Received: 17 August 2023 | Revised: 19 December 2023 | Accepted: 4 March 2024

DOI: 10.1111/exsy.13589

ORIGINAL ARTICLE

Expert Systems WILEY

Predicting word vectors for microtext

Iti Chaturvedi[1] | Ranjan Satapathy[2] | Curtis Lynch[1] | Erik Cambria[3]

[1]Information Technology, James Cook University, Townsville, Australia

[2]Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore, Singapore

[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore

Correspondence
Iti Chaturvedi, Information Technology, James Cook University, Townsville, Australia.
Email: iti.chaturvedi@jcu.edu.au

Funding information
James Cook University

Abstract

The use of computer-mediated communication has resulted in a new form of written text called Microtext, which is very different from well-written text. Most previous approaches deal with microtext at the character level rather than just words resulting in increased processing time. In this paper, we propose to transform static word vectors to dynamic form by modelling the effect of neighbouring words and their sentiment strength in the AffectiveSpace. To evaluate the approach, we crawled Tweets from diverse topics and human annotation was used to label their sentiments. We also normalized the tweets to fix phonetic variations, spelling errors, and abbreviations manually. A total of 1432 out-of-vocabulary (OOV) texts and their IV texts made it to the final corpus with their corresponding polarity. To assess the quality of the corpus, we used several OOV classifiers such as linear regression and observed over 90% accuracy. Next, we inferred word vectors using a novel four-gram model based on sentiment intensity and reported accuracy on both open domain and closed domain sentiment classifiers. We observed an improvement in the range of 4–20 on Twitter, Movie and Airline reviews over baselines.

KEYWORDS
microtext, sentiment analysis, word vectors

1 | INTRODUCTION

Artificial intelligence performs emotion recognition using voice, body language, gait, eye tracking and measurements of pulse and breathing rates (Wang et al., 2023). The software '4 Little Trees' can capture individual student reactions during an online lecture using a webcam[1] and microphone. This can help teachers respond in real time and make the class more interactive and personalized. Emotion recognition devices are being installed in public places in the UK. The Piccadilly Circus in London, for example, has a screen with hidden cameras that can automatically gather live information of the audience. It can display advertisements corresponding to large scale analysis of age, gender, and mood.

Psychological literature says that there is no significant relationship between a person's speech and his emotional state (Akhtar et al., 2020). Furthermore, it is not reliable because the same emotions may be expressed in different ways. A scowling expression, for example, is commonly labelled as anger. However, the person may just be confused or trying to concentrate (Stanley, 2018). In fact, people only scowl 30% of the time when they are angry. The emotional response of Americans and Japanese to violent films had significant differences. Japan also has cultural rules about who is allowed to show their emotions in public and to whom and when.

Academic labs and companies are creating large personality databases. To capture high quality interview recordings, it is necessary to have a full-frontal view of the candidate with no occlusions, adequate lighting, and sound proofing. Next, to automate the system and make decisions, you need human labelling of expressions and spoken content so that you can train the AI model. Annotation of the database is often done by a

chosen group of recruiters and may not generalize well to new people (Valdivia et al., 2018). Any bias in the training data will perpetuate into high stake applications such as recruitment. For instance, visual bias in data can include colour, ethnicity, and presence of glasses. The quality of annotation is particularly affected by the presence of micro-texts such as abbreviations and slang in audio recordings. In this paper, we propose a method to predict sentiments in unknown words without the need for annotation.

With the advent of social media, microtext has become omnipresent in today's society. The digital evolution gave microtext a medium to propagate. Natural Language Processing (NLP) techniques are relying on data crawled from the internet, hence it is an absolute necessity to build a corpus for this task. While many NLP researchers and engineers are struggling with the difficulties imposed by bad language (Eisenstein, 2013), there has been relatively little consideration of why the language in social media is so different from our other corpora. According to Jones (2010), the rise of microtext is because of the following reasons:

1. people are unsure of the correct spellings;
2. it is faster;
3. it is become the norm;
4. people want to represent their dialects and accents.

Often various signals, especially sentiments, are grounded in natural language, and due to the noise present in the data, they become quite hard to pick up. A simple affect word such as 'love' might get altered to 'luv' or 'loveee', which creates a massive problem for any NLP task of text.

A robust finding from the sociolinguistics literature is that non-standard forms that mark social variables, such as regional dialects, are often inhibited by informal registers (Labov, 1972). For example, while the Pittsburgh spoken dialect sometimes features the address term yinz (Johnstone et al., 2006), one would not expect to find many examples in financial reports. Similar results for Singaporean dialect, that is, Singlish for NUS SMS data have been reported in Satapathy, Li, et al. (2019). Abbreviations and phonemic-variants are found as geographically-associated lexical items (Eisenstein et al., 2010). A key to the abbreviations is that it can act as constituents, as in 'smh (shake my head) at your ignorance'. Another form of non-standard language is expressive lengthening (e.g., coooollllll), found (Brody & Diakopoulos, 2011) to indicate subjectivity and sentiment. In running dialogues—such as in online dating sites: the symbols '*' and '$' means 'Starbucks'. The affect words, for example, 'luv' and 'lyk' instead of 'love' and 'like' respectively, are being used. To detect sentiments in these texts, we need a corpus with sentiments associated with each text. Thus, we develop a corpus for microtext normalization with the focus on sentiment analysis. The sentences with no OOV words or 'Neutral' polarity were removed. This stringent removal brought the total number of tweets in the corpus to 1432.

The next Section 2 discusses the related work, followed by the corpus annotation procedure in Section 3. We propose a word vector inference approach for OOV in Section 4. Then we employ machine learning algorithms to evaluate the corpus in Section 5. We provide our conclusions in Section 6.

2 | RELATED WORK

Since the beginning of human history, people have been considered by nature as social animals who are highly susceptible to opinions as practically all undertakings and behaviours are influenced by them. Microtext normalization (Satapathy et al., 2020) needs to deal with jango such as 'l8' (late) and 'msg' (message), which are typically self-created and are not yet formalized in linguistics. This section gives a brief introduction to the work done in developing the corpus for microtext normalization and sentiment analysis. In this section, we divide the related work into Corpus 2.1 and Sentiment Analysis 2.2 subsection.

2.1 | Corpus for microtext normalization

To normalize the microtext or bad language, we need models to be trained on such a corpus. There have been efforts around the research community to create a corpus for normalizing microtext. A corpus for machine translation (Wang & Ng, 2013) has been developed by the name NUS SMS corpus[2]. It contains more than 71,000 messages, focusing on Singaporean English (Singlish) and Mandarin Chinese. The framework (Alegria et al., 2014) contains 1164 tweets in the Spanish language for microtext normalization. It includes a set of tweets in Spanish where OOV words and their normalized forms are present. In Lusetti et al. (2018), authors have developed a dataset for Swiss German microtext normalization. They show their results with the character-level Encoder-Decoder model. The authors shared this corpus as a shared task. We found that using an auto-encoder for normalizing tweets has very low accuracy due to the large number of unknown words and small training corpus (Chaturvedi, Su, & Welsch, 2021; Young et al., 2018).

