Opinion-Fact Classification

Sarath Chandra Reddy (MT19037), Mani Kumar Reddy (MT19065), Murali Krishna (MT19123)

Problem Definition

In the present-day technology huge amount of data is being generated every day. So, it's turning out to be a challenging task to handle text-based data. In the world of text-based sentences it is not that simple to differentiate between fact and opinions. So, our project is to build the model that classifies/identifies facts from/and opinions in the given text by using various machine learning and deep learning techniques.

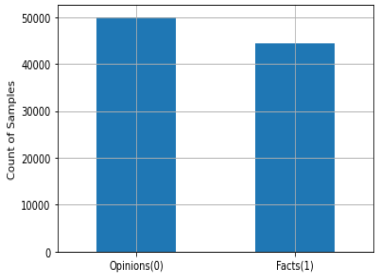
2. Background & Literature Review

There has been a wide range of applications developed over the last decade that mine social media data for diverse objectives to obtain meaningful insights. In terms of data consumption, we can broadly classify this active research area into two categories. The first category, event detection, represents research that captures factual data, such as identifying breaking news from social media streams. The second category, opinion mining, aims at capturing user opinion and/or towards events or entities. In the paper titled "Beyond opinion classification: Extracting facts, opinions and experiences from health forums" by Jorge Carrillo et al. They have considered the sentences or data from online health forums (eDiseases Dataset) which consists of opinions, facts and experiences. They defined fact as "something that can be checked and backed up with evidence" and opinions was defined as "judgement viewpoint or statement that is not conclusive and also which is not always true and can be proven". Along with this, they have also considered experience which is "something someone has lived through and that leaves an impression on his or her".

They have used the following feature types, lexical features (BOW, TF-IDF, Noun phrases), positional features (the position of the sentence within the post and thread), network features (no. of replies), sentiment based features (no. of positive/negative words, no. of adjectives),grammatical features (verb tense, part-of-speech, negation), word-embeddings (Word2Vec). The evaluation metrics used are accuracy, f-measure, recall, precision. They have only used SVM with different combinations of the feature set mentioned above. Any other machine learning or deep learning technique wasn't used in the paper. From the above techniques we have considered lexical features for feature representations and tried out various machine learning and deep-learning techniques.

3. Dataset Used & Proposed Algorithm

The dataset we will be using for this project is hand annotated. We considered the data from "movies" domain and annotated them into opinions and facts. Here, the plot of a movie is considered as fact. whereas the review of an individual for a movie is considered as opinion. The dataset contains 94,379 samples which are facts or opinions. Dataset has opinion count of 50,000 whereas facts of 44,379. The dataset has train, cross-validation & test splits.



We this available data we propose to classify opinions from facts using various machine learning and deep learning techniques.

4. Data Pre-Processing

- Stop-Word removal
- Case Conversion
- Tokenization, Lemmatization
- Removal of alpha-numeric words and special characters.
- Removal of words of length less than 3.

5. Models Applied & Methodology

5.1 KNN (K Nearest Neighbors - Baseline)

The KNN algorithm considers the k nearest neighbors for a test vector, takes the majority vote of class labels of its

neighbors and assigns it to the given test vector. Here, we performed train-cv-test split on the data**.** As, "k" is a hyper-parameter we found out the best "k" using grid-search cv technique applied on CV on basis of accuracy metric. Then after we predicted the labels for test data using the same "k" value. We have used both Bag-of-Words and TF-IDF embedding.

5.2 Naïve Bayes

Naïve Bayes learns the probability distribution of classes from the given training data. When a test sentence is given it calculates the probability of sentence belonging to each class. P(C|Test) is maximized. The class which gives maximum P(C|Test) is the class label of test data. The smoothing factor (alpha) is the hyper-parameter in Naïve Bayes. we found out the best "alpha" using grid-search cv technique applied on CV data on basis of accuracy metric. Then after we predicted the labels for test data using the same "alpha" value. We have used both Bag-of-Words and TF-IDF embedding.

5.3 SVM (Support Vector Machines)

SVM gives the hyperplane that best separates the data. It draws the margin maximizing hyper plane between two classes. The popular hyper parameters we have in SVM are kernel and "C" (regularization parameter). We found out the best hyper-parameters using grid-search cv technique applied on CV data on basis of accuracy metric. Then after we predicted the labels for test data using the same hyper-parameters values. We have used both Bag-of-Words and TF-IDF embedding.

