Sentiment Analysis is Challenging

An Evaluation of Model Performance on Person SenTiment

Authors

Jerry Kang - 113181366
Joseph Hess - 107683340
Paul Kogan - 112813501

Overview

The overall objective of this project is to determine whether our current language models are adequate for complex sentiment analysis. We are specifically using three different types of models to test this: Deep Averaging Network (DAN), Masked Language Model (MLM), and Large Language Model (LLM).

The difficulty of the task lies in our dataset. We are using a special dataset called PerSent that is incredibly difficult to perform sentiment analysis on. The reason is that the dataset includes a target subject for every document and the dataset gives us the sentiment towards that subject. However, the author's overall sentiment does not necessarily equal the sentiment towards the target subject. Each sentence could have a negative sentiment towards the subject, but the overall sentiment could be positive. The three language models we're using are not specifically trained for analyzing sentiment towards a specific subject.

We want to test if any of the three types of models are adequate for doing more complicated sentiment analysis. We are also interested in doing hyperparameter tuning to test for the maximum performance level of these models for complex sentiment analysis. We will be evaluating our models on PerSent's test datasets and computing the precision, recall, and f1 score performance metrics. We went with f1 score because it is the most standardized and reliable metric for model performance.

As expected, the models listed in order of performance from worst to best are: DAN, MLM (DistilBert), and LLM (GPT3). The DAN had an overall f1 score of around 0.24, the DistilBert was around 0.32, and GPT3 was around 0.59 for the Random test dataset.

Ideas

Idea 1: Build a model from scratch

In this experiment, we built a DAN from scratch and attempted to see how well it performed on the task. We did use pretrained, frozen GLoVe embeddings, but the hidden layers of the DAN were trained entirely on the PerSenT dataset.

The nature of the PerSenT dataset makes it unlikely that the DAN will perform very well, since the author's overall sentiment in the article need not match their sentiment towards the subject of the article, as the paper discusses. However, building a DAN will give a good baseline to compare against better models, to see to what extent they are able to differentiate between the overall sentiment of the article as indicated by the DAN and the sentiment towards the main subject of the article.

Idea 2: Fine-tune a pretrained model

In this experiment, we used a pre-trained DistilBert model from hugging face's transformers library. We also did a little bit of fine tuning and hyperparameter tuning to improve performance. The fine tuning included changing the amount of layers that we froze, and changing the number of training epoch iterations and batch size.

Idea 3: A Few-Shot learning approach

A few-shot learning approach with a LLM, such as the one used in this task, aims to leverage the model's pre-existing knowledge and adapt it to perform a specific task with just a few examples, as opposed to traditional machine learning methods that require large amounts of labeled data to train. This saves time and resources as it does not require extensive fine-tuning. This approach involves providing the LLM with a few labeled examples from the training dataset, followed by the target text and main entity. The examples are formatted as a prompt, which is then fed to the model. The model then (essentially) learns from the examples and applies it to the target text.

Experimental Setup

Experiment 1

For this experiment, we built DAN models on top of pretrained GLoVe word embeddings. The DAN, or Deep Averaging Network, is a deep neural network that trades off the ability to examine the arrangement of the individual tokens in an input for increased efficiency in training and inference. The DAN first averages together its embeddings for all of the tokens in the input, then passes this average through some number of activated hidden layers and a final classification layer. In the PerSenT dataset, the challenge comes precisely from those arrangements, so we do not expect a DAN to perform well. However, we believe that the experiment is still worth doing, since it will allow us to see if the more powerful models are really making good use of the extra information they have available to them.

