Cognitive Computation manuscript No.
COGN-D-20-00042R3

Hybrid Deep Learning Models for Thai Sentiment Analysis

Kitsuchart Pasupa · Thititorn Seneewong Na Ayutthaya

Received: 28 February 2020 / Accepted: 22 August 2020

Abstract Background: Many people use social media in their daily life for entertainment, business, personal communication and catching up with friends. In social media marketing, sentiment analysis is one of the most popular research topics because it can be employed to perform brand or market research monitoring and to keep an eye on the competitors. Machine learning algorithms have been utilised to carry out the task. In addition, sentiment analysis is essential in cognitive computing. Currently, there are still a limited number of Thai sentiment analysis research. **Methods:** This paper proposes a framework for sentiment analysis in Thai along with Thai-SenticNet5 corpus. The framework employs different types of features, namely, word embedding, part-of-speech, sentic features, and all combinations of these features. Furthermore, we fused deep learning algorithms–Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BLSTM)–in different ways and compare it to several other fused combinations. Three datasets in Thai were used in this work: ThaiTales, ThaiEconTwitter and Wisesight datasets. **Results:** The experimental results show that combining all three features and fusing deep learning algorithms were able to improve overall performance. The best hybrid deep learning was BLSTM-CNN that achieved F_1-scores of 0.7436, 0.7707, 0.5521, on ThaiTales, ThaiEconTwitter, and Wisesight datasets, respectively. **Conclusion:** According to the experimental results, we conclude that feature combination and hybrid deep learning algorithms can improve the overall performances.

Keywords Sentiment Analysis · Thai Sentiment · SenticNet5 · Deep Learning · Hybrid Model

1 Introduction

Social media connect us with other people, sharing aspects of our life that we value. Unarguably, this means of communication has become a part of one's life in this digital age. In the past decade, the number of social media users has increased tremendously, so has the number of platforms that have been designed and developed to better suit users' needs. These include social network, microblog, web forum, photo/video sharing site, etc. The blooming of social media brings an overwhelming amount of data to the network, combinations of text, image, video, etc. People share their experiences and opinions through

K. Pasupa (Corresponding Author)
Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang,
Bangkok 10520, Thailand
E-mail: kitsuchart@it.kmitl.ac.th

T. Seneewong Na Ayutthaya
Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang,
Bangkok 10520, Thailand
E-mail: thititorn.tao@gmail.com

these means. Therefore, such raw data can provide useful information when extracted. Because of these reasons, social media has become the subject of interest of many researchers [1].

At present, a large amount of raw data in social media appears in the form of text. It is impossible for a human to analyse all of those text messages and manually extract useful information from them. In order to achieve this goal, computational analysis is required to manage the massive data. A specific technique known as Natural Language Processing (NLP) has long been employed to enable computer to understand human languages.

Many companies have invested buckets of their money in digital media marketing, hoping to increase their sales volume [2]. To ensure the worthiness of the budget spent, they also use social monitoring as a tool to trace customers' satisfaction and other feedbacks as well as keep track of customers' desire of new services and/or products. To process social monitoring, NLP technique has been applied to sentiment analysing the text messages that customers have provided. The findings from the analysis allow suppliers to gain better insights into customers' attitudes, either positive, negative or neutral, towards their products. Particularly in the case of a negative comment, quick response and prompt action from suppliers can demonstrate their trustworthiness. Thus, analysis of customers' sentiment is necessary not only for improving the products/services but also the credibility and good image of the companies. Many commercial social media marketing tools have been developed, such as ZocialEye[1], Evolve24[2], and many more, to assist the companies to keep track of their customers [3]. More details on sentiment analysis can be found in [4].

At the present, deep learning model is the predominant technique for constructing prediction models in various fields of study including NLP. Various deep learning algorithms have been adopted to deal with different types of data. For example, researchers use Convolutional Neural Network (CNN), the prevailing algorithm for computer vision and text analysis, to extract local structure while resort to Long Short Term Memory (LSTM) and Bi-directional LSTM (BLSTM) to manage sequential data and various linguistic patterns, respectively [5,6,7]. Furthermore, previous works have suggested that the quality of a prediction model can be improved when multiple features are integrated into an analysis [8,9,10]. This is because each feature can be complimentary to each other. Because of this reason, Pasupa and Seneewong Na Ayutthaya (2019) incorporated word embedding, part-of-speech (POS) and sentic features into various deep learning models that represent words as vectors, identify POS, and associate words with feeling, respectively [11]. This study's findings reaffirm the aforementioned claim–integrating multiple features can enhance performances–by performing sentiment analysis on Thai children tales. Apart from combining features, this study also compared the efficiency of various deep learning models. The comparison showed that CNN is the most efficient model for sentiment analysis. In addition, Seneewong Na Ayutthaya and Pasupa (2018) also attempted to fuse deep learning models, BLSTM and CNN, in order to, firstly, examine sequences of words, and secondly, explore local features of the text [12]. It was found that the fusion of deep learning models resulted in a higher accuracy of the sentiment analysis. However, these two works were only on a single dataset of 40 Thai Children Tales. It should be noted that most Thai sentiment analysis research studies mentioned above and in Section 2 were conducted on only one dataset and undertook different pre-processing steps. Consequently, it is hard to compare all of them together and reveal the most efficient way to construct models appropriate for sentiment analysis research because of those different experimental frameworks. Additionally, deep learning models have already been fused in various ways, e.g. [13,14,15]. All of those works did not compare their combined networks against all of the others but only against individual ones. Thus, models hybridised in different manners should be compared against each other.

[1] https://wisesight.com/zocialeye

[2] https://evolve24.com/

In this paper, we aimed to construct a Thai sentiment analysis framework that fuses CNN and BLSTM in different ways on three datasets. According to our literature review, there are a number of research gaps that have been addressed in this work. Our contributions are as follows:

1. To the best of our knowledge, this is the first Thai sentiment analysis study which drew its findings from more than one set of data. Precisely, we performed sentiment analysis on three different datasets that had been collected from different sources: (i) Wisesight dataset collected from various social media platforms such as Facebook, Twitter, web forum; (ii) Thai Economy Twitter dataset collected solely from Twitter; and (iii) 40 Thai Children Tales. The first two datasets were from social media while the latter was from the literature. Hence, the writing styles were different.
2. To effectively analyse human sentiment, we propose that instead of relying on only one feature, a combination of features–word embedding, POS, and sentic features–should be incorporated into the analysis to increase the accuracy of human sentiment predictions. This is to strengthen the statement of our previous work [11].
3. According to the literature, there have been different approaches to fusing models, but the hybrid models were only compared against other individual models running on different datasets without comparing them to each other. Therefore, we compared different combinations of deep learning techniques, for example, CNN-BLSTM, BLSTM-CNN, BLSTM+CNN, and BLSTM×CNN on the same framework and with several different datasets. We demonstrated that among different combinations of deep learning techniques, BLSTM-CNN generated the most reliable results and is thus the best method for doing Thai sentiment analysis.
4. The current Thai-SenticNet used in this study is the latest version that we have updated from the previous Thai-SenticNet2, proposed in [16]. Unlike SenticNet2, this corpus drew its information from SenticNet5 and includes more words. To achieve this goal, it incorporated LEXiTRON [16], Volubilis [17], and Thai-Wordnet [18] to render the translation of texts between Thai and English languages. As a result, we successfully constructed the Thai-SenticNet5.

