CS W182 Final Project (NLP)

Anirudhan Badrinath
3034761432
abadrinath@berkeley.edu

Eric J. Michaud
3032858077
ericjmichaud@berkeley.edu

Charlie Faramarzi
3033672113
charlie.faramarzi@berkeley.edu

1 Introduction

Yelp reviews are commonly used as an overall public assessment of a widely used institutions such as a restaurants and beauty salons. Similar to the prediction task of Amazon rating scores given text reviews [13], we create a natural language processing (NLP) model to accurately model the sentiment of a review into a quantitative measure of rating (in stars) of the institution.

We built a robust model for predicting, given a restaurant review in plain text, the corresponding rating (1-5 stars), for the review – a standard text classification task. We are given a data set of around 500,000 chosen Yelp reviews as the primary training set. Our model was trained and evaluated based on its classification accuracy as well as distance, in number of stars, between its prediction and the true rating of the review.

We aimed to make a classifier which is not only accurate on our given Yelp dataset, but robust to common perturbations of text-based data. This was achieved through novel data augmentation methods, such as cropping, merging, or scrambling reviews based on heuristics and cosine similarity. Our data augmentation strategy was informed by our analysis of the typical structure of a review, through which we determined that the beginning and conclusion of a review contain more information relevant to the classification task. Overall, we created a model that ranks in the top 20% for both released test set tasks in terms of accuracy with 63.2% and 53.6% accuracy on Challenges #8 and #5 respectively.

This challenge of making a robust text classifier has many practical applications in the design of recommendation systems and content moderation.

2 Related Work

Sequence modeling has been widely studied for the purposes of natural language processing (NLP). Traditional approaches have included the classic long-short term memory (LSTM) model [6]. Recently, a shallow neural network for language representations in vector space (known as word2vec) was developed [10]. However, these approaches have lacked context-based representation for words.

Transformers were introduced in 2017 as a novel sequence transduction model, relying on an encoder-decoder configuration with an attention mechanism [14]. Unlike many prior approaches, such as convolutional or recurrent methods as described above, the transformer doesn't use any recurrent approaches or convolutional layers. Currently, the most widely used pre-trained transformer-based models in natural language processing are BERT [3], BART [7], and GPT/GPT-2, developed by OpenAI. The transformer architecture is especially useful for pretraining on massive data-sets. The pretrained model can then be fine-tuned downstream to greatly increase accuracy on more specific prediction or classification tasks, even on text containing different writing styles (academic writing vs. forum posts) or domain-specific language[15].

Newer complex transformer-based models such as XLNet have been devised to further improve upon BERT [16]. It doesn't train via masking a proportion of the input words; instead, it uses permutation-based language modeling. As a result, XLNet is autoregressive as opposed to BERT, which presents many advantages. While it outperforms BERT in many NLP tasks, it is far more computationally expensive.

Pre-trained transformers models like BERT excel at capturing the context-dependent meaning of a word in a general case, but often fail to account for underrepresented linguistic features such as punctuation frequency that may be more insightful or prevalent in specific types of text [1].

For out-of-distribution generalization, ensembling techniques and data augmentation methods are used. Ensembling multiple models has been shown to increase robustness by reducing the impact of any one model's inability to detect and compute accurate class probabilities on out-of-distribution data. Several ensembling techniques exist, such as Bayseian voting and modifying inputs via bootstrap aggregation. To implement bootstrap aggregation ("bagging") multiple independent observations are taken and passed to each predictor in the ensemble [4]. The "shallow" equivalent to bagging is a random forest of decision trees.

3 Background

Since this task is largely based on the robustness of the deep learning method to distributional shift, we explore varied novel data representations for reviews, methods of data augmentation, pre-trained ensemble methods, and handpicked underrepresented linguistic features to maximize the accuracy on the perturbed test set. We ensure that while the accuracy over the best-case distribution is not necessarily maximized (i.e. over the raw training set), the accuracy over a worst-case distribution is preserved.

Throughout all experiments, we rely on a pre-trained machine learning method, typically a transformer-based language model, that feeds into a simpler post-processing head such as a multi-layer perceptron (MLP)[11], support vector machine (SVM) [5], or even a long-short term memory (LSTM) layer [6].

To increase robustness, we introduce three methods of data augmentation particular to NLP: cropping text in various regions, merging different similarly themed texts as measured by cosine similarity, and scrambling sentence orders and stop words. For the purposes of data augmentation, we refer to the length of the text T as $|T|$ and any particular character as T_i. The corresponding review in stars is denoted as y.

