Journal Pre-proof

Transformer based Deep Intelligent Contextual Embedding for Twitter sentiment analysis

Usman Naseem, Imran Razzak, Katarzyna Musial, Muhammad Imran

PII: S0167-739X(20)30306-X
DOI: https://doi.org/10.1016/j.future.2020.06.050
Reference: FUTURE 5716

To appear in: *Future Generation Computer Systems*

Received date : 23 January 2020
Revised date : 22 June 2020
Accepted date : 25 June 2020

Please cite this article as: U. Naseem, I. Razzak, K. Musial et al., Transformer based Deep Intelligent Contextual Embedding for Twitter sentiment analysis, *Future Generation Computer Systems* (2020), doi: https://doi.org/10.1016/j.future.2020.06.050.

Transformer based Deep Intelligent Contextual Embedding for Twitter Sentiment Analysis

Usman Naseem[a], Imran Razzak[b], Katarzyna Musial[a], Muhammad Imran[c]

[a] *Advanced Analytics Institute, University of Technology, Sydney, Australia*
[b] *School of Information Technoology, Deakin University, Geelong, Australia*
[c] *College of Applied Computer Science, King Saud University, Riyadh, Saudi Arabia*

Abstract

Along with the emergence of the Internet, the rapid development of handheld devices has democratized content creation due to the extensive use of social media and has resulted in an explosion of short informal texts. Although a sentiment analysis of these texts is valuable for many reasons, this task is often perceived as a challenge given that these texts are often short, informal, noisy, and rich in language ambiguities, such as polysemy. Moreover, most of the existing sentiment analysis methods are based on clean data. In this paper, we present $DICE_T$, a transformer-based method for sentiment analysis that encodes representation from a transformer and applies deep intelligent contextual embedding to enhance the quality of tweets by removing noise while taking word sentiments, polysemy, syntax, and semantic knowledge into account. We also use the bidirectional long- and short-term memory network to determine the sentiment of a tweet. To validate the performance of the proposed framework, we perform extensive experiments on three benchmark datasets, and results show that $DICE_T$ considerably outperforms the state of the art in sentiment classification.

Keywords:

*Imran Razzak is the corresponding author. Phone: +61 3 522 73849 (imran.razzak@deakin.edu.au

Email addresses: usman.naseem@student.uts.edu.au (Usman Naseem), imran.razzak@ieee.org (Imran Razzak), katarzyna.musial-gabrys@uts.edu.au (Katarzyna Musial), dr.m.imran@ieee.org (Muhammad Imran)

1. Introduction

Over the years, people have posted their opinions, thoughts, or attitudes on social media platforms, such as Twitter, Live Journal, Facebook, Instagram, and LinkedIn, as well as on instant messaging tools, such as WhatsApp, Skype, and WeChat. These posts include recommendations and real-time feedback on products. The proliferation of these posts has provided researchers with new opportunities to investigate public opinion through social media. Companies have also used such real-time content to understand how their customers perceive their products and to develop the appropriate strategies. Such information can also be used to either address the limitations of a product or observe the people's attitudes before launching a new product. Published contents on social media can be subconsciously classified into positive and negative texts. For example, "your curly and shiny hair look great" will most likely provide positive sentiments given that the words "curly," "shiny," and "great" are most likely positive, whereas the words "hair" and "look" are natural. Therefore, this message brings a positive sentiment. However, using technology to discover the feelings or emotions contained in a social media post is a challenging task. Since 1998, previous studies have explored computational semantic analysis systems that aim to understand the nature or meaning of language.

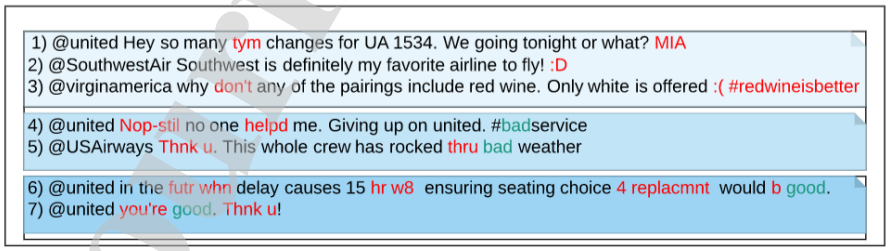

Figure 1: Words with different meanings and polarities in tweets

Although sentiment analysis is a widely investigated topic, the number of

techniques that fully utilize the polysemy, context, semantics, sentiments, and syntax contained in a text remains limited. This case is especially true in the case of short texts, such as tweets. With the limited amount of information contained in these texts (e.g., tweets are limited to 280 characters; however, tweets have 33 characters on average, and only 1% of published tweets actually hit the prescribed character limit), analyzing their content presents a very challenging task especially given that they contain polysemy and figurative words [1, 2]. In addition, the language used on social media is ubiquitous, unstructured, and informal as social media posts often contain abbreviations, misspelled words, non-standard punctuation, slang, and unique words that are impossible to analyze without using an intelligent pre-processing framework. Tweets should also be unitized in terms of their polysemy, syntax, semantics, and sentiment knowledge of words to improve the results of sentiment analysis. Word2Vec[3] and GloVe[4] have been widely used in semantic analysis as they consider the semantic information and distributed word representation in tweets. However, these techniques are unable to take the polysemy and noise contained in tweets into account. Figure 1 shows an unstructured text with polysemy and words with opposite polarity. The meanings of words *"good"* and *"bad"* depend on the context in which they are used. Traditional word embeddings are unable to deal with polysemy and assign the same representation of words irrespective of their context and meaning. Moreover, traditional word embeddings are unable to capture the sentiment of words (i.e., *"good"* and *"bad"* are given the same representation despite having opposite polarity). Polysemy, sentiment polarity, and out of vocabulary (OOV) words (due to the unstructured and low quality of text) negatively impact the results of sentiment analysis. To address these challenges, this work proposes a word representation framework that can capture polysemy in a context, represent the complex attributes of words (including their syntax, semantics, and sentiment), and address issues related to OOV words. Finally, we forward the improve word representation to BiLSTM with attention for sentiment classification. By using a collection of airline-related tweets, we show that the proposed framework considerably outperforms the state of the

art (SOTA). The key contributions of this work are as follows:

- We propose an intelligent tweet pre-processor that removes noise by performing spell correction, sentiment aware tokenization, word segmentation, and normalization.
- We present a novel **Transformer**-based word representation technique fused with **D**eep **I**ntelligent **C**ontextual **E**mbedding (DICE) to address language ambiguity, explore the deep relationships among words, and capture the polysemy, semantics, syntax, and sentiment knowledge of words.
- We perform extensive experiments on several datasets to show that the proposed framework considerably outperforms the state of the art models.

The rest of the paper is organized as follows. Section 2 summarizes the relevant work. Section 3 describes the architecture of the proposed framework. Section 4 evaluates and analyzes the framework. Section 5 concludes the paper.

2. RELATED WORK

Much research interest has been directed toward sentiment analysis, particularly the challenges encountered in detecting word sentiment [5]. While previous studies have attempted to address this problem, they mainly rely on traditional sentiment classification methods, including lexicon-based methods [6], which are economical, expendable, and simple. The limitations in traditional sentiment classification stem from its dependence on human effort in labelling documents, time-intensive activities, low coverage, and limited effectiveness resulting from the colloquial and unstructured text in tweets [7, 8, 9]. Many researchers claim that employing a mix of lexicon-based and machine learning methods can produce improved results [10].

Tweets that express a sentiment or give opinions can be detected by classifiers that incorporate various features, such as machine-learning methods. The lexicon-based method manually or automatically generates a list of negative and

positive terms to deduce any polarity in the message being analyzed [11, 12]. Meanwhile, hybrid approach methods combine these aforementioned approaches to yield performance improvements [13].