This paper is arranged as follows: Section 2 describes the papers related to our work in the literature. The proposed framework is explained in Section 3 that shows the data preprocessing step, feature extraction process, and hybrid deep learning models. Section 4 describes the experimental setup and datasets, followed by the results and discussion Section 5. Finally, we conclude our work in Section 6.

2 Related Works

Many researchers have investigated sentiment analysis in various types of learning problem, such as supervised learning [11,12,19], unsupervised learning [20], semi-supervised learning [21], and reinforcement learning [22]. Regarding sentiment analysis, most studies following this line of research have been conducted with texts in English [23,24]. Few researches were on other languages, e.g. German, French, Japanese and Chinese [25,26,27,28]. Recently, a framework for multilingual sentiment analysis called BaBelSenticNet was proposed [29]. It translates SenticNet corpus via statistical machine translation tool into 40 languages based on WordNet and its multilingual version. As for Thai, even though Sentiment Analysis was introduced to examine Thai texts a decade ago [30], the number of Thai sentiment analysis researches is still limited because to analyse Thai text effectively, a limiting factor is that multiple preprocessing steps are required. Hence, it is a challenge for researchers to create Thai NLP to deal with the lack of word delimiter and sentence boundary marker, nostalgic Thai slangs, etc. To do so, a fine-grained corpus of Thai language that enables researchers to identify word segmentation, tag part-of-speech, manage named identity recognition and analyse syntactic parsing, among others, is essential. Unfortunately, up till now, we still have inadequate tools and limited resources supporting Thai sentiment analysis [31]. In 2010, Thai text sentiment analysis was first conducted using Term of Frequency as an input feature by [32,33]. Since then, Thai sentiment analysis research continued [34,35,36,37,38]. Most of the studies rely on corpora that applied Dictionary-based technique to create feature extraction. In those corpora, words are categorised into three groups, positive, neutral and negative, and are tagged with either –1, 0, or +1 label. However, the 3 labels are, in fact, insufficient to describe human sentiment. Therefore, Lertsuksakda *et al.* (2014) proposed using a corpus that incorporates a better defined weight, ranging from –1 to +1, in Thai sentiment analysis [16]. The corpus was constructed based on SenticNet2 proposed by Cambria and his colleagues [39]. To translate English terms into Thai and verify the Thai meanings gained from the translation process, this study adopted a bi-directional translation technique. Then, Sentic features were extracted from sentences and used to analyse sentiment in Thai children tales [19, 40].

Deep learning has played a major role in sentiment analysis tasks. An important feature used with deep learning is word embedding which is a feature that transforms words into vectors. Each dimension of this kind of vectors represents a meaning or a context of the word. Word embedding can be done by a Word2Vec model [41]. Besides word embedding feature, there have been several more features used in sentiment analysis tasks, features such as term frequency, POS , and sentic. One of our previous works used combinations of word embedding with other features such as POS-tag feature that identifies the grammatical type of a word in a sentence and sentic feature that represents the emotion of a word in vector form [11]. Those combinations truly improved the performance of our model for sentiment classification in Thai Tales. Also, consolidating sentiment information into text embedding process can obtain better representations for sentiment analysis [42].

Conventional deep learning models are such as CNN and LSTM. A CNN model is a feed-forward neural network, a long-time favourite for computer vision task. One of the processes in a CNN model is computation of groups of pixels that are components of image data, but for NLP tasks, it processes groups of words instead [43]. An LSTM model is one of Recurrent Neural Network (RNN) models that can learn sequential data, sequences of words in an NLP task. Normally an LSTM model learns sequential data in forward direction, but in some cases, a pattern needs to be learned in backward direction, so BLSTM was developed to handle it; BLSTM learns sequential data in both forward and backward directions. Many research studies have shown that BLSTM performed better than LSTM [44,45,46]. CNN, LSTM, and BLSTM models have been used in sentiment analysis tasks. Ouyang *et al.* (2015) used a CNN model to perform sentiment classification on a movie review dataset [47] and showed that the CNN model was more accurate than shallow classification algorithms such as Naïve Bayes or Support Vector Machine [48]. Nowak *et al.* (2017) compared LSTM against BLSTM models in sentiment classification of an Amazon book review dataset and found that the BLSTM model was more accurate than the LSTM model in this task [49].

Besides research studies using individual models, there have been studies that used combinations of models to improve performance. Wang *et al.* (2016) [13] shows that a combination of a CNN model with an LSTM model yielded a lower error measure than individual CNN and LSTM models alone did in predicting the valence-arousal value (dimensional sentiment in numerical form) of Stanford Sentiment Treebank (English language) [50] and Chinese Valence-Arousal Text (Chinese language) [51]. Lin *et al.* (2017) shows that a combination of BLSTM with CNN provided the best performance among BLSTM, CNN, CNN-LSTM, LSTM-CNN, and LSTM in predicting the type of customer feedback in an IJCNLP 2017 Shared Task on Customer Feedback Analysis dataset [14]. Minaee and his colleague [15] show that an ensemble between CNN and LSTM model provided a higher accuracy than individual CNN and LSTM models alone in sentiment classification of an IMDB review dataset [52] and a Standford Sentiment Treebank dataset.

Sentiment analysis is normally performed at coarse level, i.e., document or sentence level. Recently, aspect-level sentiment analysis has been proposed [6]. There might be multiple feelings in a sentence, for example, “Bad service but really good food”. In this case, there are two aspects which are “service” and “food”. It is clearly seen that this customer has a positive sentiment toward food but a negative sentiment towards the service of this restaurant. Therefore, aspect-level sentiment analysis aims to understand the sentiment in a certain aspect term. This can be achieved by integrating attention mechanism in learning models [53]. The mechanism imitates human's attention behaviour in reading and focusing on a context word that draws their attention. Wang *et al.* (2016) employed the attention mechanism on LSTM and proposed a model called "Attention-based LSTM with Aspect Embedding" [6]. The aspect (target word) embedding is concatenated with word embedding vector and fed into the LSTM layer while it is concatenated with hidden state vector and fed into the attention layer. Ma *et al.* (2017) proposed an Interactive Attention Network that utilises two LSTM models to separately learn both context and target words [54]. Then, hidden states of both models are interactively learned through the attention mechanism and combined together. Both researches show that employing attention mechanism in LSTM can clearly improve the overall performance of aspect-level sentiment analysis on SemEval 2014 Task 4 [55].

3 Proposed Framework

The framework for Thai sentiment analysis in this experiment consisted of 3 main parts: (i) Data preprocessing, (ii) Feature extraction and (iii) Learning model, as shown in Figure 1.

