CSC594 Final: Data Imbalance and Few-Shot Learning With BERT and Multiclass Classification

Clayton Cohn

November 21, 2020

1 Introduction

This assignment involved choosing a research topic and investigating it in a manner similar to that of a formal research paper that one would prepare for submission to a journal or conference. For my topic, I had originally chosen the comparing of various Bidirectional Encoder Representations from Transformers (BERT) models on a corpus of short-answer essays written by Chicago high school students but later pivoted to researching data imbalance and few-shot learning with the same dataset. The high school students whose essays comprised the corpus were given a variety of sources from which to ascertain the various causes of skin cancer and then were asked to write a short-answer response to convey what they had learned. The various causal chains extracted from the sources is illustrated in Figure 1. Any mention by a student of a particular concept and its precipitating an additional concept constitutes a causal chain. For example, a student's mentioning that, "an increase in exposure to direct sunlight leads to an increase in the likelihood of sunburn," indicates that concept 2 causes concept 5 (in Figure 1). As such, this sentence would be labeled with the causal relation "R-2-5."

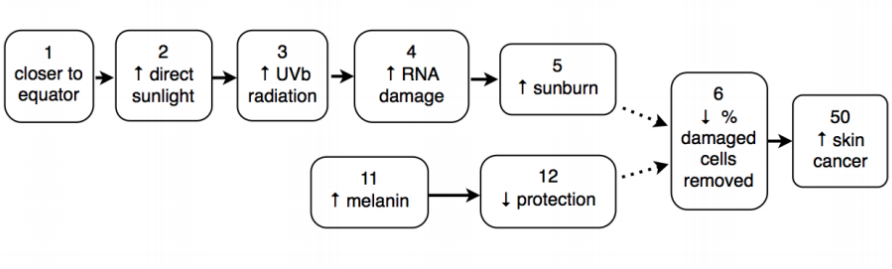


Figure 1: *Causal Model for Skin Cancer Concepts* [Hughes, 2019]

The task I am concerned with for this paper is identifying the causal chain (if present) in each sentence, making this a multiclass sentence classification task. For his doctoral dissertation, Simon Hughes had used the combination of a sequence tagging model with a bidirectional Recurrent Neural Network to achieve an F1-score around 84% [Hughes, 2019]. He also incorporated other grammatical tasks such as dependency parsing and part-of-speech tagging. I am quite familiar with this dataset, as I have spent a considerable amount of time modeling it in the last year while working on my thesis [Cohn, 2020], as well as while contributing to [Cochran et al., 2020]. My goal has been to try and recreate Dr. Hughes' results with only a language model (BERT). Unfortunately, I have so far been unable to do so for a variety of reasons—mainly that the dataset is imbalanced and sparsely populated, which is why I chose to address this in my final project.

2 Related Work

There has (surprisingly) been a rather limited amount of research conducted *specifically* (i.e., for the sole purpose of) addressing BERT's application to imbalanced datasets. A Google Scholar search for "allintitle: imbalanced bert" (i.e., all papers whose titles include the words "imbalanced" and "BERT") yielded only two results–both of which touted a method known as *cost-sensitivity* [Tayyar Madabushi et al., 2020]. With cost-sensitivity, the instances of the underrepresented classes are cost-weighted to account for their sparsity. By doing this, the model is able to prioritize the correct classification of the underrepresented classes, as reducing the cost of these instances yields the greatest reduction of the loss function [Tayyar Madabushi et al., 2020]. Unfortunately, due to time constraints, implementing cost-sensitivity was not feasible for this assignment.

An additional Google Scholar search "imbalanced bert" (i.e., each word appearing somewhere in the entire article) was conducted and yielded nearly two thousand papers. Most of these were from open access databases such as arXiv, however, and were therefore not peer-reviewed. This has been typical in my (limited) research experience, as the publication process is often lengthy, and BERT is only about two years old. [Oak et al., 2019] was able to achieve promising results in detecting malware from labeled action sequences of Android applications. [Oak et al., 2019] achieved an F1-score of 0.919 on a 180,000-instance dataset consisting of only 0.5% malware (the other 99.5% of the instances were action sequences of non-malicious Android applications). While this result was good enough for a new state-of-the-art (SOTA) for this particular dataset, it was not really relevant to my research for this assignment. Although [Oak et al., 2019] modeled an imbalanced dataset, the task itself was one of binary classification. As such, it was more akin to outlier detection than multiclass classification and thus did not provide me with any actionable implementation.