Among previous studies on sentiment analysis, Go et al. [14] performed binary classifications where they classify tweets as either positive or negative, applied distant supervision to build a machine-learning classifier and automatically tag tweets, classified positive and negative tweets based on emoticons, and examined and isolated retweets that contain both negative and positive sentiments by using SVM, NB, and MaxEnt classifiers. Pang et al. [15] applied these approaches to analyze movie reviews with unigrams, POS tags, and bigrams and found that both polarity classification and POS tags cannot add negation to unigrams as a specific feature. They argued that using NB with bigrams as features is the most effective method for sentiment analysis with 82.7% accuracy. Barbosa and Feng [16] used a two-step classifier that initially deciphers whether a tweet contains opinions or not and then classifies such tweet as either positive or negative. They also employed three unique sentiment detection tools to obtain the necessary information used for annotating tweets. They obtained a training dataset containing 200,000 tweets by removing those tweets with different sentiment polarities as assigned by various detection tools. They trained their classifiers with syntax (e.g., retweets, URLs, and hashtags) and meta-index features (e.g., POS tags and polarity of words detected by the MPQA lexicon [17]). They normalized the frequency of each feature based on the number of tokens of a tweet. Their SVM classifier achieves accuracies of 81.9% in detecting subjectivity and 81.3% in detecting polarity. They also found that meta-features are crucial in detecting polarity, whereas syntax features are crucial in detecting subjectivity.

In the same vein, Mohammad et al. [18] applied an SVM classifier on a dataset taken from the SemEval-2013 evaluation campaign. Each tweet was represented as a feature vector comprising POS, hashtags, punctuation, word/character n-grams, emoticons, negation, and emphatic lengthening. Their SVM classifier that performed better as compared to the baseline trained on unigrams was

that which was trained using those features. Their classifier obtained F-scores of 69.02% and 89.93% in message- and term-level analyses, respectively. Lexicon and n-grams were identified as the most useful features. Kiritchenko et al. [19] proposed a linear-kernel SVM for sentiment analysis based on an approach that involves a classification of supervised statistical text. They also used various semantic, surface-form, and sentiment features that are derived via a tweet-specific lexicon. This linear-kernel SVM outperformed the MaxEnt classifier. Kouloumpis et al. [20] investigated the utility of various features (especially linguistic ones), applied AdaBoost to detect sentiment polarity, and collected their training data by using existing hashtags in sentiment-indicative tweets. They found that POS is not a good sentiment indicator and that a combination of lexicon, n-grams, and microblogging features shows the best performance in sentiment analysis.

Recent studies have combined several classifiers to improve the performance of individual classifiers. Such ensemble classifier approach has also been applied in analyzing the sentiment of tweets. For instance, Da Silva et al. [10] combined MNB, RF, SVM, and LR and explored the potential of feature hashing and bag of words in feature representation. They found that the ensembles formed by RF, LR, SVM, and MNB can improve classification accuracy. Meanwhile, using feature hashing is less effective than using training ensembles in representing the features of lexicons and bag of words. These approaches were evaluated by using the STS, HCR, Sanders, and OMD datasets.

Polarity- or polarity-score-annotated lists of words have also been leveraged in lexicon-based methods to determine the opinion score of a text in question. These methods do not require any training data. Lexicon-based approaches have been extensively applied in analyzing forum posts, product reviews [21, 22], blogs, and other forms of conventional texts. However, compared with machine-learning methods, lexicon-based approaches have been rarely applied in sentiment analysis due to the unique nature of tweets, which are known for their plethora of peculiarities, colloquialisms (e.g., “gr8” and “YOLO”), and non-static nature (e.g., hashtags and frequently emerging expressions).

SentiStrength [23] is a lexicon-based algorithm that has been widely used in analyzing social media text as this tool can effectively measure the strength of a sentiment contained in informal texts, such as tweets that use human-encoded lexicon and contain phrases or single words that are frequently used on social media. When measuring text sentiment, SentiStrength employs a list of boosting words, emoticons, and negations in conjunction with an almost 700-word sentiment lexicon. MySpace comments were initially used to test this approach. Thelwall et al. [23] extended this algorithm by integrating new sentiment words and idioms into the lexicon and then performed strength boosting via emphatic lengthening. SentiStrength was then compared with many machine-learning methods and tested on six separate datasets, one of which includes tweets. Ortega et al. [24] proposed a three-step Twitter sentiment analysis technique that employs pre-processing, polarity detection, and rule-based classification, of which polarity detection and rule-based classification are based on SentiWordNet and WordNet. This approach obtains favorable results when tested on the SemEval-2013 dataset. However, the effectiveness of this approach has not been compared with that of existing technologies for tweet sentiment analysis. Saif et al. [25] proposed SentiCircles, a lexicon-based approach that evaluates word patterns that co-occur in various contexts, updates word polarity, and pre-assigns scores. They used three separate datasets to evaluate the performance of this approach and found in several experiments that SentiCircles outperforms both SentiWordNet and MPQA-based methods.

Deep learning plays an important role in natural language processing (NLP). Bengio et al. [26] developed the neural network language model to learn the representations of words based on their previous contexts. Following the development of this model, many studies have examined NLP by using deep learning approaches. For instance, Milokov et al. [3] used skip-gram models with a one-layer word embedding architecture and applied continuous bag of words (CBOW) in local and linear contexts that are solved by GloVe [4] and dependency-based word embeddings. Jianqiang et al. [27] found that the results of tweet sentiment analysis can be improved by employing DCNN along with

GloVe embedding. By contrast, Santos et al. [28] used a character-to-sentence deep convolutional neural network to analyze the sentiments of short texts and reported that the application of this technique can enhance the accuracy of sentiment analysis. Given that the aforementioned methods do not consider those contexts with polysemy, Liu et al. [29] proposed context-sensitive embeddings for general word embeddings where one vector is assigned to each word. McCann et al. [30] applied contextualized word vectors that use a neural machine translation encoder to compute contextualized representations. Peters et al. [31] applied deep contextual word representations to learn the complex attributes of words used in a specific context. Xia et al. [32] explored opinion-level contexts with finely defined intra- and inter-opinion features to address the contextual polarity ambiguity in sentiment analysis and applied the Bayesian model to address such polarity in a probabilistic manner.

Sentiment-specific embeddings have also been proposed in the bid to integrate sentiment information into traditional word embeddings. In line with this objective, Tang et al. [33] proposed various HyRank models and CW-based sentiment embeddings that work by considering the sentiment polarity and context of tweets. Meanwhile, Yu et al.. [34] proposed refined pre-trained embeddings (Re(*)) as a novel method for sentiment embedding. Refining is performed based on the intensity score of an external knowledge resource. Razaeinia et al. [35] proposed improved word vectors (IWV) for sentiment analysis that are obtained through a combination of parts of speech, word embeddings, and lexicon. Susanto et al. [36] proposed a new version of "hourglass of emotions" [37] based on the sentiment analysis model of Pulchik. Based on this hourglass, Cambria et al. proposed sentic computing [38] as an interdisciplinary method to fill the gap between statistical NLP and various disciplines, which is necessary for understanding human language. This approach can also be applied to analyze text at the sentence and concept levels via the top-down and bottom-up methods used in NLP. Specifically, when applying the top-down approach, sentic computing takes advantage of certain models, such as semantic networks [39] and conceptual dependency representationss [40], to encrypt meaning, whereas

when applying the bottom-up approach, sentic computing uses sub-symbolic methods, such as deep neural networks, for targeted-aspect-based sentiment analysis by exploiting common knowledge [41] and applies multi-kernel learning [42] to infer syntactic patterns from data. Multi-task learning trains various problems in parallel to enhance the generalization of the network across all problems. Multi-task learning has been used in different NLP problems and has been proven very effective in detecting sentiment and sarcasm [43] and in target-dependent sentiment analysis [44].

Recent studies have classified tweets while addressing the ambiguities of their language by applying deep contextual embedding (DICE), deep contextual word representation, DICE+ (hybrid word representation), and other models [45, 11, 46]. This study serves as an extension of our previous work [45, 11, 47], where we present a powerful and context-sensitive transformer-based words representation that takes complex word attributes, such as semantics, OOV, polysemy, and syntax, into account as well as word sentiments in analyzing tweet sentiment. Compared with SOTA, our proposed model shows improvements in classification performance.

