JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

EXTRACTING SUMMARY FROM AN ELABORATIVE TEXT USING EXTRACTIVE APPROACH

[1]Prof Jagadish N, [2]Reetik Raj, [3]Aditya Ranjan, [4]Himanshu Malik, [5]Maithili Shetty K
[1]Assosiate Professor, [2]Student, [3]Student, [4]Student, [5]Student
Department of Information Science and Engineering
Acharya Institute of Technology, Bengaluru, India

***Abstract* — We presently live in a data-centric world, where data is produced at a rapid pace. To put things in perspective, 18 billion texts are created per day on various platforms. People today are extremely busy with their daily obligations. How to make content more condensed and accessible without sacrificing its factual meaning is the question that arises. Summarization is the process of selecting important information from a large text by carefully reading the entire text and creating a summary without losing any important details. Research papers and news articles must be summarised, which calls for subject-matter expertise. Automatic text summarization aims to automate this process by reducing long texts into shorter versions. Text summarization has advanced significantly as a result of recent improvements in deep learning methods, particularly neural networks. These models can produce summaries that are comparable to those created by humans because they have been trained on large datasets.**

Keywords— Extractive text summarization, BERT

INTRODUCTION

Owing to the vast amounts of textual data that are generated online every day from a variety of sources, including social media sites, web applications, and other information-focused programs. One must read the entire article and create a summary without losing important details to extract the essential information from the large text. Human text summarization is very laborious and time-consuming, and it also requires knowledge in that field. Due to computers' ever-increasing computational power, summarization performed by computers aims to overcome these drawbacks. There are two ways to summarise texts: one is abstractive, and the other is extractive. Abstractive text summarization produces summaries that mimic how people summarise the information by including words and phrases that don't appear in the original text. Although this method is highly desirable, it is challenging to automatically produce because it either requires multiple GPUs to train for deep learning over a long period or complicated algorithms and rules. On the other hand, extractive text summarization only uses the content from the source material and outputs a summary using the text's raw sentence structures, paragraphs, or phrases. The extractive approach is used in our text summarization implementation. Automatic extractive text summarization of lectures is a useful tool as it can extrapolate the important points without laborious manual work. Transcripts from video lectures are available in the context of many MOOCs, but it can be difficult to find the most important information from each lecture. There have been numerous attempts to address this issue as of late.

PROBLEM STATEMENT

We currently live in a very fast-paced world, where content is consumed at a rapid pace and individuals are very occupied in their day-to-day chores. With the abundance of information available in various forms, it can be challenging for individuals to sift through and extract the most relevant and important information from lengthy documents or articles.
This creates a need for developers to come up with solutions that can make content more easily accessible and consumable in a shorter time frame without compromising on its meaning and quality. In other words, there is a growing demand for efficient and effective content summarization techniques that can extract the essential information from a document and present it in a concise and easily understandable format.
Developers need to take into account the specific needs and preferences of their target audience when designing such tools. The challenge lies in striking a balance between summarizing the content sufficiently and ensuring that the key information is not lost or distorted in the process. Thus, developers must use robust algorithms and natural language processing techniques to accurately identify and extract the most important information, and present it in a format that is easily consumable and accessible to their audience.

HISTORY

The history of text summarization can be traced back to the early days of information retrieval and natural language processing.

In the 1950s and 1960s, researchers began exploring automatic summarization techniques. One of the early approaches was based on the extraction of key sentences or phrases from a text. This method involved identifying important sentences based on features like word frequency, position in the document, and grammatical structure. The selected sentences were then combined to create a summary.

In the 1990s, with the rise of the internet and the increasing volume of information available, the need for automated text summarization became more prominent. The research focused on developing more advanced algorithms and techniques for summarization.

One significant development was the emergence of statistical and machine learning-based approaches. These methods employed algorithms to analyze large collections of documents and learn patterns from them. They used techniques such as clustering, classification, and ranking to identify important sentences and generate summaries.

Another approach to text summarization was based on natural language processing and computational linguistics. These methods aim to understand the meaning of the text and generate summaries that capture the essence of the original content. They employ techniques such as semantic analysis, syntactic parsing, and discourse analysis to extract important information and generate coherent summaries.

In recent years, advancements in deep learning and neural networks have had a significant impact on text summarization. Techniques like recurrent neural networks (RNNs). LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) architecture that has been widely used in text summarization tasks. Its history can be traced back to the development of RNNs and the need for capturing long-term dependencies in sequential data.

The concept of RNNs was introduced in the 1980s as a way to process sequential data by maintaining an internal state or memory. However, traditional RNNs faced the problem of vanishing or exploding gradients, which made them difficult to train effectively on long sequences.

In 1997, Sepp Hochreiter and Jürgen Schmidhuber proposed LSTM as a solution to the vanishing gradient problem in RNNs. The LSTM architecture introduced the idea of memory cells, which allow the network to selectively remember or forget information over time. The memory cells, along with various gates, including the input gate, forget gate, and output gate, enable LSTM to capture and retain long-term dependencies in sequential data.

LSTMs gained significant attention and popularity in the field of text summarization due to their ability to handle the contextual information in text sequences effectively. Summarization involves understanding the meaning of a document and generating a concise summary while preserving the most important information.

BERT (Bidirectional Encoder Representations from Transformers) is a natural language processing model that was introduced by Google in 2018. It represents a significant milestone in the field of language understanding and has had a profound impact on various NLP tasks.

Before BERT, NLP models relied on a unidirectional approach, processing text in one direction, such as left-to-right or right-to-left. However, this limited their ability to understand the context of a word based on both preceding and following words. BERT addressed this limitation by introducing a bidirectional approach, allowing the model to capture the context from both directions. BERT was introduced by Google in 2018 as a pre-training technique for language understanding.

The development of BERT was motivated by the idea of pre-training and fine-tuning. During the pre-training phase, BERT is trained on a large corpus of unlabeled text, such as books, Wikipedia articles, and web pages. It learns to predict missing words in sentences by considering the surrounding words on both sides. This pre-training process allows BERT to learn a rich representation of language and capture various linguistic patterns.

After pre-training, BERT can be fine-tuned on specific downstream tasks, including text summarization. Fine-tuning involves training BERT on smaller labeled datasets that are specific to the target task. In the case of text summarization, the model can be fine-tuned on summarization datasets, where it learns to generate summaries based on the encoded representations of input documents.