The work of Han and Baldwin (2011) is a well-known reference to the task of lexical normalization of tweets, albeit their study focused on English tweets for one-to-one normalization. But there is some work which concentrates on enhancing the current state-of-the-art NLP tools such as sentiment analysis by using microtext normalization techniques as a preprocessor to the input text. In Satapathy, Singh, and Cambria (2019) the authors propose to tackle phonemic variation class normalization by utilizing the International Phonetic Alphabet (IPA) transformation of input microtext. The works have also depicted the change in the sentiment associated with the text before and after normalization. For the machines to understand 'bad language', we need a corpus to train and test accuracy (Yue et al., 2023). The works cover a wide range of microtext normalization corpus, but they miss out the application to sentiment analysis.

Recently, in Lee and Ashok (2022) looked at the narration of unknown words by a screen reader. The increased use of casual language has made such systems unusable. For unknown words, the reader automatically switches to a spelling model. In Satapathy et al. (2020) the authors review different approaches to microtext normalization. For example, we can perform grammar correction on informal text. Another approach is to translate new words to possible pronunciations and then map them to known words. However, the use of phonetics is restricted to resource-rich languages such as English. Our method shows significant improvement over baselines on multilingual text such as on Twitter.

2.2 | Sentiment analysis

Recent advances in neural language modelling (LM) modelled language to be distributions over sequences of characters (Graves, 2013; Kim et al., 2015; Radford et al., 2017). In Satapathy et al. (2017), authors add Soundex with a lexicon to normalize the microtext and show the increase in accuracy on sentiment analysis. Similar work (Satapathy, Li, et al., 2019) to normalize microtext is present with applications in sentiment analysis tasks. The framework can handle semiotic class normalization, but the accuracy is less for phonemic variation class normalization.

Learning to predict the next character is the basis of OOV normalization. Such models learn internal representations that capture syntactic and semantic properties: even though trained without an explicit notion of word and sentence boundaries, they have been shown to generate grammatically correct text, including words, subclauses, quotes and sentences (Karpathy et al., 2015; Sutskever et al., 2014). More recently, Radford et al. (2017) shows that individual neurons in a large Long-Short-Term-Memory (LSTM) are attributed to special semantic functions, such as predicting sentiment, without explicitly being trained on a sentiment label set.

Julio et al. (2013) describes TASS, an experimental evaluation workshop within SEPLN to foster research in the field of sentiment analysis in social media, specifically focused on the Spanish language. Commonsense knowledge has shown remarkable improvement in OOV tasks. For example, 'If you are injured, you should go to the hospital' (Zhou et al., 2019). ConceptNet is an ontology of commonsense concepts such as 'glasses usedFor reading'. It contains 21 million edges over 8 million nodes showing cause and effect relationships (Liu et al., 2021). Creating a commonsense dataset can be done by both expert knowledge and crowdsourcing from non-experts. Bidirectional Encoder Representations from Transformers (BERT) is a pretrained commonsense embedding model (Devlin et al., 2019).

In Mielke and Eisner (2019) the authors used a recurrent neural network to correct misspelled words. Such a model will not work on abbreviations of short forms such as '2night'. They evaluate their method using a bits per character metric on open vocabulary such as WikiText that does not contain any unknown words. Instead we aim to predict the influence of unknown words on neighbouring words and show significant improvements in accuracy of sentiment prediction in closed domains such as Movie Reviews (Chaturvedi, Thapa, et al., 2021).

In Sasaki et al. (2021) they consider subword semantics to predict rare words. For example, 'untracked' can be predicted from 'tracked'. For this, they use a self-attention mechanism for characters in each word (Chaturvedi et al., 2022). They show that their method is faster than baselines. Similarly, in Prokhorov et al. (2019) the authors align graph embeddings from a semantic network of words with known word embeddings to predict rare words. We feel that character level modelling will increase the complexity of the model, hence we manually curated a corpus for text normalization. For dealing with new words, we use the two neighbouring words on the left and right in a sentence.

3 | A CORPUS FOR ENGLISH MICROTEXT NORMALIZATION

This section discusses the procedure followed to create a corpus for English Microtext normalization (CEMt-Norm). We started by crawling Twitter using Twitter's streaming API. There were no constraints on what keywords to use as a filter. So, the data we got was across the globe and covered different topics, starting from a well-wished tweet to a debate on nuclear energy. We collected around 10K tweets in a day. Next, we employ three annotators to clean each tweet and write it is corresponding in-vocabulary(IV) text. It followed a majority voting to decide the final tweet and its IV text. We dropped the IV text where any of the three annotations were different, and also remove their corresponding tweet from the corpus to maintain coherence. A total of 1432 tweets and their IV text made it to the final corpus. A word of caution is that this corpus contains some usage of foul language as it is collected from public sources.

3.1 | Preprocessing

We used the basic modules such as a dictionary, affix analysis, number and date detection to analyze tokens in the corpus, and a token is considered as an OOV if there is no match in any of the modules. As a first step, we tokenized the sentences to get tokens. The tokenizer's rules were tuned to remove usernames (), hashtags (#), e-mail addresses, URLs, and emoticons. As we aim to detect only the sentiments and not their intensities, emoticons were redundant for this matter. As a second preprocessing step, we used Pyenchant[3] to identify OOV words in the corpus. Pyenchant is used to check the spelling of words and suggest corrections for words that are misspelled. It can use many popular spell-checking packages to perform this task, including ispell, aspell, and MySpell. It is quite flexible at handling multiple dictionaries and multiple languages. The corpus contains sentences with at least one OOV word.

3.2 | Exploration of corpus

The corpus has 1432 OOV text with its corresponding IV text. The corpus also contains the sentiment of the text. The corpus comprises a total vocabulary of 15,798 words. 3541 of 15,798 words are OOV. We calculated the OOV words using Pyenchant tool as a dictionary. The words which return **False** were regarded as OOV. We also calculated the unique set of vocabulary for both total words and OOV. The unique words in total vocabulary are 4679, and unique OOV words are 1861, which is about 40% of total unique words.

Average unique words per sentence: $4679/1432 \approx 3.3$.
Average unique OOV words per sentence: $1861/1432 \approx 1.3$.

