Develop New Transformer Architecture For Question and Answering(QandA)

Nirbhay P. Tandon

June, 2021

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background Of The Study

1.2 Aims And Objectives

1.3 Scope Of The Study

1.4 Significance Of The Study

1.5 Structure Of The Study

Chapter 2

Literature Review

We have divided this section into 3 key subsections. Section 2.1 deals with the early days of Question-Answering based systems using RNNs and the common challenges that were faced for them. We then move on to section 2.2, where we see the use of LSTMs

2.1 Question Answering Using Neural Nets

2.2 Question Answering Using LSTMs

2.3 Question Answering Using Transformers

Outlined below are some of the most important pieces of research that relate directly to the work done for Transformers and Q&A based systems.

1. Work done in the field of Long short-term and gated recurrent (Hochreiter et al., 2001) and (Zhou et al., 2016) neural networks, in particular, has been established as a state of the art approach in sequence modelling, transduction problems such as language modelling and machine translation. In their paper Attention Is All You Need,(Vaswani et al., 2017) the team set out to resolve problems in the parallelization and increased compute times of recurrent models. The inherently sequential nature of RNNs causes issues in memory constraints, leading to reduced batch sizes.

   The architecture for a *Transformer* in this paper is outlined as having

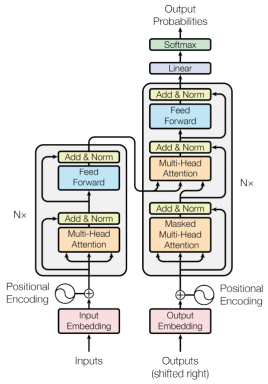


Figure 2.1: Transformer Architecture built by (Vaswani et al., 2017)

an encoder that maps input sequences to a continuous representation. The architecture can be seen above in Figure 1. This is then decoded into an output sequence of symbols one at a time. Each step is auto-regressive, i.e. it consumes the previously generated symbols as additional input when creating the next. This is similar to an ensemble model. Stacks of 6 encoder layers and 6 decoder layers is used.
The encoder layers each have 2 sub-layers of a multi-head self-attention and the other a simple, position-wise fully connected feed-forward network layer.
The decoder layer is similar to the encoder layer and has an additional 3rd sub-layer that performs multi-head attention over the output of the encoders. There is also normalization and the outputs are prevented from attending to subsequent positions.
The attention mechanism can be described as mapping a query to a set of key-value pairs. This can be seen from Figure 2, below.
The evaluations performed on the Wall Street Journal dataset(Marcus et al., 1993), using 40k sentences, showed that even without task-specific tuning the model had better results with a fraction of the training cost.

2. The paper on BERT, which is *Bidirectional Encoder Representations from Transformers* (Devlin et al., 2018), introduces a new language model. This model is truly fascinating in many ways. First and foremost it is designed to pre-train deep bidirectional representations using

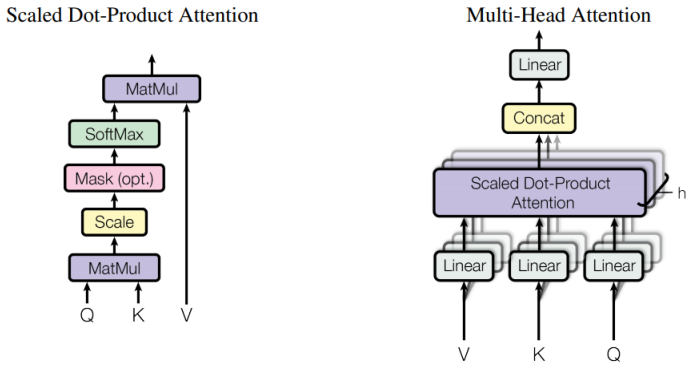


Figure 2.2: Scale Dot and Multihead Attention
Models Vaswani et al. (2017)

unlabelled data. This is done by jointly conditioning context in all layers to the right and left. This pre-training allows the model to be fine-tuned simply using one additional output layer. These features make this model conceptually simple and very powerful empirically.