Today, text summarization finds applications in various domains, including news aggregation, document summarization, social media analysis, and content curation. Researchers continue to work on improving the accuracy, coherence, and efficiency of text summarization systems, exploring new approaches and techniques to address the challenges of summarizing diverse types of texts.

OBJECTIVES

The main objective of text summarization is to condense a long piece of text into a shorter form while still conveying the main ideas and key points of the original content. There are several specific objectives that text summarization can achieve, including:

1. **Saving time**: Text summarization can help people quickly understand the main points of a long piece of text, reducing the time and effort required to read and comprehend the entire document. □

2. **Improving information retrieval**: Text summarization can be used to create a condensed version of a document, making it easier to search and find specific information within the text. □

3. **Reducing the amount of text that needs to be read**: Text summarization can help people quickly understand the main points of a long piece of text, reducing the number of text that needs to be read to get a general understanding of the content. □

4. **Improving clarity and understanding**: Text summarization can help to clarify and simplify complex or technical information, making it easier to understand and remember.

METHODOLOGY

The extractive approach to text summarization involves selecting a subset of sentences or phrases from the original document and combining them to create a summary. The main steps involved in this methodology are as follows:

1. **Input**: Depending on the input format, there are a variety of techniques that can be used to convert input in different formats, such as text, speech, textual image, or video, into text for text summarization. The following strategies are listed for each input format:

Text input: When using text input, the summarization system can receive the input text directly. The implementation allows for the upload of a PDF file. The text's contents are read and then passed for tokenization using the help of the PyPDF2 python library.

Speech input: Speech recognition technology can use the input audio to convert spoken words into text. The subsequent layers can then receive the text output. This can be done using speech recognition APIs such as Google Cloud Speech-to-Text or Amazon Transcribe. However, we encountered an issue, the transcribed text does not have any punctuation. Punctuation is required at the end of sentences for the text to be tokenized. deepmultilingualpunctuation, a Python package for deep multilingual punctuation prediction, is utilized to introduce punctuation.

Video input: The vast majority of people watch videos on YouTube. To summarize these videos, we need to get the audio transcribed into text. For videos posted on YouTube, captions are automatically generated using speech recognition technology. The youtube_transcript_api package can be used to parse YouTube captions. This package outputs transcribed text and requires the YouTube video ID as input. Similar problems exist with this input method as with speech. While the majority of YouTube captions lack punctuation, some of them do. The same package for deep multilingual punctuation prediction is used to incorporate punctuation into the text.

2. **Text Preprocessing**: BERT uses special tokens [CLS] and [SEP] to understand input properly. The text always begins with the [CLS] token, which is unique to classification tasks. To distinguish between two sentences, a [SEP] token is used at the end of the sentence. Even if we only have one sentence, both tokens are always
necessary.

3. **Tokenization**: BERT accepts numerical values rather than working with character arrays. Hence we require to convert words into tokens. The tokens are then passed into the BERT model, which produces a sequence of contextualized embeddings for each token.

The following operations are necessary to produce tokens and prepare the text for the BERT model:

Token embedding: Each word within a sentence is represented as a token. Larger words could be fragmented into several shorter words or characters. This is because Bert's vocabulary is fixed with a size of ~30K tokens. Words that are not part of the vocabulary are represented as subwords and characters. When given a sentence, the tokenizer will decide whether to keep every word as a single unit or break it up into smaller units.

Sentence Embedding: Each word or subword token is then mapped to its corresponding vocabulary index, which is an integer value that represents its position in the vocabulary. The vocabulary indices in BERT are important because they enable the model to efficiently look up the embedding vector for each word or subword in the input text.

Positional Embedding: It is used to encode the position of each token in the input sequence, which enables the model to capture the temporal relationship between tokens and generate contextualized representations. Without position embedding, the summarization algorithm may have difficulty distinguishing between words that have the same spelling but different meanings, depending on their position in the input text.

4. **BERT Encoder layer**: Each transformer block in the encoder layer has two sub-layers: a multi-head self-attention layer and a position-wise feed-forward layer. The encoder layer is made up of a stack of identical transformer blocks. It can have either 12 layers or 24 layers. Each transformer block contains a multi-head self-attention layer that enables the model to focus on various portions of the input sequence and identify the relationships between various tokens. To do this, each token's attention score is calculated

depending on how similar the tokens in the sequence are to one another. As a result, the model might give tokens that are more pertinent to the current context greater weights while giving irrelevant or redundant tokens lower weights. The feed-forward layer in each transformer block applies a non-linear transformation to each token's embedding to capture higher-order interactions between tokens. It does this by applying a neural network to each position in the sequence independently. The output of each transformer block is a sequence of contextualized embeddings that capture the meaning and context of each token in the input sequence.

5. **Mean of Sentence Embedding list**: It creates a single vector by dividing the total number of word vectors in a phrase by the word count. The complete statement is represented by this vector

6. **KMeans**: K-means is a popular clustering algorithm that can be used to cluster similar text data together. K-means clustering aims to partition a set of data points into a fixed number of clusters, where each data point belongs to the cluster whose mean (centroid) is closest to it. The KMeans algorithm receives its input from the mean of the cluster embedding list. The number of clusters may be predetermined or may vary according to the number of sentences in the text.

7. **Summary Generation**: The closest sentences to the cluster centroid are chosen to be a part of the summary. The chosen sentence must be arranged so that it flows similarly to how the original text does. Upon successful rearrangement using the sentence index number, sentences need to be concatenated to generate a summary.