3. PROPOSED MODEL

In this section, we present our proposed sentiment analysis framework DICET that comprises three main components, namely, an intelligent preprocessor, a text representation layer, and a bi-directional long- and short-term Memory (BiLSTM) with attention modelling. The complete architecture of $DICE_T$ is shown in Figure 2. As discussed in the previous sections, tweets often have a complex nature. Therefore, to facilitate sentiment analysis, we propose an intelligent preprocessor that deals with noisy, unstructured, and informal tweets. Given that text ambiguity can lead to misinterpretation, we use the words representation model to concatenate different embeddings, such as bi-directional encoder representations from Transformers (BERT) $V_{Transformer}$, word embeddings V_{GloVe}, contextual embedding $V_{context}$, POS embedding V_{POS}, lexicon

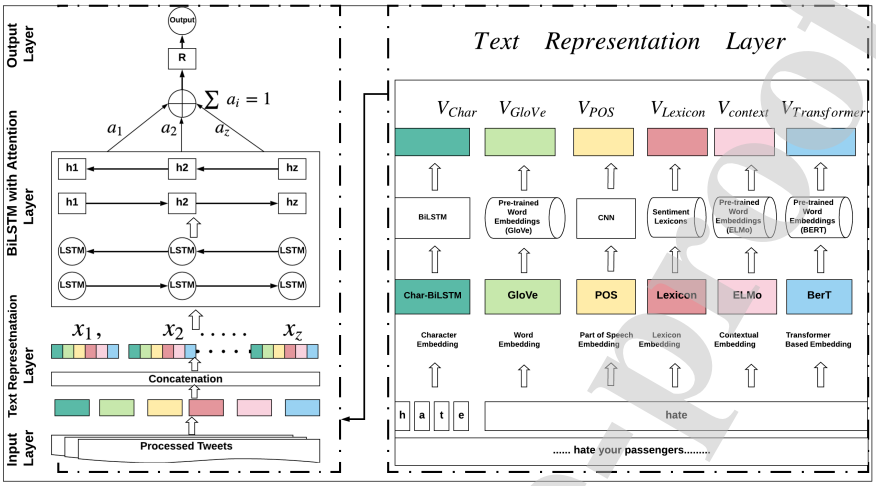

Figure 2: Overview of the proposed model

embedding $V_{Lexicon}$, and character embedding V_{Char}, which considers the polysemy in the context, syntax, and semantics knowledge of words. We describe these components in detail in the following section.

3.1. Input Layer: Intelligent Pre-processor

As discussed in the previous sections, the language used on social media is ubiquitous, unstructured, and very informal at times as they extensively use abbreviations, misspelled words, non-standard punctuation, slang, and words that require extensive preprocessing. We design an intelligent tweets pre-processor that can deal with the noise from informal and unstructured tweets by correcting spelling mistakes, conducting sentiment-aware tokenization (i.e., by replacing emoticons and slangs with actual words), and segmenting hashtags to learn better features. For sentiment-aware tokenization, we use Potts's tokenizerr[1], which can capture basic sentiment-related expressions and identify the recent expressions and slangs being used in social media. To correct misspelled words, we follow the method of Gimpel et al. [48], except that we use the Viterbi algorithm

[1]http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py

instead of the metaphone algorithm to find the most likely sequence of POS tags for unseen sentences. This approach efficiently detects the most probable tag sequence via dynamic programming. Results show that the Viterbi algorithm outperforms the metaphone algorithm. We also apply word segmentation to segment the words from hashtags, normalize words (including stop words), and remove punctuation marks, mentions (@), URLs, and special characters from tweets. Figure 3 shows an example of an intelligent pre-processor.

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

Figure 3: Intelligent pre-processor

3.2. Transformer-Based Deep Intelligent Contextual Embedding

Context may alter the meaning words in a sentence. For instance, the words *good* and *bad* in Figure 1 may have different meanings depending on their context (polysemy). While humans can comfortably understand such language complexities, building a model that can decipher the various nuances and meanings of words based on their context is a difficult task. Traditional word embeddings are also unable to deal with polysemy and assign the same representation of words irrespective of their context and meaning. Traditional word embeddings are also unable to capture the sentiment of words.

A tweet T_x has unique characteristics, including an unstructured and informal nature, low-quality text, similar words with different meanings, words with different sentiments, and a sequence of tokens $(t_1, t_2, ...t_k)$, where x denotes the number of tweets and k represents the number of tokens in a tweet to be classified as label y from a set of fixed labels $y_1, y_2, ...y_k$. Whereas at the input of learner a set of n hand-labeled tweets $(T_1, y_1), ..., (T_n, y_n)$ given and a learned classifier $f(T)$ predicts the label (class) y of a tweet. Our proposed method concatenates representations from the BERT $V_{Transformer}$, applies contextual embeddin $V_{context}$ (ELMo) to deal with polysemy and OOV words, applies word embeddings V_{GloVe} to deal with semantic and syntactical issues, applies part of

speech V_{POS} embedding to effectively capture syntactical information, and applies lexicon embedding $V_{Lexicon}$ to deal with the sentiment knowledge of words in a specific context. Each embedding is discussed in the following sections.

3.2.1. Transformer-Based Embedding

Devlin et al. [49] proposed BERT as a contextualized word representation model that is based on a masked language model and is pretrained by using bidirectional transformers. Given the nature of language modelling where future words cannot be seen, language models (LM) have been previously restricted to a combination of two unidirectional LMs (from right to left and left to right). BERT utilizes masked LM to predict random words that are masked in a sequence and to subsequently learn bidirectional representations. This model also demonstrates the best performance in many natural language processing tasks without requiring huge efforts for task-specific changes. BERT also incorporates information from both directions (bidirectional representations) instead of relying on one direction, which is crucial in words representation in natural language. In our experiments, we use the BERT base model with 12 encoder layers (i.e., transformer blocks) as well as feed-forward networks with 768 hidden units and 12 attention heads to capture transformer-based contextual word representations from both directions. BERT outputs the $\mathbf{V}_{Transformer}$ vector in addressing the problems faced by sequence-to-sequence (seq2seq) methods that use encoder–decoder schemes given that the contextual representations of encoders show uncertainty when dealing with long-range dependencies and handling polysemy.

3.2.2. Contextual Embedding

The quality of a word representation is measured by how such representation adds syntax information and address polysemy in a model. ELMo [31] is an embedding technique based on a language model that considers different aspects of words, including their usage in a specific context. The use of context-based word representation helps address polysemy, capture words with different sentiment

valances, and deal with sarcastic texts.

These embeddings are based on representations learned from the bi-language model (BiLM). Unlike traditional word embeddings, ELMo evaluates the different aspects of words based on their usage in a certain context. Therefore, we use ELMo to identify the same words with different meanings in a post. The log-likelihood of sentences in both forward and backward language models is considered when training BiLM, and the final vector is computed after concatenating the hidden representations from the forward language model $\overrightarrow{h}_{n,j}^{LM}$ and the backward language model $\overleftarrow{h}_{n,j}^{LM}$, where $j = 1, ..., L$, given by eqn. (1).

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

where θ_x and θ_s are the token representation and softmax parameters respectively which are shared between forward and backward directions, and $\overrightarrow{\Theta}_{LSTM}$ and $\overleftarrow{\Theta}_{LSTM}$ are the forward and backward LSTM parameters respectively. ELMo abstracts the representations learned from an intermediate layer from BiLM and then executes a linear combination for each token in a downstream task. BiLM contains *2L+1* set representations as given below.

$$R_n = (X_x^{LM}, \overrightarrow{h}_{n,j}^{LM}, \overleftarrow{h}_{n,j}^{LM} | \quad j = 1, ..., L)$$

$$= (h_{n,j}^{LM} | \quad j = 0, ..., L)$$

where $h_{n,0}^{LM} = x_n^{LM}$ is the layer of tokens and $h_{n,j}^{LM} = [\overrightarrow{h}_{n,j}^{LM}, \overleftarrow{h}_{n,j}^{LM}]$ 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 eqn. (2).

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

where s^{task} are weights which are softmax normalized for the combination of different layer representations and γ^{task} is a hyper-parameter for optimization and scaling of the ELMo representation. Our architecture is based on a pre-trained ELMo embeddings[2] that are obtained by using the one billion word benchmark,

[2]https://tfhub.dev/google/elmo/3

which contains approximately 800M tokens of news crawling data taken from the EMNLP 2011 workshop on statistical machine translation. ELMo yields a word representation that can capture and assign words with different sentiment valances and syntax information of the context to different vectors.

