Stance Detection: A CNN and LSTM Approach

Praveen Naik S., Siddesh L. C., Shreyas Pandith, Nagaratna B. Chittaragi, Shashidhar G. Koolagudi
Dept. of Computer Science and Engineering, National Institute of Technology Karnataka
Surathkal, India siddeshlc8@gmail.com, praveennaik.s98@gmail.com,shreyaspandith98@gmail.com,
nbchittaragi@gmail.com, koolagudi@nitk.edu.in

***Abstract*—Social media has become one of the main channels for people to share their views and communicate with society. We can often detect and recognize from these statements whether the person is in favor, against or neutral towards a given subject. These opinions and thoughts of people from social media are useful for various companies. In our paper, we present our effort to produce a state-of-the-art Stance Classifier based on Convolutional Neural Networks and Long Short Term Memory Networks(CNN and LSTM). We use a broad set of labeled data with word embedding to predict stance of unlabeled tweets. We compared our model with the SemEval2016 Stance detection Task to verify the efficiency and scalability of our model which outperformed all teams score.**

Keywords*—*Demonetisation, Stance, Dataset, Veracity

I. INTRODUCTION

In today's generation, social media platforms such as Twitter, Linkedin, Facebook, Instagram have gained a lot of popularity. They provide people the medium to communicate and connect with family, friends, and colleagues and express their opinions and thoughts freely and this makes people express themselves on these platforms very often. We can find views on almost any topic may it be politics, sports or movies. The researchers use the name Big Data, to this kind of data, characterized by "3V" which stands for, Variety, Volume, and Velocity. It can also be referred to as "5V," i.e., for Value and Veracity. Analysis of these opinions and views in big data is referred to as stance detection or opinion mining.

Stance detection is the process of automatically determining from the text/statement whether the author is in favor or against or is neutral towards a given target. The target in this paper is 'Demonetisation' which was implemented by the Government of India on 8th November 2016. The Government declared that the currency in denominations of 500 and 1000 was invalid. The government believed that this decision was taken to eliminate black money and the use of counterfeit cash which was used to fund illegal activities and terrorism. People all over the country had different reactions to this event, and many of them used social media like Twitter to express their opinions.

Consider for example the following tweet: 'Demonetisation has caused a lot of problems to people across the nation'. We can say that the author of this tweet is most likely to be against the target demonetization.There have been several experiments and research going on in the field of opinion mining on social media and online texts. Stance detection can provide a lot of information about the tweets present in social media like Twitter and can benefit many other tasks such as information retrieval, text summarization, etc. Stance detection provides us with a collective opinion of people about a particular target. These results are helpful in making necessary changes and coming up with more practical policies in the future.

The main contribution of this paper is the LSTM approach of stance detection with results showing the percentage of tweets which are in favor, against or neutral about the subject and A Resource of the dataset obtained from tweets on 'Demonetization' with tweet level annotation for stance towards this target. The model has been used for the target of demonetization. The results obtained in this paper can be used by the government to implement policies which are reliable and can benefit the common man. The paper is organized as follows: Section II focuses on the prominent work done in regard to the concerned field. Section III elucidates the proposed methodology and the model used along with the steps taken to obtain the necessary results. Section IV pertains to the results and the analysis of the proposed methodology. Section V includes the conclusion of the paper and provides the scope for future work.

II. LITERATURE SURVEY

The author in [4] presented an SVM-based stance detection approach on sports-related tweets. They have carried out experiments with SVM classifiers for each target with various features including those based on hashtags, unigrams, bigrams, external links, positive and negative emoticons, and named entities. The 10-fold cross-validation results of the SVM classifiers using hashtag + unigrams use+named entities are promising and better than the other combinations used and tested. Therefore, the findings suggest that features based on unigrams, hashtags, and named entities can be used to improve stance detection performance on sports-related tweets.