5.4 Decision Trees

Decision tree is a tree-based classifier which is highly interpretable. Here each internal node represents a feature on which decision is made. Uses "Gini-impurity" (which works on the concept of entropy) technique in deciding the what features to be considered as internal nodes. The depth of the tree is the hyper-parameter in Naïve Bayes. we found out the best "depth" using grid-search cv technique applied on CV data on basis of accuracy metric. Then after we predicted the labels for test data using the same "depth" value. Here, we have used both Bag-of-Words and TF-IDF embedding.

5.5 LSTM (Long Short-Term Memory)

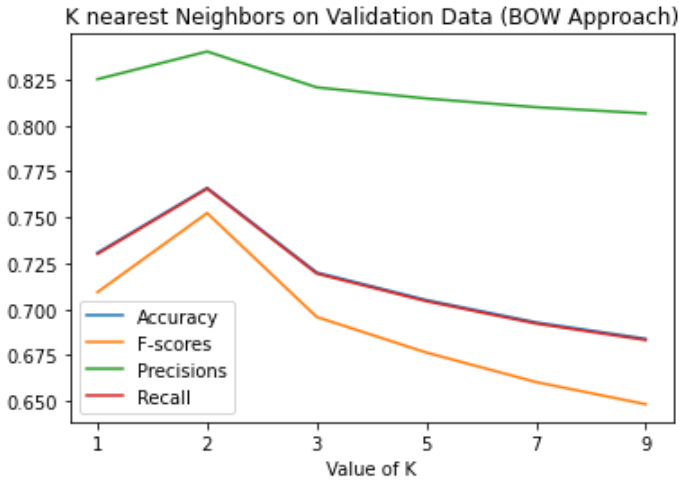
LSTM is the deep-learning technique which is most popular with time-series data. LSTM overcomes the problem of long-term dependencies which prevail in recurrent neural networks thus solving the problem of vanishing-gradient. Here, we considered the rank of the word in the vocabulary when the vocabulary is sorted in descending order based on word occurrences. We have used an embedding layer in the architecture which generates a fixed size context-vector for every word in the sentence considering the words present in its context. We used the SoftMax activation function at the end with 2 activation units as number of labels to be predicted are two with binary cross entropy loss for happening of back propagation through time.

6. Results Achieved on Test Data

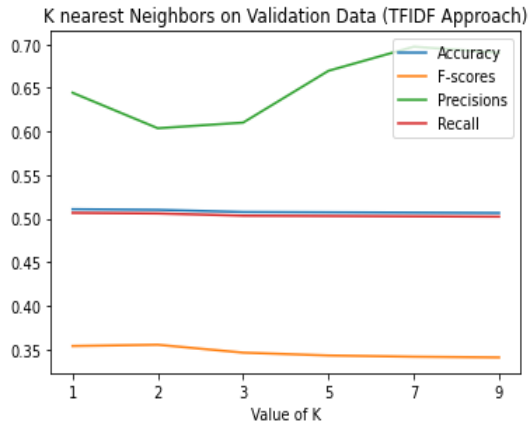
model	word-embedding	precision	recall	f-score	Acc%
K-NN	BOW (baseline)	0.832	0.754	0.739	75.45
K-NN	TFIDF (baseline)	0.6387	0.506	0.351	50.6
Naïve Bayes	BOW	0.814	0.795	0.792	79.50
Naïve Bayes	TF-IDF	0.811	0.788	0.7844	78.85
D-tree	BOW	0.9062	0.905	0.9046	90.46
D-tree	TF-IDF	0.9192	0.918	0.9176	91.76
SVM	BOW	0.9576	0.956	0.9569	95.7
SVM	TF-IDF	0.9542	0.953	0.9539	95.4
LSTM	Rank of word in the vocabulary	0.9866	0.987	0.9867	98.62

7. Graphs & Plots

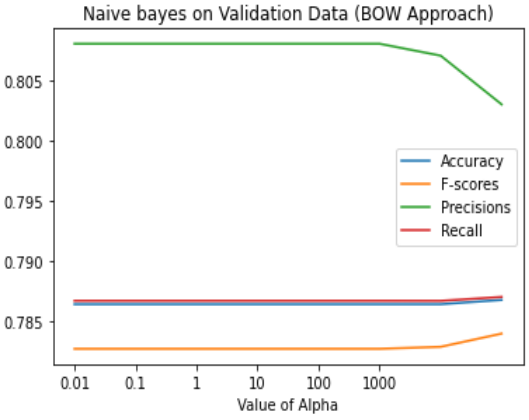
7.1 K-NN- K vs Metrics plot (BOW embedding)