BERT employs language pre-training (Dai and Le, 2015) which has shown significant advantages in many applications e.g. paraphrasing, language level inference etc. These tasks aim to highlight the relationships between sentences through contextual understanding as well as by using tokenized outputs.

BERT was tested on The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) which has a large number of diverse NLU tasks. BERT performed extremely well on the 11 NLP tasks that the authors ran it against. it showed an average accuracy improvement of 4.5% and 7% when compared to the previous state of the art models. Results of BERT and its significant gains make it one of the best candidate models for NLU tasks.

Several advancements have been made to the BERT model to make it fast, better at understanding and even performing application-specific tasks such as Q&A.

The SQuAD 2.0 dataset has inspired an LSTM based FastQA (Weissenborn et al., 2017) model architecture. This architecture takes cues from

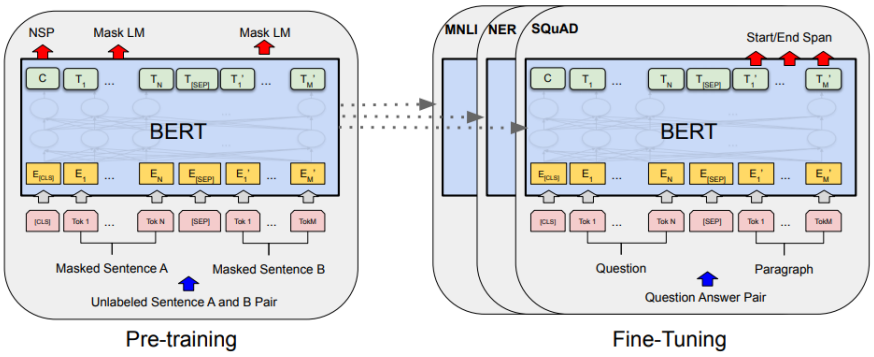


Figure 2.3: Pre-training and Fine Tuning procedures for BERT (Devlin et al., 2018)

the work done by Hochreiter and Schmidhuber in their work on Long Short-Term Memory Architecture (Hochreiter et al., 2001), to create a model specifically for end-to-end question answering systems.

A common theme with BERT is that it takes a long time to train. Especially in the BERT based RoBERTa architecture Liu et al. (2019). RoBERTa takes on average 4-5 times more time to train than BERT, however, it also shows a maximum of 20% improvement over BERT, depending on the application. ALBERT, which is A Liter BERT, Lan et al. (2019) was developed specifically to deal with memory limitations and reduce training times. ALBERT's XXL implementation has been documented to perform better in models trained on the BOOKCORPUS and Wikipedia ones by at least 2% across multiple applications. In a smaller amount of time.

However, none of these model architectures has been able to address all the problems and serve as more generic solutions to multiple end-to-end sequence encoding problems across various applications.

Our work is primarily focused on building a fast model that allows for higher accuracy in responses specifically with Q&A systems.

2.4 Comparison Of Techniques

2.5 Summary

Chapter 3

Research Methodology

DONT DO ANY ANALYSIS HERE

3.1 Data Selection

The SQuAD 2.0 Dataset (Rajpurkar et al., 2018), was developed with funding from Facebook to help address some major issues with existing datasets. Most datasets focus on questions that can be easily answered or use of automatically generated, unanswerable questions which are easily identifiable. The SQuAD 2.0 dataset resolves this by combining the SQuAD dataset along with 50,000 crowd worker generated unanswerable questions. The key feature of these being that the unanswerable questions must look similar to answerable ones. For a model to be successful on this new dataset, it must be able to answer all possibly answerable questions as well as determine when no answers are provided for a question in the given paragraph and abstain from answering. A comparative study was done for a Natural Language Understanding(NLU) task that obtained an 86% score on SQuAD 1.1, only got 66% on the new 2.0 dataset. The dataset helps bridge the gap between true NLU and machine understanding by using the concept of Relevance. Through comparisons with various datasets such as RACE, MCTest, QASENT etc. they have identified the missing links like negative examples, antonyms and helped fill the gap. This dataset forces the models to understand whether a paragraph span has the answer to the question posed.

