COMPSCI5106 Text-as-Data (M)

Assessed Coursework Report

Saad Mohammed Anis (2738059M)

12.03.2023

Q1 – Dataset

(a) The dataset, "Twitter US Airline Sentiment" is a collection of tweets that are aimed at each of the major United States airlines, each labeled with the sentiment portrayed in the tweet (negative, neutral, or positive), the name of the airline the tweet was aiming at (United, Virgin America, etc.), and if the sentiment was negative, the reason for the negativity (Bad Flight, Customer Service Issue, etc.). This dataset was chosen to develop a sentiment analysis classifier. Being a twitter corpus, it has the advantage of being highly informal, diverse, and brief, which comprises a large portion of how people communicate on the internet. Furthermore, since the tweets are aiming specifically at brands, a classifier can be used effectively by businesses, brands, and political parties to analyze their brand sentiment on social media and online forums.

(b) The label to be predicted is the sentiment of the tweet, i.e., the emotion portrayed in the tweet. This can be either be negative, neutral, or positive. No preprocessing was required to create these labels. The input text will be the text of the tweet. The only preprocessing performed was the removal of mentioned usernames (@VirginAmerica, for example), website URLs, and certain punctuation from the tweets.

(c) The unsplit dataset contains 14,640 samples. A random sampling of 10,000 was chosen from the dataset and divided into a 60-20-20 split to form the training, validation, and testing sets respectively as shown in the table below. 16% of the tweets were labeled positive, 21% were labeled neutral, and 63% were labeled negative. There is a slight imbalance in the dataset since most of the samples are labeled negative. This makes sense in context since users might more often tweet to a company to issue a complaint.

	Positive	Neutral	Negative	Total
Training	958	1232	3810	6000
Validation	324	424	1252	2000
Testing	322	426	1252	2000
Total	1604	2082	6314	10000

Q2 – Clustering

(a) Cluster 0: top five tokens: thank, dm, follow, send, great.

- @JetBlue I will. Thank you!
- @AmericanAir thanks
- @united I will. Thanks.

Cluster 1: top five tokens: flight, cancel, flightle, delay, late.

- @united are you telling me that you are now Cancelled Flighting my flight ??
- @USAirways Just Cancelled Flight every flight I have why don't you.
- @SouthwestAir Can I get any help with Cancelled Flighting my flight reservation?

Cluster 2: top five tokens: hour, plane, wait, hold, delay.

- @JetBlue don't I always?
- @SouthwestAir Can't DM you because you don't follow me.
- @USAirways "We can't help you. We don't put people up in hotels when you miss a flight it's against policy."

Cluster 3: top five tokens: booking, problem, flight, reflight, helpful.

- @united My flight was Cancelled Flighted and I'm needing some help reFlight Booking Problems.
- @VirginAmerica And now the flight Flight Booking Problems site is totally down. Folks, what is the problem?
- @united No. The entire problem here is that I was never sent anything via email and only given a Flight Booking Problems number over the phone.

Cluster 4: top five tokens: service, co, customer, fly, help.

- @SouthwestAir I've been on hold with customer service for over an hour. Can you help?!
- @SouthwestAir has the WORST customer service of any airline I've ever flown.
- @AmericanAir hey! Tried calling customer service and was told there's a 2 hour wait. This has been for the past 4 hours. Thanks! You suck!

(b) Based on these results, some connections can be noticed. Although all the five labels have the element of customer support and flights in common, the tone and topics of the tweets vary between the different clusters. Cluster 0 seems to have a very positive tone with 'thank' and 'great' among the tokens. Cluster 1 seems to focus on cancelled flights with example tweets containing the phrase 'cancelled flight'. Cluster 2 seems to be about example tweets about holding calls for long hours. Cluster 3 heavily focuses on customers' booking problems. Cluster 4 specifically mentions the customer service and some tweets complain about the quality of customer service of various carriers.

(c) The confusion matrix mapping the clusters to the labels:

Cluster	Positive	Neutral	Negative	Total
0	749	290	256	1295
1	161	452	1604	2217
2	68	146	1496	1710
3	24	44	182	250
4	602	1150	2776	4528

(d) The confusion matrix reveals some useful information. Every cluster apart from cluster 0 is dominated by negative labels. Only the 0th cluster is dominated by positive labels. Over 70% of the 1st, the 2nd, and the 3rd clusters have negative labels. Over 60% of the labels of the 4th cluster are negative. In every cluster, neutral tweets are neither the majority nor the minority. The 1^{st}, 2^{nd}, 3^{rd}, and the 4^{th} clusters have only 4-13% of positive labels. From this information, it can be revealed there is a strong separation between positive and negative labels, while very little effect on the neutral labels. Primarily, clusters seem to have been divided by topics rather than sentiment, but coincidently clustering most of the positive labels into the 0^{th} cluster.