We trained three separate models with varying hyperparameters to see if any could perform well on the PerSenT dataset. Each model was trained for 50 epochs with a batch size of 16 using the AdamW optimizer. We chose the 100 dimensional word embeddings from the GloVe website’s 6B token pre-trained embedding file, as the larger sets were likely too large to work with effectively on available hardware. These embeddings have a 400k vocabulary size. We gave each DAN model a dropout layer used only in training, with a dropout probability of 0.2. The hidden layers were each 100-dimensional. We varied the number of hidden layers and the learning rate to explore the effects of those hyperparameters on the DAN. In particular, we trained model with 4 hidden layers and a learning rate of 3e-5, 8 hidden layers and a learning rate of 3e-5, and 4 hidden layers and a learning rate of 3e-2.

We used PerSenT’s full training dataset to train the DANs, with the dev dataset used for evaluation, and the fixed and random test datasets used for final testing. We used each dataset’s “Document” and “True Sentiment” columns for our models. The “Document” column contains the full, unmasked document in text form, and the “True Sentiment” column is a text label of “Negative,” “Neutral,” or “Positive.” We tokenized and embedded these datasets to work with a DAN model, and used a dictionary to convert the sentiment label to categories for classification. The training, evaluation, fixed test and random test datasets have 3355, 578, 827, and 579 entries, respectively.

We evaluated the models automatically using accuracy, and macro precision, recall, and f1 scores. We chose accuracy as it is a standard metric, and used precision, recall, and f1 score to better ensure that the model is not finding some pathological pattern in the data that inflates its accuracy without providing any usefulness. We chose macro precision, recall, and f1 as these are the analogues of these metrics in a multiclass situation, and can be more effective than micro scores.

We also looked at the model’s loss value over each epoch, to help understand how the model was changing throughout its training.

Experiment 2

We used the DistilBert uncased pre-trained model from hugging face for this experiment setup. DistilBert is a transformer based model and is a much smaller version of the popular model Bert. Using the actual Bert model would have been much more difficult due to Google Colab's space limitations. Therefore, we went with the DistilBert uncased. The tradeoff for the smaller model size was a couple percent worth of accuracy. In our case, this tradeoff is inevitable because Bert's large model size would have been impractical to work with.

We are using the PerSent dataset to train and evaluate our model. Each training sample in the dataset contains the document, the main subject, and the individual sentiment of each sentence in the document. Note however, that we are not using the individual sentiment data, only the overall sentiment. The training set, validation set, fixed test set, and random test set are length 3355, 578, 827, 579 respectively.

We used automatic evaluation to compute our performance metrics by testing the model on PerSent's test dataset and evaluating the precision, recall, and f1 score. We went with these metrics because they are standard for machine learning evaluation and are very easy to compute with scikit-learn's built in functions for precision, recall, and f1.

Experiment 3

In this experiment, we used the OpenAI API to perform sentiment classification by leveraging a few-shot learning approach to demonstrate how to leverage pretrained models like GPT3 for specific tasks without requiring extensive fine-tuning or additional training. We used the 'text-davinci-003' engine because it is the most powerful available. To utilize the few-shot learning approach, we first crafted a prompt that included a few labeled examples from the training and development datasets. Each example contained the masked text of an article, its title, the main entity (subject), and the sentiment label (Neutral, Positive, or Negative). These examples were provided to the model to help it understand the context and task requirements. We then appended the target text (i.e., the article we wanted to classify) and its metadata (title and main subject) to the prompt, and formed a query to the model to determine the sentiment towards the main subject in the target text. This formatted prompt was sent to the OpenAI API, which returned a completion containing the predicted sentiment as a single word.

