

# DICE: Deep Intelligent Contextual Embedding for Twitter Sentiment Analysis

Usman Naseem and Katarzyna Musial

Advanced Analytics Institute, School of Computer Science

University of Technology Sydney, Australia

Email: usman.naseem@student.uts.edu.au and katarzyna.musial-gabrys@uts.edu.au

**Abstract**—The sentiment analysis of social media-based short text (e.g., Twitter messages) is very valuable for many good reasons, explored increasingly in different communities such as text analysis, social media analysis, and recommendation. However, it is challenging as tweet-like social media text is often short, informal and noisy, and involves language ambiguity such as polysemy. The existing sentiment analysis approaches are mainly for document and clean textual data. Accordingly, we propose a Deep Intelligent Contextual Embedding (DICE), which enhances the tweet quality by handling noises within contexts, and then integrates four embeddings to involve polysemy in context, semantics, syntax, and sentiment knowledge of words in a tweet. DICE is then fed to a Bi-directional Long Short Term Memory (BiLSTM) network with attention to determine the sentiment of a tweet. The experimental results show that our model outperforms several baselines of both classic classifiers and combinations of various word embedding models in the sentiment analysis of airline-related tweets.

## I. INTRODUCTION

Understanding the content and sentiment of information published online, including the social media platforms, where people share their opinions and views, is crucial from the perspective of improving the services, products, and recommendations for those users. Although many different approaches have been proposed, we are still not able to fully utilize the polysemy in the context, semantic information, sentiment, and syntax of the published information. This is especially vivid in the case of short statements and descriptions as tweets. With limited information available, e.g., tweets are limited to 140 characters, the analysis becomes very challenging.

Additionally, the language used on social media and blogs is ubiquitous in nature as it is unstructured and very informal at times. Users write in their own words, and use abbreviations, different punctuations, incorrect spelling, emoticons, slang words, and URLs, etc. All of those language imperfections cause a lot of noise in the data, and one of the major challenges is to handle this unstructured and informal text by applying appropriate cleaning and pre-processing mechanisms.

Other than the noisy nature of tweets, utilizing the context of tweets in terms of polysemy, semantics, syntax and having sentiment knowledge of words are crucial for Twitter sentiment analysis. To represent the semantic information within tweets context, distributed word representation models like Word2Vec[16] and GloVe[19] have been used. Even though compelling improvements have been achieved by using distributed word representation in deep neural network models,

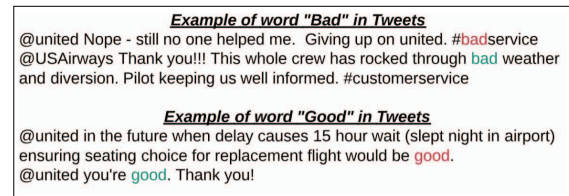


Fig. 1. Words with different meanings and polarities in the context of tweets

there are still some limitations such as inability to identify and handling polysemy as well as noise within the context of tweets. Our research should focus on making sure that learned representations: (i) capture polysemy in the context; (ii) represent complicated attributes of words usage including both semantics and syntax; and (iii) sentiment of words.

Examples of polysemy and words with opposite polarity are shown in Fig. 1 where the meaning of words like good and bad changes according to its context which traditional word embeddings are unable to capture but assign the same representation of a word irrespective of its context and meaning. In addition to polysemy, traditional word embeddings fail to capture sentiment information of words like good and bad which results in similar word vector representations having the opposite polarities. Thus, ignoring polysemy within the context and sentiment polarity of words in a tweet reduces the performance of sentiment analysis.

In this research, we propose DICE (Deep Intelligent Contextualized Embedding) to solve the issues of polysemy, semantics, syntax and sentiment for a Twitter sentiment analysis. The input of DICE is fed to BiLSTM with attention for Twitter sentiment analysis. Also, our intelligent tweets pre-processor compliments our proposed model to learn better features by removing noise from unstructured and informal text.

