Research Proposal

Nirbhay P. Tandon
MSc. Data Science
Research Title:
Develop New Transformer Architecture For
Question & Answering(Q&A)

Abstract

Attention-based Transformer architectures have become the norm of current Natural Language Processing applications. Google began this trend back in 2017 with their paper *Attention Is All You Need*, by introducing the Transformer architecture that works solely on attention mechanisms. The purpose of our work will be to explore a new kind of Transformer architecture. Compare & contrast its performance against other architectures such as BERT, DistilBERT, ALBERT etc. that are also based on SQuAD 2.0 Dataset. Through our research, we aim to produce a new Transformer Architecture that has a better performance than existing models for Question Answering in a conversational manner.

Contents

List of Figures

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 & 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 & chatbots in general. However, there is a behaviour of overfitting & lack in contextual understanding of the question. This, coupled with long training times & 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 the purpose of Question Answering. An advancement on the work done by the team at Google (Vaswani et al., 2017).

We have divided our research proposal into 7 sections. In Section 1, we shall take a look at briefly introducing the concept & why our work is necessary. Next, in Section 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 & modelling techniques. In Section 3, we briefly outline the aim of our proposed research. In Section 4, we define in some detail the work that we will do to establish our research & how we plan to quantify the work that shall be done. Section 5, highlights our goal, which is to produce a new transformer architecture that performs better at Question Answering based tasks & is capable of doing so in a conversational manner. In Section 6, we have outlined the minimum hardware requirements along with the resources available to the author. Finally, in Section 7 we submit a Gantt Chart to outline our plan against the number of weeks.

2 Background & Related Research

In this section, we shall highlight why this research has been conducted, what has led us this far & some of the interesting challenges that it poses. In 2.1, we briefly look at the history of Natural Language Processing & how some of the challenges were addressed. In 2.2, we look at the most interesting & latest research that has gone into creating the Transformer

architecture, identify some of the common patterns & use that information to strategise our model in other sections.

2.1 Background

The area of Natural language Processing has taken significant leaps in the last two decades. Work was done towards improving the ability of machine learning models to first recognize words, then sentences, followed by contextual understanding has led to several interesting & novel approaches in the field. From early on neural networks to creating Long Short-Term Memory architectures(Schmidhuber and Hochreiter, 1997) by Sepp Hochreiter & Jurgen Schmidhuber helped 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*. In their paper Attention Is All You Need(Vaswani et al., 2017) Vasvani et. al demonstrated how replacing the *encoder-decoder* based recurrent layers with *multi-headed self-attention* based ones removes the need for recurrence and convolutions entirely. We look into this in a bit more detail later.

2.2 Related Research

Outlined below are some of the most important pieces of research in the field of Natural Language Processing based neural networks, long short-term memory architectures & transformers.

1. The SQuAD 2.0 Dataset by (Rajpurkar et al., 2018), was developed with funding from Facebook to help address some major issues with existing datasets. Key pain points of these being a 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 most important 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 & abstain from answering. A comparative study done for a natural language understanding 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 natural language understanding & 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 & 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 & gated recurrent (Hochreiter et al., 2001) & (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 & increased compute times of recurrent models. The inherently sequential nature highlights the issues in memory constraints, leading to reduced batch sizes. The architecture for a *Transformer* in this paper is out-

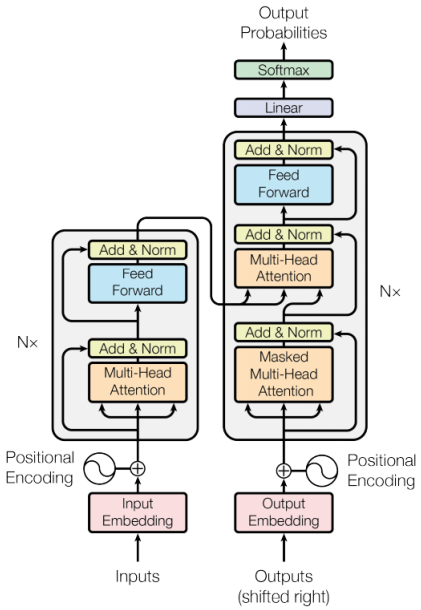


Figure 1: Transformer Architecture built by Vaswani et al. (2017)