After performing this process for all the samples in the test datasets, we obtained the predicted sentiment labels and compared them with the true labels from the dataset. Various performance metrics were calculated to evaluate the effectiveness of the model. We chose to calculate accuracy because it is a simple, straightforward metric that calculates the proportion of correct predictions out of the total predictions made; it gives a general sense of the model's performance. It should be noted, however, that accuracy might not be the best metric in cases where the dataset is imbalanced, as it can be misleading. Since our task is a multi-class classification problem, we also calculated the ROC-AUC score. The Receiver Operating Characteristic (ROC) curve is a plot that displays the true positive rate (sensitivity) against the false positive rate (1-specificity) at various classification thresholds. The Area Under the Curve (AUC) is a single scalar value that measures the overall performance of the model across all possible classification thresholds; a higher ROC-AUC score indicates better performance. We also found the F1 score, which is the harmonic mean of precision and recall–it provides a single value indicating the trade-off between the two–and is particularly useful for imbalanced datasets. Finally, we used a Confusion Matrix to understand the types of errors the model is making. This matrix provides a detailed view of the results, showing the number of true positives, true negatives, false positives, and false negatives for each class; it can help us identify patterns in misclassifications. By using this combination of metrics, we obtain a comprehensive assessment of the model's performance on the task.

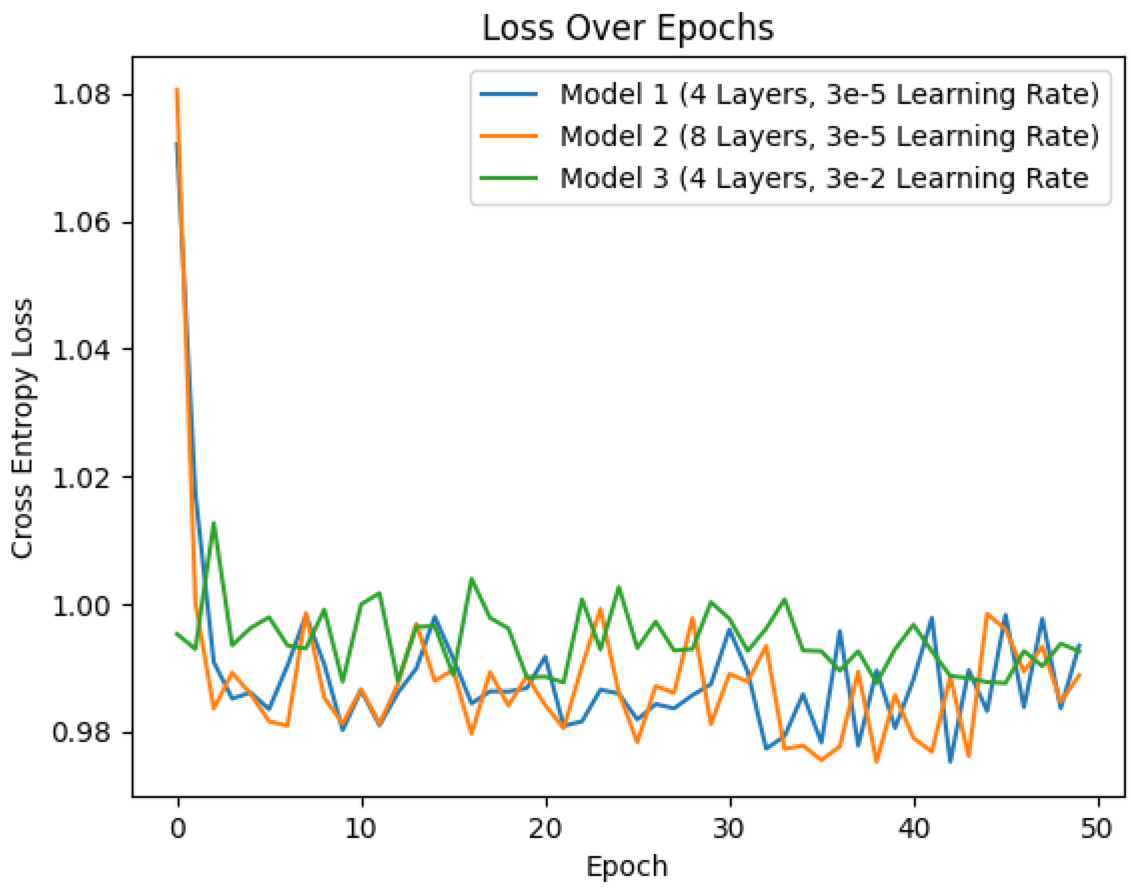
Results

Experiment 1

Best Results for DAN (precision, recall, and f1 are macro scores):

Test	Model	Accuracy	Precision	Recall	F1
Fixed	Model 2	0.44	0.155	0.346	0.211
Random	Model 3	0.51	0.198	0.365	0.250

Before evaluating these test metrics, we first looked at the loss in each model for each epoch over the course of training.