3.2.3. GloVe Embedding

To get the advantage of the considerable data repetition, we use GloVe, which is an unsupervised learning model that obtains word vector representations by aggregating global word-word co-occurrence statistics that show how frequently a word appears in a context, thereby making this model a count-based model [4]. GloVe uses ratios of co-occurrence probabilities, whereas Word2Vec (CBOW and skip-gram) is a prediction-based model that considers only the local context and does not leverage the benefit of the global context. Such limitation prevents Word2Vec from learning substantial scale repetitions and patterns.

According to Peter [31], ELMo embeddings should be concatenated with traditional word embeddings, such as Word2Vec and GloVe. Therefore, in our framework, we use both pre-trained GloVe and Word2Vec embeddings of 300 dimensions that are trained on 840 billion tokens from a common crawl. GloVe provides better results compared with Word2Vec in our case. Specifically, GloVe outputs the vector VGloVe which has 300 dimensions in addition to the word semantics and syntactical information of tweets.

3.2.4. Part of Speech (POS) Embedding

The state-of-the-art word embedding models can capture distributional semantic meanings in different NLP-related tasks. However, these models are unable to incorporate and capture complex semantic representations because some words can have various senses depending on their contexts. Grammatical language characteristics are defined to address these ambiguities. POS helps capture the syntactic characteristics of individual words. POS tagging is an important step in which each word in a defined context is assigned with the appropriate POS tag. This procedure has demonstrated an excellent perfor-

mance in NLP-related tasks [35]. POS gives useful information about a word, its neighbors, and its different syntactic types (e.g., verbs, nouns, adverbs, and adjectives). The primary purpose of POS embedding is to detect nouns/NN, verbs/VB, adjectives/JJ, and adverbs/RB and to generate their respective representations. Figure 4 illustrates the generation of POS word vectors. In our proposed model, we use the Stanford parser for POS tagging. Each POS-tagged token is then transformed into vector $\mathbf{V}_{POS}$ with 50 dimensions

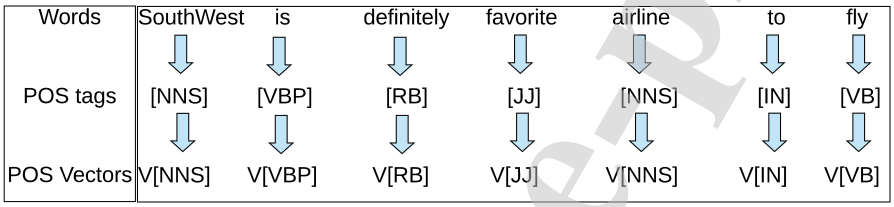

Figure 4: POS embedding V_{POS}

3.2.5. Lexicon Embedding

Words in a language have associated sentiments that can be easily understood by humans but are difficult to interpret by machines. Conventional word representation models are also unable to capture the sentiment of words. To capture such linguistic ambiguity, we use lexicons to extract the sentiments of words and generate the corresponding vectors. The lexicon embedding is based on the sentiment scores extracted from a lexicon dictionary that presents a list of words, terms, and phrases. Each lexicon contains a word–sentiment pair where each word has a sentiment score ranging from -1 to 1, with scores of less and more than 0 represent negative and positive words, respectively. Given that many sentiment and emotion lexicons are available, selecting the correct lexicon or an appropriate combination of multiple lexicons is crucial. We select a combination of six lexicons to extract sentiments in our lexicon embedding. If any token is unavailable in these lexicons, then we assign a 0 score to that token. Our lexicon embedding outputs the vector $\mathbf{V}_{Lexicon}$. Table 1 summarizes the characteristics of sentiment lexicons used in our model, which we describe as

follows:

1. **SenticNet 5.0** [39]:

   SenticNet is a concept-based sentiment lexicon with different versions. The latest version of this lexicon is SenticNet 5.0, which gives the sentiment and semantic information of 100,000 concepts and has been commonly used in sentiment analysis. The parser of this lexicon yields the sentiment score and sentic vector of each concept found in the given text. The sentiment value is a real number, whereas the sentic vector contains scores related to emotions.

2. **VADER** [50]:

   The valence aware dictionary and sentiment reasoner (VADER) lexicon is mainly used to classify the sentiments and intensity of social media text that are sensitive to both positive and negative sentiments. VADER uses the human-centric method and combines qualitative analysis with empirical validation by human raters and crowd knowledge. VADER works well on social media text and does not require any training data. This lexicon is fast and does not suffer from speed–performance tradeoffs.

3. **Bing Liu Opinion Lexicon** [51]:

   The widely used Bing Liu opinion lexicon was introduced by Bing Liu in 2012. This lexicon comprises 2,006 and 4,683 positive and negative words, respectively, including slang, misspelled words, and morphological variants.

4. **SemEval Twitter English Lexicon** [18]:

   SemEval Twitter English Lexicon was presented in semEval-2015 task-10 (subtask E) on Twitter sentiment analysis. This lexicon has sentiment scores ranging from 0 to 1 and contains a list of approximately 1,500 single words, 2-word negation expressions, and their relations with negative and positive polarities. The terms used in this lexicon are taken from Twitter and include general English words, hashtags, misspellings, and other different categories commonly used on Twitter.

5. **NRC Sentiment140 Lexicon** [19]:
NRC-Canada proposed the NRC Sentiment140 Lexicon, which computes the polarity of each word and uses the pointwise mutual information (PMI) measure and sentiment tags of words in a tweet to find the bigrams. Instead of hashtags, a huge corpus of 1.6 million tweets with negative and positive emotions is used to compute the sentiment of words.
6. **Large-Scale Twitter-Specific Sentiment Lexicon (TS-LEX)** [52]:
TS-LEX was built by using the learning representation learning approach. The sentiment information of a text is integrated into a neural network along with its loss function to learn sentiment-specific phrase embedding. An existing phrase embedding model is tailored, and the network is trained from a huge corpus of positive and negative emoticons without annotation. Unlike other lexicons, TS-LEX outputs a word representation that represents a semantic, syntactical, and sentimental relationship among words. An urban dictionary is used to expand a list of the sentiment of common words, and this list is used to train the classifier and determine the probability for a word to be positive or negative.

Table 1: Characteristics of the used sentiment lexicons

Lexicon Name	**Construction based**	**Corpus Used**	**Statsitics**
SenticNet 5.0	Dictionary	WordNet	100K concepts in four categories
VADER	Dictionary	SentiWordNet	75000 lexical features with scores
Bing Liu	Dictionary & Manually	WordNet	Positive words = 2006 Negative words = 4783
SemEval Twitter English Lexicon	Corpus	Tweets	1500 words
NRC Sentiment140 Lexicon	Corpus	Tweets	1.6M Positive & Negative emoticons
TS-LEX	Corpus & Dictionay	WordNet Affect	Positive words with values =1,78,781 Negative words with Values = 1,68,845

3.2.6. Character Embedding:

The prefix and suffix information of any word provides character-level features that can help achieve a closer representation among words of the same category. These features help us deal not only with OOV words but also with unseen words.

We obtain character-level representations by using Bi-LSTMs to produce a character-enhanced embedding for each unique word in a post [53]. In this experiment, we set the maximum character length to 25 and set forward and backward LSTMs parameters to 25, thereby resulting in a 50-dimensional vector that is useful in capturing and handling issues related to OOV words.