The author in [3] was focused on detecting stance towards Hillary Clinton and Donald Trump that was contesting the political campaign for the 2016 U.S.Two groups of features namely sentimental ,structural and Context-based features were used to detect stance of the tweet.They used packages which had polarity labeled to each english word which helped them in sentiment analysis.Difference between positive and negative words were also considered.The frequency of hashtags present ,the frequency of screen names (often called mentions) in each tweet,the frequency of exclamation marks, question marks, periods, commas were used as structural features.Their hypothesis was that the context-based features would capture some domain related information.Thus they manually created a list of entities related to the Party

Presidential Primaries for the Democratic and Republican parties from Wikipedia.The paper exploited various types of context-based features considering different types of relationships between the target and the entities around the target. Target by name ,target by pronoun,target party,target party colleagues which were used for stance detection.

The author in [2] proposed a simplistic two-phase approach, with intuitive features and traditional SVM learning, to solve the problem of stance detection over various targets from social media. In the first phase, they borrowed from the subjectivity literature, and put forward a novel syntactic feature, to differentiate and classify the neutral vs. non-neutral tweets.In phase two, they use features from the sentiment polarity detection literature that apply in the current context, and put forward a novel semantic feature, to classify the non-neutral tweets into favor vs. against.
Empirically, on the benchmark dataset of SemEval 2016, they demonstrated the effectiveness of the system they developed, where the target in the test dataset was part of the training dataset (SemEval 2016 Task A), as well as where it is not (SemEval Task B). For both tasks, they outperformed the literature by substantial F-score improvements, 5.46 for Task A and 5.29 for Task B.

The author in [5] proposed a joint learning approach for stance detection with sentiment classification. They consider stance detection as the main task, and sentiment classification as the auxiliary task. This approach leveraged the auxiliary representation to support the execution of the main task. The idea of their approach lied in that the auxiliary long short term memory network (LSTM) layer that is shared by both the main and auxiliary tasks to achieve a mutual semantic representation through parameter sharing.

The author in [1] proposed a two-phase approach of stance detection using attention embedding at each phase and encoding using LSTM.In the first phase, they performed subjectivity analysis of the tweets, using a combination of LSTM and attention embedding. In the second phase,they performed sentiment analysis on the subjective tweets, again using a combination of LSTM and attention embedding.one. At each phase, they made use of two components,a bi-directional LSTM and an attention mechanism. The bi-directional LSTM was used for feature encoding. The attention logic used augmentation of the word embeddings with target topics, and subsequently passing it through a linear layer for computing attention of each word in the text in the context of the topic under consideration.

III. Proposed Methodology

The proposed approach includes the three important stages namely: Data Acquisition, Data pre-processing, Supervised training and the current section includes the brief discussions of the same.

A. Dataset Acquisition

We use two datasets for our model. One is from SemEval 2016 Dataset for stance detection in tweets and other dataset containing tweets about Demonetisation. We collected tweets related to the Demonetisation that was implemented in India in 2016. We used Twitter Scraper API to collect tweets using the keywords 'demonetisation' over six months after Demonetisation was realized.Each tweet is handled in JSON format after which the content of the tweet and the tweet id extracted from it. A total of 4000 tweets has gathered for the training model.

The tweets manually annotated with one of the following stance tags: 'FAVOR,' 'AGAINST' and 'NONE.' Some keywords and hashtags, such as IAmWithModi, ByeByeBlackMoney and 'Cashless India' are direct indicators that the author is in favor of demonetization. Similarly, hashtags such as StopDemonetisation, FailedModi, and ModiSurgicalStrikeOnCommonMan are clear indicators that the author is against demonetization.

