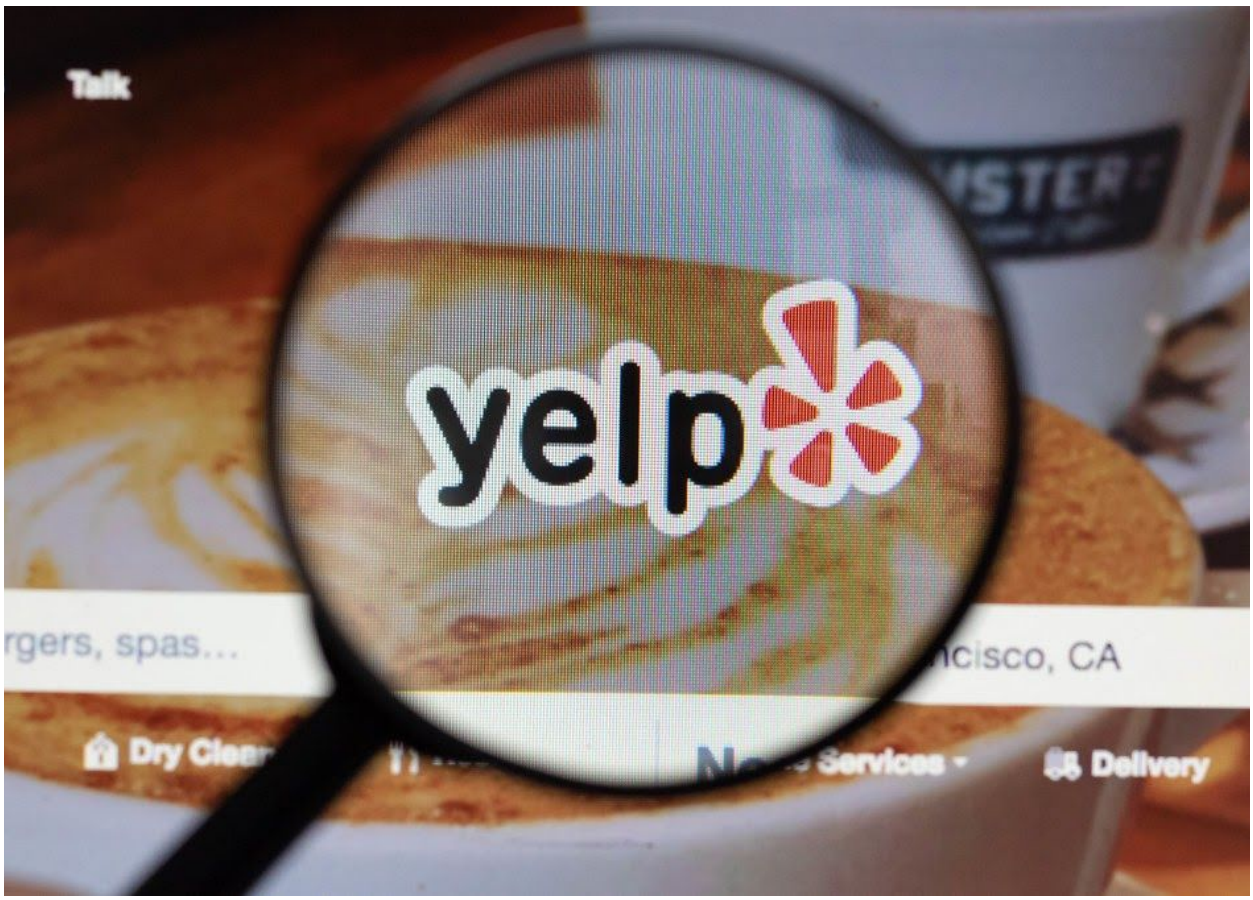
ILS-Z534-SEARCH

DATA ANALYSIS OVER YELP DATABASE

SENTIMENT ANALYSIS & TOPIC MODELLING



Github Link - https://github.com/BAgrawal03/Sentiment-Analysis-on-Yelp-Database.git

Group Members

Bhumika Agrawal - bagrawal@iu.edu

Devanshi Mittal - dsmittal@iu.edu

Abstract

In this project report, we have used a deep learning based model to perform sentiment analysis of a review based on its text only and combine it with any category identified from tips data.

