Saint Petersburg Electrotechnical University
(ETU)

Direction	09.04.01 Computer Science and Engineering
Profile	Computer Science and Knowledge Discovery (In English)
Faculty	Computer Science and Technology
Department	Computer Science and Engineering

Accepted for defense

Head of the department Kupriyanov M.S.

MASTER'S
GRADUATE QUALIFICATION WORK

TOPIC: English automatic text summarization:A deep learning approach

Student		____________ *signature*	Chahedaoui Ahmed Amin
Supervisor	PhD, Associate Professor *(academic degree, title)*	____________ *signature*	Posov I.A.
Consultant	PhD, Associate Professor *(academic degree, title)*	____________ *signature*	Pashkovskii. E. A.

Saint Petersburg2021

TASK FOR THE GRADUATE QUALIFICATION WORK

Approved

Head of the department VT

____________Kupriyanov M.S.

«__»______________2021_.

Student Chahedaoui Ahmed Amin Group 5300

Topic: English automatic text summarization:A deep learning approach

Institution: Department of Computer Science and Engineering

Initial data (technical requirements):

Google Scholar Citations

Contents:

Current Research Analysis and Literature Review, Using deep learning to train a model that detects the processes of sentences and passes along some information it extracted from it with Bert Model.

List of reporting materials: the text of the GQW, illustrations, the text for the qualification thesis, thesis defence presentation working procedure

The task was given

«___»______________2021_.

The GQW was submitted for the defense

«___»______________2021_.

Student _________ Chahedaoui Ahmed Amin

Scientific advisor PhD, Associate Professor *(Academic degree, title)* ____________________Posov I.A.

CALENDER PLAN FOR THE GRADUATE QUALIFICATION WORK

Approved
Head of the department VT
____________Kupriyanov M.S.
«__»______________2020.

Student Chahedaoui Ahmed Amin Group 5300

Topic: : English automatic text summarization: Adeep learning approach

Nº	Stages	Deadline
1	Literature review, proposed methodology	11.20 – 12.20
2	Automatic summarization	12.20 – 01.21
3	Deep neural networks	01.21 – 02.21
4	Contribution of BERT for Extractive Text Summarization on Lectures	02.21 – 03.21
5	Implementation and Evaluation , Analysis of the Social Content of the Order and the Socio-political Conditions for the Implementation of the Project	04.21 – 05.21
6	Conclusion, Cross Reference	05.21 – 06.21

Student ___________________ Chahedaoui Ahmed Amin

Scientific advisor PhD, Associate Professor ______________________ Posov I.A.
(Academic degree, title)

Summary

Explanatory note 88p., 45 fig., 3 tables, 48 sources.

Keywords: ATS, TF, IDF, HMM, ANN, NLP, ANLP, ROUGE, RNN, CNN, AI, BERT, DistilBERT.

Aim / Goal of the Project:

Implement a text summarization for texts of unspecified topics, invent a user interface to view the summary and adjust its parameters, and implement it.

Subject and Object:

In this study, the subject of research is the automatic text summarization. The object of this work is to Study existing types of text summarization, Study different approaches to text summarization including outdated and state-of-the-art approaches, select an approach to implement, implement a summarization, select a form to display a summary.

Final / Expected Outcome:

Implement a user interface to display a summary and allow adjusting it.

Model Description:

The model that is used in this thesis is DISTILBERT model. The. DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than *bert-base-uncased*, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark.

Adequacy of the model:

This work has the potential to reach a high scale of adequacy, adhering to the fact that the hypotheses discussed and the final outcomes are analogous and correlated

Abstract

Information is the oil of the 21st century, and analytics is the combustion engine. Therefore, we are surrounded by data that is produced every day and will remain unuseful for humans unless making it available with new tools and technologies. Automatic text summarization has becomes an important way of finding relevant information precisely in large text in a short time with little efforts. Text summarization approaches are classified into two categories: extractive and abstractive. Through this work, we have examined approaches to text summarization and to evaluate summarization results.

Keywords: Deep learning , Automatic text summarization , Abstractive summary , Extractive summary.

Table of Contents

Definitions And Abbreviations

ATS	Automatic Text Summarization
TF	Term Frequency
IDF	Inverted Document Frequency
HMM	Hidden Markov Models
ANN	Artificial Neural Networks
NLP	Natural language processing
PCA	Principal component analysis
CA	Classical Arabic
BERT	Bidirectional Transformer
ML	Machine learning
GEMS	Generative Modelling for Evaluation of Summaries
BLEU	Bilingual evaluation understudy
RNN	Recurrent Neural Networks
AI	Artificial Intelligence
NN	Neural Network
CNN	Convolution Neural Networks
FRNN	Fully recurrent neural network
LSTM	Long Short-Term Memory
DistilBert	Distil Bidirectional Transformer

Introduction

The information age quickly revolutionizing the way transactions are completed and has changed our whole environment. Now We are surrounded by data, not in a metaphoric sense; it is a physical reality. Internet, television and radio signals are circulation in the air and we receive them by our mobile devices everywhere. Data servers in counts of millions have connected all the planet like a spider web. A considerable portion of this stream of new data is textual. It covers almost almost of the available text corpus, like books, journals and newspapers, it is also connected to millions of organizations, companies, universities and research centers, and individual users that are creating content every day in the websites, blogs, social networks etc. These information is left unusable unless we find a way to make it available for users. Automatic Text Summarization is an attempt to decrease the size of a data while keeping its valuable content. in contrast, is composed with a selection of sentences and text part, phrases, paragraphs from the source text. In general, topic identification, interpretation, summary generation, and evaluation of the generated summary are the key challenges in text summarization and even more challenging with English text. Around 1950's the resources were not sufficient to make an abstractive summarization systems due the difficulties to design and replicate an abstractive algorithm that is close to how a brain receive process and regenerate information. Until recently, neural network models also showed that if we make texts as humane as possible, they may be the most accurate text summaries.

Chapter 1

1 Automatic Summarization

1.1 Introduction

We currently have fast access to a large amount of information. However, most of this information is redundant, trivial, and may not convey its intended meaning. Therefore, it is very important to use automatic text summarization to extract useful information that skips secondary and irrelevant data. The implementation of summarization can improve the readability of the document, reduce the time to find information and allow more information that is applicable to a specific area In this chapter, we will introduce you to the different types and methods used by automatic text summarization., then we talk about the different techniques of NLP and the methodologies of it, in the end we discussed how the evaluation of summary works.

1.2 Definition:

A summary is short text that contains important information from the original of text l and usually too little. Automatically summarize text: This is the shortening process of text documents using resume creation software. The abstract provides the most important or relevant information in the original content. When someone needs to manually summarize a very long content quickly and accurately, the abstract becomes crucial. However, when this operation is done through a computer, we call it automatic text summation (ATS).

The Text summarization divided into two important categories: extractable summarization and abstract summarization.

1.2.1 Extractive Summarization

Our method is based on the extraction of various parts such as sentences and phrases From part of the text, and compiling those parts into an autobiography and other things, Important phrases or expressions will be identified from the source text and extracted In the final summary. The method of identifying key phrases and measuring their importance is to propose topics and propose indicators [8]. Topic presentation technology assumes that the body of the text has a specific topic. Research the corpus to find words that are closest to the topic, and measure their frequency to gauge the importance of sentences.

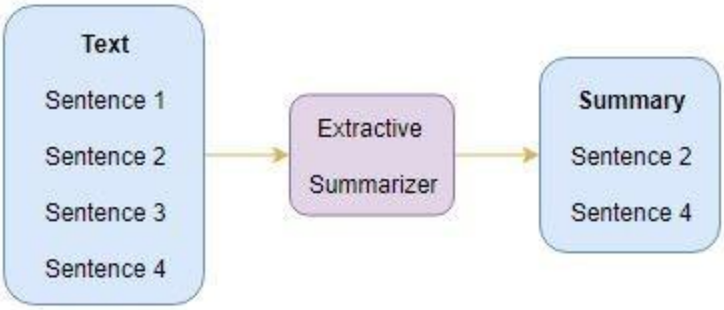

Figure 1-1: Text summarization by extraction

1.2.2 Abstractive Summarization

Abstractive Summarization is a method of automatically creating abstraction through abstraction, which comes from artificial intelligence (AI). The creation of a resume through the application must be done through an understanding of all or part of the text. Therefore, the abstract is based on understanding. If you can find the main meaning of the text original and explain it in fewer words, the abstract is closer to what people usually do. You get the text, compare it with your memory and related information, and Then recreate the core with short text. Therefore, abstract abstracts are more difficult than retrieval methods.

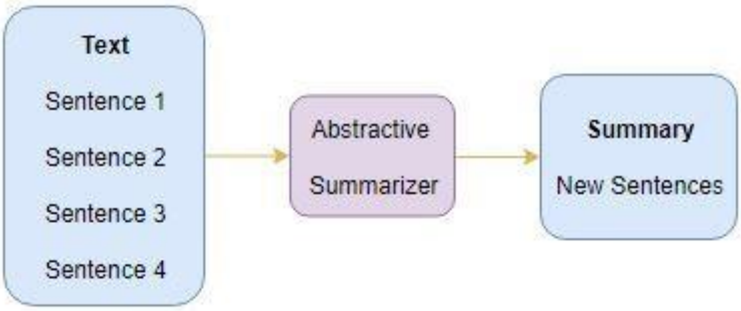

Figure 1-2: Text summarization by abstraction.

1.3 Automatic summarization approaches

Under the two methods cited above, several approaches were proposed where the most common are the following:

1.3.1 Surface level approaches

This method analyzes keywords and phrases such as "final", "important", "in this article" or the complete sentences they contain, and then rearranges them to form a coherent abstract. It is developed [9]. Words are selected based on their frequency (important sentences contain common words), their position (words and phrases, titles and titles are related), and specific words in the original document.

1.3.2 Machine learning based approaches

The main idea of the Corpus-Based approach is that instead of looking for term frequency using the original content, a corpus is made from similar contents and relevance of a word is calculated for example by the Term Frequency Inverted Document Frequency formula.

1.3.3 Cohesion based approaches

The superficial and subject-level methods do not consider the relationship between sentences in the document. For example, in the sentence "I asked him to write a report," the pronoun "he" might be added to the final text without even mentioning the person mentioned. It's hard to understand. Text cohesion defines the relationship between terms in a document and the terms that define text cohesion. This method uses lexical strings [10]. They are grammatically independent sequences of words that represent the coherent structure of the text. For example, the word string can be expressed as follows: [Roman-uppercase-resident]. The context was lost after the summary was created.

1.3.4 Machine learning based approaches

Machine learning methods rely on machine learning algorithms to write resumes. The machine learning method treats the summarization process as a classification problem, in which sentences are divided into summary sentences and non-summary sentences according to the characteristics of the sentences. Hidden Markov Model (HMM) [13] [14] [15] [16] and Bayesian rule [17] are examples of combined machine learning methods that extract materials to provide a set of training documents and their Summary.

1.3.4.1 Naïve-Bayes method

The naive Bayes method is first used to determine whether to use the Bayesian classifier to extract sentences. The system can learn from data [18]. Some features used by the system include capitalized words, sentence length, phrase structure, and word position. Sentences are scored based on these characteristics, and formulas are used to calculate the score The method naïvebayes classifier also used in the DimSum [19].

1.3.4.2 Neural Network method

Artificial Neural Network (ANN) is very popular and powerful machine learning algorithms. ANN is used to create news fragments of any length. A summary of the recommendations with the most categorized articles. Due to the integration of functions, the network has becomeShows the importance of various features used to determine the final score of each proposal [22].

For example, the features are selected according to position of document or position of the sentence. In ANN architecture presented in Figure 5, the following seven features are used:

f1 = Paragraph follows title (Paragraph Position)

f2 = Paragraph location in document

f3 = Sentence location paragraph

f4 = First sentence in paragraph

f5 = Sentence Length

f6 = Number of thematic words in sentence

f7 = Number of title words in sentence

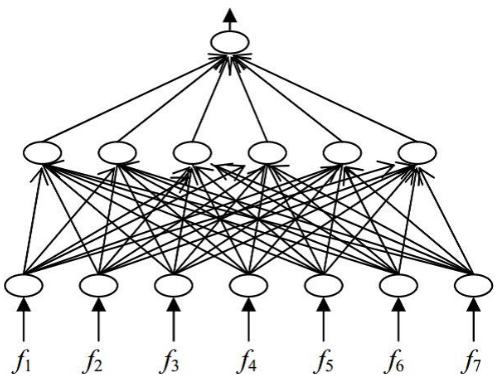

Figure 1-3:The Neural Network after Training [3]

1.4 Natural Language Processing

NLP is the field of computer science that aims to create concepts, find methods, and create software that can understand, learn, and create natural human languages to interact with humans and enabled computers through written and spoken language. Help computers determine how people use language. In the past ten years, English has become more and more important in natural language processing research. This work focuses on various aspects related to this language processing, such as: B. Automatic translation, ATS, and the representation of English features and the types of tasks that deal with them.

1.4.1 English language

English is a West of Germanic language, first used in England in the early middle Ages, and eventually became the main language of the international discourse in the modern world. It was named after Angles, and one of the ancient Germanic people who immigrated to Great Britain, which was later named Britain. Both names come from England, a peninsula in the Baltic Sea. English is actually most closely to Frisian and Lower Saxon, and its vocabulary is greatly influenced by the Germanic languages, especially Northern Norwegian and Latin. And French. English was developed 1400 years ago. The earliest form of English was a group of West Germanic (Ingweon) dialects brought to England by Anglo-Saxon immigrants in the 5th century, collectively referred to as Old English. In the 11th century, Norman conquered England; it was an era when Old French influenced English, especially through the Old Nordic languages. Early modern English began at the end of the 15th century, when the printing press in London came out, the King James Bible and The beginning of a huge change in sound.

1.4.2 English naturel language processing and machine learning

Recently, English natural language processing tools are developed using machine learning algorithms. Machine learning (ML) falls within artificial intelligence(AI) domain . The goal of such technologies is to give computers the ability to learn without explicit programming. ML has been successfully applied in many difficult and complex computing tasks (such as ANLP) without designing and programming explicit algorithms. In addition, the range of ML algorithms that yield satisfactory results has led ML to become vastly involved in NLP in general.

1.4.3 NLP processing steps

To extract an information from any content depends and rely on some levels of text extraction, or a full-up NLP techniques. And there are five primary steps involved in the NLP process.

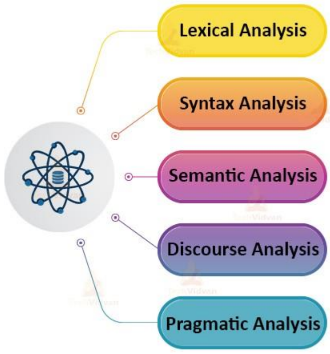

Figure 1-4: NLP processing steps

NLP plays a vital role for supporting human-computer of interaction. With more research of this field, we hope to see a more advancements that will make machines smarter in recognizing and more understanding a human language. Therefore, the operational question when evaluating of NLP algorithm or system is how it will produce the results designed for it. In the next section, we will discuss some final evaluation methods.

1.5 ATS Evaluation

The summaries are generally is more suitable for specific needs, and existing methods of evaluation must be adjusted accordingly. However, scoring abstracts is challenging because there is no perfect abstract for document or set of documents, and there is no defined abstract. The summary is largely an open question [27]. Therefore, Human believes that summaries are an expensive generalization method and a difficult task for him, which means that automated tasks become more difficult and difficult to evaluate. Therefore, it is clear that we need complex assessment methods that can overcome all these difficulties.

1.5.1 Evaluation methods

Tons of numerous approaches for outline evaluation are projected within the last 2 decades. This prevents America from being exhaustive. we are going to instead concentrate on the strategies that are wide used throughout recent analysis campaigns (Especially at the text analysis conference recently organized by NIST): Callback-oriented Gusting Evaluation Sub-Research (ROUGE), which can be a fully automated method, PYRAMID (hybrid method) and provided by manual analysis A series of indicators that we can talk about alternative of strategies that have not yet been widely used.

1.5.2 Manual Methods

The most obvious and easiest way to score abstracts is to ask reviewers to rate quality. For Document Understanding Conference, judges must score the scope of the abstract, which means they must perform a comprehensive assessment of the scope. The candidate's abstract covers the text provided as input. In the future structure, especially in the TAC (Text Analysis Conference), a demand-oriented summary should be created. The judge should then evaluate the requirements in the opinion [27].

1.5.3 Automatic Methods

In order to facilitate the summarization evaluation process, several metrics were proposed. We detail here the most popular measures: Rouge and Pyramid.

1.5.4 ROUGE

ROUGE is AN instructively live wide utilized in report area [28]. It measures how much of the data in human summaries are reproduced in automatic summaries. Although, it's a really straightforward live (which computes the amount of common n-grams within the human and automatic summaries), it includes measures to automatically confirm the standard of a outline by scrutiny it to different (ideal) summaries created by humans. The Recall indicates share of the main content to the human outline that's reproduced within the automatic summary and valuate the completeness, or what quantity data within the reference summaries are coated by the subject of summary automatic. It is computed as:

Recall = number of overlapping words / Total words in reference summary (1.2)

The Precision indicates the percentage of the content in the automatic summary that is relevant, and evaluate the correctness, or how much information in the summary are also in the reference summaries. We notice that there is a tradeoff between precision and recall (increasing one tends to decrease the other). Precision can be computed as:

Precision = number of overlapping words / Total words in reference summary (1.3)

F-measure combines both measures, being a unique indication of the system performance. The basic way how to compute the F-score is to count a harmonic average of precision and recall:

$$F = 2 * P * R / (P + R) \quad (1.4)$$

Below is a more complex formula for measuring the F-score:

$$F = (\beta^2 + 1) P R / (\beta^2 P + R) \quad (1.5)$$

1.5.5 Generative Modelling for Evaluation of Summaries (GEMS)

This method [29] suggests the use of signature for analyzing that how they are captured in automatic summaries. Such as nouns or verbs query terms and terms of reference summaries. The distribution of the signature terms is calculated in the source document and then the possibility of a summary being biased towards such signature-terms is gained.

1.5.6 Pyramid

As we have seen, the ROUGE measurement is based on finding exact matches between certain sequences in the candidate abstract and certain reference abstracts. Therefore, if reformulation techniques are used to prepare candidate abstracts, these methods will be ineffective. This estimate is more accurate than ROUGE, but requires manual labor to determine the final content unit (SCU) used. Compare the information in the abstract, assign verbal expressions to them, and calculate the weights required for the evaluation. However, according to [32], this method seems to be better than ROUGE's association with human judgment, which may be because it considers certain semantics.

1.6 Conclusion

Automatic text summarization (ATS) is one of the best challenging and also interesting tasks in the domain of natural language processing (NLP), and one of the most important processes. In this chapter, we presented the different types and approaches used the automatic summarization (ATS) of texts, then we talked about how evaluation of summary works and then we explained steps and techniques used in Naturel language processing. The next chapter will focus on deep neural networks and how they can be used in the NLP tasks. We will more focus on BERT model.