3.3 | Corpus annotation

The corpus contains around 1432 text and is annotated by three experts (Cohen's kappa $= 0.79$) in terms of polarity (positive, negative or neutral). We also asked three independent experts to normalize each of the OOV text to IV text. The normalization task is important, so all the reviewers agreeing was important to build a corpus. The final corpus did not contain sentences with at least one expert disagreeing on the normalized text. The agreement is based on exact matching. All the annotators agreed on the final normalization, so no metric to measure inter-annotator agreement needed for normalization. There are 39.9% negative, 38% positive, and 22.1% neutral tweets in this corpus. The two most challenging cases we identified during the annotation process are the normalization of abbreviations and phonetic variation.

1. In the text, '2many of yall gals thinking all the shit' '2' is 'too' but in 'woow im sorry 2 hear dat … i hope she ok' '2' is 'to'.
2. One-to-many forms of abbreviations: like together is written as 2gethr, tgthr, togethr, and many more.
3. We realized there is no standard way to write a word in abbreviations. People rely on creativity to make abbreviations. For example: '*$' means 'Starbucks'. In such cases, experts had a discussion and normalized according to the unified decision.

4 | BI-GRAM SENTIMENT VECTOR TRANSFORMATION

Word vectors or embeddings represent words as a numeric vector. A word vector with dimension d can represent d unique features. This allows words with similar meaning to have similar representations. Features relate words to one another. This allows for vector mathematics such as 'King - man = Queen - woman'. For sentiment analysis, we can use the vectors for clue words belonging to positive and negative emotions, creating a classification manifold. A limitation of such a model is that it is unable to represent sentiments in new, unknown words. Figure 1a illustrates an example of vector mathematics using the word vectors. We consider two features extracted from co-occurrence data. The difference in angles between 'King' and 'Man' is the same as that between 'Queen' and 'Women'. Increasing the number of features or dimensions in vector space will result in higher accuracy of the classifier.

Humans can control the intensity of their emotions during speech. The intensity conveyed by my individual words in a sequence is a reliable way to estimate the intensity of the sentence. Different emotions such as 'Happy' or 'Sad' are characterized by different parameters such as the curvature or the stiffness of their manifold. For example, it is more difficult to identify subtle emotions such as 'sadness' and for new words which almost appear to be neutral. In the past, researchers have looked at the intensity of the entire sentence. However, by looking at each pair of consecutive words in a bi-gram, we can estimate intensities of new unknown words without the need for expensive annotation.

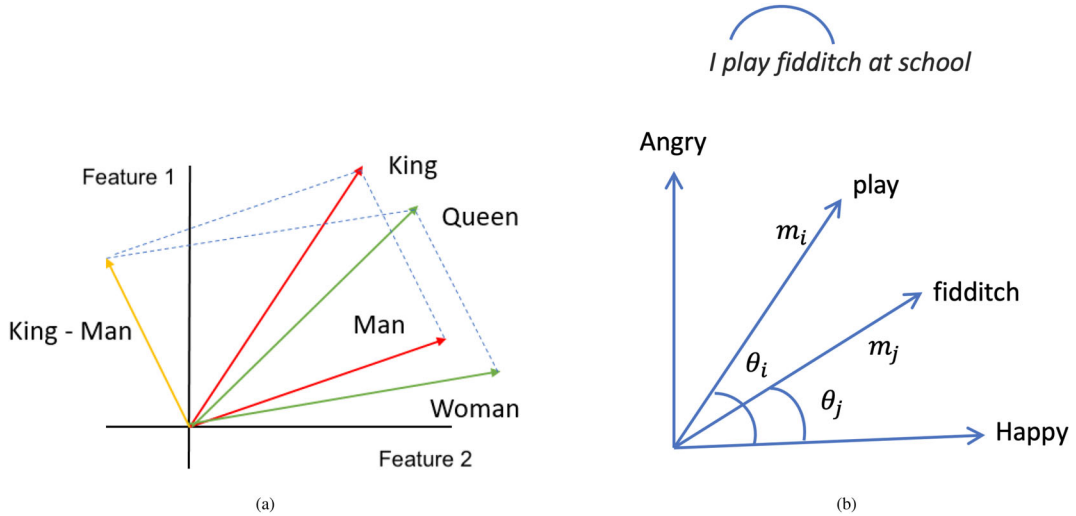

FIGURE 1 (a) Example of word vector mathematics. The two dimensions represent features extracted from co-occurrence frequencies. (b) Vector representation of bi-gram in OOV text. Each dimension is an emotion.

4.1 | Bi-gram control system

We consider a transformation of word vectors for a pair of consecutive words from the original pre-trained Glove embeddings generated from an open-domain corpus. Each word vector has a different angle in vector space θ_i hence the combined effect of the bi-gram can be modelled as a two-inertia control system described by:

$$\begin{aligned} \dot{\theta}_i &= w_i \\ \dot{w}_i &= a_i(t)\left(\theta_j - \theta_i\right) + d_1, \end{aligned} \tag{1}$$

$$\begin{aligned} \dot{\theta}_j &= w_j \\ \dot{w}_j &= a_j(t)\left(\theta_i - \theta_j\right) + b_j(t)u + d_2 \\ a_i(t) &= \frac{k(t)}{m_i(t)}, b_j(t) = \frac{1}{m_j(t)}, \end{aligned} \tag{2}$$

where, w_i is the gradient or frequency of word i in the training data, k is co-occurrence frequency of the two words, m_i is the frequency of word i, u is the intensity of emotion dependent, d_i are constants and t is the sentence index. Here, the dot symbol is the rate of change with respect to the sentences.

Figure 1b illustrates an example of an OOV 'I played fidditch at school'. Each pair of consecutive words is a bi-gram such as 'played fidditch'. Here the word 'fidditch' is OOV and can have multiple meaning based on the context. The word vector representation assumes a numerical representation for each word with an angle θ_i. Each dimension in vector space represents an emotion or feature so that similar words are closer together. For example, here we consider the level of 'Happiness' and 'Anger' as two dimensions for the words. Hence, it is possible to infer word vectors of OOV in bi-grams from its neighbours.

4.2 | AffectiveSpace co-efficients

The pre-trained Glove vector representation for each word assumes that each training sample is independent of the other. In practice, sentences often follow a logical sequence where emotional intensity may increase of decrease. We hence transform the word vectors such that the emotional transforms from 'Happy' to 'Suprise', then progresses to 'Anger' and lastly to 'Sadness'. To determine such a transformation we make use of the AffectiveSpace (Cambria et al., 2011, 2015) of emotions and compute the 'Stiffness (B)', 'Curvature (E)', 'Mode (D)', and 'Variance (C)' of vectors for each of the four emotions. Here 'Stiffness' is computed as the range of values for concepts in AffectiveSpace for each emotion. The 'Mode' is the highest value and 'Variance' is the spread of values. To compute curvature, we consider a two-dimensional plane in the vector space for all concepts and consider the average curvature for the emotion.