After individually creating six vectors, we concatenate these vectors into a single vector $\mathbf{V}_{DICE_T}$, which contains polysemy, word semantics, syntax, and sentiment knowledge of words in a tweet. This vector can be formulated as

$$\mathbf{V}_{DICE_T} = \mathbf{V}_{Transformer} \oplus \mathbf{V}_{context} \oplus \mathbf{V}_{GloVe} \oplus \mathbf{V}_{POS} \oplus \mathbf{V}_{Lexicon} \oplus \mathbf{V}_{Char}$$

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

3.3. BiLSTM with Attention Layer

The recurrent neural network has been successfully applied for sequential data. LSTM is a popular recurrent neural network architecture that outperforms its RNN counterpart in capturing long-term dependencies. The idea was to propose an adaptive mechanism of gating which finalizes the amount to keep the earlier state and remember the obtained features of input data. The forget, input, and output gates are the three gates of a typical LSTM cell that determine the flow of information at the current time step. LSTM processes a sequence $S = (x_1, x_2, ..., x_Z)$, where z is the length of the input text in a sequence. An LSTM cell can be formulated as

$$f_i = \sigma(W_f[x_i, h_{i-1}] + b_f)$$

$$I_i = \sigma(W_z[x_i, h_{i-1}] + b_z)$$

$$C_i = tanh(W_C[x_i, h_{i-1}] + b_C)$$

$$C_i = f_i * C_{i-1} + z_i * C_i$$

$$o_i = \sigma(W_o[x_i, h_{i-1}] + b_o)$$

$$h_i = o_i * tanh(C_i)$$

We place a BiLSTM layer [54] above $DICE_T$ along with an attention layer to capture information from both directions. A BiLSTM layer takes an input of vector $\mathbf{V}_{DICE_T}$ with a sequence of x_z tokens and then produces a hidden representation hi at a given time i by concatenating the hidden representations from both forward $\overrightarrow{h_i}$ and backward $\overleftarrow{h_i}$ LSTMs. This layer can be formulated as

$$h_i = [\overrightarrow{h_i} \parallel \overleftarrow{h_i}]$$

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

Not all words equally contribute to understanding the meaning of a sentence. We therefore adopt the attention mechanism to enforce the contribution of essential words. Attention assigns a weight a_i to each token through a softmax function. The representation $\mathbf{R}$, which is a weighted sum of all tokens, is then calculated as

$$R = \sum_{i=1}^{z} a_i h_i,$$

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.

We use the representation $\mathbf{R}$ generated from an attention layer and feed this representation to a fully connected softmax layer to obtain the class probability distribution. We then minimize the binary cross-entropy loss function $\mathbf{L}$, in which loss increases as the predicted probability $\mathbf{p}$ diverges from the actual label $\mathbf{y}$. This function is formulated as

$$L = -(y \log(p) + (1 - y) \log(1 - p))$$

Table 2: Summary of parameters

Name	Details
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
Gausian Noise	$\sigma = 0.3$

4. Experimental Analysis

We conduct a series of experiments to analyze the performance of our proposed framework. This section initially discusses the experimental settings, the employed datasets, and the experiment results. Afterward, the proposed framework is compared with state-of-the-art word representation models for Twitter sentiment analysis. We perform 10-fold cross-validation and use the accuracy metric to evaluate and compare the performance of these models. Accuracy can be computed as

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

4.1. Parameter settings

We first discuss the parameter settings used in the experiment as follows before presenting the results:

- **Regularization**: : To avoid the overfitting problem, we use Gaussian noise at our input layer and dropout [55] at connections of the network to randomly turn off neurons in the network. Moreover, to avoid over-fitting and ensure the robustness of our method, we adopt the L_2 regularization technique to reduce the size of large weights.

- **Training**: Binary cross-entropy loss function was used to train our model using the rectied linear unit (ReLu) by back-propagation method with the batch size of 128. We use the Adam [56] optimizer to tune the learning rate and apply the 10-fold cross-validation technique to evaluate the classification results.

- **Hyper-parameters**: We use the grid search optimization technique to find the hyperparameters used in our experiments. These parameters are given in Table 2.

4.2. Datasets

We use airline-related datasets given the relatively few studies that conduct airlines sentiment analysis. Specifically, we use the US airlines, airlines, and Emirates airlines datasets. The US airlines dataset is publicly available from CrowdFlower, while we manually crawl and label the airlines and Emirates airlines datasets. We have made all these datasets publicly available. Table 3 presents the statistics of these datasets.

- **US Airline**: This publicly available dataset is taken from the Kaggle datasets originally released by CrowdFlower. This dataset contains 14,640 tweets. We only consider the positive and negative tweets from this dataset given that we are exploring a binary classification problem in this work. After iteration, we obtain a total of 11,541 tweets for our study. These tweets are related to six major US airlines, namely, American Airlines, United Airlines, US Airways, Southwest Airlines, Delta Airlines, and Virgin Airlines.

Table 3: Distribution of tweets in all datasets

Dataset Name	**Positive**	**Negative**	**Total**
US Airline	2,363	9,178	11,541
Airlines	11,670	4,784	16,454
Emirates	17,860	4,312	22,172

- **Airlines Dataset**: The tweets in the airlines dataset are collected by using Tweepy, an official python Twitter API library. This dataset contains only two months' worth of tweets published between December 2015 and January 2016. We follow the guidelines of Mohammad et al. [57] in annotating these tweets and instruct our annotators to label these tweets as either positive or negative. We first obtain a set of 200 tweets and ask four annotators to annotate these tweets collectively in order for them to develop a general understanding and agreement on the standard for annotation. We use Cohen's Kappa(κ) to calculate the inter-annotator agreement (IAA). In the second phase, each annotator is given the same set of 1000 tweets for annotation. Any disagreement arising among these annotators is eventually resolved. In the third phase, each annotator is given the same set of 500 tweets, and a good IAA score and minor disagreement are observed. In the last phase, the remaining tweets are equally divided among all annotators. The airlines dataset has 16,454 tweets related to three airlines, namely, Cathay Pacific, United Airlines, and Singapore Airlines.

- **Emirates Airline** : This dataset contains tweets related to Emirates airlines. We use the same method described above to collect and annotate this dataset. A total of 22,172 tweets are used from this dataset.

Table 4: Ablation analysis of the proposed model

Experiment	Model\Dataset	US Airline	Airlines Dataset	Emirates Airline
Experiment 1	BERT+GloVe+Lexicon+POS	0.909	0.892	0.904
Experiment 2	ELMo+GloVe+Lexicon+POS	0.917	0.909	0.921
Experiment 3	ELMo+GloVe+Lexicon+POS+Character	0.928	0.919	0.936
Experiment 4	BERT+ELMo+GloVe+Lexicon+POS+Character	0.956	0.948	0.962

4.3. Results

We perform an ablation analysis of our proposed model with and without our intelligent pre-processor. Table 4 presents the results for all embedding

layers in this framework. We conduct four experiments by using a combination of different embeddings. In experiment 1, we combine BERT, GloVe, lexicon, and POS, whereas in experiment 2, we replace BERT with ELMo. Results show that ELMo marginally outperforms BERT. In experiment 3, we include character embedding to achieve a closer representation among words of the same category. The prefix and suffix information of any word provides character-level features. As a result, the selected combination can deal with problems related to OOV words. We then consider all possible word embeddings and find that the resultant combination can consider the aforementioned challenges.

Table 4 shows the performance gains of the proposed model on all datasets when the intelligent pre-processor is used. Such gain may be ascribed to the fact that we have worked on improving the quality of the text that can help $DICE_T$ learn better features. The performance of $DICE_T$ slightly drops for all datasets when we exclude BERT. The results of the experimental analysis show a performance drop in both cases if either character embedding or ELMo is excluded. A significant drop in performance is also observed when we exclude ELMo from our representation layer. Figure 5 illustrates these results. The strengths of $DICE_T$ can be ascribed to its combination of different components that ensure diversity and contribute to the increased accuracy of sentiment classification.

4.4. Discussion

We then compare the performance of $DICE_T$ with that of baseline methods. To validate the gain in performance, we select state-of-the-art baseline methods based on our meta-analysis as they have the best accuracy among all techniques available thus far. As a baseline method, we have selected models from weighted representation model, continuous words representation models and finally, hybrid and sentiment specific words representation models which have been used for Twitter sentiment classification task earlier. We used TF-IDF with different machine learning algorithms. We also compare our model with continuous word representation techniques, such as the deep convolutional

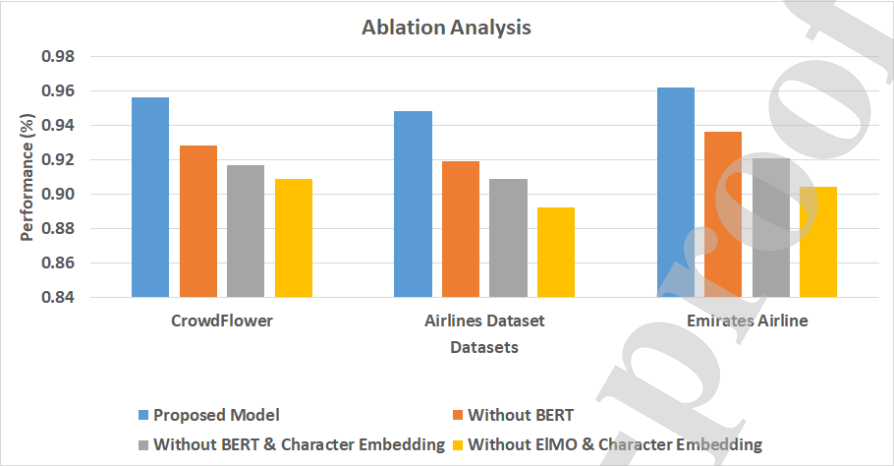

Figure 5: Ablation analysis with different combinations of embeddings

neural network (DCNN)[3], which uses GloVe for word representations [27], and CharSCNN/SCNN4 [28], which utilizes character embedding (CharSCNN) and Word2Vec (SCNN). We also compare the performance of our framework with that of hybrid and sentiment-specific models, such as hybrid ranking [33], refined embeddings Re(*) [58], IWV [35], and transfer-based contextual embedding [45, 11].