Chapter 2

2 Deep Neural Networks

2.1 Introduction

Automatic text summarization became popular, and until recently text summarization was dominated by unsupervised information retrieval models dominated In [33], he showed that persistent neural models are promising to summarize the text. This marked the start of the widespread use of neural network-based text synthesis models, due to their superior performance compared to conventional techniques. . In this chapter, we present a different deep learning model as well as the different types of neural networks used in automatic text summarization, and then we detail the structure of the recurrent neural network (RNN) with explication for how it works.

2.2 Machine learning

Machine learning (ML) is an application of artificial intelligence (AI) that allows the system to automatically learn from experience and make improvements without explicit programming, while ML focuses on developing data that can be accessed and used Computer programs for learning. The learning process starts with observations or data (such as examples), direct experience or explanations, to find patterns in the data based on the examples we provide and make more informed decisions in the future. The main goal is to give the computer the ability to learn and adapt automatically without human intervention or assistance The corresponding procedure. ML algorithms are often classified as supervised, unsupervised reinforcement learning.

2.3 Machine learning methods

2.3.1 Supervised Learning

A set of training examples are provided with correct answers (goals) and, based on this training set, the algorithm generalizes to correctly answer all possible inputs. This is also called learning models.

In essence, supervised learning refers to the learning that we use well-classified data to train or train machines. This means that certain dates have been marked as correct answers. Then, the machine will receive a new set of examples (data) used to monitor the learning algorithm, used to analyze the training data (a set of training examples), and output the correct results from the decomposed data [4]. Supervised learning is divided into two types of algorithms:

- Classification: is a problem that is used to predict which class a data point is part of which is usually a discrete value.
- Regression: is a problem that is used to predict continuous quantity output. A continuous out- put variable is a real-value, such as an integer or floating point value.

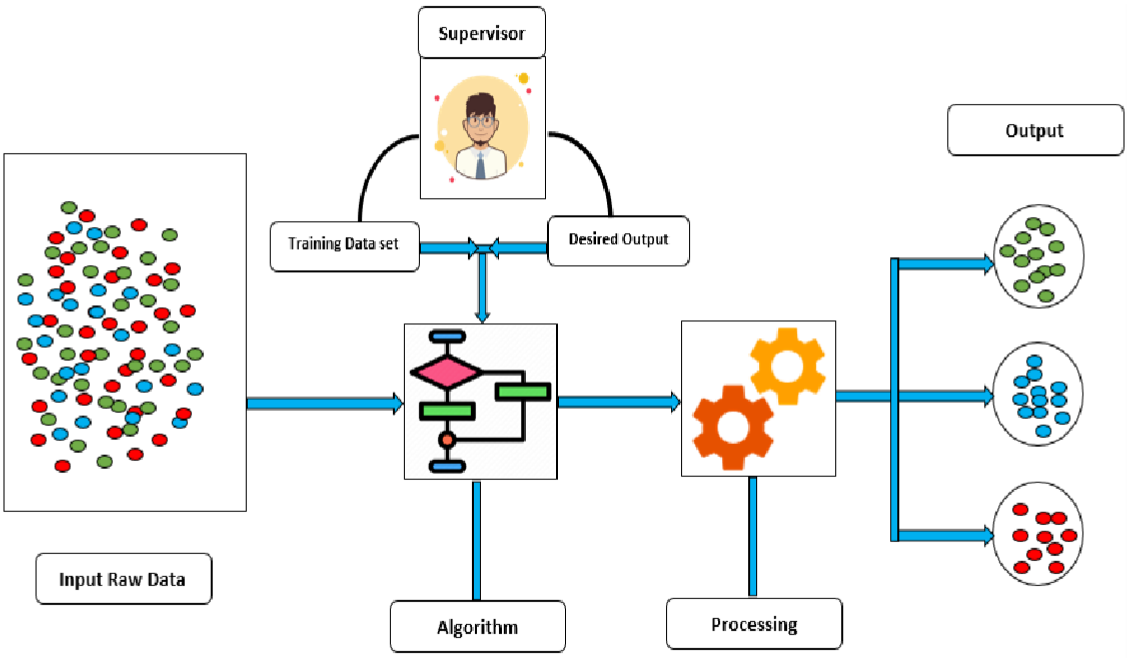

Figure 2-1 :Supervised machine learning, the algorithm learns from labeled data

2.3.2 Unsupervised Learning

Correct responses are not provided, but instead the algorithm tries to identify similarities between the inputs so that inputs that have something in common are categorized together. The statistical approach to unsupervised learning is known as density estimation. Unsupervised learning is machine learning that uses unclassified or labeled information, which allows algorithms to operate on information without guidance. Here, the machine’s job is to group unclassified information based on similarities, patterns, and a differences without the prior of data training. Without a teacher, this means that the machine will not learn. So the machine can't find it the hidden structure in the data that has not been marked by us [4].

Unsupervised learning has or classified into two algorithms categories:

- Clustering: the process of organizing objects into groups whose members are similar in some way.
- Association: The problem of learning association rules arises when we want to discover some rules that describing the large amounts of data. For example, people who buy X will also buy Y.

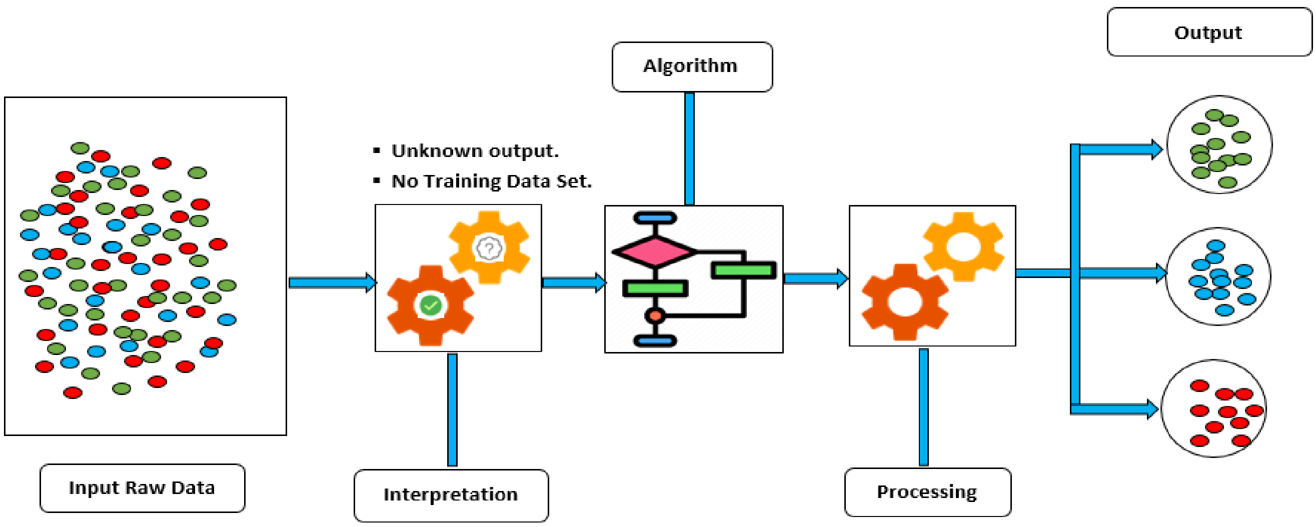

Figure 2-2: Unsupervised machine learning

2.3.3 Reinforcement Learning

Reinforcement learning is the field of machine learning introduced in [34] and [35]. It's about taking the right steps under the right circumstances to maximize your rewards. Various software and machines use it to find the best behavior or path at a given point in time. The difference between reinforcement learning and supervised learning is that in supervised learning, the learning data has an answer key. Reinforcement learning will not give the answer, but the reinforcement agent will decide what to do to complete the task. If there is no training data set, then you can safely learn from experience. Reinforcement learning is similar to many topics called machine learning, planning and mountaineering. In a sense, reinforcement learning is both a problem and a type of solution. They are in a certain type of problem and the field and problem they are in. Both apply to be investigated. Method maximize Numerical reward signal. Basically, this is a feedback problem, because the actions of the training system affect subsequent input.

2.4 Deep learning

Deep learning is like a subset of ML in which artificial neural networks (ANN) (algorithms inspired by human brain) learn from the large amounts of the data. As we have learned from our own experience, deep learning algorithms will perform the task over and over and change it every time. Slightly improved the prognosis [36]. We say "deep learning" because neural networks have multiple (deep) layers that can be learned [36]. Any problem that needs to be solved by "thought" is a problem that deep learning can learn to solve. Hence, neural networks are sometimes referred to deep neural networks as well. Deep learning models with neural network architectures are trained, so that the models can learn features automatically by analyzing the labeled examples. Hence, the performance of a deep learning system increases with the increase of the amount of data. In traditional machine learning approaches, the performance becomes constant after certain steps, while the performance of the deep learning systems increases with the increase of the amount of data.

2.5 Neural Network

A neural network is a computer system with interconnected nodes that function like neurons in the human brain. Using algorithms, they can identify, group and classify hidden patterns and correlations in the original data, and continue to learn and improve over time. A type of machine learning that simulates the human brain, which creates an artificial neural network that uses an algorithm to enable a computer to learn by merging new data. NN consists of a series of interconnected neurons. The number of neurons that make up a neural network can be hundreds or even millions, and can be divided into multiple levels. The hidden layer remains between the input layer and the output layer. The number of hidden levels can be zero or more. There are different types. NN is used for machine learning and deep learning.

2.5.1 Artificial Neural Networks (ANN):

Artificial Neural Network (ANN) is like computational model that based on the structure and function of biological for neural networks [37]. When the neural network changes or learns in a sense based on the input and output, the information transmitted through the network will affect the structure of the ANN. There are input and output layers and a hidden layer in most cases, the hidden layer accept input, transform it and then passes it to the output layer. Figure 2.3 shows the structure of ANN. Artificial neural networks are regarded as nonlinear statistic of data modeling tools, which are used to model complex a relationships between inputs and outputs or discover patterns. Artificial neural network is also called neural network.

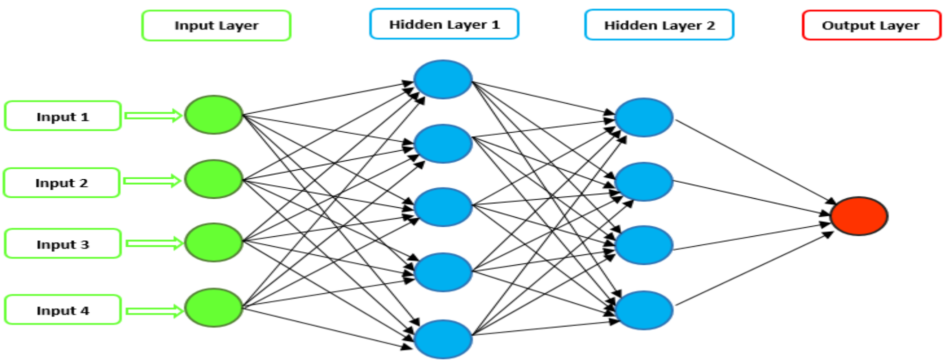

Figure 2-3 : A schematic artificial neural network (ANN) with two hidden layers and a single neuron output

The neural network is divided into several layers:

- **Input layer**: The neurons in the input layer receive information, which is thought to explain all problems to be analyzed.

- **Hidden layer**: Actually this layer is intermediate layer, and the neural network can use the intermediate layer to model nonlinear phenomena. Because there is no direct connection with the outside world. The output of each hidden layer is the input of the block of the next layer.

- **Output layer**: The output layer is the last layer of the network. Produce results, which are predictions.

2.6 Types of ANN

2.6.1 Convolutional neural network (CNN)

Convolutional Neural Network (CNN) is a special type of the artificial neural network that uses the perception of the supervised algorithm of the learning machine to learn data in order to analyze the data. A CNN is also known as a ConvNet Like other kinds of artificial neural networks, it has an input layer, an output layer and various hidden layers. Some of these layers are convolutional, using a mathematical model to pass on results to successive layers. These layers take the input, transform the input, and then feed the transformed input into the next layer. The layers can detect patterns by using filters. Each layer has a large number of filters. A filter is represented by a matrix, which are moved over the original matrix and multiplied with the corresponding index of the original matrix, and sums them up to convolve the required features. This simulates some of the actions in the human visual cortex. CNNs are a fundamental example of deep learning, where a more sophisticated model pushes the evolution of artificial intelligence by offering systems that simulate different types of biological human brain activity. CNNs can be applied to the image processing [38], a natural language processing (NLP) and other tasks.

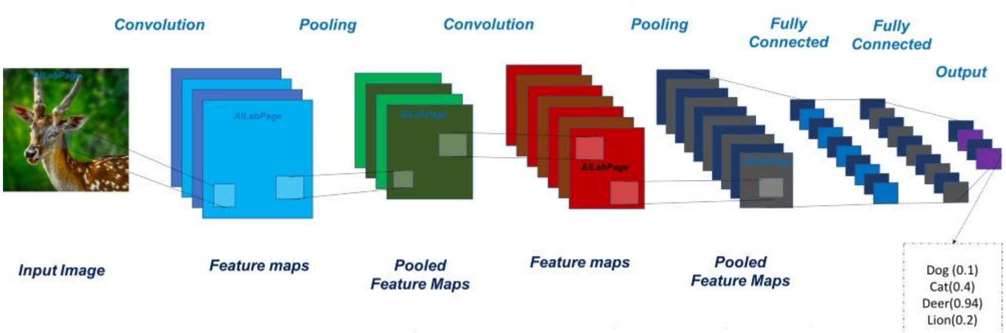

Figure 2-4: CNN structure used for image recognition[6].

This progression is one among the foremost vital strides of the task and it contains 3 sections

- **Convolution:** The essential role of Convolution is to extricate highlights from the image. Convolution saves the spatial connection between pixels by learning image highlights utilizing little squares of information.

- **Pooling:** :Pooling is also known as subsampling or down sampling and it reduce the size of the spatial dimension by sub-sampling their inputs, which is applied after the convolutional layers Pooling layer is normally inserted periodically between two successive convolutional layers. Pooling is very important to provide a fixed size output matrix and reduce the complexity of the output dimensions.

- **Flattening:** The matrix changed over into a linear array that to enter it into the hubs of our neural system.

2.6.2 Recurrent Neural Networks (RNN)

Recurrent Neural Network (RNN) [39] is a type of neural network that uses the output of the previous step as the input of current step. In traditional NNs, all inputs and outputs are independent of each other, but for example, in the case where the next word in a sentence needs to be predicted, the previous word is needed, so the previous word must be stored. For example, we take a sequence of two words. We are told to predict the third word of the sequence. It would be a lot easier for the model to predict the third word if it knew the first two words. Thus, RNN came into existence, which solved this issue with the help of a Hidden Layer. The Hidden state is the most important feature of RNN; it stores some information about the sequence. RNN has a "memory" that is used to store all the information about the calculated content. It uses the same parameters for each input because it performs the same work on all inputs. Or a hidden layer is used for output [40]. Compared with other neural networks, this reduces the complexity of the parameters.

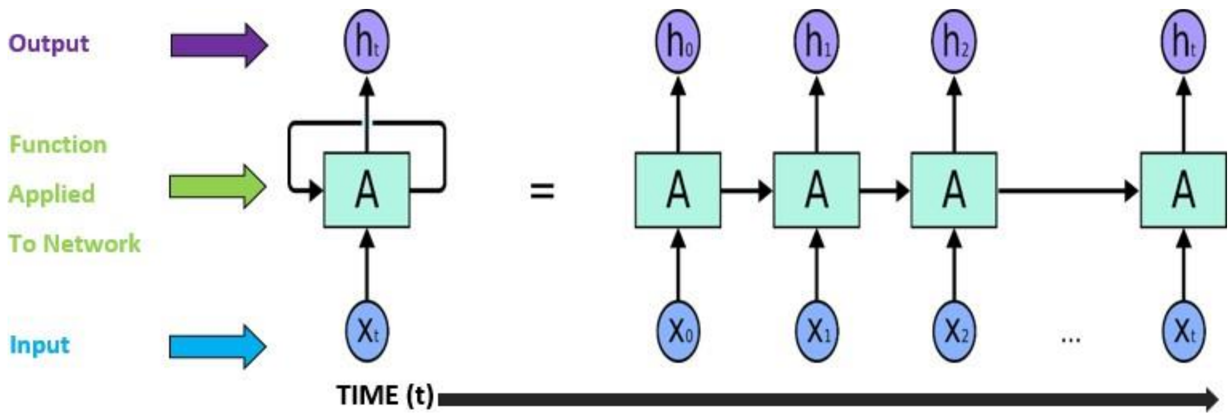

Figure 2-5:An unrolled recurrent neural network[7].

The above figure shows a RNN being unfolded into a full network. By unfolding we simply mean that we are repeating the same layer structure of network for the complete sequence.

- X_t: is the input at time step t. X_t is a vector of any size N.

-This is the Hidden state at time t. This is the "memory" of network. It is calculated that based on the previous hidden state and entered into the current step.

Represented by At= f (W Xt +U At -1). Here W and U are weights for input and previous state value input. And f is the non-linearity applied to the sum to generate final cell state.

2.7 Natural Language Processing

Natural Language process (NLP) may be a field of computer science, computer science and linguistics as all those such arena brought it into play. typically it deals with the interactions between machines and human languages that accomplish task on analyzing, understanding and generating the language, that human use naturally so as to move with computers in each oral Associate in Nursing written contexts exploitation natural human languages rather than laptop languages [41]. it's an interdisciplinary space supported versatile arena of study as well as laptop engineering, that provides ways for model illustration, algorithmic program devise and accomplishment; linguistics, that categorizes linguistic forms and practices; mathematics, which provides formal models and methods; psychology, that studies models and theories of human behavior; statistics, which offers procedures for predicting measures supported sample records; and biology, that travels round the underlying architecture of linguistic processes within the human brain [42].

2.7.1 What is Text Summarization in NLP?

Let's see what a text summary is first before we see how it works.

We have two different approaches that are used for text summarization:

- Extractive Summarization
- Abstractive Summarization

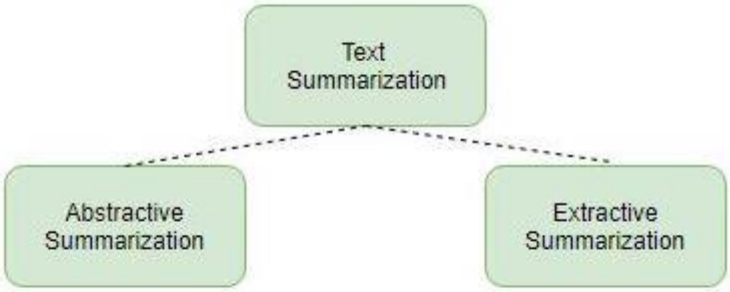

Figure 2-6: Text summarization types

2.7.2 BERT Model for NLP

BERT (Bidirectional Encoder Representations from Transformers). it's caused a stir within the Machine Learning community by presenting progressive ends up in a large type of natural language processing tasks, together with Question responsive (SQuAD v1.1), linguistic communication logical thinking (MNLI).