Examples of tweets with different stances towards the target are:
Target: Demonetisation

Tweet:@narendramodi thanks for demonetization we are with you
Stance: FAVOR

Tweet: @PMOIndia Modi Ji you played a prank with the People in your country, people are hassled. Show mercy on 500/1000 and take demonetization back
Stance: AGAINST

Tweet: Ministers are confused about how to do Hindu Muslim politics on demonetization, this is everyone's unity. everyone's progress.
Stance: NONE

B. Data pre-processing

Before supplying the tweets to any training stage, they are pre-processed using the following procedure:

- Replacement of symbols like /(){}[]\ ,; from tweets by space.
- Bad Symbols replacements like the replacement of 0̂-9a-z +_.
- All tweets are lowercased.
- The Stopwords of English Literature Removal.

C. Model Description

We attempted to produce a state-of-the-art Twitter sentiment classifier using Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTMs) networks.

The input to the model are the tweets, which are extracted and tokenized in words. Each word mapped to a vector representation, i.e., a word embedding, such that an entire tweet can do assigned to an s × d sized matrix, where d is the dimension of the embedding space and d is the number of words in the tweet (we chose d = 100). We follow zero padding strategy such that all tweets have the same matrix

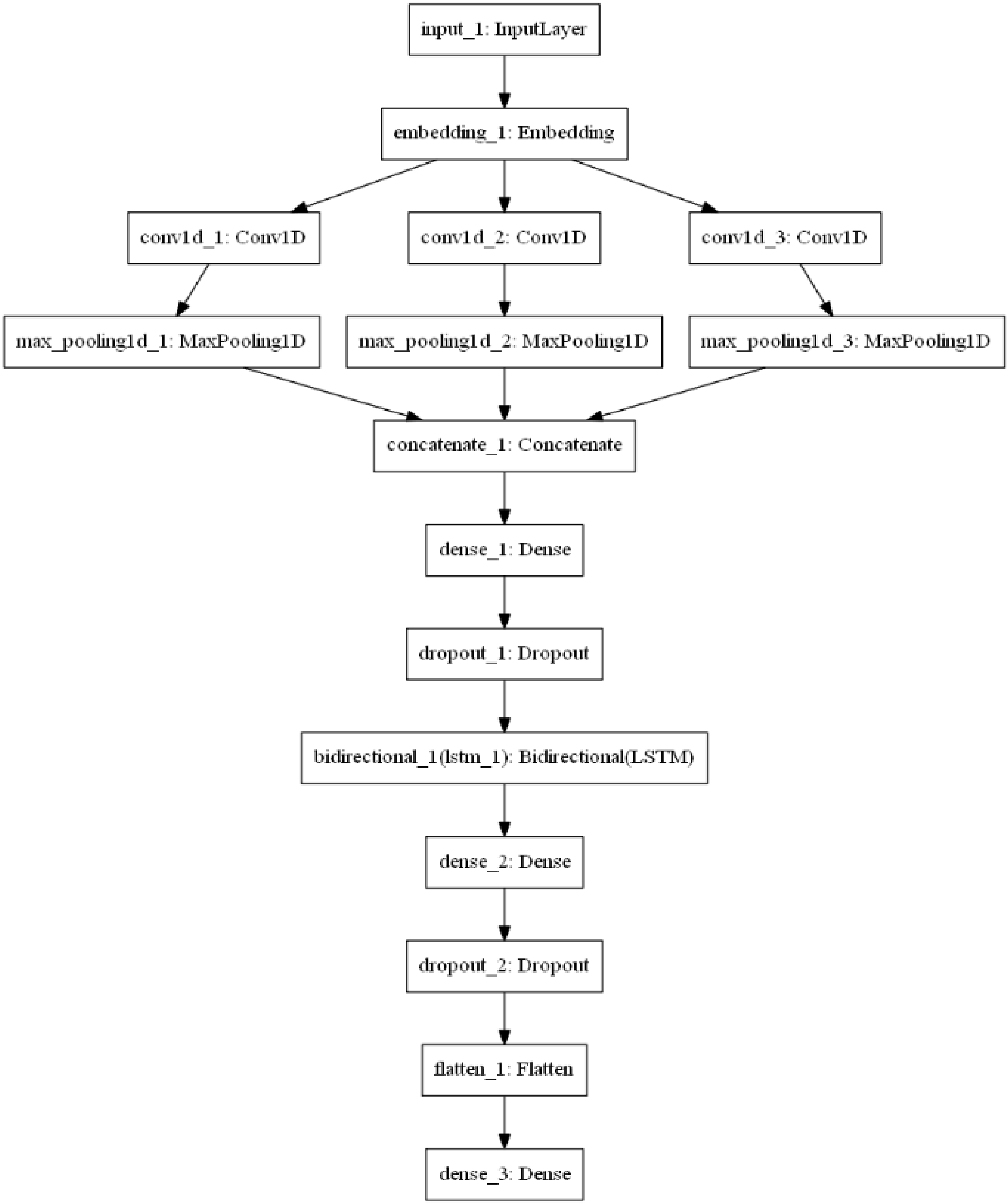

Figure 1: Flow diagram

Layer (type)	Output Shape	Param #	Connected to
input_5 (InputLayer)	(None, 100)	0	
embedding_7 (Embedding)	(None, 100, 100)	5000000	input_5[0][0]
conv1d_13 (Conv1D)	(None, 98, 200)	60200	embedding_7[0][0]
conv1d_14 (Conv1D)	(None, 97, 200)	80200	embedding_7[0][0]
conv1d_15 (Conv1D)	(None, 96, 200)	100200	embedding_7[0][0]
max_pooling1d_13 (MaxPooling1D)	(None, 1, 200)	0	conv1d_13[0][0]
max_pooling1d_14 (MaxPooling1D)	(None, 1, 200)	0	conv1d_14[0][0]
