Mixture of Hierarchical Unified Neural Domain Experts (MHUNDE): Transforming Scouting in Major League Baseball

Stanford CS224N Custom Project

Aman Malhotra
Department of Computer Science
Stanford University
amanm27@stanford.edu

Eish Maheshwari
Department of Computer Science
Stanford University
eishm@stanford.edu

Amol Singh
Department of Computer Science
Stanford University
asingh11@stanford.edu

Abstract

Although many baseball prospects dream of having a career in the MLB, the majority of them do not make it to the major leagues. It is a tough but essential task to effectively identify talent in the prospect pool. We aim to implement an effective pre-trained and fine-tuned deep learning model to predict whether a baseball prospect will have a major league career given scouting reports written on the player. We address small sample size and class imbalance in the scouting report data through multiple data augmentation strategies, experiment with a novel hierarchical pre-training approach on a variety of domains, ranging from Wikipedia text to general sports articles to unlabeled scouting reports, and leverage a mixture of experts by integrating insights from various pre-trained models. Our best model configuration uses a shuffled sentences data augmentation technique and a mixture of domain experts to improve accuracy by 36.9% and F1 score by 66.1% compared to the best previously published benchmarks on this task.

1 Key Information

Our TA mentor is Gaurab Banerjee. No external collaborators/mentors and no project sharing.

2 Introduction

In baseball, among other sports, there has been a clash between traditional coaches, who rely on their experience and knowledge of the sport, and more open-minded coaches, who embrace the analytics revolution sweeping across sports since the Moneyball era in the early 2000s. Teams invest a sizable amount of resources into good scouting departments who watch players in live games and take written notes on their skills. These notes directly impact deciding whether a young prospect will receive the chance to pursue a career in the major leagues. However, given the massive population of baseball prospects, a tiny fraction of which make it to the major leagues, do such written scouting reports have predictive power for major league success? By training a deep learning model on scouting reports, the recruitment process can be streamlined by learning the nuances of scouting language.

This problem is made especially challenging since language in the scouting reports is very niche and domain-specific to baseball with significant vocabulary and content variation from report to report.

Additionally, there are a limited number of players with scouting reports available (less than 10,000 examples) and most of these players are not drafted, resulting in a small dataset with a heavy class imbalance (around 75-25 negative-positive split). Existing modeling approaches use a handful of standard baselines (e.g. Bag of Embeddings, CNN, LSTM) but struggle to perform well. The goal of our project is to improve existing modeling approaches with a three-pronged approach.

1. Fixing class imbalance and small sample size through data augmentation
2. Using a novel hierarchical pre-training approach on domains of varying specificity to expose the model to general sports jargon and specific domain language
3. Using a mixture of experts approach to combine knowledge from multiple independent pre-trained models.

Our experiments yield accuracy and F1 scores that outperform all published baselines on this task.

3 Related Work

3.1 Data Augmentation

Data augmentation is a powerful strategy for creating synthetic samples from existing training data. Often when working with small datasets or a large class imbalance, this helps reduce over-fitting and class biases. Wei et al. introduced easy data augmentation (EDA), a set of simple operations (synonym replacement, random insertion, random swap, and random deletion) to modify existing text [1]. EDA was found to improve performance on both convolutional and recurrent neural networks; however, these gains were minimal, limited to only a 0.8% increase in accuracy, and a more effective strategy is needed.

Zhang et al. proposed mixup, which constructs synthetic training examples as linear combinations of existing features and labels [2]. The potential of mixup is realized in creating hybrid-labeled training examples since positive and negative samples can be combined (with a corresponding mixed label). Thus, mixup can expand a dataset of $O(n)$ examples to $O(n^2)$ examples, in addition to considering linear associations between examples. Zhang et al. found mixup to improve generalization of image-base CNNs, especially on adversarial examples. However, there is limited work on mixup in language tasks since mixing text data is semantically complex. Sun et al. has suggested using a pre-trained transformer (ex: BERT-base) for tokenizing text in mixup applications [3].

