OPTIMIZING AND IMPROVING QUESTION ANSWERING(QA) SYSTEM PERFORMANCE USING LANGUAGE HEURISTICS AND KNOWLEDGE DISTILLATION

Prasanth Yadla

Department of Computer Science, North Carolina State University, Raleigh, NC, USA

ABSTRACT

Question Answering (QA) systems are advanced platforms designed to automatically respond to human queries expressed in natural language by utilizing pre-structured databases or collections of unstructured text documents. These systems represent a convergence of Natural Language Processing (NLP) and Information Retrieval (IR). Despite significant progress, challenges persist in reducing training time for large-scale datasets and improving model performance across diverse scenarios.

This research builds upon the BERT implementation of QA systems, introducing key innovations to address existing limitations. We employ Knowledge Distillation, a regularization technique, to compress the learned representations of deep learning models, making them more efficient. Additionally, we integrate Data Augmentation to enrich the training dataset by generating diverse linguistic variations, thereby enhancing the model's robustness. Furthermore, Linguistic Post-Processing is applied to refine predictions, leveraging domain-specific heuristics to minimize false positives and improve reliability.
The proposed system is validated using the Stanford Question Answering Dataset (SQuAD 2.0). By combining data augmentation, knowledge distillation, and linguistic knowledge, we aim to optimize the pipeline, reducing computational overhead while maintaining high accuracy. These advancements have broad applications, including real-time chatbot systems, domain-specific question answering, and efficient information retrieval for large-scale datasets.

KEYWORDS

Natural Language Processing, Question Answering, Knowledge Distillation, Language Heuristics, Deep Learning

1. INTRODUCTION AND BACKGROUND

Question answering (QA) systems have advanced significantly in recent years. A notable example is IBM's Watson, which achieved a major milestone by outperforming Jeopardy champions Brad Rutter and Ken Jennings by a wide margin. QA systems are also utilized in chatbots to provide answers within specific domains or topics.

Typically, QA systems undergo a pre-training phase where they process a corpus or documents containing the required information, which forms the dialog context. Based on this training, the system predicts answers to user questions. For instance, a model trained on a passage about the

David C. Wyld et al. (Eds): DSML, ARIA, NLP, CSEN, BIBC, EDTECH - 2024
pp. 71-81, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.142405

Apollo Space Program could answer questions like, 'What space station was the manned mission in 1973?'

However, training on large passages or corpora, particularly in real-time chatbot interactions, can be time-intensive, leading to delays while users wait for responses. Reducing the training time for extensive context documents is crucial for enabling immediate system responses. Our proposed approach integrates language text augmentation, linguistic post-processing, optimization, and dimensionality reduction techniques into traditional QA systems. This enhances both accuracy and performance, addressing the challenges of real-time applications.

We are surrounded by massive amounts of information in full-text documents via the web. Usually, we are interested in knowing the answer to our question rather than looking at the document [1]. QA systems are useful in retrieving useful information from the web and providing insights. The QA process can be broken into two parts:

1. Information Retrieval: Finding the document containing the answer to the question
2. Reading Comprehension: Given the document find the answer to the question

Here, we are concerned with the reading comprehension part of the Question-Answering System. We primarily rely on the Stanford Question Answering Dataset (SQUAD 2.0). SQUAD is a reading comprehension dataset consisting of 10,000+ questions posed by crowd workers on a set of Wikipedia articles, where the answer to each question is a segment of text from the corresponding reading passage.

2. RELATED WORK

For this project, we use Bidirectional Encoder Representations from Transformers (BERT) as our baseline model. BERT is the state-of-art model for SQuAD 2.0, which is designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers of Transformers [3]. This model achieves excellent performance on various NLP tasks with two noteworthy pre-training tasks, Masked Language Modelling (MLM) and Next Sentence Prediction (NSP).

Recently, Google AI Language pushed their model into a new level on SQUAD 2.0 with N-gram masking and synthetic self-training. Compared to BERT’s single word masking, N-gram masking training enhanced its ability to handle more complicated problems.

However, pre-training tasks is usually extremely expensive and time-consuming. For us, data augmentation, as an effective technique to improve the model performance in many Natural language Processing tasks [4] can also help break the limitation of dataset size and achieve better performance.