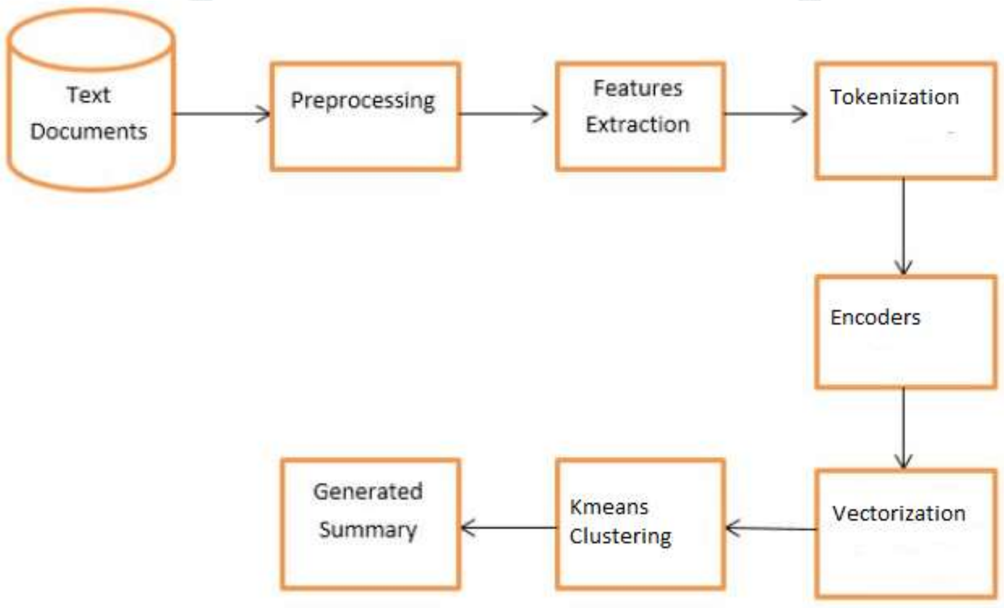
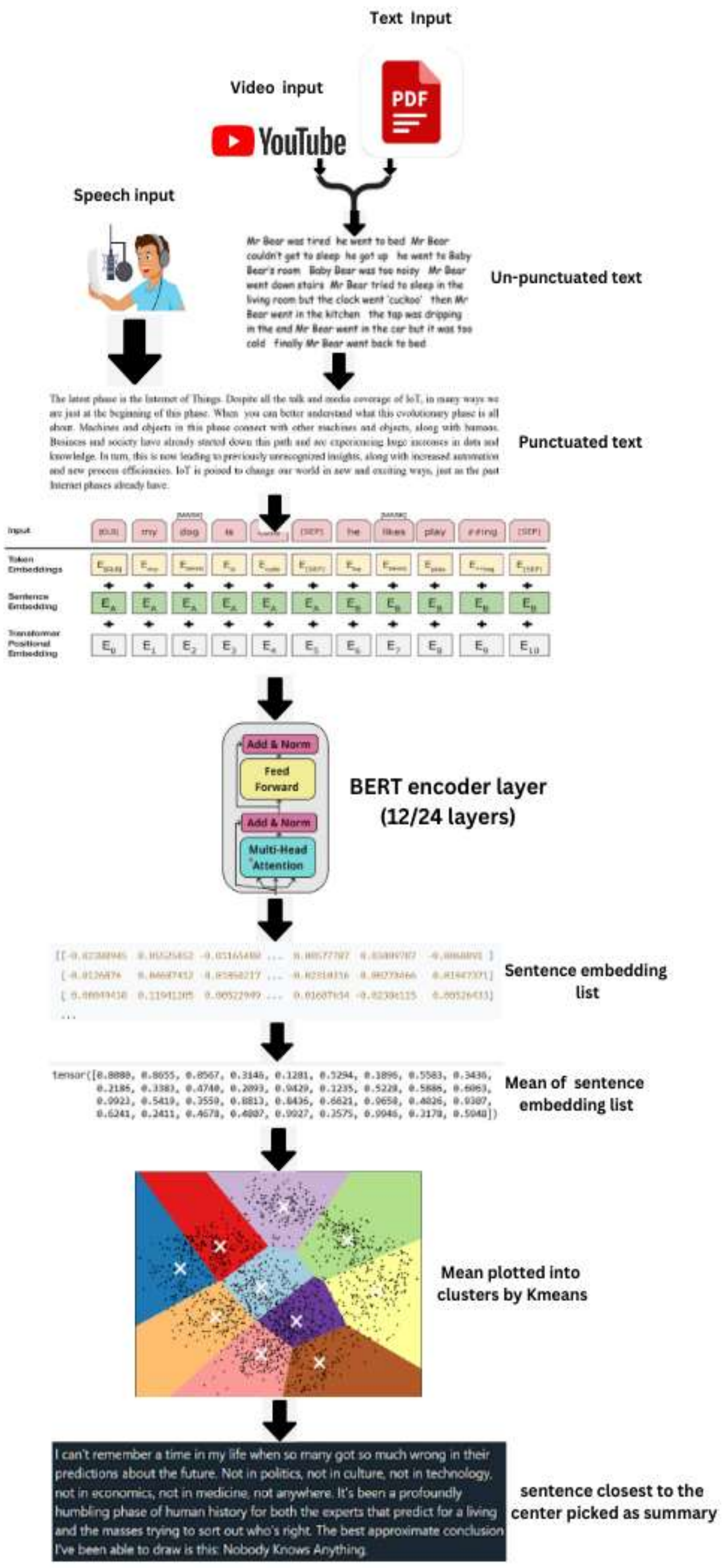


Fig 1. Block Diagram



Mr Bear was tired he went to bed Mr Bear couldn't get to sleep he got up he went to Baby Bear's room Baby Bear was too noisy Mr Bear went down stairs Mr Bear tried to sleep in the living room but the clock went 'cuckoo' then Mr Bear went in the kitchen the tap was dripping in the end Mr Bear went in the car but it was too cold finally Mr Bear went back to bed

The latest phase is the Internet of Things. Despite all the talk and media coverage of IoT, in many ways we are just at the beginning of this phase. When you can better understand what this evolutionary phase is all about. Machines and objects in this phase connect with other machines and objects, along with humans. Business and society have already started down this path and are experiencing huge increases in data and knowledge. In turn, this is now leading to previously unrecognized insights, along with increased automation and new process efficiencies. IoT is poised to change our world in new and exciting ways, just as the past Internet phases already have.