max_pooling1d_15 (MaxPooling1D)	(None, 1, 200)	0	conv1d_15[0][0]
concatenate_5 (Concatenate)	(None, 1, 600)	0	max_pooling1d_13[0][0] max_pooling1d_14[0][0] max_pooling1d_15[0][0]
dense_15 (Dense)	(None, 1, 30)	18030	concatenate_5[0][0]
dropout_9 (Dropout)	(None, 1, 30)	0	dense_15[0][0]
bidirectional_5 (Bidirectional)	(None, 1, 200)	104800	dropout_9[0][0]
dense_16 (Dense)	(None, 1, 30)	6030	bidirectional_5[0][0]
dropout_10 (Dropout)	(None, 1, 30)	0	dense_16[0][0]
flatten_5 (Flatten)	(None, 30)	0	dropout_10[0][0]
dense_17 (Dense)	(None, 3)	93	flatten_5[0][0]

Total params: 5,369,553
Trainable params: 5,369,553
Non-trainable params: 0

Figure 2: Model Architecture

dimension X R$\hat{}$(s'×d), where we chose s' = 80. We then apply several convolution operations of different sizes to this convolutional matrix.

A unique convolution involves a filtering matrix w R$\hat{}$(h×d) where h is the size of the convolution, indicating the number of words it spans. The output c R$\hat{}$(s'h+1) is, hence, a concatenation of the convolution over all possible window of words in the tweet. In the paper, we used three filter sizes, and we used a total of 200 filtering matrices for each filter size.

Us later apply a max-pooling operation to each convolution. The max-pooling method selects the most important features for each convolution, autonomously of where in the tweet this feature is located. In other words, CNN's composition efficiently extracts the most significant n-grams in the embedding space, which is why we believe these methods are good at sentence classification. Ultimately softmax layer gives the final classification probabilities.

To reduce overfitting, we added a dropout layer after the max-pooling layer and later the fully connected hidden neural network layer, with a dropout probability of 50% during training.In the subsequent phase, the output of the CNN model given to the bi-directional LSTM model.