The experiments are carried out on airline related tweets. The accuracy results for Twitter sentiment analysis, when using our proposed method, is higher than the state-of-the-art methods. Our contributions can be summarized as follows:

- An Intelligent tweets pre-processor is designed to standardize the noisy nature of tweets by spell correction, sentiment aware tokenization, word segmentation and normalization.
- A Deep Intelligent Contextual Embedding (DICE) which addresses the language ambiguity and is devised to

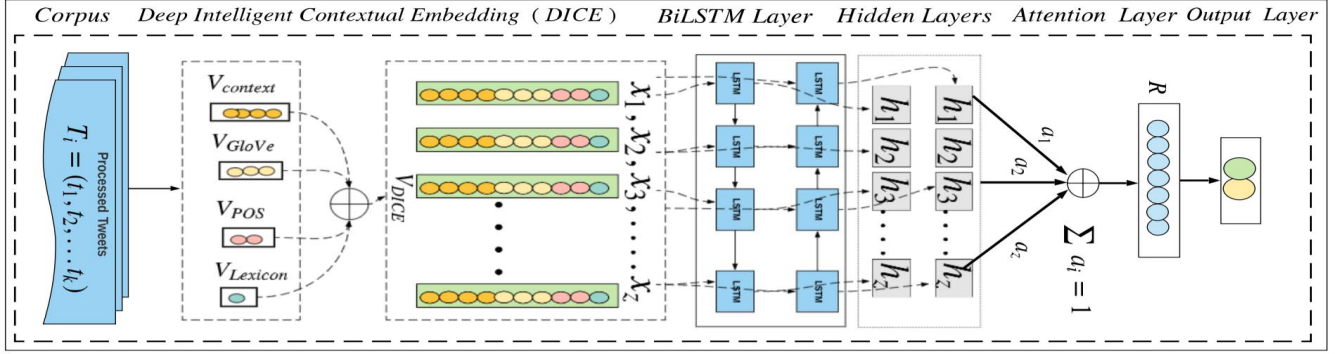


Fig. 2. DICE with BiLSTM and Attention layer

comprehensively capture polysemy in context, semantics, syntax and sentiment knowledge of words.

- Extensive experiments are conducted on several real-world datasets to evaluate above design. All the results prove that our model constantly outperforms other state-of-the-art methods.

The rest of the paper is organized as follows. Section II summarizes the relevant work. Section III describes the model architecture which includes our intelligent pre-processor, deep intelligent contextual embedding and deep neural network with attention. Section IV presents evaluation and analysis of model. Section V provides the conclusion of this research.

## II. RELATED WORK

Sentiment analysis has attracted a lot of interest in the research community. Traditional methods for sentiment classification such as lexicon based methods [5] are simple, computationally economical and expendable but have some limitations such as their reliability on human efforts to label documents which is time consuming, have low coverage and not much effective in case of tweets where text is unstructured and informal. Several researchers claimed that using machine learning methods and hybrid of machine learning with lexicon based gives better performance [6].

Deep learning have played an important role in natural language processing (NLP). Bengio et al.[1] proposed a method which used neural network language model (NNLM) to learn word representations based on prior contexts of every word. After this breakthrough, different investigations were conducted in NLP using deep learning. Mikolov et al.[16] proposed continuous bag of words (CBOW) and skip-gram models which uses a one layer architecture of word embeddings which was based on linear and local context which was solved by dependency based word embeddings and global vectors (GloVe)[19]. Jianqiang et al. [12] improved the accuracy results by using GloVe embedding with deep convolutional neural network (DCNN) for Twitter sentiment analysis, whereas Santos et al.[7] used character to sentence Convolutional Neural Network (CharSCNN) for sentiment analysis of short text which improved the accuracy results. All above methods ignore usage of polysemy in the context, Liu

et al.[14] proposed context sensitive embeddings to overcome the issue of polysemy in general word embeddings which assigns one vector to each individual word. McCann et al. [15] proposed contextualized word vectors (CoVe) by computing contextualized representations using neural machine translation encoder. Most recently, Peters et al.[20] proposed deep contextual word representations for learning complex attributes of a word use in a context.

To integrate the sentiment information into traditional word embeddings, researchers proposed sentiment specific embeddings. Tang et al.[23] proposed several hybrid ranking models (HyRank) and developed sentiment embeddings based on C&W, which considers context and sentiment polarity of tweets. In Yu et al. [26] proposed sentiment embeddings by refining pre-trained embeddings  $Re(*)$  using the intensity score of external knowledge resource. Razaieinia et al [21] proposed improved word vectors (IWV) by combining word embeddings, part of speech (POS) and combination of lexicons for sentiment analysis. Recently, Cambria et al.[2] proposed context embeddings for sentiment analysis by conceptual primitives from text and linked with commonsense concepts and named entities. Unlike above mentioned work, our proposed method is a context sensitive which considers complex attributes of words such as polysemy, semantics, syntax as well as sentiments of words for Twitter sentiment analysis.