3.2 Data Pre-processing And Transformation

3.3 Existing Models And Benchmarks

Chapter 4
Architecture Creation

4.1 Drawbacks Of Current Architectures

4.2 Proposed Architecture Improvements

4.3 Architecture Refinement

Appendices

Chapter A

Research Proposal

A.1 Introduction

Question-Answering based systems have gained a lot of popularity, especially in the form of "chatbots". These systems depend highly on contextual understanding of the input, the training corpus and the question asked. They use this knowledge to output an answer that can help the user with whatever their query is. Recurrent neural networks and architectures based on them, have been able to provide great advancements in the field of Question-answering and chatbots in general. However, there is a behaviour of over-fitting and lack of contextual understanding of the question. This, coupled with long training times and extremely complex mathematical model designs, have often kept the field of Natural Language Processing slightly obscured from the masses.

We wish to change that. Through our work, we would like to aim at creating a sustainable, fast, easy to understand Transformer architecture for Question Answering. An advancement on the work done by the team at Google (Vaswani et al., 2017). To be able to do so, let us first understand what a *Transformer* is. A Transformer is a form of transduction model that relies solely on self-attention to figure out how to represent its inputs and outputs. It does so without the use of any sequence aligned recurrent neural networks(RNNs) or convolutions.

Through our research proposal we wish to highlight why we are going to be performing this research and putting in the effort to devise a new architecture. We have divided our research proposal into 7 sections. In Section A.1, we shall take a look at briefly introducing the concept and why our work is

necessary. Next, in Section A.2, we outline the background work that has already been done in this field and how some of the papers relate to the work that has been done. We use this as an opportunity to highlight some of the shortcomings in current architectures and modelling techniques. In Section A.3, we briefly outline the aim of our proposed research. In Section A.4, we define in some detail the work that we will do to establish our research and how we plan to quantify the work that shall be done. Section A.5, highlights our goal, which is to produce a new transformer architecture that performs better at Question Answering based tasks. In Section A.6, we have outlined the minimum hardware requirements along with the resources available to the author that will be used to conduct this research. Finally, in Section A.7, we submit a Gantt Chart to outline our plan against the number of weeks.

A.2 Background and Related Research

In this section, we shall highlight what has led us this far and some of the interesting challenges that it poses. In A.2.1, we briefly look at the history of Natural Language Processing and how some of the challenges were addressed. In A.2.2, we look at the latest research that has gone into creating the Transformer architecture, identify some of the common patterns and use that information to strategise our model in later sections.

A.2.1 Background

The area of Natural language Processing has taken significant leaps in the last two decades. Work done towards improving the ability of machine learning models to first recognize words, then sentences, followed by contextual understanding has led to several interesting and novel approaches in the field. From early on neural networks to creating Long Short-Term Memory architectures (Schmidhuber and Hochreiter, 1997) by Sepp Hochreiter and Jurgen Schmidhuber in the mid-'90s that resolved the vanishing gradient problem of classical neural networks, we have come a long way.

The latest advancements in this field come from Google's research lab in the form of *Transformers*. We look into this in a bit more detail later. However, no model can be successful without a good dataset to train on. This is where the SQuAD 2.0 dataset (Rajpurkar et al., 2018) comes in. This dataset is what forces the machine learning models to do contextual

understanding. One might even say that it forces the models to "think" for themselves before answering a task.

Let us now look at some of the key research that has been done in this regard.

A.2.2 Related Research

Outlined below are some of the most important pieces of research that relate directly to the work done for Transformers and Q&A based systems.