BERT is the key technical for innovation to applying the bidirectional coaching of Transformer, a well-liked attention model, to language modelling This is usually indistinguishable from previous attempts to check text consistency from left to right or from left to right and from right to left. The results show that the language model after two-way training can better understand the context of the language. The flow of these one-way language models. the researchers detail a completely unique technique named cloaked lumen (MLM) which permits biface coaching in models within which it had been antecedently impossible[43].

2.7.3 How BERT works

BERT actually uses a related attention conversion mechanism that can check the linguistic relationship between words in a very complete text. The original form of the converter contains two independent mechanisms: an encoder for validating input text and a decoder for a prediction tasks. Since the purpose of BERT is to obtain a language model, only one encoder mechanism is needed [43].

Transformer encoder reading all sequence of words at once. Thus it's thought-about bidirectional, though it might be a lot of correct to mention that it’s non-directional.

This characteristic permits the model to be told the context of a word supported all of its surroundings (left associated right of the word) .

chart below can be a high-level for description of electrical device encoder. The sequence of tokens actually they are a input, which are 1st embedded into vectors then processed within the neural network. And about output we can say it's a sequence of vectors of size H, within which every vector corresponds to an input token with identical index. When training language models, there's a challenge of process a prediction goal. Many models predict future word in an exceedingly sequence . BERT uses 2 training strategies:

2.7.3.1 Masked LM (MLM)

Before entering the word sequence in BERT, replace 15% of the words in each sequence with [MASK] [50]. Then, the model will try to predict the initial value of the masked word based on the context provided by the other unmasked words in the sequence. Technically speaking, predicting the output of word requires the following operations:

1-Add a classification layer on top of the encoder output.

2-Multiply the output vector by the embedding matrix, and then converts it to a dictionary dimension.

3- Use Softmax to calculate the probability of each word in the dictionary.

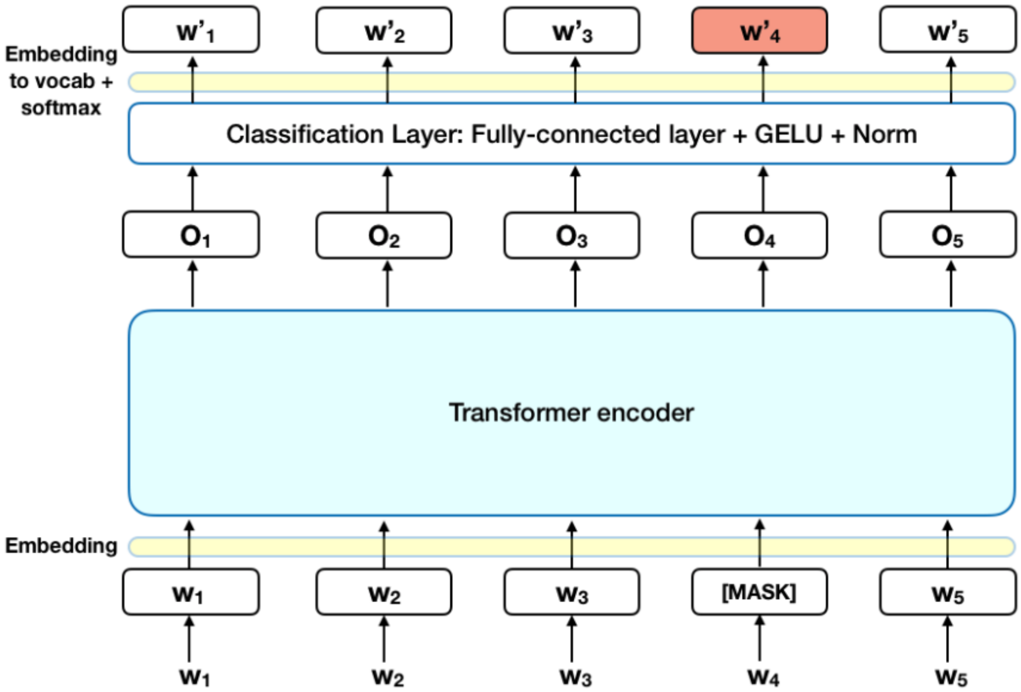

Figure 2-7: The mechanism of Masked LM Bert model

BERT loss rate only considers the prediction of the masked value, and ignores the prediction of unmasked words. As a result, the convergence speed of the model is slower than that of the directional model, and this property is compensated for by its more knowledge of the context (see result #3)

2.7.3.2 Next Sentence Prediction (NSP)

During the BERT training process, this model receives a combination of sentences as input, and learns to predict whether the second sentence in the pair will become a future sentence in the original document. During training, 50% of the entries are paired, and the second sentence is the next sentence in the original document, or 50% of random sentences are selected from the corpus based on the second sentence. the idea is that the random sentence are going to be disconnected from the primary sentence[43].

2.8 Extractive Summarization with BERT

Generally, in order to use BERT to extract a resume(extractive summarization), you need to create an illustration for each sentence. However, BERT is trained as a masked language model, the output vector is token-based, not sentence-based. Although

BERT has segmented insertion to represent completely different sentences, it only has two labels (sentence A or sentence B), rather than extracting multiple sentences in a summarization. BERT input stream and attachments to generate an accessible summary. Encoding Multiple Sentences As illustrated in Figure 1, we insert a [CLS] token before each sentence[44] and about [SEP] token must be after sentence In vanilla BERT.

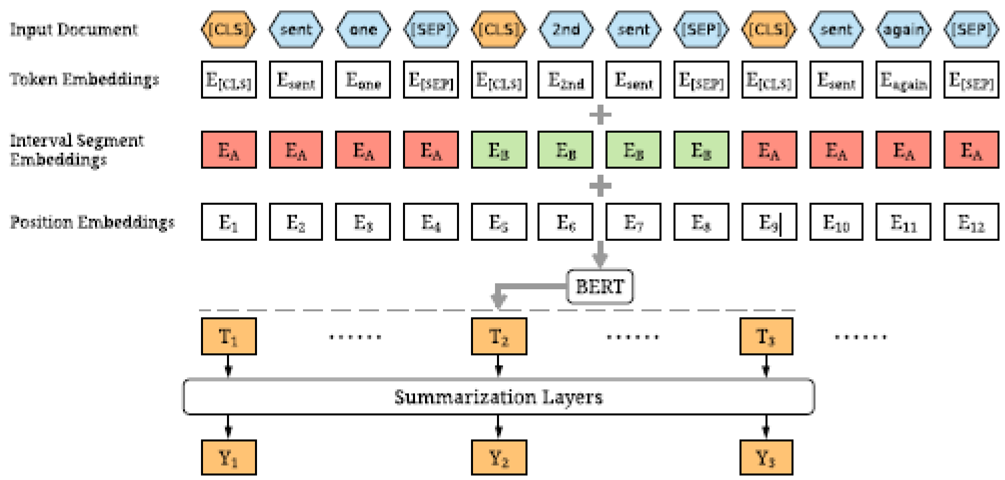

Figure 2-8: The overview architecture of the BERTSUM model[50].

The [CLS] is employed as a logo to aggregate options from one sentence or a combine of sentences. We tend to modify the model by mistreatment multiple [CLS] symbols to urge options for sentences ascending the symbol. Interval phase Embeddings we tend to use interval segment embeddings to differentiate multiple sentences at intervals a document. For sent i we will assign a phase embedding Semitic deity i is odd or even. For example, for (sent1; sent2; sent3; sent4; sent5) we are going to assign (EA;EB;EA;EB;EA). The vector Ti that is that the vector of the i-th [CLS] image from the highest BERT layer are used because the illustration for sent i.

2.9 Fine-tuning with Summarization Layers

After obtaining vectors of sentence from model BERT, we created many specific output layers on top of the BERT output to capture the document level parameters for the main idea (extracting summaries), For every sentence senti, we'll calculate the ultimate predicted score ^ Yi. The loss of the entire model is that the Binary Classification Entropy of ^ Yi against gold label Yi. These report layers are collectively fine-tuned with BERT[44].

2.9.1 Simple Classifier

Like in the original BERT document, a simple classifier only adds a line layer to the BERT output, and uses the Sigmoid function to obtain the expected score, and the Sigmoid function to obtain the expected score:

$$\hat{Y}_i = \sigma(W_o T_i + b_o) \quad (1)$$

2.9.2 Inter-sentence Transformer

Instead of a the simple of sigmoid classifier, but applies more converter levels to sentence views only, and extracts document-level functions from the BERT results, which are focused on the generated tasks from the output of Bert .:

$$\tilde{h}^l = \mathrm{LN}(h^{l-1} + \mathrm{MHAtt}(h^{l-1})) \quad (2)$$

$$h^l = \mathrm{LN}(\tilde{h}^l + \mathrm{FFN}(\tilde{h}^l)) \quad (3)$$

where $h^0 = \mathrm{PosEmb}(T)$ and T are the sentence vectors of output by BERT, PosEmb is the function of adding positional embeddings (indi- cating the position of each sentence) to T ; LN is the layer normalization operation, MHAtt is the multi-head attention operation the superscript l indi- cates the depth of the stacked layer.

The final output layer is still a sigmoid classi- fier

$$\hat{Y}_i = \sigma(W_o h^L + b_o) \quad (4)$$

2.9.3 Recurrent Neural Network

Though the Transformer model achieved nice results on several tasks, there are proof that continual Neural Networks still have their advantages, especially once combining with techniques in Transformer. Therefore, we tend to apply associate degree LSTM layer over the BERT outputs for learning the specific of summarization features. For stabilization of the training, pergate layer normalisation [44] is applied among every LSTM cell. At time step i,and about the input to the LSTM layer is the output of BERT Ti, and also the output is calculated.

$$\begin{pmatrix} F_i \\ I_i \\ O_i \\ G_i \end{pmatrix} = \mathrm{LN}_h(W_h h_{i-1}) + \mathrm{LN}_x(W_x T_i) \quad (5)$$

$$\begin{aligned} C_i = \sigma(F_i) \odot C_{i-1} & \\ + \sigma(I_i) \odot \tanh(G_{i-1}) & \quad (6) \\ h_i = \sigma(O_t) \odot \tanh(\mathrm{LN}_c(C_t)) & \quad (7) \end{aligned}$$

If F_i, I_i, O_i are forget gates, input gates, and output gates; so G_i is the hidden vector and C_i is the memory vector; and the output vector is h_i; LN_h, LN_x, LN_c are there difference layer normalization and operations; Bias terms are not shown.

The final output layer is also a sigmoid classifier:

$$\hat{Y} i = \sigma(W_o h_i + b_o) \qquad (8)$$

2.10 Conclusion

In this chapter, we have discussed the various methods of deep learning that are employed in text summarization process and also the techniques that are developed over the years. We have also discussed the activation functions, which have been used to evaluate and show the results obtained by the deep learning approaches. We have focused on BERT model architecture which has been commonly used for the text summarization purpose. we have found that this model gives results either better. It has been found that Since BERT's goal is to generate a language model, only the encodermechanism is necessary. the encoder-decoder models with BERT model are the most widely used approaches for the summarization purpose. In the following chapter, we will detail our deep text summarization model.

Chapter 3

3 Contribution of BERT for Extractive Text Summarization on Lectures

3.1 Introduction

In training, the automatic text summarization of the lecture extractor is a powerful tool that can be used to infer key points without manual intervention or work. Video conference logs can be used in many MOOCs, but it is difficult to find the most valuable information about each conference. Many attempts have been made to solve this problem, but almost all solutions are implemented in ancient natural language. Processing algorithms that often need to be maintained due to poor generalization capabilities. Due to these limitations, many regular results of these tools may appear randomly when creating content. In the past year, many new deep learning methods have emerged, and these methods have shown the latest results on many tasks, such as automatic summarization of text extraction. Due to the need for more modern musical instruments in the classroom therefore, the meeting summary service provides a RESTful API like tool that extracts each meeting log to demonstrate that the implementation can be extended to other domains. Conference summary, methods of building services, modelling results and indicators, and sample summaries comparing them with common tools such as TextRank. The lecture summarization service uses extractive summarization. In general, The Extractive text summarization adopts the structure of the text, sentence or phrase, and generates a summary using only the content of the source material. Only the Summarization is recommended for the initial implementation of the service. Our contribution consists of adapting, testing , The extractive summarization appraoches used BERT model. we use a variant of BERT to classify sentences. And try to apply the PCA to 2D using k-means clusterization for make a summary . Summarization process starts with preprocessing, division on traning and testing dataset and then fitting and testing models. The main objective is to explain the extractive text summarization pipeline that contain the classification of sentences.

3.2 Towards Deep Learning

After highlighting all previous research projects, you have not yet implemented deep learning in your transcript as a lecture summary. Even for the most modern projects, there are many reasons not to use it.) Has become the standard method for many data-intensive natural language processing applications. Expensive computing resources and a tendency to overmatch [45] Considering this, the researcher Vaswani proposed An excellent architecture called Transformer, which gets rid of the use of RNN and Convolutional Neural Network (CNN), uses an architecture composed of power grids and attention mechanisms Although the Transformer architecture solves some of the problems of RNN and CNN, it still performs well on many NLP tasks. At the end of 2018, Google researchers built an unsupervised learning architecture based on the Transformer architecture called BERT (Bidirectional Encoded Transformer Representation), which almost surpassed the performance on various tasks, All NLP models. Released some pretrained models that can be used for communication Training in many different fields and tasks. Another component missing from the early research project is the dynamic or custom pivot size function. The conference may wish to set the number of sentences for each lecture and provide more or less information as needed. Since the BERT model generates bids, these suggestions can be grouped by K to achieve dynamic meeting results. With this in mind, Lesson Pivot Service took the exact same approach by creating dynamic dashboards. Put the focus phrases in a group rather than a fixed-size static abstract [45].

3.3 Method

Lecture summarization service method consists of two main components. One of the functions is conference registration and summary management, which enables users to create, edit and delete and retrieve saved items. Model K-means for summary, Each part will be described in detail below, and describe the motivation and realization of each feature.

3.4 Extractive Text Summarization with BERT and K-Means

When creating a summary of a saved meeting, the meeting summary engine uses a pipeline that symbolizes the incoming paragraph text with pure sentences, passes the tokenized BERT model sentence to output from the output embedding, and then K-means groups the sentences selected in attachments and inline, and the closest focus is the candidate's final suggestion. BERT converts them to numbers (768 numbers for each sentence, in fact we used DistillBert, it is a faster and smaller version of BERT). Then the standard approach is to classify sentences based on that numbers. We applied PCA to reduce 768 numbers to just 2 numbers for each sentence. This is a Principal Component Analysis, it is used to reduce dimension. It tries to find points in lower dimensions in such a way that distances between points are roughly the same [45].

After we have sentences-points on the plane, we cluster them, that is we find groups of points close to each other. A cluster is a set of sentences with similar senses. So we can select only one sentence from each cluster to create a summary. These clusters were searched by the K-Means algorithm.

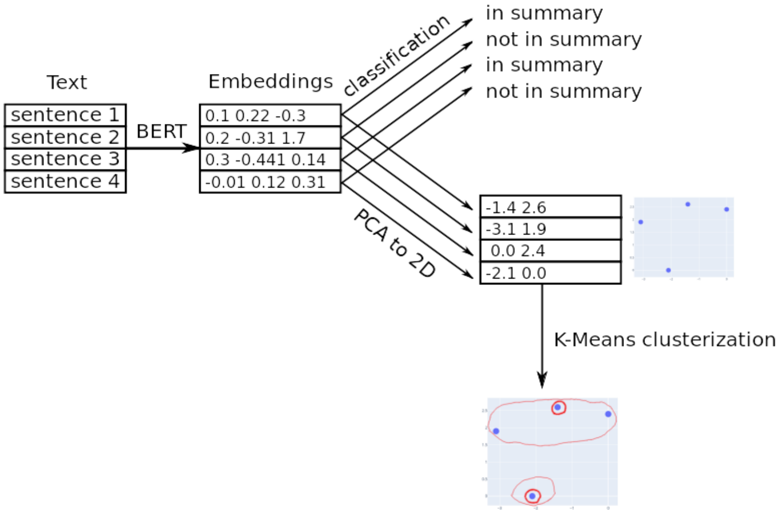

Figure 3-1: Extractive Text Sumarization Pipeline

3.5 Textual Tokenization

Tokenization is a method for dividing texts into tokens; Words are often separated from each other by blanks (white space, semicolons, commas, quotes, and periods). These tokens could be individual words (noun, verb, pronoun, and article, conjunction, preposition, punctuation, numbers, and alphanumeric) that are converted without understanding their meaning or relationships. The list of tokens becomes input for further processing[46].

Due to the difference in the text quality of meeting minutes, a combination of several tokenization methods was used in the model before input. For transcripts derived from Udacity, a special parser was created to convert the data from the ".srt" file into a standard paragraph format, which is a special format that contains the timestamp of the related phrase. The NLTK library for Python is used to extract sentences from speeches and share the content passed to subsequent models. The final step in marking the text is to delete or edit candidate suggestions to keep only these suggestions in the final resume that does not require additional context[46].

3.5.1 Text summarization with NLTK in python

Text synthesis is a subfield of Natural Language Processing (NLP) that deals with extracting summaries from huge chunks of text. There are two main types of techniques used to summarize text: NLP based techniques and deep learning based techniques. In this article, we will see a simple NLP-based technique for summarizing a text. We will not be using any machine learning libraries in this article. Instead, we'll use the Python NLTK library to summarize Wikipedia articles[46].

3.5.2 Text Summarization Steps

We can see from the above paragraph that this mainly motivates others to work hard and never give up. To summarize the above paragraph using NLP-based techniques, we need to follow a set of steps, which will be described in the following sections:

3.5.2.1 Convert Paragraphs to Sentences

First we need to convert the entire paragraph into sentences. The most common way to convert paragraphs to sentences is to split the paragraph facing a point. So if we divide the paragraph under discussion into sentences, we will get the following sentences:

1. So, keep working
2. Keep striving
3. Never give up
4. Fall down seven times, get up eight
5. Ease is a greater threat to progress than hardship
6. Ease is a greater threat to progress than hardship
7. So, keep moving, keep growing, keep learning
8. See you at work

3.5.2.2 Text Preprocessing

After converting paragraph to sentences, we need to remove all the special characters, stop words and numbers from all the sentences. After preprocessing, we get the following sentences:

1. keep working
2. keep striving
3. never give
4. fall seven time get eight
5. ease greater threat progress hardship
6. ease greater threat progress hardship
7. keep moving keep growing keep learning
8. see work

3.5.2.3 Tokenizing the Sentences

We need to tokenize all the sentences to get all the words that exist in the sentences. After tokenizing the sentences, we get list of following words:

```
['keep',
 'working',
 'keep',
 'striving',
 'never',
 'give',
 'fall',
 'seven',
 'time',
 'get',
 'eight',
 'ease',
 'greater',
 'threat',
 'progress',
 'hardship',
 'ease',
 'greater',
 'threat',
 'progress',
 'hardship',
 'keep',
 'moving',
 'keep',
 'growing',
 'keep',
 'learning',
 'see',
 'work']
```

Figure 3-2: Tokenizing The Sentences

3.6 BERT for Text Embedding

The BERT architecture was chosen because it has better performance than other NLP algorithms when implementing the proposal. BERT is based on transformer architecture, but its goal is specific to pre-training. In the first step, 10% to 15% of the words in the training data are randomly masked to try to predict the masked words. In the other step, the input sentence and the candidate sentence are used to predict whether the candidate sentence follows the input sentence correctly. ...Even with many GPUs, training may take several days [45]. In this case, Google released two BERT models for public use, one with 110 million parameters and the other with 340 million parameters. Finally, the BERT model was selected as the conference result service.