lined as having 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 tree architecture. Stacks of 6 encoder layers & 6 decoder layers is used.
The encoder layers each have 2 sub-layers of a multi-head self-attention & the other a simple, position-wise fully connected feed-forward network layer.
The decoder layer is similar to the encoder layer & has an additional 3rd sub-layer that performs multi-head attention over the output of the encoders. There is also normalization & 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(Vaswani et al., 2017). This can bee seen from 2. 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.

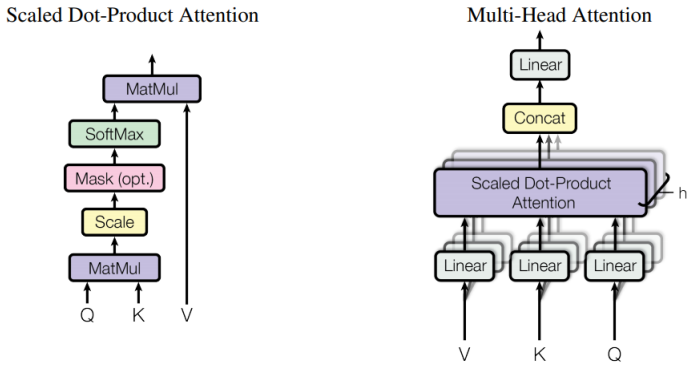


Figure 2: Scale Dot & Multihead Attention ModelsVaswani et al. (2017)

3. bert Language model pre-training has been shown to be effective for improving many natural language processing tasks (Dai and Le, 2015; Peters et al, 2018a; Radford et al, 2018; Howard and Ruder, 2018). The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations. The authors present BERT fine-tuning

results on 11 NLP tasks. 4.1 GLUE

The General Language Understanding Evaluation (GLUE) benchmark (Wang et al, 2018a) is a collection of diverse natural language understanding tasks. The authors present BERT fine-tuning results on 11 NLP tasks. The General Language Understanding Evaluation (GLUE) benchmark (Wang et al, 2018a) is a collection of diverse natural language understanding tasks. To fine-tune on GLUE, the authors represent the input sequence as described, and use the final hidden vector where C belongs to set RH, corresponding to the first input token ([CLS]) as the aggregate representation. The only new parameters introduced during fine-tuning are classification layer weights W belongs to RKxH, where K is the number of labels. Results: The best performing method concatenates the token representations from the top four hidden layers of the pre-trained Transformer, which is only 0.3 F1 behind fine-tuning the entire model. This demonstrates that BERT is effective for both finetuning and feature-based approaches. Using only the RND strategy performs much worse than the strategy as well Conclusion: Recent empirical improvements due to transfer learning with language models have demonstrated that rich, unsupervised pre-training is an integral part of many language understanding systems. These results enable even low-resource tasks to benefit from deep unidirectional architectures. The authors' major contribution is further generalizing these findings to deep bidirectional architectures, allowing the same pre-trained model to successfully tackle a broad set of NLP tasks. (Devlin et al., 2018)

4. fastqa (Weissenborn et al., 2017)
5. roberta (Liu et al., 2019)
6. alert (Lan et al., 2019)
7. distilbert Sanh et al. (2019)
8. schmidhuber (Schmidhuber, 2015)
9. levy (Levy et al., 2017)
10. contextual (Akbik et al., 2018)
11. lstm (Hochreiter et al., 2001)
12. zhang (Zhang et al., 2017)

3 Aims & Objectives

The main aim of this research is to propose a new transformer architecture that can perform better at conversational Question & Answering from the SQuAD 2.0 dataset(Rajpurkar et al., 2018). We shall:

1. Implement the existing models that are available via libraries such as HuggingFace(Team), PyTorch & Tensorflow, run the SQuAD 2.0(Rajpurkar et al., 2018)
2. Obtain F1, validation, etc. scores for existing models & treat them as our benchmark scores
3. Identify drawbacks of the current architectures
4. Design our architecture & evaluate its performance
5. Fine-tune the architecture, re-evaluate & report improvements
6. Compare the results of our Transformer model with the benchmark scores.