At this point, I decided to explore few-shot learning. Although sparsely populated, the skin cancer dataset does have more than two instances in nearly half of its classes. Therefore, I thought that if I was able to find a way to adapt BERT for few-shot learning that this would mitigate a large part of the multiclass data imbalance that I had been trying to remedy. Searching Google Scholar for "allintitle: bert "zero-shot"" and "allintitle: bert "few-shot"" yielded five results. Of the five, the most interesting was a paper called "CG-BERT," which adapted BERT for text generation to improve few-shot intent detection via data augmentation. However, like cost-sensitivity, this method was not implementable due to the time constraints of the assignment.

3 Methodology

The first thing I did was combine the data to form a single set. Previously, the data had been pre-separated into training and test sets, but this was not conducive to performing cross-validation. As such, I created a Jupyter notebook called *SkinCancerCombine.ipynb* that coalesced the two sets. The dataset also needed to be cleaned, as some instances had incorrectly placed newline characters that hindered parsing by Pandas. Additionally, I wanted to establish a baseline from which to try and improve upon. This was done by running the *BERTClassification.ipynb* notebook on the unadulterated dataset with five-fold cross-validation across all fifty-four classes (relations). The distribution of the dataset is shown below. Each relation is listed with its corresponding number of instances. There were 12,639 total instances. One can clearly see the imbalance present in the distribution of data, which strongly resembles a Zipfian (thefreedictionary.com) distribution. Figure 2 shows the distribution across all fifty-four classes in order of frequency of occurrence. One can see that the "O" class (no relation) contains nearly half of the 12,639 instances, while the right-most twenty classes are barely visible due to their only consisting of one or two instances each.

O: 5791	R-12-3: 288	R-2-4: 91	R-12-2: 20	R-2-6: 5	R-50-3: 2	R-4-11: 1
R-5-50: 1051	R-2-3: 284	R-5-4: 76	R-11-3: 15	R-50-2: 4	R-3-2: 2	R-5-12: 1
R-2-50: 753	R-4-5: 268	R-11-50: 70	R-6-5: 9	R-2-1: 4	R-4-RH: 2	R-2-11: 1
R-1-2: 638	R-3-4: 267	R-4-6: 63	R-5-5: 8	R-4-4: 2	R-3-12: 2	R-3-11: 1
R-1-50: 620	R-11-12: 231	R-3-5: 55	R-11-5: 7	R-2-2: 2	R-12-12: 1	R-50-1: 1
R-5-6: 587	R-2-5: 135	R-3-6: 34	R-1-4: 6	R-6-4: 2	R-11-4: 1	R-4-12: 1
R-3-50: 473	R-4-50: 132	R-1-5: 26	R-50-5: 6	R-12-4: 2	R-50-4: 1	
R-6-50: 469	R-1-3: 95	R-12-50: 25	R-12-5: 5	R-5-11: 2	R-6-3: 1	