## III. PROPOSED MODEL

In this section we will describe our proposed model which is based on A ) Intelligent Pre-Processor, B) Deep Intelligent Contextual Embedding (DICE) C) Bi-directional Long Short Term Memory (BiLSTM) with Attention. The complete architecture of our proposed model is given in Fig.2. At input layer, our model gets an input of a processed tweet. Then in second layer, firstly, all words in a tweet are POS tagged and are assigned a vector. Secondly, using embeddings from language model, vector of each word is extracted which contains polysemy and syntax information and gives us a context embeddings. Thirdly, using GloVe embeddings, vector of words is created which captures word semantics information and the at next step, sentiment score of each word in a tweet is extracted from lexicons and lexicon vector is generated. All

UnProcessed Tweets	Processed Tweets
@virginamerica why <b>don't</b> any of the pairings include red wine. Only white is offered :( <b>#redwineisbetter</b> @united Hey so many <b>tym</b> changes for UA 1534. We going tonight or what? <b>MIA</b>	why <b>do not</b> any of the pairings include red wine only white is offered <b>sad red wine is better</b> hey so many <b>time</b> changes for ua 1534 we going tonight or what <b>Missing In Action.</b>

Fig. 3. An Example our Tweets Pre-Processor which compliments our model to learn better features

four vectors are then concatenated at third layer to produce our deep intelligent contextual embeddings (DICE) which is fed to BiLSTM with attention for tweets sentiment analysis. Below we describe each of the main components.

#### A. Intelligent Pre-processor

Our intelligent tweets pre-processor is designed to remove the noise from informal and unstructured tweets by correction of spelling mistakes, sentiment aware tokenization such as replacing emoticons and slangs with actual words and word segmentation of hashtag words in tweets which compliments our model to learn better features. For sentiment aware tokenization we benefited from Potts's tokenizer<sup>1</sup> which is able to capture basic sentiment related expressions but our tokenizer is also able to identify more recent expressions and slangs being used in social media. For spell correction we borrowed idea proposed by Gimpel et al.[8] but we used Viterbi algorithm which is more effective in our case instead of metaphone algorithm. Lastly, we performed word segmentation to separate words in hashtags, normalized all words, removed punctuations, stops words, mentions (@), URLs and special characters. Examples of our Intelligent pre-processor is shown in Fig. 3.

#### B. Deep Intelligent Contextual Embedding (DICE)

Given a Tweet  $T_i$  with a sequence of tokens  $(t_1, t_2, t_3, \dots, t_k)$ , where  $i$  represents the number of a tweet and  $k$  represents the number of tokens in a tweet. At our deep intelligent contextual embedding layer we concatenated contextual embedding (ELMo), Word Embeddings (GloVe), Part of speech (POS) Embedding and Lexicon embedding. Detail of each embedding is given in the following sections.

1) *Contextual Embedding*: Words representation quality is measured by how it adds syntax information and handle polysemy into a model, which improves semantic word representation. Deep contextual embeddings [20] are embeddings proposed from language model (ELMo) which considers different aspects of words according to its usage in the context.

ELMo embeddings is based on the representation learned from Bi language model (BiLM). Log-likelihood of sentences in both forward and backward language models is involved in training process of BiLMs and final vector is computed after the concatenation of hidden representations from forward language model  $\vec{h}_{n,j}^{LM}$  and backward language model  $\overleftarrow{h}_{n,j}^{LM}$ , where  $j = 1, \dots, L$  and is given by equation 1.

$$BiLM = \sum_{n=1}^k (\log p(t_n | t_1, \dots, t_{n-1}; \Theta_x, \vec{\Theta}_{LSTM}, \Theta_s) + \log p(t_n | t_{n+1}, \dots, t_k; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s)) \quad (1)$$

where  $\theta_x$  and  $\theta_s$  are the token representation parameters and softmax parameters respectively which are shared between forward and backward directions. And  $\vec{\Theta}_{LSTM}$  and  $\overleftarrow{\Theta}_{LSTM}$  are the forward back backward LSTM parameters respectively. ELMo abstracts the representations learned from intermediate layer from BiLM and execute linear combination for each token in a downstream task. BiLM contains  $2L+1$  set representations as given below.

