A Trade-off between ML and DL Techniques in Natural Language Processing

Journal of Physics: Conference Series **1831** (2021) 012025 doi:10.1088/1742-6596/1831/1/012025

A Trade-off between ML and DL Techniques in Natural Language Processing

Bhavesh Singh, Rahil Desai, Himanshu Ashar, Parth Tank, Neha Katre

Dwarkadas J. Sanghvi College of Engineering, Mumbai - 400056, India

E-mail: bhavesh.singh@djsce.edu.in, rahil.desai@djsce.edu.in, himanshu.ashar@djsce.edu.in, parthtank60@gmail.com, neha.mendjoge@djsce.ac.in

Abstract. The domain of Natural Language Processing covers various tasks, such as classification, text generation, and language model. The data processed using word embeddings, or vectorizers, is then trained using Machine Learning and Deep Learning algorithms. In order to observe the tradeoff between both these types of algorithms, with respect to data available, accuracy obtained and other factors, a binary classification is undertaken to distinguish between insincere and regular questions on Quora. A dataset called Quora Insincere Questions Classification was used to train various machine learning and deep learning models. A Bidirectional-Long Short Term Network (LSTM) was trained, with the text processed using Global Vectors for Word Representation (GloVe). Machine Learning algorithms such as Extreme Gradient Boosting classifier, Gaussian Naive Bayes, and Support Vector Classifier (SVC), by using the TF-IDF vectorizer to process the text. This paper also presents an evaluation of the above algorithms on the basis of precision, recall, f1 score metrics.

Key words— Natural Language Processing ,Machine Learning, Deep Learning, Bi-directional LSTM, Extreme Gradient Boosting classifier, Gaussian Naive Bayes, SVC

1. Introduction

Artificial Intelligence has been the primary subject of research for the past few decades. The term artificial intelligence, itself conveys the meaning - making a machine simulate human intelligence. Along with artificial intelligence, conventional machine learning and deep learning have also been a focus of researchers. Machine learning and deep learning both are a subset of artificial intelligence.

Machine Learning is referred to as an application of artificial intelligence (AI) that enables the system to automatically learn from experience [1]. The system does not require to be programmed explicitly. Machine Learning can be used to synthesize the fundamental relationships within a large variety of data. This type of dataset is used to solve real-time problems such as Big Data Analytics, Evolution of Information [2]. Deep Learning, in turn, is a subset of Machine Learning. Deep learning algorithms are able to process a large number of features, hence preferable to compute huge datasets and unstructured data. Deep learning facilitates computer systems to analyze and extract important data from the raw data. Both these techniques have enabled computers to perform various intricate problems such as speech recognition, object de-

tection, and recognition [3].

Natural Language Processing is one of the subfields of artificial intelligence (AI). It deals with the interactions of human and computer languages. NLP helps the computers to analyze natural language in the form of speech and text. Machine learning algorithms have been long used to comprehend the text documents and have provided decent results. As the ML algorithms can process only numerical data, the text needs to be converted to numerical form. Term Frequency-Inverse Document Frequency (TF-IDF) represents words in a vectorised numerical format. Deep learning, on the other hand, has enabled computers to make better representation of data and features for intricate tasks [4]. DL algorithms make use of techniques such as glove, word2vec to convert the text data to numerical form. The significant difference between these two techniques is that word2vec relates the target data to the context data and ignores the co-occurrence of some context words, whereas, glove makes use of these co-occurrences to gain essential information.

The paper is organised as follows. Section 2 mentions related works in natural language processing tasks. The system flow, dataset information, and data preprocessing techniques are detailed in section 3. Section 4 describes the algorithms used for performing classification. The results of classification and the conclusion are described in Section 5 and 6 respectively.

2. Related Works

Vandan Mujadia et.al. [5] studied different neural network architecture and machine learning techniques to determine insincere questions asked on Quora. The dataset consisted of about 900 samples, including questions signifying hate speech, sexual content, among other insincere topics. The deep learning approach involved passing word level input features to neural networks, which used LSTMs and pre-trained GloVe embeddings to perform classification, with categorical cross entropy and Adam as the loss function and optimization function respectively. However, this approach gave an accuracy of merely 48.51%. The machine learning approaches involved TF-IDF vectorization to process the text at word and character level. Some of these algorithms were Linear SVM, k-Nearest Neighbours (k-NN), Gradient Boost (GB), Random Forest (RF), and Adaptive Boosting (Adaboost). Adaboost gave the best performance with 66.33% accuracy, and an ensemble model of RF, GB and 3-NN classifiers displayed an accuracy of 62.37%.