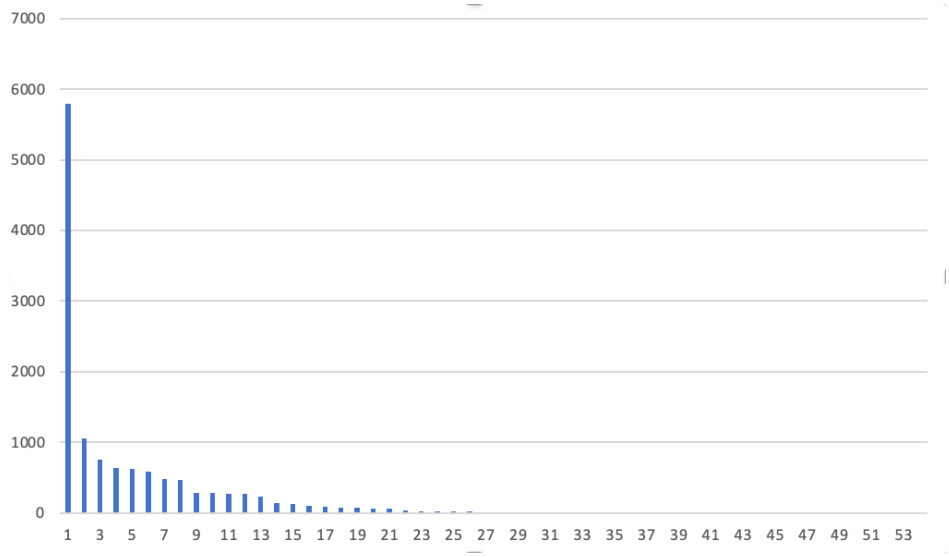


Figure 2: *Class Distribution of Causal Chains in Skin Cancer Dataset*

Once a baseline was established with all of the data, I started to implement various data balancing approaches. The first method I applied was the simplest: omitting instances from classes with fewer than ten instances. While a seemingly primitive approach, this is a rather common approach taken when class sizes are large and numerous in quantity. [Song et al., 2019] took this approach, for example, when applying Chinese BERT to the classification of Traditional Chinese Medicine cases. In this case, the authors only considered the five most abundant classes. For my work, limiting the classes to the ones consisting of ten instances or more still left me with twenty-six trainable classes (the smallest of which consisted of fifteen instances).

Secondly, I tried reducing the size of the "O" class to see if that would have any effect on improving the model's recall, thinking that a reduction of the most prevalent class would lead to fewer false negatives and thus precipitate a greater recall. To do this, I generated a random sample (without replacement) of one thousand instances from the "O" class, as this was close to the quantity of the next most-frequently occurring class. Only one round of sampling was performed, and the model was not evaluated to determine how sampling bias affected its performance.

For the third task, I explored data augmentation. While I was unable to apply the aforementioned generative BERT variant to generate "silver label" (i.e., computer-generated) instances, I was able to apply a different type of data augmentation technique typically reserved for ensembles: bootstrap aggregating, or *bagging*. With bagging, data is augmented by within-class instances via random sampling with replacement, which creates similar (but different) training batches. Twenty percent of each class' instances were withheld from sampling in order to create a stratified test set.

Lastly, I explored few-shot learning. While not an initial goal at the outset of this work, it seemed particularly relevant considering the sparse population of the dataset. Over half of the classes (twenty-eight our of fifty-four) consisted of fewer than ten instances, and only fifteen classes had over one hundred instances. To experiment with few-shot learning, I used the same group of twenty-six classes that had at least fifteen or more instances. This time, I distilled each class down to fifteen instances that were collected via random sampling without replacement.

For all of the previous methods, all experiments were conducted with "bert-base-uncased." Additionally, all experiments used the Adam optimizer with a learning rate of 0.00002, a warm-up rate of 0.1, and a batch size of sixteen (as per [Devlin et al., 2018]'s guidance). Likewise, training (fine-tuning) was conducted for four epochs universally (except for the few-shot instances, which I will explain in the next section).

4 Results

The results from implementing all five methods with five-fold cross-validation are illustrated below in Figure 3. Each cluster of columns represents an accuracy metric (accuracy, micro-F1, macro-F1, etc.), and each column in each cluster represents a different method. All of the methods are color-coordinated. "Cropped" refers to the method whereby instances were discarded if they belonged to a class consisting of only one or two instances. "O 1000" refers to the method where the "O" (no relation) class was reduced to one thousand via random sampling (without replacement). The details of each method from Section 3 are enumerated in the subsections that follow.