Introduction

Sentiment analysis is critical in identifying customer behaviour and analyzing what products or service aspects customers like or dislike. This information can further be manipulated to profit a business or for marketing purposes.

Proposed Algorithms

The main algorithm used in the project is Long Short Term Memory (LSTM) with word embedding and LSTM with TF-IDF using Keras.

LSTM

As we read any sentence, we retain the entire sequence of words and draw conclusions. The traditional machine learning approaches or even neural networks lack this capability. LSTM overcomes this problem and performs well if the sentence has some contradictory words/points that change the overall meaning.

Eg: “The travel bag has a very chic and pretty look but it isn’t very practical.” This review might not get identified well using other approaches but, this is exactly where LSTM shines.

LSTM with TF-IDF

Another approach we used was the Term Frequency - Inverse Document Frequency, incorporated into the LSTM model. The TF-IDF vectorizer gives us a sparse matrix of the TF-IDF of each word in the document and using those as the features of our LSTM model we tried to predict the sentiment attached to each review. It is important to note that the TF-IDF feature space will be really large so in order to deal with it we choose the most

frequent 500 words across the text documents and applied the TF-IDF transformation of those 500 words as the features for our LSTM model.

LDA

To make the approach semi-supervised, we implemented a combination of supervised and unsupervised methods. Looking at filtering documents on the basis of topics by using an unsupervised approach and further performing sentiment analysis on documents associated with each topic using a supervised approach is something we want to do in the future to implement Aspect based analysis. Hence, we tried performing the unsupervised approach i.e topic modelling using LDA.

Experiment Design

Data Collection

The dataset was provided by the online Yelp Dataset challenge,consisting of five parts which includes reviews,tips,user,business and checkin data. For the purposes of this project, we have used reviews and tips data only. It comprises about 8,021,122 reviews and 1,320,761 tips by 1,968,703 users. Due to our computational challenges, we have used about 50K reviews for the purposes of this project.

The review data includes attributes such as user_id,business_id,stars,useful,funny,cool,text and date. The attributes we used were business id, business categories, review text and rating. The review text was the corpus for our analysis; rating was the identifier for discriminating positive or negative sentiment; business id served as the key for data segregation and the business categories served as the identifier for analyzing sub-categories.

Data Preprocessing

The following steps were followed for data preprocessing -

1. Converting text to lower-case
2. Removing the punctuations and special characters.
3. Stemming

4. Dividing the dataset into train and test sets with a 3:1 ratio.

LSTM Architecture

The LSTM architecture we have used has 3 basic layers -

- Embedding Layer - This layer encodes the input data and maps every word to a unique integer. The Embedding layer is initialized with random weights and will learn an embedding for all of the words in the training dataset.
- LSTM Layer - The core layer which implements the dropout and continual memory classification.
- Dense Layer - This acts as an output layer with softmax activation function.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 1002, 150)         4500000
_________________________________________________________________
lstm_1 (LSTM)                (None, 200)               280800
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 402
=================================================================
Total params: 4,781,202
Trainable params: 4,781,202
Non-trainable params: 0
_________________________________________________________________
```

For LSTM, we have used 50k reviews and trained the model for 3 epochs due to computational challenges.

LSTM with TF-IDF Architecture

Similar to the first approach, we used Keras and as can be seen tried to create the input layer as the LSTM layer takes TF-IDF of most frequent 500 words which is fed to a second LSTM layer size 200. This gets fed to the final layer with softmax activation which yields a vector of 2 labels: positive and negative.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 500, 100)          40800
_________________________________________________________________
dropout_1 (Dropout)          (None, 500, 100)          0
_________________________________________________________________
lstm_2 (LSTM)                (None, 200)               240800
_________________________________________________________________
dropout_2 (Dropout)          (None, 200)               0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 402
=================================================================
Total params: 282,002
Trainable params: 282,002
Non-trainable params: 0
_________________________________________________________________
```

