

QUORA QUESTION PAIRS ANALYSIS USING PERT

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
ARTIFICIAL INTELLIGENCE

Submitted by

Shubham Agarwal (2K21/AFI/09)

Under the supervision of

Mrs. Minni Jain



COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

JUNE, 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, SHUBHAM AGARWAL, Roll No. – 2K21/AFI/09 students of M.Tech (Computer Science and Engineering), hereby declare that the project Dissertation titled “Quora Question Pairs Analysis using PERT” which is submitted by me to the Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Shubham Agarwal

Date: 01/06/23

2K21/AFI/09

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled “Quora Question Pairs Analysis using PERT” which is submitted by Shubham Agarwal, Roll No. – 2K21/AFI/09, Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Mrs. Minni Jain

Date: 01/06/2023

Assistant Professor

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to Mrs. Minni Jain for her continuous guidance and mentorship that she provided me during the project. She showed me the path to achieve my targets by explaining all the tasks to be done and explained to me the importance of this project as well as its industrial relevance. She was always ready to help me and clear my doubts regarding any hurdles in this project. Without her constant support and motivation, this project would not have been successful.

Place: Delhi

Shubham Agarwal

Date: 01.06.2023

2K21/AFI/09

Abstract

An innovative design called The Transformer in NLP tries to tackle sequence-to-sequence problems while skillfully managing long-range relationships. It doesn't use convolution or sequence-aligned RNNs; it just uses self-attention to compute representations of its input and output. The encoder-decoder design is the foundation of the transformer concept. After conducting in-depth study, researchers put forward the BERT and GPT transformer-based models, which significantly improved the bulk of NLP tasks including text creation, text summarization, and question answering, among others. But as time went on, a number of these models' drawbacks became apparent. PERT was recommended as a way to get around one of these drawbacks. In this project work, we fine-tune the pre-trained model on the similarity and paraphrasing task and analyze how the model performs in comparison to the other previously introduced methods.

Contents

Candidate’s Declaration	i
Certificate	ii
Acknowledgement	iii
Abstract	iv
Content	vi
List of Tables	vii
List of Figures	viii
1 INTRODUCTION	1
1.1 Overview	1
1.2 Problem Formulation	1
1.3 Objectives of the project	2
1.4 Challenges in Human Language	3
1.5 Natural Language Processing	4
1.6 Natural Language Understanding	6
1.7 Encoder-Decoder Model	7
1.8 Seq2Seq Model	10
1.9 Transformers	11
1.10 Language Models	12
1.11 Pre-trained Models	13
1.12 Transfer Learning	14
2 LITERATURE REVIEW	16
2.1 Technological Flow in NLP	16
2.2 Description of some publicly available models	17
2.2.1 N-Gram and other classical models	17
2.2.2 Transformer models	17

3	METHODOLOGY	26
3.1	Dataset	26
3.2	Model Architecture	28
3.2.1	Overview	28
3.2.2	Implementation Details	29
4	RESULTS	33
5	CONCLUSION AND FUTURE SCOPE	34
5.1	Summary	34
5.2	Future Scope	34

List of Tables

2.1	Summary of Features of Models	25
4.1	Comparison on QQP dataset	33

List of Figures

1.1	NLP flowchart	5
1.2	Encoder-Decoder Architecture	8
1.3	Image Captioning Architecture	8
1.4	Semantic Segmentation Architecture	9
1.5	Machine Translation Architecture	9
1.6	Sequence-to-Sequence model	10
1.7	Transformer Model	11
1.8	General Transfer Learning Diagram	15
2.1	BERT model flowchart	19
2.2	BART model	19
2.3	Training with permutations	20
2.4	Structure of MPNet	22
2.5	TaCL teacher-student architecture	23
3.1	Datasets in GLUE benchmark	26
3.2	QQP dataset	28
3.3	PERT model flowchart	29
3.4	Tokenization	30
3.5	Train-Test split	31
3.6	Model hyper-parameter initialization	31
3.7	Model training	31
3.8	Evaluation function	32
3.9	Testing model	32
3.10	Output of test questions	32

Chapter 1

INTRODUCTION

1.1 Overview

In this report, we have discussed different pre-trained language models and how they perform different natural language understanding applications. The motivation for this contribution comes from the idea of the challenges we face in creating a software that reliably ascertains the intended meaning of text or speech data since human language is rife with ambiguity. We use PERT, a pre-trained language model which was developed to solve the limitation of previously introduced language models like BERT and give advanced results. We test the model for the similarity and paraphrasing tasks of Quora Question Pair and try to understand how well the computer algorithm understands human language.

1.2 Problem Formulation

At what other place but on Quora can a scientist assist a chef deal with a math issue in exchange for food advice? A place to learn and exchange information about anything is Quora. It serves as a forum for queries and connections with experts who offer insightful observations and thorough responses. People are better able to grasp the world and learn from one another as a result.

It's hardly surprising that many questions on Quora are identical in wording given that more than hundred million individuals use the site each month. Multiple inquiries with the same objective might make readers feel as though they must respond to various variations of the same inquiry, while also making searchers spend extra time looking for a suitable solution to their problem. Canonical questions are highly valued on Quora given that they provide active authors and visitors with improved service and more long-term value.

In order to detect replicating queries, Quora presently applies a Random Forest approach. In this research, algorithms are tested to solve this natural language processing issue by determining the possibility that question pairs contain copies using cutting-edge methodologies. This will improve the overall user experience for Quora authors, searchers, and visitors by offering it less complicated to locate compelling responses to queries.

NLP tasks like paraphrasing and similarity detection on MRPC, STS-B and QQP datasets play a very significant role as it makes the basis for many other language and text related applications. Some may consider these applications as insignificant but they help in identifying how well our algorithm is performing and is able to grasp the meaning and context of the language or text.

For example, in the most basic operation of information retrieval like google search, they help in identifying relevant documents or passages that have similar meanings or convey the same information as a given query. By determining the similarity between queries and documents, search engines can provide more accurate and relevant search results. Similarly, it helps in the question answering system by matching the user's queries with pre-existing knowledge to find relevant answers. Paraphrasing can help in generating concise summaries by expressing the same ideas using different words or phrases. This idea is applied in text summarization or text generation applications.

Various models have been proposed starting from the RNN neural network to encoder-decoder models and even the more complex transformer based language models. But every model has its limitations leading to suboptimal results and solving only a certain kind of problem. As a result, we need to create a teaching model to ensure that the encoding generated by the model must be associated with the correct answer and it must be capable of differentiating with invalid input, which are modified versions of the correct answer.

1.3 Objectives of the project

QQP is a similarity detection task in NLP where in every query, we need to predict whether the given questions are similar to each other or not. We witness a variety of techniques, some of which have resulted in a paradigm shift in issue resolution. The survey article examines the proposed methodologies, their outcomes, and their disadvantages.

In this research work, we aim to use and tune the PERT model introduced in 2022 to analyze its performance on the QQP dataset and how the model is different from the previously proposed methods.

1.4 Challenges in Human Language

Human language is a powerful and dynamic tool that facilitates communication, knowledge sharing, and the expression of human experiences. Languages are complex but crucial to express thoughts, ideas, emotions, and convey information. For decades, researchers and developers have been trying to teach computers to comprehend and then communicate back using human language but it is easier said than done.

Computers face several challenges in understanding and processing human language like its inherent complexity, ambiguity in the language and context dependencies in an extract. Ambiguity is the term for instances when a word, phrase, or sentence may have more than one conceivable meaning or interpretation in various languages. Due to the context in which it is used, ambiguity results. There are many different types of ambiguity, including semantic ambiguity (many viewpoints of an expression or sentence), syntactic ambiguity (several readings of a sentence structure), and lexical ambiguity (alternative meanings of the same term). People are susceptible to ambiguity's negative effects. It could result in misconceptions, confusion, and poor communication. Ambiguity necessitates more work and context to clarify, which might result in communication blunders or inefficiencies. Ambiguity can be particularly difficult when communicating between cultures or when people are speaking at various language levels.

Homophones, homonyms, sarcasm, idioms, metaphors, variances in syntax and usage, and alterations in sentence structure are only a few examples of the irregularities in human language.

- Homophones: Words with the same sound but distinct interpretations and frequently different spellings are known as homophones. For instance, while having different meanings, the words "write" and "right" sound the same. Due to the fact that the intended meaning may be highly dependent on context, homophones can cause misunderstanding and ambiguity in communication.
- Homonyms: Homonyms are words that have the same spelling or pronunciation but different meanings. They can be further classified into homophones (same sound, different meaning) and homographs (same spelling, dif-

ferent meaning). For example, "bat" (referring to a flying mammal) and "bat" (referring to a sports equipment) are homonyms.

- **Sarcasm:** Sarcasm is a type of linguistic irony in which the speaker says a point but means to imply something quite different. To express the intended sarcasm, it frequently includes employing tone, context, and non-literal language. Sarcasm involves recognising the contradiction between what is actually meant and what is meant by it, which can be difficult for linguistic models or AI systems to grasp.
- **Idioms:** Idioms are phrases that have metaphoric meanings that are not the same as their literal interpretation. They can be difficult for foreigners to grasp since they are exclusive to certain languages or cultures. For instance, "kick the bucket" refers to passing away rather than actually kicking a bucket.
- **Metaphors:** Metaphors are figurative language devices that compare two things by implying that one is like the other. They use nonliteral language to conjure up an eye-catching picture or idea. For instance, the phrase "time is money" compares the worth and significance of time to money using a metaphor.
- **Deviations to Syntax and Changes in Sentence Structure:** Although language provides for some freedom in the arrangement of sentences and syntax, deliberate phrase structure changes can express certain ideas or writing styles. Word order changes, omissions of particular words, and the use of non-standard language may be examples of these deviations used for emphasis, poetic impact, or aesthetic reasons. Poetry, literature, everyday speech, and some dialects all include them.

In addition to language expertise, understanding and successful use of homophones, homonyms, sarcasm, idioms, metaphors, deviations from syntax, and alterations in sentence structure need contextual comprehension, cultural familiarity, and pragmatic interpretation. These subtleties give human language depth, richness, and complexity, but they can be difficult for non-native speakers or language processing algorithms to understand.

