**Name:** Siddharth Pal

**Course:** AIML

**Roll No**: 25

Title: **“Text Summarization”**

**Email**: [siddharthpal819@gmail.com](mailto:siddharthpal819@gmail.com)

Department of AIML & Data Science.

School of Emerging Science and Technology

Dissertation Submitted for the partial fulfillment of the **B.Sc. as a part of M.Sc. (Integrated) Five Years Program AIML/Data Science** degree to the Department of AIML & Data Science.

**Project Dissertation**

**Text Summarization**

submitted to

****

By

**Siddharth Pal**

**Semester-VI**

**M.Sc. (Integrated) Five Years Program AIML/Data Science**

Department of AIML & Data Science.

School of Emerging Science and Technology

Gujarat University

**June ,2022**

**DECLARATION**

This is to certify that the research work reported in this dissertation entitled

“**Text Summarization**” for the partial fulfilment of B.Sc. as a part of M.Sc. (Integrated) in Artificial Intelligence and Machine Learning/Data Science degree is the result of investigation done by myself.

**Siddharth Pal**

|  |  |
| --- | --- |
| Place: Ahmedabad | Name of Student |
| Date: |  |

**ACKNOWLEGEMENT**

On this occasion of submitting my project work, I would like to thank all the people who have made this possible. The role of Department of AIML, Gujarat University in shaping my professional capability is very important.

I thank Dr. Ravi Gor, coordinator of the Department of AIML, Data Science & Actuarial Science, for all his support.

Rashmi Ma’am my guide has always guided me on the right path and has been a motivating source for innovative research work. This work would have not been possible without her guidance, support, and encouragement. With the help of her guidance, I magnificently overcame the difficulties during my work and learned at each step.

I am heartily and loving thankful to my parents for their sacrifice, support and endless love which encouraged me to efficiently overcome the difficulties in my pursuit of the project and who always helped me in my work.

Last but not the least, I would not be able to complete the work without the blessings of almighty and would like to thank God for giving me strength and patience to carry out my work with full of dedication.

**-Siddharth Pal**

**Index**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Content** | **Pg no.**     |  | | --- | |  | |  | |  | |  | |  | |  | |  | |
| 1 | Abstract & Key Words | 5 |
| 2 | Introduction | 7 |
| 3 | Methodology | 11 |
| 4 | Data Analysis | 13 |
| 5 | Model Explanation | 16 |
| 6 | Result | 22 |
| 7 | Bibliography | 26 |
|  |
|  |
|  |  |  |

**Chapter 1**

**Abstract & Key Word**

**Abstract**

The amount of text data available online is increasing at a very fast pace hence text summarization has become essential. Most of the modern recommender and text classification systems require going through a huge amount of data. Manually generating precise and fluent summaries of lengthy articles is a very tiresome and time-consuming task. Hence generating automated summaries for the data and using it to train machine learning models will make these models space and time efficient. Extractive summarization and abstractive summarization are two separate methods of generating summaries. The extractive technique identifies the relevant sentences from the original document and extracts only those from the text. Whereas in abstractive summarization techniques, the summary is generated after interpreting the original text, hence making it more complicated. In this paper, we will be presenting a comprehensive comparison of a few transformers’ architecture based pre-trained models for text summarization. For analysis and comparison, we have used the BBC news dataset that contains text data that can be used for summarization and human generated summaries for evaluating and comparing the summaries generated by machine learning models.

Abstractive text summarization has achieved success in switching from linear models via sparse and handcrafted features to nonlinear neural network models via dense inputs. This success comes from the application of deep learning models on natural language processing tasks where these models are capable of modelling intricate patterns in data without handcrafted features. In this work, the text summarization problem has been explored using Sequence-to-sequence recurrent neural networks and Transfer Learning with a Unified Text-to-Text Transformer approaches. Experimental results showed that the Transfer Learning-based model achieved considerable improvement for abstractive text summarization.

**Keywords:** News Summarization, NLP, Abstractive Summarization , Deep Learning , Transformers

**Chapter 2**

**Introduction**

**Introduction**

Summarization is closely related to data compression and information understanding both of which are key to information science and retrieval. The technology of text summarization can improve information extraction systems and also allows readers to quickly view a large number of documents for important information. Indeed, automatic summarization has been recently recognized as one of the most important natural language processing (NLP) tasks, yet one of the least solved one. In the literature, there are two main approaches to text summarization. While extractive methods are arguably well suited for identifying the most relevant information, such techniques may lack the fluency and coherency of human-generated summaries.

