QuesBELM: A BERT based Ensemble Language Model for Natural Questions

Raj Ratn Pranesh
Birla Institute of Technology, Mesra
Mesra,India
raj.ratn18@gmail.com

Ambesh Shekhar
Birla Institute of Technology, Mesra
Mesra,India
ambesh.sinha@gmail.com

Smita Pallavi
Birla Institute of Technology, Mesra
Mesra,India
smita.pallavi@bitmesra.ac.in

***Abstract*—A core goal in artificial intelligence is to build systems that can read the web, and then answer complex questions related to random searches about any topic. These question-answering (QA) systems could have a big impact on the way that we access information. In this paper, we addressed the task of question-answering (QA) systems on Google's Natural Questions (NQ) dataset containing real user questions issued to Google search and the answers found from Wikipedia by annotators.**

In our work, we systematically compare the performance of powerful variant models of Transformer architectures- 'BERT-base, BERT-large-WWM and ALBERT-XXL' over Natural Questions dataset. We also propose a state-of-the-art BERT based ensemble language model- QuesBELM. QuesBELM leverages the power of existing BERT variants combined together to build a more accurate stacking ensemble model for question answering (QA) system. The model integrates top-K predictions from single language models to determine the best answer out of all. Our model surpassed the baseline language models with the Harmonic mean score of 0.731 and 0.582 for the long answer(LA) and short answer(SA) tasks respectively, reporting an average of 10% improvement over the baseline models.

***Keywords*-Ensemble model; Question Answering; Deep learning; Natural Language Processing; Transformer Architecture**

I. INTRODUCTION

Over the last few years there has been a considerable development in the domain of Natural Language Processing. Several research work on various Natural Language Processing and Natural Language Understanding tasks such as text classification, text summarization, machine translation, text generation and many more has been conducted. The large part of their successes in this field accounts to the rapid progress and development of large-scale and powerful language models which tends to surpass the performance of previous approaches. Presence of large-scale training datasets also contribute as major factors behind the performance of these language model. For example- SQuAD [6], CoQA [11].

Question answering is a benchmark task in the field of natural language understanding (NLU). It refers to the ability of machine to give an answer based on the document collection covering wide range of topics. In this paper, we explore the question answering (QA) task using the Google Natural Questions [3] (NQ) dataset for our experiments. The Natural Questions (NQ) dataset, is consists of questions asked by people based on the Wikipedia page. Existing natural language models focus on extracting answers from a short paragraph, whereas answering Natural Questions involves reading an entire page of content for proper context. A good answer will be both succinct and relevant.

The reason behind selecting our specified models and ensembling them to generate a strong learner model was that we wanted to evaluate how capable each language model is in learning the contextual features and answering the questions. In our experiments we found that the large language model such as ALBERT-xxl perform better than comparatively smaller model such as BERT-base. The F1 score of ALBERT-xxl on NQ dataset is reported 0.700 and 0.555 for long answers and short answers respectively which is an improvement of **11.7%** and **17.6%** for BERT-base and **3.0%** and **2.52%** for BERT-large model.

Motivation: Though pre-trained language models achieved significant results in the question answering. But there always exists room for improvement. In past few years, ensemble learning has been highly successful in large number of Machine Learning tasks [1], especially in Natural Language domain. For the question answering task we designed an ensemble language model by integrating heterogeneous pre-trained transformer language models. We have proved in our experiments that the ensembles surpasses the performance of single models. QuesBELM reported a significant rise in the harmonic mean by **4.4%** and **5.81%** for long answers and short answers respectively for existing large language model performing individually on the problem set.

In our approach we used stacking as the meta-generalization ensembler [2] in order to combine top-K predictions produced by each large model. We selected the heterogeneous pre-trained bi-directional transformer models, fine-tuned them for the SQuAD v2.0 task, and then combined their top-K predictions to determine the best possible prediction for the question answering task as the ensemble's output. Each model present in the ensemble model learns during training in an idiosyncratic manner, so the output results (for the same input) of each model is expected to

978-1-7281-9180-5/20/$31.00 © 2020 IEEE

vary. This behavior is very similar to one in humans, for example different people have different opinions considering the fact that everyone is coming from a diverse background.

II. Related Work

Introduction of Pre-trained Contextual Embeddings (PCE) models played a very crucial role in the enhancement of machine's ability in NLP and NLU. With the improvement in contextual understanding of language model researchers are able to achieve high accuracy in question answering task.