1.5 Natural Language Processing

It is extremely challenging to create software that reliably ascertains the intended meaning of text or speech data since human language is rife with ambiguity. The abnormalities of human language include homophones, homonyms, sarcasm, idioms, metaphors, deviations to syntax and use, and changes in sentence structure, to name

just a few.

The field of computer science known as "natural language processing" (NLP) aims to enable computers to grasp the spoken and written meaning of the language in a manner that is similar to that spoken by humans. NLP aims to develop algorithms and tools that can understand, analyze, and generate natural language text or speech. NLP fuses the statistical methods, machine learning tech. and/or deep learning models with computational grammar or rule-applied model techniques of human language. By applying some technologies, computers gain the power to interpret human language which can be in any form like text or audio data and fully "understand" what is the hidden meaning said or written, including the intentions and mood contained in the input.

NLP has several components, including:

1. Text Preprocessing: Text preprocessing involves cleaning and transforming raw text data into a format that can be used by NLP algorithms. This includes tasks like tokenization, stemming, lemmatization, and stop-word removal.
2. Natural Language Understanding (NLU): NLU is the ability of a computer to understand and interpret human language. This involves tasks like semantic analysis, syntactic analysis, and entity recognition.
3. Natural Language Generation (NLG): NLG is the ability of a computer to generate human-like language. This includes tasks like machine translation, summarization, and dialogue generation.
4. Machine Learning: Machine learning is a key component of NLP, as it provides the algorithms that power many NLP applications. Common machine learning techniques used in NLP include neural networks, decision trees, and support vector machines.

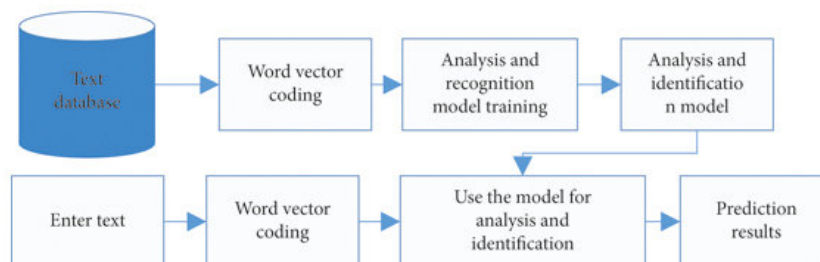


Figure 1.1: NLP flowchart

The role and importance of NLP are significant in many areas, including:

1. **Communication:** NLP plays a critical role in improving communication between humans and machines. NLP-powered chatbots and voice assistants have become increasingly popular in recent years, allowing people to interact with technology in a more natural and intuitive way.
2. **Information Retrieval:** NLP is used in search engines to provide more accurate and relevant search results. NLP techniques like keyword extraction and topic modeling can help identify the most relevant content for a given search query.
3. **Sentiment Analysis:** NLP is used to analyze and understand the sentiment behind text data. This can be used for a variety of applications, including social media monitoring, customer feedback analysis, and market research.
4. **Language Translation:** NLP is used to power machine translation systems, which can automatically translate text from one language to another. This has significant applications in business, international relations, and education.

Overall, NLP is a rapidly evolving field with a wide range of applications and significant potential to improve human-machine interaction and communication.

1.6 Natural Language Understanding

The method by which the machine "comprehends" the information is called natural language understanding (NLU). It's important to recognise the differences between NLU and NLP, NLU is a component of NLP. NLU is an artificial intelligence technology that may be used to understand text and any other form of disorganized linguistic input. It derives context and meaning from natural language inputs. In essence, it describes the method through which a computer decodes human input and purpose and "decides" how to respond appropriately. The word "intent" here denotes a mapping between a user's words and the actions that an AI tool, such a chatbot, should do. The objective of NLU is to provide methods and algorithms that can comprehend, analyze, and express the meaning of natural language text or speech.

Some of the basic operations of NLU include:

1. **Tokenization:** Tokenization is a way of dividing and scaling down a text or document into tokens, which might be words or sentences. This is a crucial stage in NLU since it aids in locating the fundamental meaning units found in natural language.

2. Part-of-Speech (POS) Tagging: The act of labeling each token in a phrase to its correct category of speech, such as a noun, verb, adjective, etc., is known as part-of-speech (POS) tagging. This makes it easier to determine a sentence’s essential meaning and grammatical structure.
3. Named Entity Recognition (NER): NER is the process of locating and categorizing named entities—such as individuals, companies, and places—in a sentence or document. Understanding the surroundings and meaning of a document depends on knowing this.
4. Sentiment Analysis: Sentiment analysis involves determining if a document has a good, negative, or neutral emotional tone. This aids comprehension of a sentence’s or document’s underlying meaning.
5. Semantic Analysis: A sentence or document’s meaning is examined through the process of semantic analysis. Understanding the links between words and phrases as well as the circumstances in which they are used is necessary for this.

Numerous applications, including chatbots, virtual assistants, and search engines, depend on NLU. NLU can increase human-machine interaction and communication by giving robots the ability to understand human language. It can also assist automate a number of formerly manual processes.

1.7 Encoder-Decoder Model

In the field of Artificial Intelligence and deep learning, the encoder-decoder model is a special type of neural network architecture that has shown significant growth in various fields like computer vision, speech recognition, natural language processing and audio processing.

Its architecture consists of two main components: an encoder and a decoder. The encoder processes the given input sequence or signal, such as an image or an audio waveform, into a fixed-length representation, which can also be referred as the “latent representation”. This representation is designed, such that it extracts the important information from the given input sequence in a condensed form. The decoder is responsible to use this representation as its input and produce an output sequence or signal depending on the desired response.

The central component or the building block used in the construction of the encoder-decoder architecture is the neural network. Since there are different kinds

of neural networks like RNN, LSTM, CNN etc., we can insert any type of neural network in the box depending on the application. Encoder-decoder architecture is most suitable for the use case where the input is a sequence of data and the output is another sequence of data. They became popular due to their ability to take a variable-length of input and process it to produce a similar variable-length output.

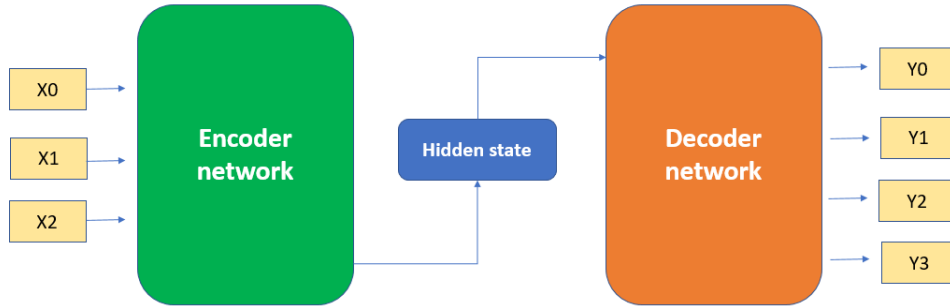


Figure 1.2: Encoder-Decoder Architecture

The Encoder-Decoder model's success may be credited to its capacity to recognise underlying patterns and connections among different elements of input sequences and transfer those relationships to a numerical representation.

Let us look at some examples of how we can use CNN, RNN and LSTM in encoder-decoder architecture to solve different kinds of problems:

- By using CNN as Encoder and RNN/LSTM as Decoder, this architecture is used for tasks like image captioning. The model takes an image as the input and produces a sequence of words as its output where the words are used to describe the image. CNN layers are used to process and extract features from the image, while the RNN/LSTM layers use these features to generate a corresponding text sequence.

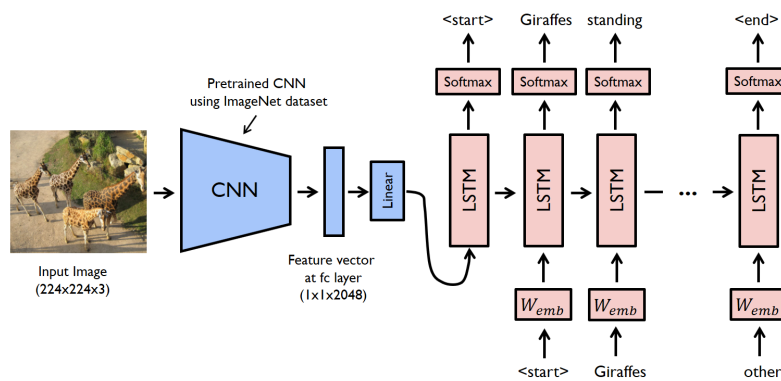


Figure 1.3: Image Captioning Architecture

- By using CNN as Encoder and CNN as Decoder, we can use the encoder decoder architecture for generating semantic segmentation of an image. The model takes an image as the input and the output is a semantic segmented image by classifying each pixel of the given input image to its class label. Decoder plays an important role of semantically distributing the features learnt by the encoder on the pixel plane to get the desired classification.

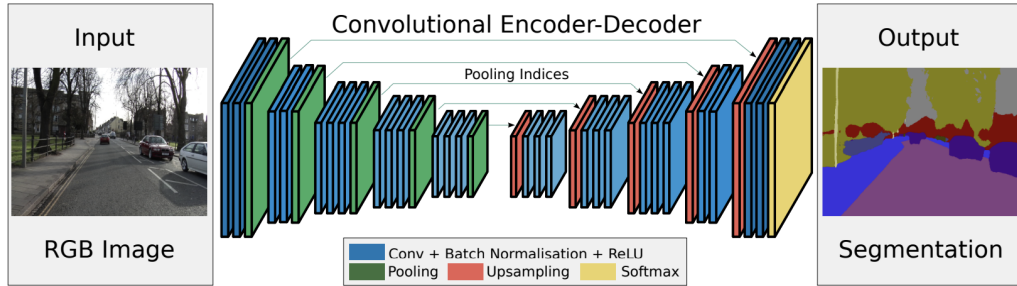


Figure 1.4: Semantic Segmentation Architecture

- By using RNN/LSTM as Encoder and RNN/LSTM as Decoder, applications like machine translation and chatbots can be created. Both input and output are a sequence of words of different lengths. Input text is converted into small and compact integral representation which is then used at the decoder end of the model to assemble a meaningful string of words as the corresponding output.

