Deep Learning Assignment 2
Sparrsh Nagdda
S3810177
17th October 2022

Problem Definition and Background

Memes have undoubtedly had one of the biggest impacts on the world through social media in the last decade. Memes can convey a lot of information about the current political climate. It can also subtly change the perception of its reader. A similar task to identify persuasion techniques in memes was announced in SemEval 2021(Task 6). SemEval is an international workshop on semantic evaluation.

Our task is to develop and analyse a model which takes a meme with image and text on it as input and predicts which persuasion techniques are involved in the meme. This was subtask 3 of task 6. It should be duly noted that the winners at the workshop scored an F1 score of 0.5811 on the test set. This just goes to show that the problem in question is a difficult one to solve by a deep learning approach and results might not be very high. Another important thing to keep in mind is that a meme can have multiple techniques assigned to it and so, the nature of the problem is a multi label classification problem. The approach involved to solve this problem has been inspired from the implementations of the participants at the workshop, particularly Peiguang et al., 2021 and Zhu et al., 2021.

Evaluation Framework

As evident from the reports of the participants, F1 score is a suitable measure for the problem at hand. When we begin exploring the data, we notice that some labels are in majority or in other words, the labels are not perfectly balanced. This rules out accuracy as a metric since the model will otherwise tend to predict the majority label more often than not. Precision calculates the number of total true positives divided by the total true and false positives. Recall calculates the true positives divided by the sum of the true positives and false negatives.

F1 Score takes the best features of both precision and recall and combines it into a single metric. It is the harmonic mean of precision and recall. There are 2 types of F1 score: macro F1 and micro F1. Macro f1 treats every class equally and hence, cannot handle imbalanced data. Micro F1, on the other hand, involves calculation of true positives, false negatives and false positives. It can also account for partially correct prediction and hence, best equipped for a multi label classification problem. Hence, micro f1 score is chosen as the evaluation metric to evaluate the models. Our aim would be to have a f1 score above 0.35.
As regards the loss function, binary cross entropy is chosen since we have a multi label classification problem involving 22 unique classes. Since we are using binary cross entropy as a loss function, we will use the sigmoid activation function in all the classifiers of the models.

Approach and Justifications

In this task, the data is already split between train, val and test sets. We observe that the train set is very small. Hence, we do not split the train set further to have additional data for testing. Also, given that the input in question are memes and contain text as well, we cannot use data augmentation or perform undersampling or oversampling to help solve the class imbalance.

We continue with data exploration and preprocessing. We get insight about the number of words in a meme, with the maximum being around 100. We perform some sanity checks which involve checking for missing labels, missing images and ensuring that the images are not corrupted(all values are integers). We noticed that there were 29 images with no label and we opted to drop them. To have further info about special characters, we get the number of special chars in the train split. There are 273 memes whose text contains special characters. We conclude that special characters might change the tone or the implication of what is trying to be conveyed and so we leave them. However, we get rid of blank spaces or ‘“\n”. This is because the language models are trained on strict English

corpuses. “\n” is more of a system language and so we opt to drop it. However, we do not touch the validation and test sets when it comes to removing the blank spaces and dropping unlabelled memes. The labels were encoded between an integer between 0 and 21 in the train data.
We load the data via a custom data generator, which ensures that irrespective of the split or presence of missing labels, each label component of the tensor will contain 22 binarized labels. This is done via the loader to ensure consistency and prevent data leakage if explicitly done for validation or test. Now, we move on to model architecture.

The implementation derives a lot of inspiration from Figure 3 of Zhu et al’s paper for this solution of subtask 3 for task 6. It gives a lot of insight into how to handle mixed data in the model by creating 2 different branches; one to handle the image component and one to handle the text component. Then, we concatenate the pooled outputs from the 2 different branches and pass it through a classifier. We considered Resnet50 and EfficientNetB7 for the CNN component and BERT and Expert Bert for the text part. Zhu et al. made do with VGG or Resnet18. However, we use more powerful models to try and improve the performance.

Notably, the dataset we have is very small which makes transfer learning a viable option. Our CNN’s use the ImageNet weights whereas the BERT and Expert Bert are trained on Wiki corpuses. It is important to note that Bert was favoured over other models like Word2Vec since BERT understands the context of words too. Moreover, since our words are less than 128, we can use the standard BERT preprocess methods. We do not tinker with the image resolution since we want to maximise performance by keeping the dimension on which the CNN was trained for Imagenet. The images in the dataset have clarity in that resolution. The scaling to (224, 224, 3) in case any image was smaller or bigger is done in the data loader. We trained 4 baseline models with 2 variations:
i) Model with no dense layer before the classifier
ii) Model with one dense layer before the classifier

This variation was considered because some of the models with no dense layer before the classifier were very noisy and as a result, had a high generalisation gap. Hence, a total of 8 baseline models were trained out of which the better variant was selected for hyper parameter tuning. As a result, we had 4 models for tuning. To select the better variant, priority was given to a graph which was less noisy and had less generalisation gap. The models, thus selected for tuning were :