For the Natural Question (NQ) dataset, in the paper [5] researcher have already applied BERT model and reported a F1 score of 64.7 and 52.7 for long and short answers respectively. The authors stated that there is plenty of room for improvement. In the paper [1], the authors emphasis on the importance and significance of ensemble machine learning models in various field and how it is being used for generating high quality results.

In our paper, we designed an ensemble language model **QuesBELM** to tackle the shortcomings of heterogeneous language models. In the following sections we have elaborated our steps involved in designing QuesBELM.

III. Data Preprocessing

The Natural Questions is a question answering dataset provided by Google's Natural Question[1], which contains 307,373 training examples, 7,830 development examples,and 7,842 test examples. Each example contains a query from Google with its respective Wikipedia page containing annotated passages present on the page that answers the query. It also contains one or more short spans from the actual answer passage. The task is to select the best short and long answers from Wikipedia articles to the given questions.

We use WordPiece embeddings with 30,522 token vocabulary to tokenize every example in NQ dataset alongwith generating multiple instances for each example. In parallel each instance is concatenated by special tokens by which each instance has the following format:

[CLS]<query>[SEP]<candidate answer>[SEP]

where [CLS] is used for the classification and [SEP] acts as separator between multiple sentences, where <query> contains tokenized query and <candidate answer> contains the answer retrieved by start and end indices of the contents from the page. We restrict each instance by limiting it to 512 tokens either by truncating or padding them. For each document we generated instances by moving a window of 512 tokens with a stride of 128 tokens.

In case if there is annotated long answer span but no short spans then the the indices points to the long span. If both the span are not available we set start and end indices to [CLS], considering them as null instances. After going through the data 98% of the generated instances were null because 51% of the document as didn't contained as positive answer. Remaining 2% of training data consisted of correct as well as positive answer.

Training the remaining dataset was considered not good enough. It was needed to train model to classify most probable answers. To continue, we sampled the data using hard negative sampling. At first we sampled the data using uniform sampling method, applied the model and tried classifying negative answers. Stored those negative answers along with their probability. Next, we generated a distribution using probability of negative answers to select negative data. We removed unnecessary data and added useful data to our training data. Next we sampled this data with equal number of positive and negative answers to train the model. It led to a training set of approximately 469,000 instances of 512 tokens each. Following [5] we used 9 HTML tag's special tokens[2] in the document to provide a notion of which segment of document the model is reading. These tokens are atomic meaning they cannot be split further according to WordPiece. For training we compute each instance with target answer type like "short" for instances that contain all annotated short spans and "yes" or " no" for yes/no annotations in a long answer span and "noanswer" for null instances.

Training set instance: Our training instances consists of a four-tuple format

<I, B, E, A>,

where 'I' is a context of 512 word-piece ids which comprises of document, question, markup and tokens, 'B, E' ϵ {0, 1, . . . , 511} are inclusive indices that points to the begin and end of the targeted answer span, and 'A' ϵ {0, 1, 2, 3, 4} denotes the annotated answer type, representing the following labels: "short", "long", "yes", "no", and "no-answer".

IV. Methodology

In this section, we elaborate the features and architecture of baseline models and our state-of-the-art ensemble model framework QuesBELM.

A. Baseline Model

BERT$_{base}$-uncased [9] In this model we used BERT-base on the NQ dataset. The experiment setting is taken as described by authors [5]. The model is fine-tuned on SQUAD v2.0 [7] dataset and trained on training instances as described in the data preprocessing section. The model returns start, end and type probability of document span as a softmax over scores computed by the BERT model. Finally all document spans are ranked and the maximum scoring span in the document is predicted as short answer span. For the long answer span, we take the top level node that contain the predicted short answer span. The model was evaluated on