We first looked at the loss (Cross Entropy Loss, for these models) over each epoch of training for each of the three models. We see that all three models rapidly converge to a similar value. The first two models, which have 4 and 8 hidden layers, respectively, and both have a learning rate of 3e-5, perform almost identically. The third model, with 4 hidden layers and a learning rate of 3e-2, exhibits slightly worse loss than the other two throughout most of its training.
We did not expect to see such rapid convergence in our models. This likely occurred because the pre-trained GloVe embeddings we are using are doing much of the heavy lifting, alleviating the need for the DAN layers to find smaller variations to amplify.
Next, we tested the models against the test sets.

For the Fixed Test:

Model	Hidden Layers	Learning Rate	Accuracy	Precision	Recall	F1
Model 1	4	3e-5	0.44	0.152	0.340	0.207

Model 2	8	3e-5	0.44	0.155	0.346	0.211
Model 3	4	3e-2	0.44	0.153	0.343	0.207

For the Random Test:

Model	Hidden Layers	Learning Rate	Accuracy	Precision	Recall	F1
Model 1	4	3e-5	0.51	0.192	0.374	0.249
Model 2	8	3e-5	0.51	0.180	0.360	0.237
Model 3	4	3e-2	0.51	0.198	0.365	0.250

Overall, the DAN models conformed to our expectations that a DAN would not be able to exhibit good performance on the PerSenT dataset. We were surprised by how quickly each model converged, but the fact that they did converge so quickly, but were still not able to achieve acceptable performance, indicates that they are in fact not suitable for the task at hand.
We note that the models perform better overall on the random test dataset than they do on the fixed test dataset. This is likely due to the fact that the random test dataset more closely resembles the training dataset than the fixed test dataset.
Notably, despite exhibiting slightly higher loss than the other two models throughout most of its training, Model 3 managed to outperform them slightly on the random test.

Experiment 2

	Precision	Recall	F1
Fixed Test	0.21505	0.3292	0.23745
Random Test	0.326275	0.36005	0.3126

From the results alone, we can see that the DistilBert model is no better than random choice for sentiment analysis with an average f1 score of .3126 for random test. We can also see that the performance on the fixed test was much worse than the random test. This is likely because those documents were focused on a small number of specific main subjects. We will also see later that DistilBert is very bad at understanding the subject sentiment vs overall sentiment. Another interesting thing is that the recall for the fixed test was close to a random choice performance (.33). One of the reasons for this unimpressive performance is because we are using the distilled version of Bert, which has weaker capabilities. However, we expected a low performance score from these models because we know the dataset is supposed to be incredibly difficult to analyze. Another failure of DistilBert was that the training f1 score was much higher than the test score, averaging around .6. This much better performance on the training and validation sets could be a sign of overfitting to the training data. It would explain why it performed much worse on the test set than expected.

