Context-Aware Sentiment Analysis using Tweet Expansion Method

Bashar Tahayna[1,*], Ramesh Kumar Ayyasamy[1] & Rehan Akbar[2]

[1]Department of Information Systems, Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Jalan Universiti, Bandar Barat, 31900 Kampar, Perak, Malaysia
[2]Department of Computer and Information Sciences
Universiti Teknologi, Persiaran UTP, 32610 Seri Iskandar, Perak, Malaysia
*E-mail: bashar.tahayna@1utar.my

Abstract. The large source of information space produced by the plethora of social media platforms in general and microblogging in particular has spawned a slew of new applications and prompted the rise and expansion of sentiment analysis research. We propose a sentiment analysis technique that identifies the main parts to describe tweet intent and also enriches them with relevant words, phrases, or even inferred variables. We followed a state-of-the-art hybrid deep learning model to combine Convolutional Neural Network (CNN) and the Long Short-Term Memory network (LSTM) to classify tweet data based on their polarity. To preserve the latent relationships between tweet terms and their expanded representation, sentence encoding and contextualized word embeddings are utilized. To investigate the performance of tweet embeddings on the sentiment analysis task, we tested several context-free models (Word2Vec, Sentence2Vec, Glove, and FastText), a dynamic embedding model (BERT), deep contextualized word representations (ELMo), and an entity-based model (Wikipedia). The proposed method and results prove that text enrichment improves the accuracy of sentiment polarity classification with a notable percentage.

Keywords: *embedding; neural networks; sentiment analysis; tweet enrichment; deep learning.*

1 Introduction

Twitter and other social microblogging platforms have become key news and information sources [1-3]. Today, the volume of tweets is growing at around 30% per year in estimation.[1] This massive information space has led to a rapid development of sentiment analysis research. The sentiment analysis problem can be simply defined as the act of analyzing a given text within large collections to classify the concealed sentiments.

On Twitter, people express their opinions in natural language by tweeting on a product, service, or any other subject, using multiple linguistic features that

[1] https://www.internetlivestats.com/twitter-statistics

Received November 4th, 2021, Revised April 6th, 2022, Accepted for publication June 7th, 2022.
 ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2022.16.2.3

represent their intention. "Tweets are often shortened and not trivial to understand without their context", according to Setiawan *et al.* [4]. The use of word variations increases the likelihood of vocabulary mismatch and can make tweets difficult to understand. This problem is not straightforward to solve; there are several complicated dimensions to take care of and common issues such as: a) the unstructured nature of the data; b) multi-lingual aspects; c) incomplete sentences; d) idiom, jargon, or ad-hoc words; e) lexical vs semantic vs syntactic attributes; f) ambiguity and implicative referencing or meanings; and many more. Initially, keyword-based sentiment analysis methods were developed to perform global sentiment classification based on bag-of-words frequency models. The difficulty with these approaches is that they do not account for the 'sentimental' links between words in the local context, therefore they usually cannot disclose the transmitted 'implicit meaning'.

In lieu of this, other research efforts have been directed toward utilizing local information within the text. For example, aspect-based sentiment analysis (ABSA) is conducted using aspect-polarity rather than merely the presence of a bag of words within a document. Yet, many existing solutions do not take into account the contextual 'intention' or meaning of words related to the aspect. Linguistic features that create semantic ambiguity and misinterpretation of the user's intention and additional factors, such as non-standard use of written language, slang, abbreviations, or implicit discourse referring to an aspect indirectly, affect the ability to accurately analyze the user's actual intention and the effectiveness of a sentiment analysis system.

To overcome these issues, we propose an enrichment model to expand and relate a given word to its lexical or semantic context. To obtain the expansion terms, external knowledge-based processing (i.e., ontologies, WordNet, Wikipedia, and other online corpora) can be utilized to add semantic models to the text. We believe that when numerous external sources are used in the enrichment process, external knowledge-based tweet expansion will provide the most gain. This is because ontologies like WordNet and ConceptNet provide semantically related words. On the other hand, several corpus-based external sources employ concept proximity to assess term-relatedness [5]. To extract sentiments from a tweet, we adopted various pre-trained word and sentence embeddings without any additional feature engineering. At the same time, we use the same model to evaluate our tweet enrichment representation on the tweet data set [6]. The hope is to obtain a correct and improved polarity classification based on the trained data set by considering the expansion and the contextual 'intention' related to a word.