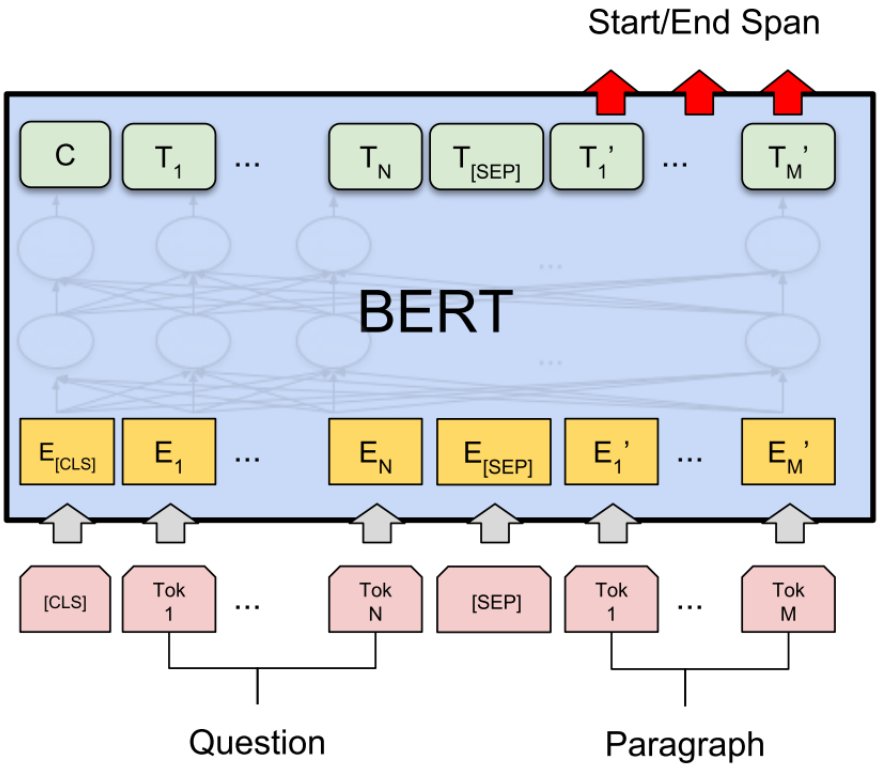

Figure 1: Architecture

Wei and Zou [4] proposed Easy Data Augmentation technique (EDA) with four simple operations to enhance the training dataset for text classification. In particular, they perform synonym replacement, random insertion, random swap, and random deletion. However, no one has tried this method for a more challenging task: question answering. Inspired by their method, we designed a similar technique for the SQuAD v2.0 dataset. In the QANet paper [6], researchers augmented the data by adding back-translated data and tuned the augmented data ratio to achieve great performance. We decided to use the data augmentation technique [4] to improve the SQuAD 2.0 training set.

Besides, researchers found linguistic patterns and knowledge base can benefit question answering [26]. Their findings inspired us to design a series of linguistic rules to help our model better predict answers. There also has been work on Training BERT Models with Knowledge Distillation and Augmentation [5]. Our work is largely derived from the same.

3. PROPOSED APPROACH

Our first step in the design of the QA system pipeline is to pre-process the data. Our novelty also lies majorly in this step. We use the Stanford Question Answering Dataset (SQUAD 2.0) readily available. An example of this is the following:

Context:

Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ bee-YON-say) (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the

late 1990s as lead singer of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".

Question:

When did Beyonce start becoming popular?

Text Span:

In the late 1990s

An example data augmentation change would be to **replace "lead" with "leading"**. This replacement will not change the answer span from the context. Similarly, replacing the words with corresponding synonyms should not change the context and would lead to a more richer training dataset. We hope that this would increase the diversity of the existing training dataset and prevent our model from overfitting or output a wrong text span as a prediction.

Our automated pipeline fully utilizes BERT model. We finetune the original model by experimenting with a variety of parameters, including learning rate, dropout, batch size and training epochs. Finally, throughout our experimentation, we develop a variety of "squad features" utilizing various NLP techniques. BERT stands for Bidirectional Encoder Representations from Transformers. BERT relies on several layers of Transformer blocks.
With this new vocabulary, we generate the embedding for BERT using GloVE, Sentence embedding and positional embedding.

