# Project Report : CS 7643 — Bring in the ExBERT

Alexander Fan
Georgia Tech
afan8@gatech.edu

James Fan
Georgia Tech
jfan83@gatech.edu

James Song
Georgia Tech
jsong441@gatech.edu

## Abstract

*DistilBERT, a distilled version of the BERT model, has gained popularity for its efficiency and computational advantages. However, its small and specialized architecture poses difficulties for further optimization on multi-domain knowledge-based tasks. In this study, we present a re-engineered version of DistilBERT, enhanced with a Switch Transformer Mixture of Experts (ST-MOE) layer for Q&A. The ST-MOE layer introduces a novel mechanism for expert selection and combines feedforward networks within each expert. Through a further reduction of transformer layers and the strategic use of auxiliary losses, our novelly modified DistilBERT with ST-MOE (**ExBERT**) surpasses the performance of vanilla DistilBERT (**V-DB**) by **9-13%** on cross-domain Q&A. The cross-domain accuracy achieved with ExBERT, especially when trained on smaller datasets (¡4,000 samples), further supports the view that excellent performance can be achieved with model engineering using light-weight architectures.*

## 1. Introduction/Background/Motivation

The goal of this project was to generate an architecture modification to DistilBERT for superior cross-domain performance on Q&A tasks, while retaining its desirable attributes such as lightweight and computational efficiency. This model takes a sample of reference texts as input and can intelligently answer questions that are asked on the content of the input text.

BERT was a transformer model first presented by Devlin et al. in 2018 [1]. Marked by incredible accuracy and adaptability to a wide variety of NLP tasks, one of its few disadvantages was relative computational complexity. The distillation of BERT (i.e. Teacher-Student model) resulted in 97% of the accuracy of BERT, with 60% of the parameters ( 60M vs. 110M), and a 40% increase in training times.

Our team's preliminary results on vanilla DistilBERT (**V-DB**) suggested that DistilBERT performs best when trained on larger training sets (>40,000 examples). Indeed, DistilBERT performs exceedingly well when fine tuned datasets such as the Stanford Question and Answering Dataset (SQuAD; 80,000 samples). Interestingly, our team's preliminary results regarding the efficiency of *multi-domain* training for *single cross-domain* tasks– resulted in less impressive prediction accuracy (see 3.1).

However, architecture-specific improvements for multi-domain applications of DistilBERT remains a sparsely studied topic. Still, a large body of work exists to inform our approach towards such an endeavor, namely in multi-domain applications of transformers [2], and pre-training schemes for DistilBERT [3].

The idea to incorporate a sparse model, namely switch-transformer mixture-of-experts (ST-MOE), originated from the success of the Google Brain team in applying sparse models to single-domain tasks [4], as well as the utility of switch-transformers in the context of task-adaptation. Additionally, the sparse model would be less prone to overfitting–an *acute* challenge in the case of DistilBERT. Alone in its original form however, ST-MOE was only *particularly* accurate on knowledge-heavy tasks (such as Q&A) after training on a large labeled dataset.

Therefore, we hypothesized that incorporating ST-MOE into the DistilBERT architecture may help counteract some of its deficiencies, namely overfitting, while improving task adaptation during training on *multi-domain* datasets. To test this hypothesis, we examined the prediction performance of four variations of our novel DistilBERT with ST-MOE (**ExBERT**) model after training on varying mixtures of SQuAD and BioASQ samples.

For this project, the Stanford Question Answering Dataset (SQuAD) and the BioASQ data were chosen. The SQuAD samples represent a generalized selection of Wikipedia articles, and BioASQ is a Q&A dataset containing excerpts from biomedical publications. While SQuAD consists of 80,000 question-answer pairs, we tested 4,000 sample *mixtures* comprising of samples from both SQuAD and BioASQ (itself 4,000 samples) to be able to examine cross-domain performance as a function of fine tuning on a range of data mixtures from 100:0 to 0:100%

(SQuAD:BioASQ sample/feature ratios). Qualitatively, SQuAD Q&A tasks appeared to require more "reasoning" capabilities compared to Q&A tasks on BioASQ, which was almost purely "knowledge". However, we believe that the scientific literary-*style* of technical writing such as that found in BioASQ poses unique, if consistent, contextual and linguistic characteristics.