3.1 Data preprocessing

3.1.1 Data Cleansing

To boost the performance of sentiment classification, a text must be passed through a cleansing process to get rid of noise that can affect the performance of other downline processes, especially before the text is processed by a tokenisation process. The text input in our work had to be processed in the following ways: (a) changing any English words from upper case to lower case and (b) since the text data used in this experiment, such as Thai Economy Twitter and Wisesight datasets, were collected from social media, there occurred some Uniform Resource Locators (URLs) in the text; many URLs were written with characters and numbers that were long and did not have any useful meanings, i.e., they are quite noisy, so we changed every URL to "xxurl".

3.1.2 Word Tokenisation

Each word in a sentence must be split before it is fed into the model as input. The process that splits each word in a sentence is called word tokenisation. Thai language is not like English language in the sense that two adjacent English words are separated by a space. Therefore, to split Thai words in a sentence needs a special technique. In this experiment, we split words in a sentence by using a technique based on a Maximum Matching algorithm from PyThaiNLP Library [56]. Maximum Matching algorithm implemented in PyThaiNLP is a dictionary-based approach. The algorithm scans series of input characters and match them with words in a dictionary [57,58,59]. Then, it employs breadth-first search to select segmented series that contain a minimum number of word tokens. Word tokens that are not in the dictionary will be segmented into Thai character clusters. A Thai character cluster is an unambiguous unit that is smaller than a word. It is an indivisible unit. This process is performed by the character clustering algorithm proposed by Theeramunkong and his colleagues [60]. The algorithm utilises a set of simple rules based on types of Thai characters. After a word tokenisation process, all tokens (except emoticons) are fed into a spell check algorithm proposed by Peter Norvig [61]. The algorithm will find possible permutations of the original words within a two edit-distance (i.e, inserts, replaces, transposes, deletes). Words that have the highest frequency in a list are selected. It is noted that the list contains words which are edited and matched with words in the dictionary.

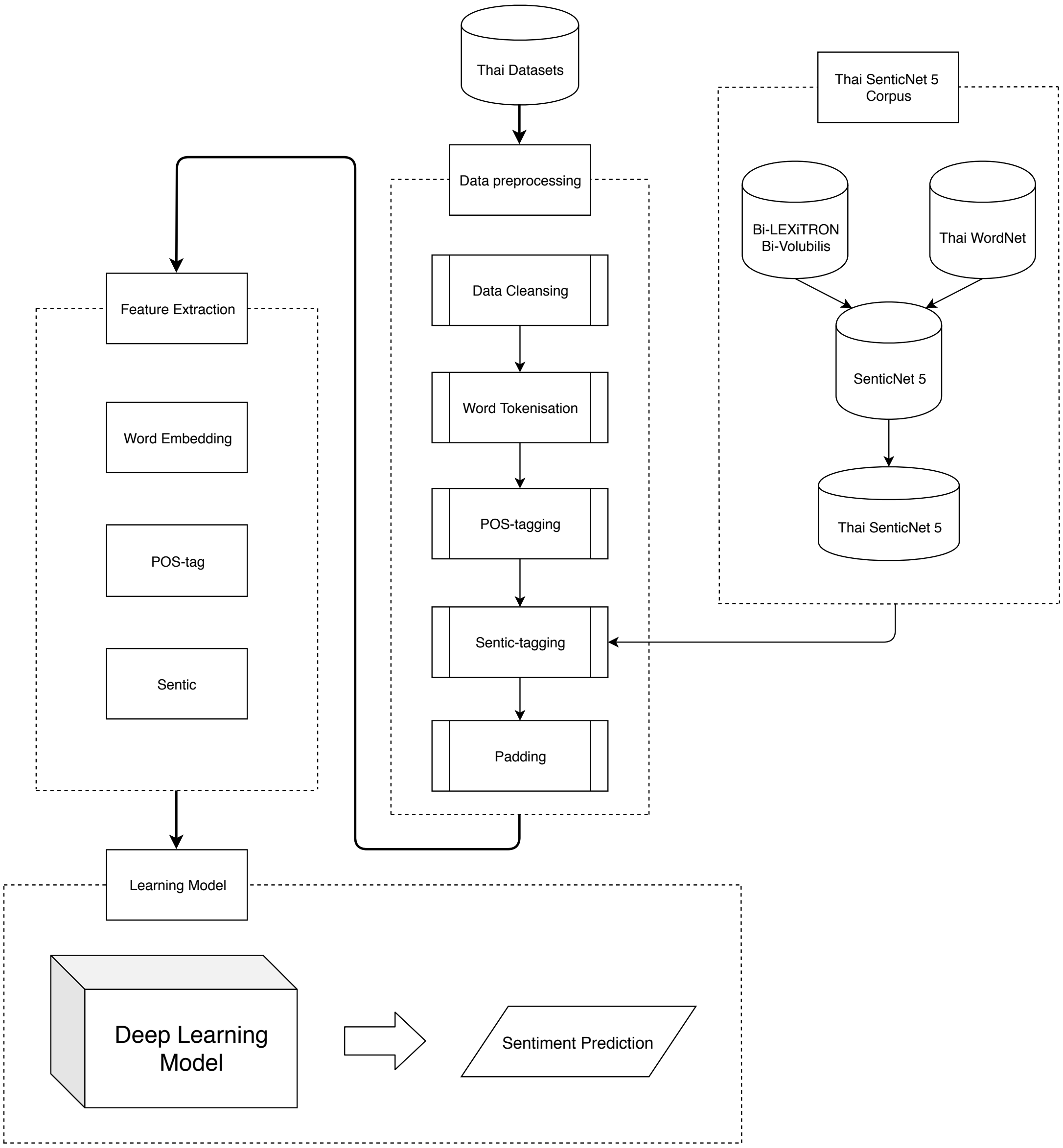


Fig. 1 Thai Sentiment Analysis Framework.

3.1.3 POS-tagging

In this experiment, we used a POS-tagging process to identify the types of words in a sentence needed for construction of POS-tag features. POS-tagging used a model based on Perceptron Tagger from PyThaiNLP library to do the tagging. This model categorises types of words into 47 types based on ORCHID corpus which is a Thai POS Tagged corpus [62]. However, 47 types of words seem too complex and can be difficult for a model to learn. Therefore, the 47 types of words from ORCHID categorisation were mapped to 17 types of words based on Universal POS tags known as Universal Dependencies (UD) [63]. Those 17 types of words were simple, easily comprehensible and preferred in many languages. However, in the mapping process, only 15 types of word were mapped because no type of words based on ORCHID could be mapped into two types of UD: ‘symbol’ (SYM) and ‘other’ (X) as shown in Table 1. Please note that the mapping function was a part of PyThaiNLP library.

Since social media data often express sentiment or emotion of the users with emoticons, we added an “EMOJI” type as one of POS-tag types to identify emoticons in sentences. Emoticon is directly associated with emotion and so beneficial to sentiment analysis [64]. All the emoticons were mapped to their name in English via emoji library [65]. In addition, we used a padding process, explained in Section 3.1.5, and the words padded by this process would be another “PAD” type. Therefore, we categorised POS-tag types into 19 types that were not exactly the same as the original 17 UD types.