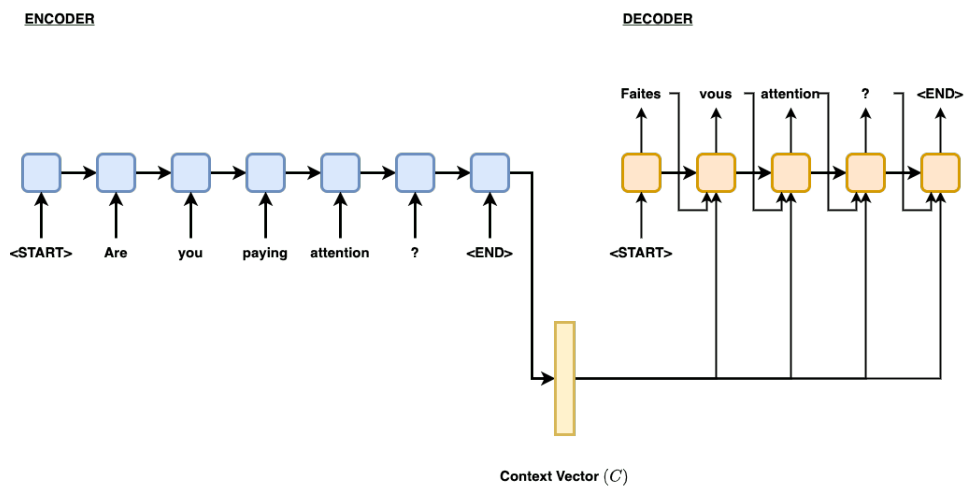


Figure 1.5: Machine Translation Architecture

- Similarly, Encoder-Decoder models are also used in speech recognition applications, where the audio signal of changeable length is processed to compose a transcription of the input signal.

1.8 Seq2Seq Model

Although the Encoder-Decoder model and Seq2Seq model are used interchangeably in Natural Language Processing, strictly speaking, they are not exactly the same things. The Encoder-Decoder model is a general architecture which is more flexible than the Seq2Seq model because it allows for different types of encoder and decoder architectures. The Seq2Seq model is more limited because it specifically uses RNN or LSTM or transformers for both the encoder and decoder.

As in any NLP task, a set of words or letters or characters is provided as input to the encoder, it processes them and builds a well structured coded description. The decoder then maps it to the output sequence using the RNN or LSTM layers.

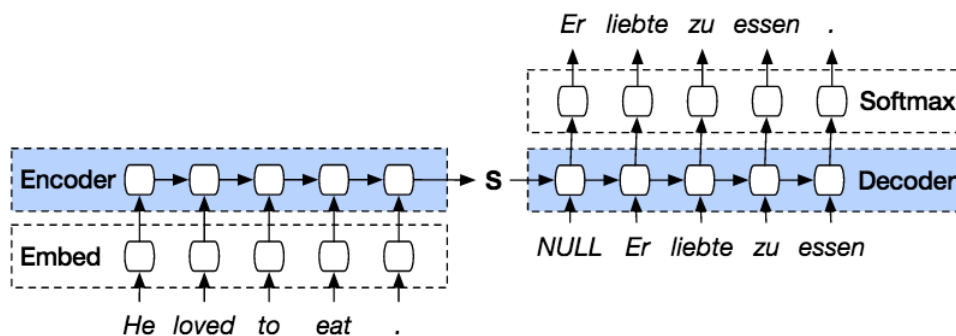


Figure 1.6: Sequence-to-Sequence model

Seq2Seq models offer the capacity to handle input and output sequences of varying lengths, which makes them ideal for applications like machine translation where the input and output languages may have various sentence lengths and structures. Their capacity to develop accurate representations of the input sequences is a further benefit. These representations may subsequently be used for later tasks, such as sentiment analysis or named entity identification.

On a number of NLP tasks, Seq2Seq models have produced state-of-the-art results and have since become a crucial part of many NLP applications. Seq2Seq models now perform better than ever because of recent developments in the field such as attention processes and transformer-based models, making them a viable subject for further study.

1.9 Transformers

If we think of a machine translation application, in which the input sequence is a sentence in English and we want to translate that sentence in Spanish. Simple translation of word by word in the order of appearance may result in an output that a human speaker would consider grammatically incorrect. One of the main questions in sequence transduction is the learning of representations for both the input and output sequences in a robust manner, so that no distortions are introduced. Unlike feed forward networks where one feature of the input is considered independent to the other, to tackle the problem of sequences, RNNs or CNNs or LSTMs were used. In an RNN network, each element is seen with respect to the prior element and thus the output is influenced by previous features of the input. In machine learning, this ability is termed as the ‘memory’ of the network.

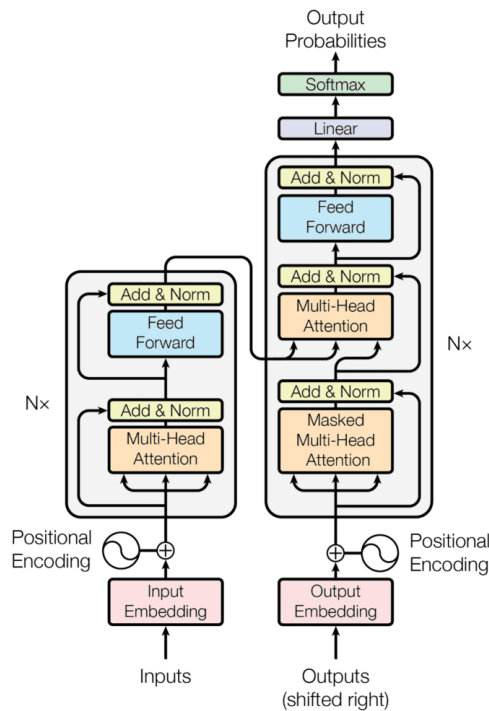


Figure 1.7: Transformer Model

But it is observed that ‘memory’ is not enough in the model. Thus, transformers were introduced. Transformers are a special type of artificial neural networks. As the name suggests, they are responsible for the transformation of the input sequence to the output sequence in our respective application. It uses ‘self - attention’ to help further improve the performance of the model in their respective problem statement. The aim of the attention mechanism is to provide more importance to features that have higher weight values. This allows the model to capture complex dependencies

between the input to create a new context vector.

Transformers are renowned for their capacity to manage lengthy data sequences, which is crucial for NLP activities like language production and machine translation. They are also very parallelizable, which makes them ideal for training with several GPUs on big datasets. In general, Transformers has had a big influence on machine learning, especially in the domain of natural language processing. They already serve as the basis for a large number of NLP operations and will probably continue to be a significant field of study and development in the years to come.

1.10 Language Models

To calculate the chance that a given set of words will appear in a sentence, language modeling employs a number of statistical and probabilistic methods. Language models investigate corpora of text data to offer a basis for their word predictions. A language model estimates the likelihood that a given word combination is "valid." Validity in this context has nothing to do with grammatical correctness. The language model learns that this means that it mimics how people talk (or, to be more accurate, write).

Depending on the goal of the language model, there are many probabilistic methods for modeling language. The quantity of text data they evaluate and the math they employ to do so vary between the various categories.

Several of the typical methods include:

- N-gram: The percentage of times the last word comes after the n-1 gramme, leaving the last word is known as an n-gram. It adheres to the Markov principle.
- Unigram: It assesses every word or phrase separately. Information retrieval problems including language processing are frequently handled by unigram models.
- Bidirectional: They evaluate text in both forward and backward orientations. By utilizing every other word in the text, these models can accurately anticipate any word in a phrase or body of text.

Natural language processing is built on language models. It is crucial to NLP activities like:

- Parsing - Analyzing a given string or text and adhering to the linguistic structure and syntax norm is known as parsing.
- Information Retrieval - It involves searching for information or metadata in a given document.
- Optical character recognition - OCR is the process of using computers to identify text in photographs and turn it into machine-readable text. It is frequently used in tasks like document scanning and record digitisation.

1.11 Pre-trained Models

The idea of pre-trained language models is to build a "black box" that can be instructed to perform any particular task in that language after it has learned the language.

Pre-training is a notion that relates to transfer learning. The goal of transfer learning is to apply previously acquired information to new, purpose driven tasks. A sizable amount of unannotated data is first put into the language model. This data is used by the model to learn how different words are used, how the language is written, and other linguistic activities. The model is then fine-tuned for a particular NLP job. To improve the word embeddings or representations even further and produce the final model, it is fed with a smaller task-specific dataset.

By using pre-trained models and transfer learning, it allows us to speed up the training process rather than training the model from scratch. Plus the fine tuning tasks do not require high computational power, so it can be done on simple personal computers.

Over the years, a wide range of unsupervised pre-training objectives have been investigated for training the neural network on a sizable unlabeled text corpus. The two most effective pre-training goals among the researched areas are Autoregressive (AR) and Autoencoding (AE) modeling methods of language.

By using an autoregressive model, autoregressive language modeling (AR) aims to judge the probability distribution of tokens in text. For instance, the likelihood of a text sequence $x = (x_1, x_2, x_3, \dots, x_T)$ is factored into a forward or a backward product via AR language modeling,

An AR language model is ineffective at simulating complex bidirectional situations since it is only taught to convey unidirectional context. However, downstream tasks of understanding language frequently call for bidirectional context data. This produces a gap between AR language modeling to get effective pre-training. A notable example is XLNet.

Instead of doing explicit density estimation, autoencoder (AE) based pretraining seeks to restore the original data from contaminated input. The model is trained to restore the original tokens from the corrupted version by replacing a piece of the input token sequence with the special symbol [MASK]. The model is permitted to exploit the bidirectional context of the input data since the training does not employ probability density. This improves performance by filling the gap of lack of bidirectional information in AR language modeling. The real data is missing from the fine tuning objective, though, as a result of the pretraining's use of a fictitious symbol like [MASK]. Pretrain-finetune discrepancy is the effect of this. Additionally, unlike AR models, the model is unable to learn the joint probability distribution using the product rule since the predicted tokens in the input are hidden.

