Study of Domain-Specificity in Customer Reviews Classifications using Supervised Learning.

Nalinda Kulathunga, Daniel Vrinceanu and Yunjiao Wang.

Department of Mathematics, Texas Southern University, Houston, TX 77004.

1. Abstract

In data science, machine learning tools have been successfully used in forecasting, specially in sentiment analysis such as customer review classification problems. Usually, a model used for a classification problem in a certain company is trained using their own data expecting better accuracy. However, since the sentiment data such as customer reviews have many common characteristics disregard of their source, there is a possibility of training models from different sources without loosing the required accuracy. In this work, we investigate the company(domain)-specificity of customer reviews by using data from three different sources; online shopping, restaurants and movies. We apply pre-processing techniques used in Natural Language Processing (NLP) such as lemmatization and embedding. Results show that there is negligible domain-specificity when models are trained using customer reviews data based on the sources used in this analysis. Furthermore, less complex models like the Multinomial Naive Bayes (MNB) show better accuracy than complex models like Conversational Neural Nets (CNN).

2. Method

Data Sets : We obtained the "Sentiment Labeled Sentence Data Set" from the center of machine learning and Intelligent systems at University of California, Irvine [1]. It consists of approximately 3000 positively or negatively labeled customer reviews from three companies; yelp.com, amazon.com and IMDb.com. In addition, a combined data set was formed by randomly selecting reviews from each of the mentioned data sets.

Pre-processing : For the basic pre-processing, the Natural Language Tool-Kit (NLTK) [2] was used. Figure 01 shows the word counts after removing stop words and punctuations. The *wordnet* dictionary in NLTK was used for the lemmatization. The *CountVectorizer* in Scikit-learn library [3] was used for the vectorization of the sentences. The importance of words towards the corresponding labels was found using *term frequency - inverse document frequency (tf-idf)*. Word embedding was used for the preparation of text data for CNNs.

Model Selection : As the benchmark model, we used the Naive Guesser which classifies the reviews assuming all reviews are positive. Then we used three different models; Multinomial Naive Bayes (MNB), Stochastic Gradient Descent (SGD) and a Convolutional Neural Net (CNN) in the training. Fine-tuning of model's hyper-parameters was done using *GridSearchCV* in scikit-learn [3] for both MNB and SGD. The CNN structure used for this analysis is shown in Table 02.

Investigation of Domain Specificity : Four data sets were used in this part of the analysis to check the domain specificity. Data sets are; *"comb"* - data set with the reviews from all three companies, *"Amaz"* - data set with only amazon.com reviews, *"Yelp"* - data set with only yelp.com reviews, *"IMDb"* - data set with only IMDb.com reviews. The best performing model selected using the whole data set was used for the testing of domain specificity.

3. Quality Assurance (QA)

Table 01: Input vectors before and after cleaning the data.

Index	Raw Input (X')	Processed Input (X)	Label (Y)
0	Your staff spends more time talking to themsel...	your staff spend more time talk to themselves ...	0
1	I've had this for nearly 2 years and it has wo...	ive have this for nearly 2 years and it have w...	1
2	I'm probably one of the few people to ever go ...	im probably one of the few people to ever go t...	0
3	Everything about this product is wrong.First	everything about this product be wrongfirst	0
4	The only thing I wasn't too crazy about was th...	the only thing i wasnt too crazy about be thei...	0
5	Chinese Forgeries Abound!.	chinese forgeries abound	0

Figure 01: Occurrence by word without stop words.

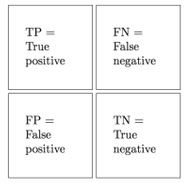

Figure 02: Confusion matrix representation.

$$Precision := \frac{TP}{TP+FP} \qquad Recall := \frac{TP}{TP+FN} \qquad F_\beta = (1+\beta^2)\cdot\frac{\text{precision}\cdot\text{recall}}{(\beta^2\cdot\text{precision})+\text{recall}}$$

3. Results - I (CNN architecture)

CNNs are mostly used in image classifications. However, by applying pre-processing techniques such as one-hot encoding, embedding and padding, text data can be fed in to CNNs and classifications can be performed successfully. In addition to the conventional layers in a CNN such as *convolution* and *pooling* layers, an *embedding* layer should be added as the input layer. This embedding layer can be easily implemented using *TensorFlow keras* [4]. Few drop-out layers were introduced to avoid the possible overfitting. Table 02 shows the CNN architecture used in this work.