Table 5 compares the proposed framework with the state of the art.

$DICE_T$ shows a considerable gain in performance compared with the existing methods for sentiment analysis on all airline-related Twitter datasets subjected to standard pre-processing. $DICE_T$ also has an accuracy ratio (Δ) that is 1.49%, 0.95%, and 1.80% higher than those of other models for the US airlines, airlines, and Emirates airlines datasets, respectively. Given that our proposed framework shows consistent improvements over all other methods, we conclude that $DICE_T$ presents a robust solution to the sentiment analysis problem. The core reasons behind such improvement may include (i) the improvements in the quality of text after noise removal, learning-sentiment-aware tokenization,

[3]https://nlp.stanford.edu/projects/glove/

Table 5: Comparison of the proposed framework with other models

Model\Dataset	US Airline	Airlines Dataset	Emirates Airline
TFIDF with SVM[14]	0.792	0.781	0.808
TFIDF with NB[14]	0.831	0.836	0.827
TFIDF with DT[14]	0.861	0.853	0.876
TFIDF with RF[14]	0.871	0.874	0.901
TF-IDF with Ensemble[10]	0.816	0.831	0.829
DCNN (GloVe)[27]	0.839	0.846	0.853
CharSCNN Pre-trained[28]	0.865	0.862	0.875
SCNN Pre-trained[28]	0.836	0.842	0.861
CharSCNN Random[28]	0.810	0.821	0.839
SCNN Random[28]	0.820	0.830	0.847
HyRank[33]	0.848	0.846	0.868
Re(word2vec)[58]	0.853	0.852	0.872
Re(GloVe)[58]	0.860	0.859	0.875
IWV [35]	0.884	0.875	0.890
DICE [45]	0.936	0.931	0.939
DICE+ [11]	0.942	0.939	0.945
$\mathbf{DICE_T}$(Proposed)	**0.956**	**0.948**	**0.962**
Δ Compared to Previous Best	**1.49%**	**0.95%**	**1.80%**

and spelling correction by using our proposed intelligent pre-processor, which helps learn a better representation, and (ii) handling the language ambiguities by capturing deep relationships within the text. Unlike Word2Vec, GloVe, and Fasttext, $DICE_T$ can handle words with different meanings in the same context (polysemy). Meanwhile, unlike IWV, refined embedding, and HyRank, $DICE_T$ can handle issues related to OOV words and can capture the sentiment knowledge of words. Specifically, $DICE_T$ learns high-quality representations by adding polysemy and sentiment knowledge of words, handling issues related to OOV words, and understanding the semantics and syntactical information of words to obtain improved classification results. For comparison, Table 6 presents

the results with and without using our proposed intelligent pre-processor on all baselines.

Table 6 shows that the accuracy of $DICE_T$ improves across all datasets when we use our proposed intelligent pre-processor. Such improvement validates the robustness of our proposed pre-processing combination. Figure 6 illustrates such performance improvement and Figure 7 visualizes the word cloud of most common words in a) Positive, b) Negative, and c) All Tweets.

Table 6: Performance comparison with and without the intelligent pre-processor

Model\Dataset	With Intelligent Pre-processor			Without Intelligent Pre-processor		
	US Airline	Airlines Dataset	Emirates Airline	CrowdFlower	Airlines Dataset	Emirates Airline
TFIDF with SVM	0.792	0.781	0.808	0.790	0.777	0.802
TFIDF with NB	0.831	0.836	0.827	0.829	0.832	0.821
TFIDF with DT	0.861	0.853	0.876	0.860	0.851	0.872
TFIDF with RF	0.871	0.874	0.901	0.869	0.871	0.897
TF-IDF with Ensemble	0.816	0.831	0.829	0.814	0.830	0.825
DCNN (GloVe)	0.839	0.846	0.853	0.832	0.841	0.848
CharSCNN Pre-trained	0.865	0.862	0.875	0.862	0.858	0.870
SCNN Pre-trained	0.836	0.842	0.861	0.831	0.840	0.857
CharSCNN Random	0.810	0.821	0.839	0.809	0.819	0.835
SCNN Random	0.820	0.830	0.847	0.817	0.829	0.842
HyRank	0.848	0.846	0.868	0.841	0.842	0.863
Re(word2vec)	0.853	0.852	0.872	0.851	0.849	0.868
Re(GloVe)	0.860	0.859	0.875	0.859	0.857	0.871
IWV	0.884	0.875	0.890	0.880	0.872	0.885
DICE	0.936	0.931	0.939	0.915	0.905	0.919
DICE+	0.942	0.939	0.945	0.928	0.919	0.936
DICE_T	0.956	0.948	0.962	0.946	0.941	0.950

We have the following **key observations**

- $DICE_T$ can capture the complex attributes of language, including polysemy, semantics, syntax, and sentiment of words.
- $DICE_T$ captures and complements data characteristics as well as extracts useful features instead of using the one-word representation model.
- A unique combination of sentiment lexicons can assign relevant sentiment to words and improve the classification results.

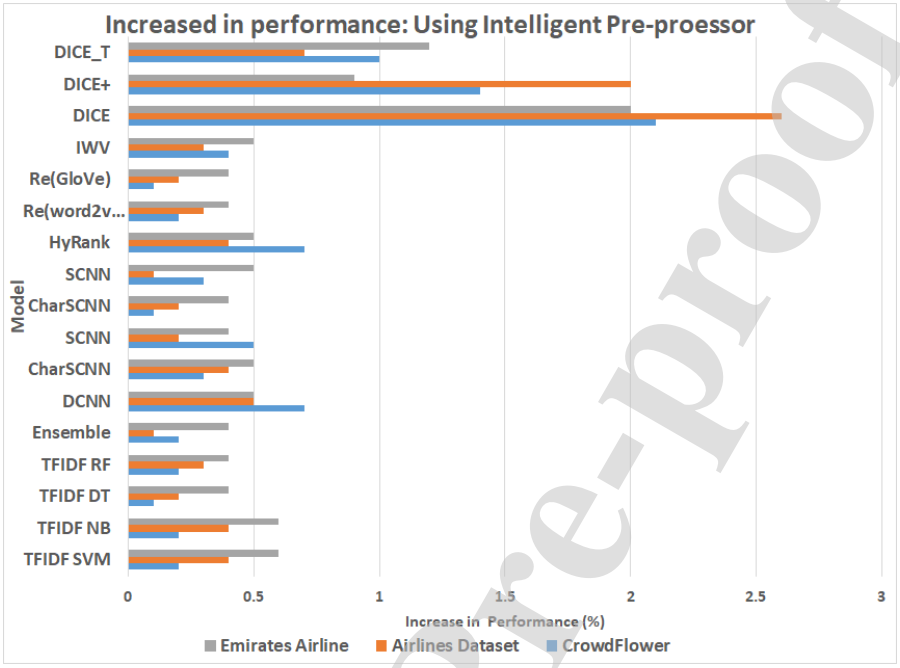

Figure 6: Performance improvements after using the intelligent pre-processor

5. Conclusion

We propose $DICE_T$, which can effectively handle the complex attributes of words, their usage 660 in the noisy context of tweets, and other deep relationships. DICET can also handle language ambiguities, including polysemy, semantics, syntax, OOV words, and sentiment knowledge by learning word representations. Results show that our sentiment analysis framework achieves a considerable gain in performance on the US airlines (+1.49%), airlines (+0.95%), and Emirates airlines (+1.80%) datasets. We also notice a considerable decline in performance when ELMo is excluded from the proposed combination. The experimental results also confirm that $DICE_T$ can effectively manage language complexities and can be used to perform data analytics tasks with low-quality and ambiguous data. In the future, we plan to explore different ways of learning implicit and explicit relationships to capture complex data characteristics,