ALBERT, RoBERTa, BART, and other novel strategies are being explored and tested to address the merits and downsides of the traditional language pre-training targets, such as the absence of dependent knowledge or the high memory and computational needs.

1.12 Transfer Learning

Transfer learning is a technique for problem-solving in artificial intelligence and machine learning that emphasizes transferring knowledge learned from one problem to a different one that is independent but still presents similar difficulties. For instance, while attempting to recognise trucks, one may employ the abilities learned when studying to identify automobiles. Similar to this, researchers in natural language processing found that after training a language algorithm for predicting the next word, they could simply take the trained model, extract the last layer that estimates the next word, and substitute it with an alternate last layer. This new last layer would then be trained to perform the necessary task, such as sentiment analysis or summarization.

Transfer learning has helped us to a large extent by generalizing the neural network model into many different domains. By training the model on large and diverse datasets, it captures the general features and patterns present in the dataset. This

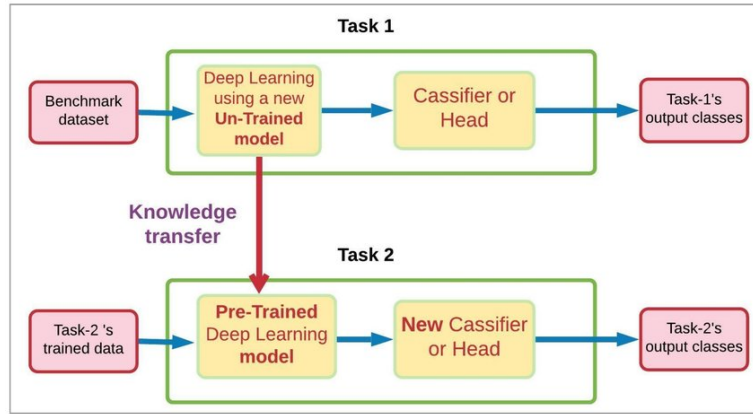


Figure 1.8: General Transfer Learning Diagram

allows the model to generalize knowledge and apply it to many different domains. Similarly, by taking advantage of the pre-trained weights and knowledge that the model learns during its initial training phase, we can reduce the overall time and computational resources required for training a model from scratch for any downstream task. Another advantage of transfer learning is the breakthrough towards extracting appropriate features from complex data. By training on a large dataset, the model acquires the ability to learn different patterns from the dataset and uses it to find hidden features.

Machine learning professionals may use transfer learning to generalize their models and expertise to new domains, manage sparse data, and construct models more quickly. It has developed into a useful method in the industry, allowing for more effective and efficient machine learning solutions.

Chapter 2

LITERATURE REVIEW

2.1 Technological Flow in NLP

Natural Language Processing and Time Series Analysis have been some of the most challenging branches in the applications of AI. Both applications contain data in which a particular order of data is important and it requires memory because the previous data has some influence on the next data. Earlier, recurrent neural networks (RNNs)[1] were introduced where the output at any point is affected by current data and the previous output. This feature enabled the model to retain some previous information like a memory and perform time series predictions.

Following RNNs, Long short-term memory (LSTM)[1] was introduced to encounter the issue of vanishing gradient in RNNs due to a long series of data. LSTMs were very effective in solving problems related to complex sequences. A special class of LSTM architecture called the Sequence to Sequence architecture or the Encoder-Decoder model provided a significant gain in solving natural language related problems like question answering, text summarization or machine translation etc.

After that, Transformers[2] were introduced whose architecture was inspired from the encoder-decoder model. Just like the encoder-decoder model, transformers have two parts. The left half in the transformer architecture is used to simply map the input sequence to another representation of that sequence which is then passed to the right half. The right half of the transformer uses the encoder output and previous time step's output to produce the final result. Transformers also implement the attention mechanism instead of the recurrent or convolution techniques. Attention highlights the important part of the information in the current input.

Release of Transformers completely restructured the way NLP tasks are performed. They are much faster and more efficient in comparison to their predeces-

sors. GPT[3] and BERT[2] were the first big NLP based pre-trained transformer models. They provided highest results for classic tasks like text classification and recognition of named entities and generation tasks like question answering and text summarization. However as the technology advanced, more and more drawbacks started surfacing for these transformers like high computational cost, large memory requirements, skipping masked tokens, lack of linguistic and dependent knowledge in the learned representations etc.

2.2 Description of some publicly available models

2.2.1 N-Gram and other classical models

N-grams[2] are frequently employed in Statistical Natural Language Processing models to create a collection of features from the input text data that will be utilized in the model. By changing the value of "n", various other linear models, including those with unigram, bigram, and trigram features, were tested using this method. Similar to this, Support Vector Machines (SVM)[4] were just another extensively employed method for solving prediction issues. SVMs enable the use of various kernel functions that aid in non-linear data separation. Later, deep learning techniques for comparable problems like natural language inference (NLI) were established. Because figuring out whether one question is comparable to its pair is equivalent to figuring out whether one question is implied by the other, the problem of detecting questions in the same order is comparable to an NLI task. The earliest continuous bag of words (CBOW) architecture was used in deep neural networks before complexity levels like LSTM and BiLSTM[4] were added. Every question in the sample was given a single word vector by each of these models, and the resultant label prediction was calculated using both representations.

2.2.2 Transformer models

Pre-trained language models have recently demonstrated improved performance in various NLP tasks. GPT[3] and BERT[2] use transformers as their core for comprehensive and generative pre-training of the model and afterwards achieve significant performance gain by fine-tuning themselves on the required downstream tasks. This is because transformer architecture which include multi-head self-attention modules and feed-forward components have already shown greater reliability than LSTMs in many NLP tasks. Since then, several improvements to all these pre-trained language models have already been implemented to further increase their performance.

BERT

A ground-breaking methodology in Natural Language Processing (NLP) called BERT (Bidirectional Encoder Representations from Transformers)[2] has revolutionized several language processing jobs. It marks a huge advancement in comprehending word and phrase context, allowing for more accurate and sophisticated language interpretation.

The Transformer design, which uses self-attention techniques to record contextual dependencies in both directions, is the foundation of BERT. This two-way method enables BERT to comprehend the meaning of a word by taking into account the words around it, leading to more accurate representations. Pre-training and fine-tuning are the two steps in the training process for BERT. It gains knowledge from a vast corpus during the pre-training phase by anticipating hidden words in a phrase and comprehending the connections between related sentences. BERT is able to acquire rich contextual representations as a result.

BERT's capacity to manage the difficulties of polysemy (expressions with numerous meanings) and disambiguation of word senses is one of its main advantages. BERT can properly determine the true significance of ambiguous words by taking the complete sentence context into account. BERT is also very good at tasks like sentiment analysis, named entity identification, question answering, as well as categorization of texts because it can capture complex syntactic and semantic links.

Numerous NLP applications have been changed by the rich representations that the context-dependent word embeddings generated by BERT give. Researchers and practitioners can benefit from the model's capacity to transfer information and increase performance even in the absence of sufficient task-specific data by combining pre-trained BERT models and optimizing them on particular tasks.

A key component of several cutting-edge models and applications, BERT has significantly advanced the NLP industry. It has revolutionized natural language comprehension thanks to its capacity to collect in-depth contextual information, manage nuanced linguistic constructions, and transfer learning across tasks. BERT and its variations continue to push the limits of language comprehension and spur new developments in several real-world applications as NLP develops.

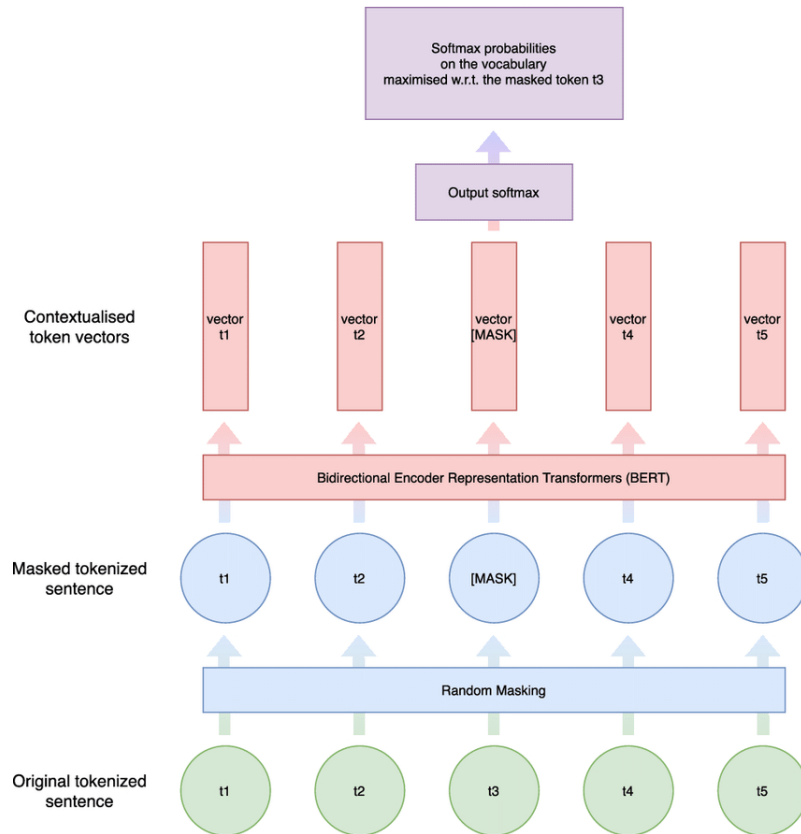


Figure 2.1: BERT model flowchart

BART

There has been remarkable success in quite a few NLP and NLU based projects and workload with the help of the self-supervised methods like BERT (bidirectional encoder representation from transformer) and GPT (generative pre-trained transformer). But being an early approach to tackle the problems of NLP, both models have some drawbacks which prevent them from being the best. Thus the model of BART[5] was introduced which is a transformer-based approach to generalize BERT, GPT and some other pre-training schemes by combining the bidirectional encoder and auto-regressive decoder.