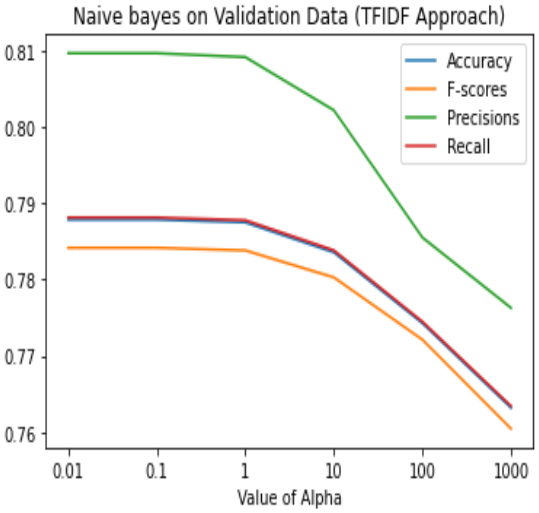
7.2 K-NN- K vs metrics plot (TFIDF embedding)



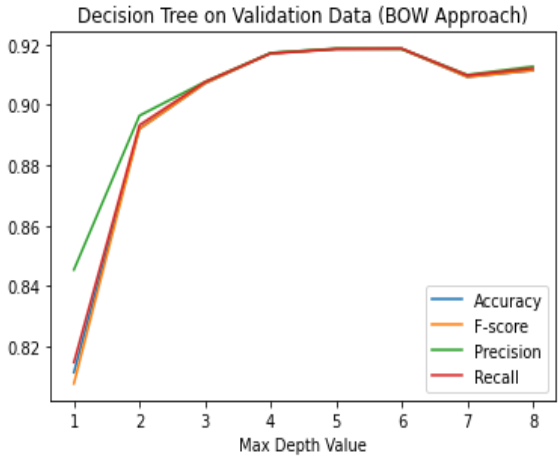
7.3 Naïve Bayes – alpha vs metrics plot (BOW)



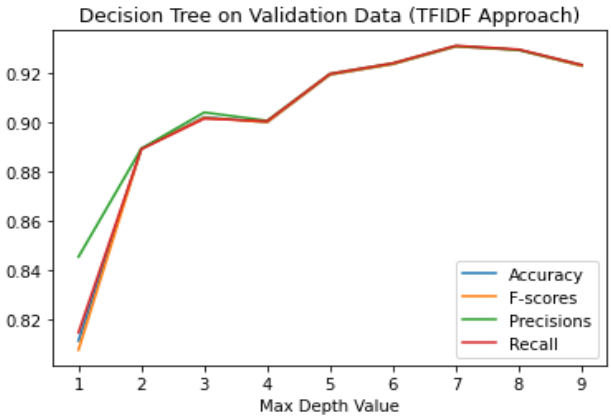
7.4 Naïve Bayes – alpha vs metrics plot (TFIDF)



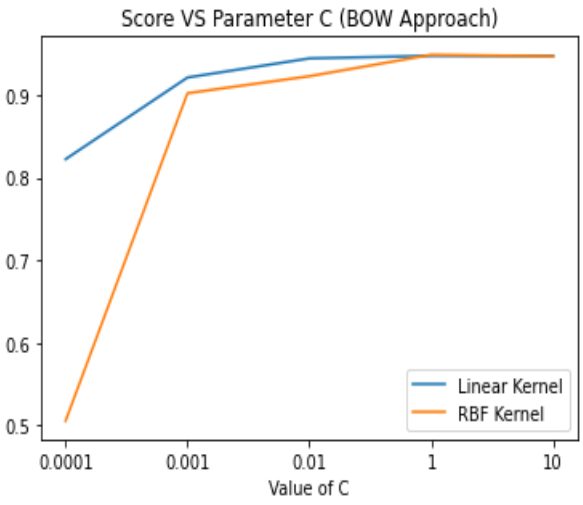
7.5 Decision Trees – depth vs Accuracy plot (BOW)



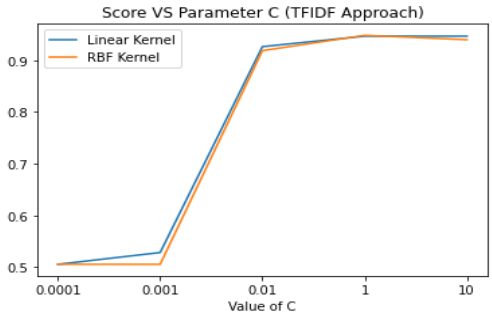
7.6 Decision Trees – depth vs Accuracy plot (TFIDF)