Let we now describe the architecture of the LSTM system. Its main building pieces are two LSTM units. LSTMs are a member of the recurrent neural networks (RNN) group, which are neural networks that are built to deal with sequential data by sharing their internal weights across the sequence. For each component in the series, that is for each word in the tweet, and the RNN uses the ordinary word embedding and its previous hidden state to compute the next hidden state.

One disadvantage from the LSTM is that it does not sufficiently take into account post word information because the sentence is read-only in one direction, ie, forward direction. To resolve this problem, we employ what is known as a bidirectional LSTM, that is two LSTMs whose outputs accumulated together. First LSTM reads the sentence forward, and the second LSTM learns it backward. We concatenate the hidden states of each LSTM after they processed their own final word which gives a vector, which is fed to a fully connected hidden layer of size 30, and then passed Through a softmax layer to provide the final classification probabilities. Here again, we use dropout to reduce over-fitting; we add a dropout layer before and after the LSTMs, and after the fully connected hidden layer, with a dropout probability of 50during training.

Figure 1 and 2 illustrate the structure of our model.

D. Supervised training

We trained our model with both the dataset we mentioned in section 3.1 The training stage uses the human labeled data provided by SemEval-2016. After the tweet pre-processing and vector conversion the vector is given two the input layer. After evaluating the model with SemEval 2016 dataset and when we got better results with that dataset we moved to Demonetisation dataset. We trained the final model with the demonetization dataset.

IV. Experimental Analysis and Results

We perform pre-processing, followed by a CNN and LSTM based machine learning. The table I the performances delivered by the system for tweet stance detection task.

TABLE I. RESULTS AND ANALYSIS

No	Task	Accuracy(%)
1	SemEval2016	75.4
2	Demonetisation	69.3

We believe that our system outperforms all the other methods for all the accuracy scores that have reported in the literature and the results of the SemEval 2016 task. The accuracy obtained by our system exceeds all best-performance points delivered by all the models presented in the SemEval 2016 Task and the 0.74 of the model in [2].

The table II shows the results of SemEval2016 and our system out-performs their models.

V. Conclusions

Stance detection is being used in various applications today. With the bounty of on-going research in this field, it has become one of the most important tasks on big data. We started by presenting the dataset collected from twitter for stance detection on Demonetisation. We described the methods used for annotating each tweet with a stance towards the target. We attempted to produce a state-of-the-art Twitter sentiment classifier using Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTMs) networks.

TABLE II. RESULTS AND ANALYSIS

No	Teams	Official metric
	Benchmark system by organizers	0.6910
1	MITRE	0.6782
2	pkudblab	0.6733
3	TakeLab	0.6683
4	PKULCWM	0.6576
5	ECNU	0.6555
6	CU-GWU Perspective	0.6360
7	IUCL-RF	0.6360
8	DeepStance	0.6354
9	UWB	0.6342
10	IDI@NTNU	0.6247
11	Tohoku	0.6221
12	ltl.uni-due	0.6173
13	LitisMind	0.6144
14	JU_NLP	0.6060
15	NEUSA	0.6012
16	nldsucsc	0.5936
17	WFU/TNT	0.5922
18	INESC-ID	0.5758
19	Thomson Reuters	0.4619

References

[1] Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. “Topical Stance Detection for Twitter: A Two-Phase LSTM Model Using Attention”. In: (2018), pp. 529–536.

[2] Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. “Twitter Stance Detection—A Subjectivity and Sentiment Polarity Inspired Two-Phase Approach”. In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2017, pp. 365–372.

[3] Mirko Lai et al. “Friends and enemies of Clinton and Trump: using context for detecting stance in political tweets”. In: (2016), pp. 155–168.

[4] Saif Mohammad et al. “Semeval-2016 task 6: Detecting stance in tweets”. In: (2016), pp. 31–41.

[5] Qingying Sun et al. “Stance detection via sentiment information and neural network model”. In: *Frontiers of Computer Science* 13.1 (2019), pp. 127–138.