Regarding the emoticons, their sentic vector was computed according to its name mapped by emoji library. In the case that an emoticon's name was longer than one word, that emoticon sentic vector would be represented by an average across all sentic vectors of all words.

3.1.4 Sentic-tagging

A feature that represents the emotion of a word is called a sentic feature. The sentic value of a word was encoded in a 5-dimensional vector that consisted of four affective dimensions and a polarity as presented by [66]. The emotion represented by each dimension is explained in detail in Section 3.2.3. Sentic vectors that represent English sentiment lexicon can be obtained from SenticNet, which was now in the 5.0 version. For our use in Thai language, we then needed to construct a new Thai-SenticNet. This Thai-SenticNet was constructed based on the two following concepts: bi-directional translation and Thai-Wordnet.

1. Thai words were aligned with English words and their alignment was verified with a bi-directional translation [67] technique based on Bi-LEXiTRON [16] and Bi-Volubilis [17] corpuses. Furthermore, several words were added from Thai-WordNet [68], Thai words that aligned with English words in Synset[3].
2. Constructing new words by deleting some stop words [69] from our corpus (details about the corpus are below) and adding them to the corpus to make it more comprehensive.

In this work, we used the following corpuses to map English sentic vectors to Thai sentic vectors: (i) Bi-directional LEXiTRON (Thai-English) [16], (2) Bi-directional Volubilis 11K (Thai-English) [17], (3) Volubilis-100K [17], (4) Thai-WordNet [18] and (5) SenticNet5 [70]. Our final corpus contained 23,093 Thai words that had been successfully verified and 15,247 sentic vectors. We called it a Thai-SenticNet5.

The purpose of Thai-SenticNet5 was to accurately map the sentic values of English words in SenticNet5 to the sentic values of corresponding Thai words in our constructed corpus. There were several Thai to English dictionaries or corpuses such as LEXiTRON, Volubilis, and Thai-WordNet. We wanted our constructed corpus to contain as many Thai words as possible, so we combined the three Thai corpuses mentioned above under the constraint that each translated word had to be successfully verified by the Bi-directional Translation Technique such that the meaning of every Thai word in the corpus would be the same when it is translated into English then translated back into Thai [16].

First, we considered Thai words in Volibilis-100K which was the biggest corpus in this experiment with 107,607 entries. Then, we put the same words that had more than one entry in the corpus, i.e., listed in different POS sections in this corpus, in one entry, reducing the number of entries or words in the corpus to 100,107 words. Words in this modified Volubilis-100K were not verified by the Bi-directional translation technique; therefore, we selected only the Thai words in this corpus and matched them with translated English words from the following corpuses that had already been verified by the bi-translation technique:

1. Thai WordNet corpus created by aligning Princeton WordNet's Synsets with Thai words by using a Bi-lingual dictionary;
2. LEXiTRON-Volubilis-Bi corpus created by merging LEXiTRON-Bi [16] that had 2,871 words with Volubilis-Bi [17] that had 11,065 words. The merged corpus was called LEXiTRON-Volubilis-Bi. Please note that the Thai words in Volubilis-Bi were a subset of the 11,820 entries in Volubilis-100K.

Afterwards, we merged only the Thai words from the modified Volubilis-100K that had 100,017 words with a list of Thai words from LEXiTRON-Bi (we did not merge the Thai words from Volubilis-Bi because

[3] A group of English words that have synonymous meaning with Thai words.

they were a subset of the words in Volubilis-100K) and got 100,118 Thai words. Furthermore, to get even more Thai words, we used a technique that delete stop words from each relevant entry in the list and added these entries with deleted stop words into the list and got 119,281 Thai words. Let us call this list ThaiWordList. After that, we started to create a verified dictionary with the two corpuses by aligning the Thai words in ThaiWordList with a set of English words in Thai-WordNet and LEXiTRON-Volubilis-Bi corpuses. We matched each Thai word in ThaiWordList with a set of English words in Thai-WordNet under the condition that the Thai word had a corresponding English word in synset in Thai-WordNet. If the Thai word did not have any corresponding English word in the synset in Thai WordNet, we matched it with a set of English words in LEXiTRON-Volubilis-Bi instead. Finally, we got 23,093 matches and started to create the Thai-SenticNet5 corpus by mapping a set of English words with the corresponding Thai word to SenticNet5 corpus to get a set of sentic values and an average sentic value for a Thai word. That was how we got the 15,247-word Thai-SenticNet5 corpus with a sentic vector for every Thai word in the corpus.

This construction of Thai-SenticNet5 corpus was shown as pseudocode in Algortihm 1. Table 2 shows the number of verified words and the number of words that had a sentic vector in each of the mentioned corpuses. It can be seen that the number of verified words that had an associated sentic vector increased after every step of the construction process. Thai-SenticNet5 is available for download at `https://github.com/dsmlr/ThaiSenticNet5`.

3.1.5 Padding

A set of words had to be transformed into vector data before they were fed into the model. The vector data were chunks of data (batches) and every sample vector in each batch had to be of the same size. Therefore, we needed to pad some words in the set with ⟨PAD⟩ in order that every sample vector would be of the same size.

3.2 Feature Extraction

3.2.1 Word Embedding

Deep Learning model is a subset of neural network models. They are mathematical models that cannot learn directly from raw text data. Indeed, it can only learn from vector data, therefore a word must be transformed into a vector first. This kind of transformation is called word embedding which can be done by models such as Word2Vec, GloVe, ULMFiT. Conventional Word2Vec models are such as the following two models: Continuous Bag-of-Words model that uses context words (words surrounding the target word) as input to predict the target word and Skip-Gram model that uses the target word to predict the context words [71]. An efficient Word2Vec model should be trained with a large corpus. GloVe model [72] learns word vectors by using information from word co-occurrence probabilities at the global level (whole dataset) and gives good results when it is trained on a large corpus. However, training a model on large corpus consumes a lot of time. Fortunately, there has been a research study that proposed a technique for fine-tuning a language model that can transfer knowledge gained from one task to any other tasks in NLP which means that the language model does not need to be trained from scratch. This technique is called Universal Language Model Fine-tuning (ULMFiT) [73].

The datasets used in this experiment were relatively small, so it was difficult to train and get an efficient model from scratch, hence we used a pre-trained language model to transform words into vectors. The pre-trained language model was from thai2fit library [74]. It was an ASGD Weight-Dropped LSTM model [75] trained by a ULMFit method on Thai Wikipedia dataset. The number of dimensions of the pre-trained embedded word vectors was 300.