In this paper, we focus on how to enrich tweets to reduce vocabulary mismatch using contextualized word embeddings or sentence embeddings. We enrich tweets by using different embedding techniques using a state-of-the-art model, which is a hybrid deep learning model that combines CNN and an LSTM network to predict the sentiment from a given text [6]. The F1-score of the proposed enrichment method outperforms the standard baseline representation by a significant margin. This paper is organized as follows. Section 2 discusses issues related to previous research. Section 3 describes our framework for sentiment analysis of tweet messages. Sections 4 and 5 describe the feature extraction and embedding techniques, respectively. Section 6 gives an overview of the hybrid model used in this paper. Finally, the last section provides the experimental results, followed by our conclusions.

2 Related Works

Word2Vec endeavors to connect words with vectors space [7]. The spatial distance between vectors depicts the similitude connection between words. Global Vectors (commonly known as GloVe) is another word embedding model [8]. It learns word representation using a matrix factorization algorithm for word-context matrices. Word2Vec and GloVe both handle whole words, but they struggle with unfamiliar words (out-of-vocabulary words). The context is irrelevant for Word2Vec and GloVe word embeddings. This means that a single word has the same representation even if the context is different. GloVe is simply a better version of Word2Vec. FastText (based on Word2Vec) is a word-fragment-based program that can generally tolerate unknown words while still producing one vector per word [9]. FastText employs n-grams to ensure that local word ordering is preserved. ELMo [9] and BERT [10] handle this issue by providing context-sensitive representations. In other words, f (word, context) gives an embedding in ELMo or BERT. BERT employs a transformer, which is an attention mechanism that learns contextual associations between words (or sub-words) in a text. [10]. ELMo is a character-based system that generates vectors for each character that may be merged using a deep learning model or simply averaged to get a word vector. Hybrid machine learning approaches have recently been proposed as possible models for decreasing sentiment errors on increasingly complicated training data. Combining two (or more) procedures tends to incorporate the benefits of both and as a result addresses some shortcomings of using an individual/singular machine learning model. In order to produce more accurate findings, a hybrid technique was devised in an experiment that used both a support vector machine and semantic orientation [3,11]. In recent years, a new type of deep learning approach has been utilized for sentiment analysis in a variety of languages. The authors of [12] utilized a combined model of CNN and LSTM to solve the sentiment classification problem. The authors of [13] defined a hybrid model based on deep neural network architectures including CNNs and LSTMs, ranking first in all of the five English sub-tasks in SemEval-2017. Cho, *et al.* [14] proposed word representation using high-level feature embedding that works at the character-level CNN and bi-LSTM. They concluded that capturing local and global information of any word token is very efficient to enhance the sentiment polarity estimation. By modeling the interactions of words throughout the composition process, Wang, *et al.* [15] used LSTM for Twitter sentiment categorization. Using the same hybrid model, Guggilla, *et al.* [16] suggested a solution to determine if a given phrase is factual or emotional by using Word2Vec and linguistic embedding features. To improve phrase and sentence representation, Huang, *et al.* [17] proposed encoding syntactic information (for example, POS tags) in a tree-structured LSTM. Yu & Jiang [18] looked further into a generalized solution to phrase embedding. The authors studied cross-domain sentiment classification and came up with an architecture of two separate CNNs to train the hidden feature representations of labeled and unlabeled data sets. In this work, we obtained our results using a state-of-the-art hybrid model [12,15-18] that combines recurrent neural networks (RNN) and convolutional neural networks (CNN).