3.2 Pre-training

We also reviewed prior work on domain adaptation through pre-training. First, Kitaev et al. [4] showed that pre-training improves performance on the final task dataset, even when the pre-training data is not a perfect fit for the target application. However, Gururangan et al. [5] argue that the pre-training set shouldn't be completely irrelevant, because blindly exposing the model to more data without considering the relevance of the domain can worsen performance on the final task. Furthermore, in addition to domain-adaptive pre-training (DAPT) on domains differing from that of the final task, Gururangan et al. also propose and show the efficacy of task-adaptive pre-training (TAPT) on a small subset of unlabeled examples from the final dataset. However, researchers only pre-trained on domains of similar topic specificity as the target task, which may not generalize the model as well.

3.3 Mixture of Experts

The mixture of experts (MOE) approach was inspired by and developed with the divide-and-conquer paradigm. For MOE, the problem domain for a task are divided up, where different models are trained to become experts in different aspects of the feature space [6]. Ensemble learning, a similar concept, combines the results from multiple models trained on the same dataset.

Mixture of experts was introduced over three decades ago, with much of the initial research performed by Jacobs et al. initially introducing the concept of breaking down problems into individual tasks and then later introducing the softmax-based gating function [7] [8]. Following this paper, significant research has been performed in the development of different expert architectures (e.g. mixture of

Gaussian processes [9]) and gating functions (e.g. introduction of sparsity into gating functions [10]). However, there is limited work on how to divide the problem space, specifically, how to specialize experts to develop a strong collective understanding of the target domain.

3.4 Prediction Task

Danovitch introduced the dataset of labeled scouting reports and used them on this same task. He developed several models, all of which had limited performance. The best model (TextCNN) had an accuracy of 0.6902 and F1 of 0.5642 [11].

Model	Accuracy	F1
Bag-Of-Embeddings	64.65%	53.78%
TextCNN	69.02%	56.42%
LSTM+SelfAttn	68.64%	54.65%
BCN	73.52%	43.33%
HAN	66.00%	54.07%

Figure 1: Previously best published model performances from Danovitch

4 Approach

4.1 Data Augmentation

4.1.1 Simple Augmentation

To address overfitting from a small sample size and class imbalances, we tried several simple data augmentation strategies on labeled reports (implementations based off textaugment library [12]).

1. Easy Data Augmentation (EDA) - combination of synonym replacement, random insertion, random swap, and random deletion of words. We developed a hybrid strategy where these techniques were randomly applied to each sentence of a scouting report.
2. Shuffled Sentences - randomly swaps the order of sentences in a given scouting report.
3. Contextual Embeddings - use a bidirectional language model (ex: BERT) to stochastially replace words, while retaining semantic meaning [13].

Because the words of each scouting report are semantically dependent on other words in the same sentence, augmentation was performed on the sentence level. Because there were significantly fewer positive than negative training examples, only positive examples were augmented until there was no class imbalance.

4.1.2 Mixup Transformer

We also developed a modified mixup-transformer architecture for use with pre-trained models such as BERT. This is motivated by the potential of mixup to dramatically increase total training set size, as well as to learn patterns in combined data. Synthetic training examples (x', y') were constructed as

$$x' = \lambda x_i + (1 - \lambda) x_j$$
$$y' = \lambda y_i + (1 - \lambda) y_j$$

where (x_i, y_i) and (x_j, y_j) are randomly drawn samples from the training set (after tokenization through BERT) and $\lambda \in [0, 1]$ is the mixup strength. The mixed-up example is fed through a BERT-base and then a 2-layer feed-forward layer (composition of linear and ReLU activation layers with dropout). Since mixed labels range between $\hat{y} \in [0, 1]$, the model is trained as a regression task, but evaluated as binary classification.

