CSE256 Assignment2 Chowdhury, Ranak Roy, A53317421

October 30, 2019

1 Supervised: Improving the Basic Classifier

In this section, the implementation, performance and the feature engineering of the supervised learning model have been discussed.

1.1 Implementation

The supervised task is implemented using Logistic Regression. The reviews and their corresponding label, which can be either positive or negative is read from the training and the validation sets. Since there are only two classes, this is a binary classification problem. The goal is to learn a simple decision boundary on the training set than can classify the reviews according to their sentiment. All the data points on one side of the decision boundary will be classified as positive while those on the other side will be classified as negative. Logistic Regression is used when the dependent target variable is categorical.

1.2 Performance Analysis

Table 1: Training, validation and test set accuracy for different feature engineering

Data Portion Used	**No Feature Engineering**	**TF-IDF**	**Hyperparameter Tuning**	**TF-IDF + Hyperparameter Tuning**
Training Set	0.98210	0.90201	0.99935	0.97447
Validation/Dev Set	0.77729	0.76638	0.78074	0.80131
Test Set	0.77936	0.78304	0.77979	0.78497

Table 1 highlights the performance of the logistic regression classifier on the training, validation and unlabeled data set for different methods of feature engineering used. The first column implements the classifier without any feature engineering. This classifier uses the default hyperparameters for regularization and optimization from *sklearn* and uses the basic *CountVectorizer* to create the features. The *CountVectorizer* provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary. An encoded vector is returned with a length of the entire vocabulary and an integer count for the number of times each word appeared in the document. *CountVectorizer* implements a simple and effective model for thinking about text documents in machine learning is called the Bag-of-Words Model, or BoW. The model is simple in that it throws away all of the order information in the words and focuses on the occurrence of words in a document. This can be done by assigning each word a unique number. Then any document we see can be encoded as a fixed-length vector with the length of the vocabulary of known words. The value in each position in the vector could be filled with a count or frequency of each word in the encoded document. This is the bag of words model, where we are only concerned with encoding schemes that represent what words are present or the degree to which they are present in encoded documents without any information about order.This gives the poorest performance on the test set as is expected because the neither good features nor well tuned hyperparameters have been chosen. When we implement both the Feature Engineering methods (TF-IDF and Hyperparameter Tuning), the performance on both the Validation and the Test set improves. We present more discussion about our Feature Engineering Methods in the following subsections.

1.3 Feature Engineering 1: TF-IDF

This is an acronym that stands for "Term Frequency–Inverse Document" Frequency which are the components of the resulting scores assigned to each word.

- **Term Frequency:** This summarizes how often a given word appears within a document.
- **Inverse Document Frequency:** This downscales words that appear a lot across documents.

TF-IDF are word frequency scores that try to highlight words that are more interesting, e.g. frequent in a document but not across documents. The TfidfVectorizer will tokenize documents, learn the vocabulary and inverse document frequency weightings, and allow us to encode new documents. Column 2 of Table 1 shows that the performance on the test set improves after we perform TF-IDF.

1.4 Feature Engineering 2: Hyperparameter Tuning

As the second Feature Engineering method, I tuned the hyperparameters of the Logistic Regression classifier. I performed a grid search on the three hyperparameters, namely *C*, *tol*, and *solver*. The explanation of each of these hyperparameters are provided below.

- **C:** Refers to the inverse of regularization strength, where smaller values specify stronger regularization. I used $C = 7$.
- **tol:** Refers to the tolerance for stopping criteria. I used $tol = 0.1$.
- **solver:** The possible options for the solver are lbfgs, sag, newton-cg, and saga. They refer to the algorithms that are used in the optimization problem. I used the newton-cg solver. The newton-cg method is better than the typical gradient descent because it uses the quadratic approximation (i.e. first and second partial derivatives). The geometric interpretation of Newton's method is that at each iteration one approximates $f(x)$ by a quadratic function around a point, and then takes a step towards the maximum/minimum of that quadratic function (in higher dimensions, this may also be a saddle point).

The second column of Table 1 shows that Hyperparameter Tuning improves performance on the training, validation and the unlabeled set.