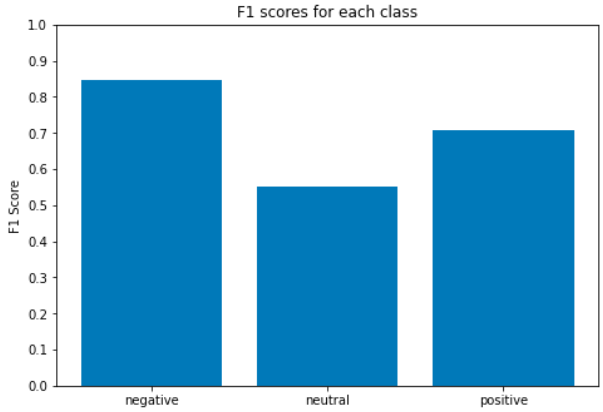
Q3 – Comparing Classifiers

(a) The five classifiers evaluated on the training and validation sets. The best performing classifier on the validation set in terms of the macro F1 is highlighted.

Classifier	Split	Accuracy	Precision	Recall	F1
Dummy Classifier (Most Frequent)	Train	0.635	0.212	0.333	0.259
	Validation	0.626	0.209	0.333	0.257
Dummy Classifier (Stratified)	Train	0.470	0.331	0.331	0.331
	Validation	0.456	0.320	0.320	0.320

Logistic Regression (TF-IDF)	Train	0.869	0.889	0.776	0.819
	Validation	0.762	0.748	0.623	0.659
Logistic Regression (One-Hot)	Train	0.940	0.933	0.909	0.920
	Validation	0.768	0.721	0.694	0.706
SVC (One-Hot)	Train	0.920	0.915	0.869	0.890
	Validation	0.772	0.743	0.657	0.686

The following bar chart shows the individual F1 scores for each class of the one-hot Logistic Regression classifier predicted on the validation dataset.



As expected, every classifier fit the training set better than the validation set. The Dummy Classifiers had F1 scores of 0.257 and 0.320 on the validation datasets, which all the other classifiers easily beat. The SVC model trained with the one-hot vectorized training set resulted in a validation F1 score of 0.686. The regularization parameter was set to its default value of 1.0. The kernel was also set to its default 'rbf' value. The model has a high recall value (0.93) for negative labels, but relatively lower recall values (0.44 and 0.60) for the other labels suggesting that the model was more likely to label tweets as negative. The Logistic Regression model trained on the TF-IDF vectorized data performed worse with a validation F1 score of 0.659. The recall values had an even greater disparity than the SVC model with the recall for negative labels being 0.95 and the recall for neutral and positive being 0.36 and 0.56 respectively. This also shows a high bias for labeling tweets as negative. The best performing model was the Logistic Regression model trained on the one-hot vectorized dataset. Its training F1 score was 0.920 and its validation F1 score was 0.706. This model does not share the negative labeling bias as strongly as the other two classifiers, having a 0.87 recall for negative labels, 0.53 for neutral labels, and 0.68 for positive labels.

(b) A Naïve Bayes classifier 'MultinomialNB' from Scikit Learn was used as the additional classifier. According to Scikit Learn, although fractional vector values may work, discrete integer values are preferred, hence leading to the use of the count vectorized dataset for training.

Accuracy	Precision	Recall	F1
0.744	0.749	0.583	0.620

Although the model performed significantly better than the Dummy Classifiers, it performed worse than all the other models with an F1 score of 62%.

Q4 – Parameter Tuning

The following parameters were chosen in the parameter grid.

- For C, the regularization parameter of the classifier: 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000.
- For 'max_features', the number of tokens to use: None (all), 915, 1830, 3660, 7318.
- For 'sublinear_tf', whether replace term frequency (tf) with 1 + log(tf): True, False.
- For 'tokenizer', which tokenizer to use: None, 'text_pipeline_spacy' (custom-made tokenizer using spaCy).

The search produced the following parameters as the best option.

- C: 1
- max_features: 1830
- sublinear_tf: False
- tokenizer: None

Classifier	Accuracy	Precision	Recall	F1
Baseline	0.762	0.748	0.623	0.659
Tuned	0.798	0.783	0.697	0.730

The value of 1 was chosen for the C regularization. The default value performed the best, which means that the data did not require strong or weak regularization. The max_features parameter, for which the values were ranged from 915 to all the terms, 1830 terms proved to be the best option. It is possible that the ignored tokens contained a lot of spelling errors and were not that useful to the classification. The term frequencies were also not scaled logarithmically since sublinear_tf was False. This log-scaling might not have been helpful since the tweets are very short documents that may not have an extremely high count of the same terms. The default tokenizer performed better than the custom-built spaCy tokenizer. Since the custom tokenizer removed a lot stop words, punctuation, and links, it's possible that it may have lost some useful information in the process.