Our project demonstrated significant improvements in performance with multi-domain trained ExBERT, and generated insights and parameter characterizations on: 1) A novel hybrid architecture based upon DistilBERT (ExBERT) with switch-transfer-gated sparse layers with excellent cross-domain adaptivity (compared to DistilBERT), 2) the effect of mixed- domain training on cross-domain Q&A tasks for DistilBERT and ExBERT, and 3) the sensitivity of our two language models to training on *small* multi-domain datasets.

We feel that the results of this study convincingly motivate the strategic inclusion of MOE layers, even in already-efficient distilled language models, to improve multi-domain training efficiency, especially on limited training data.

## 2. Approach

The primary problem we were trying to solve was to investigate and address deficiencies in DistilBERT performance on cross-domain tasks. The project aims were achieved in three stages. **First,** we designed and implemented a novel hybrid ST-MOE/DistilBERT model. **Second,** we compared the performance of this new model to V-DB after fine tuning on a variety of mixed domain datasets, which led to results on the previously understudied regime of multi-domain training. **Third**, the ExBERT model was tuned to maximize performance on dual-domain question and answer tasks.

### 2.1. Creating the ExBERT Model

The forward pass through our novel ExBERT model involves processing input tensors through a modified DistilBERT layer, with *four to six* transformer blocks. Notably, an ST-MOE block is utilized with *8-16* experts and either one (preceding) dropout layer, or two (preceding and following) the MOE block (**Fig. 1**). The output of the ExBERT model comprises start and end logits, hidden states, and attention values. Additionally, a final loss value is calculated, from the summation of total loss and auxiliary loss, introduced by the ST-MOE modifications. Gating number (the number of experts that tokens would be sent to) was kept at 2, with a routing threshold of 0.2, training capacity factor of 1.25 (each expert could take 25% more tokens if necessary), balance loss coefficient of 1e-2, and router z-loss coefficient of 1e-3.

A primary problem we encountered with DistilBERT was its tendency to overfit after Epoch 3, particularly on small datasets. While this is working as intended (according to its authors), one of our primary challenges was to optimize the model so that ExBERT would achieve better performance through improved accuracy on the same number of epochs. This proved to be a successful approach–On the other hand, attempting to lower learning rates and increase dropout for DistilBERT/ST-MOE to fine tune over a larger number of epochs proved unsuccessful–The model had trouble fitting the data, and we suspected that a more sophisticated approach to learning rate scheduling may be necessary for this method to work properly.

Additionally, we retained *some* of the published MOE parameters that we intuited would *not* necessarily need to be retuned–for example, z-router loss was optimized by the authors for stable training and convergence. Instead, we suspected (correctly) that DistilBERT parameters would have to be modified to accommodate both the insertion of ST-MOE and heavy changes to its original architecture (see **2.4**; Fig. 1).

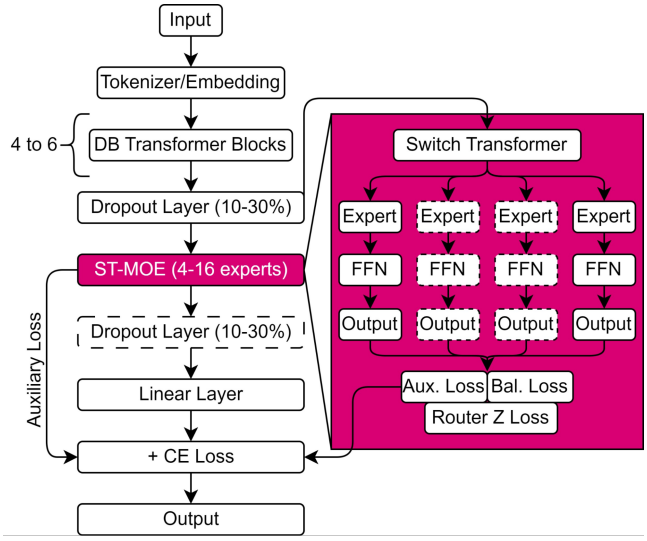


Figure 1. Schematic of ExBERT architecture, and ST-MOE block.

### 2.2. Mixed Domain Performance

#### 2.2.1 Tokenization, Training and Evaluation

The datasets were imported using the Hugging Face Load framework and tokenized using the DistilBERT AutoTokenizer from the Transformers library. Based on the Hugging Face DistilBERT guide [5], sequence length overflows (inputs exceeding DistilBERT's max feature length) were handled by returning overflowing_tokens and offset_mapping in the Hugging Face Tokenizer call. By mapping 1) tokenized features to their original example and 2) tokens to their original character positions in the example,

the models could be trained on long sequences without any truncation or loss of data. Overflows occurred occasionally in both the SQuAD and BioASQ datasets.