The exhaustive list of hyperparameters we intend to tune and use in our BERT model are batch size, max sequence length, learning rate, number of train epochs, base/large pre-trained model, uncased/cased model. We will try to vary each hyperparameter while keeping the others constant and record the evaluation metrics (F1- score and EM score) for each combination and infer out a table for the same. *However, due to computational capability limitations, we restrict our work to using pre-trained models only. We do not perform training and hyper-parameter tuning.*

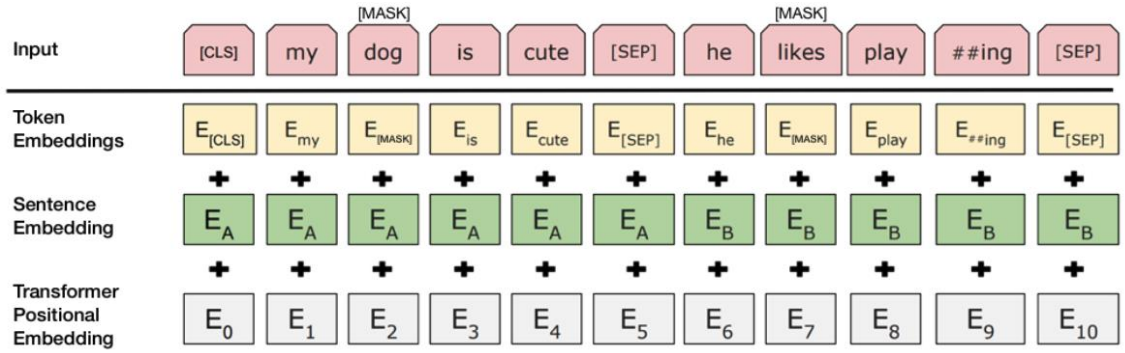

Figure 2: Word Embedding in BERT

Post processing using linguistic knowledge:

During prediction time, the probability for a text span from word *i* to word *j* being the answer is

$$P(i, j) = \text{Softmax}(\text{startlogit}(i) + \text{endlogit}(j))$$
where, $i < j$

If we were to incorporate linguistic knowledge to the predictions, an example definition would be the following: For "When" questions, if Text(i) belongs to ['before', 'after', 'about', 'during' ., etc] which are common prepositional words used when answering "when" questions.

4. EXPERIMENTAL DESIGN

There are two Question answering (QA) models being used in this research - Fine-tuned BERT and fine-tuned DistillBERT. Both models are trained on SQUAD2.0 dataset. We have used the dev dataset to perform inference the two fine-tuned models and compare their performances. We have also used the augmented dev set along with linguistic post-processing on each model and reported observations on the same.

4.1. Dataset Collection

We plan to apply our model to Stanford Question Answering Dataset (SQUAD v2.0). The input to the model is a question with a context paragraph, and the output should be the span of the text in the paragraph that can answer the question. Some features of SQUAD include the following:

1. It is a closed dataset meaning the answer to the question is a part of the context and is a continuous span. Therefore, the task can be simplified as finding the start and end index of the answer in the context.
2. The distribution of the question length is centered (median) around 10 words, and the distribution of the context length is centered around 110 words

4.2. System Design

We extract the questions, context and question ids from the SQUAD 2.0 dev set. Each question and its associated context are tokenized using the model-based tokenizer before being fed to the model as inputs. The model generates the probability distribution for the start and end indexes of the answer. We extract the combinations of all the answers that can be generated using a start, end index pair. We then select the top 5 answers (the one with the highest start, end logit sum, higher the sum means higher the probability). We also add an empty string as a possible answer for the question. We select the most probable answer as our prediction and store in a JSON file. We also save the difference between the null prediction and the best text prediction in a null_odds file. These values are used to calculate the null threshold which helps us improve our model scores.

4.2.1. Data Augmentation

Our implementation takes the list of contexts from SQUAD 2.0 dataset. For each context, we ignore determiners, proper nouns and punctuation. With the remaining words, with 50\% probability, we pick a synonym of the word and replace the word with a randomly selected synonym. We do that for all the words over the context. In this way, we form the augmented dataset using synonym and random replacement.