Algorithm 1 Thai-SenticNet5 Corpus Construction

```
Require:
    LEXiTRON, Volubilis-Bi, Volubilis-100K, Thai-WordNet, SenticNet5,
Ensure:
    ThaiSenticNet5
LEXiTRON-Bi $\leftarrow$ BiDirectionalTranslation(LEXiTRON)    ▷ Perform Bi-directional Translation
LEXiTRON-Volubilis-Bi $\leftarrow$ LEXiTRON-Bi $\cup$ Volubilis-Bi    ▷ Merge two corpuses
ThaiWordList $\leftarrow$ Thai(Volubilis-100K) $\cup$ Thai(LEXiTRON-Bi)    ▷ Merge all Thai words into one list
ThaiWordListNoStopWord $\leftarrow$ ExtractWordByStopword(ThaiWordList)    ▷ Remove stop words
ThaiWordList $\leftarrow$ ThaiWordList $\cup$ ThaiWordListNoStopWord    ▷ Merge two lists
ThaiEngCorpus-Verify $\leftarrow \emptyset$    ▷ Initialise ThaiEngCorpus-Verify
for Each thaiword $i$ in ThaiWordList do
    WordSet $\leftarrow$ synset instances of ThaiWordList$_i$ in Thai-WordNet
        ▷ Get synset instances of $i$ from Thai-WordNet
    if Size of WordSet $> 0$ then
        ThaiEngCorpus-Verify $\leftarrow$ ThaiEngCorpus-Verify $\cup$ {ThaiWordList$_i$: WordSet}
            ▷ Add Thai word with its synsets
    else
        WordSet $\leftarrow$ words of ThaiWordList$_i$ in LEXiTRON-Volubilis-Bi
            ▷ Get words of $i$ from LEXiTRON-Volubilis-Bi
        if Size of WordSet $> 0$ then
            ThaiEngCorpus-Verify $\leftarrow$ ThaiEngCorpus-Verify $\cup$ {ThaiWordList$_i$: WordSet}
                ▷ Add Thai word with its English words
        end if
    end if
end for
ThaiSenticNet5 $\leftarrow \emptyset$    ▷ Initialise Thai-SenticNet5
for Each thaiword $i$ in ThaiEngCorpus-Verify do
    senticVectorsList $\leftarrow \emptyset$
    for Each engword $j$ in ThaiEngCorpus-Verify$_i$ do
        if found engword $j$ in SenticNet5 then    ▷ Align engword with SenticNet5
            senticVectorsList $\leftarrow$ senticVectorsList $\cup$ SenticNet5(engword $j$)
                ▷ Add sentic vector of engword $j$ into a list
        end if
    end for
    senticVector = Average(senticVectorsList)
    ThaiSenticNet5 $\leftarrow$ ThaiSenticNet5 $\cup$ {thaiword $i$: senticVector}    ▷ Add word with its sentic vector
end for
```

3.2.2 POS-tagging

In this experiment, we used a POS-tag feature that represented a type of word in a sentence in one-hot vector form. The number of dimensions of our one-hot vector was related to the number of types of POS. For each POS-tag type, the corresponding one-hot vector would have a value of 1 in a dimension while the other dimensions would have a value of zero. We categorised POS-tag into 17 UD POS-tag types as shown in Table 1 and two additional types–EMOJI and PAD.

3.2.3 Sentic

Sentic feature is a five-dimensional vector composed of four affective dimensions based on Hourglass of Emotion theory. The theory follows psychological principles that rely on activities of the brain while the condition of the mood changes [76,77]. The four dimensions are sensitivity (*Snst*), aptitude (*Aptit*), attention (*Attnt*), and pleasantness (*Plsnt*). We were able to calculate the polarity of a word by,

$$p = \sum_{i=1}^{N} \frac{Plsnt(\kappa_i) + |Attnt(\kappa_i)| - |Snst(\kappa_i)| + Aptit(\kappa_i)}{3N}, \tag{1}$$

where N is the total number of word concepts (concepts that describe objects or actions perceived) and κ_i is the i-th input concept. Here, p was in the range $[-1, 1]$ implying extremely negative to extremely positive emotion.

This research obtained the sentic values from SenticNet5 corpus. The corpus is a sentiment lexicon at concept-level. It employs BLSTM to infer primitives by lexical substitution. Feature vectors were extracted using Thai-SenticNet5 Corpus as explained in Section 3.1.4. For any Thai words that do not exist in the corpus, the feature vectors would be represented by 5-D zero vectors.

3.3 Learning Model

3.3.1 Bi-directional Long Short-term Memory

One of powerful algorithms in the RNN family is BLSTM which is able to learn sequential data both in forward and backward directions. It is an extension of LSTM. It has been known that RNN has a gradient vanishing problem for long data sequences [78]. Therefore, LSTM has been introduced to deal with this problem.

LSTM processes data in a forward direction with an ability to remember and forget the information. LSTM model consists of the following components: forget gate (f_t), input gate (i_t), input modulation gate ($\tilde{c_t}$), cell state (c_t), output gate (o_t) and hidden state (h_t). Forget gate (f_t) enables the model to be able to reset itself–to forget the old information at an appropriate time. When there is a new incoming sample (x_t), the sample will be considered together with the previous hidden state (h_{t-1}) whether how much information should be forgotten. Sigmoid function is employed for this task. Its value ranges from 0 to 1, corresponding to completely forget or remember the previous information, respectively. This can be explained by

$$f_t = sigmoid(W_f[h_{t-1}, x_t] + b_f). \tag{2}$$

Input gate (i_t) is a gate that decides which information will be updated, again considering together with the previous hidden state (h_{t-1}). The sigmoid function will decide how much new information should be updated based on values of 0 to 1,

$$i_t = sigmoid(W_i[h_{t-1}, x_t] + b_i). \tag{3}$$

Input modulation gate ($\tilde{c_t}$) is similar to candidate cell state that learns both new information (x_t) and the previous hidden state (h_{t-1}). It utilises the tanh activation function and create a vector of new candidate values,

$$\tilde{c_t} = tanh(W_c[h_{t-1}, x_t] + b_c). \tag{4}$$

Cell state (c_t) is a long-term memory cell that is a combination of old information (c_{t-1})–that is dropped by a forgot gate–and new information–that is a product of input gate i_t and input modulation gate ($\tilde{c_t}$),

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c_t}. \tag{5}$$

Output gate (o_t) has a role to decide what the next hidden state should be. It sends information to the hidden state (h_t) that is restricted to an interval $[0, 1]$ by a sigmoid function,

$$o_t = sigmoid(W_o[h_{t-1}, x_t] + b_o). \tag{6}$$

The last component is hidden state (h_t) referred to as an output of LSTM. It carries the information on what LSTM has seen,

$$h_t = o_t \cdot tanh(c_t). \tag{7}$$

On the other hand, BLSTM processes data in both forward and backward directions. The architecture of BLSTM is shown in Figure 2.