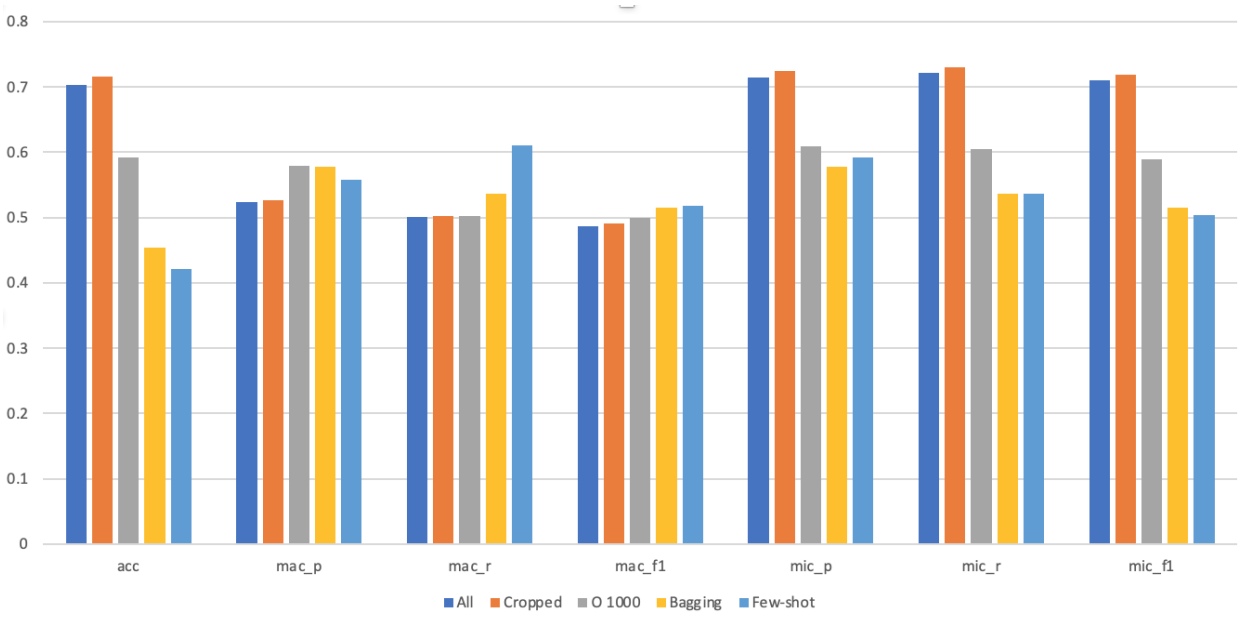


Figure 3: *Comparison of Metrics Across Multiple Methods*

4.1 Full Dataset (All)

I first conducted five-fold cross-validation on the entire set of causal relations (regardless of frequency of occurrence) in order to set a baseline (base case) from which to compare the other methods. The results are illustrated in Figure 4. One that can see that the accuracy and micro metrics are in the low seventies; however, the macro metrics are significantly lower. This is a result of the data imbalance present in the dataset. Micro statistics are calculated based on the mean of all of the instances across every class. Thus, increased weight is placed on the classes with the most instances. The "O" relation, for instance, achieved precision, recall, and F1-score metrics of 92.71%, 92.47%, and 92.59%, respectively. This is likely due to the model optimizing for the best-represented class. The macro statistics weigh all classes equally, and it is these metrics that I am hoping to improve in the subsequent experiments.

4.2 Reducing the Class Number (Cropped)

The results from reducing the number of classes to include only those which contained ten or more instances are illustrated in Figure 5. Surprisingly, these results are very similar to the base case metrics depicted in Figure 4. Initially, I thought that discarding the one and two-shot instances would significantly augment the model's performance. Cropping the classes ensured that each class had better representation, and it also reduced the number of classes by 52% (by omitting twenty-eight out of fifty-four classes). Still, performance metrics are almost identical to the base case. There appears to be a slight improvement in the cropped dataset uniformly across all metrics, but this is negligible, as the difference in performance is