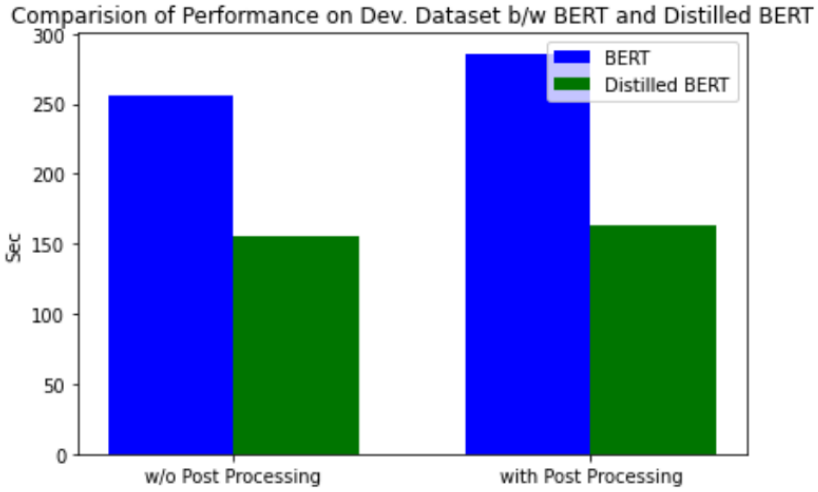

Figure 3: Performance on Dev. Dataset for BERT and DistillBERT

4.2.2. Post Processing

When considering post processing, we simply check the question type (why, when, where, other etc.), and if an answer associated with it consists of specific words, we increase the probability of that answer. This way, an answer which might have a low probability due to inaccuracy in a model, will have its probability increased and may be included in the best set of answers. Thereby, it will be chosen as the answer span of the question given the context.

	Distilled BERT (dev set)	BERT (dev set)
With Post processing	EM Score :- 64.625 F1 Score :- 68.530	EM Score :- 69.434 F1 Score :- 73.823
Without Post Processing	EM Score :- 64.575 F1 Score :- 68.547	EM Score :- 69.342 F1 Score :- 73.800

Figure 4: EM and F-1 score with and without post-processing

5. EVALUATION AND RESULTS

Following are the results obtained from each configuration of the model and dataset and post processing steps.

5.1. Performance Results

The latency measured is for inference over the SQUAD 2.0 dev set containing questions and contexts. The units are seconds.

- Latency for BERT without Post Processing: 256.47
- Latency for Distilled BERT without Post-Processing: 155.74
- Latency for BERT with Post Processing: 286.12
- Latency Distilled BERT with Post Processing: 162.94

Performance Results Analysis

We find that in the case of without Post-Processing, Distilled BERT takes less time to infer, compared to the standard BERT. This is consistent with the theoretical backing that Distilled BERT is faster. In the case of post-processing as well, we find that Distilled BERT takes less time than the standard BERT model. In this case, post-processing has incurred some cost in time.

5.2. Analysis of Post Processing using Linguistic Knowledge

We can find the results of this experiment in Figure 4.

When comparing the EM score with Distilled BERT and BERT, we find that the former is slightly less than the latter. This is consistent with the theory that Distillation compromises slightly on accuracy, to achieve better performance than BERT.

However, with post processing, we slightly get better Exact Match (EM) score and F1 score. This validates that Linguistic Post-Processing helps filter out better results.

	Distilled BERT (aug. dev. data)	BERT (aug. dev. data)
With Post processing	EM :- 51.579, F1 :- 55.440	EM :- 51.983, F1 :- 57.488
Without Post Processing	EM :- 51.065, F1 :- 55.481	EM :- 51.966, F1 :- 57.508

Figure 5: EM and F-1 score with augmented data and post-processing

The total improvements on the EM and F1 scores are limited because we only performed post processing for the “when", “Where", “Whose", “Which" questions. Only about 14\% of the questions in the dataset are these types as shown in Figure 6. For the other types of questions such as “What" or “How", it is hard to think of a good linguistic rule to apply on the predicted answers since the answers to these questions can have varied forms.

The figure below shows the distribution of question types in SQUAD 2.0.

5.3. Data Augmentation Analysis

In general, on Augmented Data, (results on figure 5) we find that the EM Score and F1 score (in 50s) less than the standard vanilla Dev. Data (in 60s). This shows that the model does not perform as effectively on the augmented data. This is true, because the models were trained on the vanilla data, and changing the testing dataset would change it’s inference and accuracy on it as well.

6. LIMITATIONS

The limitations with respect to Implementation in General are as follows.

1. Training of BERT with SQUAD 2.0 Dataset is inherently compute intensive. This is because each question has answer spans that vary and training on every combination would be time consuming. Therefore, we use pre-trained SQUAD 2.0 trained BERT models available online. We use case and uncased, along with distilled and native BERT.

2. The questions starting with When, where, whose and which amounts to 15 - 20\% of questions, in post processing. We could include more variety of question rules linguistically.