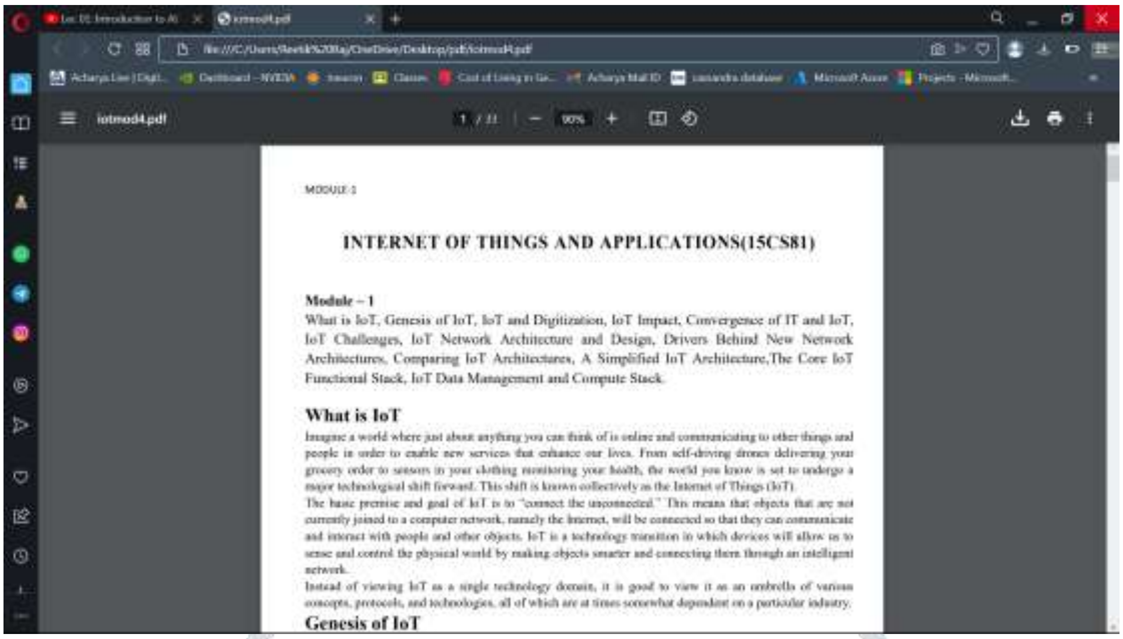
```
[[-0.02380945  0.00525852 -0.03165480 ...  0.00577787  0.03809787  -0.0068891 ]
 [-0.0126876   0.0668743  0.05868217 ... -0.02310316  0.0027366   0.0144737]]
 [ 0.00049438  0.11941205  0.00522949 ...  0.01607634 -0.02306115   0.00526433]]
 ...
```

```
tensor([0.8080, 0.8655, 0.8567, 0.3146, 0.1281, 0.5294, 0.1896, 0.5583, 0.3436,
        0.2186, 0.3383, 0.4740, 0.2893, 0.9429, 0.1235, 0.5228, 0.5886, 0.6063,
        0.0923, 0.5419, 0.3550, 0.8813, 0.8436, 0.6621, 0.0658, 0.4026, 0.9307,
        0.6241, 0.2411, 0.4678, 0.4807, 0.9927, 0.3575, 0.9946, 0.3178, 0.5048])
```

I can't remember a time in my life when so many got so much wrong in their predictions about the future. Not in politics, not in culture, not in technology, not in economics, not in medicine, not anywhere. It's been a profoundly humbling phase of human history for both the experts that predict for a living and the masses trying to sort out who's right. The best approximate conclusion I've been able to draw is this: Nobody Knows Anything.

Fig 2. System architecture

SNAPSHOTS



MODULE-1

INTERNET OF THINGS AND APPLICATIONS(15CS81)

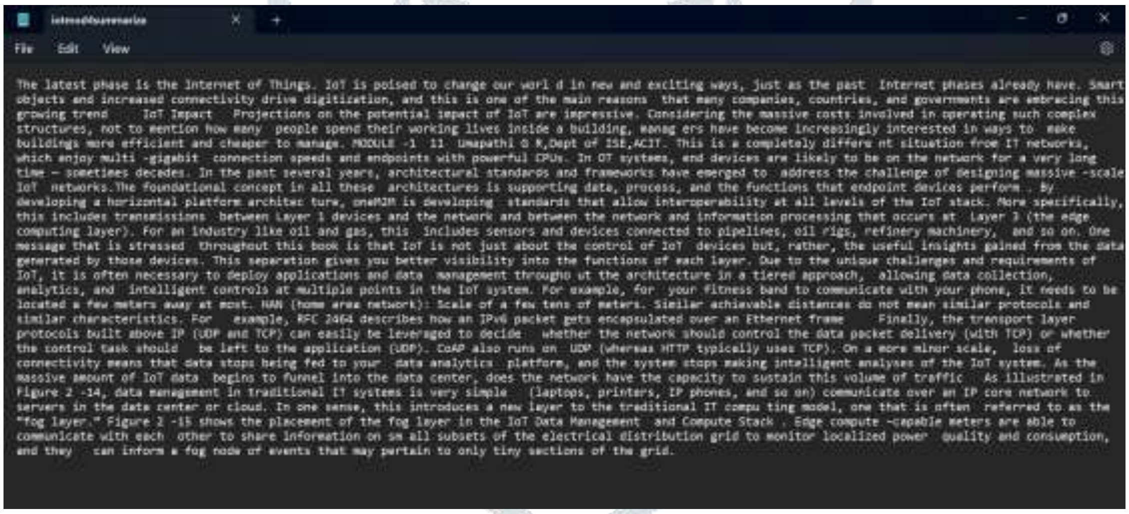
Module – 1
What is IoT, Genesis of IoT, IoT and Digitization, IoT Impact, Convergence of IT and IoT, IoT Challenges, IoT Network Architecture and Design, Drivers Behind New Network Architectures, Comparing IoT Architectures, A Simplified IoT Architecture,The Core IoT Functional Stack, IoT Data Management and Compute Stack.

What is IoT
Imagine a world where just about anything you can think of is online and communicating to other things and people in order to enable new services that enhance our lives. From self-driving drones delivering your grocery order to sensors in your clothing monitoring your health, the world you know is set to undergo a major technological shift forward. This shift is known collectively as the Internet of Things (IoT).
The basic premise and goal of IoT is to "connect the unconnected." This means that objects that are not currently joined to a computer network, namely the Internet, will be connected so that they can communicate and interact with people and other objects. IoT is a technology transition in which devices will allow us to sense and control the physical world by making objects smarter and connecting them through an intelligent network.
Instead of viewing IoT as a single technology domain, it is good to view it as an umbrella of various concepts, protocols, and technologies, all of which are at times somewhat dependent on a particular industry.