2 Semi-supervised (Expanding Labeled Data)

In this section, the implementation, performance analysis, effect of variation of labeled and unlabeled data, data labeling strategy, comparison between supervised and semi-supervised setting, feature analysis, and error analysis have been discussed.

2.1 Implementation

Logistic Regression classifier with the same hyperparameter setting and TF-IDF used for the supervised setting are used for the semi-supervised setting as well.

2.2 Performance

Table 2: Training, validation and test set accuracy for the semi-supervised setup

Data Portion	Accuracy
Training Set	0.99505
Validation/Dev Set	0.76201
Test Set	0.78792

Table 2 illustrates the performance of the semi-supervised model on the training, validation and test set of the data. As evident from this table, the classifier trained under semi-supervised setting surpasses the performance of that trained under supervised setting on the training and test data. The unlabeled dataset is much larger compared to the training set. So the classifier leverages this large collection of unlabeled reviews to better learn the parameters. So it is understandable that the performance of the semi-supervised classifier is better than that of the supervised one.

2.3 Labeling Strategy

The trained classifier is first used to predict the labels of the unlabeled data. Then, I used the *predictproba* function of *sklearn* to estimate the predicted probability of each instance to belong to a particular class. I used only those instances which have shown a probability of greater than 90% to belong to a particular class. So I used only those instance which have been predicted by the model with a high degree of confidence to retrain my classifier. Then I used those instances and their corresponding predicted labels to merge them with the training data and labels. I then used this updated training set to retrain my classifier. I started by predicting the performance of the classifier on the first 10% of the unlabeled data and used the most confident predictions out of those 10% data to be merged with the training set. I continued this process of predicting, merging and retraining until I utilized whole of the unlabeled data.

2.4 Labeled Data Size Analysis

Table 3: Training and validation accuracy with variation of the amount of labeled data

Fraction of labeled Data	**Training Set Accuracy**	**Validation Set Accuracy**
10%	0.74182	0.69432
20%	0.79550	0.74236
30%	0.82300	0.71834
40%	0.85552	0.75109
50%	0.87669	0.75109
60%	0.89917	0.74672
70%	0.91684	0.76201
80%	0.93955	0.76855
90%	0.95417	0.78384
100%	0.97447	0.80131

Table 3 shows how the performance of the classifier on the training and the validation set varies with the changing fraction of the labeled training data used. As illustrated by the table, the performance of the classifier on the training and the validation set improves as we increase the fraction of the labeled training data. The model becomes better at approximating the true parameters of the task, as it is fed with more data.

2.5 Features Analysis

The features used in this model are the words. So we show a list of most representative words that belong to each of the two classes as learned by the logistic regression classifier in the supervised and the semi-supervised setting.

List of top-8 and bottom-8 words learned by the classifier in the supervised setting:

- **Top 8 words (Positive Class):** friendly, love, awesome, best, delicious, excellent, amazing, great
- **Bottom 8 words (Negative Class):** worst, horrible, terrible, not, disappointing, rude, average, disappointed

List of top-8 and bottom-8 words learned by the classifier in the semi-supervised setting:

- **Top 8 words (Positive Class):** : favorite, awesome, excellent, delicious, best, love, amazing, great
- **Bottom 8 words (Negative Class):** not, worst, horrible, terrible, disappointed, went, asked, average

Most of the important features representative of each class are same in both the supervised and the semi-supervised setting. The model successfully learns to distinguish words with positive connotation from words with negative connotation.

2.6 Comparison Supervised vs Semi-supervised

Table 4 shows how the performance of the classifier on the training, validation and test set varies between the supervised and the semi-supervised setting. As evident from this table, the classifier trained under semi-supervised setting surpasses the performance of that trained under supervised setting on the training and test data. The unlabeled dataset is much larger compared to the training set. So the classifier leverages this large collection of unlabeled

Table 4: Comparison between the Training, validation and test set accuracy for supervised and semi-supervised setup

Data Portion Used	**Supervised**	**Semi-supervised**
Training Set	0.97447	0.99505
Validation/Dev Set	0.80131	0.76201
Test Set	0.78497	0.78792

Table 5: Training and validation accuracy with variation of the amount of unlabeled data