There are problems arising from a linguistic standpoint as well.

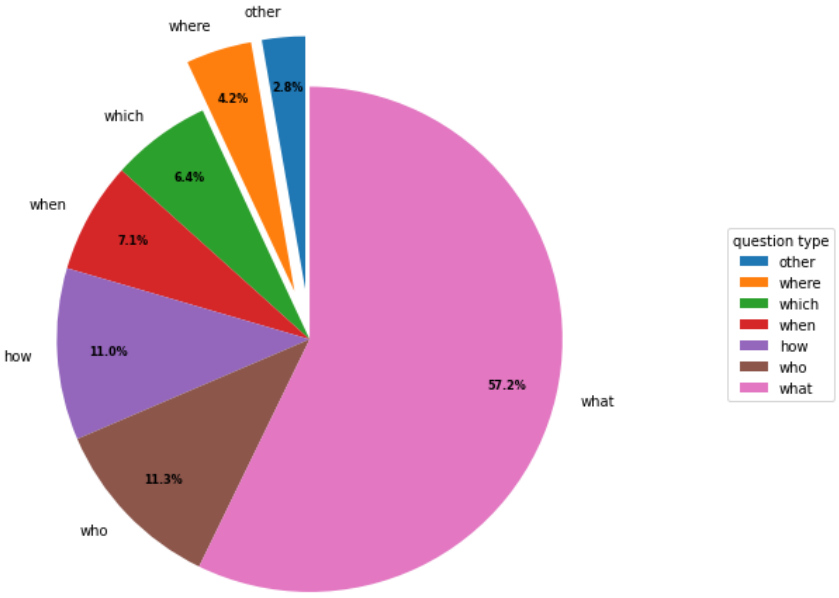

Figure 6: question type replacement chart

- Context: *...there were rich and well socially standing Chinese while there were less rich Mongol and Semu than there were Mongol and Semu who lived in poverty and were ill-treated.*

    - Question: *There were many Mongols with what unexpected status?*
    - Answer: *lived in poverty and were ill-treated vs.* ***Prediction****: less rich.*
    - Analysis: This error is caused by model's lack of knowledge in linguistic structure. Human never makes this mistake because the predicted answer and the human answer are indeed semantically similar at the first glance. However, the head of "less" is not "rich" but "Mongols", meaning "less" and "rich" is not forming a phrase here. Thus, a human will never choose "less rich" as an answer, but BERT does not understand the underlying linguistic structure and lead it to make the mistake.
    - Fix: There is an easy fix to this error. We can parse the sentence using dependency parse and check whether the predicted words are in the same clause. If not, we can directly abandon the possible answer as a candidate, and search an answer from other predicted possible answers.

- Context: *...the Commission has a monopoly on initiating the legislative procedure, although the Council is the de facto catalyst of many legislative initiatives. The Parliament can also formally request the Commission to submit a legislative proposal but the Commission can reject such a suggestion...*

    - Question: *Who is the sole governing authority capable of initiating legislative proposals?*
    - Answer: *The Commission vs. Prediction: The Council*

- Analysis: the model is sort of weak at distinguishing two close entities in the paragraph, even though the difference between "the Commission" and "the Council" is quite obvious to human readers. Both "the Commission" and "the Council" are predicted candidates by the model. The model maybe does not understand the transition word "although", which signals a contrast relationship.
- Fix: We can fix this by checking which candidate answer is the actor of "initiating the legislative procedure" in the context. We can obtain the original linguistic structure of the context and pick the one that can be the actor of action in the query.

7. CONCLUSION AND FUTURE WORK

From doing a comparative study between Distilled BERT and BERT in various environments, Data and configurations, we arrive at the following points.

BERT for QA:

- It takes more time and resources to train and infer using the BERT model.
- It is more accurate, having slightly higher EM score than Distilled BERT.
- Linguistic Post Processing helps improve the EM score by a slight margin
- Inference on Augmented Data (without Training on the same) would slightly reduce the EM score

Distilled BERT for QA:

- Since it's a lighter weight version of BERT, it computes inference faster than BERT.
- It compromises accuracy, as the EM score is slightly less than BERT
- Linguistic Post Processing helps improve the EM score by a slight margin
- Inference on Augmented Data (without Training on the same) would slightly reduce the EM score

There is much **scope for future work**. Some of them include the following:

- Training on Augmented Data would be an important step for future work, so that it can handle all forms of data
- Incorporating other types of questions in post-processing would also improve the accuracy and EM score of the model inference. Some work has already been done in this regard. We take the help of Penn TreeBank Discourses to tag the question text. Once we obtain the tags, it is more easier to identify the type of question.