$$R_n = (X_n^{LM}, \vec{h}_{n,j}^{LM}, \overleftarrow{h}_{n,j}^{LM} \mid j = 1, \dots, L) \\ = (h_{n,j}^{LM} \mid j = 0, \dots, L)$$

where  $h_{n,0}^{LM} = x_n^{LM}$  is the layer of token and  $h_{n,j}^{LM} = [\vec{h}_{n,j}^{LM}, \overleftarrow{h}_{n,j}^{LM}]$  for each bi directional LSTM layer. ELMo is a task specific combination of these features where all layers in  $M$  are flattened to single vector and is given by equation 2.

$$ELMo_n^{task} = E(M_n; \Theta^{task}) = \gamma^{task} \sum_{j=0}^L s_j^{task} h_{n,j}^{LM} \quad (2)$$

where  $s^{task}$  are weights which are softmax normalized for the combination of different layers representations and  $\gamma^{task}$  is a hyper parameter for optimization and scaling of Elmo representation. Our architecture is based on a pre-trained ELMo embeddings with a 1,024 dimensions obtained using the 1 Billion Word Benchmark which contains about 800M tokens of news crawl data from WMT 2011 [4]. ELMo gives us context Vector,  $\mathbf{V}_{context}$  of 1024 dimensions, which has the polysemy and syntax information of tweets context.

2) *GloVe Embedding*: GloVe is an unsupervised learning model for obtaining word vector representations by aggregating global word-word co-occurrence statistics which counts how frequently a word appears in a context. GloVe uses ratios of co-occurrence probabilities.

As recommended by Peters et al. [20] that it is favourable to concatenate ELMo embeddings with traditional word embeddings such as Word2Vec and GloVe. In our model we have used pre-trained GloVe embedding of 300 dimensions which are trained on 840 billion token from common crawl because it gives better results as compared to Word2Vec in our case. GloVe outputs a vector,  $\mathbf{V}_{GloVe}$  of 300 dimensions, which has word semantics information of tweets context.

3) *Part of Speech (POS) Embedding*: POS tagging is an important step in which each word in the context is assigned with the appropriate POS tag. Using POS has shown good results in NLP related tasks. POS gives us the useful information about a word, neighbors and different syntactic categories of words such as verbs, nouns, adverbs and adjectives etc. In our proposed model we have used stanford parser for POS

<sup>1</sup> <http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py>

tagging which generates POS tags. Each POS tagged token is then transformed to a vector,  $\mathbf{V}_{POS}$  of 50 dimensions.

4) *Lexicon Embedding*: The lexicon based embedding is based on the extraction of sentiment scores from sentiment lexicon which is a list of words, specific terms and phrases. Using these lexicons can be useful in analyzing the text for sentiment analysis. Each lexicon contains a pair of word-sentiment where each words has its sentiment score which is between  $-1$  to  $1$ , where value less than  $0$  represent negative words and positive for values above  $0$ . There are many sentiment and emotion lexicons resources are available so it is very important to select a right one or appropriate combination of lexicons. We selected a combination of 6 different lexicons after experimenting with different lexicons for extracting sentiments in our lexicon embedding. If any token is not available in any of these lexicons then we assigned a score of zero to that token. Our lexicon embedding outputs a vector,  $\mathbf{V}_{Lexicon}$  of 6 dimensions.

We have used following six Lexicons in our model.

- 1) SenticNet 5.0 [3]
- 2) VADER [11]
- 3) Bing Liu Opinion Lexicon [10]
- 4) SemEval Twitter English Lexicon [18]
- 5) NRC Sentiment140 Lexicon [13]
- 6) Large-Scale Twitter-Specific Sentiment Lexicon [24].

After the creation four vectors individually, we concatenated all of them to get one vector  $\mathbf{V}_{DICE}$ , which is clean and contains polysemy, word semantics, syntax and sentiment knowledge of words in a tweet and given by equation 3.

$$\mathbf{V}_{DICE} = \mathbf{V}_{context} \oplus \mathbf{V}_{GloVe} \oplus \mathbf{V}_{POS} \oplus \mathbf{V}_{Lexicon} \quad (3)$$

where element-wise symbol  $\oplus$  denotes vectors concatenation.