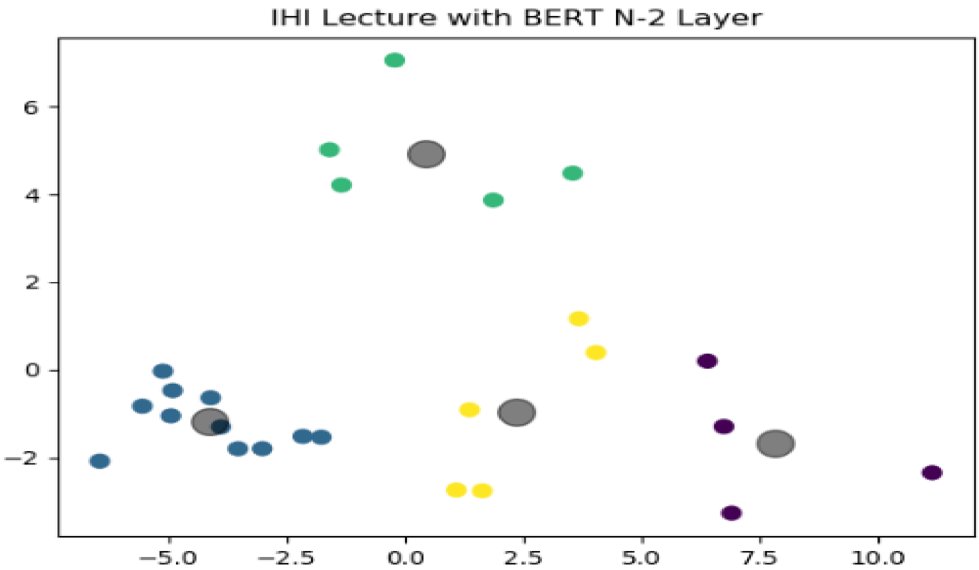

Figure 3-3: Introduction to Health Informatics lecture with BERT N-2 layer embeddings

Using the pre-built standard BERT model, multiple levels can be selected for the tab. Use the BERT level [cls] to create the N x E matrix required for grouping, where N is the number of sentences and E is the size of the attachment. However, the result of level [cls] may not provide the best inline rendering for the collection. ...Due to the nature of the BERT architecture, the output is output to other layers embedded in the network N x B x E, where W corresponds to the token word. To get around this issue, the embeddings can be averaged or maxed to produce an N x E matrix. After experiments with Udacity extractive summarizations on Udacity lectures, it was determined that the second to last averaged layer produced the best embeddings for representations of words Ultimately, this is determined by visually inspecting the groups during the initial activation process. Figures 3 and 4 show examples of differences between two different graphs. Compared with the last layer [cls] of the BERT network, the better representation of the proposal in the N-2 layer is to move the last layer. . The classification problem when the model was first formed.

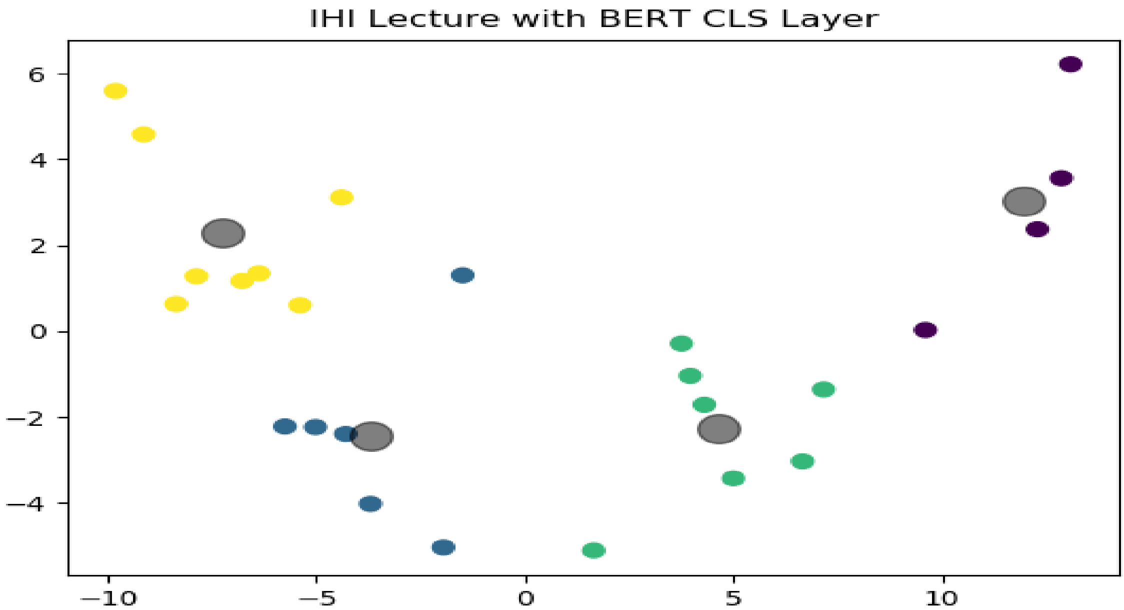

Figure 3-4: Introduction to Health Informatics lecture BERT [cls] layer embeddings

For the lecture summarization service, the basic application and BERT implementation use the pytorch Prerained BERT library organized by Huggingface. Basically, the library is the author of Pytorch implemented by a model previously trained by Google. On high of the original BERT model, the pytorch pretrained BERT library conjointly contains the OpenAi GPT-2 model, that may be a network that expands on the first BERT architecture. once examining the sentence embeddings from each the GPT-2 and original BERT modelit was evident that BERT embeddings were additional representative of the sentences, Euclidean distances between clusters. Below is an example of clustering with the GPT2 embeddings.

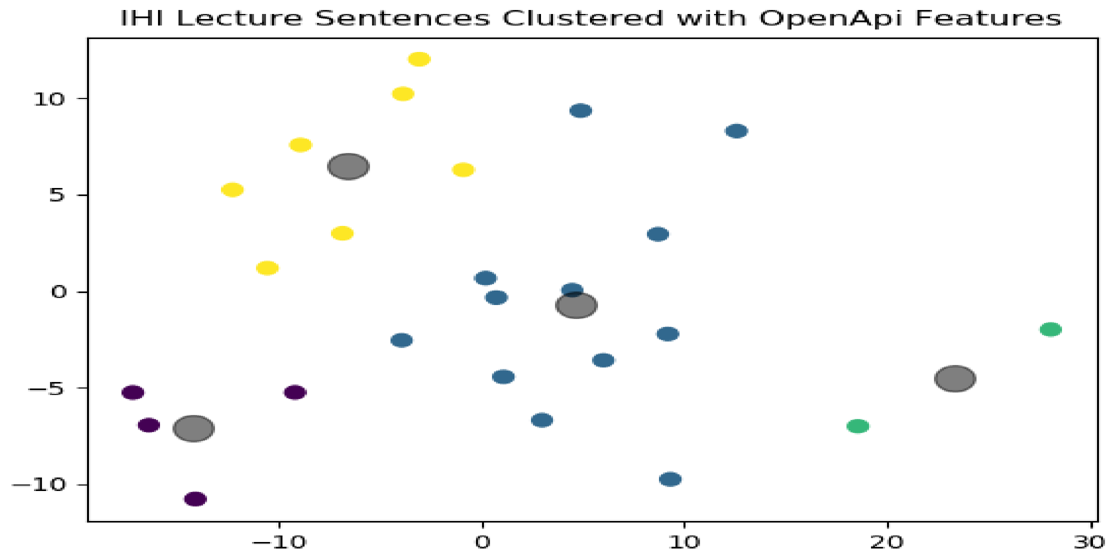

Figure 3-5: IHI GPT-2 embeddings

3.7 Clustering Embeddings

Finally, after embedding the N-2 layer, prepare the N x E matrix for merging. From the user's point of view, they might provide a parameter K, which might represent the amount of clusters and requested sentences for the ultimate summary output. Throughout experimentation, each K- suggests that and mathematician Mixture Models were used for clustering, utilizing the Scikit Learn library’s implementation. Thanks to models’ terribly similar performance, K-Means was finally elite for clump incoming embeddings from the BERT model. From the clusters, the sentences nearest to the centroids were selected for the ultimate summary [45].

3.8 Results

This section focuses on the results of the BERT model and compares the results with other methods such as TextRank. Since there is no summary of the golden truths of the lecture, no indicators other than manual comparison and cluster quality were used, which was discussed in detail. Some of the initial flaws found in the BERT lecture summary are the same as other methods, such as B. Full summary of large lecture. The complexity of dealing with contextual words and spoken language in speech scripts is more common in lectures.

3.9 Model Weaknesses

For large lectures categorized as having 100 or more sentences, the challenge is to make a small number of sentences sufficient to represent the entire lectures. If the percentage of sentences to be summarized is higher, more context is preserved, which makes reading easier. Understand the user's summary. One hypothesis for solving the big reading problem is to include a few sentences close to the center of gravity in the group. This provides more background information for the resume and improves the quality of the output. This also avoids having to add more clusters, which may have fewer representatives, depending on the convergence of the centroid. The problem with this method is that it directly conflicts with user relationship parameters, adds more suggestions than requests, and reduces user experience. Therefore, this method is not included in the service. Another disadvantage of the current method is that sometimes sentences containing words that require additional context are selected, such as "this", "these", "these" and "also". Although the brute force solution is to delete sentences that contain these words, usually such changes have greatly reduced the quality of the summarization. Taking a lot of time, one possible solution is to use NLTK to find parts of speech and try to replace pronouns and keywords with your own keywords. Initially I tried to do this, but some readings include contextual words mentioned in two or three sentences in the past, so it is difficult to determine which element is actually the actual context [45].

3.10 Reinforcement Learning – TD(0)

In an intensive learning course, the content is organized as a dialogue between the two authors. This leads to another problem in summarizing the content of the dialogue. The following are examples of BERT and TextRank, summarizing this content of the TD(0) conference. By reducing the supply from 40 to 5. In this example, you can see that the BERT model focuses on the advantages of this accumulation The maximum probability corresponds to TD (0). It can also correctly select the TD(0) definition and merge the abstract of the correct abstract data. Although TextRank contains the word Maximum Probability, these sentences are quite random, which makes TD(0) difficult to understand from this content.

3.11 BERT Output

Okay, this is the rule we want to call the TD(0) rule. The name of the rule is different from TD(1). So the coincidence here, at least what we have been talking about, is that if we are in certain states of St-1 and have passed, we don't know which state we will be in, so we really assume that we are waiting for us to get The next reward state of plus the estimated present value of the next reward state. That is Assume that the probability of the maximum likelihood estimation is as long as the probability that the data shows before the next state is converted to the state.

3.12 DistilBERT

- **Student architecture**

Within the gift work, the student DistilBERT has an equivalent general design as BERT. The token type embeddings and also the partaker are removed whereas the amount of layers is reduced by an element of 2. Most of the operations utilized in the electrical device architecture (linear layer and layer normalization) are extremely optimized in modern algebra frameworks and our investigations showed that variations on the last dimension of the tensor (hidden size dimension) have a smaller impact on computation potency (for a set parameters budget) than variations on different factors just like the variety of layers. therefore we have a tendency to target reducing the amount of layers [47].

- **Student initialization**

Additionally to the previously delineate improvement and branch of knowledge choices, a vital element in our coaching procedure is to seek out the proper format for the sub-network to converge. Taking advantage of the common spatiality between teacher and student networks, we have a tendency to initialize the coed from the teacher by taking one layer out of two [47].

3.13 A Visual Guide to Using BERT for the First Time

In recent years, the development of machine learning models for processing languages has accelerated rapidly. This advancement left the research laboratory and began to use some leading digital products. A good example is the recent announcement that the BERT model is now the main force behind Google search. Google believes this move (or progress in understanding the natural language used in search) is "the biggest improvement in the past five years, and one of the greatest achievements in search history." This following information is a simple guide on how to use the BERT option to classify sentences. This is a very simple example, but sufficient to show some key concepts [48].

3.13.1 Sentence Sentiment Classification

The goal is to create a model that will contain a sentence and 1 (indicating that the sentence has a positive feeling) or 0 (indicating that the sentence has a negative feeling). The model contains two models:

● DistilBERT processes the quotes in the following models and conveys you from them Some information obtained. DistilBERT is an open source version of BERT developed by the HuggingFace team. This is a lighter and faster version. BERT roughly corresponds to its attributes.

- The following model is the basic logistic regression model of scikit Learn, which uses the output processed by DistilBERT and classifies sentences as positive or negative (1 or 0).

The data that we pass between the two models is a vector of size 768. We can think of this vector as an insertion of sentences that can be used for classification.

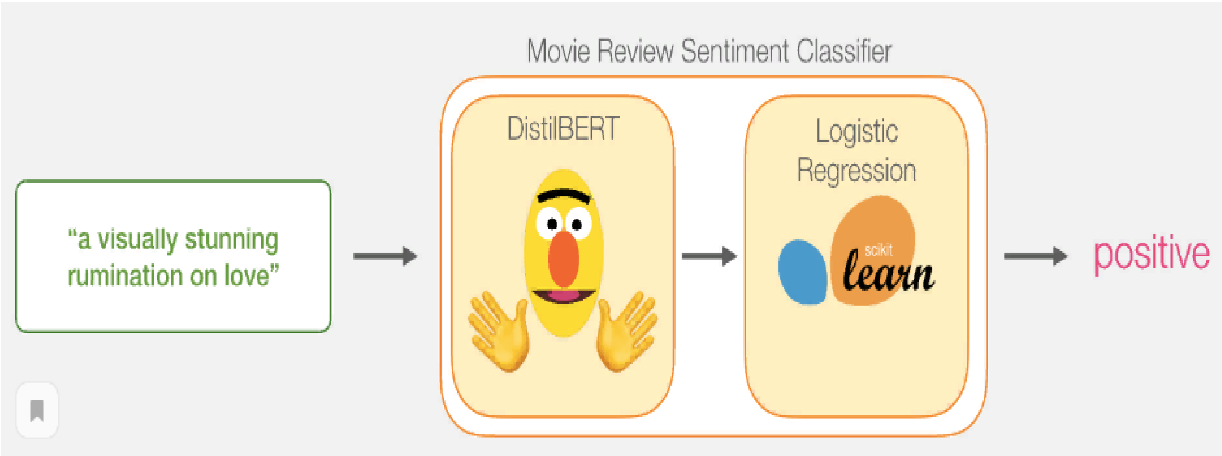

Figure 3-6: Movies Review Sentient Classifier

3.13.2 Model Training

Although we are using two models, we will only run one logistic regression model. For DistillBERT, we use a model that has been established and has English commands. However, this model is wrong. Send to the proposal for classification. However, we can indeed classify sentences according to the overall goal on which BERT is based. This is especially true for the BERT output in the first position (associated with the [CLS] token). The second learning objects BERT the classification of the next sentence. This goal obviously trains the model to encapsulate the meaning of the entire sentence at the exit of the first position. And the implementation of DistilBERT and the previous training version of the model [48].

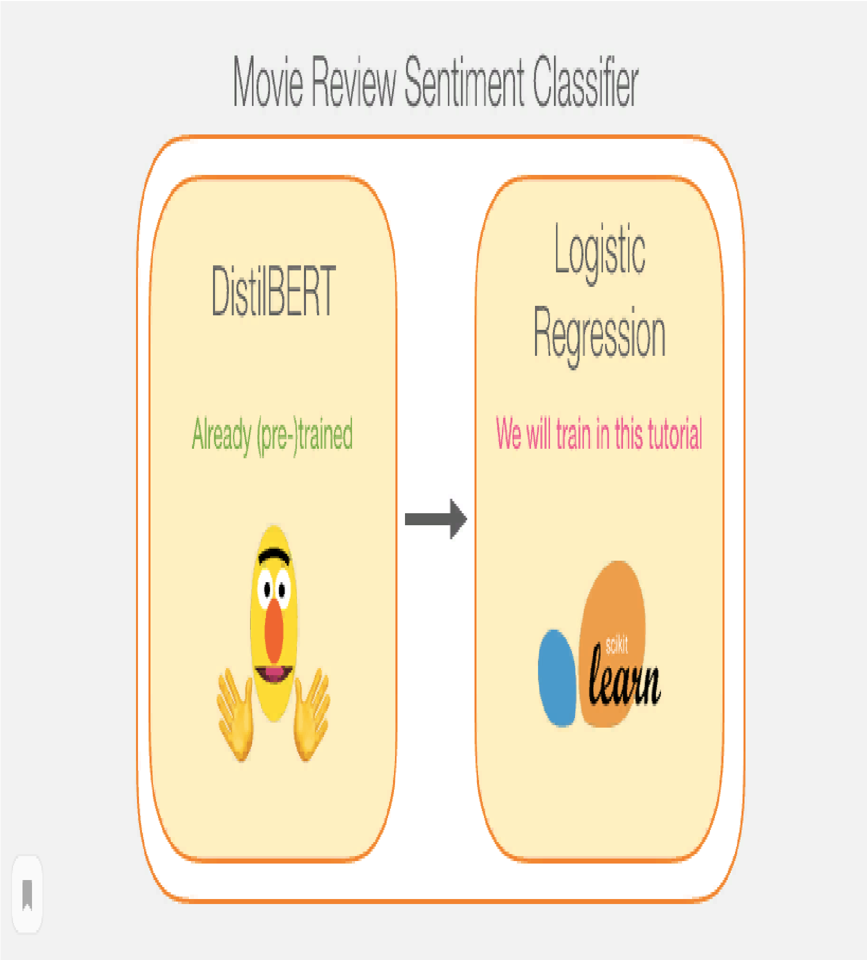

Figure 3-7: Model training for logistic regression

3.13.3 Tutorial Overview

For generate sentence embeddings , we need first to use the trained of DistilBert for 2,000 sentences(for example).