1. Resnet-50 and Bert En Cased L-24 H-1024 A-16 with no dense layer before the classifier
2. Resnet 50 and Expert Bert with one dense layer before the classifier
3. Efficient Net and Expert Bert with no dense layer before the classifier
4. Efficient Net B7 and Bert En Case L-24 1024 A-16 with one dense layer before the classifier

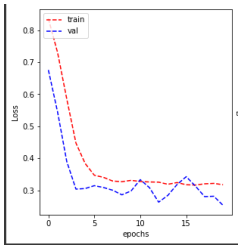
Experiments and Tuning

The 4 models were tuned using a Bayesian Optimizer. It is an automated tuning process which runs a number of trials based on certain values for parameters we decide. Due to visible overfitting, a dropout layer was added to each of them with possible values of {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6}.
If there was a dense layer before the classifier, that was added as a parameter too with possible values of {128,256,384,512}. Additionally, the learning rate was considered to be a hyperparameter too and adamW is the optimizer inplace of adam. The reason for this was to make the graphs less noisy and also follow the original BERT implementation. The tuning seeked to optimise val_binary_crossentropy and make it as minimum as possible. The tuned parameters for the 4 models were as follows :

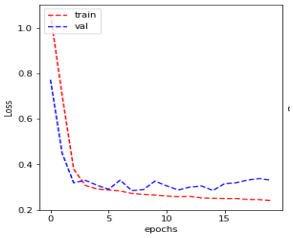
a. {'rate_dropout': 0.5, 'learning_rate': 5e-05}
b. {'units': 512, 'rate_dropout': 0.6000000000000001, 'learning_rate': 5e-05}
c. {'rate_dropout': 0.6000000000000001, 'learning_rate': 5e-05}

d. {'units': 384, 'rate_dropout': 0.5, 'learning_rate': 3e-05}

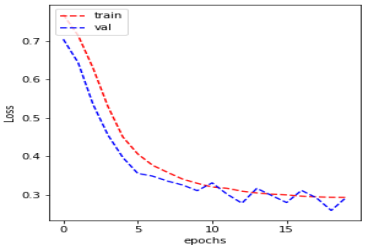
The 4 loss graphs were :



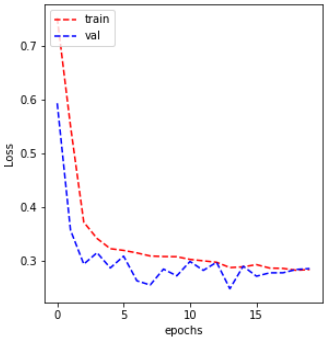
Loss v Epochs for Model 1



Loss v Epochs for Model 2



Loss v Epochs for Model 3



Loss vs Epochs for Model 4

Ultimate Judgement, Analysis and Limitations

From the graphs above, we have to choose a model. We use 2 factors to determine the best model : less overfitting and minimum generalisation gap. We see that Model 1 has a large generalisation gap, in Model 2 it begins to overfit considerably, in Model 3 the generalisation gap stays par for the course but begins to overfit at the end. Also, Model 3 has visible spikes in learning. In lieu of this, Model 4 is chosen because the generalisation gap decreases a lot as the model learns more epochs and overfitting also decreases. There is still minute overfitting towards the end but the generalisation gap is almost 0.

We now load the test set via the data loader and predict the test set labels. We then apply post processing and get a micro f1 score of 0.41 through the classification report. This means that we achieved our target. This means that our implementation was not far off from the participants at the conference in spite of not using Text CNN or other complex techniques.

Judging by the f1 score of the other participants, this problem was always going to be a challenging one not netting a large result. The problem in question was fairly complex.
Data imbalance severely impacted the model. However, choosing f1 micro mitigated the impact it had on the model. The dataset was also very small and it can be argued that a larger dataset would have given a better result. Assuming we had more hardware capabilities, we could have implemented text CNN which would propel us to an F1 score close to that of the winner's. Another of our model's limitations could be that it might output the majority class incorrectly. Some of the minor classes are not predicted correctly at all. Many of the minority classes had a f1 score of 0. This is a major flaw. Also, in this particular task, the labels themselves are highly technical which makes artificial data generation a challenging affair.

Citations

1. Zhu, Xingyu, et al. YNU-HPCC at SemEval-2021 Task 6: Combining ALBERT and Text-CNN for Persuasion Detection in Texts and Images. 2021. Accessed 17 Oct. 2022
2. Tian, Junfeng, et al. MinD at SemEval-2021 Task 6: Propaganda Detection Using Transfer Learning and Multimodal Fusion. 2021. Accessed 17 Oct. 2022.
3. Rosebrock, Adrian. “Keras: Multiple Inputs and Mixed Data.” PyImageSearch, 4 Feb. 2019, pyimagesearch.com/2019/02/04/keras-multiple-inputs-and-mixed-data/.