Fraction of unlabeled Data Predicted on	**Fraction of unlabeled data predicted with confidence $> 90\%$ (used for training)**	**Training Set Accuracy**	**Validation Set Accuracy**
0%	0%	0.97447	0.80131
10%	8.29%	0.98141	0.79039
20%	13.16%	0.98630	0.77948
30%	19.23%	0.98904	0.77729
40%	26.40%	0.99090	0.77074
50%	34.31%	0.99220	0.76856
60%	42.5%	0.99316	0.76973
70%	50.68%	0.99401	0.77074
80%	59.13%	0.99449	0.76856
90%	67.66%	0.99483	0.77293
100%	76.34%	0.99505	0.76201

reviews to better learn the parameters. So it is understandable that the performance of the semi-supervised classifier is better than that of the supervised one.

Table 5 shows how the performance of the semi-supervised classifier on the training, validation and test set varies as the fraction of the unlabeled data is changed. The first row shows the performance of the supervised classifier since we did not use any o the unlabeled data. We do not use the whole of the unlabeled data that our classifier predicted on but we only use those fraction of instances from that unlabeled data, which have been predicted with a confidence of greater than 0.9. This ensures that we do not train our model from data whose labels have a low probability of being right. The performance on the training set improves progressively as we increase the fraction of unlabeled data in our training set. But there is no general trend on the performance of the classifier on the validation set.

2.7 Error Analysis

These are examples of some of the reviews which have been misclassified by the semi-supervised classifier:

- If you want your fancy flavored iced coffee, sure go here. But if you want a good version of their namesake-doughnuts, then go elsewhere. Specifically, go south to Fresh
- We decided to try hash house a go go after reading several positive yelp reviews.We were a little put off when we got to the Imperial Palace location at approximately
- Had an awsome late lunch at this bbq restaurant this afternoon. I have a gluten alergy so it was a relief to be able to talk to the owner
- You are guaranteed a headache because of all of the annoying, loud advertisements in your face while you're filling up your car. It's out of control.
- This location is fairly large compared to other's I've been to. I prefer this location over the one in Chandler. There is a lot more to choose from
- The atmosphere, granted I say is pretty nice to be sipping on your glass of white wine under some realistic clouds as your overheard and looking at people pass

- Cheese steak = 5 starsService = 2 starsWaiting time = 0 starsWe really want to love this place, but they're making it very hard for us to do so. We
- My experience at the Stonewall Institute was very positive. I highly disagree with the negative reviews posted below. I found the staff to be very kind, knowledgeable, caring. I

In all the above reviews, we find that none of them contained any of the 8 most representative words that belong to the positive and negative class. So these reviews lacked a strong feature which is a good representation of its corresponding class. This could be a potential reason why the classifier misclassified these examples.

3 Semi-supervised (Designing Better Features)

In this section, I discuss how I came up with better feature designs to improve my classifier and show relevant comparisons.

3.1 Description

In order to utilize the corpus of unlabeled text to learn something about the word semantics, and use it to identify the words that are likely to indicate the same sentiment, I used the Word2Vec model. This model helps to learn better representations of the huge corpus of unlabeled dataset than the bag of words model. In this way, I can transfer the knowledge acquired from unlabeled documents to spread the labels to words that do not even appear in the training data.

3.2 Motivation/Justification

Word embedding is one of the most popular representation of document vocabulary. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc. The objective of this representation is to have words with similar context occupy close spatial positions, so that we can capture the semantic meaning of the words in the vectors.

3.3 Analysis

Table 6: Comparison between the Training, validation and test set accuracy for the BOW and the Word2Vec semi-supervised setup

Data Portion Used	**Bag-of-Words**	**Word2Vec**
Training Set	0.99505	0.98412
Validation/Dev Set	0.76201	0.77397
Test Set	0.78792	0.78879

Table 6 shows how the performance of the classifier on the training, validation and test set varies between the Bag-of-Words and the Word2Vec model under the semi-supervised setting. As evident from this table, the Word2Vec model outperforms the Bag-of-Words model in the training, validation and the test set. This is because the Word2Vec model captures the semantics of the words rather than just counting their occurrences. Therefore, the Word2Vec model learns a better representation of the corpus than the Bag-of-Words model.

4 Kaggle

Username: ranakroychowdhury
Display Name: Ranak Roy Chowdhury
Email Address: rrchowdh@eng.ucsd.edu