Genesis of IoT

Fig 3. Summarising a PDF file

This is a PDF file that talks about the Internet of Things, its impact, and some of its architecture. This PDF contains 33 pages of textual information.



```
The latest phase is the Internet of Things. IoT is poised to change our worl d in new and exciting ways, just as the past  Internet phases already have. Smart
objects and increased connectivity drive digitization, and this is one of the main reasons  that many companies, countries, and governments are embracing this
growing trend     IoT Impact   Projections on the potential impact of IoT are impressive. Considering the massive costs involved in operating such complex
structures, not to mention how many  people spend their working lives inside a building, manag ers have become increasingly interested in ways to  make
buildings more efficient and cheaper to manage. MODULE -1  11  Umapathi G R,Dept of ISE,ACIT. This is a completely differe nt situation from IT networks,
which enjoy multi -gigabit  connection speeds and endpoints with powerful CPUs. In OT systems, end devices are likely to be on the network for a very long
time — sometimes decades. In the past several years, architectural standards and frameworks have emerged to  address the challenge of designing massive -scale
IoT  networks.The foundational concept in all these  architectures is supporting data, process, and the functions that endpoint devices perform . By
developing a horizontal platform architec ture, oneM2M is developing  standards that allow interoperability at all levels of the IoT stack. More specifically,
this includes transmissions  between Layer 1 devices and the network and between the network and information processing that occurs at  Layer 3 (the edge
computing layer). For an industry like oil and gas, this  includes sensors and devices connected to pipelines, oil rigs, refinery machinery,  and so on. One
message that is stressed  throughout this book is that IoT is not just about the control of IoT  devices but, rather, the useful insights gained from the data
generated by those devices. This separation gives you better visibility into the functions of each layer. Due to the unique challenges and requirements of
IoT, it is often necessary to deploy applications and data  management througho ut the architecture in a tiered approach,  allowing data collection,
analytics, and  intelligent controls at multiple points in the IoT system. For example, for  your fitness band to communicate with your phone, it needs to be
located a few meters away at most. HAN (home area network): Scale of a few tens of meters. Similar achievable distances do not mean similar protocols and
similar characteristics. For   example, RFC 2464 describes how an IPv6 packet gets encapsulated over an Ethernet frame     Finally, the transport layer
protocols built above IP (UDP and TCP) can easily be leveraged to decide   whether the network should control the data packet delivery (with TCP) or whether
the control task should   be left to the application (UDP). CoAP also runs on  UDP (whereas HTTP typically uses TCP). On a more minor scale,  loss of
connectivity means that data stops being fed to your  data analytics  platform, and the system stops making intelligent analyses of the IoT system. As the
massive amount of IoT data  begins to funnel into the data center, does the network have the capacity to sustain this volume of traffic   As illustrated in
Figure 2 -14, data management in traditional IT systems is very simple   (laptops, printers, IP phones, and so on) communicate over an IP core network to
servers in the data center or cloud. In one sense, this introduces a new layer to the traditional IT compu ting model, one that is often  referred to as the
"fog layer." Figure 2 -15 shows the placement of the fog layer in the IoT Data Management  and Compute Stack . Edge compute -capable meters are able to
communicate with each  other to share information on sm all subsets of the electrical distribution grid to monitor localized power  quality and consumption,
and they   can inform a fog node of events that may pertain to only tiny sections of the grid.
```

Fig 4. Summarised PDF file

The model has summarized 33 pages of a PDF file into 30 sentences approximately.

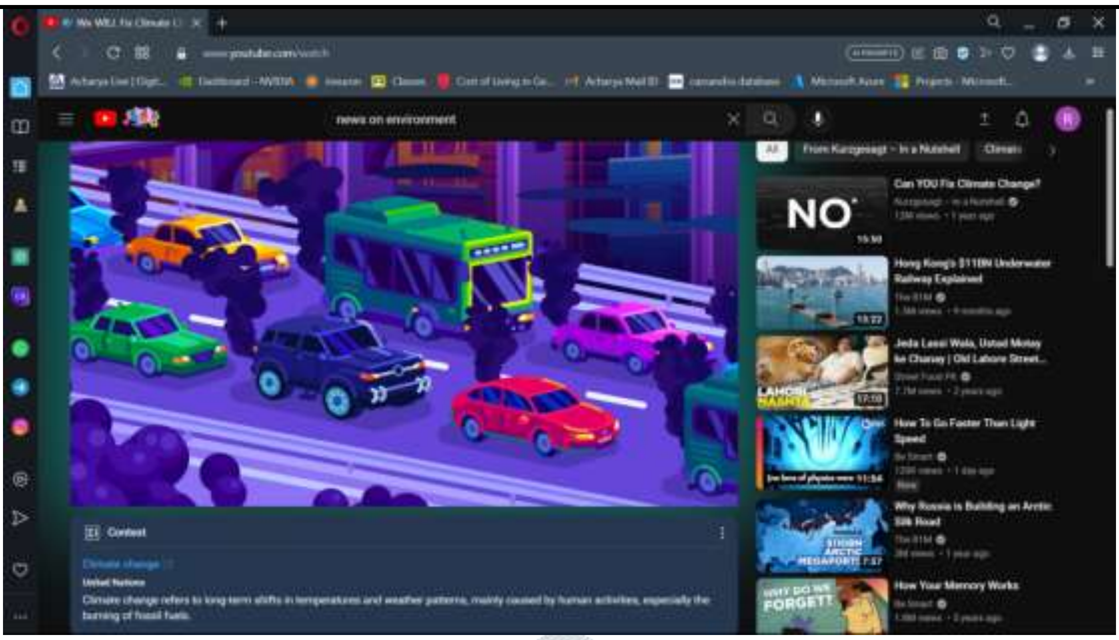
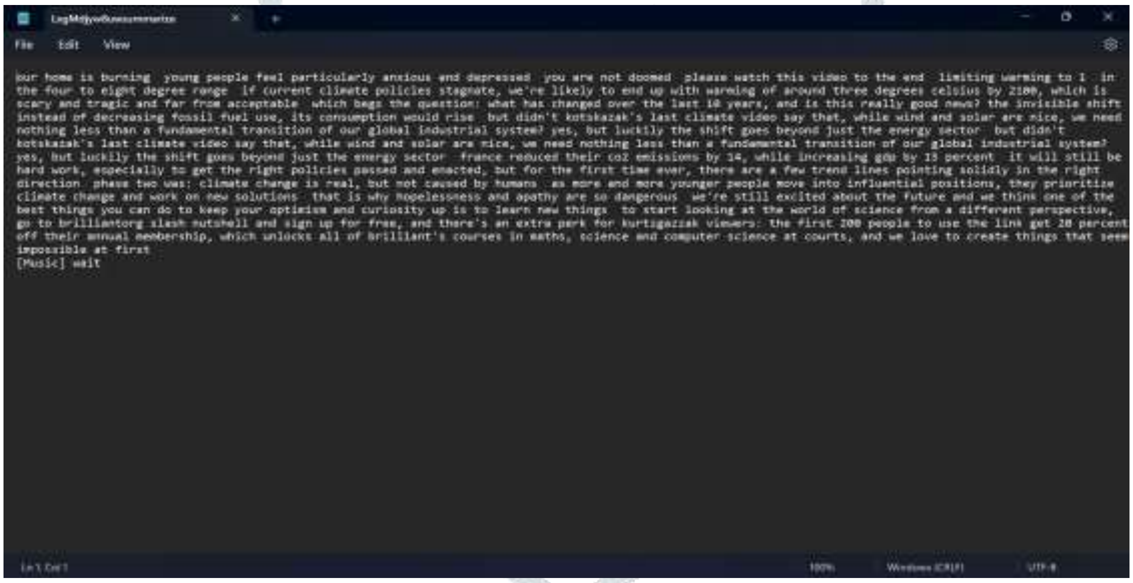