1. The SQuAD 2.0 Dataset (Rajpurkar et al., 2018), was developed with funding from Facebook to help address some major issues with existing datasets. Most datasets focus on questions that can be easily answered or use of automatically generated, unanswerable questions which are easily identifiable.
   The SQuAD 2.0 dataset resolves this by combining the SQuAD dataset along with 50,000 crowd worker generated unanswerable questions. The key feature of these being that the unanswerable questions must look similar to answerable ones. For a model to be successful on this new dataset, it must be able to answer all possibly answerable questions as well as determine when no answers are provided for a question in the given paragraph and abstain from answering. A comparative study was done for a Natural Language Understanding(NLU) task that obtained an 86% score on SQuAD 1.1, only got 66% on the new 2.0 dataset. The dataset helps bridge the gap between true NLU and machine understanding by using the concept of Relevance. Through comparisons with various datasets such as RACE, MCTest, QASENT etc. they have identified the missing links like negative examples, antonyms and helped fill the gap. This dataset forces the models to understand whether a paragraph span has the answer to the question posed.

2. Work done in the field of Long short-term and gated recurrent (Hochreiter et al., 2001) and (Zhou et al., 2016) neural networks, in particular, has been established as a state of the art approach in sequence modelling, transduction problems such as language modelling and machine translation. In their paper Attention Is All You Need,(Vaswani et al., 2017) the team set out to resolve problems in the parallelization and increased compute times of recurrent models. The inherently sequential nature of RNNs causes issues in memory constraints, leading to

reduced batch sizes.

The architecture for a *Transformer* in this paper is outlined as having

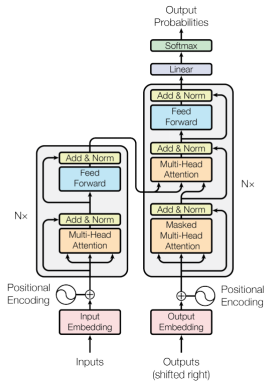


Figure A.1: Transformer Architecture built by (Vaswani et al., 2017)

an encoder that maps input sequences to a continuous representation. The architecture can be seen above in Figure 1. This is then decoded into an output sequence of symbols one at a time. Each step is auto-regressive, i.e. it consumes the previously generated symbols as additional input when creating the next. This is similar to an ensemble model. Stacks of 6 encoder layers and 6 decoder layers is used.

The encoder layers each have 2 sub-layers of a multi-head self-attention and the other a simple, position-wise fully connected feed-forward network layer.

The decoder layer is similar to the encoder layer and has an additional 3rd sub-layer that performs multi-head attention over the output of the encoders. There is also normalization and the outputs are prevented from attending to subsequent positions.

The attention mechanism can be described as mapping a query to a set of key-value pairs. This can be seen from Figure 2, below.

The evaluations performed on the Wall Street Journal dataset(Marcus et al., 1993), using 40k sentences, showed that even without task-specific tuning the model had better results with a fraction of the training cost.

3. The paper on BERT, which is *Bidirectional Encoder Representations from Transformers* (Devlin et al., 2018), introduces a new language

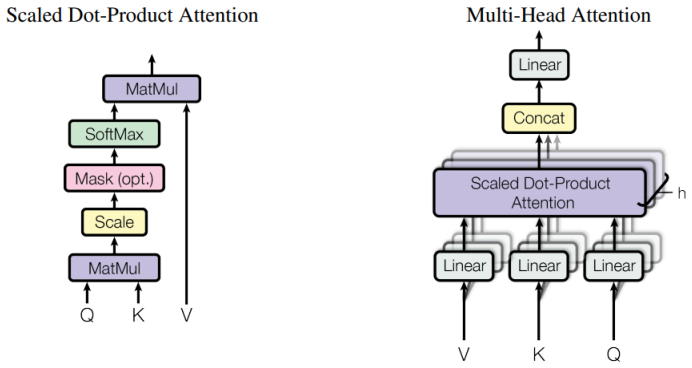


Figure A.2: Scale Dot and Multihead Attention
Models Vaswani et al. (2017)

model. This model is truly fascinating in many ways. First and foremost it is designed to pre-train deep bidirectional representations using unlabelled data. This is done by jointly conditioning context in all layers to the right and left. This pre-training allows the model to be fine-tuned simply using one additional output layer. These features make this model conceptually simple and very powerful empirically.