[1] https://ai.google.com/research/NaturalQuestions/dataset

[2] https://github.com/google-research-datasets/natural-questionsdata-statistics

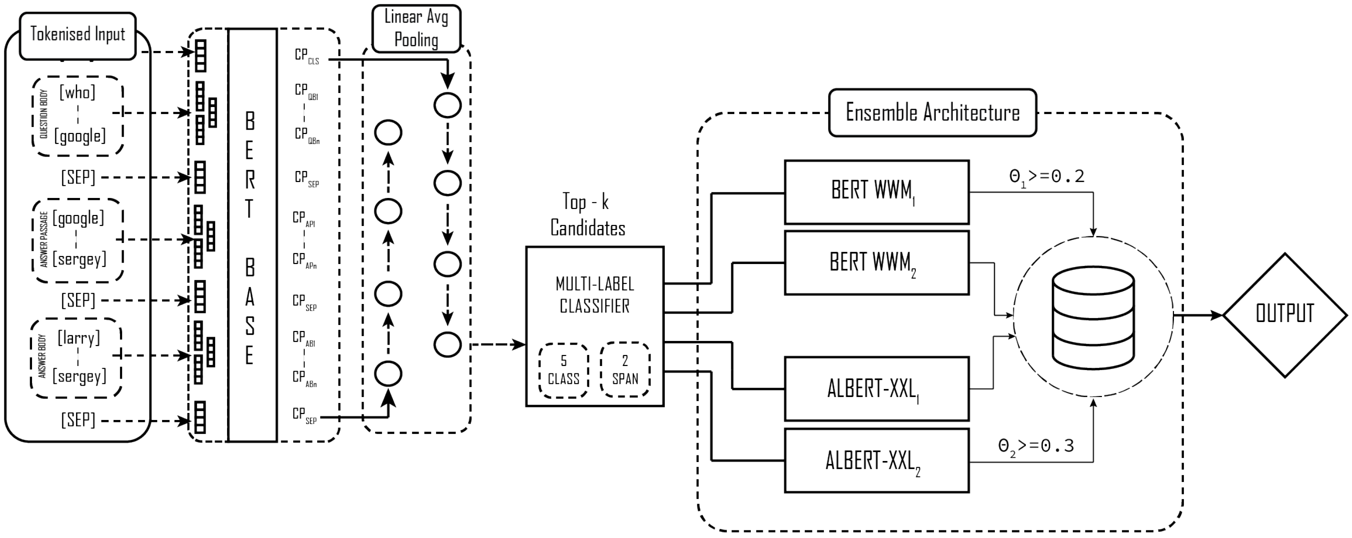

Figure 1. QuesBELM Architecture

NQ dev and test set with the hyperparameter setting reported at Table I.

BERT$_{large}$-WWM [9] BERT$_{large}$(uncased) with whole word masking is used as baseline model for the question answering task. The training and evaluation of model is done in a similar manner as described above in the BERT$_{base}$ model with hyperparameter setting for the experiment reported in the Table I.

ALBERT-XXL-v2 [10] We also used ALBERT-XXL a lighter BERT as a baseline model. The training and evaluation of model is done in a similar manner as described above in the BERT-base model with hyperparameter setting reported at TableI.

B. Ensemble Architecture

An ensemble of weak classifiers is an attitude wherein a few classifiers are consolidated together for much learning in the merging stage called the Meta state to enhance the general classifier execution. A parallel blend of the meta level learner classifiers is known as stacking which is composed of two phases. Firstly, different models (as in our study, two BERT-large-WWM and two ALBERT-XXL) are learned in the base of the data set. Then, the output of each of the model are collected to create a new dataset. In the new dataset each instance is related to the real value that it is suppose to predict. Following phase witnesses the meta-learning, in order to provide the final output. We opted to create the ensemble to leverage each individual model's strengths by manipulating the BERT models through specialization to achieve significant performance improvement.The basic idea evolves from the following concept :

$$\mathbf{S} \ \epsilon \ \{\text{Naive Learning Algorithms}\},$$

$$\mathbf{P} \ \epsilon \ \{\text{Partitions of training set } T_i\}$$

where we choose $s_i \ \epsilon \ S$ based on how well the objects agree with the partition T_i into the training and testing data set. This leads to meta generalization whereby one assigns to the partitions and roles played by the learning algorithm on the training data sets for extrapolation. We have implemented stacking as a net generalization tool involving multiple learning algorithm. Stacking uses sets of partitions of T as

$$\text{P} = \{T_i^1, T_i^2\}$$

and train one learning algorithm B on T^1 thus observe errors that the algorithm makes on input-output pairs in T_i. Next, several mappings of T_i are created $(B(T_i(.))$ and these parallel learning algorithms can be stacked together to enable multiple models depicted as weak learners but combined to produce a more powerful model. In our experiment, one Bert-base-uncased, two Bert-large-WWM and two Albert-xxlarge models were used . Later, prediction was done by 4 models only so Bert-base used for candidate generation shall not be included in the ensemble model. The parameter tuning and empirical evaluation of two groups of model is done here by Bagging followed by Stacking approach. Bagging (bootstrap re-sampling the training set)will handle homogeneous BERT large and Albert respectively with less variance, then Stacking will calculate their weighted average which will be stronger, less biased handling both models in parallel.