Experiment 3

Test	Accuracy	F1	Confusion Matrix	ROC-AUC
Random Test	0.60	0.5911	$\begin{matrix} 4 & 2 & 0 \\ 1 & 17 & 5 \\ 1 & 11 & 9 \end{matrix}$	0.6503
Frequent Test	0.42	0.3337	$\begin{matrix} 6 & 6 & 1 \\ 5 & 15 & 0 \\ 3 & 14 & 0 \end{matrix}$	0.5434

For the Random Test data, the model correctly classified 60% of the samples (4 Negative, 17 Neutral, and 9 Positive). The F1 score is 0.5911 and ROC-AUC score is 0.6503, indicating moderate performance in classifying sentiment. For the Frequent Test data, the model correctly classified 42% of the samples (6 Negative, 15 Neutral, and 0 Positive). The F1 score is 0.3337 and ROC-AUC score is 0.5434, indicating worse performance in classifying sentiment. The model seems to have difficulty classifying negative and positive sentiments correctly, as there are relatively higher misclassifications between these two classes.

Overall, the model performs moderately well on the Random Test data, but its performance drops when dealing with the Frequent Test data, which contains multiple articles about a small number of popular entities. This suggests that the few-shot learning approach might need more examples or a more specific prompt to better handle sentiment classification for popular entities.

Analysis and Discussion

Chosen Model: DistilBert

Hypothesis 1

The model should fail on a document where the sentiment towards the main subject is neutral but the predicted sentiment is negative. This happens when the subject is saying something that is of negative sentiment, but the negativity is not directed towards the subject.

Example Input

Since early 2009, said Norm Miller vice president of analytics for the CoStar Group, some 2 300 auto dealerships have closed around the country as new car sales plunged more than 40 percent and the government after taking ownership stakes in General Motors and Chrysler forced them to end long standing franchise contracts. The closings put 70 million square feet of buildings and land on the market according to CoStar, a commercial real estate research company based in Bethesda Md. But in the last five quarters Miller said 649 of those shuttered dealerships found new owners and were put to new uses including the sale of Whitehall Ford here for $1.1 million. In the first quarter of this year 152 dealerships were sold for a combined total of $300 million he said. Prices ranged from $500 000 to $9 million Miller said though most sales were for $1 million to $3 million. The numbers Miller said represent only the first wave of real estate investment in a market segment that was blasted by the recession. "The good news is that there are steady sales. And there are some noticeable trends " he said. "Schools are buying dealerships and converting them. Lumber Yards are buying dealerships. Some are being turned into retail centers.

Output and Explanation

The model predicted this as a negative sentiment, but it is actually neutral because the main subject is Norm Miller. The model couldn’t identify this targeted sentiment and as a result, predicted wrong. The hypothesis was correct.

Modification

Removed all mentions of Norm Miller and replaced it with the word he. The model still predicted the new document as negative, which is now correct given that Norm Miller is no longer the targeted subject. This confirms that the model behaves as we expected and is unable to understand subject sentiment.

Hypothesis 2

The model should incorrectly predict a document where the main subject is the one the sentiment is being directed towards. With this sort of document, the problem becomes no different than normal sentiment analysis so the model should predict correctly.

Example Input

All this makes it easy to forget that one potential candidate is missing: the one time presumptive favorite Victoria Reggie Kennedy. She has definitively ruled out any temporary appointment and there is nothing to suggest that she wants to run either. That is too bad because she is the natural inheritor of the Kennedy mantle. For such a long-awaited race the assumed stampede of candidates has been slow to materialize. Joe Kennedy is out. Marty Meehan is out. Mike Capuano is on the fence. No one is ever going to replace Teddy Kennedy. The flood of obituaries reminded us eloquently that he devoted decades to becoming the master of the Senate. He often said that he came into his own as a senator only upon the death of his presidential ambitions. We can forget the notion that we are electing someone to fill his shoes because that is impossible.

Output and Explanation

The model correctly predicted this as a positive sentiment.even though there were many words in the document that hinted at a negative corpus. These include mentions of obituaries, Victoria not running for the office position, and death.

Hypothesis 3

The model should incorrectly predict a document where the overall sentiment is positive but the main content contains negative sentiment wording.