BERT employs language pre-training (Dai and Le, 2015) which has shown significant advantages in many applications e.g. paraphrasing, language level inference etc. These tasks aim to highlight the relationships between sentences through contextual understanding as well as by using tokenized outputs.

BERT was tested on The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) which has a large number of diverse NLU tasks. BERT performed extremely well on the 11 NLP tasks that the authors ran it against. it showed an average accuracy improvement of 4.5% and 7% when compared to the previous state of the art models. Results of BERT and its significant gains make it one of the best candidate models for NLU tasks.

Several advancements have been made to the BERT model to make it fast, better at understanding and even performing application-specific tasks such as Q&A.

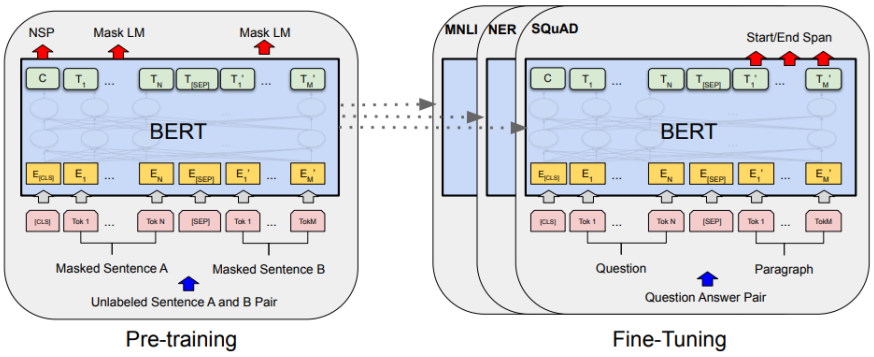


Figure A.3: Pre-training and Fine Tuning procedures for BERT (Devlin et al., 2018)

The SQuAD 2.0 dataset has inspired an LSTM based FastQA (Weissenborn et al., 2017) model architecture. This architecture takes cues from the work done by Hochreiter and Schmidhuber in their work on Long Short-Term Memory Architecture (Hochreiter et al., 2001), to create a model specifically for end-to-end question answering systems.

A common theme with BERT is that it takes a long time to train. Especially in the BERT based RoBERTa architecture Liu et al. (2019). RoBERTa takes on average 4-5 times more time to train than BERT, however, it also shows a maximum of 20% improvement over BERT, depending on the application. ALBERT, which is A Liter BERT, Lan et al. (2019) was developed specifically to deal with memory limitations and reduce training times. ALBERT's XXL implementation has been documented to perform better in models trained on the BOOKCORPUS and Wikipedia ones by at least 2% across multiple applications. In a smaller amount of time.

However, none of these model architectures has been able to address all the problems and serve as more generic solutions to multiple end-to-end sequence encoding problems across various applications.

Our work is primarily focused on building a fast model that allows for higher accuracy in responses specifically with Q&A systems.

A.3 Aims and Objectives

The main aim of this research is to propose a new transformer architecture that can perform better at Q&A using the SQuAD 2.0 dataset. We shall:

1. Implement the existing models that are available via libraries such as HuggingFace (Team), PyTorch and Tensorflow on the dataset
2. Obtain F1, validation, etc. scores for existing models and treat them as our benchmark scores
3. Identify drawbacks of the current architectures
4. Design our architecture and evaluate its performance
5. Fine-tune the architecture, re-evaluate and report improvements
6. Compare the results of our Transformer model with the benchmark scores.

A.4 Research Methodology

To implement this research we shall break the project down into 5 phases. These are outlined below.

A.4.1 Research Dataset

We have selected the Stanford Question Answering Dataset (SQuAD). This is as a reading comprehension dataset based on Wikipedia articles. It is based on questions posed by crowd-workers on a set of articles. The answer to every question is a segment of text or span, from the corresponding reading passage, or the question might be unanswerable (Rajpurkar et al., 2018).