Rushali Dhumal Deshmukh et.al. [6] proposed two neural network architectures for tagging parts of the speech, using Marathi as the language. Thirty two tags were observed, containing examples of Common Nouns, Proper Nouns, Auxiliary Verbs, Intensifier particles, and others. The corpus contained words taken from 1500 sentences. The first architecture was a multi-layer perceptron model, with the hidden layers using ReLu activation, trained for 5 epochs and a batch size of 256. Dropout was also applied in order to reduce overfitting. The softmax layer at the output accurately determines the correct tag from the given 32 options. It achieved an accuracy of 85% on test data. The second architecture involved an embedding layer, and a bi-directional LSTM layer of size 32. A time distributed layer finally classifies the output to identify the correct tag. A vector sized 100 was used for allocating the word embeddings. Categorical cross entropy was used as the loss function, and 'rmsprop' served as the optimizer. The model achieved a testing accuracy of 97%.

Batool Armouty et.al. [7] used Support Vector Machine (SVM) to extract keywords from Arabic news documents. Whether a word is a keyword or not would depend on their position in the document, and part of speech. The dataset used comprised approximately 850 Arabic news documents, with an associated set of keywords, which contains five keywords on average. In or-

der to pre-process the text, numbers and punctuations were removed, following which stemming was performed and stopwords were removed. Word tokens with the same stem were grouped together, and TF-IDF and First Occurrence values were calculated for every token. In order to make the data fit for an accurate classification, downsampling was applied to balance the data. The Radial Basis Function (RBF) kernel was used, as the data exhibits nonlinear dependencies. The SVM achieved the best F1 score of 0.64, compared to Naive Bayes and Random Forest, which showed F1 scores of 0.56 and 0.63 respectively.

3. Methodology

In this section, the flow of the system is illustrated with the help of a diagram, following which the dataset used for the purpose of classification is explained. Data preprocessing and techniques used to quantify the text data such as TF-IDF and GloVe Embeddings are discussed. Figure 1

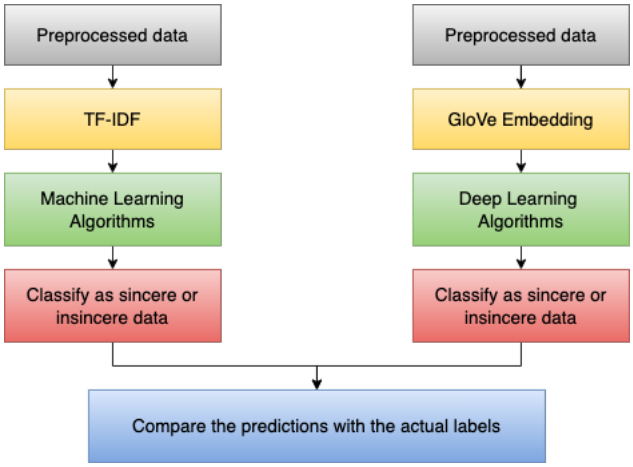

Figure 1: System Flow.

illustrates the system architecture of the proposed approach. The input data is preprocessed to balance out the sincere as well as insincere data and also, to remove the and modify the spelling errors to make the data suitable for further processing. The preprocessed data is then passed on to the TF-IDF vectorizer in case of machine learning algorithms and GloVe Embeddings in case of deep learning algorithms to convert the textual data to numerical features. Machine learning algorithms such as XGBoost, Naive Bayes and Kernel SVM and Deep learning algorithms such as Bi-LSTM and 1D CNN are used to classify the data as sincere or insincere.