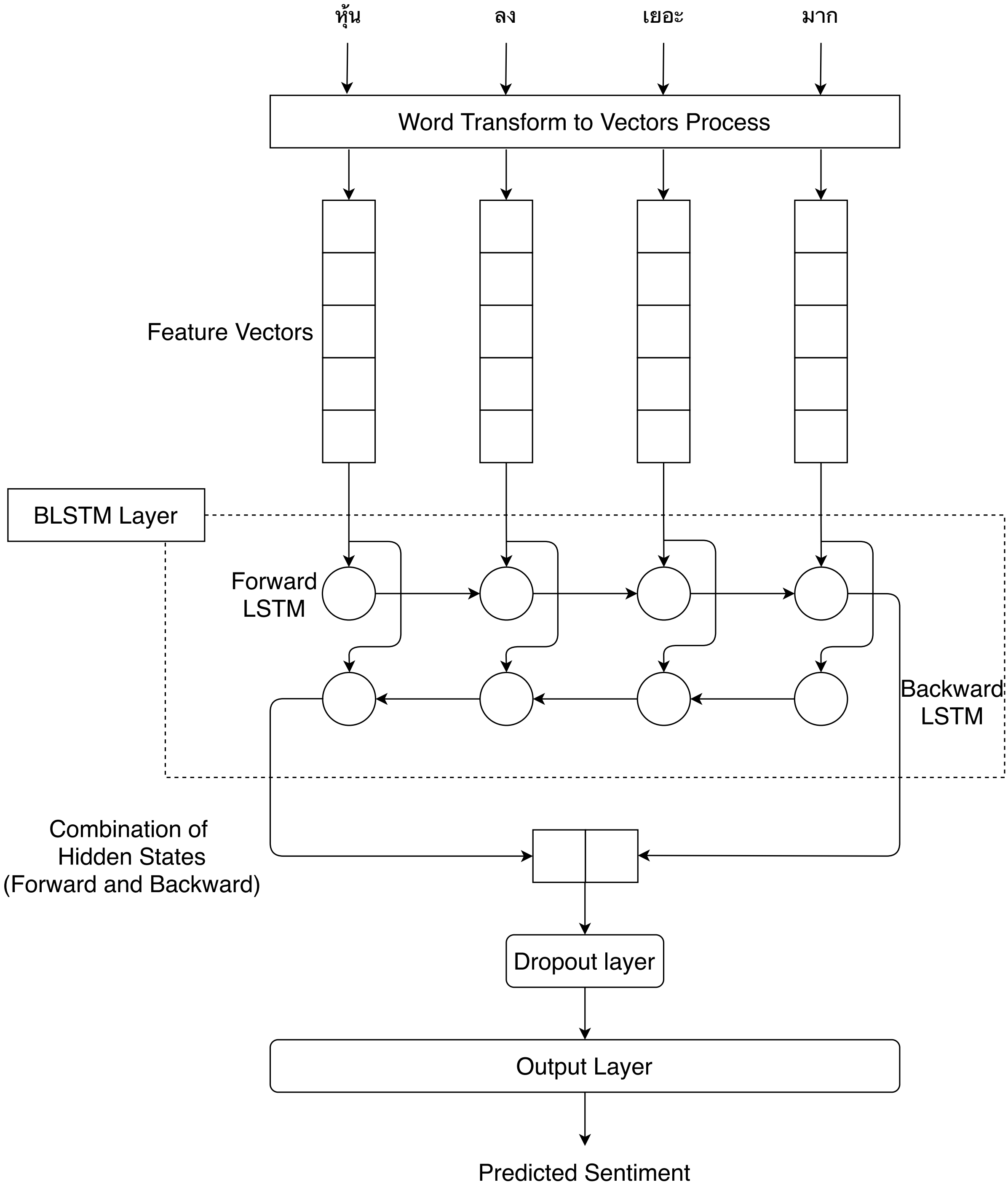


Fig. 2 Bi-directional Long Short-term Memory for Sentiment Analysis

In this study, once a sentence was fed into the model, it went through the embedding layer that converted the sentence into a word embedding feature that was further fed to a dropout layer. Then, it was combined with POS-tag and sentic vectors as shown in Figure 3. Then, we fed the combined features to a BLSTM layer. The hidden states of both forward and backward directions–the last output of BLSTM–would be concatenated and fed to the dropout layer to prevent over-fitting problem [79] before pushed on to the output layer. Softmax activation function was used to predict output classes as output probabilities range.

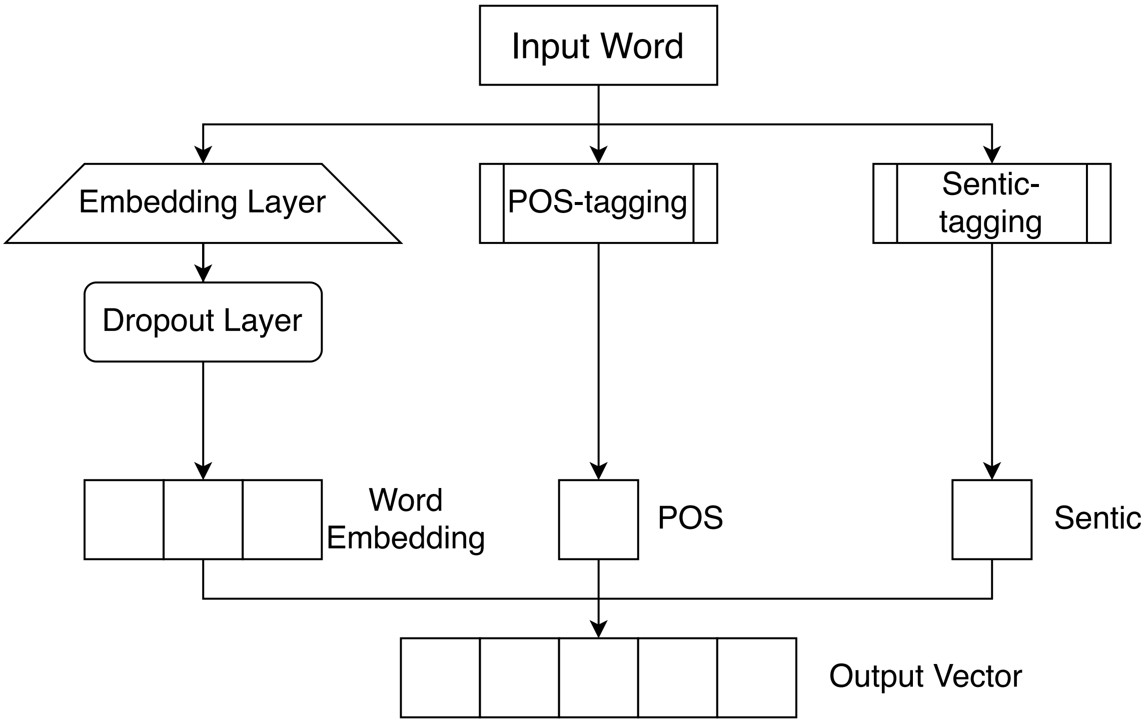


Fig. 3 Word transform to vectors process

Comparing the operation of the BLSTM model to human reading behaviour, it would be like reading each word from the beginning to the end of the sentence and analysing the sentence in both forward and backward directions. This allows human to interpret and analyse the meaning of the sentence including the use of grammar that might contain patterns in both forward and backward directions.