Table 02: CNN architecture with embedding as input layer.

Layer (type)	Output Shape	Param #
embedding_4 (Embedding)	(None, 200, 200)	800000
conv1d_1 (Conv1D)	(None, 200, 64)	38464
conv1d_2 (Conv1D)	(None, 200, 32)	6176
flatten_4 (Flatten)	(None, 6400)	0
dropout_1 (Dropout)	(None, 6400)	0
dense_4 (Dense)	(None, 180)	1152180
dropout_2 (Dropout)	(None, 180)	0
dense_5 (Dense)	(None, 1)	181

```
The maximum number of words considered was 200
Embedding dimension: 200
Drop—out rate : 0.2
The activation function used was a "sigmoid"
```

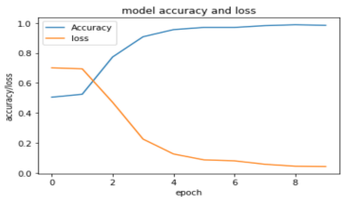

Figure 03: Accuracy and loss during the training of CNN.

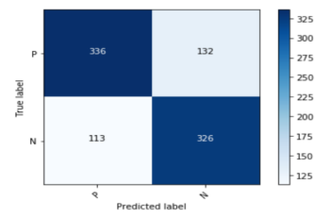

Figure 04: Confusion matrix after testing the CNN.

4. Results - II (SGD Classifier)

In gradient descent algorithms, models are trained by minimizing the Mean Square Error (MSE). Consider the matrix representation of the problem: $\mathbf{Y} = \theta\mathbf{X}$, where $\mathbf{Y}$ represents labels, $\mathbf{X}$ represents features and θ represents the weights. The MSE(θ) and the gradients are defined as;

$$MSE(X,\theta) = \frac{1}{m}\sum_{i=0}^{m}(y_{pred}^{(i)} - y^{(i)})^2 = \frac{1}{m}\sum_{i=0}^{m}(\theta^T.x^{(i)} - y^{(i)})^2 \qquad \frac{\partial}{\partial\theta_j}MSE(X,\theta) = \frac{2}{m}\sum_{i=0}^{m}(\theta^T.x^{(i)} - y^{(i)})x_j^{(i)}$$

Here, $\mathbf{m}$ stands for number of data instances, $\boldsymbol{i}$ stands for the index related to length of the data set and $\boldsymbol{j}$ stands for the number of features (considering $\boldsymbol{j} = 0$ for the bias term). Weights are updated in each iteration so that the MSE(θ) reaches a global minimum with a rate η, which is also called as the learning rate.

$$\theta^{(next)} = \theta^{(now)} - \eta\nabla_\theta MSE(\theta)$$

SGD is a special case of gradient descent where in each iteration gradient is calculated from randomly selected instances of the data set. The *SGDClassifier* in scikit-learn library [3] was used for the training in this work.

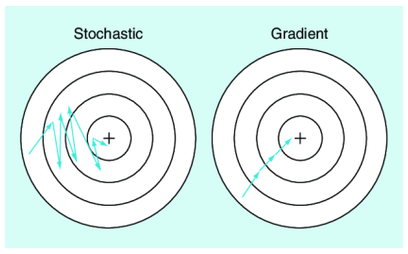

Figure 05: Approaching the global minimum in SGD and GD. (Fig. ref.: Deep learning and virtual drug screening, Xudong Huang et. al.)