We hope to establish our proposed transformer architecture as a competent enough contender to be used within both industry & academia.

4 Research Methodology

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

4.1 Research Dataset

We have selected is the Stanford Question Answering Dataset (SQuAD). It was created as a reading comprehension dataset with the help of crowd workers. It is based off of questions posed by these workers on a set of Wikipedia articles, where 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 & over 50,000 unanswerable question, that were written by crowd workers to look similar to unanswerable questions. The challenge being that a model should be able to correctly answer the answerable questions & abstain from answering the unanswerable ones. In their paper *Know What You*

Don't Know: Unanswerable Questions for Squad, Rajpurkar et al. (2018) the authors have described how they 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 of the aforementioned question is quite simple for humans to comprehend. The challenge is for us contextualize this & 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 & Context lists.
2. Splitting the data into separate training & validation sets of question & answers using the 80/20 rule, also known as the Pareto principle. We will have 80% training data & 20% test data.
3. Tokenization of the split data to generate "context-question" pairs
4. Generating indexes for when an answer begins & ends in the dataset
5. Adding answer tokens based on their encoded positions

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) & Albert Lan et al. (2019), on the above dataset (Rajpurkar et al., 2018).

We shall use the F1, Exact Match & Accuracy scores to determine if our model has performed better or not.

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(Rajpurkar et al., 2018) to obtain 1st pass performance benchmarks
5. Document architecture performance, identify pros & cons

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 & implement them in the architecture
4. Run the SQuAD 2.0 dataset(Rajpurkar et al., 2018) to obtain new performance benchmarks
5. Document architecture performance, identify pros & cons

4.5 Model Evaluation

The training shall be carried out on each of the models for 100 epochs. We shall also look at the specificity/recall of these results to better understand if our work was done correctly or not. F1, accuracy, recall, specificity

5 Expected Outcomes

We expect that our created model is at-par, if not better performing than existing models for Question Answering based problems.

6 Requirements & 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 & 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 & 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 & any changes to the same will be notified/highlighted in the subsequent reports.

7 Research Plan

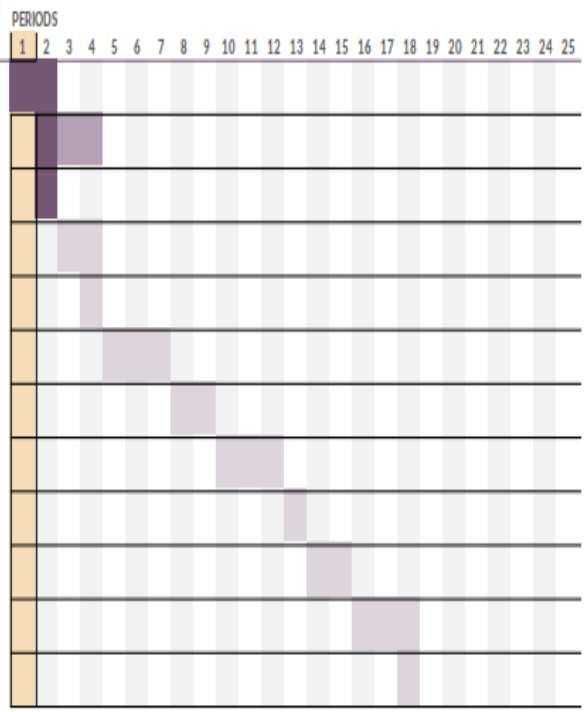
Shown on the next page is the Gantt Chart highlighting the research stages & timelines.

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			



References

A. Akbik, D. Blythe, and R. Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649, 2018.

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.

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.

O. Levy, M. Seo, E. Choi, and L. Zettlemoyer. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*, 2017.

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. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

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.

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`.

Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1004. URL `https://www.aclweb.org/anthology/D17-1004`.

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.