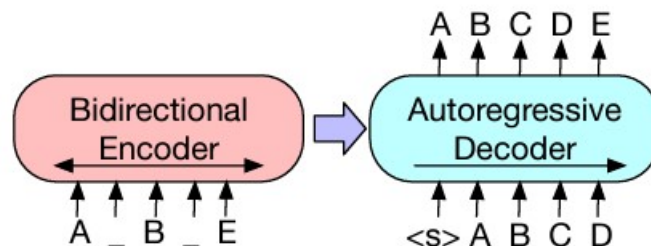


Figure 2.2: BART model

For training the sequence-to-sequence model, it first corrupts the user provided text by replacing some text in the document by mask symbols. The corrupted document is passed to the bidirectional encoder to learn a representation and then the left-to-right autoregressive decoder is utilised to generate the original document. It also provides freedom of noise i.e. to apply every conceivable method of record corruption technique like token concealing, token elimination, word infilling, sentence arrangement etc. BART surpasses previous PLMs in text summarization and text production roles while performing akin to RoBERTa in other disproportionate activities.

XLNet

Both autoregressive (AR) and autoencoding (AE) approaches have some disadvantages in their pre-training objectives as the autoregressive model only encodes a uni-directional context and the autoencoding model fails to calculate the joint probability like the AR models. Thus, XLNet[6] was proposed.

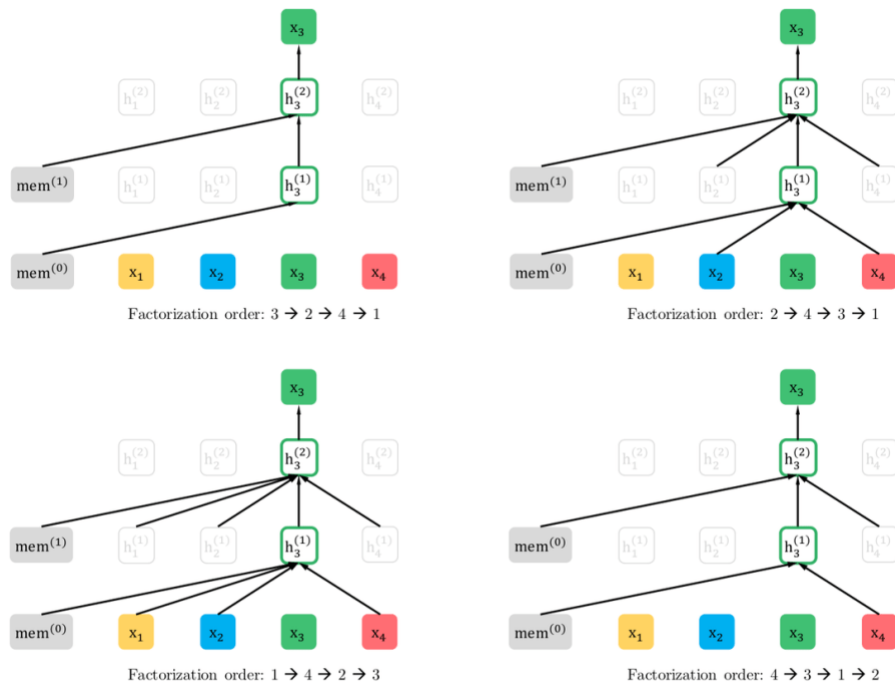


Figure 2.3: Training with permutations

XLNet is an autoregressive approach that adopts the best of both AR and AE models. Unlike using a fixed order like unidirectional or bidirectional, it considers all the possible permutations of the factorization order. This permits the model to understand the significance of every token from each position. Also since it is an AR model, it can use the product principle to generate the combined likelihood of

the tokens it needs to predict.

XLNet uses the permuted language modelling (PerLM) objective for pre-training thus removing its dependency on the masked token and solving the problem of fine tune discrepancy. By integrating segment reoccurring way and comparative encoding strategy, the performance of the model improved significantly. XLNet outperforms the previous pre-training objectives by a large margin in many NLP tasks.

ALBERT

As the technology advanced, it became possible to train large networks and it was observed that expanding the architecture of the model when pre-training performed better on downstream tasks. However, as the models became larger, the quantity of parameters associated with the model also grew exponentially and memory limitation became a problem. As a result the final model was degraded in performance and took longer to train.

Therefore, A Lite BERT (ALBERT)[7] architecture was introduced which had remarkably fewer parameters than BERT. This was because it adopted two variable compression techniques: factorized embedding parameterization and cross-layer parameter sharing. As the name suggests, cross-layer parameter sharing allows layers at a deeper level of the network to share parameters with the layers at shallow level. In BERT, the embedding size of tokens was tied to the hidden layer size due to which as the model size increases, embedding size also increases and parameters of the model increases. Thus, factorized embedding parameterization unties and fixes the embedding size. Together these two techniques reduce the parameters in the model by 18 times.

RoBERTa

Hyperparameter tuning is one of the most critical and challenging jobs in learning algorithms. The goal of hyperparameter tuning is to find a set of values of hyperparameters for a particular learning algorithm that makes the model efficient and enhances the performance. This is done by a hit and try method in which the model is trained on different combinations of hyperparameter values and observe their performance to choose the best model.

It was observed that BERT was severely undertrained and by increasing the training time and tuning the hyperparameters, it can achieve better performance.

Thus, an improved version of BERT was introduced called RoBERTa[8] (Robustly optimized BERT approach). Along with hyperparameter tuning and more training, RoBERTa makes two more changes in the architecture: removing the NSP loss technique of BERT, as it was observed that it improves the success of the model on subsequent roles and unlike BERT which implements static masking (fixed pattern of masking in each input), RoBERTa implements dynamic masking. With adaptive masking, a fresh mask pattern is developed each time a data stream comes. RoBERTa showcases a significant improvement over the original BERT in almost all the NLP tasks.

MPNet

The Masked Language Modelling (MLM) method for pre-training adopted by BERT has a disadvantage as it does not consider the dependency of masked tokens. It assumes them to be independent. This is known as the Output dependency problem. To overcome this, the Permuted Language Modelling (PLM) method was introduced in XLNet. But by taking different permutations of the user provided text, the positional knowledge of tokens is lost producing the positional discrepancy in the initial training of the variables. This is known as the Input consistency problem.

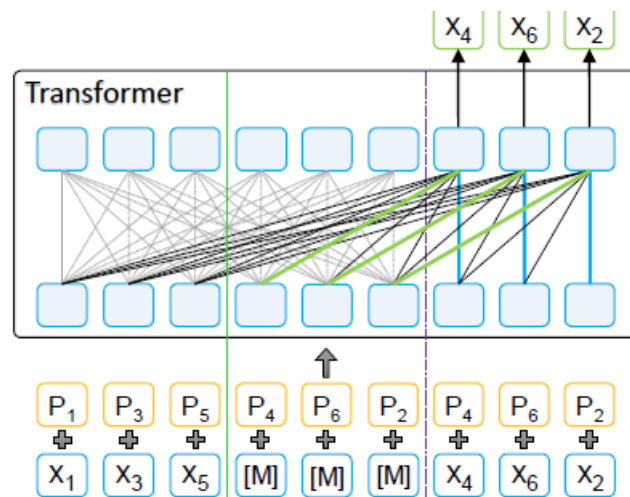


Figure 2.4: Structure of MPNet

MPNet[9] produces a unified view of MLM and PLM while addressing the issues faced in both methods. In MLM, the tokens are divided into masked and non-masked tokens and in PLM, after permutation the tokens are divided into predicted and non-predicted tokens. Therefore by keeping the non-masked and non-predicted tokens the same, in the overall perspective of the input, non-masked symbols are placed on the left, then the concealed symbols and finally the anticipated symbols on

the extreme right. With this method, the model takes the dependency of predicted tokens to solve masked token's discrepancy and by taking positional information of all tokens, it avoids the positional discrepancy.

TaCL

Although BERT revolutionized the way computers understand natural language. But as the technology advanced, more and more drawbacks of BERT surfaced. One of the drawbacks suffered by Masked Language Modelling based pre-trained models is the anisotropic distribution problem. It states that the token representations learned by the model lie only in a small dimensional space. This causes the representations to be less discriminative and unable to differentiate between similar tokens.

TaCL (Token-aware Contrastive Learning)[10] is a new pre-training approach that motivates the model to learn more isotropic and discriminative representations of the tokens. This approach uses two models initialized with pre-trained BERT. One model acts as a student and the other model acts as a teacher. The training objective of this network is to contrast the masked token's representations produced by the student model with the actual representations produced by the teacher model. The result of this training approach brings some improvement in the performance of the model in comparison to BERT across many NLP tasks.

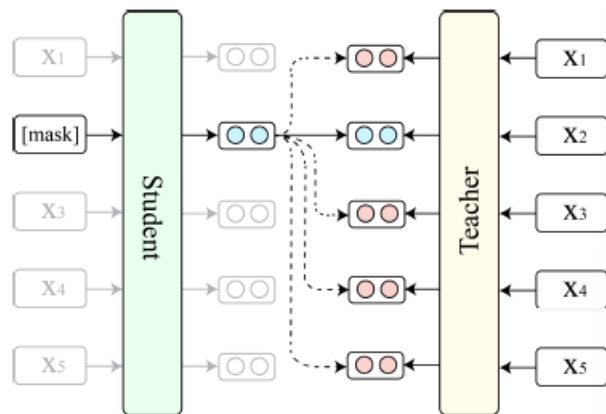


Figure 2.5: TaCL teacher-student architecture

SpanBERT

SpanBERT[11] is a new masked language modelling based pre-trained model which introduces new masking techniques and training objectives. As the term indicates, in this strategy a contiguous span of fixed length of text is masked randomly rather than masking single tokens or masking whole words. The training objective introduced is called span-boundary objective (SBO). In continuation to the masking technique,

SBO aims to teach the learning algorithm to correctly anticipate the entire masked span in one go using the context it learned from other tokens in the input text. Also SpanBERT omits the Next Sentence Prediction (NSP) algorithm of BERT as it is observed on multiple occasions that removing it increases the success of the model on subsequent roles. SpanBERT outperformed BERT on most of the NLP tasks but it performed exceptionally well on answer creation problems SQuAD 1.1 and 2.0.