The training loop was implemented using the Hugging Face Trainer framework. After fine tuning, evaluation was conducted on single-domain test splits taken from the SQuAD or BioASQ datasets. Across all experiments, our primary metric for performance was the percent exact match (%EM) with the ground truth answers in each dataset.

#### 2.2.2 Mixed-Domain Fine Tuning

Current convention is to pre-train models on a general dataset before fine tuning for a domain-specific application, necessitating multiple models if the application domains differ significantly. To our knowledge, there has not been a systematic investigation into concurrently fine tuning a DistilBERT model on two vastly different datasets.

SQuAD and BioASQ were selected because they differ significantly in both semantics and structure. It was hypothesized that the pre-trained DistilBERT model checkpoint would be more compatible with SQuAD's wikipedia-derived context examples compared to the highly technical research excerpts in BioASQ. An open question was whether a single model could be fine tuned to perform well on both datasets and how the mixed-domain dataset should be constructed.

To get a baseline, we began by assessing model performance after fine tuning on the SQuAD and BioASQ datasets separately. Surprisingly, the models showed stronger in-domain and out-of-domain performance on BioASQ (see results).

After establishing this baseline, we began experimenting with dual-domain datasets. A variety of casting and mapping functions were used to align the feature definitions of both datasets to facilitate mixing. We created 6 datasets with different SQuAD:BioASQ ratios by concatenating randomly sampled chunks of examples from each dataset. We considered the possibility that the order of the mixed datasets (SQuAD then BioASQ) might also have an effect on the tuned models and generated a further 6 datasets which underwent shuffling to randomize the example order. In the preparation of these mixed datasets, care was taken to only sample from the 'training' split so that the models would be naive to the validation and test splits. We fine tuned the V-DB and ExBERT architectures on all 12 datasets, yielding 24 different models for comparison. We saw no evidence of interference between the two different domains on model performance regardless of whether the mixed dataset was shuffled (see results). This was the case for both V-DB and ExBERT, which performed similarly.

### 2.3. Tuning ExBERT

After the architecture revisions to modify V-DB with an ST-MOE layer, it was anticipated that significant hyperparameter tuning would be needed to extract better performance from the ExBERT architecture. We also noted while fine tuning on the mixed datasets that ExBERT consistently overfit the data by epoch 2. We subsequently performed a narrowing hyperparameter search to optimize ExBERT performance on a single mixed dataset (see results). Ultimately, the number of experts, number of transformer blocks, learning rate, and the inclusion of extra drop out layers were tuned to attain a group of ExBERT models which were more performant across all dataset mixture ratios.

### 2.4. Problems Encountered

Several challenges were addressed throughout this project. Instead of exclusively tuning off-the-shelf models from the Transformers library, a key aim of the project was to implement a novel architecture by augmenting DistilBERT with a switch transformer mixture of experts layer. This presented challenges with defining an appropriate loss function to facilitate stable learning of the experts. Integrating the ST-MOE auxiliary loss facilitated stable training of ExBERT. Mixing datasets was also non-trivial. In particular, the BioASQ dataset required extensive modification for compatibility with BERT-based models. This was accomplished using the casting and mapping tools within the Hugging Face Datasets framework. Fine tuning ExBERT on relatively small datasets presented a hyperparameter search challenge. Given the number of training examples relative to the complexity of the model, overfitting was a persistent problem. Modifying the number of transformer blocks and the learning rate went far to counteract this problem.

### 2.5. Code Sources

#### 2.5.1 Code Repositories Used

The DistilBERT model was a pre-trained checkpoint imported from the Hugging Face Transformers library (DistilBERT-base-uncased). The ST-MOE layer was a package implemented in pytorch imported from [6] based on the original paper by Zoph et al. [4]. Initial fine tuning of the vanilla DistilBERT model was based on a guide published by the Hugging Face team [5, 7]. All datasets were imported using the Hugging Face Load framework. Losses and other statistics were collected using the Wandb library imported into the notebook, and the graphics and visuals were created in a Wandb project and Microsoft Excel.