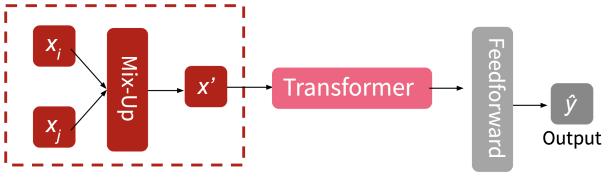

Figure 2: Mixup data augmentation framework. x_i and x_j are tokenized embeddings of 2 scouting reports. Inputs are linearly interpolated before fed through a pre-trained transformer (ex: BERT) and feedforward layer with dropout. BERT weights are unfrozen during fine-tuning.

4.2 Hierarchical Domain Adaptation

Next, we explored domain adaptation through hierarchical pre-training. This is motivated by niche baseball vocabulary that varies greatly from report to report in our small dataset, so solely training on scouting reports might not allow the model to generalize well. Rather than using domains of similar specificity like previous work [5], we developed a novel hierarchical pre-training framework that increases in domain specificity to the final task through multiple pre-training stages. Our domains were Wikipedia text, general sports articles, and unlabeled scouting data. The intuition is that each hierarchy level adds more domain specific knowledge in addition to more general knowledge to reduce over-fitting. Models consist of the same general structure, with various pre-training stages removed. This maintains the hierarchy of pre-training on general domains first, then more task-relevant texts. After pre-training, we fine-tune the model on labeled scouting reports. Our baseline is a model without data augmentation or pre-training. We implemented model pre-training using the Hugging Face Transformers library (tokenization w/ BertTokenizer, pre-training with AutoModelForMaskedLM) based on an example [14] from Hugging Face.

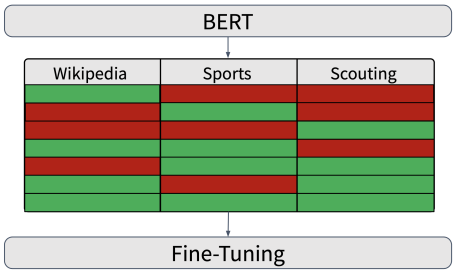

Figure 3: Hierarchical pre-training framework. Each row is a distinct model either pretrained on a dataset (green) or not (red). A BERT transformer is pre-trained (using a masked language modeling objective) before being fine-tuned on labeled scouting reports for binary classification.

4.3 Mixture of Hierarchical Experts

The intuition behind the mixture of experts involves creating a think tank of domain-specific experts that each contribute to the final output. Scouting reports are a good fit for this approach as they consist of an extreme variety of content. There is information ranging from baseball-specific jargon to the criminal history of candidates. For example, one scouting report mentions how a player is a "Level 1 sex offender." Having a model pre-trained with Wikipedia text will better discern the context behind a less baseball-specific scouting report. Other reports mention sport specific injuries (ex: surgeries) or performance of the athlete in other sports, a task handled well by a model pre-trained on sports articles. Finally, all scouting reports contain a level of technical analysis of the player using baseball specific terminology and acronyms like Double-A. Using this combination of models, we hypothesize that our models can learn different aspects of the feature space effectively to accurately classify players. For the gating function, our approach initially utilized a 3-layer multilayer perceptron (MLP) with 16 neurons in the hidden layer. This gating function evolved into one inspired by the Jacobs et al. [8] using a learned set of weights on the original input and an applied softmax to generate the weighting on each of the output models in the final computation.

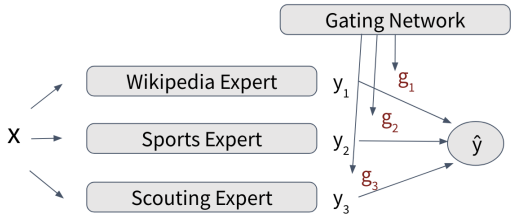

Figure 4: Mixture of hierarchical experts framework. Predictions from individual domain experts (each well-versed in their respective domains through targeted pre-training) are passed through a learned gating network to produce a final ensembled prediction.

5 Experiments

5.1 Data

For our main binary classification task, we use a dataset [15] published by Jacob Danovitch of Carleton University, which contains 7778 *labeled* written scouting reports with 2114 positive and 5664 negative examples. The labeled data was divided into a train, dev, and test set with an 80-10-10 split. Models were fine-tuned and evaluated on this data.