Abstractive text summarization is the task of generating a summary consisting of a few sentences that capture the salient ideas of the input text document. The adjective ‘abstractive’ is used to denote a summary that is not a mere selection of a few existing passages or sentences extracted from the source, but a compressed paraphrasing of the main contents of the document, potentially using vocabulary unseen in the source document. Abstractive summarization has shown the most promise towards addressing issues in extracting important information from the text documents, but Abstractive generation may produce sentences not seen in the original input document. Motivated by neural network success in machine translation experiments, the attention-based encoder-decoder paradigm has recently been widely studied in abstractive summarization. By dynamically accessing the relevant pieces of information based on the hidden states of the decoder during the generation of the output sequence, the model revisits the input and attends to important information. Recent abstractive document summarization models are yet not able to achieve convincing performance. In this paper, we investigate the Transfer learning for abstractive text summarization to address a key challenge in summarization, which is to optimally compress the original document while preserving the key concepts in the original document. The rest of this paper is organized as follows: provides an overview of the existing works and approaches. The approach to be investigated is introduced. Experimental setting, data sets used and results.

* **There are two approaches for text summarization:**

1. **Extractive summarization**: In extractive summarization, most important phrases or sentences from the text are identified and selected based on a score that is computed depending on the words in that sentence.

Graphical user interface, text, application, email

Description automatically generated

1. **Abstractive Summarization**: Abstractive summarizers do not select sentences from the originally given text passage to create the summary. Instead, they produce a paraphrasing of the main **contents** of the given text, using a vocabulary set different from the original document. This is very similar to what we as humans do, to summarize.

**Graphical user interface, text, application

Description automatically generated**

* Here, abstractive summarization approach has been used to summarize the text.

**Objective**

To summarize the given article provided by the user with the help of NLP and deep learning techniques.

**Need For Text Summarization**

Automating summarization eliminates manual efforts. Shorter texts, which are summaries of longer texts, would reduce reading time. With the ever-growing amount of data, text summarization would reduce the size of files and hence solve the problem of storage. A shorter text or summary would provide more significant insights. Moreover, accurate summaries are very useful when it comes to text mining and data analysis

**Chapter – 3**

**Methodology**

**3.1 Dataset**

The CNN / Daily-Mail Dataset is an English-language dataset containing just over 300k unique news articles as written by journalists at CNN and the Daily Mail. The current version supports both extractive and abstractive summarization, though the original version was created for machine reading and comprehension and abstractive question answering.

The average token count for the articles and the highlights is

Article: 786

Highlights: 56

The CNN/Daily Mail dataset splits are as follows:

* Train Set: 287,113
* Validation Set: 13,368
* Test set: 11,490
* **ID**:  a string containing the heximal formatted SHA1 hash of the URL where the story was retrieved from
* **Article**: a string containing the body of the news article
* **Highlights**: a string containing the highlight of the article as written by the article author

**Chapter – 4**

**Data Analysis**

**4.1 Data Pre-processing**

* 5000 samples have been used to train the model (Random sampling has been used)
* **Lower casing**: - To convert the input text into the same casing format so that all capital, lower case and mixed case are treated similarly.
* **Eliminate Punctuation**: - HTML tags and links- Removal of punctuations, links and tags that do not add meaning to the text such as “!"#$%&\'() \*+,-./:;<=>?@[\\]^\_{|}~`” to standardize the text.
* **Eliminate Stop words and frequently occurring** **words**: - Removal of common words such as ‘the’, ‘a’, etc. that are frequently used in a text but do not provide valuable information for downstream analysis.
* **Stemming**: - Reducing the inflected words to their root form.
* **Lemmatization**: - Reducing derived words to their base or root form while making sure that root words belong to the language.
* **Checking the Data distribution of the Headlines and the article set**

**Chart, histogram

Description automatically generated**

**Headlines having length in range [0,150]: 99.84 %**

**Text having length in range [0, 1100]: 89.69 %**

**Mean headline length: 51.30**

**Mean text length: 694.30**

**Chapter – 5**

**Model Explanation**

**4.1 The Transformer Model**

It is possible to formulate most NLP tasks in a “text-to-text” format – that is, a task where the model is fed some text for context or conditioning and is then asked to produce some output text. This approach provides a consistent training objective both for pre-training and fine-tuning. Specifically, the model is trained with a maximum likelihood objective regardless of the task.