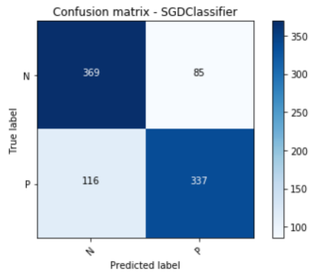

Figure 06: Confusion matrix after testing the SGD classifier.

5. Results - III (Best Model - MNB)

MNB classifier is one of the popular and simple yet very effective model based on the Bayes Theorem [5]. The term "Naive" comes due to the assumption; strong independence in the feature space. Out of all the classifiers tested in this work, MNB classifier showed the best performance after the training based on the whole data set. It was then used to test the domain specificity. The MNB classifier with optimized hyper-parameters was trained and tested with different combinations of training and testing data (Table 03: data set names, Table 04: performances). Confusion matrix for each test is given below:

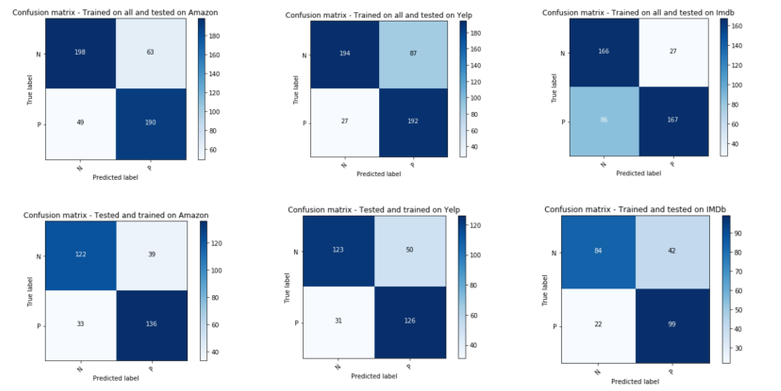

Table 03: Names of the different data sets used for training and testing.

- Comb - Amaz : Trained on Combined reviews and tested on Amazon.
- Comb - Yelp : Trained on Combined reviews and tested on Yelp.
- Comb - IMDb : Trained on Combined reviews and tested on IMDb.
- Amaz - Amaz : Trained on Amazon reviews and tested on Amazon.
- Yelp - Yelp : Trained on Yelp reviews and tested on Yelp.
- IMDb - IMDb : Trained on IMDb reviews and tested on IMDb.

Table 04: Model performances.

Combination	Accuracy	Precision	Recall	F_β score
Comb - Amaz	0.786	0.780	0.780	0.780
Comb - Yelp	0.772	0.790	0.770	0.770
Comb - IMDb	0.747	0.770	0.750	0.750
Amaz - Amaz	0.780	0.780	0.780	0.780
Yelp - Yelp	0.754	0.760	0.750	0.750
IMDb - IMDb	0.741	0.750	0.740	0.740

6. Discussion

Among the classifiers used to predict the customer reviews data, Multinomial Naive Bayes classifier showed the best accuracy. MNB is a simple but powerful model when compared to CNN and SGD in text classification problems. We used the MNB classifier for the testing of domain specificity. Table 04 shows the accuracy, precision, recall and F-score when MNB is trained and tested on different combinations of training and testing data sets. The testing accuracy shows slightly better when trained on combined data and tested using individual company data. With this results, we can conclude that the model training for a customer reviews classification problem in a certain company can be successfully done using combined reviews without necessarily using reviews form that particular company.

Therefore, we conclude that costumer reviews data shows weak domain specificity.

Reference :

[1]. Sentiment Labeled data set (2019), https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences

[2]. Natural Language Tool-kit (2019), https://www.nltk.org/

[3]. Scikit-learn Libraries (2019), https://scikit-learn.org/stable/

[4]. TensorFlow - word embeddings (2019), https://www.tensorflow.org/tutorials/representation/word2vec

[5]. Bayes theorem, https://en.wikipedia.org/wiki/Bayes%27_theorem

Acknowledgment :

This work has been supported by National Science Foundation through an Excellence in Research award (CNS-1831980).

Contact: nalinda05kl@gmail.com