The first augmentation is straightforward and entails cropping text at location $T_{p_i|T|}$ with probability p_i. Given a uniform distribution over p, we sample to generate different possible ways to crop the text in each batch. For example, p_1 could correspond with cropping the introduction and would be sampled between 0 and 0.3. Correspondingly, we choose which region to crop using a random number r. The second augmentation is by merging parts of two texts T_1 and T_2 with both texts having the same rating $y_1 = y_2$. Further, we ensure that the texts are contextually similar by measuring the cosine similarity of T_1 and T_2 to ensure that it is at least some β [8]. The final augmentation is to scramble sentence orders and swap stop words and subjects (i.e. he is replaced with she, the is replaced with a). Since these are particularly risky augmentation that can remove or reorder references that BERT and other language models pick up, it is applied with a low probability on shorter texts [2].

We explore various types of data representation types that are gathered from the typical format of a review. The first is an introduction-explanation-conclusion type of review that is traditionally associated with longer reviews. The second is the introduction-only reviews associated with 1-star or 5-star reviews. The final review type has all parts bar a conclusion. As such, we apply simple techniques to segment the reviews into its logical components on which to apply machine learning methods. We refer to this technique as textual relevance segmentation.

A core component discussed is the ensembling of both pre-trained language models and trained post-processing heads (i.e. an MLP or SVM). Ensemble methods are typically used to reduce variance in predictions, which is particularly important in robustness against distribution shift. To ensemble pre-trained models, we simply run multiple forward passes of many different (different architectures and training sets) language models and concatenate them. The intuition behind is relies on the notion that different models are trained on different amounts and types of texts, producing a unique and valuable hidden state representation. It allows the model to react appropriately to perturbations in types of vernacular (i.e. formal language as opposed to informal in reviews). Since some language models are trained on research publications while some are trained on informal blogs, the representations are uniquely valuable. To ensemble the trained machine learning heads, we predict on all generated hidden-state representations with the different methods. Then, we use soft voting.

Finally, we add a handful of handpicked features to address underrepresented linguistic features such as the frequency of punctuation and capitalization, sentence and review length, and others. While this may seem rudimentary, an all capital review could easily be classified as a likely 1-star or 5-star review.

4 Approaches and Results

The first baseline was a simple DistilBERT-based sentence transformer available in the `sentence_transformers` Python library followed by an SVM head. No data augmentation or ensembles were used. Table 1 includes evaluations of both accuracy and mean absolute error after 1 and 10 epochs. Over 10 epochs of SGD, we achieve a reasonably good accuracy on the (unperturbed) validation set. While the SVM was a principled approach, it was quite basic in the features that were represented.

Table 1: Performance of SVM on Transformer State

Metric	Value at 1 Epoch	Value at 10 Epochs
Accuracy	0.67	0.73
MAE	0.43	0.40

We opted for an MLP, which is a standard head. The MLP produced an accuracy of around 70% on the validation set. However, the standard MLP struggled with many perturbations from the training data.

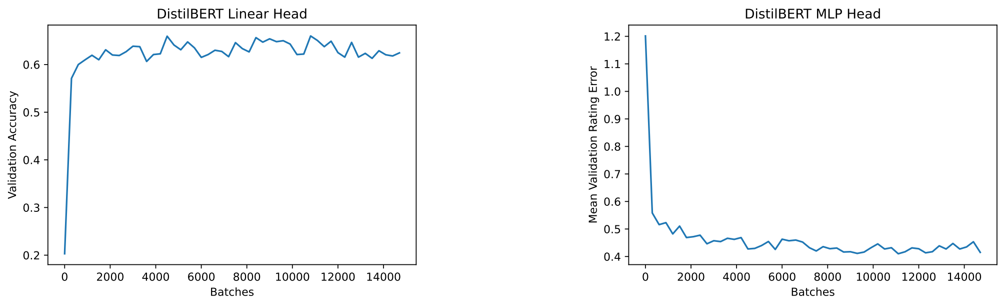
For the introduction of the ensemble-based models, we chose to use two pre-trained BERT models, DistilBERT [12] and RoBERTa [9]. Since RoBERTa is a robust method as described by Liu et al [9], we hoped to solve some issues of distributional shift. The accuracy of the approach increased to around 75.66% on the validation set. Unfortunately, on perturbed data, the method still struggled with nearly the exact accuracy as the previous method.

To account for the lingering issue of robustness, we introduced data augmentation into the training process. Text cropping, which was the first data augmentation technique, proved quite successful and provided uniquely perturbed data. We used both the second and third augmentation with lower probability. Further, to tackle the vast class imbalance as shown by approximately 75% of the training set being 1 or 5-star reviews, we randomly dropped some 1-star and 5-star reviews.