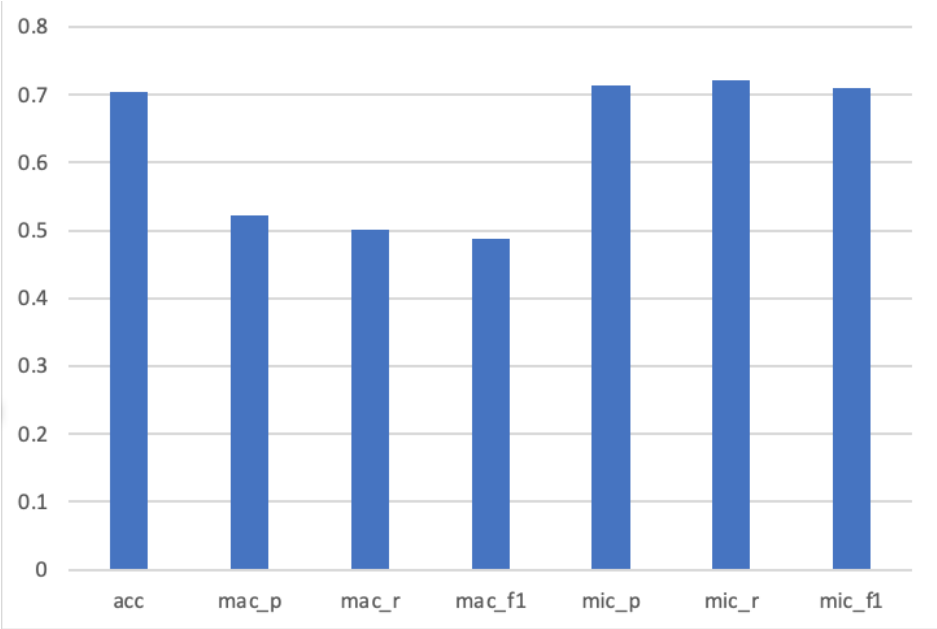


Figure 4: *Performance Metrics Across All Classes (Base Case)*

statistically insignificant. My hypothesis is that there was little difference in performance due to the model's optimizing its predictions in favor of the prevalent class; however, I do not currently have evidence to support this, and further research must be conducted to discern why these results are so similar.

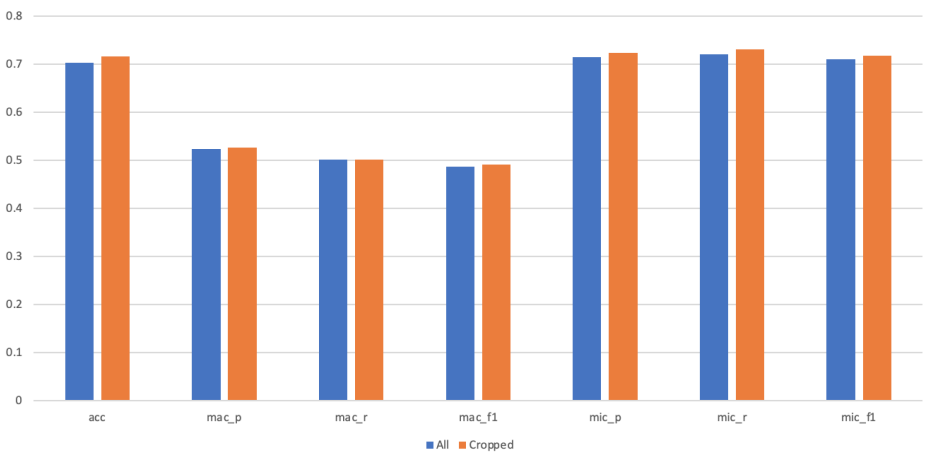


Figure 5: *Performance Metrics of Reduced Class Number*

4.3 Reducing the Prevalent Class (O 1000)

The results from reducing the size of the prevalent "O" class are illustrated in Figure 6. While there was a slight improvement in the macro statistics relative to the base case, none of the deviations were statistically significant. Additionally, the accuracy and micro metrics dropped considerably. The latter is not particularly surprising, as reducing the weight of the most prevalent (and best-performing) class could reasonably be expected to aid in balancing the data between classes and cause the micro and macro metrics to edge closer to each other. Unfortunately, however, these results show a decrease in micro metrics without a corresponding increase in macro metrics, indicating that reducing the prevalent class did not help the

model better generalize to other classes.