ConvBERT

In English and many other languages, sometimes the same word expresses multiple meanings and sometimes different words show the same meanings. This makes the language ambiguous. To attend to this ambiguity, BERT had the self-attention block. When the BERT model is processing the input text, for each word it processes, the self-attention block warrants the system to view more expressions in the input text and grasp the correct meaning of each word. This allows it to learn a better version of representation of each word in the sentence. However, since BERT is highly dependent on its self-attention blocks as it has a block for each layer in the architecture, BERT requires large memory and more computation time and power.

It was observed that a large portion of the attention blocks can be substituted by internal dependencies in the text thus reducing the weight of the model. Since convolution operation best summarizes the local features, ConvBERT[12] was introduced. In ConvBERT, convolution operation is integrated with the self-attention to make mixed attention. Experiments show that the model achieved comparable performance while reducing the computation cost.

Model	AE/AR	Training	Masking	Novel Idea
ALBERT	AE	MLM + SOP	Token	Factorized embedding parameterization Cross-layer parameter sharing
XLNET	AR	Permutation LM (PLM)		Bidirectional style in autoregressive
RoBERTa	AE	MLM	Dynamic	Hyperparameter tuning for BERT
BART	AE	Reconstruct corrupt text	Any	Reconstruct text with noise using whole transformer
SpanBERT	AE	Span MLM	N-gram	Continuous span masking
ConvBERT	AE	Replaced token detection		Mix Convolution in attention block
TaCL	AE	NSP + MLM + unsupervised contrastive learning	Token	Learn isotropic and discriminative distribution
MPNet	AE	MLM + PLM	Token	Unifying MLM and PLM

Table 2.1: Summary of Features of Models

Chapter 3

METHODOLOGY

3.1 Dataset

An assortment of tools for developing, testing, and analyzing natural language understanding systems may be found in the General Language Understanding Evaluation (GLUE)[13] benchmark. By using pre-existing databases from a wide variety of database sizes, text genres, and difficulty levels, it is a standard of nine expression or phrase-pair comprehension tasks. It is employed for comparing and contrasting various NLP models across a range of natural language features.

Dataset	Description	Data example	Metric
CoLA	Is the sentence grammatical or ungrammatical?	"This building is than that one." = Ungrammatical	Matthews
SST-2	Is the movie review positive, negative, or neutral?	"The movie is funny , smart , visually inventive , and most of all , alive ." = .93056 (Very Positive)	Accuracy
MRPC	Is the sentence B a paraphrase of sentence A?	A) "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ." B) "The island reported another 35 probable cases yesterday , taking its total to 418 ." = A Paraphrase	Accuracy / F1
STS-B	How similar are sentences A and B?	A) "Elephants are walking down a trail." B) "A herd of elephants are walking along a trail." = 4.6 (Very Similar)	Pearson / Spearman
QQP	Are the two questions similar?	A) "How can I increase the speed of my internet connection while using a VPN?" B) "How can Internet speed be increased by hacking through DNS?" = Not Similar	Accuracy / F1
MNLI-mm	Does sentence A entail or contradict sentence B?	A) "Tourist Information offices can be very helpful." B) "Tourist Information offices are never of any help." = Contradiction	Accuracy
QNLI	Does sentence B contain the answer to the question in sentence A?	A) "What is essential for the mating of the elements that create radio waves?" B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field." = Answerable	Accuracy
RTE	Does sentence A entail sentence B?	A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members." B) "Yunus supported more than 50,000 Struggling Members." = Entailed	Accuracy
WNLI	Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun?	A) "Lily spoke to Donna, breaking her concentration." B) "Lily spoke to Donna, breaking Lily's concentration." = Incorrect Referent	Accuracy

Figure 3.1: Datasets in GLUE benchmark

Any system competent of processing sentences and phrase pairs and delivering appropriate predictions is allowed to compete since the GLUE benchmark's structure is model-neutral. The benchmark problems are chosen to favor models that use sharing of parameters or other forms of transfer learning approaches to communicate knowledge across tasks. The main objective of GLUE is to stimulate research into

the creation of comprehensive and reliable natural language understanding systems.

The common goal of all researchers is to develop a generalized model that, rather than being task-specific, can be used in a variety of situations. Thus, the purpose of developing this criteria was to enable researchers to compare the quality of different models. The ultimate performance score of the model is determined by scoring it on each of the nine tasks after training it on each one.

In this study, we concentrate on the semantic similarity and paraphrasing task and use the Quora Question Pair dataset for our research work. The Quora Question Pairs dataset, which is made available as part of a Kaggle competition, comprises the train set of 404,290 samples of question pairs and the testing set of 2,345,795 samples of question pairs. We decided it would be best to build our custom test set through the training set given because the offered test set is missing labels for every question combination. This would allow us to undertake a deeper error analysis on our prediction models and evaluate performance in measures other than accuracy. As a result, we restricted the scope of our data analysis to the 404,290 question pair queries in the training set.

The following fields are present in each sample point:

- id: distinct ID of every sample point
- qid1, qid2: unique ID of both the questions
- question1, question2: written textual data of both the questions
- is_duplicate: boolean value to determine whether or not the questions are identical copies of one another (0 signify not similar, 1 signify similar).

Due to the fact that 255,027 (63.08%) of the 404,290 question pairs receive a negative label and 149,263 (36.92%) a positive label, our dataset is unequal. Despite the fact that each pair of questions is distinct, not all questions inside the question pairs are. Several of the inquiries were repeated. Our dataset's character set was not entirely ASCII. Among 8000+ question pairs, we found that non-ASCII symbols were used in 6000+ of the questions. Moreover, two pairings included having one of its questions be just an empty string.

questions (sequence)	is_duplicate (bool)
{ "id": [1, 2], "text": ["What is the step by step guide to invest in share market in india?", "What is the step by step guide to invest in share market?"] }	false
{ "id": [3, 4], "text": ["What is the story of Kohinoor (Koh-i-Noor) Diamond?", "What would happen if the Indian government stole the Kohinoor (Koh-i-Noor) diamond back?"] }	false
{ "id": [5, 6], "text": ["How can I increase the speed of my internet connection while using a VPN?", "How can Internet speed be increased by hacking through DNS?"] }	false
{ "id": [7, 8], "text": ["Why am I mentally very lonely? How can I solve it?", "Find the remainder when 23^{24} is divided by 24,23?"] }	false
{ "id": [9, 10], "text": ["Which one dissolve in water quickly sugar, salt, methane and carbon di oxide?", "Which fish would survive in salt water?"] }	false
{ "id": [11, 12], "text": ["Astrology: I am a Capricorn Sun Cap moon and cap rising...what does that say about me?", "I'm a triple Capricorn (Sun, Moon and ascendant in Capricorn) Wha..."] }	true
{ "id": [13, 14], "text": ["Should I buy tiago?", "What keeps children active and far from phone and video games?"] }	false

Figure 3.2: QQP dataset

3.2 Model Architecture

3.2.1 Overview

Auto-encoding based Pre-trained Language Models (PLM) are majorly pre-trained using the masked language modelling task. In the MLM pre-training task, a few of the symbols in the user provided text is substituted with the masking symbol (i.e. [MASK]) and the parameters of the architecture are trained to find the correct symbol from the vocabulary that fits the masked tokens. But it has a drawback. The encoded semantic representation of the input text only captures the contextual features of the tokens around the masked token and the contextual features of the artificial symbol are absent from the real representation. This creates a discrepancy leading to a sub-optimal depiction of the user provided text and the model lacks in performing the task optimally.

Thus a question of pre-training an auto-encoding PLM with tasks other than MLM arises. To answer this question, a new pre-training task was proposed called the Permuted Language Model (PerLM). The objective of PerLM is to permute a portion of the input text randomly and train the models to recover the original positions of the tokens of the input text.

PERT[14] is an auto-encoding model based on PerLM. It employs whole word masking[2] and N-gram masking[2] where either all the tokens of a word or N tokens are permuted and the model learns to find their original locations. This approach permits the model to develop a combination of short and long text inference and it should have an improved performance in reading comprehension tasks and named entity recognition tasks.