Q5 – Context Vectors Using BERT

(a) The evaluation metrics on the validation set after using the first context vectors on the Logistic Regression model are tabulated below.

Accuracy	Precision	Recall	F1
0.824	0.787	0.753	0.767

(b) Below is the table with the evaluation metrics on the validation set using the 'roberta-base' model.

Accuracy	Precision	Recall	F1
0.820	0.766	0.766	0.764

(c) The following sets of hyperparameters were chosen.

- 'distilbert-base-uncased' with a learning rate of 1e-6, epoch of 20, and batch size of 8. Since DistilBERT is smaller and faster than BERT, a high number of epochs was chosen.
- 'bert-base-uncased' with an LR of 1e-4, 5 epochs, and 32 batch size. Being the most downloaded model on Hugging Face, this model was given a priority.
- 'cardiffnlp/twitter-roberta-base-sentiment' of LR 1e-5, epoch 3, and batch size 16. This model was already finetuned for sentiment analysis with TweetEval.

The learning rates, batch sizes, and epochs were finalized through a manual trial-and-error process.

Model	L. Rate	Epoch	Batch	Acc.	Prec.	Recall	F1

roberta-base	1e-4	1	16	0.820	0.766	0.766	0.764
distilbert-base-uncased	1e-6	20	8	0.831	0.784	0.786	0.785
bert-base-uncased	1e-4	5	32	0.843	0.802	0.798	0.798
cardiffnlp/twitter-roberta-base-sentiment	1e-5	3	16	0.867	0.832	0.835	0.831

(d) The first approach vectorized the text using a base model and used the vector of the start token in a Logistic Regression model to perform the classification. The second approach fine-tuned the existing base model with the new training data, thereby improving the existing model's performance on the current task. Hence, all the end-to-end Trainer models performed better than the pipeline model. In terms of inputs, both the Trainer model and the Logistic Regression models used the first context vectors since BERT classification tasks only use the [CLS] vector. The difference in scores may be due to the classification models itself, where the hyperparameters of the Trainer model performed better. Furthermore, Logistic Regression may have not be the apt choice for this problem since Logistic Regression assumes a log-linear relationship between the features and the output.

Q6 – Conclusions And Future Work

(a) The best performing model was the roBERTa-base model which was finetuned for sentiment analysis on Twitter data. This model resulted in an F1 score of 0.831 on the validation dataset. On the test set, it performed slightly worse with an F1 of 0.820.

Accuracy	Precision	Recall	F1
0.862	0.822	0.820	0.820

Prediction\Actual	Negative	Neutral	Positive
Negative	1158	98	17
Neutral	66	228	28
Positive	28	40	277

(b) The test results were used to manually examine a random sample of the misclassified text. For the data predicted as negative while it the actual was neutral, the examples show that some of the data is actually negative (example: *"@AmericanAir you wont allow calls? My husband has a ticket but it looks like all the seats are taken? I cant even call."*) or passive aggressive (example: *"@united I'll be impressed if I actually get a response!"*). Similarly, tweets that were predicted negative but labeled positive weren't always a straightforward misclassification. Some of these examples included sarcasm which seemed to be labeled incorrectly. (Example: *"@AmericanAir Yes, thank you. Just not how I wanted to start my vacation!"* and *"@united They finally gave in a let him on. After they threatened to send him back to Vegas on coach. Thnx."*)
Some of the tweets that were predicted as neutral but were in fact negative were not straightforward. Some tweets didn't make sense without context and seemed to be labeled incorrectly as negative (examples: *"@AmericanAir Close down"* and *"@united GRK13575M is the file reference"*). For some positive tweets that were predicted neutral, the classifier seemed to be performing correctly while the labeling seemed incorrect (Examples: *"@SouthwestAir thanks do yall expect to be operational tomorrow out of*

Nashville?" and *"@united Hmmm...seems like this could be something to be changed to be more #flyerfriendly."*).