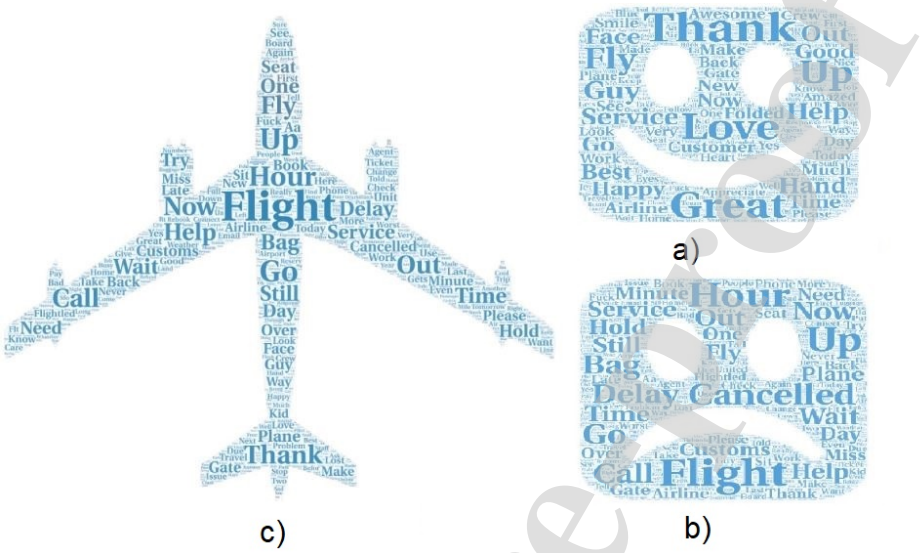

Figure 7: Word Cloud of a) Positive, b) Negative and c) All Tweets

which may help improve the performance of $DICE_T$. Learning the representation of words hierarchically to capture additional information also presents an interesting avenue for future research.

References

[1] U. Naseem, P. Eklund, K. Musial, M. Prasad, Biomedical named-entity recognition by hierarchically fusing biobert representations and deep contextual-level word-embedding, International Joint Conference on Neural Networks.

[2] Z. Saeed, R. A. Abbasi, O. Maqbool, A. Sadaf, I. Razzak, A. Daud, N. R. Aljohani, G. Xu, What's happening around the world? a survey and framework on event detection techniques on twitter, Journal of Grid Computing 17 (2) (2019) 279–312.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: C. J. C.

Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 3111–3119.

[4] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: In EMNLP, 2014.

[5] D. Recupero, E. Cambria, Eswc'14 challenge on concept-level sentiment analysis, Vol. 475, 2014, pp. 3–20. doi:10.1007/978-3-319-12024-9_1.

[6] F. Chiavetta, G. Lo Bosco, G. Pilato, A lexicon-based approach for sentiment classification of amazon books reviews in italian language, 2016, pp. 159–170. doi:10.5220/0005915301590170.

[7] Z. Saeed, R. A. Abbasi, A. Sadaf, M. I. Razzak, G. Xu, Text stream to temporal network-a dynamic heartbeat graph to detect emerging events on twitter, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2018, pp. 534–545.

[8] Z. Saeed, R. A. Abbasi, M. I. Razzak, G. Xu, Event detection in twitter stream using weighted dynamic heartbeat graph approach, arXiv preprint arXiv:1902.08522.

[9] Z. Saeed, R. A. Abbasi, I. Razzak, Evesense: What can you sense from twitter?, in: European Conference on Information Retrieval, Springer, 2020, pp. 491–495.

[10] N. F. da Silva, E. R. Hruschka, E. R. Hruschka, Tweet sentiment analysis with classifier ensembles, Decis. Support Syst. 66 (C) (2014) 170–179. doi:10.1016/j.dss.2014.07.003.
URL http://dx.doi.org/10.1016/j.dss.2014.07.003

[11] U. Naseem, S. K. Khan, I. Razzak, I. A. Hameed, Hybrid words representation for airlines sentiment analysis, in: Australasian Joint Conference on Artificial Intelligence, Springer, 2019, pp. 381–392.

[12] U. Naseem, I. Razzak, P. Eklund, K. Musial, Towards improved deep contextual embedding for the identification of irony and sarcasm, International Joint Conference on Neural Networks.

[13] V. N. Khuc, C. Shivade, R. Ramnath, J. Ramanathan, Towards building large-scale distributed systems for twitter sentiment analysis, in: SAC '12, 2012.

[14] A. Go, R. Bhayani, L. Huang, Twitter sentiment classification using distant supervision, Processing (2009) 1–6.
URL http://www.stanford.edu/~alecmgo/papers/TwitterDistantSupervision09.pdf

[15] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: Sentiment classification using machine learning techniques, in: The Conference on Empirical Methods on Natural Language Processing, Association for Computational Linguistics, 2002, pp. 79–86.

[16] L. Barbosa, J. Feng, Robust sentiment detection on twitter from biased and noisy data, Vol. 2, 2010, pp. 36–44.

[17] J. M. Wiebe, R. F. Bruce, T. P. O'Hara, Development and use of a gold-standard data set for subjectivity classifications, in: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, Association for Computational Linguistics, 1999, pp. 246–53.

[18] S. Mohammad, S. Kiritchenko, X. Zhu, Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets, in: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, 2013, pp. 321–327.
URL http://aclweb.org/anthology/S13-2053

[19] S. Kiritchenko, X.-D. Zhu, C. Cherry, S. Mohammad, Nrc-canada-2014: Detecting aspects and sentiment in customer reviews, in: SemEval@COLING, 2014.

[20] E. Kouloumpis, T. Wilson, J. D. Moore, Twitter sentiment analysis: The good the bad and the omg!, in: ICWSM, 2011.

[21] P. D. Turney, Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 417–424. doi:10.3115/1073083.1073153.
URL https://doi.org/10.3115/1073083.1073153

[22] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, Lexicon-based methods for sentiment analysis, Computational Linguistics 37 (2) (2011) 267–307. arXiv:https://doi.org/10.1162/COLI_a_00049, doi:10.1162/COLI_a_00049.
URL https://doi.org/10.1162/COLI_a_00049

[23] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, A. Kappas, Sentiment strength detection in short informal text, J. Am. Soc. Inf. Sci. Technol. 61 (12) (2010) 2544–2558. doi:10.1002/asi.v61:12.
URL https://doi.org/10.1002/asi.v61:12

[24] R. Ortega Bueno, A. Fonseca Bruzón, Y. Gutiérrez, A. Montoyo, SSA-UO: Unsupervised sentiment analysis in twitter, in: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 501–507.
URL https://www.aclweb.org/anthology/S13-2083

[25] H. Saif, Y. He, M. Fernandez, H. Alani, Contextual semantics for sentiment

analysis of twitter, Information Processing & Management 52 (1) (2016) 5–19.

[26] Y. Bengio, Deep learning of representations: Looking forward, CoRR abs/1305.0445. arXiv:1305.0445.
URL http://arxiv.org/abs/1305.0445

[27] Z. Jianqiang, G. Xiaolin, Deep convolution neural networks for twitter sentiment analysis, IEEE Access PP (2018) 1–1. doi:10.1109/ACCESS.2017.2776930.

[28] C. N. dos Santos, M. A. de C. Gatti, Deep convolutional neural networks for sentiment analysis of short texts, in: COLING, 2014.

[29] P. Liu, X. Qiu, X. Huang, Learning context-sensitive word embeddings with neural tensor skip-gram model, in: Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, AAAI Press, 2015, pp. 1284–1290.
URL http://dl.acm.org/citation.cfm?id=2832415.2832428

[30] B. McCann, J. Bradbury, C. Xiong, R. Socher, Learned in translation: Contextualized word vectors, in: NIPS, 2017.