#### 2.5.2 Modifications

Extensive data preprocessing was conducted to combine the raw datasets in different configurations for the dual-

domain fine tuning experiments. This was accomplished using tools from the Hugging Face dataset framework. The ExBERT model was implemented by importing both the DistilBERT model checkpoint and the ST-MOE layer, and integrating them in a custom pytorch model. Training loops were written to conduct hyperparameter optimization, and evaluation loops were implemented to compare performance between many differently tuned models.

## 3. Results and Discussion

### 3.1. Establishing a Baseline: Vanilla DistilBERT and ExBERT

We started by evaluating the performance of both models after fine tuning on the SQuAD and BioASQ sets individually. All fine tuned models were tested on both datasets to gain insight on out-of-domain performance. SQuAD-tuned models performed surprisingly well on BioASQ (∼40% EM) whereas BioASQ-tuned models had next to no ability to output correct answers for SQuAD (∼5% EM).

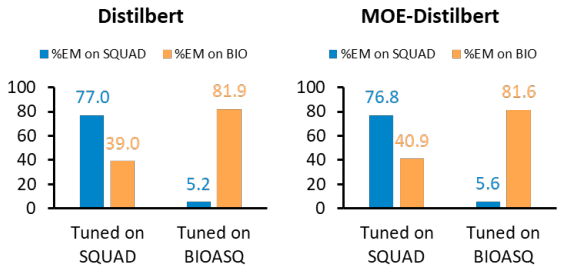


Figure 2. Prediction performance of V-DB and *non-optimized* ExBERT models, fine tuned on SQuAD database and BioASQ databases. Performance was measured based on percent of exact matches (%EM) on final validation on SQuAD (blue) and BioASQ (orange) datasets.

The implication is that both models were able to learn feature representations from SQuAD that were relevant to parsing BioASQ text. This may indicate that the sentence structures typical of biomedical text are less varied than the Wikipedia excerpts used in SQuAD.

In general, there was no significant difference between the performance of V-DB and ExBERT on either dataset. This result is consistent with the hypothesis that routing to different experts would only yield better performance on a more heterogeneous mixture of datasets across multiple domains.

### 3.2. Model Performance on Mixed Domain Datasets - Shuffled vs. Serial Fine Tuning

Next, we evaluated the effect of fine tuning models on mixed datasets. Examples from the SQuAD and BioASQ datasets were combined via concatenation into 6 different mixtures of training data. The ratio of SQuAD:BioASQ examples was varied from 0% SQuAD and 100% BioASQ to 100% SQuAD and 0% BioASQ in 20% increments. Across all datasets, size of the fine tuning dataset was held constant and equal to the max length of the original BioASQ set ( 4K examples). In these concatenated mixed datasets, the SQuAD examples always preceded the BioASQ examples.

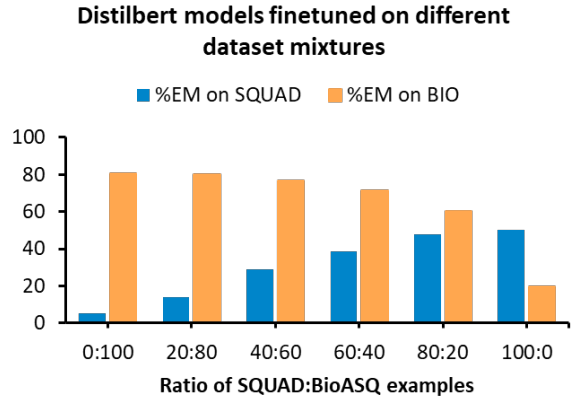


Figure 3. Bar graphs of V-DB model fine tuned on varying ratios of SQuAD-to-BioASQ databases (SQuAD:BioASQ). Final validation was performed on both SQuAD (blue) and BioASQ (orange).

Separate models were fine tuned on each of the 6 different mixed datasets. After fine tuning, the models were evaluated on unmixed test sets taken from the original SQuAD and BioASQ datasets and the % exact match for both datasets was visualized (Fig. 3).

Generally, the performance in a domain (SQuAD or BioASQ) improved the more that domain was represented in the dataset used for fine tuning. Performance on BioASQ degraded gradually, with the 60:40 model (only trained on ∼1600 BioASQ examples) attaining 90% the performance of the 0:100 model.