3.1. Dataset

The Insincere Questions Dataset, available on Kaggle [8], was used to perform classification. Any question containing a visible agenda, bias, unprecedented harsh statements, or sexually explicit content, defines an insincere question. The questions contained in the dataset are labelled '1' if insincere, and '0' if sincere. Training data consisted of 1306122 tuples in total, of which 1225312 questions were insincere, and 80810 insincere questions. Noisy data and outliers present in the dataset were addressed during preprocessing. The significant words in the sincere and insincere questions are represented in the form of word clouds in the fig 2 and 3.

3.2. Data Preprocessing

The dataset used for the research contained 1225312 tuples for sincere data and 80810 tuples of insincere data. The dataset was not adequately balanced; being biased towards the insincere

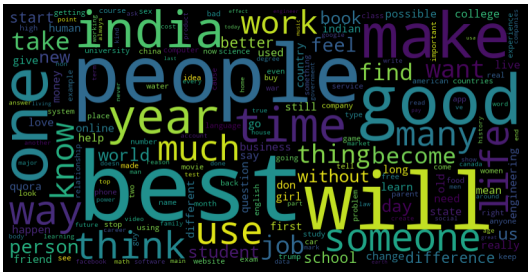

Figure 2: Sincere class dataset

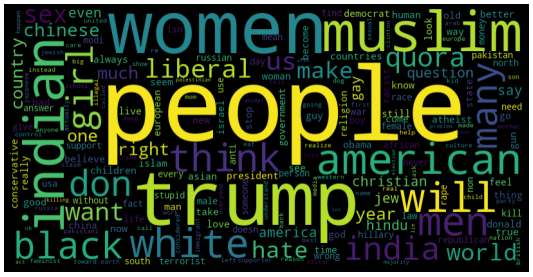

Figure 3: Insincere class dataset

data. To process the data, the punctuations and the tuples containing misspelled words were removed. Contractions such as “aren’t” and “ain’t” were modified to “are not” and “ am not”. Also, the tuples containing words which were not a part of glove embeddings were removed. The dataset obtained after preprocessing was balanced, containing 84496 tuples of sincere data and 70919 tuples of insincere data.

3.3. TF-IDF

Machine Learning algorithms can only process numerical inputs. To cater to this requirement, in this paper, TF-IDF is used. TF-IDF helps in quantifying the text document so that it can be utilized as an input to the algorithms. In this technique, each word in the text document is assigned with a weight. This weight signifies the importance of that word in a corpus or text document. TF-IDF works in collaborative methods of TF (Term Frequency) and IDF (Inverse Document Frequency). Term Frequency denotes the frequency of a word in the document, whereas, Inverse Document Frequency helps in gaining the informativeness of a particular term in the document. So, IDF can be used to compute the uniqueness of a particular term [9].

3.4. Glove Embedding

As machine learning algorithms, even deep learning algorithms require numerical data as input. In this paper, glove embeddings are used for the purpose of converting textual data to numerical features. Glove refers to global vectors used for word embeddings. It is an unsupervised learning approach for representing words. Glove uses the co-occurrences of words; how frequently two words appear in a corpus to form a linear sub-structure of word vectors. The primary focus of glove embeddings is to maximize the probability of occurrence of a context word given a word from the corpus. Glove makes use of dynamic logistic regression to achieve this. The resultant word vectors can be provided as an input to the deep learning models [10].

4. Approaches

To draw conclusive empirical results about the performance of Machine Learning and Deep Learning approaches on natural language problems, following algorithms were used along with the given dataset.

4.1. Machine Learning algorithms

Before passing the input to the classifiers, the corpus of words was converted to feature vectors using the Term Frequency - Inverse Document Frequency (TF-IDF) vectorizer. The stopwords in the content were removed, and the maximum features were set at 1000, in order to optimize execution time.

4.1.1. Naive Bayes classifier: The Naive Bayes algorithm utilises the Bayes' Theorem for performing classification tasks [11]. To predict a particular element's membership, it calculates the probability of membership to all the classes, based on data points being associated to a particular task. The Gaussian Naive Bayes' classifier is an ideal use case when all the data points are distributed among classes in a normal (Gaussian) distribution.

4.1.2. Extreme Gradient Boosting (XGBoost) classifier: The XGBoost classifier uses the Gradient Boosting decision tree algorithm. It has a fast execution speed when compared to other gradient boosting implementations, with an impressive performance for classification applications. New models are iteratively added to correct the prior models' errors, subsequently boosting the reliability of the final prediction. Gradient boosting ensures the minimisation of the loss function while adding new models, by using the gradient descent algorithm.