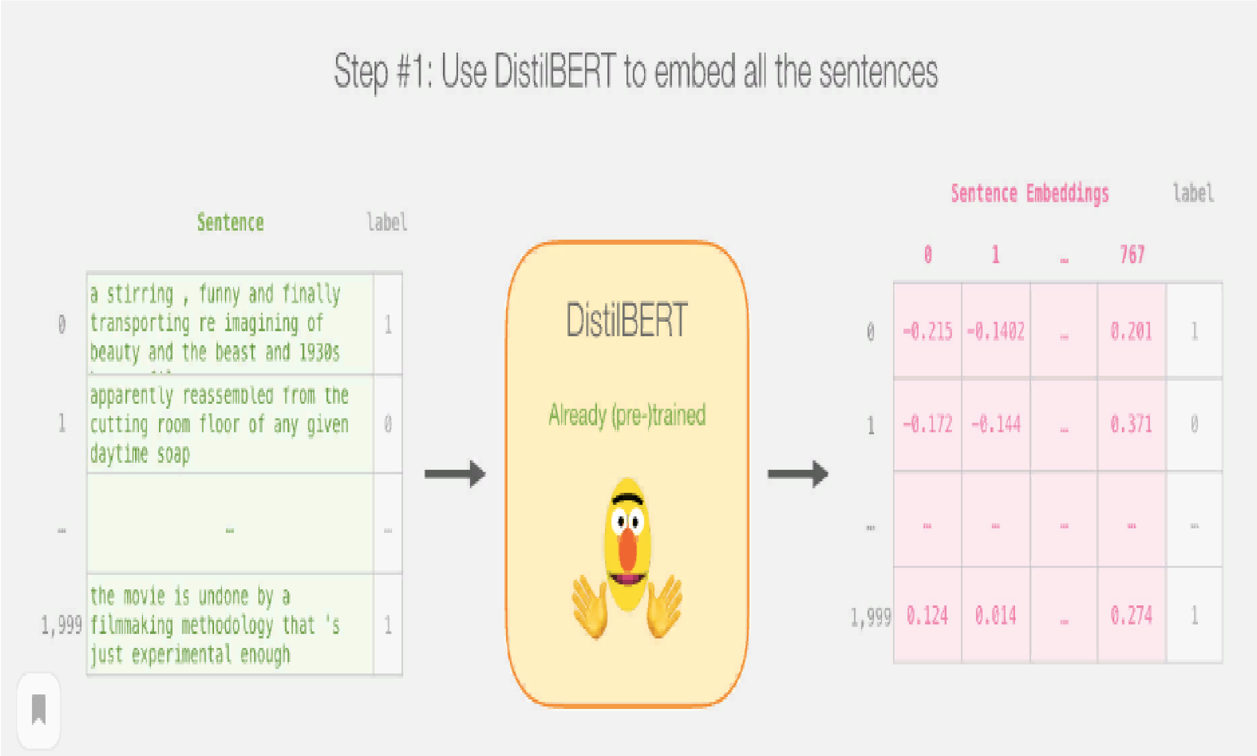

Figure 3-8: Using DistilBert to embed all the sentences

After this step We will not touch distilBERT. It's all Scikit Learn from here. We do the usual train/test split on this

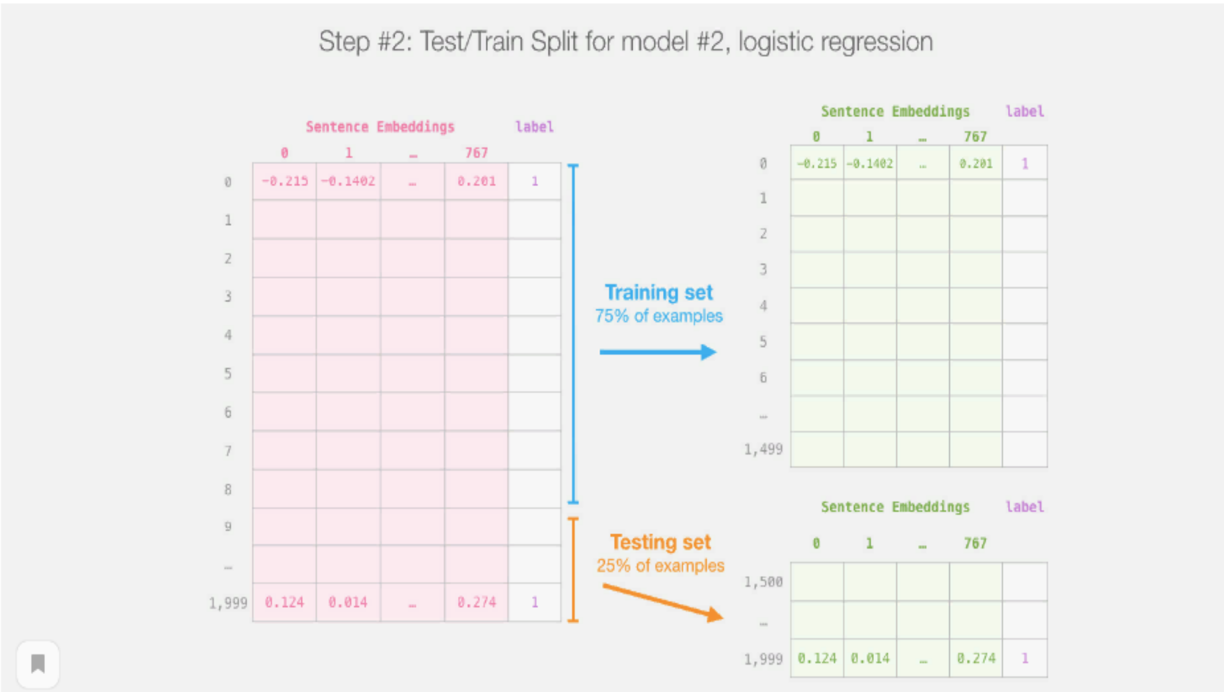

Figure 3-9: Test/Split for model logistic regression

By dividing training/testing into distilBert inference (model 1), it is possible to create a data set, train it well, and evaluate logistic regression (model 2). Note that sklearns train/testplit will actually shuffle the examples before splitting. Not only is the first one needed. Examples that appear in the data set account for 75%.

So we are going to train the logistic regression model on the training set as shown in this photo:

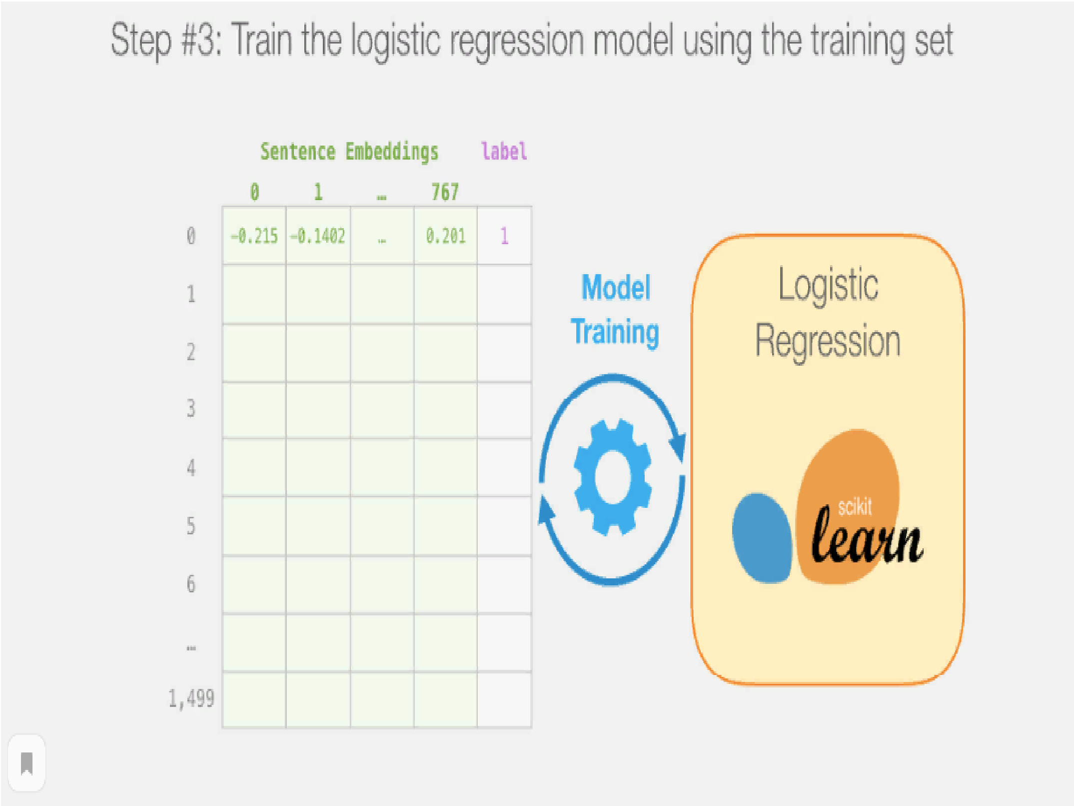

Figure 3-10: Train the logistic regression model using the training set

3.13.4 How a single prediction is calculated

Firstly , Before we dig into the code and giving an explication for how to train the model, let's look some information at how a trained model calculates its prediction.

Let's try to classify the sentence "a visually stunning rumination on love". The first step is to use the BERT tokenizer to first split word into the tokens. Then, we need to add the special tokens needed for our sentence classifications (these are [CLS] at the first position, and [SEP] at the end of the sentence).

The third step the tokenizer does is to replace each token with the special id of the sentences from the table of embedding which is a component we get with the trained model.

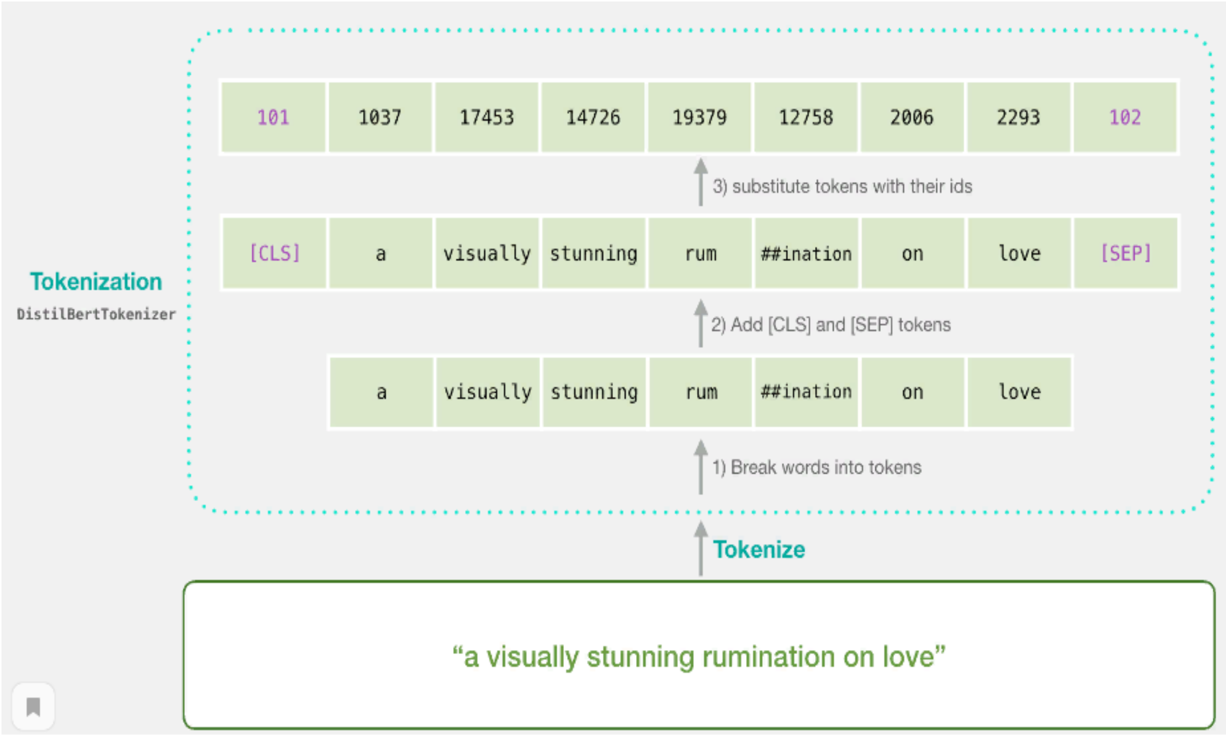

Figure 3-11: The archetecture of DistilBertTokenizer

Remember you also that the tokenizer does all these steps in a single line of code:

```
[   tokenizer.encode("a visually stunning rumination on love", add_special_tokens=True) ]
```

Our input sentence is now the proper shape to be passed to DistilBERT.

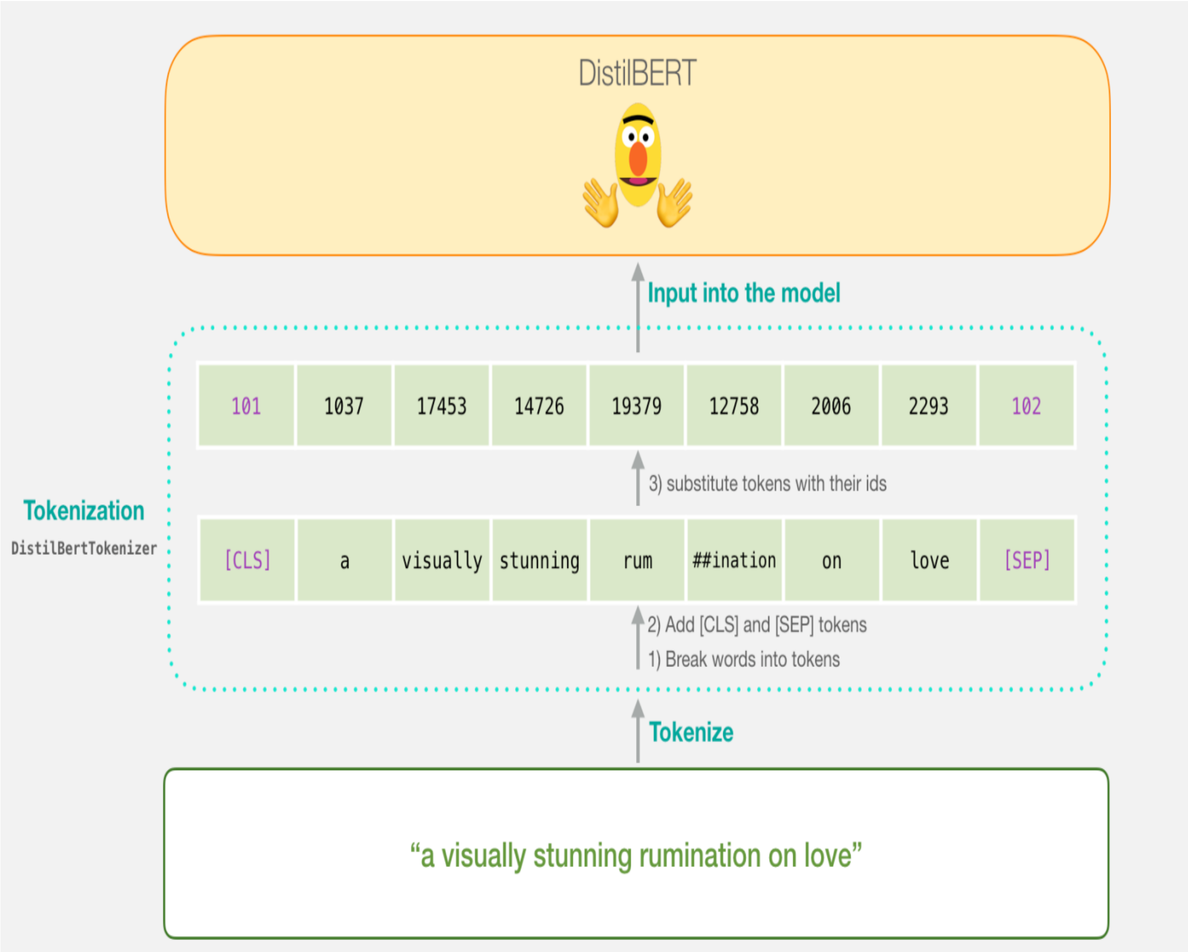

Figure 3-112: Input into the model DistilBert

3.13.5 Flowing Through DistilBERT

We need to pass the vector of imput through DistilBERT, they have the same fonction or the same work just like BERT. The output would be a vector for each input token. each vector is made up of 768 numbers (floats).

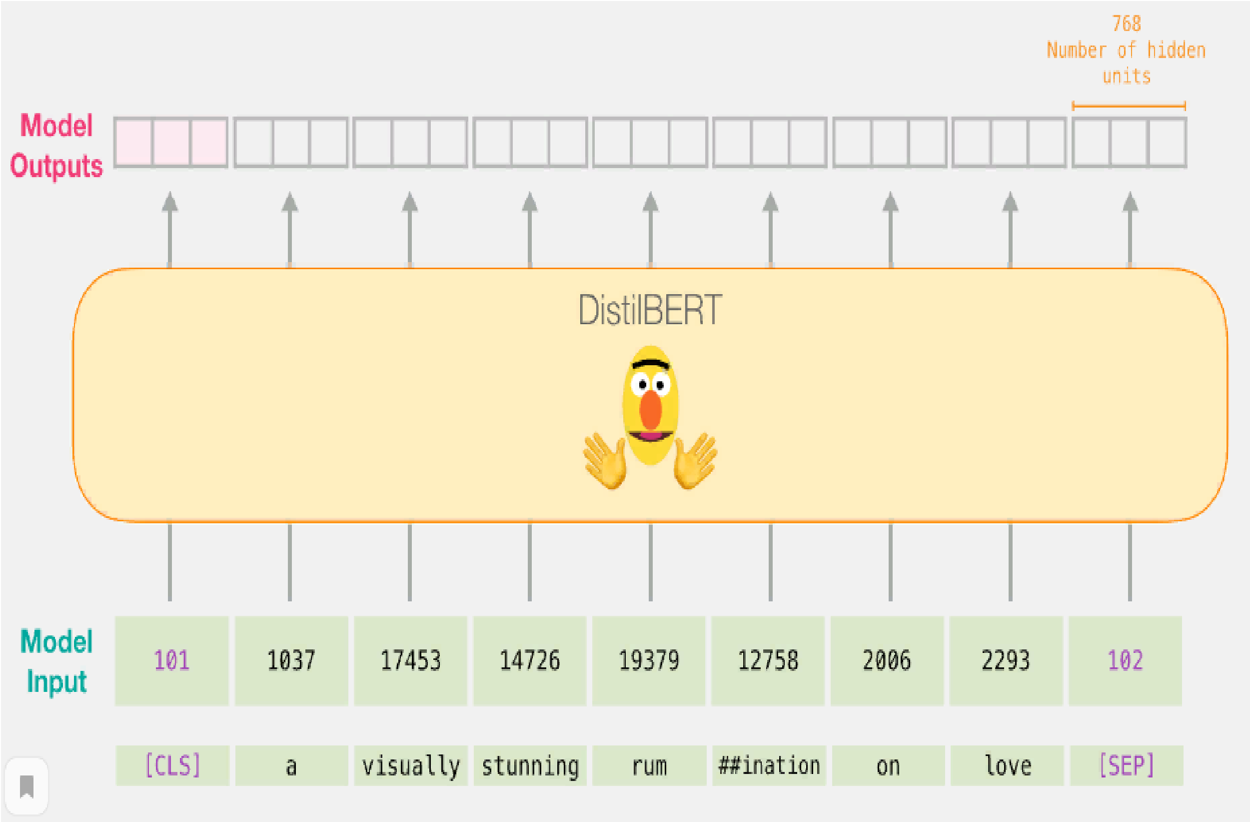

Figure 3-13: Flowing Through DistilBERT

Because this is a sentence classification task, we ignore all except the first vector (the one associated with the [CLS] token). The one vector we pass as the input to the logistic regression model.