For task-adaptive pre-training (TAPT), we use 1397 *unlabeled* scouting reports from Danovitch's dataset. For one pre-training task, we use the wikitext dataset [16] provided by Hugging Face, which includes over 1.8 million text samples from Wikipedia articles. For more domain-adaptive pre-training (DAPT), we use the "Sports articles for objectivity analysis" dataset [17] from the UCI Machine Learning library, which contains text from 1000 general sports articles.

5.2 Evaluation method

The original BERT paper used GLUE as the main evaluation metric, which is the overall macro-average of the unweighted averages of the task specific metrics, often accuracy and F1 score, for each task [18]. Therefore, we use the F1 score and accuracy as the main two metrics for the test group. In addition, the recall is key towards our project domain as we want to be able to capture all the great players that should make the MLB, even at the cost of taking some false positives.

5.3 Experimental details

5.3.1 Baseline model selection

We evaluated the performance of three baseline parameter initializations (all from Hugging Face): `bert-base-uncased`, `microsoft/SportsBERT`, and `xlm-roberta-base`. These were chosen since they are designed to be fine-tuned for sentence-level tasks like sequence classification. For each model, we evaluated accuracy, precision, and recall on our task after fine-tuning on labeled scouting reports (without augmentation). After performing a hyperparameter grid search, we used a per-device batch size of 16, 3 epochs, AdamW optimizer, and a learning rate of $2e-5$. Each epoch took 5 minutes. The best baseline was used in all future experiments.

5.3.2 Data augmentation

Simple augmentation methods (EDA, shuffled sentences, contextual embeddings) were applied to positive training examples to remove class imbalances. The augmented training data was used to fine-tune the best performing baseline model with the settings as above. Performance of the mixup transformer was evaluated by performing hyperparameter searches for the best mixup strength (λ) and dropout probability in the feedforward layer (p). Mixup was used to create 2500 positive, 2500 negative, and 5000 hybrid-label training examples. Each epoch took about 6 minutes.

5.3.3 Hierarchical pre-training

We evaluated performance of the 7 pre-trained models (along with a baseline without pre-training). Pre-training models were trained first, then weights were uploaded to Hugging Face. These weights were then imported to fine-tune classification models. Classification models used a BERT-like transformer with a linear layer as a sequence classification head on top of the pooled outputs. Additionally, these models were trained on the augmented dataset produced by the best data augmentation technique. Both pre-training and classification took 10-15 minutes to train.

5.3.4 Mixture of experts

The 7 pre-trained models were evaluated in different groupings of 3 experts to determine the best division of the feature space defined by pre-training. The final models and gating function were fine-tuned on labeled scouting reports. The MLP for the gating function was trained for 50 epochs and with a learning rate of 0.02 with a batch size of 50.

5.4 Results

5.4.1 Baseline model selection

`bert-base-uncased` and `SportsBERT` had very similar accuracy (0.767 and 0.778 respectively), which already performed comparably to the best published benchmarks on the task [11]. However, the former had balanced precision and recall while the latter had higher precision and lower recall. We opted to prioritize recall over precision, so true talent would not be passed up on. Thus, we chose `bert-base-uncased` as our baseline parameter initialization for all future models. `xlm-roberta-base` had similar accuracy, but did not predict any positive examples, resulting in zero precision and recall. This suggests that training on this larger baseline model worsened the model's ability to learn nuances in the scouting report domain language to classify any positive examples, which is understandable given the strong class imbalance.