### C. BiLSTM Layer

We have placed BiLSTM layer [22] on top of our DICE with attention layer for sentiment analysis to capture the information from both directions. A BiLSTM takes an input of a vector  $\mathbf{V}_{DICE}$  with a sequence of  $x_z$  tokens and produces hidden representation  $h_i$  at a given time  $i$  by concatenating the hidden representations from both forward  $\vec{h}_i$  and backward  $\overleftarrow{h}_i$  LSTM and is given by equation 4.

$$h_i = [\vec{h}_i \parallel \overleftarrow{h}_i] \quad (4)$$

where  $\parallel$  in equation 4, denotes the concatenation of outputs from both forward and backward LSTM.

### D. Attention layer

Not all words play an equal role in understanding the meaning of the sentence. We used attention mechanism [25] to enforce the contribution of important words. Attention assigns a weight  $a_i$  to each token through a softmax function and finally, representation  $\mathbf{R}$  which is a weighted sum of all tokens is calculated and given by equation 5.

$$\mathbf{R} = \sum_{i=1}^z a_i h_i, \quad (5)$$

TABLE I  
SUMMARY OF ALL PARAMETERS

Name	Details
Loss function	Binary Cross Entropy
DICE dimension	1380
Train-Test Split	0.80-0.20
Optimizer	Adam
Learning Rate	0.001
Back-Propagation	ReLU
batch Size	128
Dropout	0.25
$L_2$ Regularization	0.0001
Hidden Layer Dimension	150 each
Gaussian Noise	$\sigma = 0.3$

where,

$$a_i = \frac{\exp(e_i)}{\sum_{t=1}^z \exp(e_t)}, \quad \sum_{i=1}^z a_i = 1$$

$$e_i = \tanh(W_h h_i + b_h)$$

where  $W_h$  and  $b_h$  are learned parameters,  $h_i$  is the concatenation of the representations of the forward and backward LSTM, introduced in equation 4.

### E. Output Layer

We used representation  $\mathbf{R}$  generated from an attention layer and fed to fully connected softmax layer to get the class probability distribution. We minimized binary Cross-entropy loss function  $\mathbf{L}$  in which loss increases as the predicted probability  $\mathbf{p}$  diverges from the actual label  $\mathbf{y}$ , is given by equation 6. The grid search optimization technique was used to find the optimal parameters which are given in Table I.

$$\mathbf{L} = -(y \log(p) + (1 - y) \log(1 - p)) \quad (6)$$

## IV. EXPERIMENTAL RESULTS

In this section, we describe the datasets and experimental evaluations to show the effectiveness of our proposed model.

### A. Datasets

We have used three airline related Twitter datasets, two of them crawled and labeled by us and one is publicly available. We chose airline related datasets because limited work has been done on airlines sentiment analysis using deep learning methods. We call our three datasets as Dataset 1, Dataset 2 and Dataset 3 and the tweets distribution is given in Table II.

1) *Dataset 1*: This dataset is publicly available and taken from the Kaggle Datasets originally released by CrowdFlower. Total number of tweets given in the dataset is 14,640. Since we are dealing with binary classification problem so we are considering only positive and negative so after filtering the total number of tweets is 11,541. The tweets are related to six major US Airlines: American airline, United airline, US Airways, Southwest airline, Delta airline, and Virgin airline.



TABLE II  
TWEETS DISTRIBUTION IN ALL DATASETS

Dataset Name	Positive	Negative	Total
Dataset 1	2363	9178	11541
Dataset 2	11670	4784	16454
Dataset 3	17860	4312	22172

2) *Dataset 2*: The data was collected by using Tweepy, an official python Twitter API library. Dataset contains only two months worth of tweets starting from December 2015 until January 2016. We followed guidelines by Mohammad et al. in [17] for annotations of tweets and instructed annotators to label tweets as positive and negative. First, a set of 200 tweets were given, four different annotators annotated the tweets collectively so that all of them can have a general understanding and agreement on the standard for annotation. We used Cohens Kappa ( $\kappa$ ) for calculating inter annotator agreement (IAA) between annotators. In the second phase a same set of random 1000 tweets were given to each annotator for annotation. Disagreement was observed and resolved. In the third phase, again another same set of 500 tweets were given to all annotators and this time we achieved good IAA score and minor disagreement was observed. Finally in our last phase, large set of remaining tweets were equally divided among all annotators for annotation of tweets. Total number of tweets in dataset 2 are 16,454. The tweets are related to three airlines in dataset 2 which are Cathay Pacific, United airline and Singapore airline.

3) *Dataset 3* : We used the same method to collect and annotate our third dataset as stated above. Total number of tweets given in our dataset 3 are 22,172. Dataset 3 contains tweets related to Emirates airline.