V. EXPERIMENT

In this section, we present our proposed model which is a 3-stage pipeline. The first stage is the fine tuning of variants of BERT model on the Stanford Question Answering Dataset (**SQuAD v2.0**). To apply data augmentation in the question answering task, we represent the input question and the answer passage as a single packed sequence as illustrated in Figure 1. Assuming the question uses 'Q' embedding and the answer passage uses 'A' embedding. The start vector S ϵ R^H and an end vector E ϵ R^H during fine-tuning. The probability of the word 'w' being in the answer span in the current paragraph is computed as

$$P(w) = \frac{exp(S \cdot C_i)}{\Sigma_j exp(S \cdot C_j)}$$

. Similarly the end of the answer span is computed. The score of a candidate from position **i** to position **j** is defined as S_i·C$_i$ + E_j·C$_j$ and the maximum scoring span where j i is used as a prediction. Both BERT$_{large}$-WWM and ALBERT-XXL were fine-tuned for 4 epochs with a learning rate of 3e-5 and a batch size of 32. The second stage marks the fetching of most probable candidates by BERT$_{base}$-Uncased model. The uncased BERT model was adopted for mapping the word-context pairs as shown in Figure 2 and for ranking of the most probable candidates from each document for reduction of long answer probabilities, firstly to top-10 and further to top-4 candidates. This is progressed by implementing Selection + Ordering utility. The selection utility performs rankings in which every candidate included in the top-K is more qualified than every candidate not included. The ordering utility prefers rankings that the excluded ones are operationalized by the difference in their qualifications (here locally optimized threshold value). Also, amongst all inclusive candidates in the top-K group, the more qualified ones are ranked above as per order.The third stage involves parallel performance by the two BERT$_{Large}$-WWM models and two ALBERT-XXL models. Next, the averaging aggregations of the weighted profiles was stacked together to generate the final output of the model.The threshold value fixation is done for both the larger transformer models as (0.2, 0.3) by the golden section search whereby scores obtained by the two points under evaluation is calculated. Next, the inner dividing point at the golden ratio is taken closer to the worse point. Similarly, for all points, we take the better point and the new point and evaluate scores to reach to optimal value to be fixed. This would ascertain that the threshold value computed is not prone to local optima. The final prediction of the resultant to the question is calculated if (null score - the score of best non-null) > threshold.

Table I
HYPER PARAMETERS SETTING

Parameter	Bert-base	Bert-large	Albert-XXL
# Batch Size	24	24	24
# Learning Rate	3e-5	3e-5	3e-5
# Max Query Length	64	64	64
# Max Sequence Length	384	384	512
# Training Epochs	4	4	4
# Doc Stride	128	128	128

Model	Long Answer Score			Short Answer Score		
	P	***R***	***F1***	***P***	***R***	***F1***
DocumentQA*	48.9	43.3	45.7	40.6	31.0	35.1
DecAtt + DocReader*	54.3	55.7	55.0	31.9	31.1	31.5
BERT$_{base}$	64.1	68.3	66.2	63.8	44.0	52.1
BERT$_{large}$ WWM	69.3	66.5	67.9	57.0	51.4	54.1
ALBERT-XXL	70.3	69.7	70.0	59.1	52.3	55.5
QuesBELM(this work)	**73.8**	**72.3**	**73.1**	**64.2**	**53.2**	**58.2**

Table II
TEST SCORES OF BASELINE AND ENSEMBLE MODEL

DocumentQA* [19], DecAtt [18], and Document Reader [21] baseline models used in NQ dataset paper [3].

The classification of the answers in the training data set was done based on the following logits :

$$\text{1 - P(noAns) = Conf(Score(longAnswer)),} \qquad \text{(a)}$$

$$\text{Score(shortAns) = max[P(shortAns),P(yes), P(no)]} \qquad \text{(b)}$$

where,

$$\textbf{P} = \text{Probability of occurrence of a class type,}$$

$$\textbf{S} = \text{Count of class type}$$

Span classification which is designated for answers without a short answer span was ignored while training the model as only generated candidates by the BERT base model were taken for further classification. Thus, the error generated by the loss of span was unfurled in training data sets. During testing phase, the span prediction was obtained by mapping the token level probability onto the word level probability considering all white space tokenizations as separators.