4.1.3. SVM: The Support Vector Machine finds its application in both classification and regression problems [12]. A hyperplane is created by using the extreme data points (support vectors) from the dataset. On training, it creates decision boundaries, which classify the example appropriately. When the data points can be separated by a straight line in the hyperplane, it is termed as linearly separable data. In this case, the linear SVM classifier is used.

4.2. Deep Learning algorithms
For creating feature vectors to provide numerical values to train the neural networks, word embeddings are required. GloVe embeddings [13] were used to transform the inputs in the Embedding layer, followed by the remaining layers in the respective neural network architectures. The dimensionality of the GloVe embeddings was chosen to be 300.

4.2.1. Bidirectional Long Short Term Memory Networks (Bi-LSTM): When looping over the cell, Recurrent Neural Networks (RNNs) can connect previous information in the time frame to the current time frame. However, they may not be able to detect long-range dependencies effectively. LSTMs are inherently better at registering long-range relations. Each LSTM cell contains four networks, unlike an RNN, which contains a single network. Bi-directional LSTMs further improve long-range connectivity in both directions, so that past and future time steps can be accessed. This neural network consisted of five layers; an input layer, an Embedding layer, a Bidirectional LSTM layer, a Global Max Pooling layer, and a Dense layers.

The input to this neural network can accommodate the maximum word length of the set of questions, with the shorter questions padded with zeros to achieve a uniform dimensionality. The embedding layer creates a feature matrix with the number of features to be 300, which generates an output having a shape 67x300, subsequently passed to the bi-directional LSTM layer. This layer contains 512 LSTM units and continues to return sequences, giving an output of size 67 x 1024, which is passed through a global max pooling layer to give 1024 values. The final Dense layer uses a Softmax function to predict whether the question was insincere or not. All the previous Convolutional and Dense layers use Relu activation. The model is trained for 5 epochs, with a batch size of 32. Adam and binary cross entropy are used as the optimizer and loss function respectively.

4.2.2. One dimension Convolution Neural Network (1D-CNN): Convolutional Neural Networks apply a kernel filter (customisable in dimensionality) to the input to transform it into a matrix of features. Every consequent convolutional layer describes the features in further detail. This neural network consisted of eight layers; an input layer, an Embedding layer, a 1D

convolutional layer, a max pooling layer, another 1D convolutional layer, a global max pooling layer, two Dense layers.

The input and embedding layer of this neural network are of dimensionalities identical to those in the above mentioned neural network (67x300). The first Convolutional block following this consists of a one dimensional Convolutional layer with 128 units, and a kernel of size 3, to give an output having a shape 65x128. This is passed through a one dimensional max pooling layer with a pool size of 3, altering the output shape to 21x128. The second convolutional block contains another one dimensional Convolutional layer having 128 units and a kernel size of 3, which gives an output of shape 19x128. This layer is followed by a global max pooling layer. The output of this block is passed through a Dense layer containing 128 units. All the above Dense and Convolutional layers had a Relu activation function. The final Dense layer determines if the question is insincere by using a Softmax activation function. This model is also trained for 5 epochs, each batch of size 32. The optimizer and loss functions remain Adam and binary cross entropy.

5. Results

The precision, F1 score, and recall are used as evaluation metrics for analysing the classification results on the dataset. The results are tabulated below in Table 1.

Table 1: Evaluation metrics on different algorithms.

Scores		**Algorithms**				
		ML Algorithms			**DL Algorithms**	
		Naive Bayes	**XGBoost**	**SVM**	**Bi-LSTM**	**1D CNN**
Precision	**Sincere**	0.84	0.74	0.85	0.89	0.91
	Insincere	0.76	0.78	0.80	0.86	0.84
F1	**Sincere**	0.78	0.82	0.85	0.90	0.89
	Insincere	0.82	0.59	0.81	0.88	0.90
Recall	**Sincere**	0.81	0.82	0.86	0.88	0.88
	Insincere	0.79	0.70	0.83	0.87	0.87