The heterogeneity of the SQuAD set continued to pose a challenge to models fine tuned on the mixed datasets. SQuAD performance gains slowed between the 80:20 and 100:0 models, and topped out at 52%, dramatically lower than the 77% attained after fine tuning on the full SQuAD set. This gap can be attributed to the number of training examples available to the model. With the mixed datasets capped at the max length of BioASQ, the 100:0 model could only be trained on 4000 SQuAD examples, about 5% the size of the original dataset.

Based on the strong out-of-domain performance of models exclusively trained on SQuAD (see Fig. 2), we wondered whether mixing in SQuAD examples during fine tuning might further boost model performance on BioASQ Q&A to a new high. This turned out not to be the case;

the model which performed best on BioASQ was the one which was fine tuned exclusively on BioASQ (Fig. 3).

The next point of interest was whether stacking the BioASQ examples on top of the SQuAD examples via concatenation was an optimal approach to generating mixed-domain datasets. The hypothesis was that fine tuning on sequential chunks of 2 different datasets might result in a loss of the model's learned weights for the initial dataset (in this case, SQuAD).

To test this, we regenerated dual domain datasets across all 6 mixture ratios but shuffled each dataset prior to initializing the trainer. The impact of shuffling versus simple concatenation was found to be minimal, indicating that the model was learning features for both dataset domains even when trained on the domains sequentially. Shuffling the data did provide a slight ($< 1\%$) improvement in the 20:80 and 40:60 models. Therefore, we elected to use the shuffled dataset mixtures throughout the remaining experiments.

### 3.3. Putting it Together: ExBERT Dual-Domain Performance

With a better understanding of the impact of concurrently fine tuning a model across two domains, the next step was to evaluate whether the ExBERT architecture could outperform V-DB. We proceeded to fine tune different ExBERT models on the same 6 mixed, shuffled datasets using the default initialization parameters for ExBERT's DistilBERT and MOE layers. Under these parameters, ExBERT performed similarly to V-DB, with slight improvements ($\sim$2%) seen in the 40:60 and 80:20 ExBERT models.

A strong tendency for ExBERT to overfit was noted when conducting these initial evaluations. Though the ST-MoE layer is sparse by virtue of its expert routing mechanism, it still contributes more parameters and complexity to the model compared to V-DB. We reasoned that especially given the small, 4K example length of our fine tuning datasets, additional model performance could be extracted by tuning hyperparameters to address this overfitting.

For the subsequent hyperparameter tuning experiments, we selected the 80:20 data mixture. Models trained with this domain data ratio exhibited a steep decline in BioASQ performance but 10% improvement in SQuAD performance. We reasoned that tuning ExBERT to better generalize on this dataset would yield a strong model across all of the mixture ratios.

### 3.4. Optimizing ExBERT Performance on Mixed Datasets

Given that high model complexity relative to the size of the training dataset can lead to overfitting, we sought to reduce the number of parameters in the ExBERT model. We experimented with different numbers of experts, but saw minimal difference in learning curves or model performance. This could be attributed to the sparse nature of the ST-MOE layer, where only a subset of experts (dictated by the gating_top_n parameter) contribute on each forward pass and have their weights updated on each backward pass.

Far more impactful was adjusting the number of transformer blocks in the DistilBERT layers of ExBERT. As expected, reducing model complexity by removing transformer blocks increased training and validation loss. However, in the 3 and 4 layer models, training and validation loss tracked more closely, and these models did not exhibit the early onset validation loss increase observed around the third epoch in the 5 and 6 layer models. All further tuning was performed on the 4 layer ExBERT model, which exhibited consistent learning behavior and maintained a relatively low validation loss.

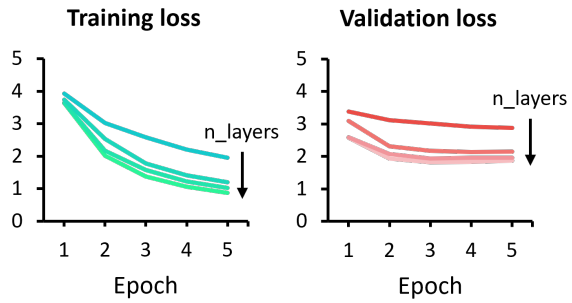


Figure 4. Training (green) and validation (pink) losses over epochs as the number of layers in the MoE model are varied.