3 Our Method

Figure 1 depicts the component-based architecture and methodology of the proposed tweet expansion framework. The framework consists of several steps including tweet pre-processing and manipulation, feature extraction, representation, and expansion, lexical simplification, and deep learning model classifier. In the following sections, we give a detailed description of each step.

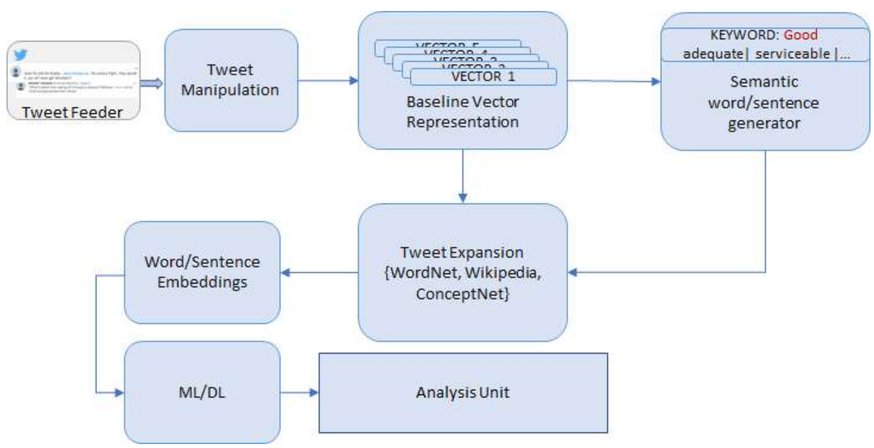


Figure 1 Proposed tweet enrichment and sentiment analysis architecture.

3.1 Tweet Feeder

To build a real-time system that can monitor a specific account on Twitter, we accessed the Twitter API using Python and retrieved all retweets of a specific tweet. For example, a recent tweet by American Airlines and people's replies are shown in Figure 2.

Figure 2 Sample of a tweet and replies from the American Airlines account.

3.2 Tweet Manipulation and Preprocessing

To generate our baseline tweets data set, we manipulated raw tweets by removing noisy URLs and applying some common text normalization procedures:

1. Stop word removal, i.e., eliminating useless encoding of words that are missing in any pre-trained word embedding.
2. Case folding, i.e., converting words or phrases to lowercase.
3. Mapping special values for their type (for example: '3pm' → 'TIME'),
4. Special character removal, i.e., removal of hashtags, numbers, punctuation marks, and characters other than letters of the alphabet.
5. Acronym normalization (for example: 'FB' → 'Facebook') and abbreviation normalization (for example: 'AFAIK' –→ 'As far as I know').
6. Spelling correction.

3.3 Tweet Expansion

We added an extra step to enrich tweet keywords using ConceptNet, WordNet and Wikipedia by producing an extended form of a particular tweet. By adding, removing, or substituting words in the original tweet, the final enrichment phase creates supporting keyword vectors of semantically related phrases (as shown in Figure 3). The ability to recognize term connections allows for a more methodical approach to tweet term selection. There are two types of term relationships: lexical and semantic. In the lexical type, relationships between fundamental word forms with the same meaning are depicted. Synonyms and morphological variations are examples of such relations. On the other hand, hyponyms/hypernyms (i.e., IS-A relationships) and meronyms/homonymies (i.e., HAS-A relationships) indicate semantic relationships between the meanings of words [19]. In addition to these formally defined relationships, word pairs can also be linked depending on their position within a tweet. When such pairings of terms regularly coexist in the same archive, they are considered to be connected to the same topic. Extracting the relationship between formal and informal terms will ensure that all terms related to one aspect of the tweet are taken into account. Tweets can be linked directly/indirectly through certain forwarding relationships. A direct link appears when some tweets contain the same terms, while an indirect relation implies that the tweets have different words, regardless of their relationship (e.g., synonyms) [15].