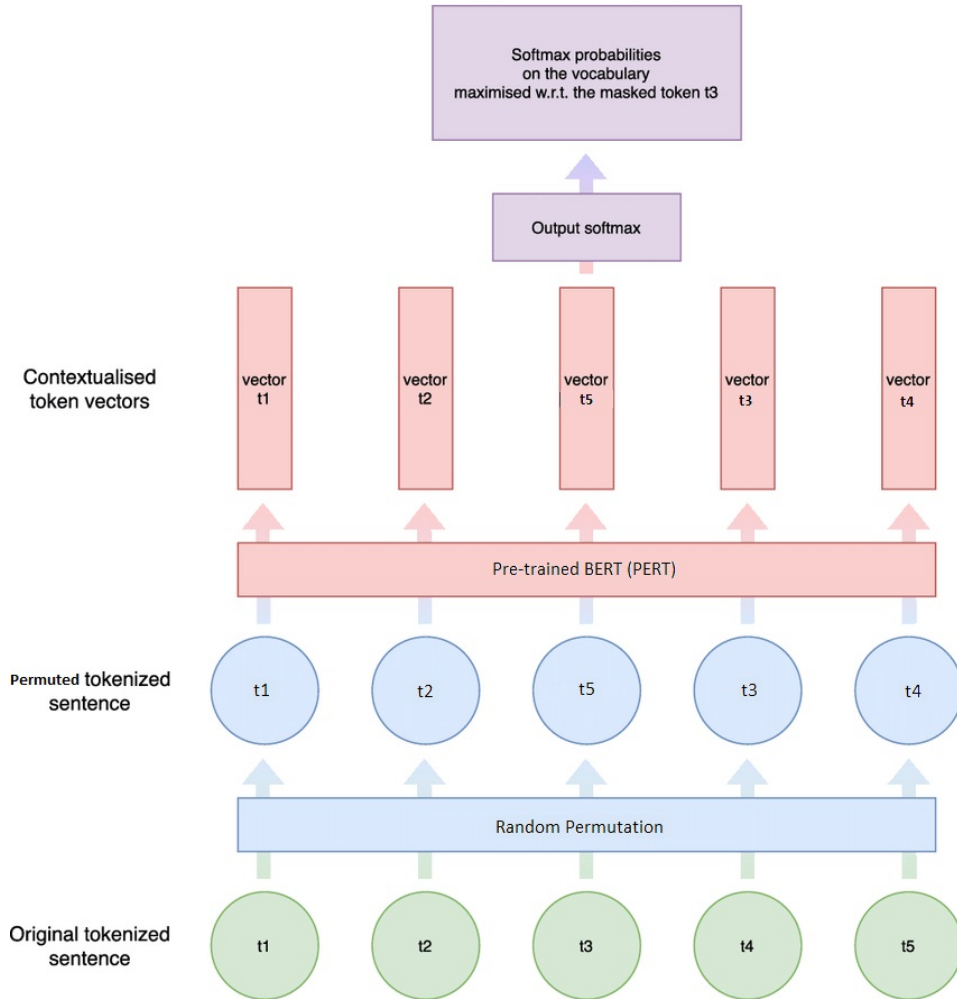


Figure 3.3: PERT model flowchart

3.2.2 Implementation Details

The models use the same BERT-like training methodology. We employ the pre-trained models from the transformer library of huggingface.

- Data: The model employs English Wikipedia with BooksCorpus[15] here as pre-training data, which are often utilized in earlier work.
- Tokenization: It employs the same WordPiece tokenizer[16] for tokenization as BERT.
- Vocabulary: With a vocabulary capacity of 30522, it directly employs the English BERT-base-uncased vocabulary.
- Hyper-parameters: During the entire pre-training procedure, the tokenizer employs a maximum sequence length of 512.
- Optimization: With a starting learning speed of $1e-4$, optimization employs a batch frequency of 416 (base-level). After the initial 10k steps, a linear

warmup plan is executed. There are 2M total training steps. It makes use of ADAM with a weight decay optimizer with a decay rate of 0.1, beta values of (0.9, 0.999), and an epsilon value of 1e-6.

Given that their primary brain architectures are similar, PERT and BERT use the same paradigm to achieve fine-tuning on diverse downstream tasks. In other words, PERT can easily integrate into any fine-tuning code already employing BERT or anything like.

The workflow of NLP applications is a little different from others because here we are working on text data. The steps are as follows:

Data Exploration and analysis: In this step, we observe the dataset and perform all the basic operations like analyzing incomplete examples and completing these examples by either filling them manually or removing these examples. All these operations are application dependent and since we need to use the entire sentence to interpret its meaning, we do not perform these operations in our application.

Text preprocessing: It is the most important step in any NLP application. During preprocessing, we perform noise cleaning (removing special symbols and characters), stopwords removal (removing less significant words from sentence), spell checking and stemming and lemmatization.

Text Representation: As the computer does not understand language characters, we need to convert these characters to represent them using numerical values for the computer to understand. This step is also known as Tokenization. In our project, we create a function to separately tokenize both questions of each example.

```
def tokenize_function(example):
    questions = example['questions']
    t1 = []
    t2 = []
    for t in questions:
        t1.append(t['text'][0])
        t2.append(t['text'][1])
    return tokenizer(t1, t2, truncation=True)

tokenized_datasets = raw_datasets['train'].map(tokenize_function, batched=True)
tokenized_datasets
```

0% | 0/405 [00:00<?, ?ba/s]

Figure 3.4: Tokenization

Before training the model, we split our dataset into ‘train’ and ‘test’ sets using the basic 80:20 ratio.

```

new_features = tokenized_datasets.features.copy()
new_features["is_duplicate"] = ClassLabel(num_classes=2, names=['not_duplicate', 'duplicate'], names_file=None, id=None)
tokenized_datasets = tokenized_datasets.cast(new_features)
tokenized_datasets = tokenized_datasets.remove_columns('questions').rename_column('is_duplicate', 'labels')
tokenized_datasets = tokenized_datasets.train_test_split(test_size=0.2)
tokenized_datasets

Casting the dataset: 0%|          | 0/405 [00:00<?, ?ba/s]

DatasetDict({
  train: Dataset({
    features: ['labels', 'input_ids', 'token_type_ids', 'attention_mask'],
    num_rows: 323432
  })
  test: Dataset({
    features: ['labels', 'input_ids', 'token_type_ids', 'attention_mask'],
    num_rows: 80858
  })
})

```

Figure 3.5: Train-Test split

Model training: We train a single PERT model i.e. the PERT-base model having 12 layers, 12 heads and 768 dimensions of the layer, which are the same as the BERT-base model. In the hyper-parameter settings for the training process, we keep the MaxLength of each question to 512, Batch size as 32 for both ‘train’ and ‘test’ set and fine-tune our model for 3 epochs.

```

training_args = TrainingArguments("./quora-saved-model", evaluation_strategy="epoch", save_strategy="no",
                                  report_to="none", num_train_epochs=3,
                                  per_device_train_batch_size=32,
                                  per_device_eval_batch_size=32)

trainer = Trainer(
    model,
    training_args,
    train_dataset=tokenized_datasets['train'],
    eval_dataset=tokenized_datasets['test'],
    data_collator=data_collator,
    tokenizer=tokenizer,
    compute_metrics=compute_metrics,
)

```

Figure 3.6: Model hyper-parameter initialization

```

trainer.train()

/usr/local/lib/python3.7/dist-packages/transformers/optimization.py:310: FutureWarning: This implementation of AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or set `no_deprecation_warning=True` to disable this warning
FutureWarning,
***** Running training *****
  Num examples = 323432
  Num Epochs = 3
  Instantaneous batch size per device = 32
  Total train batch size (w. parallel, distributed & accumulation) = 32
  Gradient Accumulation steps = 1
  Total optimization steps = 30324
  Number of trainable parameters = 109483778

[30324/30324 3:39:18, Epoch 3/3]

```

Epoch	Training Loss	Validation Loss	Accuracy	F1
1	0.297100	0.287816	0.880853	0.840544
2	0.232100	0.266344	0.898934	0.866444
3	0.158700	0.265175	0.904858	0.873222

Figure 3.7: Model training

Evaluation: Finally we evaluate our model on the test set using the GLUE benchmark metric of the huggingface library.

Chapter 4

RESULTS

F1 score is a machine learning evaluation metric which is popularly used to measure the performance of models in classification applications. It is used when there are only two classes to classify the data example into. It is calculated by combining precision and recall. Table 4.1, shows the comparison between our PERT model with other state-of-the-art models. The table provides the accuracy and the F1 Score for our model as well as the other models. As we can see, with an accuracy rating 90.4% and an F1 score of 87.3%, our PERT model performs better than other cutting-edge models. This represents a significant increase above the BERT model's previous best outcomes, which had an accuracy and F1 score around 89.6% and 85.9%, respectively.

	Accuracy	F1 score
PERT	90.4	87.3
BERT	89.6	85.9
GPT	88.5	-
XLNet	90.5	-
SpanBERT	89.5	-
ConvBERT	90	-

Table 4.1: Comparison on QQP dataset

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Summary

There have been many attempts of researchers trying to revamp the downsides faced in BERT. In this research, we investigate the novel pre-trained language model PERT, whose pre-training job is the permutation language modeling (PerLM). We conducted the experiment here on an English NLU task using the Quora Question Pairs dataset in order to assess PERT's performance. In the experiment, we must determine whether the two questions are comparable or not. The experiment's findings demonstrate that PERT produces better results than the baseline BERT and GPT models. We expect that the PERT experiment will encourage the community to create other pre-training challenges for representation learning that are not similar to MLM.

5.2 Future Scope

Further work includes combining multiple modifications in different areas and seeing how well they complement each other and further enhance their performance in understanding different languages and performing various tasks.

Additionally, due to time constraints, many various modifications, testing, and experiments have been postponed (i.e. the experiments with real data are usually very time consuming, requiring even days to finish a single run). The proposed PERT model only considers simple token level permutation of the input text, but in future we can investigate other permutation approaches such as permuting tokens within a single complete word or permutation of sentences.

Bibliography

- [1] K. M. Tarwani and S. Edem, “Survey on recurrent neural network in natural language processing,” *Int. J. Eng. Trends Technol*, vol. 48, no. 6, pp. 301–304, 2017.
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [3] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, “Improving language understanding by generative pre-training,” 2018.
- [4] L. Sharma, L. Graesser, N. Nangia, and U. Evci, “Natural language understanding with the quora question pairs dataset,” *arXiv preprint arXiv:1907.01041*, 2019.
- [5] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” *arXiv preprint arXiv:1910.13461*, 2019.
- [6] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” *Advances in neural information processing systems*, vol. 32, 2019.
- [7] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations,” *arXiv preprint arXiv:1909.11942*, 2019.
- [8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pre-training approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [9] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, “Mpnet: Masked and permuted pre-training for language understanding,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 857–16 867, 2020.