## B. Performance Evaluation

1) *Baselines*: As a baseline, we compared the performance of our model with approaches mentioned in [9] and [6] but we used TF-IDF for features extraction instead of uni-gram, bi-gram and their combination for different classifiers and their ensemble. We have also compared our model with (i) deep convolutional neural network<sup>2</sup> (DCNN) (embeddings initialized with GloVe word embeddings) [12] and (ii) CharSCNN/SCNN<sup>3</sup> [7] (embeddings initialized with character (CharSCNN) and Word2Vec (SCNN) embeddings) where resulting embeddings are fed to deep neural networks. We have also compared our model with some recently proposed sentiment embeddings such as (i) hybrid ranking [23] which incorporates sentiment and context of tweets for Twitter sentiment analysis (HyRank<sup>4</sup>) and (ii) refined embeddings Re(\*) [26] where researchers refined the traditional word embeddings by using intensity score from lexicon. Finally, we have compared our model with improved word vectors (IWV)[21] where traditional pre-trained word embeddings

<sup>2</sup><https://nlp.stanford.edu/projects/glove/>

<sup>3</sup><https://code.google.com/archive/p/word2vec/>

<sup>4</sup><http://ir.hit.edu.cn/dytang/>

TABLE III  
COMPARISON OF DIFFERENT MODELS WITH PROPOSED MODEL

Model	Dataset 1	Dataset 2	Dataset 3
TF-IDF SVM	.792	.781	.808
TF-IDF Naive Bayes	.831	.836	.827
TF-IDF Decision Tree	.861	.853	.876
TF-IDF Random Forest	.871	.874	.901
Lexicon Based Classifier	.624	.715	.691
Ensemble (NB+SVM+DT+RF)	.816	.831	.829
Ensemble + Lexicon Classifier	.825	.839	.838
DCNN	.839	.846	.853
CharSCNN (Pre-trained)	.865	.862	.875
SCNN (Pre-trained)	.836	.842	.861
CharSCNN (Random)	.810	.821	.839
SCNN (Random)	.820	.830	.847
HyRank-BiLSTM	.848	.846	.868
Re(Word2Vec)-BiLSTM	.853	.852	.872
Re(GloVe)-BiLSTM	.860	.859	.875
IWV	.884	.875	.890
<b>DICE</b>	<b>.936</b>	<b>.931</b>	<b>.939</b>
<b><math>\Delta</math> compared to previous best (IWV)</b>	<b>5.88%</b>	<b>6.40%</b>	<b>5.50%</b>

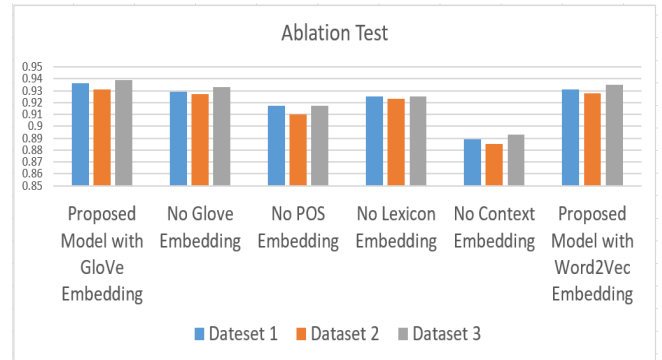


Fig. 4. Ablation Test of proposed model

were enhanced by adding POS and sentiment information from lexicons for sentiment analysis. We selected those methods because they are the state-of-the-art ones and based on the conducted meta-analysis they exhibit the highest accuracy among the techniques developed so far.

2) *Results*: Accuracy results of our model are given in Table III. As we can see that the accuracy of our model is better than existing methods for sentiment analysis when testing them on three, airline related Twitter datasets. Our model achieved better performance because it improves quality of tweets by handling noise, polysemy, semantics, syntax and sentiments within tweet context. DICE improved the accuracy ratio ( $\Delta$ ) by 5.88%, 6.4% and 5.50% when compared to previous best results of embedding based IWV method and by 50%, 30.20% and 35.89% when compared to lowest results of lexicon based classifier on dataset 1, dataset 2 and dataset 3 respectively. As our model offers consistent improvement over all other methods for all tested datasets we can conclude that it is a robust solution for sentiment analysis problem.

3) *Ablation Test*: It is evident from Fig.4 that all embedding layers in our proposed model adds to the overall performance.