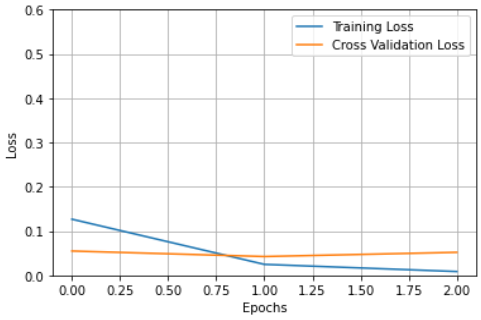
7.7 SVM - C v/s Accuracy plot (BOW)



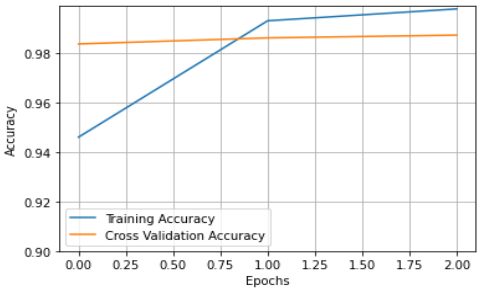
7.8 SVM - C v/s Accuracy plot (TFIDF)



7.11 LSTM - Epochs v/s Loss plot



7.11 LSTM – Epochs v/s CV accuracy plot



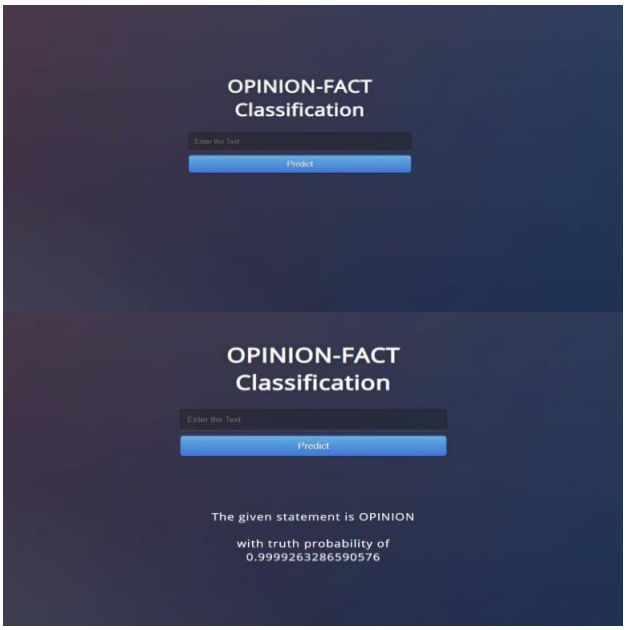
8. Analysis

The reason for better performance of SVMs compared to KNN, Decision trees and naïve bayes might be due to because the SVM always tries to draw the margin maximizing hyper-plane that separate two classes (or more depending upon the count of class labels). Because of this property the samples that are present closer to hyper-plane are now correctly classified which might be mis-classified earlier.

The best performance of LSTM might be due to non-linearity introduced by the activation functions. LSTM is best suited for time-series and text data. They can also capture the long-term dependencies in long-sentences due to the in-built forget gate. This forget gate customizes the LSTM cell about the amount of data it can retain and leave out.

9. Deployment

We deployed he best model and successfully ran our best model on local host using the flask web-frame work. The Screen shots for the same can be seen below.



(after clicking predict the text vanishes from text box)

10. Contributions

- K-NN, Naïve Bayes (BOW & TF-IDF Embedding) – Sarath Chandra Reddy (MT19037)
- Decision Trees & SVM (BOW & TF-IDF Embedding) – Kastala Murali Krishna (MT19132)
- LSTM & final deployment of best model using flask – Mani Kumar Reddy (MT19065)

11. References

[1] Most of the earlier research on opinion classification is done by Wiebe and his collegues (Weibe et al., 1999). they proposed methods for discriminating subjective and objective features.
[2] Hatzivassiloglou and McKeown proposed an un supervised model for learning positively and oriented adjectives with accuracy over 90%.
[3] A similar study was conducted by Ahmet Aker et in this paper titled “Beyond opinion classification: “extracting facts and opinions from health forums”.
[4] https://www.youtube.com/watch?v=UbCWoMf80PY&t=692s