Regardless of the number of transformer blocks used, best performance across all models was consistently achieved within 5 epochs. Using the 4 layer model, we tuned the learning rate to minimize validation loss within this training window. We assessed learning rates of 0.5X, 2X, and 5X the default V-DB rate of 2e-5. With learning rates of 2X and 5X, severe overfitting was again observed. However, the 5X learning rate achieved the lowest validation loss seen in any experiment so far. Given the model's persistent tendency to overfit, we opted for an early-stopping strategy and moved forward with the 4 layer ExBERT trained for 2 epochs at 5X default learning rate.

A final set of experiments was conducted to determine whether adding additional dropout layers could extend the stable training window of the 4 layer model by a few more epochs, even with the aggressive 5X learning rate. The model architecture was modified with dropout layers either before the ST-MOE layer, after the ST-MOE layer, or both before and after. We initialized these layers using the default V-DB dropout rate (0.1).

The additional dropout layers did not curb the overfitting behavior. A more aggressive dropout of 0.3 was also tested, but this model continued to overfit and exhibited significantly higher validation loss at all epochs. A single dropout

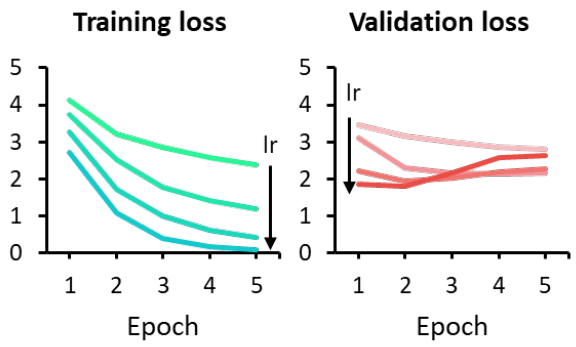


Figure 5. Training (green) and validation (pink) losses over epochs as the learning rate of the MoE model are varied from 2e-5 to 1.1e-4.

layer before the ST-MOE layer achieved a better (<0.5%) validation loss at epoch 2 compared to no dropout. This frontal dropout layer was adopted for the final ExBERT architecture used for comparison with V-DB.

The tuned ExBERT model beat V-DB across all dataset mixture ratios, including the homogenous 0:100 and 100:0 datasets (Fig. 8). Much larger (10-20%) improvements over V-DB were seen when the models were fine tuned on mixed datasets. This suggests that ExBERT's performance edge stems from fast learning on a small number of examples (e.g. hitting 50% EM on only 1600 SQuAD examples when fine tuned on 40:60).

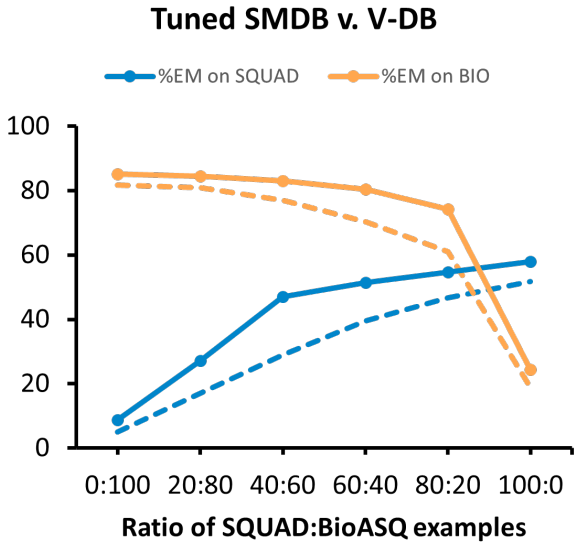


Figure 6. *Tuned* prediction performance (percent exact match) on SQuAD (blue) AND BioASQ (orange) Q&A tasks over varying dataset mixtures using ExBERT (solid lines) vs. DistilBERT (dotted lines). Dataset mixtures range from 0:100 (no SQuAD, 100% BIO) to 100:0 (100% SQuAD, 0% BIO).