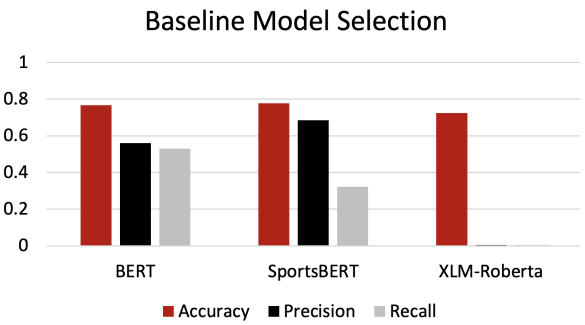

Figure 5: Performance of baseline models without pre-training or fine-tuning

5.4.2 Data augmentation

Simple Augmentation: Using EDA had a 10.4% increase in accuracy and 20.0% increase in F1 compared to no data augmentation on a fine-tuned BERT model (baseline). This suggests that adding noise in the data via EDA reduced overfitting. The best performing augmentation strategy was shuffling sentences which increased accuracy of the baseline by 13.8% and F1 by 51.4%. The drastic jump in performance by shuffling of sentences suggests that the order of sentences does not add much value to the binary classification task. Contextual embeddings had almost no improvement, likely because scouting reports contain highly specific jargon for which there are few semantically replaceable contextual embeddings.

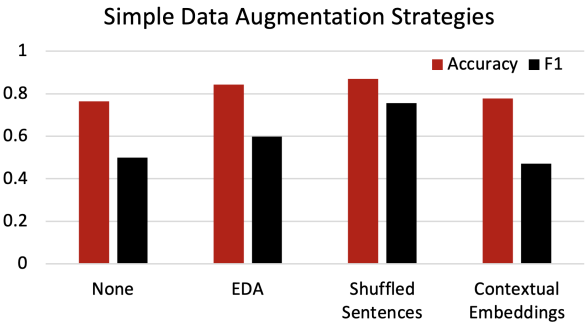

Figure 6: Performance of various simple data augmentation strategies

Mixup: Ideal performance was found with a mixup strength of $\lambda = 0.4$ and a dropout rate of $p = 0.2$. We noticed when training that the loss would first decrease then began to increase again. To combat this, we decreased the learning rate to $2e - 6$ and increased the number of epochs to 30 which fixed

the optimization issue. The best mixup model gave a 0.58% *reduction* in accuracy and only 4.14% improvement in F1 from the non-augmented baseline. Overall, there is minimal improvement on performance, suggesting that linearly combining scouting reports does not reveal new associations that increase predictive power. Interestingly, mixup models tended to be recall-biased (recall > 0.9 for all models), suggesting mixup improves identification of positive labels.

Mixup Strength		
λ	Accuracy	F1
0.0	**0.7634**	0.4986
0.1	0.7525	0.4887
0.2	0.7512	0.4941
0.3	0.7172	0.4930
0.4	0.7589	**0.5006**
0.5	0.7294	0.4847

Table 1: Hyperparameter tuning with mixup strength λ and dropout rate $p = 0.2$

Dropout Rate		
p	Accuracy	F1
0.0	0.6594	0.4566
0.1	0.7491	**0.5291**
0.2	**0.7589**	0.5007
0.3	0.7512	0.4921

Table 2: Hyperparameter tuning of mixup transformer with mixup strength $\lambda = 0.4$ and dropout rate p

5.4.3 Hierarchical pre-training

All pre-trained models were fine-tuned with shuffled sentences data augmentation since it was the best performing strategy. We evaluated different hierarchies of pre-training as an ablation study. Task-adaptive pre-training (TAPT) performed better than domain-adaptive pre-training (DAPT), as pre-training on unlabeled scouting data had the highest accuracy (0.9357) and F1 score (0.8747). This suggests scouting reports contain highly specialized jargon and patterns which are not captured in general articles outside that domain.

Pre-Training	Accuracy	F1
None	*0.8689*	*0.7548*
Wiki	0.9107	0.8311
Sports	0.9067	0.8345
Scouting	**0.9357**	**0.8747**
Wiki + Sports	0.9229	0.8485
Wiki + Scouting	0.9081	0.8065
Sports + Scouting	0.9068	0.8153
Wiki+Sports+ Scout	0.9331	0.8706

Figure 7: Hierarchical pretraining on Wikipedia, sports articles, and unlabeled scouting data

5.4.4 Mixture of experts

The highest accuracy and recall were found in experiment 1, where each expert was pretrained on a different domain (one on just Wikipedia, one on just sports articles, one on just unlabeled scouting). Ensembling with MOE improved accuracy by 1.0% and F1 by 7.1% compared to the TAPT model above. Overall, this is a 23.8% improvement in accuracy and 87.9% improvement in F1 from the BERT baseline. Ensembling with multiple experts pretrained on Wikipedia (experiments 2 and 4) had noticeably lower performance, suggesting there was not a useful transfer of semantic knowledge.