news on environment
NO
Can YOU Fix Climate Change?
Hong Kong's $11BN Underwater Railway Explained
How To Go Faster Than Light Speed
Why Russia is Building an Arctic Silk Road
How Your Memory Works
Context
Climate change
United Nations
Climate change refers to long term shifts in temperatures and weather patterns, mainly caused by human activities, especially the burning of fossil fuels.

Fig 5. Summarising a YouTube video

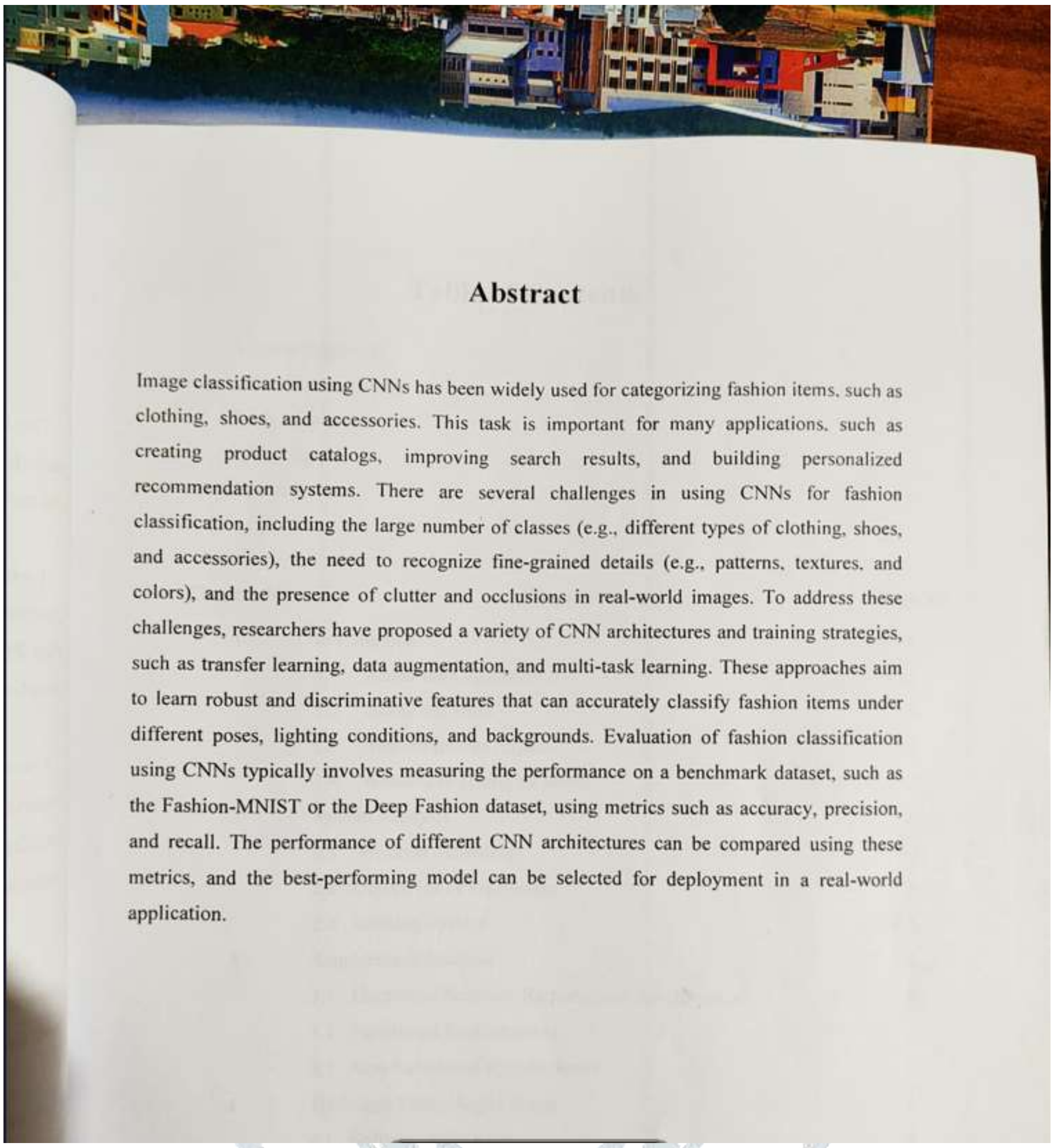
To summarize a Youtube video we pass the video ID which is "LxgMdjyw8uw" from its URL as an input. This video is 16 minutes and 10 seconds long.



```
our home is burning  young people feel particularly anxious and depressed  you are not doomed  please watch this video to the end  limiting warming to 1  in
the four to eight degree range  if current climate policies stagnate, we're likely to end up with warming of around three degrees celsius by 2100, which is
scary and tragic and far from acceptable  which begs the question: what has changed over the last 10 years, and is this really good news? the invisible shift
instead of decreasing fossil fuel use, its consumption would rise  but didn't kotskazak's last climate video say that, while wind and solar are nice, we need
nothing less than a fundamental transition of our global industrial system? yes, but luckily the shift goes beyond just the energy sector  but didn't
kotskazak's last climate video say that, while wind and solar are nice, we need nothing less than a fundamental transition of our global industrial system?
yes, but luckily the shift goes beyond just the energy sector  france reduced their co2 emissions by 14, while increasing gdp by 15 percent  it will still be
hard work, especially to get the right policies passed and enacted, but for the first time ever, there are a few trend lines pointing solidly in the right
direction  phase two was: climate change is real, but not caused by humans  as more and more younger people move into influential positions, they prioritize
climate change and work on new solutions  that is why hopelessness and apathy are so dangerous  we're still excited about the future and we think one of the
best things you can do to keep your optimism and curiosity up is to learn new things  to start looking at the world of science from a different perspective,
go to brilliantorg slash nutshell and sign up for free, and there's an extra perk for kurtigazzak viewers: the first 200 people to use the link get 20 percent
off their annual membership, which unlocks all of brilliant's courses in maths, science and computer science at courts, and we love to create things that seem
impossible at first
[Music] wait
```