5.1. Observation

For the purpose of Natural Language Processing, machine learning algorithms used TF-IDF vectorization. When vectorized using TF-IDF, words' representation depends upon their frequency in the given dataset. On the other hand, GloVe embeddings have been pre-trained on a huge corpus of data, to create a 300 dimensional vector for every word's representation. Hence, the possibility of inadequate representation of momentous words is minimised when using GloVe embeddings, making them a better choice for word representation when compared to TF-IDF. When using a Bidirectional LSTM, the entire context of a given sentence can be processed by the neural network, including long-range dependencies and time frames from the past or future, for a given time step. In one dimensional CNNs the features of input vectors are represented with

more detail, when passed through the layer. Therefore, as evident from Table 1, deep learning models perform better when compared to machine learning algorithms like SVM, where data is reduced to points in the space.

6. Conclusion and Future Scope

This paper provides a comprehensive comparison of the results obtained from machine learning and deep learning algorithms on the insincere data. As evident from the observation table, 1D CNN provided better accuracy among all the algorithms used for the research. This was followed by Bidirectional LSTMs which also performed well on the dataset. Among the machine learning algorithms, Linear SVM provided the most satisfactory results. Overall, it is evident that deep learning techniques used with glove embeddings outcast the results obtained from the machine learning algorithms along with TF-IDF vectorizer. This behaviour was observed owing to the more detailed feature representation of neural networks when compared to machine learning algorithms, as well as the superior text processing of GloVe embeddings, when compared to TF-IDF vectors. To conduct future work, the TF-IDF word counts can be increased for creating the vectors for machine learning algorithms. Further demonstration of comparison between the algorithms can be achieved by training the models with different textual datasets. If the objective is to compare machine learning and deep learning algorithms based on text processing, alternative methods, such as the Bag of Words model and Word2Vec embeddings can be used.

References

[1] Expert System. 2020. What Is Machine Learning? A Definition - Expert System. [online] Available at: <https://expertsystem.com/machine-learning-definition/> [Accessed 16 September 2020].
[2] Awad M., Khanna R. (2015) Machine Learning. In: Efficient Learning Machines. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-5990-9_1
[3] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539
[4] Richard Socher, Yoshua Bengio, and Christopher D Manning. 2012. Deep learning for NLP (without magic). In Tutorial Abstracts of ACL 2012. Association for Computational Linguistics, 5–5
[5] Mujadia, Vandan, Pruthwik Mishra, and Dipti Misra Sharma. "Classification of Insincere Questions with ML and Neural Approaches." FIRE (Working Notes). 2019.
[6] R. Dhumal Deshmukh and A. Kiwelekar, "Deep Learning Techniques for Part of Speech Tagging by Natural Language Processing," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 2020, pp. 76-81, doi: 10.1109/ICIMIA48430.2020.9074941.
[7] B. Armouty and S. Tedmori, "Automated Keyword Extraction using Support Vector Machine from Arabic News Documents," 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 2019, pp. 342-346, doi: 10.1109/JEEIT.2019.8717420.
[8] Kaggle.com. 2020. Quora Insincere Questions Classification — Kaggle. [online] Available at: ¡https://www.kaggle.com/c/quora-insincere-questions-classification/data¿ [Accessed 1 September 2020].
[9] Medium. 2019. TF-IDF For Document Ranking From Scratch In Python On Real World Dataset.. [online] Available at: ¡https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089¿ [Accessed 14 September 2020].
[10] Medium. 2018. Word Vectors In Natural Language Processing: Global Vectors (Glove). [online] Available at: ¡https://medium.com/sciforce/word-vectors-in-natural-language-processing-global-vectors-glove-51339db89639¿ [Accessed 15 September 2020].
[11] Lewis D.D. (1998) Naive (Bayes) at forty: The independence assumption in information retrieval. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0026666
[12] Suykens, J., Vandewalle, J. Least Squares Support Vector Machine Classifiers. Neural Processing Letters 9, 293–300 (1999). https://doi.org/10.1023/A:1018628609742
[13] Pennington, Jeffrey & Socher, Richard & Manning, Christoper. (2014). Glove: Global Vectors for Word Representation. EMNLP. 14. 1532-1543. 10.3115/v1/D14-1162.