Example Input

In his first full-fledged session with the media since his life fell apart Woods entered the interview room with a smile on his face and stopped to hug one of the green-jacketed club members Ron Townsend. Woods again took full blame for his personal failings but stopped short of providing many new details. He wouldn't say why he entered rehab for

45 days nor would he go into specifics about his infamous November night car crash other than to say it took five stitches to close a lip wound. He said his wife Elin would not be at Augusta. His personal life fell apart after revelations that he had multiple extramarital affairs during their 5 1/2-year marriage. Woods thanked his fellow golfers for the support he's received since announcing his return to the PGA Tour and said he was pleasantly surprised how well the fans treated him during a practice round Monday. The outing was his first before a gallery since the sex scandal made him a tawdry tabloid fixture. He even flashed a bit of uncharacteristic charm, stopping to sign autographs -- something he rarely does -- while heading to the practice range to get in a few extra swings. "The encouragement I got blew me away, " he said. "It really did. The people here over the years I know they've been extremely respectful. But today is just something that touched my heart pretty good."

Output and Explanation

The model incorrectly predicted this as a negative sentiment document because there were so many mentions of negative things that happened to the main subject Tiger Woods. However, the sentiment isn't directed towards Woods as the model clearly fails to understand. The document is actually a positive overall sentiment towards Woods.

Code

DAN drive link:
https://drive.google.com/drive/folders/1T7jhZWmFjonZib8TdGnGJjq1NQteaXu7

Code structured off of: CSE 354 HW2, CSE 354 HW3
get_performance_metrics() function modified
Notebook cells should be run sequentially to create/test the models.
Code uses GloVe embeddings: https://nlp.stanford.edu/data/glove.6B.zip
PerSenT Dataset: https://stonybrooknlp.github.io/PerSenT/
Code created/run in Google Colab. Python: 3.10.11, Numpy: 1.22.4, Pandas 1.5.3, Matplotlib: 3.7.1, PyTorch: 2.0.0+cu118, NLTK: 3.8.1, TQDM: 4.65.0, sklearn: 1.2.2

MLM drive link:
https://drive.google.com/drive/u/0/folders/1FGBZqiwxF_YpRnGl-hRM1Hz_zPDrpLwy
Original source: CSE354 HW 3
DistillBert, DatasetLoader, Trainer, Tester classes modified
Software requirements: Python 3.10.11, PyTorch 2.0.0+cu118, pandas 1.5.3, scikit-learn 1.2.2, transformers 4.28.1, cuda_11.8.r11.8/compiler.31833905_0

LLM drive link:
https://drive.google.com/drive/folders/1Srfbj9K9qIY-peNUOxySIl2andXQtLhK
This code is my own work; I have not modified anyone else's files/functions. The functions I wrote are: get_examples(), process_dataframe(), and calculate_metrics(). The model was prompted using the PerSenT datasets. The code works on the latest Python 3 versions (>3.7).

Learning Outcomes

Overall, this project helped us cement our understanding of the various models we tested, as well as their strengths and weaknesses. We expected the models' performance to rank roughly according to complexity, and were able to confirm that with our experiments.

The DAN is only able to capture the general sentiment of an article, and that hamstrings its ability to perform a task that requires looking at a specific entity within the text.

The MLM we used, DistilBERT, exhibited better performance than the DAN, although the difference was not as great as we had initially thought it would be. The gap may have been larger had we used the full BERT, however.

The LLM model we used, OpenAI's text-davinci-003, performed the best of all our experiments, as expected. While this model can leverage its few-shot learning capability to perform sentiment classification tasks without requiring fine-tuning, its performance on sentiment classification tasks depends on the quality and clarity of the input data, the complexity of the sentiment expressed, and the model's exposure to relevant contexts during training. While GPT-3 can handle many sentiment classification tasks, it may face challenges in certain scenarios, such as dealing with ambiguous, sarcastic, or domain-specific inputs.

Contributions

Joseph Hess - DAN Code + Write-up
Jerry Kang - MLM Code + Write-up
Paul Kogan - LLM Code + Write-up