Substitution or addition of tweet keywords with corresponding synonyms, hyponyms/hypernyms, and meronyms/homonyms are common forms of content change, as described in [20]. We utilized the WordNet ontology to derive lexically and semantically related words. At the same time, we implemented a heuristic function to compute how close two noun keywords are using the distance between Wikipedia articles. For ConceptNet and WordNet, we stored IDs for words that have some form of semantic relation. From Wikipedia, no such relations can be extracted, so we used a set of categories associated with any given article as semantic relations. To measure similarity between words in the WordNet ontology, we use the formula from Leacock and Chodorow [21]:

$$Sim_{wup}(C_1, C_2) = \frac{2*N}{N_1+N_2+2*N} \quad (1)$$

where C1 and C2 represent two different concepts and N1 and N2 evaluate the distances (in IS-A connections) between the concepts and a particular common concept. N measures the distance between the nearest common ancestor of the two concepts and the root node. Table 1 shows a sample of original tweets and semantically similar tweets generated by the BERT transformer. Some terms may not have a corresponding word embedding when the tweet's terms are replaced by vector representations. This problem can be solved by removing the terms or filling their corresponding vectors with random values. A specific word's embedding can also be set to UNK (unknown). Unknown words, on the other hand, will all be represented by the same vector.

Table 1 Relevant keywords extracted from external knowledge bases.

ORIGINAL_TWEET	Highly Scored Paraphrased Sentences
This is so untrue. Literally had THE WORST customer service experience of my life happened yesterday. I was actually trying to be nice & the representative was extremely rude and I felt interrogated when YOU did something wrong.	*- This is completely false. Yesterday, I literally had THE WORST customer service experience of my life. I was attempting to be kind, but the agent was really nasty, and I felt questioned as if I had done something wrong.* *- This is so fake. Literally had the WORST customer service experience of my life came yesterday. I was actually trying to be nice and the rep was exceedingly impolite and I felt questioned when YOU did something wrong.*
American Airlines is terrible with customer service, no solutions to problems and truly make you feel belittled.	*- American Airlines has horrible customer service, offers no answers to problems, and treats you like a second-class citizen.* *- Customer service at American Airlines is atrocious; there are no answers to problems, and you are treated as a criminal.*

	located in the city center	mouth watering food	delicious cuisine	busy area and crowded
located in the city center	1.000000	0.385700	0.293664	0.697138
mouth watering food	0.385700	1.000000	0.632792	0.543945
delicious cuisine	0.293664	0.632792	1.000000	0.373751
busy area and crowded	0.697138	0.543945	0.373751	1.000001

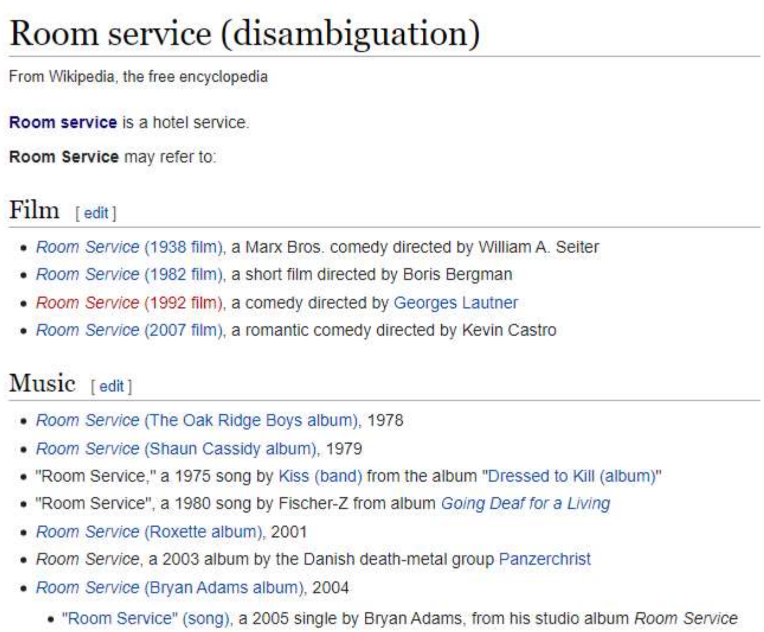
Figure 3 Confusion matrix of BERT transformer-based tweet rephrasing.