TABLE 1 Comparison of sentiment classifiers on OOV text.

Emotion	Stiffness (B)	Curvature (E)	Mode (D)	Variance (C)
Happy	1.67	0.47	0.84	0.04
Surprise	1.65	0.16	0.83	0.07
Angry	1.58	0.79	0.84	0.06
Sadness	1.28	0.71	0.69	0.04

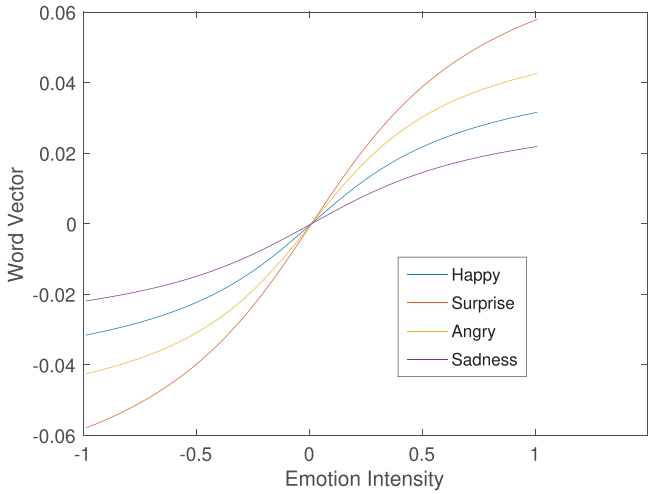

FIGURE 2 We illustrate the effect of emotional intensity on predicted word vectors using parameters from the AffectiveSpace. Sadness has maximum rigidity and Surprise shows highest increase with intensity.

We can see in Table 1 'Stiffness' is highest for 'Happiness' and gradually decreases as we progress to 'Sadness'. This corresponds to the fact that it is difficult to express an emotion such as 'Sadness'. The curvature of the AffectiveSpace for concepts with negative emotions is higher at 0.79 for 'Anger', it is lowest for 'Surprise' with only 0.16 curvature. This could suggest a lack of control, which is the highest for 'Anger'. The peak value is quite similar for all emotions except for 'Sadness' where it is slightly lower at 0.69. Similarly, the variance of values in the vector space is similar for all emotions and, highest for 'Surprise', which has a variance of 0.07. To compute these values we used over 6000 concepts for each emotion and the corresponding AffectivSpace embedding of 100 values. Next, we use the following formula to transform a word vector m_i to a new for $\widehat{m_i}$ (Blundell & Damian Harty, 2015) where u is the intensity of emotion as described in Equation (1):

$$\widehat{m_i}(t) = (m_i(t) \cdot D \cdot \sin(C \cdot \arctan\{Bu - E[Bu - \arctan(Bu)]\}) + m_i(t)) \times 0.5. \tag{3}$$

Figure 2 illustrates the predicted value of word vectors using Equation (3) and Table 1. We can see that the value of word vectors increases as the emotional intensity increases from -1 to $+1$ in AffectiveSpace. The lowest variance is observed for 'Sadness' followed by 'Happiness'. Emotions such as 'Surprise' show a maximum increase in expression with emotional intensity. This is expected as it is more difficult to detect subtle emotions such as 'Sadness' compared to 'Surprise' where a person is excited and, is easier to detect.

With OOV, it is common for word vector representation to be missing or inaccurate in Glove embeddings. Hence, we consider a four-gram model where word vectors for a group of four consecutive words in a sentence are collectively transformed using Equation (1). The weighted sum of the new and old word embeddings is then used for sentiment classification. We refer to the model as Vector Inference of Bi-gram Expressions (VIBE). Figure 3 illustrates an example of the word vector representation for the word 'Hate' and the transformed vector representation using VIBE. We can see that the expression intensity declines as we progress from 'Anger' to 'Sadness' and then 'Happiness'.

4.3 | Parameters

For the pre-trained LDA and BERT models we consider Glove embeddings of length 300 for each word. For training closed domain LSTM we used five-fold cross-validation. The deep learning model consists of a sequence input layer to accommodate sentences of variable lengths. Next, there was a word embedding layer of dimensions equal to vocabulary size. Finally, we had an LSTM layer of 50 neurons followed by the output layer of

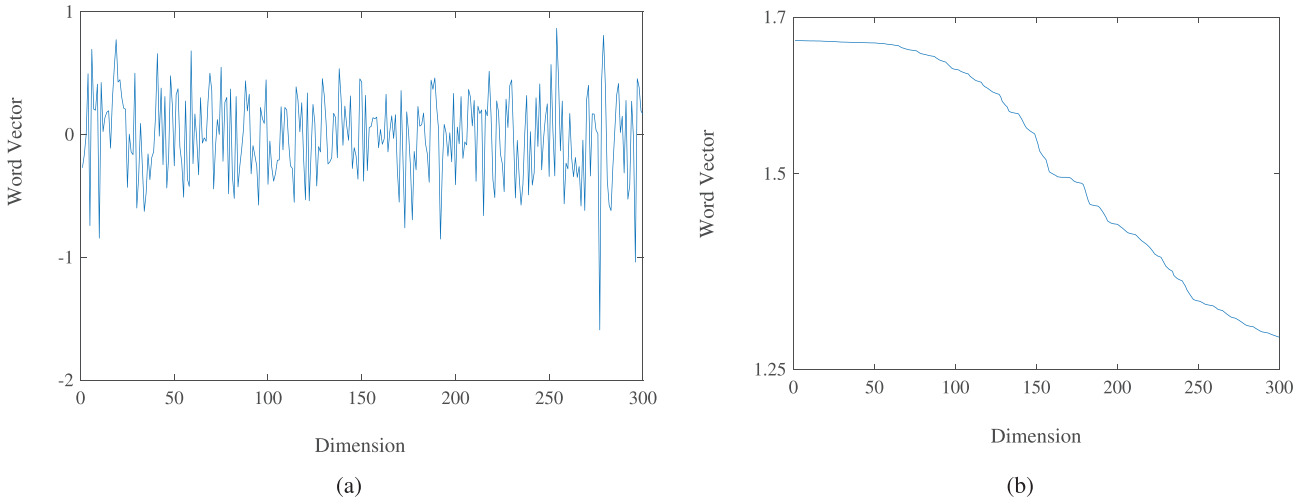

FIGURE 3 Word vector representation for the word 'Hate'. (a) The original vector is from Glove and (b) the transformed vector is inferred using VIBE.

class labels. We used a batch size of 50 sentences and trained the deep neural network until error was below 0.01. For transforming Glove to VIBE word vectors, we used the sentiment co-efficients shown in Table 1. These co-efficients together with Glove word vectors for four consecutive words in a sentence were input to a two-inertia control system described in Section 4.1. The control system was simulated over time by progressively changing the co-efficients in Equation (3) from 'Happy' to 'Sadness'. In this way, static word vectors were transformed to dynamic form where the emotional intensity varies over time.