Mixture of Experts				
Model	Accuracy	F1	Precision	Recall
1)Wiki-SP-SC	**0.9449**	0.9372	**0.9282**	**0.9495**
2)Wiki-Wiki+SP-Wiki+SP+SC	0.9079	0.8505	0.7835	0.939
3)SP-SP+SC-SC	0.9309	**0.9382**	0.9321	0.9484
4)Wiki-SP-Wiki+SP	0.9101	0.9250	0.9279	0.9246

Table 3: Evaluation of mixture of experts and some key ablation studies conducted through permuting pretrained datasets. A notation of SC = Scouting and SP = Sports has been included for brevity. Dashes (–) separate different experts and the plus (+) indicates all the datasets the model was hierarchically pre-trained on.

6 Analysis

6.1 Final Model: Error Analysis

Our best performing model (accuracy: 0.945, F1: 0.937) used shuffled sentences for augmentation and a hierarchical mixture of experts where each expert was only pre-trained on one level of specificity

(Wiki-SP-SC). Looking at the confusion matrix, we see that using MOE slightly biases recall over precision, increasing the number of false positives. This optimization reduces the chance of missing a successful player. We then examined the importance of each expert by looking at weights in the feedforward layer of the gating function, noting that all models contribute to the final output, but the model pretrained only on unlabeled scouting reports (TAPT) had a slightly greater importance (35.4%) as it has the most in-domain knowledge.

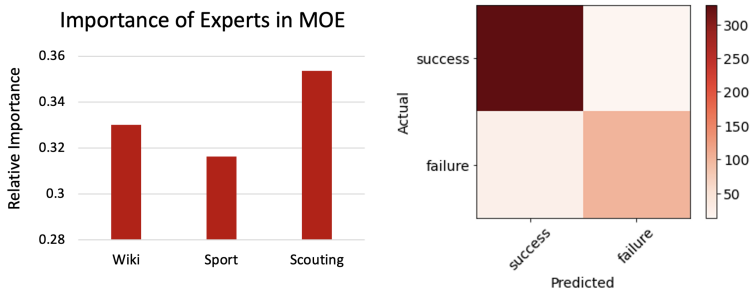

Figure 8: Importance of experts according to gating function weights (left). Performance of mixture of experts strategy (right).

6.2 Qualitative Analysis

We will examine incorrect predictions made by the best-performing MOE model. As our model was recall-biased, false positives were much more common.

Scouting Report Excerpt 1:

> *"Another aggressive Latin GPE signing. Hes a long way from GPE, but shows signs of hitting for power with a decent approach at the plate and excellent bat speed. It's unclear just yet if he'll be able to stay at third, with his below-average speed limiting his range. If he hits like it appears he can, he'll be just fine if he has to move to first base in the future."*

Label: 1
Prediction: 0

This false negative showcases issues with interpreting a mixed-sentiment scouting report. The report highlights both positive and negative attributes of the player, which makes predicting a single class challenging. Furthermore, the model struggles with a sense of time. The player "shows signs of hitting for both power and average", suggesting good potential for the future.

Scouting Report Excerpt 2:

> *"PERSON originally caught PERSON's eye with his quick right-handed bat and his outstanding bat speed. He could hit 15 homers per season if he adds strength and tones down his aggressive approach. He lacks polish at the plate, which is understandable considering he's 20 and totaled just 391 plate appearances in his first three pro seasons."*

Label: 0
Prediction: 1

This false positive highlights the difficulty of learning nuanced baseball terminology. For example, while an "aggressive signing" (as referenced in the previous example) might be beneficial, an "aggressive approach" to hitting is flaw, namely that a player cannot control their swing. Therefore, without a deep contextual understanding of the task domain, the model gets confused.