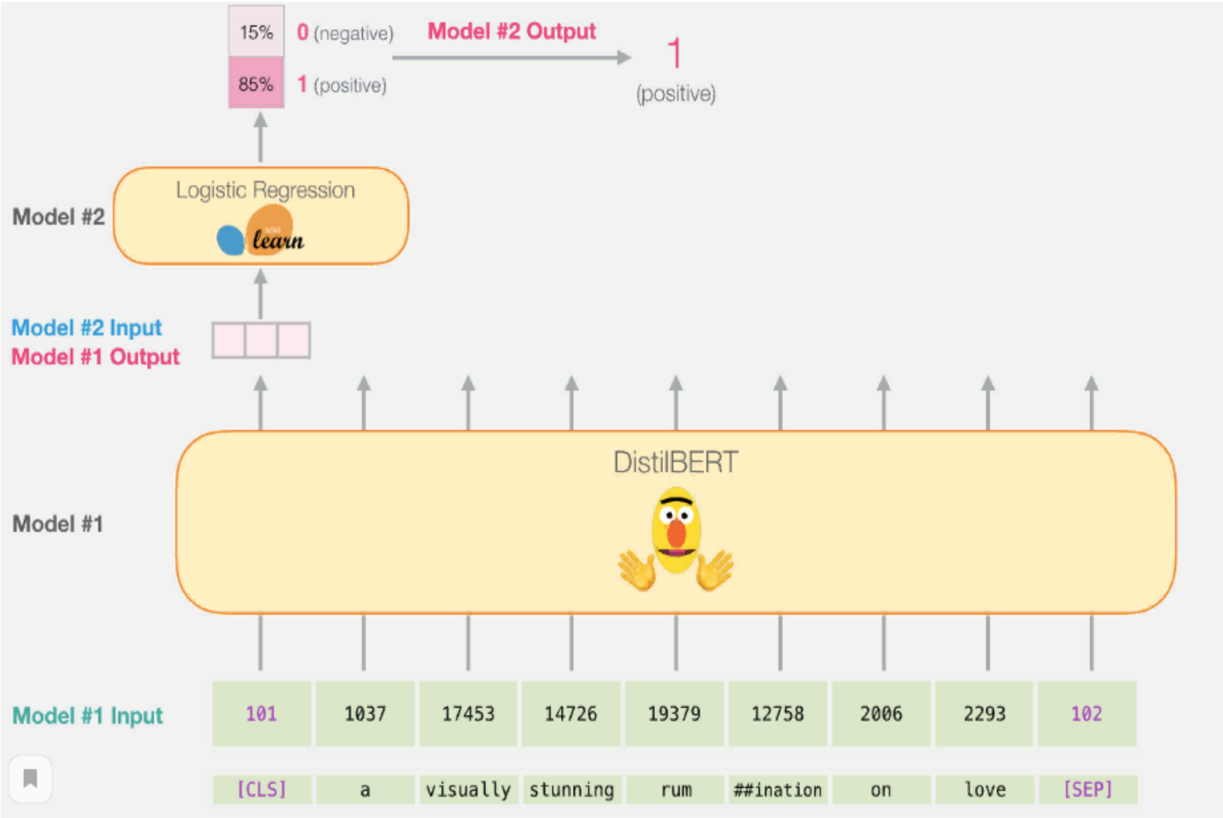

Figure 3-14: Result of output model logistic regression[48]

From this step, it's the job of model logistic regression to classify this vector based on what it learned from its training phase.

3.14 Conclusion

In the past few years, researchers in Automatic extractive summarization have been trying to solve this problem by conducting research with good results. However, most methods need to be improved because they use older natural language processing models. By using a cutting-edge deep learning NLP model called BERT, we are continuously improving the quality of summaries that combine context with key sentences, and this method is not like an old method like TextRank. The conference summary service uses the BERT model and the DistilBERT process to generate summaries for users according to their settings. Automatic aggregation and extraction services are not ideal, but compared to traditional methods, it provides higher.

Chapter 4

4 Implementation and Evaluation

4.1 Introduction

Evaluating the quality of automatically produced summaries is subjective (there is no "perfect" summary), rather difficult and remains an open problem. To make a good summary it has to be comparable and evaluated to so called good summary. To do so, there are a variety of possible bases for the comparison of summarization systems performance. Therefore, the system summary can be compared to the source text, to a human-generated summary or to another system summary. In terms of evaluating summaries on sentence level can be done semi-automatically by measuring content overlap with precision, recall, and F1 measure. An extracted sentence is considered acceptable if the same sentencewas extracted in a reference summary. This process cannot be automatized because reference sum- maries are created by human judges. Some semi-automatic evaluation methods used nowadays are: BERT and PCA that will be discussed and used as the method of evaluation in this chapter .In What follows of this section we will show the progress and the results of our text summarization approaches in the test phase, which will help us to consider the possible improvements. We start by presenting and describing the methods used in the evaluation. And then we present the evaluation result for the extractive approach .In a further experiment, we will compare the obtained result by TextRank algorithm and an extractive fine-tuned version of BERT transformer.

4.2 Software Configuration for implementation

Cloud computing services cover an enormous vary of choices now, from the fundamentals of storage, networking, and process power through to linguistic communication process and computer science as well as standard office applications. Prettymuch any service that doesn’t require you to be physically close to the pc hardware that you are mistreatment will currently be delivered via the cloud. In the recent years the only solutions for playacting Brobdingnagian computations, was mistreatment supercomputers that use arrays of CPUs to do many computations in parallel. Therefore in this work we have a tendency to use Google Colab. To make the executions faster , it has free Jupyter notebook environment that runs completely on Google Cloud and uses Google cypher Engine backend for all the computations. It is very simple to write and execute the code because it doesn't need any installation on your native machine. All we want is a web browser to access the colaboratory. Colab offers each GPU and TPU hardware accelerators for free with our Google free tier account up to a certain computational power. That is, we can change the runtime type while executing our code. The free GPU offered is 1xTesla K80, 2496 CUDA cores and free TPU includes TPU v2, eight cores, approx twelve GB RAM with a most RAM limit up to 36 GB, Consequently for our implementation we have used the GPU accelarator.

4.3 Experimental Results and Evaluation

Experiments are conducted on any lecture or text ... The lectures were collected from Wikipedia and newspaper. The selected articles cover different topics: art and music, environment, politics, sports, health, finance and insurance, science and technology, tourism, religion and education. We evaluated summarization quality automatically using BERT model (DistilBert) [47] and PCA.. We will report a general picture that contain the steeps of implementation .

4.4 General picture for implementation :

we will write about the steps that are implemented in our notebook and our work .

- we preprocess a Text using DistilBERT for processes the sentence and passes along some information it extracted from it on to the next model. DistilBERT is a smaller version of BERT developed and open sourced by the team at HuggingFace.
- using DistilBert to embed all sentences We will first use the trained distilBERT to generate sentence embeddingsgive it to BERT to get embeddings .
- reduce dimension with PCA.
- plot points and allow user analyze it.
- make clusterization .
- select sentences from the center .

4.4.1 Importing the tools of the trade

```
[        import numpy as np
         import pandas as pd
         import torch

         !pip install transformers

         import transformers as ppb # pytorch transformers ]
```

after that we can just import the text directly into a pandas dataframe .

using and execute this code   :

```
text = """ our text """
```

4.4.2 Importing the NLTK library

We will see a simple NLP-based technique for summarizing a text. We will not be using any machine learning libraries in this article. Instead, we'll use the Python NLTK library to summarize Wikipedia articles or any text[46] .

```
import nltk.tokenize
nltk.download('punkt')
all_sentences = nltk.tokenize.sent_tokenize(text)
```

also we can get a text with all sentences

```
print(f"Now we got a text with {len(all_sentences)} sentences")
```

Now we got a text with 81 sentences.

4.4.3 Importing pre-trained DistilBERT model and tokenizer

We will currently tokenize the text. Note that we're going to do things a bit otherwise here from the instance above. The example higher than tokenized and processed just one sentence. Here, we'll tokenize and method all sentences along as a batch[48].

```
[  model_class, tokenizer_class, pretrained_weights = (ppb.DistilBer
tModel, ppb.DistilBertTokenizer, 'distilbert-base-uncased') ]

[   tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
model = model_class.from_pretrained(pretrained_weights)   ]
```

and this example of pretrained model/tokenizer execution :

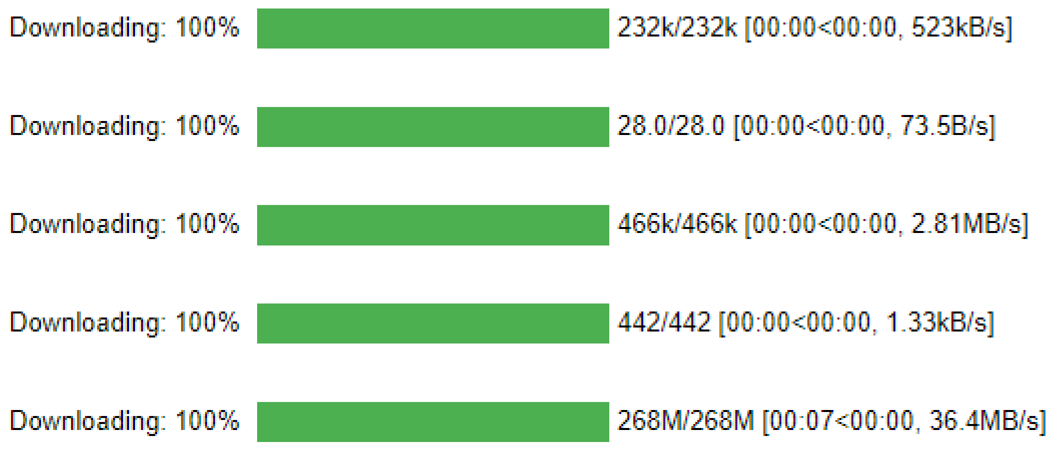

Figure 4-1: pretrained model / tokenizer execution

4.4.3.1 Tokenization

```
[tokenized = df[0].apply((lambda x: tokenizer.encode(x, add_special_
tokens=True)))]
```

```
[input_ids = tokenizer(sentences, add_special_tokens=True, return_tensors="pt
", padding="longest").input_ids ]
```

This turns each sentence into the list of ids.

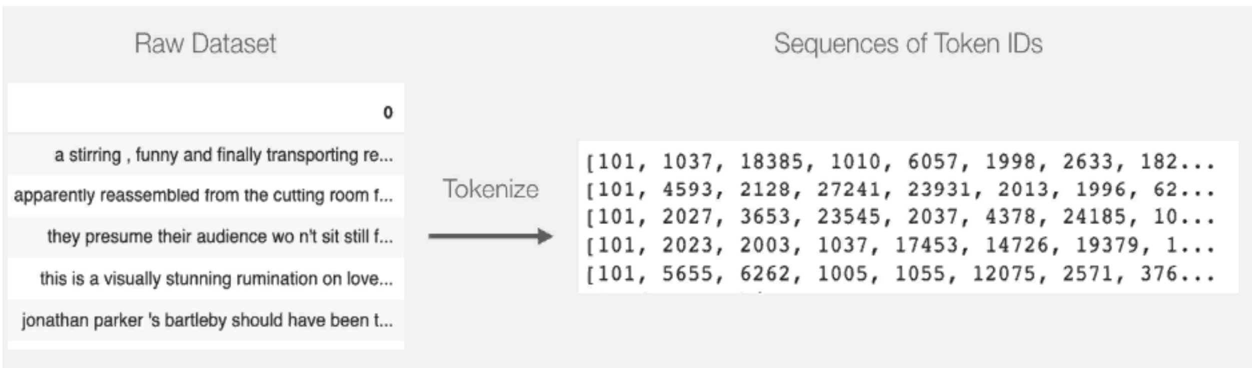

Figure 4-2: Representation sentences into the list of ids

The text is presently an inventory (or pandas Series/DataFrame) of lists. Before DistilBERT can method this as input, we'll have to be compelled to create all the vectors a similar size by artifact shorter sentences with the token id 0. you'll be able to check with the notebook for the padding step, it’s basic python string and array manipulation.

After the padding, we've a matrix/tensor that's able to be passed to BERT:

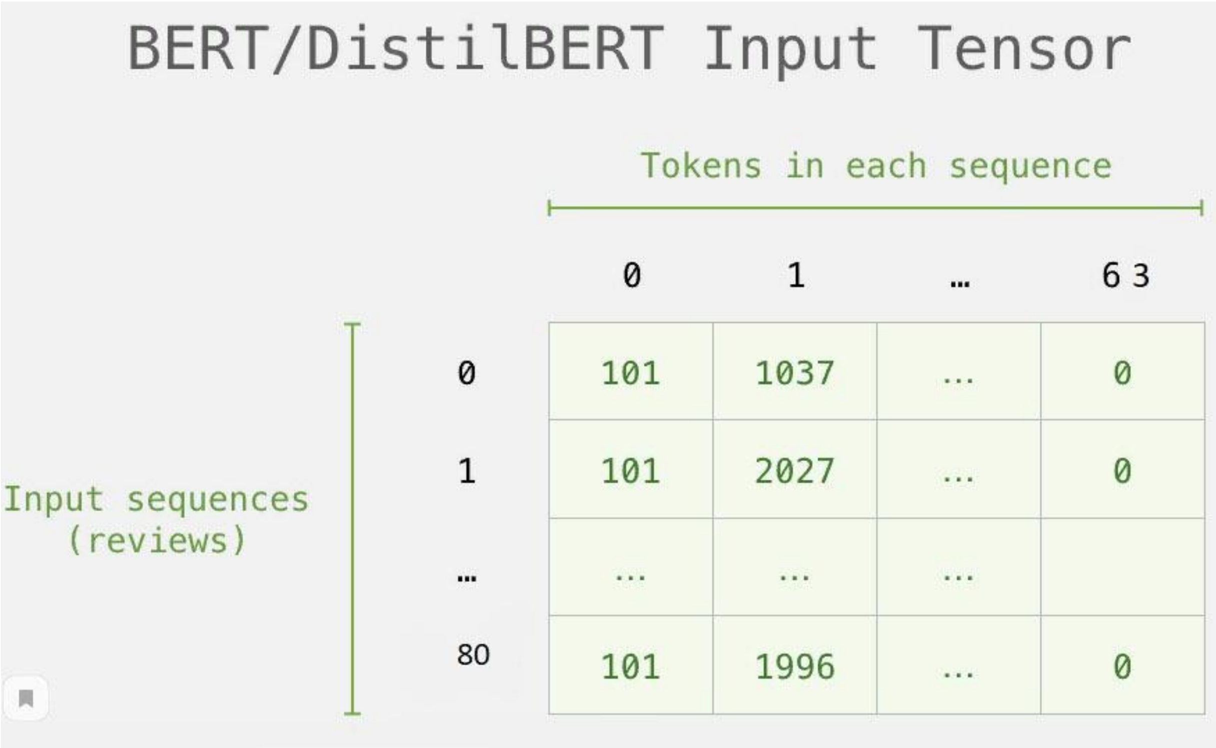

Figure 4-3: BERT/DistilBert Input Tenzer

4.4.3.2 Processing with DistilBERT

We will create an tensor input , out of the padded this token matrix, and ship that to DistilBERT.

```
[input_ids = torch.tensor(np.array(padded))

with torch.no_grad():
    last_hidden_states = model(input_ids)
```

```
last_hidden_states[0].size() ]
```

After completing this step, last_hidden_states will contain the DistilBERT output. This is a form of tuple (number of examples, maximum number of tokens in the sequence, number of hidden units in the DistilBERT model). In our example, they are 81 (because we are limited to 81 examples), 63 (this is the number of tokens in the longest 81 example sequence), and 768 (the number of hidden units in the DistilBERT model)..

Result of execution : `torch.Size([81, 63, 768])`

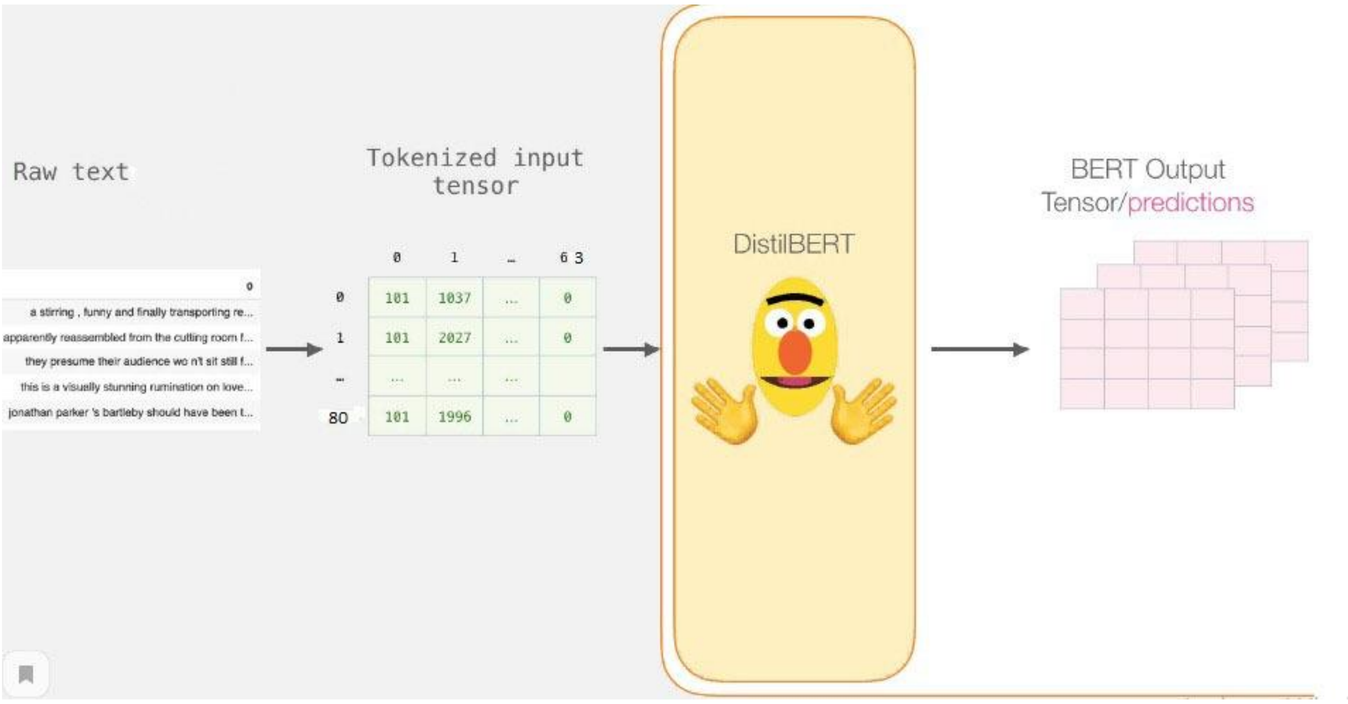

Figure 4-4: Processing with DistilBERT

4.4.3.3 Unpacking the BERT output tensor

Let's take this 3-D output tensor. we are able to initial begin by examining its dimensions:

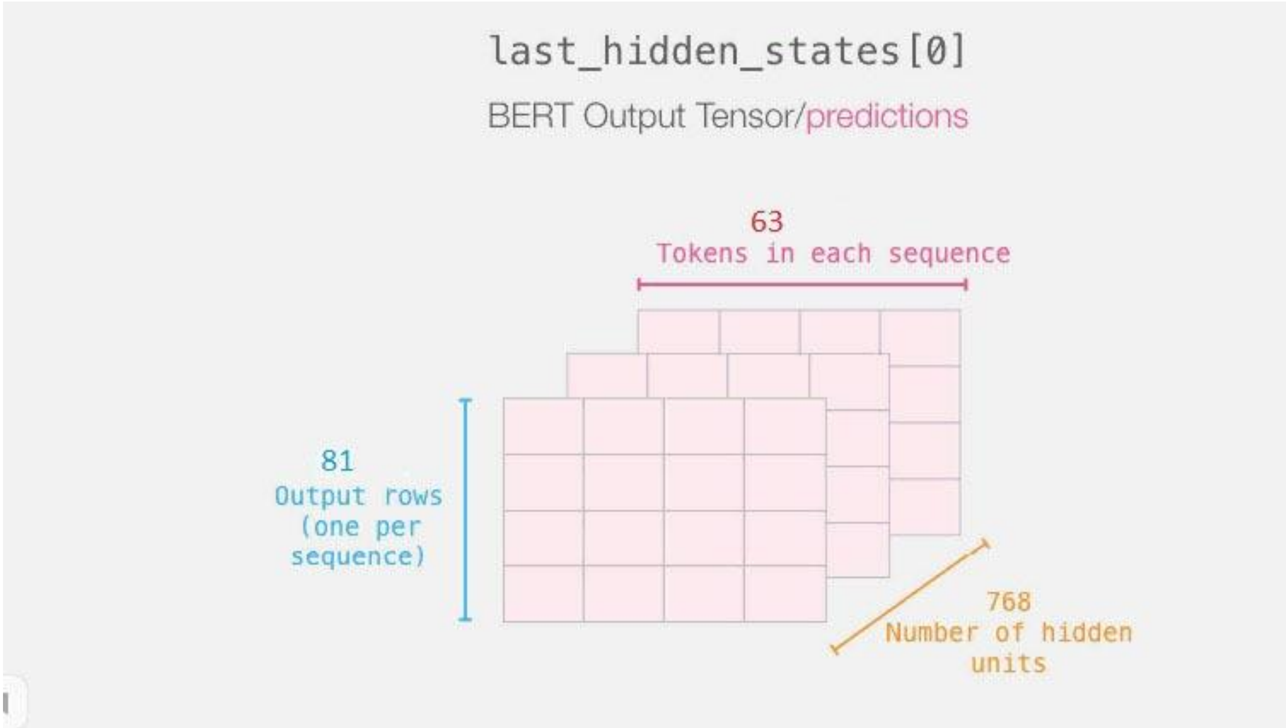

Figure 4-5: BERT output tensor/predictions

4.4.3.4 Recapping a sentence's journey

Each row is related to a sentence from our text. To recap the process path of the primary sentence, we are able to think about it as trying like this:

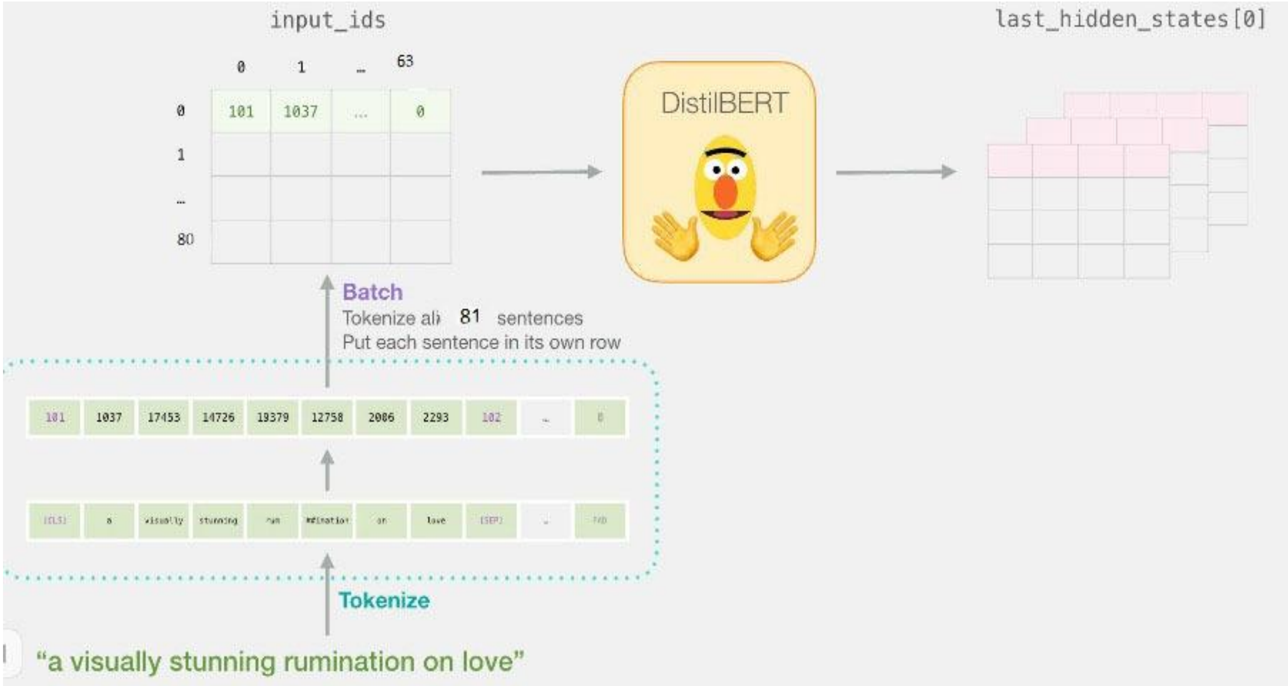

Figure 4-6: Presentation of Recapping a sentence's journey

4.4.3.5 Slicing the important part

For a classification sentence, we tend tore solely only curious about the BERT's output for the [CLS] token, therefore we choose this slice of cube and discard everything else.

However, we tend to cover this 3D tensor and obtain the 2D tensor we are interested in.

```
[features = last_hidden_states[0][:,0,:].numpy()]
```

Now, Features is a two-dimensional array of numbers that contains sentences to be embedded in the text.

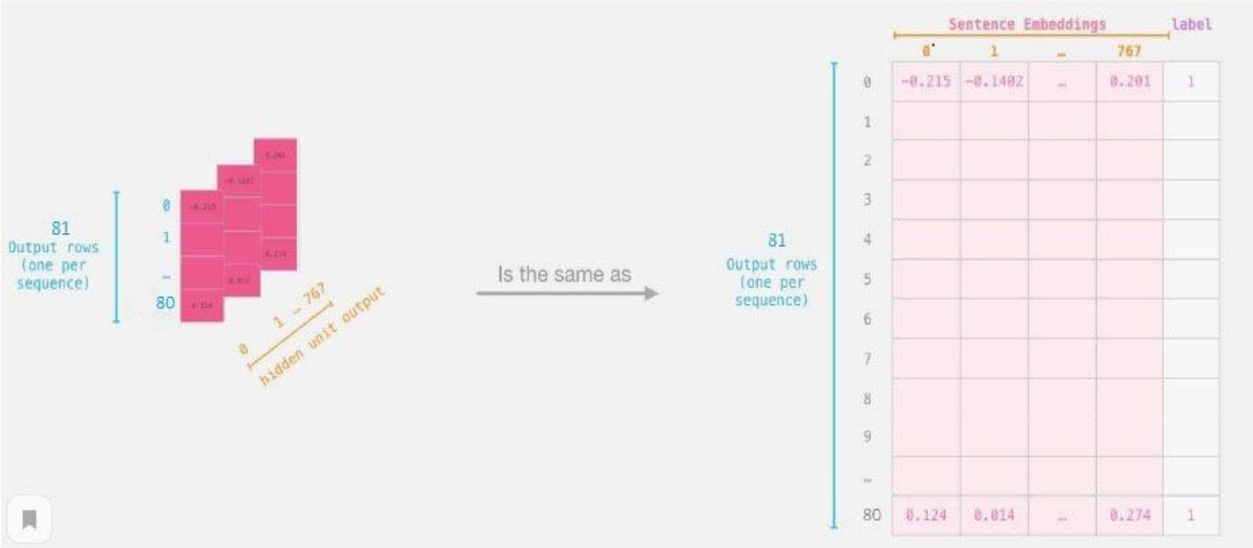

Figure 4-7: Process of Sentence classification

The tensor we have a tendency to sliced from BERTs output.

4.4.3.6 A text for Logistic Regression

Now that we've got the output of BERT, we tend to have assembled the text we'd like to coach our supplying regression model. The 768 columns are the features, and therefore the labels we simply get from our initial text.

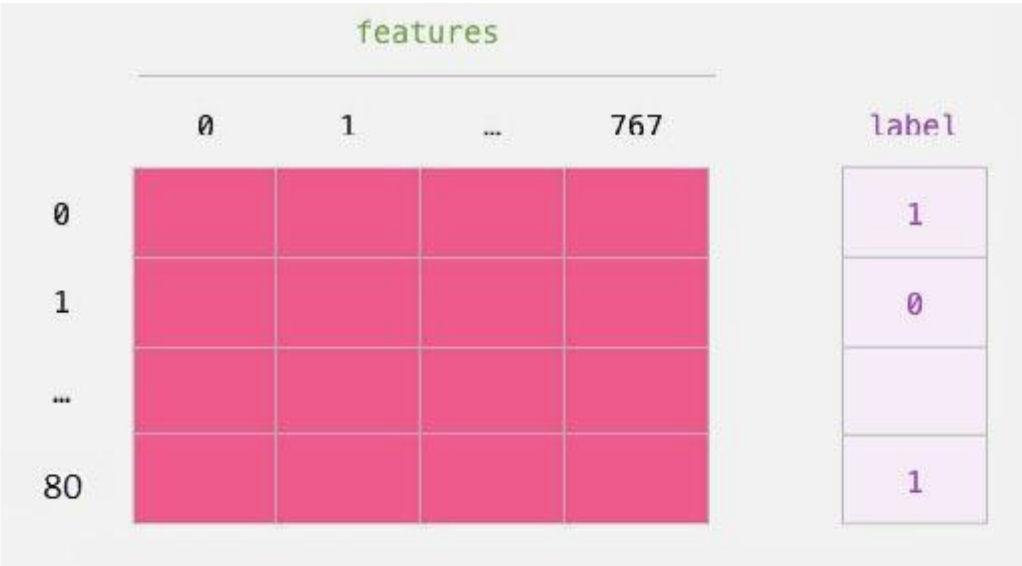

Figure 4-8: Output of bert contain a text for logistic regression

We usually use the labeled text for training logistic regression. The option is the BERT output vector of the [CLS] token (position 0) we cut out in the figure above.

Each row corresponds to a sentence in each column of our text. Corresponds to the output of the hidden module of the direct neural network in the higher electrical module of the Bert/DistilBERT device [48]. After the traditional train /test of ML, we will declare and train our logistic regression modeltext.

```
[labels = df[1]
train_features, test_features, train_labels, test_labels = train_te
st_split(features, labels)]
```

Which splits the dataset into training/testing sets:

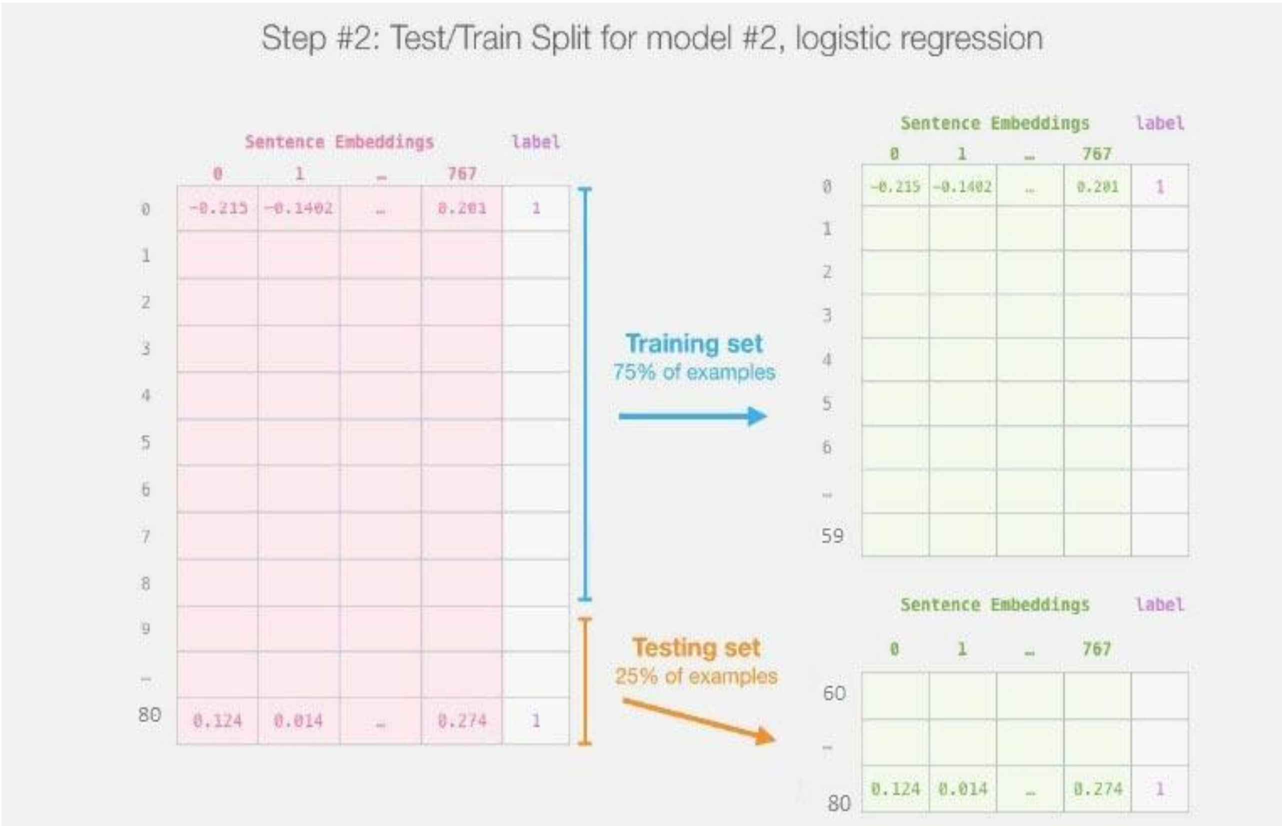

Figure 4-9: Text/Train split for model logistic regression

Next, we will train this Logistic Regression model on the training set.

```
[lr_clf = LogisticRegression()
lr_clf.fit(train_features, train_labels)]
```

now our model is already trained, we can score it against the test set:

```
[lr_clf.score(test_features, test_labels)]
```

Which shows the model achieves around 81% accuracy.

4.4.3.7 Reduce dimension with PCA

Now we run PCA to reduce 768 dimension to just 2 dimension and also In this section we will reduse the dimension of our sentences using PCA , as shown in the following picture.

	x	y	sentence
0	-3.934732	-7.500238	We often talk about "The Perl Community", but ...
1	-6.198814	-0.185216	Instead what we have is a loose, and at times ...
2	3.063171	-17.623810	Over the last few weeks I've been thinking and...
3	8.840539	-13.440082	This is not me trying to tell you how things a...
4	-3.155584	0.416067	A community is two or more people who come tog...
...	...	...	...
76	23.293865	5.277453	In a grassroots community of volunteers, like ...
77	4.807804	-13.350680	We'll then start thinking about the boundaries...
78	-9.726598	-3.704862	I hope the process will naturally bubble up, a...
79	9.534457	-5.476137	We may also find that in working this stuff ou...
80	-15.108610	5.583264	They could still appear on our map, but labell...

81 rows × 3 columns

Figure 4-10: Reduce 768 dimension to just 2 dimension using PCA.

4.4.3.8 Plot points and allow user analyze it

after reduse dimensoin we can finally plot a point that represent the sentences for our summary .this picture explain the idea for our researche that contain the representation of summaries as a set of sentences-points , that correspond to sentences of the text.

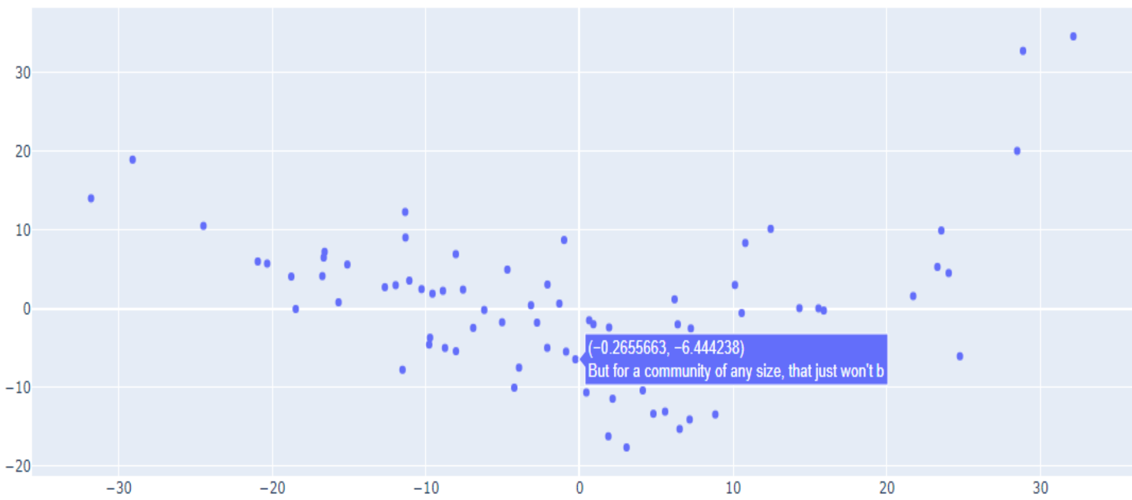

Figure 4-11: Presentaion of text summary

4.5 Conclusion,

This chapter illustrates briefly the process of implementing the adopted systems by specifying the software configurations , platforms , the environment features used, and the approaches we took to achieve our purposes We presented the explaication of the steps that are implemented in our notebook. and experiental results where we notice that performance with the classification method on extractive approach using the process of DistilBert , also we presented the summaries as a set of sentence-points , we produced the embedding for clustering using K-Means model. As a future work, we consider to enhance the informative scoring technique by implementing Bert model or DistilBert model , Can process the sentence and pass some information extracted from it to the next model. and results in a better context vector, increase the training text size and build the model and Implement pointer-generator networks and coverage mechanisms.

Chapter 5

5 Analysis of the Social Content of the Orderand the Socio-political Conditions for the Implementation of the Project

5.1 Introduction:

We can currently access a lot of information quickly, but most of the information is redundant, trivial, and may not convey its intended meaning. As a result, the use of automatic text summaries has become more common and relevant, which can extract useful information from lack of irrelevant data. ... important. Can improve the readability of documents, reduce the time required to search for information, and provide more This chapter introduces the different types and methods of automatic text summarization, then introduces various natural language processing techniques and methods, and discusses the working principles of abstract grading.