5 | EVALUATION

In this section we evaluate the methods described above. We first trained a binary classifier to detect microtext in Twitter data. Next, we consider sentiment analysis on the Twitter corpus. We also compare our approach with baselines on movie reviews and live data crawled from a blog on JetStar Airlines. Table 3 compares the *F*-measure of the proposed VIBE classifier with baselines such as pre-trained LDA (Poria et al., 2016) and pre-trained BERT (Devlin et al., 2019). These three baselines used pre-trained Glove embeddings. Next, we used the transformed VIBE embeddings with the pre-trained LDA sentiment classifier shown as VIBE in Table 3. For each sentence, the classifier score is an average over subsequences of four words. We also trained a closed-domain classifier using the LSTM model with Glove word embeddings (Radford et al., 2017). Lastly, we trained the LSTM with the transformed VIBE embeddings resulting in the VIBE-LSTM classifier.

5.1 | Detecting microtext in Tweets

In this subsection, we employ various binary classifiers to detect microtext. Since this algorithm does not require training, and we directly predict the class labels with pre-trained embeddings, hence the execution time is greatly reduced, especially in an online setting where new comments are posted daily. Table 2 shows the results for different classifiers trained on the corpus. We use the term frequency-inverse document frequency (TF-IDF) (Ramos, 2003) approach for the task of feature extraction from a given text. We first split the document into tokens, assigning weights to them based on the frequency with which it shows up in the document along with how recurrent that term occurs in the entire corpus. We used this approach to train four different classifiers. We have used three different features to train and evaluate the models, namely unigram, bigram and trigram. Table 2 shows the accuracy of different classifiers is over 90% for detection of OOV.

5.2 | Predicting sentiments in Tweets

The source of our corpus, that is, Twitter, is used by people mostly to share opinions, which makes it very suitable for sentiment analysis. We first consider the three class problems of 'Positive', 'Negative', and 'Neutral' tweets shown as Twitter3 in Table 3. We observe the highest *F*-measure of 42% by using VIBE-LSTM. This is 8% higher than the pre-trained LDA classifier (36%). It is also 15% higher than BERT and 3% higher than the LSTM model. We observe a slightly higher *F*-measure on the positive class by VIBE (58%) and LSTM performs the best on the neutral class with

TABLE 2 Accuracy of classifiers on the corpus for the task of detecting (OOV) microtext in CEMt-Norm Corpus.

Features	LR	Linear SVC	MNB	MaxEnt
Unigram	96.7	98.7	93.4	91.4
Bigram	96.7	98.6	93.9	91.5
Trigram	96.4	98.5	93.6	88.6

TABLE 3 Comparison of *F*-measure of baseline sentiment classifiers on OOV text.

F-measure	Method	Positive	Negative	Neutral	Total
Twitter3	LDA	0.5	0.38	0.19	0.36
	BERT	0.23	0.57	0	0.27
	LSTM	0.44	0.57	**0.18**	0.39
	VIBE	**0.58**	0.33	0.17	0.35
	VIBE-LSTM	0.53	**0.57**	0.16	**0.42**
Twitter	LDA	0.58	0.52	-	0.55
	BERT	0.63	0.12	-	0.38
	LSTM	0.46	0.53	-	0.5
	VIBE	**0.57**	0.58	-	**0.58**
	VIBE-LSTM	0.5	**0.61**	-	0.56
	VIBE-LSTM-OOV	0.45	0.51	-	0.48
JetStar	LDA	0.24	0.65	-	0.45
	BERT	0.09	0.25	-	0.17
	LSTM	0.05	0.9	-	0.45
	VIBE	0.14	0.78	-	0.46
	VIBE-LSTM	**0.4**	**0.9**	-	**0.65**
Movies	LDA	0.58	0.61	-	0.60
	BERT	0.52	0.46	-	0.49
	LSTM	0.63	0.64	-	0.63
	VIBE	0.08	0.65	-	0.31
	VIBE-LSTM	**0.68**	**0.69**	-	**0.69**

Note: We observe 20% improvement on JetStar dataset. Bold values are the highest in each column.

18%. For the negative class VIBE-LSTM has the highest *F*-measure of 57%. Table 4 reports the comparisons using an accuracy metric. Here again, VIBE-LSTM has the highest accuracy of 61% which is 10% higher than pre-trained LDA (51%). On all three classes positive (55%), negative (57%) and neutral (70%) VIBE-LSTM has the highest accuracy compared to other baselines.

Next, we consider the two class problem of 'Positive' and 'Negative' tweets. Here, VIBE performs the best with an *F*-measure of 58%. This is 3% higher than the pre-trained LDA classifier (55%). Not much improvement was observed compared to the open domain pre-trained sentiment model, since tweets were collected randomly from many topics. It is also 20% higher than BERT and 8% higher than the LSTM model. VIBE-LSTM shows a slightly lower *F*-measure. However, it has the highest *F*-measure on the negative class (61%). For the positive class highest F-measure is achieved by VIBE of 57%. We also reported the *F*-measure for unnormalized tweets shown as VIBE-LSTM-OOV. We can see that the *F*-measure is 48%, which is 10% lower than VIBE. This highlights the importance of text normalization for sentiment analysis. Table 4 reports the comparisons using an accuracy metric. Here VIBE has the highest accuracy of 58% which is 9% higher than pre-trained BERT (49%). VIBE also has the highest accuracy on both positive and negative class of 58%.

5.3 | Classification of movie reviews

The movie review dataset contains 1000 positive and 1000 negative movie reviews from IMDB prior to 2002 (Pang & Lee, 2004). The reviews are from 312 different authors with a cap of 20 reviews per author. Each review is labelled using the corresponding star rating for the review. We

TABLE 4 Comparison of accuracy of baseline sentiment classifiers on OOV text.