Table 1. Comparison of V-DB and ExBERT Outputs.

| | Question | Ground Truth | V-DB | ExBERT |
|---|---|---|---|---|
| SQuAD | Who had the best record in the NFC? | Carolina Panthers | New Orleans Saints and the 2011 Green Bay Packers | Carolina Panthers |
| | What injury did the Carolina Panthers lose Kelvin Benjamin to during their preseason? | torn ACL | a torn ACL | torn ACL |
| | How many yards did Newton throw for in 2015? | 3837 | 3,837 yards and rushing for 636 | 3837 |
| | Who is the head coach of the Broncos? | Gary Kubiak | John Fox | Gary Kubiak |
| BioASQ | ROSIER scale is used for which disorder? | Stroke | Stroke | Stroke |
| | Which enzyme is inhibited by a drug fostamatinib? | spleen tyrosine kinase | SYK | spleen tyrosine kinase |
| | Which is the molecular target of the immunosuppressant drug Rapamycin? | mTOR | ribosomal protein S6 kinase | mTOR |
| | What is the typical rash associated with gluten ? | Dermatitis herpetiformis | dermatitis herpetiformis that uses HLA-DQ8 transgenic NOD mice. Dermatitis herpetiformis | Dermatitis herpetiformis |

From a qualitative assessment of the predictions, the V-DB model generally created longer answers to the questions irrespective of dataset. One interesting note is that these long-winded predictions often contained the ground truth answer within the response, as seen in the last example of Table 1. Ultimately, the ExBERT model was more consistent in matching the ground truth answers.

## 4. Other Considerations

ST-MOE, works on the concept of gating to send information to the top "rated" experts, comprised our switch-transformer gated mixture-of-experts layer, which was added to (or replaced) DB transformer layers. To definitively address the issue of overfitting in ExBERT, we scaled the complexity (6, 5, and 4 transformer blocks) of ExBERT. Indeed, reducing the number of DistilBERT layers proved to significantly improve our model performance–to the extent that ExBERT was showing superior performance at epoch 2, compared to epoch 3 with V-DB.

Incredibly, ExBERT achieved its minimum losses 33% earlier than DistilBERT, with an increase of 8% EM on single-domain SQuAD and 13.1% EM on BioASQ when trained on an 80:20 (SQuAD to BioASQ) dataset.

**Impact and Future work:** Multi-domain training for cross-domain tasks remains an interesting and highly-relevant area of study: Smaller, more efficient models such as DistilBERT demonstrate only slightly degraded performance compared to full models such as BERT. By incorporating a sparse block (ST-MOE), we demonstrate that distilled models can efficiently train on small multidomain datasets for superior cross-domain performance. We believe that this is due to an improvement in task adaptation, future work should focus on exploring this effect, and counteracting the significant overfitting that is demonstrated by *both* DistilBERT and ST-MOE through the exploration of parameters such as balancing coefficient for ST-MOE and dropout rates *within* its expert NNs.

## 5. Work Division

See Page 8.

## 6. References

List and number all bibliographical references in 9-point Times, single-spaced, at the end of your paper. When referenced in the text, enclose the citation number in square brackets, for example [**?** ]. Where appropriate, include the name(s) of editors of referenced books.

## References

[1] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. https://arxiv.org/abs/1810.04805v2, 2018. Arxiv preprint, arXiv:1810.04805. 1

[2] Yanping Fu and Yun Liu. Contrastive transformer based domain adaptation for multi-source cross-domain sentiment classification. https://www.sciencedirect.com/science/article/abs/pii/S0950705122002970, 2022. Knowledge-Based Systems, 245:108649. 1

[3] Antonis Maronikolakis and Hinrich Schütze. Multidomain pretrained language models for green nlp. *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 1–8, 2021. 1

[4] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models. https://arxiv.org/abs/2202.08906, 2022. Arxiv preprint, arXiv:2202.08906. 1, 3

[5] Hugging Face. Distilbert. https://huggingface.co/docs/transformers/model_doc/distilbert#transformers.DistilBertForQuestionAnswering, 2023. Hugging Face. 2, 3

[6] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, , and William Fedus. St-moe: Designing stable and transferable sparse expert models. https://github.com/lucidrains/st-moe-pytorch, 2022. Github. 3

[7] Hugging Face. Question answering on squad. https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/question_answering.ipynb, 2023. Hugging Face. 3

Table 2. Contributions of team members.

| Student Name | Contributed Aspects | Details |
|---|---|---|
| Alexander C. Fan | Data Mixtures — Optimization | Generated scheme for testing data mixtures and mixture effects, as well as architecture tuning. |
| James C. Fan | Evaluation — Optimization | Evaluated of results, and provided targeted feedback for optimization. |
| James K. Song | Architecture — Optimization | Implemented ST-MOE and integration into ExBERT. |