5.2 Automatic summarization

A summary is short text that contains important information from the original of text l and usually too little. Automatically summarize text: This is the shortening process of text documents using resume creation software. The abstract provides the most important or relevant information in the original content. When someone needs to manually summarize a very long content quickly and accurately, the abstract becomes crucial. However, when this operation is done through a computer, we call it automatic text summation (ATS).

The Text summarization divided into two important categories: extractable summarization and abstract summarization.

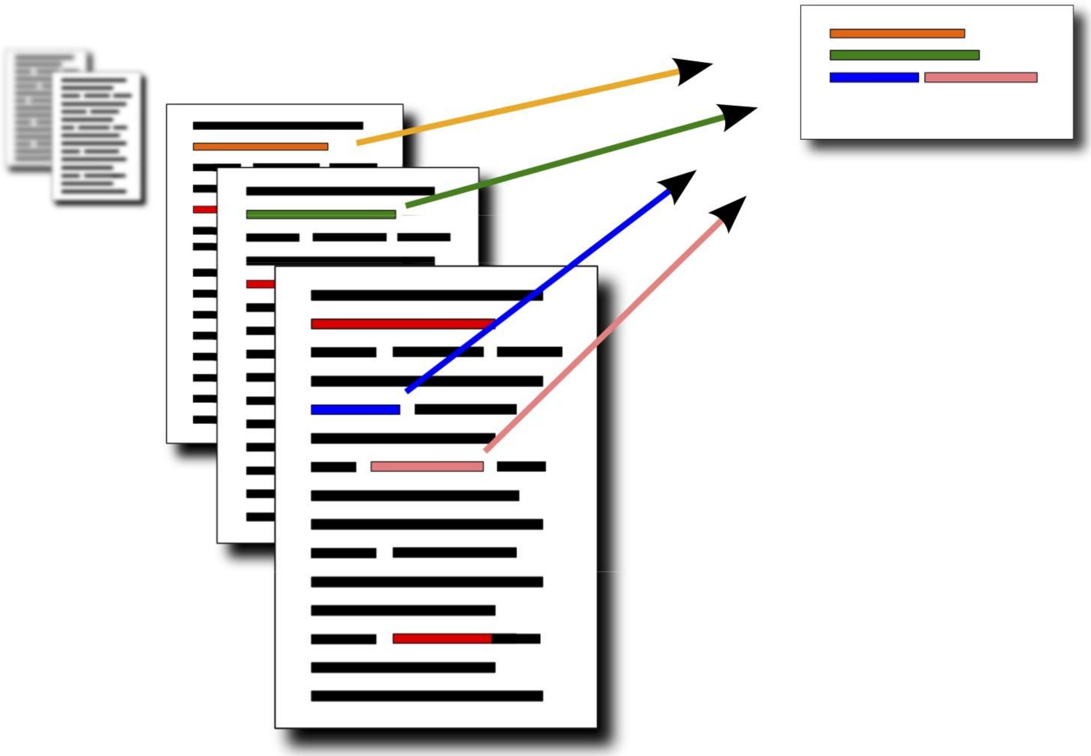

Figure 5-1: Over View Of Text Summarization

Our research involved in helping to understand what the text is about, Also Modern systems achieve results comparable to humans. There are many types of summaries and they may be adjusted by many parameters, so it is possible to give users a special interface for to experiment with summaries of a text.

5.3 Target field and motivation of this work

With the help of background knowledge and accompanying work, the missing part of existing research and projects is the conference summary service, which students with custom conference sizes can use to take advantage of the latest deep learning research. This fact has inspired the development of conference summary services. , This is a cloud service based on the BERT model that is used to dynamically adjust the size of the conference summary.

5.4 A Survey on ATS

The availability of information on the Internet continues to increase, and it is necessary to conduct in-depth research on automatic text summarization in the NLP (Natural Language Processing) community. In the past half century, different perspectives, different fields and different paradigms have been used to solve this problem. The purpose is to study some of the most suitable methods for single document and multi-document abstraction fields, with emphasis on empirical techniques and extraction techniques.

Some promising methods are also discussed, which focus on the specific details of the issues to be merged. Special attention should be paid to the automatic evaluation of the results system, because future research on aggregation depends to a large extent on the progress of the field. It is required to develop an efficient and accurate merging system. Although abstract research began about 50 years ago, there is still a long way to go in this field. Over time, the focus has shifted from abstracts of academic articles to news articles, emails, announcements, and blogs. Depending on the application, abstract methods and extraction methods were tried. Used to replicate or expand to a larger domain. On the contrary, in large-scale applications, especially in simple applications, simple batch extraction has obtained satisfactory results A summary of several documents. The recent popularity of effective cross-border news feed systems supports this claim. This review focuses on the extraction and generalization methods using statistical techniques. A distinction is made between a document and a combination of multiple documents. Because there is a lot of interesting work to do besides traditional research.

In this area, we decided to briefly discuss some of the technologies we can find Even if they only focus on the small details of the entire final process, rather than building a complete final system, they are also relevant for future research. Finally, we reviewed some of the latest trends in the automatic evaluation of the summary system. It is suggested that the future of this research field depends to a large extent on the ability to find effective methods for automatically evaluating these systems and the ability to formulate objective methods that are generally accepted by the research community.

5.5 The challenges of ATS with AI

One of the main problems of automatic text summarization is to give meaning to specific sentences when constructing the summary. This is because importance is of course subjective; novice chefs may want to find a list of key ingredients from an article on mushroom risotto, while experienced chefs may want to learn new techniques from the same article. Factual errors are unacceptable to users of AI products. Therefore, the industry is only adjusting the mining summary. In the following, I list the main problems of automatic text summarization, which are common to abstract methods and extraction methods. Information Flow. Consecutive sentences must be appropriately related; otherwise, the reader may be separated from the main text.

5.6 Evaluation methods

The aggregate rating method attempts to determine whether a resume is appropriate (and reliable) or useful relative to its source. There are currently two types of evaluation methods. The first is internal (or normative) evaluation, where users evaluate the quality of abstracts through direct analysis. Users can fluently assess the degree of coverage of the key ideas of the statement, or the degree of comparison with the ideal abstract written by the original author or the author of the abstract. None of these measures are completely satisfactory. In particular, a perfect resume is difficult to create and rarely unique. Just as there are many ways to describe events or scenarios, users can create regular or user-centric paragraphs or summaries that they see fit. In fact, empirical evidence shows that people rarely agree on which sentences or paragraphs to include in a resume. The second valuation method is the external valuation method. The user scores the quality of the summary based on the degree of its impact on other tasks.

Table 1. Relevance assessment using summaries, as opposed to full text.

Summary type	Length reduction	Time reduction	Accuracy loss
User-focused	77%	50%	5%
Generic	90%	60%	0%

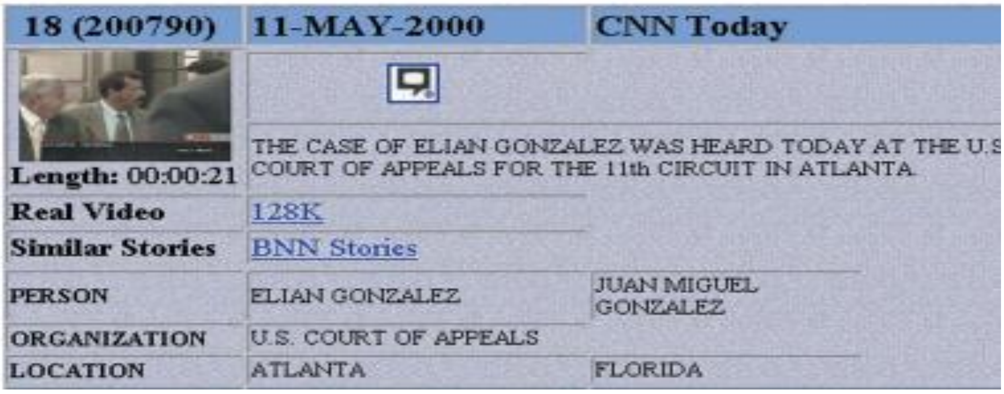

Figure 5-2: Multimedia summarization using the Broadcast News Navigator

Advance modern word processing technology. 9 The plan includes two evaluations. In a session, each user has viewed the user-centric source or summary and must decide whether it is related to the topic. In another session, the user viewed the source or regular source they must choose a topic that they think is related to the document (or choose from several topics proposed), or decide whether it is related to a topic. As shown in Table 1, automatic text summarization is very effective in this relevance score. 77% to 90% of abstracts are discarded as full texts, but the speed is almost twice as fast as abstracts (the accuracy difference of 5% is not statistically significant). Although no specific generalization methods were tested in the evaluation, the 16 generalization systems evaluated used low-level knowledge methods. They differ in their ability to generate user-centric summaries. When acquiring sentences, the most accurate user-centric systems exhibit similar behavior.

5.7 How costly is the ATS failure?

He report of Cortexica claims Thoughtly has raised a total of $2.6M in funding over 2 rounds. Their latest funding was raised on Feb 26, 2015 from a Seed round.

5.8 The future of AI in ATS

Automatic text summarization has played a critical role in helping people obtain key information from increasing huge data with the advantaged development of technology. In the past, few literatures are related to solve the problem of generating titles (short summaries) by using artificial intelligence (AI). The purpose of this study is that we proposed an AI approach for automatic text summarization. We developed an AI text summarization system architecture with three models, namely, statistical model, machine learning model, and deep learning model as well as evaluating the performance of three models. Essay titles and essay abstracts are used to train artificial intelligence deep learning model to generate the candidate titles and evaluated by ROUGE for performance evaluation. The contribution of this paper is that we proposed an AI automatic text summarization system by applying deep learning to generate short summaries from the titles and abstracts of the Web of Science (WOS) database.

5.9 Areas of development

Actually the Areas of life or development is professional, the missing part of existing research and projects is the lecture summarization service, and students with custom lecture sizes can use the latest research in deep learning. This fact is the motivation for the development of automated text summarization.

5.10 Difficulties of distribution, sale, introduction into mass use

our product is a computer program and of course have difficulties associated with its distribution, sale, introduction into mass use.

Among the difficulties that do not allow the distribution, sale, introduction into mass use of the product are as follows:

- The product does not satisfy a large percentage of consumer requirements.
- The usefulness of the product is not suitable for some areas.

- It is possible that there are competitive products that have better features than ours.
- It is possible that the quality of the product is distinguished by us.
- It is possible that the price of the product is not suitable.

In the preparation of investment projects related to the development of scientific and technical products (commercialization of research), it should be borne in mind that the project is justified from a financial point of view only in the event that the result of its implementation is of value to the consumer, If the product will be sold in the market.

5.11 Conclusion

Automatic text summarization can be defined as the process of shortening a text computationally, to create a summary that contains the most relevant parts of the original content. Obviously, the critical point is not to reduce the number of words or letters, but to train the machine to understand the grammar and the semantics of the text, in order to re-build its meaning and re-shape it in a reduced form

Techniques and methodologies for English text summarization are still immature because of the inherent quality of English language in terms of each structure and morphology during this thesis ,We tailored extractive approach of automatic summarization conducted on associate degree English text , used BERT models. Our approach supported extractive model victimization sentences classification and k-Means clusterization. Report method starts with preprocessing, division on training and testing dataset or text so fitting and testing models for each the approaches. then the ensuing summaries are evaluated with an individual's summaries .the results obtained are compared to different recent and fashionable works aiming to improve their quality. This work has been terribly helpful for US due the made info and data gained throughout our research’s and studies . we’ve emerged tons of pc science field appreciate :Artificial Intelligence with software system development .It gave US the chance to implement python programing language to explore the big field of ATS with deep learning and created US realize what we have a tendency to need to improve our work for west Germanic in the close future.

This research shows that AI can be used for ATS with different types and can be useful in real life in different ways as we showed as a monitoring or a clock or any other idea that can help to present the automatic text summarization.

6 Conclusion

Automatic text summarization is a data science task that can create short, accurate, and easy to understand summaries from longer documents. People are usually good at this a type of task because they first need to understand the meaning of the original document, and then extract the meaning and master the most important details in the document. The new description and main purpose of this work is to automatically generate abstracts and extractable abstracts in English text, and obtain abstracts that are the same as those written by humans. Thanks for recent a research in this field last year. At first, different approaches of automatic text summarization are described including the main approaches, as well as the basics techniques and methodologies of natural language processing. After that the various methods of deep learning also the different types of neural network used in ATS are presented along with the recurrent neural network (RNN) architecture as the adopted model architecture for the main approaches. Finally a contribution based on adapting, testing and comparing deep extractive approaches when dealing with English text are managed. As a conclusion, Techniques and methodologies for English text summarization are still immature due to the inherent complexity of the English language in terms of both structure and morphology In this thesis, We adapted extractive approach of automatic summarization conducted on an English text , used BERT models. Our approach based on extractive model using sentences classification and k-Means clusterization. Summarization process starts with preprocessing, division on training and testing dataset or text and then fitting and testing models for both the approaches. then the resulting summaries are evaluated with a human summaries .the results obtained are compared to other recent and modern works aiming to improve their quality. This work has been very helpful for us due the rich information and knowledge gained throughout our research's and studies . we've emerged a lot of computer science field such as :Artificial Intelligence with software development . It gave us the opportunity to implement python programming language to explore the large field of ATS with deep learning and made us realize what we need to improve our work for English language in the near future .As future work, we intend to improve and test the built models on other corpora. The following opportunity is the design of Transformer-based approaches for English text summarization.

Bibliography

1 N. Alami, M. Meknassi, S. A. Ouatik, and N. Ennahnahi, "Arabic text summarization based ongraph theory," pp. 1–8, 2015.

2 A. Thomas, "Speech recognition and hidden markov models." [Online]. Available: https://www.slideserve.com/amber/speech-recognition-and-hidden-markov-models (Accessed:25.12.20)

3 K. Kaikhah, "Text summarization using neural networks," 2004.

4 aimadhu. (2015) Supervised and unsupervised learning. [Online]. Available at: https://dataaspirant.wordpress.com/2014/09/19/supervised-and-unsupervised-learning/ (Accessed:01.01.21)

5 V. Alto. (2019-07-06) Neural networks: parameters, hyper parameters and optimization strategies. [Online]. Available at: https://towardsdatascience.com/neural- networks-parameters-hyperparameters-and-optimization-strategies-3f0842fac0a5 (Accessed:15.01.21)

6 How does convolutional neural network work. [Online]. Available at : https://mc.ai/how-does-convolutional-neural-network-work/ (Accessed:02.02.21)

7 (2018) Sentiment analysis with supervisioned and unsupervisioned learning. [Online]. Available at: https://mc.ai/sentiment-analysis-with-supervisioned-and-unsupervisioned-learning/ (Accessed:03.02.21)

8 M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut,"Text summarization techniques: a brief survey," 2017.

9 H. P. Luhn, "The automatic creation of literature abstracts," 1958.

10 M. E. R. Barzilay, Using lexical chains for text summarization, 1999.

11 Y. Jaya Kumar, O. S. Goh, H. Basiron, H. C. Ngo, Suppiah, and P.C, "A review on automatictext summarization approaches," vol. 12, pp. 178–190, 01 2016.

12 S. l. Nabiha.Azizi, Mohammed Redjimi, "Automatic arabic text summarization

approaches,"2017.

13 D. P. O. J. M. Conroy, "Text summarization via hidden markov models," 2001.

14 L. Rabiner and B. Juang, "An introduction to hidden markov models," 1986.

15 R. Nag, K. Wong, and F. Fallside, "Script recognition using hidden markov models," 1986.

16 S. L. Cailan Zhou, "Research of information extraction algorithm based on hidden markovmodel," 2010.

17 T. Nomoto, "Bayesian learning in text summarization," 2005.

18 J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," 1995.

19 B. Larsen, "A trainable summarizer with knowledge acquired from robust nlp techniques," 1999.

20 Z. Ghahramani, AN introduction to hidden markov models and bayesian networks, 2001.

21 M. L. Nguyen, S. Horiguchi, A. Shimazu, and B. T. Ho, "Example-based sentence reductionusing the hidden markov model." 2004.

22 K. Kaikhah, "Automatic text summarization with neural networks," 2004.

23 A. A. Al-Ajlan, H. S. Al-Khalifa, and A. S. Al-Salman, "Towards the development of an auto-matic readability measurements for arabic language," 2008.

24 R. V. Sumit Sahu, Bharti Dongre, "Web spam detection using differentfeatures," 2011.

25 N. Boukhatem, "The arabic natural language processing: Introduction and challenges," 2014.

26 A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions,"2009.

27 H. Saggion and T. Poibeau, "Automatic text summarization: Past, present and future," in Multi- source, multilingual information extraction and summarization. Springer, 2013, pp. 3–21.

28 C.-Y. Lin and F. Och, "Looking for a few good metrics: Rouge and its evaluation," in *Ntcir Workshop*, 2004.

29 R. Katragadda, "Gems: generative modeling for evaluation of summaries," 2010.

30 K. P. S. R. T. Ward and J. H. F. Reeder, "Corpus-based comprehensive and diagnostic mt evalu- ation: Initial arabic, chinese, french, and spanish results," 2002.

31 A. Nenkova, R. Passonneau, and K. McKeown, "The pyramid method: Incorporating human content selection variation in summarization evaluation," 2007.

32 D. Gillick, B. Favre, and D. Hakkani-Tür, "The icsi summarization system at tac 2008." in *Tac*,2008.

33 M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi, "Extractive summarization using continuous vector space models," 2014.

34 D. P. Bertsekas and J. N. Tsitsiklis, Neuro-dynamic programming, 1996.

35 R. S. Sutton, A. G. Barto *et al.*, "Introduction to reinforcement learning." 1998.

36 J. Schmidhuber, "Deep learning in neural networks: An overview," 2015.

37 Y.-Y. Chen, Y.-H. Lin, C.-C. Kung, M.-H. Chung, I. Yen *et al.*, "Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes," 2019.

38 Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," The handbook of brain theory and neural networks, vol. 3361, no. 10, p. 1995, 1995.

39 K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, Recurrent neural networks for language understanding., 2013, 2013.

40 W. Lalouani, "Neural information retrieval."

41 Eugene Charniak and Drew McDermott, Introduction to artificial intelligence, Addison-Wesley Longman Publishing Co.,Inc., Boston, MA, 1985

42 Bill Z. Manaris, Natural Language Processing: A Human-Computer Interaction Perspective, University of Southwestern Louisiana, Academic Press, New York,

vol. 47, pp. 1-66, 1998.

43 Bert explained: state of the large language model for NLP . Available at: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270/ (Accessed:15.04.21)

44 Liu, Y. (2019). Fine-tune BERT for extractive summarization. arXiv preprint arXiv:1903.10318.

45 Miller, D. (2019). Leveraging BERT for extractive text summarization on lectures. arXiv preprint arXiv:1906.04165.

46 Text Summarization with NlTk in python. Available at: https://stackabuse.com/text-summarization-with-nltk-in-python/ (Accessed:20/04/21)

47 Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

48 A visual guide to using BERT for the first time . [Electronic resource] : URL : https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/ (Accessed:01/05/21)