Fig 6. Summarised YouTube video

The YouTube video is summarized into 225 words.

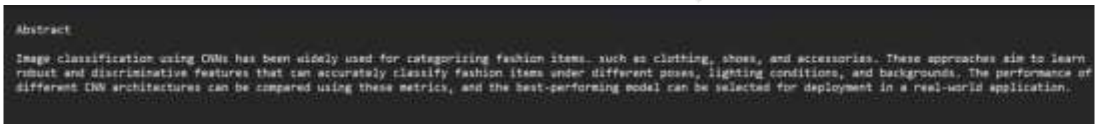


Abstract

Image classification using CNNs has been widely used for categorizing fashion items, such as clothing, shoes, and accessories. This task is important for many applications, such as creating product catalogs, improving search results, and building personalized recommendation systems. There are several challenges in using CNNs for fashion classification, including the large number of classes (e.g., different types of clothing, shoes, and accessories), the need to recognize fine-grained details (e.g., patterns, textures, and colors), and the presence of clutter and occlusions in real-world images. To address these challenges, researchers have proposed a variety of CNN architectures and training strategies, such as transfer learning, data augmentation, and multi-task learning. These approaches aim to learn robust and discriminative features that can accurately classify fashion items under different poses, lighting conditions, and backgrounds. Evaluation of fashion classification using CNNs typically involves measuring the performance on a benchmark dataset, such as the Fashion-MNIST or the Deep Fashion dataset, using metrics such as accuracy, precision, and recall. The performance of different CNN architectures can be compared using these metrics, and the best-performing model can be selected for deployment in a real-world application.

Fig 7. Summarizing textual image

It is an image in JPEG format. This image has textual information about Convolutional Neural Networks.



```
Abstract

Image classification using CNNs has been widely used for categorizing fashion items. such as clothing, shoes, and accessories. These approaches aim to learn
robust and discriminative features that can accurately classify fashion items under different poses, lighting conditions, and backgrounds. The performance of
different CNN architectures can be compared using these metrics, and the best-performing model can be selected for deployment in a real-world application.
```

Fig 8. Summarized textual image

The abstract is summarized into few lines.

RESULTS

1. **Content coverage**: The summary covers the main ideas and key information of the original text.

2. **Coherence**: The summary is coherent and organized, with sentences and paragraphs that mostly flow well.

3. **Length**: The length of the summary is appropriate for the intended purpose, with longer summaries

providing more detail and shorter summaries providing a quick overview.

4. **Information redundancy**: The summary does not contain redundant information that does not add value to the overall summary.

LIMITATIONS

1. **Lack of Coherence**: BERT-based extractive summarization, when combined with k-means clustering, may not guarantee the overall coherence and smooth flow of the generated summary. Clustering alone does not consider the contextual relationships between sentences, potentially leading to disjointed or fragmented summaries.

2. **Domain-specific adaptation**: BERT models are pre-trained on general corpora, which may not capture domain-specific knowledge effectively. Fine-tuning BERT with domain-specific datasets and incorporating domain-specific features can help improve the performance of extractive summarization in specialized domains.

3. **Scalability**: BERT, being a deep neural network model, can be computationally expensive and memory-intensive, especially for longer documents. This scalability issue can limit the feasibility of using BERT for real-time or large-scale summarization tasks.

4. **Sensitivity to Input Variations**: The effectiveness of BERT and k-means clustering in summarization can be sensitive to input variations, such as sentence order or minor changes in the input text. These variations might lead to different clustering results and subsequently impact the summary composition.

FUTURE SCOPE

The field of text summarization using extractive approaches has seen significant advancements in recent years, and there are several potential future directions and scopes for further development. Some of the potential future scopes of text summarization using extractive approaches include:

1. **Enhanced Extractive Summarization Algorithms**: Researchers can continue to develop and refine extractive summarization algorithms to improve their effectiveness and efficiency. This can involve exploring new techniques such as deep learning, natural language processing (NLP), and machine learning algorithms to better identify and select the most important sentences or phrases from a text to create a summary.

2. **Multi-Document Summarization**: Extractive summarization techniques can be extended to handle multiple documents, such as summarizing multiple news articles on a similar topic or summarizing a collection of research papers. Multi-document summarization presents unique challenges, such as identifying redundancy across documents and effectively combining information from multiple sources, and further research in this area can lead to more robust and informative summaries.

3. **Domain-specific Summarization**: Extractive summarization can be tailored for specific domains or industries, such as summarizing legal documents, scientific articles, or technical documentation. Domain-specific summarization can leverage domain-specific knowledge and language models to generate more accurate and relevant summaries that cater to the specific needs of the domain.

4. **Real-time and Streaming Summarization**: With the increasing availability of real-time data from various sources such as social media, news feeds, and live events, there is a growing need for real-time and streaming summarization techniques that can process and summarize the text in real-time as it becomes available. Further research in this area can lead to the development of efficient and scalable extractive summarization algorithms for real-time data streams.