The dataset consists of over 150,000 questions. Split into 100,000 answerable and 50,000+ unanswerable question, which were written to look similar to unanswerable questions. The challenge being that a model should be able to correctly answer the answerable questions and abstain from answering the unanswerable ones. The dataset is freely available as a part of the Transformers package in python or it can be downloaded from the SQuAD 2.0 website (Rajpurkar).

To effectively use this dataset for our purposes, let us first take a look at what its contents look like below.

Context: *"The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia."*
Question: *Who was the Norse leader?*
Answer: *Rollo*

The answer to the aforementioned question is quite simple for humans to comprehend. The challenge is for us to contextualize this and make it machine-understandable so that our model can answer it correctly.

The dataset consists of various kinds of English language examples like negation, antonyms, entity swaps, impossible conditions to answer, answerable, etc. making the dataset a well-balanced one.

To use this dataset correctly we shall perform the following pre-processing steps on it:

1. Data splitting into separate Question, Answer and Context lists.
2. Splitting the data into separate training and validation sets of question and answers using the 80/20 rule, also known as the Pareto principle. We will have 80% training data and 20% test data.
3. Tokenization of the split data to generate "context-question" pairs
4. Generating indexes for when an answer begins and ends in the dataset
5. Adding answer tokens based on their encoded positions

A.4.2 Research Benchmarks

Here we shall focus on obtaining benchmark scores for the shortlisted architectures i.e BERT (Devlin et al., 2018), DistilBERT (Sanh et al., 2019) and ALBERT Lan et al. (2019), on the above dataset.

We shall use the F1, Exact Match(EM), Recall and Training Time scores to create a benchmark to compare our architecture against. The Exact Match score will help us identify how many questions were 100% correctly answered by each model.

A.4.3 Architecture Creation

In this section we will:

1. Mathematically model a new transformer architecture
2. Code the architecture
3. Run sample dataset to identify base benchmarks
4. Run the SQuAD 2.0 dataset to obtain 1st pass performance benchmarks
5. Document architecture performance, identify pros and cons

A.4.4 Architecture Refinement

In this phase, we will focus on:

1. Reviewing the results from the previous section
2. Identifying the areas of improvement
3. Hypothesise the improvements and implement them in the architecture
4. Run the SQuAD 2.0 dataset to obtain new performance benchmarks
5. Document architecture performance, identify pros and cons

A.4.5 Model Evaluation

The training shall be carried out using the train-test loss plot to identify the optimal number of epochs for which our model needs to be run. This will also be done for the selected model architectures.

The main parameters we will use for model evaluation are F1, Exact Match(EM), Recall and Training Time.

These metrics will help us reiterate and quantify correctly if our model has improved performance or not.

A.5 Expected Outcomes

We expect that our created model is at-par, if not better, at performing Q&A than existing models.

A.6 Requirements and Resources

To successfully deliver on our research we will be utilizing the following hardware:

- EVGA GeForce RTX 2070 SUPER KO GAMING, 08G-P4-2072-KR, 8GB GDDR6, Dual Fans(Evga). This graphics card is based on the Nvidia "Turing" architecture and has 2560 CuDA cores.
- Intel 10700 processor. 8 cores, 16 threads, 16M cache(Intel).
- VENGEANCE® LPX 8GB (1 x 8GB) DDR4 DRAM 2400MHz C14 Memory Kit - Black(Corsair). 8GB x 4, 32 GB total.
- Ubuntu 20.04 Operating System
- We will also be using the latest versions of the following packages: Pandas, NumPy, SciPy, Transformers by HuggingFace, Matplotlib, Tensorflow and PyTorch. In case there are compatibility issues the appropriate versions will be mentioned. We will also mention any other packages that might be required in the course of the research.

The above hardware is available to the author and any changes to the same will be notified/highlighted in the subsequent reports.