**4.2 The Transformer: Model Architecture**

Most competitive and successful neural sequence transduction models have an encoder-decoder structure [14, 11]. Here, the encoder maps an input sequence of symbol representations (x1,..., xn) to a sequence of continuous representations z = (z1,..., zn) [14]. Given z, the decoder then generates an output sequence (y1,..., ym) of symbols one element at a time. At each step, the model is automatically regressive, with the previously generated symbols being consumed as additional input when generating the next step. The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder,

**Encoder**: The encoder is composed of a stack of N = 6 identical layers. Each layer has a multi-head self-attention mechanism, and a simple, position-wise fully connected feed-forward network. A residual connection is employed around each of the two sub-layers followed by layer normalization. That is, the output of each sublayer is Layer Norm(x+Sublayer(x)) Where Sublayer(x) is the function implemented by the sub-layer itself.

**Decoder**: The decoder also consists of a stack of N = 6 identical layers. The decoder inserts a third sub-layer which, in addition to the two sub-layers, provides multiheaded attention to the output of the encoder stack. Similar to the encoder, a residual connection around each of the two sub-layers is used, followed by a layer normalization. To prevent positions from paying attention to subsequent positions, a modified self-attention sub-layer is used in the decoder.

**Attention**: An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, the keys, the values and the output are all vectors. The output can be calculated as a weighted sum of the values, where the weight assigned to each value is calculated by a compatibility function of the query with the corresponding key. The advantage of using multi-head attention allows the model to share information from different representation subspaces at different positions. With a single attention head this is prevented by averaging. The Transformer uses multi-head attention in the following manner:

• In “encoder-decoder attention” layers, the queries come from the previous decoder layer and the memory keys and values come from the output of the encoder. This allows every position in the decoder to attend over all positions in the input sequence [15, 2].

Diagram

Description automatically generated

• The encoder contains self-attention layers. In a self-attention layer, all keys, values and queries come from the same location, in this case from the output of the previous layer in the encoder. Each position in the encoder can attend to all positions in the previous layer of the encoder.

• Similarly, self-attention layers in the decoder allow each position in the decoder to attend to all positions in the decoder up to and including that position.

**T5 Approach**

**Attention Masks:** A major distinguishing factor for different architectures is the “mask” used by different attention mechanisms in the model. Recall that the self-attention operation in a Transformer takes a sequence as input and outputs a new sequence of the same length [10]. Each entry of the output sequence is produced by computing a weighted average of entries of the input sequence. Specifically, let yi refer to the i-th element of the output sequence and xj refer to the j th entry of the input sequence.

Diagram

Description automatically generated

in practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. We compute the matrix of outputs as:

Where wi, j is the scalar weight produced by the self-attention mechanism as a function of xi and xj . The attention mask is then used to zero out certain weights in order to constrain which entries of the input can be attended to at a given output time step.

**Encoder-Decoder**: An encoder-decoder Transformer consists of two layers of stacks: the encoder, which is fed an input sequence, and the decoder, which generates a new output sequence. The encoder uses a “fully visible” attention mask. The “fully visible” masking allows a self-attention mechanism to pay attention to each input of its output. This form of masking is suitable when the attention is over a “prefix”, i.e., a context that is provided to the model that will later be used to make predictions. The self-attention operations in the decoder of the transformer use a “causal” masking pattern. Within model training process, approaching with "causal" mask let decoder prevent the model from attending to the j th entry during handling i th input sequence for j > i. This is used during training so that the model cannot “see into the future” while producing its output

**Sequence to Sequence Model**

The Recurrent Neural Network (RNN) is a natural generalization of feed forward neural networks to sequences. Given a sequence of inputs (x1…,xT ), a standard RNN computes a sequence of outputs (y1,...,yT ):

Text

Description automatically generated

The RNN can easily map sequences to sequences whenever the alignment between the inputs and the outputs is known ahead of time. However, it is not clear how to apply an RNN to problems whose input and the output sequences have different lengths with complicated and non-monotonic relationships.

Sequence learning consists of mapping the input sequence with one RNN to a vector of fixed size and then mapping the vector with another RNN to the target sequence. Although it could work in principle, since the RNN is supplied with all relevant information, it would be difficult to train the RNNs due to the resulting long-term dependencies. However, the Long Short-Term Memory (LSTM) is known to learn problems with long-range time dependencies, so an LSTM can be successful in this setting.