To create an embedding for missing words, we take a different approach; we examine tweets based on their linguistic features, i.e., three key linguistic features are retrieved (morphological, syntactical, and semantics). The use of both formally and informally linked terms in tweet expansion in combination with the other variables described above is expected to reduce the problem of tweet vocabulary mismatch. Next, three-gram combinations are extracted from unknown terms and the word embedding model is used to look up each gram. By doing this, we can compute the average of the discovered grams' embeddings. However, if none of the word's three-grams have any embedding, we try higher grams (*4, 6,…,n grams*) until we discover the matching embedding or until we reach the length of the unknown word. In the second situation, we look for additional linguistic features in an external knowledge base.

3.4 Feature Extraction

To extract features from a tweet, we substitute the term with its corresponding word embedding using different embeddings. In the ELMo embedding, the vector representing a token or word acts as a function of the whole sentence containing that token/word. As a result, different word vectors for the same word may exist in different contexts. Unlike the Glove, Word2Vec, and fastText embedding techniques, which ignore word order in the training phase and look up words and vectors in a dictionary, ELMo generates on-the-fly vectors through a deep learning model. The effectiveness of the different embeddings are shown in the result section below.

On the other hand, we reasoned that utilizing context to establish connections to identify common ground between word meanings sounds a lot like spanning nodes in graph-searching techniques, so we chose to represent the disambiguation process as several trees (one for each conceivable meaning), attempting to connect one to another. To make the model function, we needed contextual information about alternative interpretations arranged in a graph-like manner. We start with looking for terms in WordNet and ConceptNet, then we disambiguate the terms using Wikipedia. Figure 4 gives an example of term disambiguation using Wikipedia. Figure 5 gives an example of keyword relations with other terms.



Room service (disambiguation)

From Wikipedia, the free encyclopedia

Room service is a hotel service.

Room Service may refer to:

Film [edit]

- *Room Service* (1938 film), a Marx Bros. comedy directed by William A. Seiter
- *Room Service* (1982 film), a short film directed by Boris Bergman
- *Room Service* (1992 film), a comedy directed by Georges Lautner
- *Room Service* (2007 film), a romantic comedy directed by Kevin Castro

Music [edit]

- *Room Service* (The Oak Ridge Boys album), 1978
- *Room Service* (Shaun Cassidy album), 1979
- "Room Service," a 1975 song by Kiss (band) from the album "Dressed to Kill (album)"
- "Room Service", a 1980 song by Fischer-Z from album *Going Deaf for a Living*
- *Room Service* (Roxette album), 2001
- *Room Service*, a 2003 album by the Danish death-metal group Panzerchrist
- *Room Service* (Bryan Adams album), 2004
  - "Room Service" (song), a 2005 single by Bryan Adams, from his studio album *Room Service*

Figure 4 Example of Wikipedia disambiguation.

Figure 5 Keyword relatedness from ConceptNet.

3.5 Word/Sentence Embedding

We initialized word embedding using different pre-trained embeddings, i.e., we tested different word/sentence embeddings on the baseline data set (raw clean tweets). In another experiment, we generated an expanded tweet data set, which was composed of semantically related sentences derived from the baseline data set, and then applied ELMo embedding to it. On the same expanded tweets, we tested Word2Vec and Sent2Vec-based embedding. We used a Wiki2Vec model [8] to learn embedding; we created an entity-annotated corpus from Wikipedia by interpreting entity links in Wikipedia articles as entity annotations, and then used the produced corpus to train 300-dimensional skip-gram embedding using negative sampling, as proposed in [8]. By the learnt embedding, we can guarantee that similar words will be clustered together in a single vector space.

3.6 The Hybrid Model

We used the embedding technique with LSTM and CNN models because "representing words and sentences into vectors provides better performance for NLP problems" [16]. The idea behind embedding assumes that similar words and their contexts have similar vector representations, as in the sentences "Best Italian restaurant in Kuala Lumpur" and "Top Italian food in Kuala Lumpur". Most likely both sentences will be used in a similar context and also with similar words that have similar vector representations, such as 'pasta', 'favorite', or 'cousin'. These vectors were used as the input for the proposed CNN/LSTM model.