Accuracy	Method	Positive	Negative	Neutral	Total
Twitter3	LDA	0.45	0.51	0.56	0.51
	BERT	0.52	0.44	0.65	0.53
	LSTM	0.54	0.53	0.68	0.58
	VIBE	0.49	0.53	0.64	0.55
	VIBE-LSTM	**0.55**	**0.58**	**0.7**	**0.61**
Twitter	LDA	0.56	0.56	-	0.56
	BERT	0.49	0.49	-	0.49
	LSTM	0.5	0.5	-	0.5
	VIBE	**0.58**	**0.58**	-	**0.58**
	VIBE-LSTM	0.57	0.57	-	0.57
	VIBE-LSTM-OOV	0.48	0.48	-	0.48
JetStar	LDA	0.54	0.54	-	0.54
	BERT	0.18	0.18	-	0.18
	LSTM	0.84	0.84	-	0.84
	VIBE	0.66	0.66	-	0.66
	VIBE-LSTM	**0.85**	**0.85**	-	**0.85**
Movies	LDA	0.6	0.6	-	0.6
	BERT	0.49	0.49	-	0.49
	LSTM	0.64	0.64	-	0.64
	VIBE	0.49	0.49	-	0.49
	VIBE-LSTM	**0.68**	**0.68**	-	**0.68**

Note: We observe 31% improvement on JetStar dataset. Bold values are the highest in each column.

consider the two class problem of 'Positive' and 'Negative' reviews shown as Movies in Table 3. We observe the highest *F*-measure of 69% by using VIBE-LSTM. This is 9% higher than the pre-trained LDA classifier (60%). It is also 20% higher than BERT and 6% higher than the LSTM model. VIBE showed lowest *F*-measure on this dataset, this could be because of presence of too many domain specific words. VIBE-LSTM has the highest *F*-measure on both positive (68%) and negative (69%) class. Table 4 reports the comparisons using accuracy metric. Here again VIBE-LSTM has the highest accuracy of 68% which is 8% higher than pre-trained LDA (60%). VIBE-LSTM also has the highest accuracy on both positive and negative class of 68%.

5.4 | Classification of JetStar blog

In order to evaluate VIBE (available on GitHub[4]) on a domain specific dataset we crawled over 5500 comments from the website 'Why we hate JetStar'. This is a public forum where people share their experiences in booking and flying with the Australian domestic airline JetStar. We used ChatGPT to label each comment as 'positive', 'sarcasm', 'negative', and 'neutral'. This resulted in a dataset of 1782 negative, 2790 neutral, 309 positive, and 634 sarcastic comments. We merged negative and sarcastic comments for this analysis and ignored the neutral comments.

We consider the two class problem of 'Positive' and 'Negative' tweets shown as JetStar in Table 3. We observe the highest *F*-measure of 65% by using VIBE-LSTM. This is 20% higher than the pre-trained LDA classifier (45%). It is also 48% higher than BERT and 20% higher than the LSTM model. We observe a much higher *F*-measure of 90% on the negative class. This is because the dataset has many negative reviews. VIBE-LSTM also performs best on the positive class with an *F*-measure of 40%. Table 4 reports the comparisons using an accuracy metric. Here again, VIBE-LSTM has the highest accuracy of 85% which is 29% higher than pre-trained LDA (51%). VIBE-LSTM also has the highest accuracy on both positive and negative class of 58%.

Next, we visualized the vector space for word embeddings of words in the JetStar corpus in Figure 4. We ran a simple *k*-means clustering algorithm to identify clusters of embeddings. The original Glove embeddings are scattered circularly, and it is difficult to identify clusters using the *k*-means. However, the embeddings predicted by VIBE had well differentiated clusters that could be easily identified using *k*-means. We consider the vectors for only words found in the opinion lexicon of positive and negative clue words (Hu & Liu, 2004) in Figure 5. The colour corresponds to a cluster assignment by *k*-means. Figure 5a,b corresponds to negative word clusters. For example, 'awful', 'horrible', and 'nightmare' are close

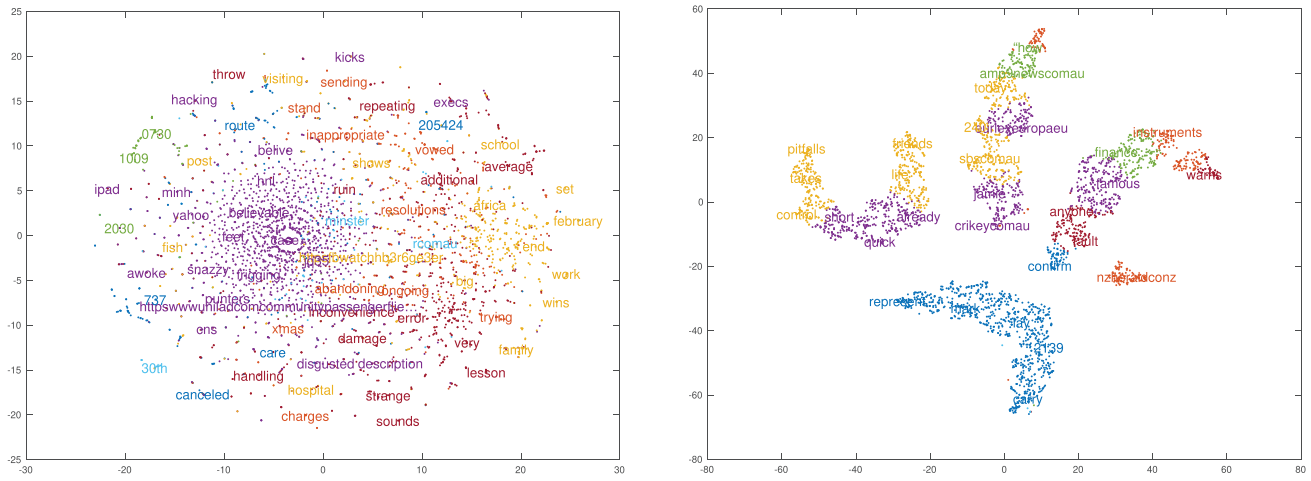

FIGURE 4 We have visualized word embeddings for over 5000 words in the JetStar dataset. (a) Glove embeddings. (b) VIBE embeddings. Colours are determined using *k*-means clustering. We find that VIBE has clearly separated clusters compared to Glove.