The training code, model and dataset used for the experiment are publicly available to facilitate reproducibility.[3]. For experiment we used NVIDIA 4 x RTX Titan(32G) GPU RAM with Core i7-9700K Processors @3.6GHz.

VI. RESULTS AND DISCUSSION

Here, we summarize the results obtained by the QuesBLEM model. We conducted our experiment initially with three language models BERT$_{base}$(12-layer, 768-hidden, 12-heads, 110M parameters), BERT$_{large}$(24-layer, 1024-hidden, 16-heads, 340M parameters.) and ALBERT-XXL(12 repeating layer, 128 embedding, 4096-hidden, 64-heads, 223M parameters) and evaluated their performance on Natural Questions dataset. On the PCE scoreboard, we reported the following individual F1 mean scores, Precision score and Recall score of all the three variants of BERT models and the QuesBELM model after fine tuning the parameters.

The harmonic mean scores of ALBERT-XXL model portrays a significant improvement of 11.7% over the individual scores of the base model of BERT when calculated for Long Bounded sentences and of approximate 17.6% for Short Bounded answers. The progressive mean score of the A Lite

[3] https://github.com/Rajratnpranesh/Ques-BELM

BERT-xxL over BERT-large-Whole Word Masked model is 3% for LB answers and was found to be 2.6% for SB answers. Upon adopting the three step modification done on the existing models, the resultant QuesBLEM Model showed an improvement of 4.4% for LB answers and 5.81% for SB answers which is quite significant taking into consideration the amount of data handled by the model.

VII. CONCLUSION

In our paper we presented a novel BERT based ensemble model for Natural Questions. We also performed a detail comparison of our model with heterogeneous Pre-trained Contextual Embeddings (PCE) models. In our study, we have established the fact that ensemble learning improves the semantic and contextual understanding of language models by helping the heterogeneous models to generalize better and tackle their shortcomings. With the optimal stacking of heterogeneous language models we were able to build QuesBELM which outperform various language model on Natural Questions dataset. Our model achieved a F1 score of 0.731 and 0.582 for long answer and short answer task respectively which is highest score out of all Pre-trained Contextual Embeddings (PCE) models. That being said,we believe that our model is far from being perfect and there is still room for improvement in the proposed design. Future work and possible experiments that can be done such as: (i) Experimenting with various model combination for designing the ensemble model, (ii) Using data augmentation for improving model's learning from data, (iii) By hyperparameter fine tuning and optimization, (iv) Multi-objective optimization,etc. Finally, we would like to conclude that question answering task is a challenging task in NLP and through stacking ensemble mode we can achieve better consistency and degree of agreement.

REFERENCES

[1] Cha Zhang and Yunqian Ma. Ensemble machine learning: methods and applications. Springer, 2012.

[2] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[3] Kwiatkowski, Tom, et al. "Natural questions: a benchmark for question answering research." Transactions of the Association for Computational Linguistics 7 (2019): 453-466.

[4] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, Kaheer Suleman. NewsQA: A Machine Comprehension Dataset. 2017.

[5] C. Alberti and K. Lee and M. Collins.A BERT Baseline for the Natural Questions,2019.

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392.

[7] Pranav Rajpurkar, Robin Jia, Percy Liang.Know What You Don't Know: Unanswerable Questions for SQuAD. 2018.

[8] Natural Question dataset by Google https://github.com/google-research-datasets/natural-questions

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[10] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In International Conference on Learning Representations, 2020.

[11] Siva Reddy, Danqi Chen, Christopher D. Manning. CoQA: A Conversational Question Answering Challenge. 2019.

[12] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretrainingapproach. 2019.

[13] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In Advances in neural information processing systems, pages 5754–5764, 2019.

[14] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. 2019.

[15] Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization. 2017.

[16] Mandar Joshi, Eunsol Choi, Daniel S. Weld, Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. 2017.

[17] The stanford question answering dataset. https://rajpurkar.github.io/ SQuAD-explorer/ .

[18] Ankur P Parikh, Oscar Tackstrom, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. 2016.

[19] Christopher Clark and Matt Gardner. Simple and effective multi-paragraph reading comprehension. 2017.

[20] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, Luke Zettlemoyer. QuAC : Question Answering in Context. 2018.

[21] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. 2017.

[22] Yi Yang, Wen-tau Yih, Christopher Meek. WikiQA: A Challenge Dataset for Open-Domain Question Answering. 2018.