We tested our hybrid model with the embedding of the enriched tweets in both word and sentence vector representations. After tuning our hybrid CNN/LSTM model, we found the best hyperparameters, i.e., the ones that achieved the highest F1-score. Soft voting was utilized to average the projected probability of class membership among the selected models, and the hybrid class prediction was picked from the class with the highest average.

4 Experiments and Results

4.1 Evaluation Metrics

To validate our results on the training data, we computed the accuracy by finding the number of correct predictions out of the entire predictions. We also used F-measure [22] to calculate the harmonic mean of precision and recall. Precision encodes the ratio of all positive examples that were correctly classified to positive samples. Recall encodes the ratio of positive samples that were correctly classified to the total number of positive samples.

4.2 Results

The aim of our experiments was to explore the effectiveness of the proposed knowledge-based tweet enrichment technique by comparing it with the baseline tweet data set using the CNN/LSTM model. We propose a hybrid model of two standard deep learning models (CNN and LSTM) using Sentiment140 [6] as the baseline data set. We conducted several experiments on the baseline data set as well as on the enriched data set in order to make a fair comparison. As shown in Table 2, the obtained results showed that classification of consumer sentiment achieved a better F-score when using the expanded tweets over the raw baseline tweets. The last column describes the accuracy differences that were gained, noted as $\text{Gain}_{\text{\_percentage}} = \text{Fscore}_{\_enriched} - \text{Fscore}_{\_raw}$.

It is very interesting that almost all word-level embeddings performed better than sentence-level embeddings on the given data set. However, regardless which embedding technique was used, the accuracy of the proposed enrichment data representations outperformed the raw baseline data set. Based on Table 2, BERT, Wiki2Vec and ELMo were the best performing embedding techniques, with an accuracy percentage close to 90%.

Figure 6 shows a comparative analysis of the accuracy/epoch curves for the tweet enrichment method with various embedding representations. From Figure 7, we may infer that the tweet enrichment procedure was critical in gaining access to factors that could not be determined straight from the raw tweet data. In Figure 7, we present ELMo embedding and compared the results of using baseline vs enriched data representations. Clearly, we can conclude that the enrichment model may be used to improve sentiment analysis on a larger scale.

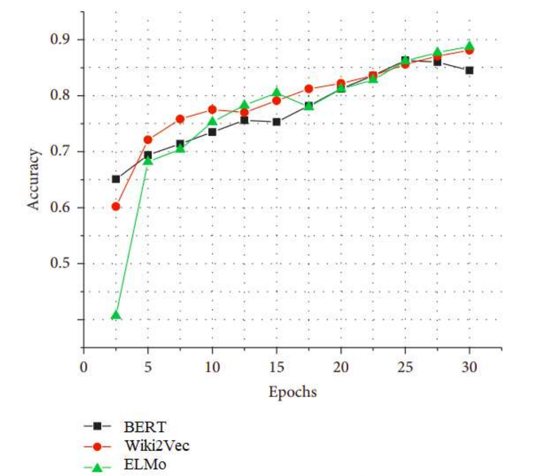


Figure 6 Accuracy of BERT, Wiki2Vec and ELMo embedding methods as the number of epochs increases (enriched tweets).

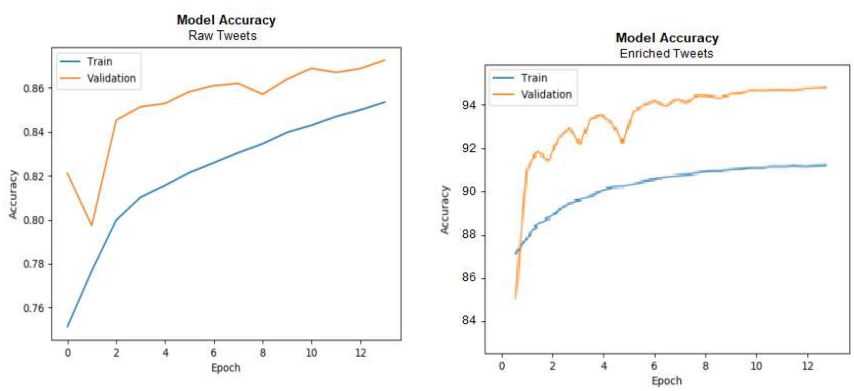


Figure 7 Accuracy of the hybrid model using the ELMo embedding method.