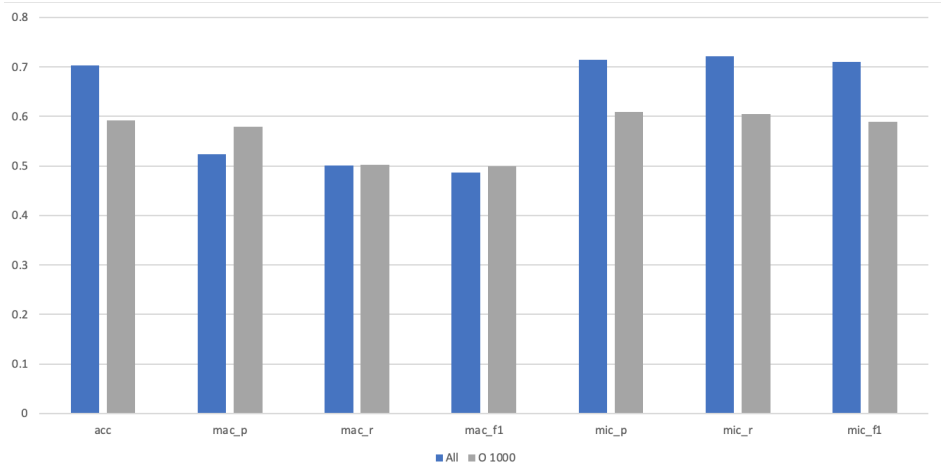


Figure 6: *Performance Metrics of Reduced Instances of Prevalent Class*

4.4 Bagging

The results from bagging are illustrated in Figure 7. While a rather primitive data augmentation technique for this particular task (i.e., no additional new instances were generated), it nevertheless yielded some interesting results. While the accuracy and micro metrics were considerably lower than those of the base case, all three macro metrics saw an increase–by 3.97%, on average. Originally, I had thought that bagging would be deleterious across all metrics, as many classes were only comprised of around twenty instances (after cropping), and would thus be prone to overfitting when creating randomly sampled classes of one thousand from only a few instances. Still, the macro metrics did see a slight improvement, indicating that the model was able to potentially better-generalize its results across all classes. The macro metrics with bagging outperforming those from the base case was statistically significant, as a two-tailed paired t-test yielded $p = 0.0002$ when evaluating all three metrics across all five cross-validation instances. This was less than the critical $p = 0.05$ value needed to reject the null hypothesis that the macro and micro metrics performed equally.

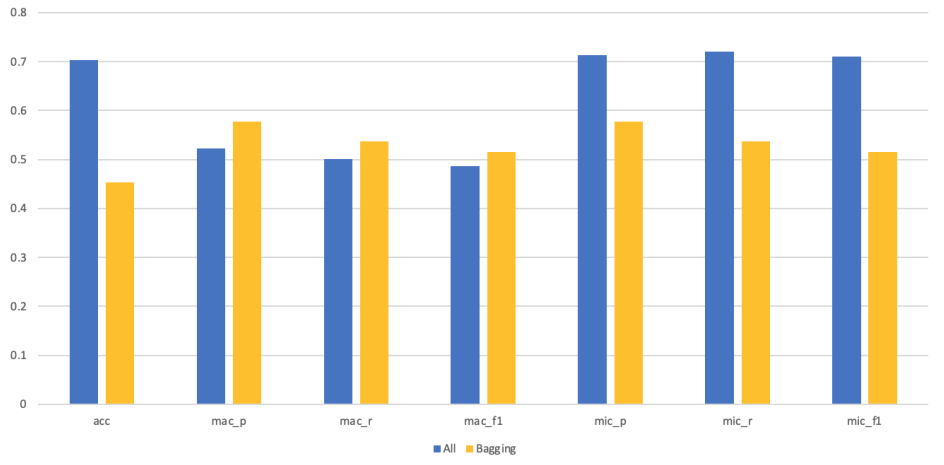


Figure 7: *Performance Metrics of Data Augmentation Via Bagging*

4.5 Few-Shot Learning

The results from few-shot learning are illustrated in Figure 8. As this was my first foray into few-shot learning (with BERT or otherwise), I was uncertain what to expect. Like the other methods, the results showed a decline in accuracy and micro metrics; however, results for macro metrics were significantly higher—5.85% on average. A two-tailed paired t-test yielded a value of $p = 0.0038$ which was below the critical value of $p = 0.05$ required to reject the null hypothesis that the macro metrics of the few-shot method and the base case were equivalent. This metric was calculated comparing the precisions, recalls, and F1-scores across all five cross-validation instances (for a total of fifteen comparisons).