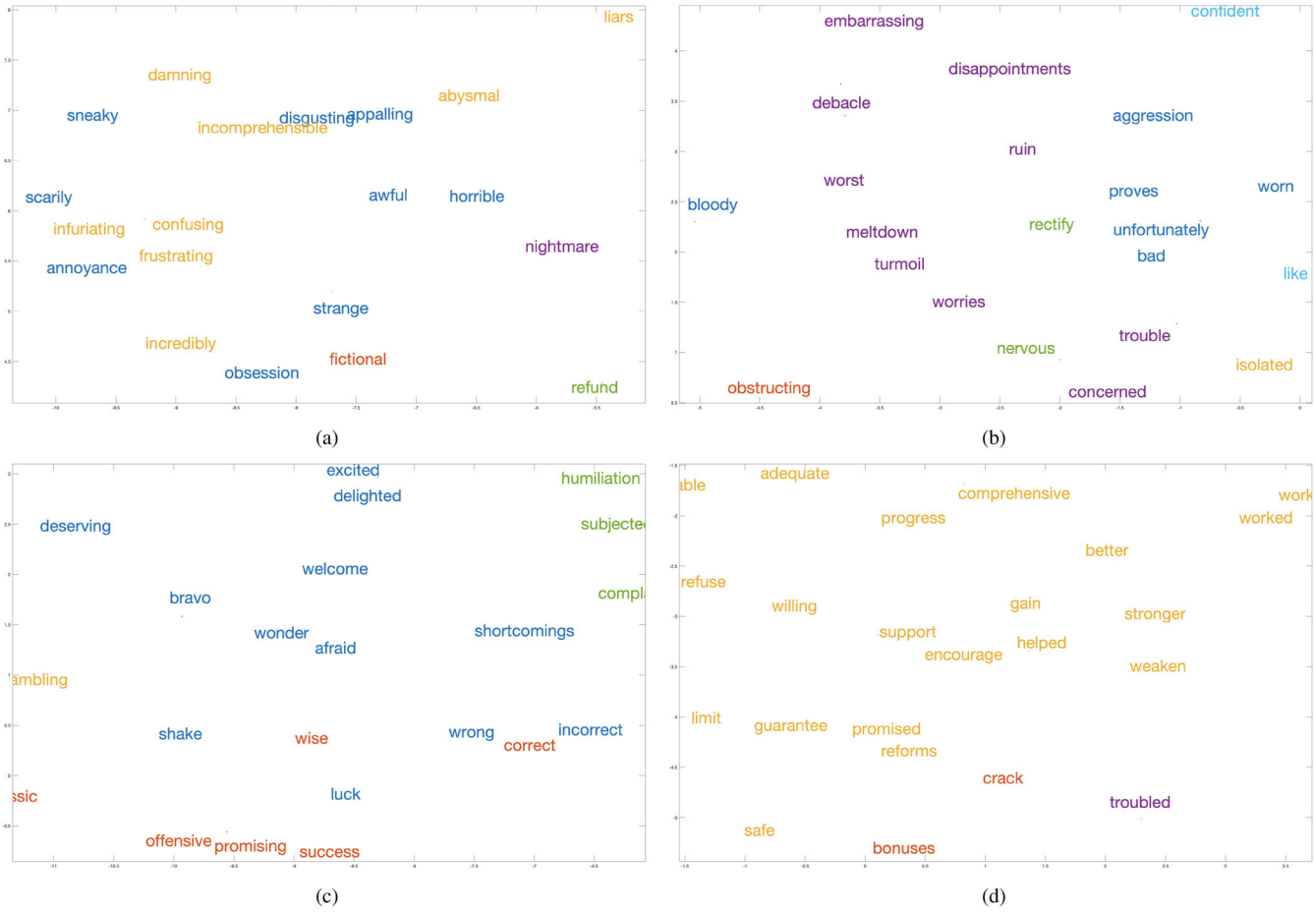

FIGURE 5 Visualization of word vectors for positive and negative clue words shown in Figure 4: (a, b) correspond to negative words and (c, d) correspond to positive words in the JetStar corpus.

to each other in the new embedding. Similarly, Figure 5c,d corresponds to positive word clusters. Here we see words such as 'delighted', 'excited', and 'deserving' close to each other. It has to be noted that the corpus is predominantly negative due to customer complaints, hence fewer positive word clusters are observed. The dataset also had a great number of microtexts, hence this result is significant for large scale analysis of consumer satisfaction.

6 | CONCLUSIONS

Recently, there has been a rise in microtext. This is because people from different cultures used varied accents to represent the same emotions. In this paper, we crawled Tweets from around the world covering different dialects and topics. The corpus is manually annotated and normalized for abbreviations and phonetic variations. Next, we model the intensity of emotions in speech using the AffectiveSpace of emotions. We transform the static Glove word embeddings into a dynamic form when the sentiment transforms from 'Happy' to 'Surprise'. The new word embeddings are used to train an LSTM classifier. On both Twitter and the Movie review dataset we see an improvement of over 8% over pre-trained LDA and over 20% over pre-trained BERT models in *F*-measure. In the JetStar airline reviews we see an improvement of 20% over pre-trained-LDA and over 45% over pre-trained BERT. Hence, we conclude that the proposed approach can overcome the challenges of new unknown words by consider the effect of neighbours on word vectors. In the future we will aim to enlarge the corpus to include additional OOV from different languages. Presence of factual information was another bottleneck to training the classifier. Future work will hence focus on improving multi-class accuracy. Lastly, we find generating normalized text is very time consuming and requires a large training corpus. We would like to extend our work to include this in the future.

ACKNOWLEDGMENT

This research is partially supported by the Ministry of Education, Singapore under its MOE Academic Research Fund Tier 2 (STEM RIE2025 Award MOE-T2EP20123-0005). This work is also partially supported by the College of Science and Engineering at James Cook University. Open access publishing facilitated by James Cook University, as part of the Wiley - James Cook University agreement via the Council of Australian University Librarians.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in GitHub at https://github.com/ichaturvedi/word-vector-inference.

ORCID

Iti Chaturvedi https://orcid.org/0000-0003-4602-2080
Ranjan Satapathy https://orcid.org/0000-0002-0733-7381

ENDNOTES

[1] www.4littletrees.com.
[2] github.com/kite1988/nus-sms-corpus.
[3] https://github.com/rfk/pyenchant.
[4] github.com/ichaturvedi/word-vector-inference.

REFERENCES

Akhtar, M. S., Ekbal, A., & Cambria, E. (2020). How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble. *IEEE Computational Intelligence Magazine*, *15*(1), 64–75.
Alegria, I., Aranberri, N., Comas, P., Fresno, V., Gamallo, P., Padró, L., Vicente, I. S., Turmo, J., & Zubiaga, A. (2014). *TweetNorm_es: An annotated corpus for Spanish microtext normalization* (pp. 2274–2278). LREC.
Blundell, M., & Damian Harty, T. (2015). *Multibody Systems approach to vehicle dynamics*.
Brody, S., & Diakopoulos, N. (2011). *Cooooooooooooooollllllllllll!!!!!!!!!!!!!! Using word lengthening to detect sentiment in microblogs* (pp. 562–570). EMNLP.
Cambria, E., Fu, J., Bisio, F., & Poria, S. (2015). *AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis* (pp. 508–514).
Cambria, E., Mazzocco, T., Hussain, A., & Eckl, C. (2011). *Sentic medoids: Organizing affective common sense knowledge in a multi-dimensional vector space* (Vol. 6677, pp. 601–610).
Chaturvedi, I., Noel, T., & Satapathy, R. (2022). Speech emotion recognition using audio matching. *Electronics*, *11*(23), 3943.
Chaturvedi, I., Su, C. L., & Welsch, R. E. (2021). Fuzzy aggregated topology evolution for cognitive multi-tasks. *Cognitive Computation*, *13*(1), 96–107.
Chaturvedi, I., Thapa, K., Cavallari, S., Cambria, E., & Welsch, R. E. (2021). Predicting video engagement using heterogeneous DeepWalk. *Neurocomputing*, *465*, 228–237.
Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding* (pp. 4171–4186). NAACL.