Finally, we compared the performance of the different learning models as shown in Figure 8. Figure 8 shows the performance of the hybrid model outperforming traditional machine learning techniques in terms of accuracy when using the enriched data set. After evaluating the performance of the proposed tweet enrichment method on several popular evaluation metrics, it can be concluded that the proposed method performs well with all embedding techniques. Therefore, the proposed enrichment method is effective in improving sentiment analysis results. We may also conclude that the hybrid CNN/LSTM model improves the accuracy of sentiment analysis compared with the traditional learning models.

Table 2 Precision, recall and F-score comparison of raw and enriched tweets.

Model	Raw vs Enriched Tweets						Gain percentage
	Precision		Recall		F-Score		
Word2Vec	0.47	0.53	0.45	0.66	0.46	0.59	13%
Sentence2Vec	0.69	0.75	0.70	0.82	0.69	0.78	9%
Glove	0.71	0.80	0.76	0.82	0.73	0.81	8%
FastText	0.63	0.73	0.71	0.87	0.67	0.80	13%
BERT	0.74	0.84	0.81	0.96	0.76	0.89	13%
ELMo	0.74	0.87	0.79	0.93	0.76	0.90	14%
Wiki2Vec	0.71	0.86	0.8	0.92	0.75	0.89	14%

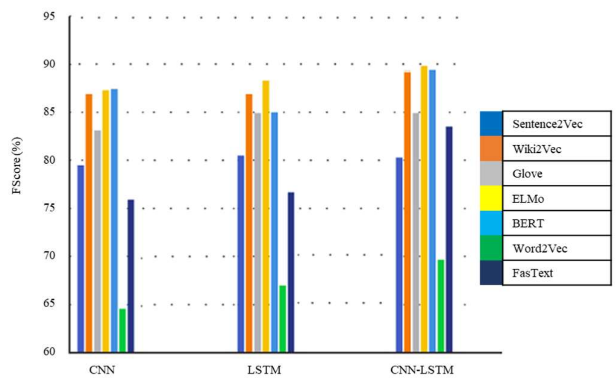


Figure 8 Comparison of the F-score results of the CNN-LSTM hybrid model, CNN and LSTM with respect to different embedding techniques.

5 Conclusion

The proposed method focuses on the task of tweet enrichment for sentiment analysis. We found that regardless of the used embedding, tweet enrichment can overcome the out-of-vocabulary problem (words that are not in the training set). We utilized a hybrid CNN/LSTM classifier for sentiment classification, but other hybrid models may perform better to solve this problem and thus we will investigate other hybrid models in a future work. Furthermore, we only looked at English tweets from the Twitter platform. However, our method does not incorporate customer sentiments in other languages.

We believe it will be a very interesting research direction to do additional research that includes tweets offered in other languages or a combination of languages. Finally, because certain aspects may have an impact on the sentiment score of a given tweet, ABSA will be taken into account in an upcoming research. We also plan to investigate other knowledge bases and test further embedding techniques as well.

Acknowledgment

This work was supported by Universiti Tunku Abdul Rahman, under the UTAR Research Fund (IPSR/RMC/UTARRF/2020-C1/R01).

References

[1] Dijck, J.V., *Tracing Twitter: The Rise of a Microblogging Platform*, International Journal of Media & Cultural Politics, **7**(3), pp. 333-348, 2011.

[2] Buccoliero, L., Bellio, E., Crestini, G. & Arkoudas, A., *Twitter and Politics: Evidence from the US Presidential Elections 2016*, Journal of Marketing Communications, **26**(1), pp. 88-114, 2020.