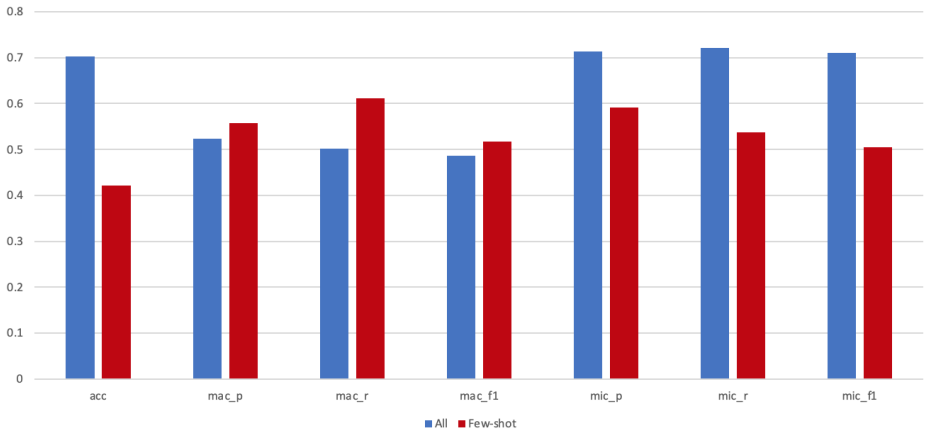


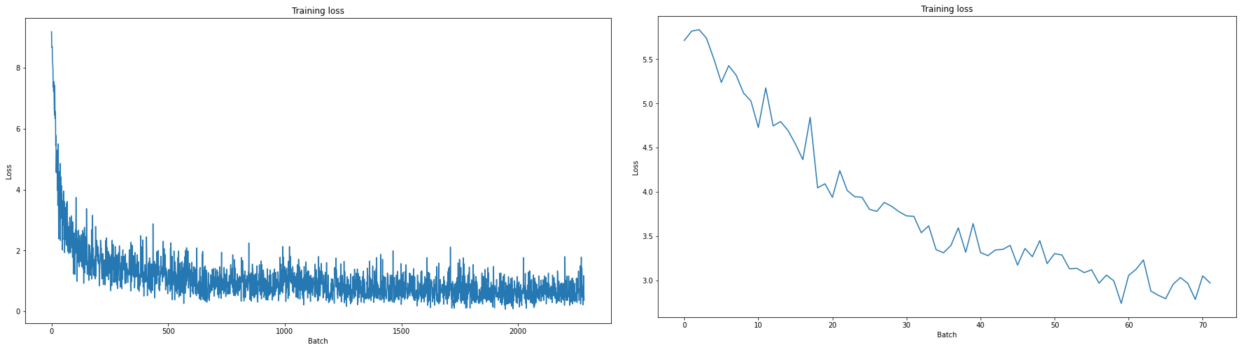
Figure 8: *Performance Metrics of Few-Shot Learning*

Additionally, it is worth noting that the recall for the few-shot method was higher than the recall in any other method—including the base case. Measuring the recall in isolation yielded a 10.94% increase in favor of the few-shot method over the base case. Surprisingly, even with only five comparisons (the five macro recall instances of the base case compared to the five macro recall instances of the few-shot method), these results were statistically significant: a two-tailed paired t-test yielded $p = 0.0494$ which was less than the $p = 0.05$ value needed to reject the null hypothesis that the macro recalls of the base case and few-shot method were equal. Recall is calculated as the true positives as a percentage of true positives and false negatives:

$$R = \frac{TP}{TP + FN}$$

Therefore, the increase in recall is indicative of either the few-shot model obtaining more true positive instances or fewer false negative instances (or both). Furthermore, this is even more impressive when considering that each of the twenty-six classes consisted of only twelve training instances (the remaining three were held out to created a stratified test set). However, it is possible that a stratified test set may outperform an imbalanced one. As such, more research needs to be conducted to determine the effects of the stratification.

Lastly, it took the few-shot model more epochs to train due to slower convergence. [Devlin et al., 2018] recommended fine-tuning for two to four epochs; however, with few-shot learning, my model had not yet converged. As shown in Figure 9, the base case in Figure 9a clearly converges at the end of the four epochs, while the few-shot method in Figure 9b appears to still be trending downward. As a result, I continued doubling the epochs until the few-shot method began converging (at n = 32). Up until that point, I ensured that the metrics (accuracy, macro, and micro) continued to improve in order to minimize the risk of over overfitting the data.