- [10] Y. Su, F. Liu, Z. Meng, T. Lan, L. Shu, E. Shareghi, and N. Collier, “Tacl: Improving bert pre-training with token-aware contrastive learning,” *arXiv preprint arXiv:2111.04198*, 2021.
- [11] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, “Spanbert: Improving pre-training by representing and predicting spans,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.
- [12] Z.-H. Jiang, W. Yu, D. Zhou, Y. Chen, J. Feng, and S. Yan, “Convbert: Improving bert with span-based dynamic convolution,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 837–12 848, 2020.
- [13] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “Glue: A multi-task benchmark and analysis platform for natural language understanding,” *arXiv preprint arXiv:1804.07461*, 2018.
- [14] Y. Cui, Z. Yang, and T. Liu, “Pert: pre-training bert with permuted language model,” *arXiv preprint arXiv:2203.06906*, 2022.
- [15] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.
- [16] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *arXiv preprint arXiv:1609.08144*, 2016.
- [17] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [18] A. Sufian, A. Ghosh, A. S. Sadiq, and F. Smarandache, “A survey on deep transfer learning to edge computing for mitigating the covid-19 pandemic,” *Journal of Systems Architecture*, vol. 108, p. 101830, 2020.
- [19] B. Gao, “Research and implementation of intelligent evaluation system of teaching quality in universities based on artificial intelligence neural network model,” *Mathematical Problems in Engineering*, vol. 2022, pp. 1–10, 2022.
- [20] <https://www.guru99.com/seq2seq-model.html>.
- [21] <https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/>.

- [22] <https://machinelearningmastery.com/the-transformer-model/>.
- [23] <https://pyimagesearch.com/2022/08/15/neural-machine-translation/>.
- [24] <https://huggingface.co/datasets/quora>.
- [25] <https://mccormickml.com/2019/11/05/GLUE/>.

List of Publications

- [1] S. Agarwal and M. Jain, "A comparative study of different BERT," communicated and accepted at International Conference on Artificial Intelligence, Blockchain, Computing and Security (ICABCS-23).
- [2] S. Agarwal, "Quora Question Pairs using PERT," communicated and accepted at 5th IEEE International Conference on Advances in Computing, Communication Control and Networking (ICAC3N-23).

PAPER NAME

ShubhamAgarwal_Thesis.pdf

WORD COUNT

10244 Words

CHARACTER COUNT

56848 Characters

PAGE COUNT

47 Pages

FILE SIZE

2.4MB

SUBMISSION DATE

May 27, 2023 11:58 AM GMT+5:30

REPORT DATE

May 27, 2023 11:59 AM GMT+5:30

● 11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 7% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 9% Submitted Works database

● Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less than 10 words)



Shubham Agarwal <shubhamagarwal904@gmail.com>

Acceptance Notification 5th IEEE ICAC3N-23 & Registration: Paper ID 725

1 message

Microsoft CMT <email@msr-cmt.org>
Reply-To: Vishnu Sharma <vishnu.sharma@galgotiacollege.edu>
To: Shubham Agarwal <shubhamagarwal904@gmail.com>

24 May 2023 at 21:28

Dear Shubham Agarwal,
Delhi Technological University

Greetings from ICAC3N-23 ...!!!

Congratulations....!!!!

On behalf of the 5th ICAC3N-23 Program Committee, we are delighted to inform you that the submission of "Paper ID-725 " titled " Quora Question Pairs using PERT " has been accepted for presentation and further publication with IEEE at the ICAC3N- 23 subject to incorporate the reviewers and editors comments in your final paper. All accepted papers will be submitted to IEEE for inclusion into conference proceedings to be published on IEEE Xplore Digital Library.

For early registration benefit please complete your registration by clicking on the following Link: <https://forms.gle/8e6RzNbho7CphnYN7> on or before 31 May 2023.

Registration fee details are available @ <https://icac3n.in/register>.
https://drive.google.com/file/d/1RGU6i4eGUI5B07zOfgRmDRfjyOhOhiC6/view?usp=share_link

You can also pay the registration fee by the UPI. (UPI id - icac3n@ybl) or follow the link below for QR code:
https://drive.google.com/file/d/1Ry-sF0apvy_0zUjM8INW02IZgWAsE0YD/view?usp=sharing

You must incorporate following comments in your final paper submitted at the time of registration for consideration of publication with IEEE:

Reviewers Comments:

The topic chosen "Quora Question Pairs using PERT" is interesting and relevant.
The formatting of paper is not proper. Formatting must be strictly as per template.
Author list formatting is not proper. All authors information must be complete and should be in proper format and as per the sequence desired.
References are not in proper format. Format and assign number to the references properly.
An overview of paper is desired to eradicate typo and grammatical error.
All figure and tables must be properly captioned and numbered as per IEEE conference template.

Editor Note:

1. All figures and equations in the paper must be clear. Equation and tables must be typed and should not be images.
2. Final camera ready copy must be strictly in IEEE format available on conference website www.icac3n.in.
3. Transfer of E-copyright to IEEE and Presenting paper in conference is compulsory for publication of paper in IEEE.
4. If plagiarism is found at any stage in your accepted paper, the registration will be cancelled and paper will be rejected and the authors will be responsible for any consequences. Plagiarism must be less than 20% (checked through Turnitin). However the author will be given fair and sufficient chance to reduce plagiarism.
5. Change in paper title, name of authors or affiliation of authors will not be allowed after registration of papers.
6. Violation of any of the above point may lead to rejection of your paper at any stage of publication.
7. Registration fee once paid will be non refundable.

If you have any query regarding registration process or face any problem in making online payment, you can Contact @ 8168268768 (Call) / 9467482983 (Whatsapp/UPI) or write us at icac3n23@gmail.com.

Regards:
Organizing committee
ICAC3N - 2023

Download the CMT app to access submissions and reviews on the move and receive notifications:
<https://apps.apple.com/us/app/conference-management-toolkit/id1532488001>
<https://play.google.com/store/apps/details?id=com.microsoft.research.cmt>

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our [Privacy Statement](#).

Microsoft Corporation
One [Microsoft Way](#)
Redmond, WA 98052



Shubham Agarwal <shubhamagarwal904@gmail.com>

Notification of acceptance of paper id 884

2 messages

Microsoft CMT <email@msr-cmt.org>
Reply-To: Arvind Dagur <arvinddagur@gmail.com>
To: Shubham Agarwal <shubhamagarwal904@gmail.com>

4 February 2023 at 23:45

Dear Shubham Agarwal,

Congratulations...

Your paper / article paper id 884: A Comparative Study of different BERT modifications has been accepted for publication in International Conference on Artificial Intelligence, Blockchain, Computing and Security .

Please ensure the following before registration and uploading camera ready paper.

1. Paper must be in Taylor and Frances Format.

Template and copyright link: <http://www.guconferences.org/ICABCS/assets/pdf/Template.zip>

2. Minimum 12 reference in the paper and all references must be cited in the text. Please follow the template.

3. The typographical and grammatical errors must be carefully looked at your end.

4. Complete the copyright form (available at template folder).

5. The regular fee will be charged up to 5 pages and after that additional Rs.1000 per additional page will be charged.

6. Maximum 8 pages are allowed.

7. Reduce the pilgrims below 10% excluding references.

8. Certificates can be issued to all unregistered authors on the payment of 1000/- INR per author.

9. Last Date of registration: 08/02/2023

Registration Link:

<https://docs.google.com/forms/d/e/1FAIpQLSegdrlPKXJpNlu-nN06HZAtYZKvHxqQL50n3CufxrKFJP75hg/viewform>

Bank Account Details :

Name of Account GU Conferences

Account Number 6717000100025845

IFSC Code PUNB0671700

MICR Code 110024298

Bank Name Punjab National Bank

Branch Punjab National Bank, Sector-63, Gautam Buddha Nagar, Noida-201301,UP

International Conference on Artificial Intelligence, Blockchain, Computing and Security
2/23/2023 to 2/24/2023

With Regards:

Dr. Arvind Dagur

Conference Chair

International Conference on Artificial Intelligence, Blockchain, Computing and Security

Download the CMT app to access submissions and reviews on the move and receive notifications:

<https://apps.apple.com/us/app/conference-management-toolkit/id1532488001>

<https://play.google.com/store/apps/details?id=com.microsoft.research.cmt>

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our [Privacy Statement](#).

Microsoft Corporation

One [Microsoft Way](#)

Redmond, WA 98052

Shubham Agarwal <shubhamagarwal904@gmail.com>
To: "minnijain@dtu.ac.in" <minnijain@dtu.ac.in>

5 February 2023 at 12:56

[Quoted text hidden]

ICAC3N-21: 3rd IEEE International Conference on Advances in Computing, Communication Control and Networking

Galgotias College of Engineering & Technology (GCET), Greater Noida U.P. (India) -201306
Greater Noida, India, December 17-18, 2021

Conference website	https://icac3n-21.in/
Submission link	https://easychair.org/conferences/?conf=icac3n21

Topics: [evolutionary computing](#) [big data](#) [machine learning](#) [networking](#)



About ICAC3N-21:

3rd IEEE International Conference on Advances in Computing, Communication Control and Networking (ICAC3N-21) will be held during **December 17-18, 2021** in **Galgotias College of Engineering and Technology, Greater Noida, India**. The conference is an international forum which aims to bring together leading academician, researchers and research scholars to exchange and share their experiences and hard-earned technological advancements about all aspects of based on their research related to Computing, Communication Control & Networking. We invite all leading researchers, engineers and scientists in the domain of interest from around the world. We warmly welcome all authors to submit your research papers to ICAC3N-21, and share the valuable experiences with the scientist and scholars around the world.

IEEE Conference Record No. #53548

Publication and Indexing

ICAC3N-21 is indexed in SCOPUS and Google Scholar. All registered & presented papers are available on IEEEXplore DigitalLibrary.



HOME ABOUT **PROCEEDINGS** CALL FOR PAPERS REGISTRATION SPEAKERS COMMITTEES PROGRAM SCHEDULE CONTACT US

CALL FOR PAPERS

Authors are invited to contribute to the upcoming conference that illustrate the latest research work, Research Projects, Surveying work, and the industrial work that explain the significance of your work.
Papers submitted to ICAC3N-21 will undergo a double-blind review process. All papers that are accepted and presented in ICAC3N-21 will be published in the ICAC3N-21 Conference Proceedings by Taylor and Francis.
It will be published in SCOPUS index Proceedings
Topics of interest for submission are mentioned below, but are not limited to:

> Track 1: Artificial Intelligence

- Intelligent systems
- Electronics and Signal