3.3.2 Convolutional Neural Network

CNN is a feed forward neural network that has, at least, a convolutional layer as a core component that automatically generates feature maps by sliding a filter over an image. Another important component is a pooling layer that is employed to reduce the size of feature map. Therefore, CNN is able to capture local features of text. The architecture is shown in Figure 4.

An input feature vector is first fed into the convolutional layer that allows the model to learn information from groups of words through a striding filter. A striding or sliding filter has a dimension of $w \times h$ where w is the length of feature vector and h is the number of words that the filter covers at a time. This leads to an output with a size of $s \times n$ where n is the number of nodes in the convolutional model. s is the number of strides that is equal to $h - (l - 1)$ and l is the number of words in a sentence. Then, the output from the convolutional layer passes through Rectified Linear Units (ReLU) activation function [80]. Because the vector from the input to the output layer has to be 1-D vector, 1-D dynamic max pooling with size of $s \times 1$ is required. It strides for n times and gives a 1-D output vector that goes to the dropout layer then to the output layer.

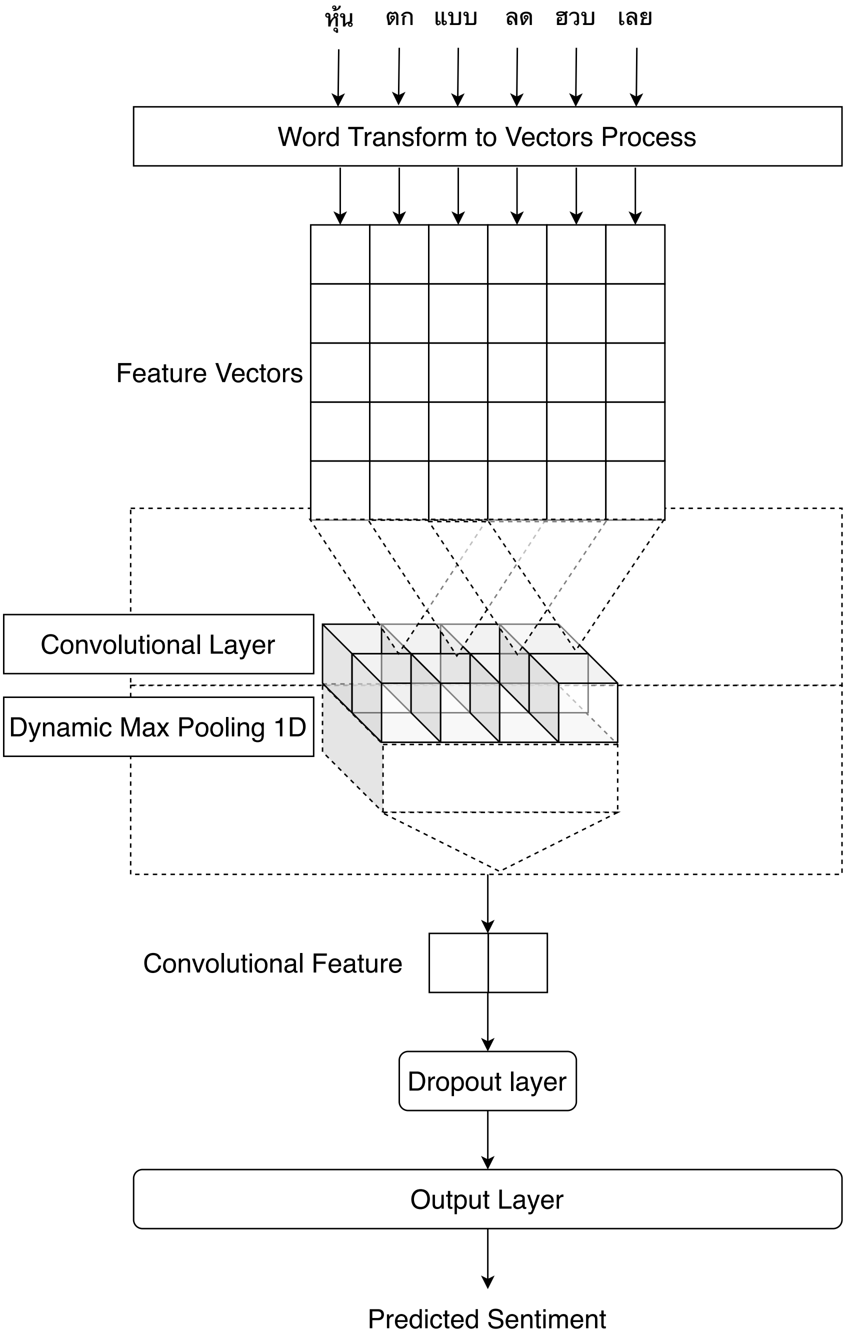


Fig. 4 Convolutional Neural Network for Sentiment Analysis

3.3.3 Hybrid Deep Learning Models

We proposed the following four different hybrids of deep learning models.

3.3.3.1 BLSTM-CNN

BLSTM-CNN is a hybrid deep learning model that combines CNN to BLSTM. The model aims to first learn sequences of words by BLSTM and capture local features by CNN. The model is shown in Figure 5. After a sentence is input into the model, it is feature-extracted and sent to BLSTM layer to learn the sequence of the sentence in both forward and backward directions. Then the output from the BLSTM–that has long-range dependency information of both directions–goes to CNN in order to extract local features of text.