The objective of the LSTM is to estimate the conditional probability p(y1,..., yM0|x1,..., xM) where (x1,..., xM) is an input sequence and (y1,..., yM0) is its corresponding output sequence whose length M0 may differ from M. The LSTM computes the conditional probability by first obtaining the fixed-dimensional representation v of the input sequence (x1..., xM) given by the last hidden state of the LSTM, and then computing the probability of (y1,… yM0) with a standard LSTM language model formulation whose initial hidden state is set to the representation v of (x1,..., xT ):

**Text

Description automatically generated**

In this equation, each p(ym|v, y1,..., ym−1) distribution is represented with a soft max over all the words in the vocabulary. The LSTM formulation from Graves has been used. It is require that each sentence ends with a special end-of-sentence symbol “”, which enables the model to define a distribution over sequences of all possible lengths.

**Chapter - 6**

**Results**

**Article Given:** Jailed: Joshua Sadler, 21, was sentenced to 12 months after admitting dangerous driving. The mother of a teenager who was killed when a friend crashed his car just four days after passing his test has spoken of her shock after the driver was jailed for only 1 2 months. Joshua Sadler, 21, lost control of his silver Renault Clio on a 60mph country road and smashed into a tree, killing from 1 seat passenger Mikey Maguire, 19. Sadler had previously been convicted of a string of serious motoring offences and twice ban ed from the road before he even had a licence. He was seen on the night of the fatal crash performing handbrake turns and showed g off in a car park. Sadler pleaded guilty to causing death by careless driving and was sentenced to 12 months in jail and banned him from driving for five years. Afterwards Mr Maguire's mother Allison Jarman, 43, described the sentence as an Insult She said: When I was in court my head was in my hands and at one point, I couldn't even listen to what was happening. I expected th e sentence to be low because I have been doing a lot of research online but when I heard it was 12 months it felt like a slop the face. "Nothing can bring Mikey back but 12 months for my son's life is shocking. I was speechless; he will be out in three months. Mikey and Joshua had known each other for years but I was under the impression that they weren't close any a ore. He had always been a bit of a troublemaker and had previous convictions for driving offences. I always said he would and up hurting someone one day. I just didn't expect it to be my son. The fatal crash happened on Hun tick Road, Lytchett Matravers near Poole, Dorset, on February 8 last year. Sadler, from Poole, lost control of his car on a bend before it ploughed down a grass bank and struck the tree. Mr Maguire was pronounced dead at the scene. Victim: Mikey Maguire pictured on holiday with his mother Allison, who described the sentence as an insult to her son's memory. Sadler had previously been disqualified from driving after being convicted of two counts of aggravated vehicle taking. He appeared in court for one of the counts and driving without 1scence and insurance in July 2010. In both incidents he lost control and caused more than $5,000 damage to one of the vehicles Les Smith, defending, read out a statement from Sadler to Bournemouth Crown Court in which he expressed regret and sorrow for the Mr Maguire's death. He added: The incident haunts me every day and will do for the rest of my life. Nothing that is said or dome to me can make me feel any worse. As well as being jailed Sadler was disqualified from driving for five years and ordered to take an extended test at the end of the ban.

**Summary Generated**:

Joshua Sadler, 21, lost control of his Renault Clio and smashed into a tree. Front seat passenger Mikey Maguire, 19, died at the scene. Sadler had previously been disqualified from driving after two counts of aggravated vehicle taking. He was twice banned from the road before he even had a licence: Mother Allison Jarman described the sentence as an insult to her son's memory."

**Chapter – 7**

**Bibliography**

**Bibliography**

* [**https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail**](https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail)
* [**https://github.com/abisee/cnn-dailymail**](https://github.com/abisee/cnn-dailymail)
* [**https://www.irjet.net/archives/V8/i12/IRJET-V8I12220.pdf**](https://www.irjet.net/archives/V8/i12/IRJET-V8I12220.pdf)
* [**https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2746634.pdf**](https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2746634.pdf)
* [**150330194118.pdf (ijarest.com)**](http://ijarest.com/papers/finished_papers/150330194118.pdf)
* [**https://arxiv.org/ftp/arxiv/papers/2108/2108.01064.pdf**](https://arxiv.org/ftp/arxiv/papers/2108/2108.01064.pdf)
* [**https://youtu.be/4Bdc55j80l8**](https://youtu.be/4Bdc55j80l8)
* [**https://www.researchgate.net/publication/353653704\_Automated\_News\_Summarization\_Using\_Transformers**](https://www.researchgate.net/publication/353653704_Automated_News_Summarization_Using_Transformers)