In terms of some negative tweets that were predicted positive, the classifier again seemed to be performing well, while the actual labels seemed wrong. (Examples: *"@JetBlue had a potentially stressful situation in reFlight Booking Problems a flight which she diffused and helped make awesome!"* and *"@united Yeah, bag is on the way. As per usual. I'm actually getting used to getting it delivered to me, its kind of nice in a sense."*) Similarly, for some neutral tweets that were predicted positive, the classifier seemed to be performing as intended while the labels were incorrect. (Examples: *"@SouthwestAir continues to prove to be the best airlines"* and *"@SouthwestAir An Oscar-worthy entrance into LA."*)

These observations reveal although the classifier misclassified some of the tweets, a lot of the tweets were still classified correctly while the true labels were wrong, implying that the classifier is possibly better than what the evaluation metrics present. The classifier can correctly classify the negative and positive tweets with a 90% and 80% precision respectively. It performs a little worse on neutral tweets which is to be expected, with a precision of 76%. According to the confusion matrix, positive tweets were very rarely classified as negative only 7% of the time. Similarly, negative tweets were also rarely classified as positive, only 2% of the time. On the other hand, only 66% of the neutral tweets were classified as neutral correctly as opposed to 93% for negative and 84% for positive.

There could be several reasons for these errors. Firstly, as mentioned earlier, it could be due to mislabeled data (noisy data). Secondly, it could be due to ambiguity, where tweets cannot really be classified without additional context. For example, *"@SouthwestAir honesty should always be the policy"* without context seems like a neutral tweet, but it could very easily have been a negative reply to the airlines about some kind presumed dishonesty.

(c) By observing the evaluation metrics and the mislabeling problems with the dataset, we can conclude that these results are good enough to be used as a sentiment analysis classifier. False positives and false negatives occur when tweets are misclassified. From (a) and (b), it can be observed that most positive and negative tweets are classified correctly, while a lot more of the neutral tweets are classified incorrectly. This can cause some issues during negative tweet filtering, where neutral tweets may also be flagged. But accidentally grouping neutral tweets as positive may not be as problematic.

(d) Sentiment classification tools can be misused by organizations to censor negative sentiments towards their brand. Justified negative sentiment might also be marked as negative and might result in limited reach of a user's tweet.

(e) The classification could be improved by taking some of the following steps:

- Increasing the amount and variety of data used for training.
- Using better tokenization techniques for traditional classifiers.
- Performing extensive hyperparameter search for BERT-based classifiers.

(f) The entire coursework took approximately 30 hours, around the anticipated time stated in the coursework specification.

Q7 – Research Paper Report

The paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" published by Devlin et al. in 2018 introduces BERT, a pre-trained language model based on transformers. The model uses bidirectional representations of text during its pre-training process, which can then be fine-tuned on specific tasks to create state-of-the-art models for various NLP-related tasks. The authors present a drawback of existing unidirectional language models which limits the self-attention to only using the previous tokens for a left-to-right architecture and vice-versa for a right-to-left architecture. This limitation results in poor performances in sentence-level tasks and question-answering tasks, where bidirectional knowledge is essential. Existing models that utilize pre-training include ELMo by Peters et al. which uses a feature-based approach that obtains more features from the pre-trained representations. ELMo performs its feature extraction from a left-to-right and right-to-left model, where the representations are concatenations of the individual representations from the two models. The other existing approach that BERT is compared to in this paper is OpenAI's GPT model, which is unidirectional and uses the fine-tuning approach. The BERT model introduces bidirectional pre-training. It also demonstrates that these pre-trained models can be easily fine-tuned to create different models for different NLP tasks. BERT uses the Transformer encoder based on the implementation by Vaswani et al. to create two versions of BERT: $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$. To implement bidirectional pre-training for BERT, two unsupervised tasks are performed: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). MLM randomly masks or replaces some words in the text and trains the model to predict the actual word based on its context. The second segment of pre-training, NSP, trains the model to predict whether two sentences are connected. Finally, the pre-training is performed on a large corpus of data including English Wikipedia and BooksCorpus by Zhu et al. Once the pre-trained model is built, fine-tuning it on specific tasks is a relatively quick process. Both models $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$ performed significantly better than the existing state-of-the-art models including ELMo and GPT. For the MNLI task of the GLUE benchmark, BERT improved the accuracy by 4.6%. $\text{BERT}_{\text{LARGE}}$ overshot GPT's score in the GLUE benchmark by 8 points, while $\text{BERT}_{\text{LARGE}}$ performed significantly better than $\text{BERT}_{\text{BASE}}$ across every task. In the SQuAD question-answering tasks, BERT demonstrated F1 scores significantly higher than other models. Finally, for the SWAG sentence-pair completion tasks, $\text{BERT}_{\text{LARGE}}$ beat ELMo by 27.1% and GPT by 8.3%. To test how well each segment of BERT performed, ablation experiments were performed where a model was trained without NSP, while another only used left-to-right representations. The significant loss of performance without NSP and the MLM proved the importance of the bidirectional pre-training tasks. Although computationally expensive and large, BERT has successfully achieved state-of-the-art results on a multitude of NLP tasks. Its bidirectional pre-training has allowed BERT to understand the context within and between sentences.