CONCLUSION

Extractive text summarization is a promising approach for condensing long documents or text into shorter summaries by selecting and rearranging the most relevant sentences or phrases. It has been widely studied and applied in various applications, such as news summarization, document summarization, and information retrieval. Extractive summarization techniques have shown significant progress and achieved state-of-the-art performance in many cases. The extractive approach offers several advantages, including maintaining the original context and ensuring that the summary is grounded in the source text. It also requires fewer computational resources compared to other approaches like abstractive summarization. However, extractive summarization still has limitations, such as potential loss of coherence and readability, reliance on sentence extraction, and inability to generate new information. Despite these limitations, extractive summarization has a bright future with many potential scopes for further research and development. This includes exploring advanced algorithms and techniques, such as deep learning, NLP, and machine learning, to improve the effectiveness and efficiency of extractive summarization. It also involves addressing challenges like multi-document summarization, domain-specific summarization, real-time and streaming summarization, evaluation metrics, summarization for low-resource languages, explainable summarization, and summarization for different modalities.

FUTURE ENHANCEMENT

1. **Cross-lingual and multilingual text summarization**: As more and more data becomes available in multiple languages, there is a growing need for cross-lingual and multilingual summarization techniques that can summarize text in different languages. Researchers are exploring methods for adapting existing summarization techniques to work with multilingual text data.

2. **Explainable text summarization**: While neural network-based summarization techniques have achieved state-of-the-art performance, their decision-making processes are often opaque and difficult to interpret. There is growing interest in developing explainable summarization models that can provide insight into how they arrive at their summaries.

3. **Interactive text summarization**: Interactive text summarization involves enabling users to interact with the summarization process, providing feedback and input during the summarization process. This approach can improve the accuracy and relevance of summaries and is an area of active research.

4. **Multi-document summarization**: Multi-document summarization involves summarizing multiple documents on a single topic, which can be challenging because of the need to identify overlapping information and summarize it effectively. Researchers are exploring new techniques for multi-document summarization that can handle large amounts of text data.

5. **Domain-specific text summarization:** Summarization techniques that are trained on generic datasets may not perform well in domain-specific settings. Researchers are exploring methods for adapting existing summarization techniques to work with specific domains, such as scientific papers, medical reports, or legal documents

REFERENCES

[1] Bert: Pre-training of deep bidirectional transformers for language understanding Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. 2019
[2]A Survey on Automatic Text Summarization N. Nazari and M. A. Mahdavi 2019
[3]A Survey of Text Summarization Extractive Techniques Vishal Gupta, Gurpreet Lehal 2018
[4]Text Summarization Using Neural Networks Kaikhah, Khosrow 2019
[5]Text Summarization with Pretrained Encoders Yang Liu and Mirella Lapata 2021
[6]BERT: A Review of Applications in Natural Language Processing and Understanding Koroteev M.V 2020
[7]Review Paper on Extractive Text Summarization Arpita Sahoo, Dr.Ajit Kumar Nayak 2019
[8]Extractive Text Summarization of Norwegian News Articles Using BERT Thomas Indrias Biniam,

Adam Morén 2017

[9]Effective summarization method of text documents Using BERT Rasim Alguliev, Ramiz Aliguliyev 2019

[10]Text Summarization Using Neural Networks Kaikhah, Khosrow 2019

[11]Chin-Zing (2018). Generating summaries from event data. Information Processing& Management, 31(5), 735–751. https://doi.org/10.1016/0306-4573 (95)00025-C

[12]Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2019. "Introduction to the special issue on summarization". Computational linguistics 28, 4 (2019), 399–408.

[13]Hovy, E., 2019. Text Summarization. In: The Oxford Handbook of Computational Linguistics, Mitkov, R. (Ed.), OUP Oxford, Oxford, ISBN-10: 019927634X, pp: 583-598.

[14]Li, P.; Lam, W.; Bing, L.; Wang, Z. Deep Recurrent Generative Decoder for Abstractive Text Summarization.

[15]Meghan J Panicker, Vikas Upadhaya, Gunjan Sethi and Vrinda Mathur "Image caption generator"vol. 10, no. 3, Jan. 2021, ISSN:2278-3075

[16]Ali Ashraf Mohammed "Image caption", using CNN and LSTM vol. 5, no. 2, May 2020.

[17]Athul Kumar, Archi Aggarwal, K S Ashin Shanly, Sudip Das and Nidhin Harilal "Imagecaptiongenerator using siamese graph convolutional networks and LSTM" Jan. 2022, ISSN:3493700.349374

[10] Yeong-Hwa, Chang,yen-Jen , Chen, Ren-hung,haung, and yi ting yu, "enhanced Image captioning with color Recognition using deep learning methods" Jan. 2022, ISSN:3493700.34937

[11] El-Kassas, W.S.; Salama, C.R.; Rafea, A.A.; Mohamed, H.K. EdgeSumm: Graph-based framework for automatic text summarization. Inf. Process. Manag. 2020, 57, 102264.

[12.] Lin, C.Y. Rouge: A Package for Automatic Evaluation Of Summaries. In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, 25–26 July 2004; pp. 25–26.

[13]. Liu, Y.; Lapata, M. Text Summarization with Pretrained Encoders. arXiv 2019, arXiv:1908.08345.

[14]. Zhong, M.; Liu, P.; Chen, Y.; Wang, D.; Qiu, X.; Huang, X. Extractive Summarization as Text Matching. ACL Anthology. 2020. Available online: https://www.aclweb.org/anthology/2020.acl-main.552/ (accessed on 9 August 2021).

[15.] Joshi, A.; Fidalgo, E.; Alegre, E.; Fernández-Robles, L. SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. Expert Syst. Appl. 2019, 129, 200–215. [CrossRef]

[16]. Sun, S.; Cheng, Y.; Gan, Z.; Liu, J. Patient knowledge distillation for Bert model compression. arXiv 2019, arXiv:1908.09355.

[17]. Iandola, F.N.; Shaw, A.E.; Krishna, R.; Keutzer, K.W. SqueezeBERT: What Can Computer Vision Teach NLP about Efficient Neural Networks? arXiv 2020, arXiv:2006.11316.

[18]. VSanh, i.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper, and lighter. arXiv 2020, arXiv:1910.0110v4.

[19]. Dong, Y.; Shen, Y.; Crawford, E.; van Hoof, H.; Cheung, J.C.K. Banditsum: Extractive summarization as a contextual bandit. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium,31 October–4 November 2018.