ACKNOWLEDGEMENTS

The authors would like to thank the Department of Computer Science at North Carolina State University, Raleigh, NC, USA.

REFERENCES

[1] @akshaynavalakha. Nlp question answering system, 2019.

[2] D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading wikipedia to answer open-domain questions, 2017.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[4] J. Wei and K. Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019.

[5] H. J. Wen Zhou, Xianzhe Zhang. Ensemble BERT with data augmentation and linguistic knowledge on Squad 2.0, 2019.

[6] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension, 2018.

[7] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter, 2020

[8] Gou, J., Yu, B., Maybank, S. J., & Tao, D. Knowledge Distillation: A Survey, 2021

[9] Liu, J., Zhang, Y., Zheng, W., et al. Towards Robust and Reliable Question Answering: A Survey, 2022

[10] Wang, C., Shou, L., Fan, H., et al. Few-shot Knowledge Distillation for Question Answering, 2021

[11] Yasaman Boreshban, Seyed Morteza Mirbostani, Gholamreza Ghassem-Sani, Seyed Abolghasem Mirroshandel, Shahin Amiriparian Improving Question Answering Performance Using Knowledge Distillation and Active Learning, 2021

[12] Gautier Izacard, Edouard Grave Distilling Knowledge from Reader to Retriever for Question Answering, 2020

[13] Minghui Qiu, et al. Cross-domain Knowledge Distillation for Retrieval-based Question Answering, 2021

[14] X. Zhu, et al. **Advances in Natural Language Question Answering: A Review**, 2019

[15] Dominika Basaj, Barbara Rychalska, Przemyslaw Biecek, Anna Wroblewska, **How Much Should You Ask? On the Question Structure in QA Systems**, 2018

[16] Hinton, G., Vinyals, O., & Dean, J. **Distilling the Knowledge in a Neural Network**, 2015

[17] Gou, J., Yu, B., Maybank, S. J., & Tao, D. Knowledge Distillation: A Survey, 2021

[18] Yuan, L., Tay, F. E. H., Li, G., Wang, T., & Feng, J. Self-Distillation Amplifies Regularization in Hilbert Space, 2020

[19] Sun, S., Cheng, Y., Gan, Z., & Liu, J. Patient Knowledge Distillation for BERT Model Compression, 2019

[20] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. TinyBERT: Distilling BERT for Natural Language Understanding, 2019

[21] Aguilar, G., Ling, Y., Lin, Z., Luo, J., & Zhang, H. Task-Oriented Knowledge Distillation for Robust Natural Language Understanding, 2020

[22] Furlanello, T., Lipton, Z. C., Tschannen, M., Itti, L., & Anandkumar, A. Born-Again Neural Networks, 2018

[23] K.S.D. Ishwari, A.K.R.R. Aneeze, S. Sudheesan, H.J.D.A. Karunaratne, A. Nugaliyadde, Y. Mallawarrachchi, Advances in Natural Language Question Answering: A Review, 2019

[24] Minghui Qiu, Liu Yang, Chen Qu, W. Bruce Croft, Ming Gao, Jun Huang, Cross-domain Knowledge Distillation for Retrieval-based Question Answering Systems, 2021

[25] Ze Yang, Linjun Shou, Ming Gong, Wutao Lin, Daxin Jiang, Model Compression with Two-stage Multi-teacher Knowledge Distillation for Web Question Answering System, 2019

[26] Danqi Chen and Adam Fisch and Jason Weston and Antoine Bordes, Reading Wikipedia to Answer Open-Domain Questions, 2017

AUTHOR

Prasanth Yadla is a seasoned Senior Machine Learning Engineer in Seattle, WA, USA., specializing in deep learning and generative AI. With over four years of experience at top tech companies like Amazon and Apple, he has contributed to groundbreaking projects, including question answering for Amazon Alexa and Apple Intelligence. Prasanth's expertise spans foundation models, distributed training, and large-scale model deployment. He holds an MS in Computer Science from NC State University and a dual major in Physics and Computer Science from BITS Pilani. His work focuses on advancing Natural language Processing models, and multimodal foundation models with an emphasis on it’s applications to Healthcare.