(a) *Base Case Loss Over Four Epochs*

(b) *Few-Shot Method Loss Over Four Epochs*

Figure 9: *Base Case Loss Versus Few-Shot Loss Over Four Epochs*

5 Discussion

Altogether, I implemented four different methods (in addition to the base case) addressing BERT's application to an imbalanced and sparsely populated dataset. Two of these methods involved data reduction, i.e., reducing the number of classes or the number of instances in the prevalent class. Additionally, one method involved data augmentation via bagging, and one method involved distilling each class to fifteen randomly sampled instances to perform few-shot learning. The most significant result from this work was that of the macro performance improvement precipitated by the few-shot model. It is very interesting that a model trained on only twelve instances of each of its twenty-six classes was able to perform better in a macro setting than the full (albeit imbalanced) model with over twelve thousand instances. I also think that the accuracy of the few-shot model is very important to note. The few-shot method yielded an accuracy of 42.17%, which may not seem like a positive result, but the fact that the model was able to achieve this with a balanced dataset was an impressive feat. Conversely, although the base model achieved an accuracy of over 70%, these results were not noteworthy when you consider the massive data imbalance that resulted in the model merely learning how to proficiently identify members of the prevalent class. Also, if one considers that there were twenty-six (equally represented) classes, then the chance of the model (randomly) predicting correctly in any one instance was 3.85% (one out of twenty-six). When considering this, the fact that the model was still able to guess correctly more than forty percent of the time is rather impressive.

This was, unequivocally, my favorite assignment so far. I definitely spent the most time on it, as I found it to be the most interesting. In many of the papers I have read during my research, multiple researchers have mentioned problems with BERT (and other models) in few-shot and data-imbalanced settings, so it was nice to be able to tackle this first-hand. As far as challenges are concerned, my only problem was time. Given that this was the final project for a course (as opposed to an ongoing research project), I had to limit the scope of my work considerably. In addition, I also had my thesis to finish (and my defense to prepare), which significantly lessened the time that I had available to contribute to this assignment. As such, I was forced to experiment with simpler methods when I would have much rather studied some more "interesting" approaches (such as sentence generation for data augmentation, for example). Still, this was definitely my favorite assignment yet.

6 Future Work

For future work, there are a few things that I would like try. Employing cost-sensitivity is something I have wanted to experiment with for some time but have unfortunately been unable to do within the time constraints and scope of my current studies and research. Generative data augmentation also seems very promising. The CG-BERT model mentioned earlier claims to have had promising results, and I would also like to try other methods of data augmentation common in NLP such as Back Translation, Random Swap, Random Insert, and Synonym Replacement. Additionally, I would like to do more experiments with BERT in few-shot settings and explore what makes BERT deviate from the prevalent class of an imbalanced dataset

when predicting few-shot instances. Lastly, I would like to see what effect test set stratification has on the model's overall performance (e.g. training the model on balanced data but then evaluating imbalanced test sets). As I continue to explore the skin cancer (and coral bleaching) dataset(s), I wonder if some relations (classes) are better suited for correct classification than others. This can only be discerned by evaluating each class individually on the same model, which I similarly think would be worth exploring in the future.

7 Resources

Google Scholar was used to gather research for this work, as that database aggregates sources from all major research databases–both published and open access. Additionally, I utilized Google Colab for all of my coding. The data balancing and k-fold cross-validation workbooks were my own, but the BERT classification workbook was adopted from a colab.research.google.com tutorial. The tutorial used Pandas, NumPy, and PyTorch/HuggingFace Transformers. The dataset is proprietary and was compiled by researchers at Northern Illinois University (NIU). I also relied heavily on my previous works [Cohn, 2020] and [Cochran et al., 2020], as this work was largely a continuation of the research I have been conducting over the past year.

References

[Cochran et al., 2020] Cochran, K., Cohn, C., Hastings, P., and Hughes, S. (2020). Transformer Models for Identifying Causal Relations in Students' Exploratory Essays [Unpublished manuscript]. College of Computing and Digital Media, DePaul University.

[Cohn, 2020] Cohn, C. (2020). Bert efficacy on scientific and medical datasets: A systematic literature review. Master's thesis, DePaul University.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805.

[Hughes, 2019] Hughes, S. (2019). *Automatic Inference of Causal Reasoning Chains From Student Essays*. PhD thesis, DePaul University.

[Oak et al., 2019] Oak, R., Du, M., Yan, D., Takawale, H., and Amit, I. (2019). Malware detection on highly imbalanced data through sequence modeling. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, AISec'19, page 37–48, New York, NY, USA. Association for Computing Machinery.

[Song et al., 2019] Song, Z., Xie, Y., Huang, W., and Wang, H. (2019). Classification of traditional chinese medicine cases based on character-level bert and deep learning. In *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pages 1383–1387.

[Tayyar Madabushi et al., 2020] Tayyar Madabushi, H., Kochkina, E., and Castelle, M. (2020). Cost-Sensitive BERT for Generalisable Sentence Classification with Imbalanced Data. *arXiv e-prints*, page arXiv:2003.11563.