We show a training curves in Figure 1 for the MLP at this stage in the development process (but prior to the introduction of our data augmentation techniques) to illustrate the evolution of the accuracy on the validation set.

Unfortunately, the MLP still struggled with the aforementioned perturbations. An observation was that the MLP struggled with varied length. To combat this, we applied the segmentation techniques based on the data representations of a review. Due to computational limitations, we asserted that each review consisted of an introduction with an optional explanation. The choice of where the introduction ended was fairly ad-hoc with a simple length-based heuristic. This increased our accuracy considerably perhaps due to the increased number of features. A visualization of this approach dubbed FCBERT is shown in Figure 2.

Finally, we use the standard approach of ensembling checkpoints of models at different phases of training. This was a fairly standard way of reducing variance without any further computational expense.



(a)

(b)

Figure 1: Training curves (evaluated on a validation dataset): fine-tuning DistilBERT with a linear classification head. This model achieves about 62% validation accuracy after training on a small fraction of the training set before performance levels off. Both the training and validation datasets were re-weighted to reduce class imbalance.

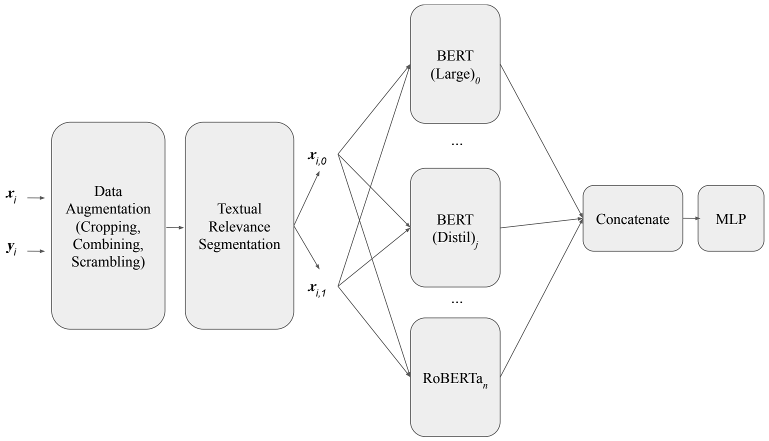


Figure 2: FCBERT: segmentation of text based on review data representation followed by ensemble of BERT in a fully-connected style. While this model had great performance, it was too slow for the submission constraints.

The results of our models on two released test sets are shown in Table 2. The baseline model is aforementioned SVM model. The final model is an ensemble of LSTM-based models operating on ensembled BERT Large outputs of segmented texts (i.e. segmented into introduction and remaining text). It was surprising that the robust BERT (RoBERTa) & MLP combination used didn't perform particularly as well with perturbations.

Table 2: Model Performance

Model / Dataset	Accuracy	MAE
SVM / Challenge 5	0.04	1.17
SVM / Challenge 8	0.62	0.596
Final / Challenge 5	0.536	0.58
Final / Challenge 8	0.632	0.492

5 Conclusion

We developed a robust Yelp review text classifier using a novel combination of ensembling and data augentation. Our final model achieved solid out-of-distribution classification accuracy of 53.6% and 63.2% on two challenge datasets. Our results emphasize the importance of data augmentation as a strategy for improving generalization performance of deep learning models. Using an ensemble of models, with different architectures and pre-trained on datasets, provides further advantages for generalization. We expect that the general techniques explored in this report will be useful in improving generalization on other tasks as well.

6 Team Contributions

Anirudhan and Eric performed experiments and made figures. All group members contributed to the text. Charlie provided additional organizational impetus. In terms of overall effort, we break down the effort as **45%** Anirudhan, **30%** Eric, and **25%** Charlie.

References

[1] Aparna Balagopalan and Jekaterina Novikova. Augmenting bert carefully with underrepresented linguistic features, 2020.

[2] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*, 2019.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] M. A. Ganaie, Minghui Hu, M. Tanveer, and Ponnuthurai N. Suganthan. Ensemble deep learning: A review. *CoRR*, abs/2104.02395, 2021.

[5] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[7] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[8] Baoli Li and Liping Han. Distance weighted cosine similarity measure for text classification. In *International conference on intelligent data engineering and automated learning*, pages 611–618. Springer, 2013.

[9] 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 pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:undefined*, 2013.

[11] Fionn Murtagh. Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5-6):183–197, 1991.

[12] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[13] Nishit Shrestha and Fatma Nasoz. Deep learning sentiment analysis of amazon.com reviews and ratings. *arXiv preprint arXiv:1904.04096*, 2019.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[16] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.