Eisenstein, J. (2013). *What to do about bad language on the internet* (pp. 359–369). NAACL.
Eisenstein, J., O'Connor, B., Smith, N. A., & Xing, E. P. (2010). *A latent variable model for geographic lexical variation* (pp. 1277–1287). EMNLP.
Graves, A. (2013). *Generating sequences with recurrent neural networks*. preprint, arXiv.
Han, B., & Baldwin, T. (2011). *Lexical normalisation of short text messages: Makn sens a #twitter* (pp. 368–378). HLT.
Hu, M., & Liu, B. (2004). *Mining and summarizing customer reviews* (pp. 168–177). KDD.
Johnstone, B., Andrus, J., & Danielson, A. E. (2006). Mobility, indexicality, and the enregisterment of Pittsburghese. *Journal of English Linguistics*, *34*(2), 77–104.
Jones, L. (2010). *The changing face of spelling on the internet*. Technical report. The English Spelling Society.
Julio, R. V., Lana-Serrano, S., Martínez-Cámara, E., & González, J. C. (2013). *Tass-workshop on sentiment analysis at sepln*.
Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). *Visualizing and understanding recurrent networks*. preprint, arXiv.
Kim, Y., Jernite, Y., Sontag, D. A., & Rush, A. M. (2015). *Character-aware neural language models*. AAAI.
Labov, W. (1972). *Sociolinguistic patterns* (Vol. 4). University of Pennsylvania Press.
Lee, H. N., & Ashok, V. (2022). *Impact of out-of-vocabulary words on the twitter experience of blind users*. CHI.
Liu, C., Cohn, T., & Frermann, L. (2021). *Commonsense knowledge in word associations and conceptnet* (pp. 481–495). ACL.
Lusetti, M., Ruzsics, T., Göhring, A., Samardžić, T., & Stark, E. (2018). *Encoder-decoder methods for text normalization* (pp. 18–28). VarDial.
Mielke, S. J., & Eisner, J. (2019). *Spell once, summon anywhere: A two-level open-vocabulary language model*. AAAI.
Pang, B., & Lee, L. (2004). *A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts* (pp. 271–278). ACL.
Poria, S., Chaturvedi, I., Cambria, E., & Bisio, F. (2016). *Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis* (pp. 4465–4473). IJCNN.
Prokhorov, V., Pilehvar, M. T., Kartsaklis, D., Liò, P., & Collier, N. (2019). *Unseen word representation by aligning heterogeneous lexical semantic spaces* (pp. 6900–6907). AAAI.
Radford, A., Józefowicz, R., & Sutskever, I. (2017). *Learning to generate reviews and discovering sentiment*. CoRR; abs/1704.01444.
Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (Vol. 242). IEEE.
Sasaki, S., Suzuki, J., & Inui, K. (2021). Subword-Based compact reconstruction for open-vocabulary neural word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3551–3564.
Satapathy, R., Cambria, E., Nanetti, A., & Hussain, A. (2020). A review of shorthand systems: From brachygraphy to microtext and beyond. *Cognitive Computation*, *12*(4), 778–792.
Satapathy, R., Li, Y., Cavallari, S., & Cambria, E. (2019). *Seq2Seq deep learning models for microtext normalization* (pp. 1–8). IJCNN.
Satapathy, R., Guerreiro, C., Chaturvedi, I., & Cambria, E. (2017). Phonetic-based microtext normalization for twitter sentiment analysis. In *IEEE ICDMW Sentire*. IEEE.
Satapathy, R., Singh, A., & Cambria, E. (2019). *PhonSenticNet: A cognitive approach to microtext normalization for concept-level sentiment analysis* (pp. 177–188). 11917 of CSoNet.
Stanley, J. (2018). *Experts say 'emotion recognition' lacks scientific foundation*. ACLU News.
Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to sequence learning with neural networks* (pp. 3104–3112). NIPS.
Valdivia, A., Luzon, V., Cambria, E., & Herrera, F. (2018). Consensus vote models for detecting and filtering neutrality in sentiment analysis. *Information Fusion*, *44*, 126–135.
Wang, P., & Ng, H. T. (2013). *A beam-search decoder for normalization of social media text with application to machine translation* (pp. 471–481). NAACL.
Wang, Z., Hu, Z., Ho, S. B., Cambria, E., & Tan, A. H. (2023). MiMuSA—Mimicking human language understanding for fine-grained multi-class sentiment analysis. *Neural Computing and Applications*, *35*(21), 15907–15921.
Young, T., Cambria, E., Chaturvedi, I., Zhou, H., Biswas, S., & Huang, M. (2018). *Augmenting end-to-end dialogue systems with commonsense knowledge* (pp. 4970–4977). AAAI.
Yue, T., Mao, R., Wang, H., Hu, Z., & Cambria, E. (2023). KnowleNet: Knowledge fusion network for multimodal sarcasm detection. *Information Fusion*, *100*, 101921.
Zhou, X., Zhang, Y., Cui, L., & Huang, D. (2019). *Evaluating commonsense in pre-trained language models*. AAAI.

AUTHOR BIOGRAPHIES

Iti Chaturvedi received the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore. She is currently a lecturer of information technology at James Cook University, Townsville, Australia, where she teaches information processing and design thinking. Her research interests include signal processing and artificial intelligence on social media. She appears on Stanford's list of most cited researchers in 2022.

Ranjan Satapathy received the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore. He is currently a research scientist at the Institute of High-Performance Computing, Singapore, where he is working on semantic intelligence and micro text. His research interests include developing algorithms for semantic AI and sustainable finance.

Curtis Lynch received the Bachelor of Information Technology, with Honours from James Cook University, Townsville, QLD, Australia. His honours thesis focused on automated animal tracking using few-shot image recognition. He is currently working as a graduate for a large mining company. His research interests are in deep learning with application to image and text data.

Erik Cambria (Fellow, IEEE) received the Ph.D. degree in computing science and mathematics from the University of Stirling, U.K., in 2012, following the completion of an EPSRC project in collaboration with MIT Media Lab, USA. He is currently a full professor with Nanyang Technological University, Singapore. His research focuses on neuro symbolic AI for explainable natural language processing in domains, such as sentiment analysis, dialogue systems, and financial forecasting. He was the recipient of several awards, including the 2018 AI's 10 to Watch and the 2019 IEEE Outstanding Early Career Award, and was featured in Forbes as One of the five People Building our AI Future.

How to cite this article: Chaturvedi, I., Satapathy, R., Lynch, C., & Cambria, E. (2024). Predicting word vectors for microtext. *Expert Systems*, e13589. https://doi.org/10.1111/exsy.13589