[31] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, CoRR abs/1802.05365. arXiv:1802.05365.
URL http://arxiv.org/abs/1802.05365

[32] Y. Xia, E. Cambria, A. Hussain, H. Zhao, Word polarity disambiguation using bayesian model and opinion-level features, Cognitive Computation 7 (3) (2015) 369–380.

[33] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, M. Zhou, Sentiment embeddings with applications to sentiment analysis, IEEE Trans. on Knowl. and Data Eng. 28 (2) (2016) 496–509. doi:10.1109/TKDE.2015.2489653.
URL http://dx.doi.org/10.1109/TKDE.2015.2489653

[34] L.-C. Yu, J. Wang, K. R. Lai, X. Zhang, Refining word embeddings using intensity scores for sentiment analysis, IEEE/ACM Trans. Audio, Speech and Lang. Proc. 26 (3) (2018) 671–681. doi:10.1109/TASLP.2017.2788182.
URL https://doi.org/10.1109/TASLP.2017.2788182

[35] S. M. Rezaeinia, A. Ghodsi, R. Rahmani, Improving the accuracy of pre-trained word embeddings for sentiment analysis, CoRR abs/1711.08609. arXiv:1711.08609.
URL http://arxiv.org/abs/1711.08609

[36] Y. Susanto, A. Livingstone, B. C. Ng, E. Cambria, The hourglass model revisited, IEEE Intelligent Systems 35 (5).

[37] E. Cambria, A. Livingstone, A. Hussain, The hourglass of emotions, in: Cognitive behavioural systems, Springer, 2012, pp. 144–157.

[38] E. Cambria, A. Hussain, C. Havasi, C. Eckl, Sentic Computing: Exploitation of Common Sense for the Development of Emotion-Sensitive Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 148–156. doi:10.1007/978-3-642-12397-9_12.
URL https://doi.org/10.1007/978-3-642-12397-9_12

[39] E. Cambria, S. Poria, D. Hazarika, K. Kwok, Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 1795–1802.

[40] S. Poria, E. Cambria, G. Winterstein, G.-B. Huang, Sentic patterns: Dependency-based rules for concept-level sentiment analysis, Knowledge-Based Systems 69 (2014) 45–63.

[41] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm, in: Thirty-second AAAI conference on artificial intelligence, 2018.

[42] S. Poria, I. Chaturvedi, E. Cambria, A. Hussain, Convolutional mkl based multimodal emotion recognition and sentiment analysis, in: 2016 IEEE 16th international conference on data mining (ICDM), IEEE, 2016, pp. 439–448.

[43] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, A. Gelbukh, Sentiment and sarcasm classification with multitask learning, IEEE Intelligent Systems 34 (3) (2019) 38–43.

[44] D. Gupta, K. Singh, S. Chakrabarti, T. Chakraborty, Multi-task learning for target-dependent sentiment classification, CoRR abs/1902.02930. arXiv:1902.02930.
URL http://arxiv.org/abs/1902.02930

[45] U. Naseem, K. Musial, Dice: deep intelligent contextual embedding for twitter sentiment analysis, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2019, pp. 953–958.

[46] U. Naseem, I. Razzak, I. A. Hameed, Deep context-aware embedding for abusive and hate speech detection on twitter, Australian Journal of Intelligent Information Processing Systems 69.

[47] U. Naseem, Hybrid words representation for the classification of low quality text, Ph.D. thesis (2020).

[48] K. Gimpel, N. Schneider, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, N. A. Smith, Part-of-speech tagging for twitter: Annotation, features, and experiments.

[49] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Min-

neapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.
URL https://www.aclweb.org/anthology/N19-1423

[50] C. J. Hutto, E. Gilbert, Vader: A parsimonious rule-based model for sentiment analysis of social media text, in: ICWSM, 2014.

[51] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, ACM, New York, NY, USA, 2004, pp. 168–177. doi:10.1145/1014052.1014073.
URL http://doi.acm.org/10.1145/1014052.1014073

[52] D. Tang, F. Wei, B. Qin, M. Zhou, T. Liu, Building large-scale twitter-specific sentiment lexicon : A representation learning approach, in: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin City University and Association for Computational Linguistics, 2014, pp. 172–182.
URL http://aclweb.org/anthology/C14-1018

[53] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, CoRR abs/1603.01360. arXiv:1603.01360.
URL http://arxiv.org/abs/1603.01360

[54] M. Schuster, K. Paliwal, Bidirectional recurrent neural networks, Trans. Sig. Proc. 45 (11) (1997) 2673–2681. doi:10.1109/78.650093.
URL http://dx.doi.org/10.1109/78.650093

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958.
URL http://jmlr.org/papers/v15/srivastava14a.html

[56] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980.

[57] S. Mohammad, A practical guide to sentiment annotation: Challenges and solutions, in: WASSA@NAACL-HLT, 2016.

[58] Liang-Chih, K. R. Lai, X. Zhang, Refining word embeddings using intensity scores for sentiment analysis, IEEE/ACM Transactions on Audio, Speech, and Language Processing 26 (2018) 671–681.

Highlights

We propose a transformer based word representation by encoding representations from transformer and Deep Intelligent Contextual Embedding (DICE).

DICE enhances the tweet quality by handling noises within contexts, and then integrates six embeddings to involve polysemy in context, semantics, syntax, Out-of-Vocabulary words and sentiment knowledge of words in a tweet.

We proposed word representation 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.

Author Bio

Usman Naseem: Usman is pursing PhD at Advanced Analytics Institute, University of Technology, Sydney, Australia. He has published more than 10 papers in well reputed journals and conferences. His area of research includes social data analytics, natural language processing.

Imran Razzak: Imran Razzak is currently Sr. Lecturer at School of Information Technology at Deakin University, Australia. Imran's research interest is machine learning and data analytics in general, particularly in healthcare industry. He is a passionate health informatician who wants to make the healthcare industry a better place through informatics. He has published more than 70 refereed research publications in international journals and conferences. He is an editorial board member of many reputable international journals as well as session co-chair, session chair and TPC member of dozens of conferences.

Kaska Musial-Gabrys received my MSc in Computer Science from Wroclaw University of Science and Technology (WrUST), Poland, and an MSc in Software Engineering from the Blekinge Institute of

Technology, Sweden, both in 2006. I was awarded my PhD in Computer Science in November 2009 from Wroclaw University of Science and Technology, and in the same year I was appointed a Senior Visiting Research Fellow at Bournemouth University (BU), where from 2010 I was a Lecturer in Informatics. I joined King's in November 2011 as a Lecturer in Computer Science. In September 2015 I returned to Bournemouth University as a Principal Academic in Computing where I was also a Head of SMART Technology Research Group and a member of Data Science Initiative. In September 2017 I joined UTS as Associate Professor in Network Science.

MUHAMMAD IMRAN received the Ph.D. degree in information technology from University Teknologi PETRONAS, Malaysia, in 2011. He is currently an Associate Professor with the College of Applied Computer Science, King Saud University, Saudi Arabia. His research interests include the Internet of Things, mobile and wireless networks, big data analytics, cloud computing, and information security. His research is financially supported by several grants. He has completed a number of international collaborative research projects with reputable universities. He has published more than 150 research articles in top conferences and journals. He has been involved in more than seventy conferences and workshops in various capacities, such as the Chair, the Co-Chair, and a Technical Program Committee Member. European Alliance for Innovation (EAI) has appointed him as the Editor in Chief of EAI Transactions on Pervasive Health and Technology. He serves as an Associate Editor for reputable international journals, such as IEEE COMMUNICATIONS MAGAZINE, Future Generation Computer Systems, IEEE ACCESS, Ad Hoc & Sensor Wireless Networks (Journal) (SCIE), IET Wireless Sensor Systems, the International Journal of Autonomous and Adaptive Communication Systems (Inderscience). He served/serving as a Guest Editor for more than a dozen special issues in journals, such as IEEE COMMUNICATIONS MAGAZINE, Computer Networks (Elsevier), Future Generation Computer Systems (Elsevier), Sensors (MDPI), the International Journal of Distributed Sensor Networks (Hindawi), the Journal of Internet Technology, and the International Journal of Autonomous and Adaptive Communications Systems

Usman Naseem

Imran Razzak

Katarzyna Musial

Muhammad Imran

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

No Competing interest