Scouting Report Excerpt 3:

> *"PERSON was among the top GPE high school pitchers, and the GPE made him their fifth-round selection in June. He is polished for a high schooler and has the*

potential for three Major League-average pitches. PERSON's fastball sits in the upper-80s to low-90s. PERSON's advanced stuff and understanding of pitching help make up for his lack of projection. He could move faster than most high school pitchers and profiles as a middle-of-the-rotation starter."

Label: 0
Prediction (Single Pretrained Model): 1
Prediction (MOE): 0

This example was incorrectly predicted as positive by the single model pre-trained on all datasets (i.e. last row in Figure 3), but was correctly predicted as negative by the best MOE model of separate domain experts. What stands out is the presence of *very* specific baseball knowledge. Even though the report seems positive on the surface, "fifth-round selection" implies he wasn't high in the draft and "middle-of-the-rotation starter" implies scouts don't project him to be one of the best pitchers. Such niche baseball knowledge is likely not found in Wikipedia or general sports articles, but could be learned by a scouting report expert.

7 Conclusion

We saw that augmenting our dataset by shuffling sentences and using a mixture of hierarchical domain experts led to a 36.9% improvement in accuracy and 66.1% increase in F1 score from the previous best published model [11], with an overall accuracy of 0.9449 and F1 of 0.9372. Surprisingly, shuffling sentences provided the biggest performance boost compared to other data augmentation methods, suggesting that sentence order does not matter in scouting reports. Furthermore, we saw that mix-up was ineffective for augmentation, suggesting that linear combinations of scouting report features do not reveal new information that increase predictive power.

Most notably, we saw improved performance from our novel hierarchical pre-training framework based on increasing topic specificity. High scores on models pre-trained on unlabeled scouting report data, as well as a heavy weight placed on the scouting expert in our mixture model, highlight the effectiveness of task-adaptive pre-training (TAPT). This aligns with the fact that scouting reports contain highly specialized jargon and patterns which might not be captured in general articles; exposing the model to such niche information is beneficial to performance on the final task.

However, our model is limited in handling mixed-sentiment reports, especially in the context of very nuanced task knowledge. In the future, we would like to collect more data (in new domain hierarchies) and test transferability to different sports domains. Additionally, we can introduce sentence-level order-based noise to reduce overfitting. Lastly, we can improve the gating function for our mixture of experts framework beyond a simple fully-connected network.

References

[1] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Protago Labs Research, Tysons Corner, Virginia, USA*, 2019.

[2] Hongyi Zhang, Moustapha Cisse, Yann Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[3] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip Yu, and Lifang He. Mixup-transformer: Dynamic data augmentation for nlp tasks. In *Salesforce Research*, 2020.

[4] Nikita Kitaev, Steven Cao, and Dan Klein. Multilingual constituency parsing with self-attention and pre-training. In *Association for Computational Linguistics (ACL)*, 2019.

[5] Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Association for Computational Linguistics (ACL)*, 2020.

[6] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, May 2012.

[7] Robert A. Jacobs, Michael I. Jordan, and Andrew G. Barto. Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive Science*, 15(2):219–250, April 1991.

[8] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the em algorithm. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, volume 2, pages 1339–1344 vol.2, 1993.

[9] Volker Tresp. Mixtures of gaussian processes. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000.

[10] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017.

[11] Jacob Danovitch. Trouble with the curve: Predicting future mlb players using scouting reports, 2019.

[12] https://github.com/dsfsi/textaugment.

[13] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Preferred Networks, Inc.*, 2018.

[14] https://github.com/huggingface/notebooks/blob/master/examples/language_modeling.ipynb.

[15] https://github.com/jacobdanovitch/trouble-with-the-curve.

[16] https://huggingface.co/datasets/wikitext.

[17] https://archive.ics.uci.edu/ml/datasets/sports+articles+for+objectivity+analysis.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.