[3] Priem, J. & Costello, K. L., *How and Why Scholars Cite on Twitter,* Proceedings of the American Society for Information Science and Technology, **47**(1), pp. 1-4, 2010.

[4] Setiawan, E.B., Widyantoro, D. H. & Surendro, K., *Feature Expansion Using Word Embedding for Tweet Topic Classification,* in 10th International Conference on Telecommunication Systems Services and Applications (TSSA) , IEEE, pp. 1-5, 2016.

[5] Kacmajor, M. & Kelleher, J.D., *Capturing and Measuring Thematic Relatedness,* Language Resources and Evaluation, **54**(3), pp. 645-682, 2020.

[6] Go, A., Bhayani, R. & Huang, L., *Twitter Sentiment Classification Using Distant Supervision*, CS224N project report, Stanford, **1**(12), p.2009..

[7] Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T., *Enriching Word Vectors with Subword Information,* Transactions of the Association for Computational Linguistics, **5,** pp. 135-146, 2017.

[8] Mikolov, T., Chen, K., Corrado, G. & Dean, J., *Efficient Estimation of Word Representations in Vector Space,* arXiv preprint arXiv:1301.3781, 2013.

[9] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. & Zettlemoyer, L., *Deep Contextualized Word Representations,* arXiv preprint arXiv:1802.05365, 2018.

[10] Devlin, J., Chang, M. W., Lee, K. & Toutanova, K., *Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding,* arXiv preprint arXiv:1810.04805, 2018.
[11] Shoukry, A. & Rafea, A., *A hybrid approach for sentiment classification of Egyptian Dialect Tweets*, First International Conference on Arabic Computational Linguistics (ACLing), IEEE. pp. 78-85, 2015.
[12] Zhou, C., Sun, C., Liu, Z. & Lau, F., *A C-LSTM Neural Network for Text Classification,* arXiv preprint arXiv:1511.08630, 2015.
[13] Sosa, P. M., *Twitter Sentiment Analysis Using Combined LSTM-CNN Models,* Eprint Arxiv, pp. 1-9, 2017.
[14] Cho, M., Ha, J., Park, C., & Park, S., *Combinatorial feature Embedding based on CNN and LSTM for Biomedical Named Entity Recognition,* Journal of Biomedical Informatics, **103**, 103381, 2018.
[15] Wang, X., Liu, Y., Sun, C., Wang, B. & Wang, X., *Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory*, Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1343-1353, 2015.
[16] Guggilla, C., Miller, T. & Gurevych, I., *CNN- and LSTM-based claim classification in Online User Comments,* Proceedings of the International Conference on Computational Linguistics (COLING), pp. 2740-2751, 2016.
[17] Huang, M., Qian, Q. & Zhu, X., *Encoding Syntactic Knowledge in Neural Networks for Sentiment Classification*, ACM Transactions on Information Systems, **35**(3), pp. 1-27, 2017.
[18] Yu, J. & Jiang, J., *Learning Sentence Embeddings with Auxiliary Tasks for Cross-Domain Sentiment Classification*, Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 236-246, 2016.
[19] Selvaretnam, B. & Belkhatir, M., *Natural Language Technology and Query Expansion: Issues, State-of-the-Art And Perspectives,* Journal of Intelligent Information Systems , **38**(3), pp. 709-740, 2012.
[20] Agarwal, B., Mittal, N., Bansal, P. & Garg, S., *Sentiment Analysis Using Common-Sense and Context Information,* Computational Intelligence and Neuroscience, 2015.
[21] Leacock, C. & Chodorow, M., *Combining Local Context and Wordnet Similarity for Word Sense Identification*, WordNet: An Electronic Lexical Database, **49**(2), pp. 265-283, 1998.
[22] Rieh, S.Y., *Analysis of Multiple Query Reformulations on the Web: the Interactive Information Retrieval Context,* Information Processing & Management, **42**(3), pp. 751-768, 2006.