A.7 Research Plan

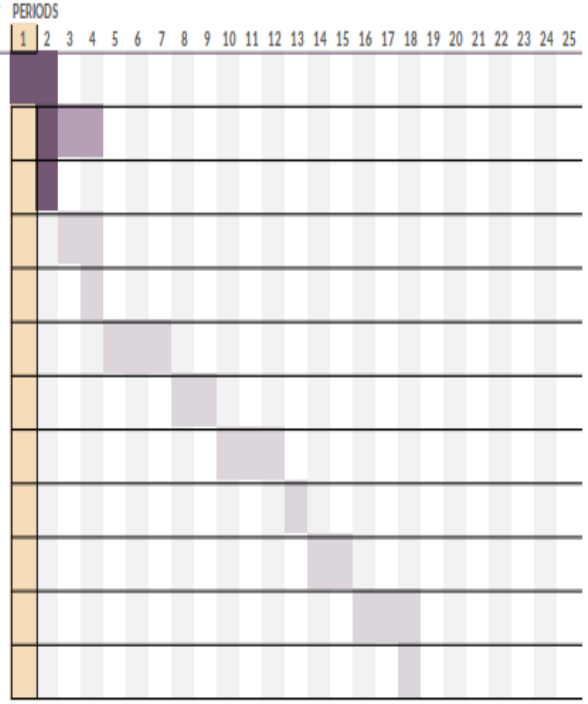
Shown on the next page is the Gantt Chart highlighting the research stages and timelines.

Compare & Contrast Existing Transformer Architectures To Develop A New Architecture

Project Planner

Period Highlight: 1 | Plan Duration | Actual Start | % Complete | Actual (beyond plan)

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Research Proposal	1	2	1	2	100%
Literature review	2	3	2	3	50%
Model & Dataset Selection	2	1	2	1	100%
Research Benchmarks	3	2			0%
Mid-thesis Report	4	1			0%
Architecture Creation	5	3			0%
Architecture Evaluation	8	2			0%
Architecture Refinement	10	3			0%
Architecture Re-evaluation & Finalization	13	1			0%
Final Architecture Conclusion	14	2			0%
Report Completion	16	3			0%
Video Presentation	18	1			



List Of Acronyms

RACE ReAding Comprehension Dataset From Examinations

Bibliography

Corsair. Vengeance® lpx 8gb (1 x 8gb) ddr4 dram 2400mhz c14 memory kit - black. URL https://www.corsair.com/uk/en/Categories/Products/Memory/VENGEANCE-LPX/p/CMK8GX4M1A2400C14. Accessed: 2021-04-16.

A. M. Dai and Q. V. Le. Semi-supervised sequence learning. *arXiv preprint arXiv:1511.01432*, 2015.

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.

Evga. Evga geforce rtx 2070 super ko gaming, 08g-p4-2072-kr, 8gb gddr6, dual fans. URL https://eu.evga.com/products/product.aspx?pn=08G-P4-2072-KR. Accessed: 2021-04-16.

S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

Intel. Intel® core™ i7-10700 processor (16m cache, up to 4.80 ghz) - product specifications. URL https://www.intel.co.uk/content/www/uk/en/products/sku/199316/intel-core-i710700-processor-16m-cache-up-to-4-80-ghz/specifications.html.

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.

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 pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

M. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. 1993.

P. Rajpurkar. Squad2.0. URL https://rajpurkar.github.io/SQuAD-explorer/.

P. Rajpurkar, R. Jia, and P. Liang. Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822, 2018. URL http://arxiv.org/abs/1806.03822.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

J. Schmidhuber and S. Hochreiter. Long short-term memory. *Neural Comput*, 9(8):1735–1780, 1997.

T. H. F. Team. Transformers. URL https://huggingface.co/transformers/. Accessed: 2021-04-16.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

W. Wang, M. Yan, and C. Wu. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. *arXiv preprint arXiv:1811.11934*, 2018.

D. Weissenborn, G. Wiese, and L. Seiffe. Fastqa: A simple and efficient neural architecture for question answering. *CoRR*, abs/1703.04816, 2017. URL http://arxiv.org/abs/1703.04816.

J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu. Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4:371–383, 2016.