Random Forest

The algorithm random forest as a baseline to measure performance against. The figure below describes the classifier summary for Random Forest model. The algorithm is implemented using Sklearn.The model was implemented by using TF-IDF and Count Vectorizer provided by Sklearn library.

```
RandomForestClassifier(bootstrap=True, class_weight='balanced',
            criterion='gini', max_depth=None, max_features='auto',
            max_leaf_nodes=None, min_impurity_decrease=0.0,
            min_impurity_split=None, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            n_estimators='warn', n_jobs=None, oob_score=False,
            random_state=None, verbose=0, warm_start=False)
```

Latent Dirichlet Allocation (LDA)

As mentioned before, the purpose of using LDA is to be able to predict the topics of the given set of documents using an unsupervised approach. LDA was performed on three kinds of data: Reviews data, Tips data, Review concatenated with tips data. We show results on tips + reviews because it was giving us better results in terms of topics. Also the text was only of the restaurants in order to perform topic modelling in a more segregated manner.

Word cloud for the tips + reviews data

Although we tried to tune the hyperparameters alpha and beta using our data, the computation power did not allow the code to run efficiently. So here are the results we obtained on the first attempt.

```
Topics found via LDA:

Topic #0:
good tacos like place pho just food taco salsa ramen

Topic #1:
beer wings bar place good like just night great chicken

Topic #2:
sushi rice good chicken like roll soup food thai rolls

Topic #3:
food good place great happy bar hour menu service like

Topic #4:
food service time order just place dont minutes like said

Topic #5:
good chicken like salad just ordered sauce food time place
```

It can be seen that topic 0 can be called Mexican Cuisine, topic 1 can be called Bar place, topic 2 can be called Asian Cuisine and so on.

Evaluation

The main metrics used for evaluation are precision,recall and F-1 score as they provide the most reliable results.

Evaluation Results

- **LSTM**

Data Size	Precision	Recall	F1 Score
10K	0.80	0.79	0.80
30K	0.81	0.81	0.81
50K	0.88	0.88	0.88

- **LSTM with TF-IDF**

Data	Precision	Recall	F1 Score
Training	0.66	0.66	0.66
Test	0.66	0.66	0.66

- **Random Forest**

Data	Precision	Recall	F1 Score
10K	0.76	0.76	0.76
30K	0.76	0.79	0.77
50K	0.79	0.80	0.80

.

Conclusion

The accuracy of LSTM when using word embeddings has the best performance rather than using LSTM with TF-IDF because in truth LSTM actually requires data to be in the form of sequences which TF-IDF is not. Hence, using TF-IDF with LSTM was a tried but failed approach and the weak results are because of the same.

The accuracy for LSTM is much better than for Random Forest too as Random Forest often suffers from over-fitting which can be especially problematic with unbalanced or biased datasets.

Limitation

LSTM is a deep learning approach which has the basic requirement of large data availability. LSTM also takes longer to train due to memory-bandwidth-bound computation. As can be observed from the results, LSTM performance improves as the data increases.

Future Work

While sentiment analysis provides efficient insights into the data,it is even more powerful when combined with other techniques to give Aspect Based sentiment analysis. The LDA approach can be used to identify the aspect terms in a dataset and then extrapolated to extract further value from the data available.

Contribution

- Bhumika Agrawal - Data preprocessing, Random Forest and LSTM.
- Devanshi Mittal - LSTM with TF-IDF and LDA.

References

- https://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://medium.com/@Intellica.AI/aspect-based-sentiment-analysis-everything-you-wanted-to-know-1be41572e238