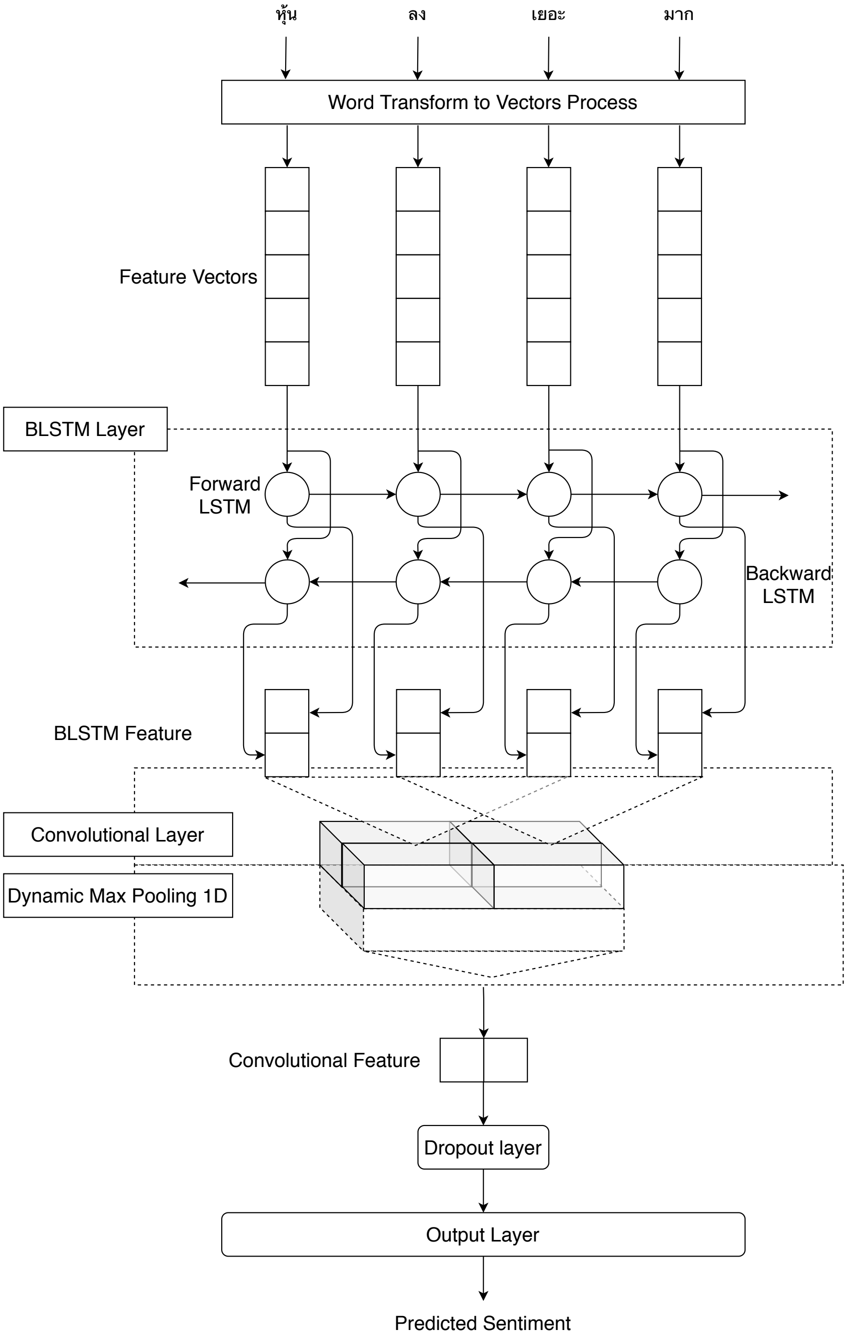


Fig. 5 BLSTM-CNN model for sentiment classification.

3.3.3.2 CNN-BLSTM

CNN-BLSTM is the other way around. The model aims to first learn local features of text by CNN and then long-range dependency between the sequence of words is learned by BLSTM. The model is shown in Figure 6. The output from convolutional layer goes to ReLU activation function. Then, the output–that has local features embedded–is fed into BLSTM layer to learn sequences in forward and backward directions. The hidden state of forward and backward directions is concatenated before it goes through the dropout layer and to the output layer.

3.3.3.3 BLSTM+CNN

This model learns the local features of a sequence of words in both directions at the same time. The

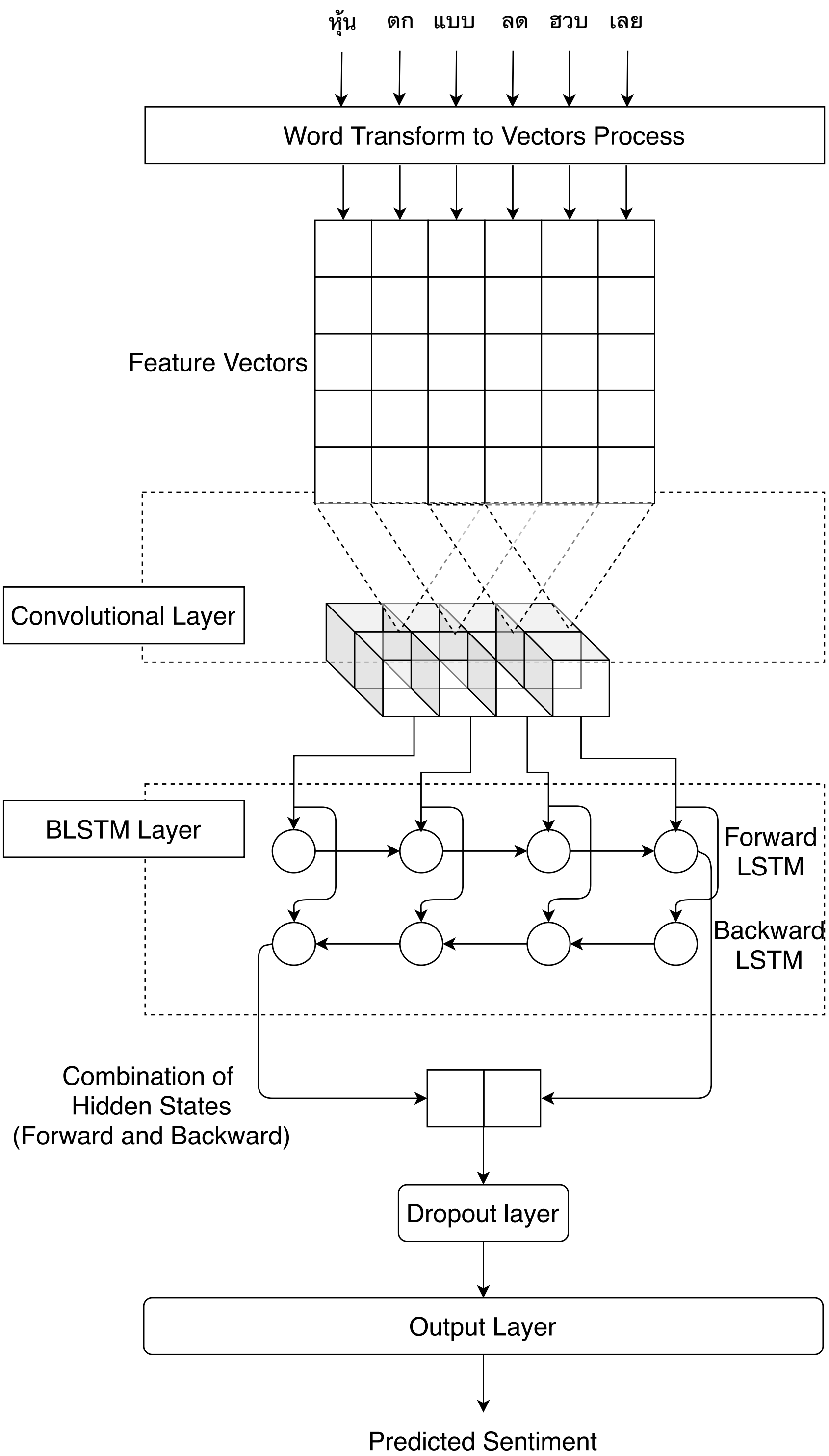


Fig. 6 CNN-BLSTM model for sentiment classification.

model is shown in Figure 7. An input sentence is feature-extracted then fed into BLSTM and CNN layers. The outputs from both layers are concatenated before they go through the dropout and output layers.

3.3.3.4 BLSTM×CNN

In this type of hybrid model, we simply ensembled both models by a soft voting scheme. The sentiment probability that BLSTM×CNN predicts is calculated by averaging the probabilities given by BLSTM

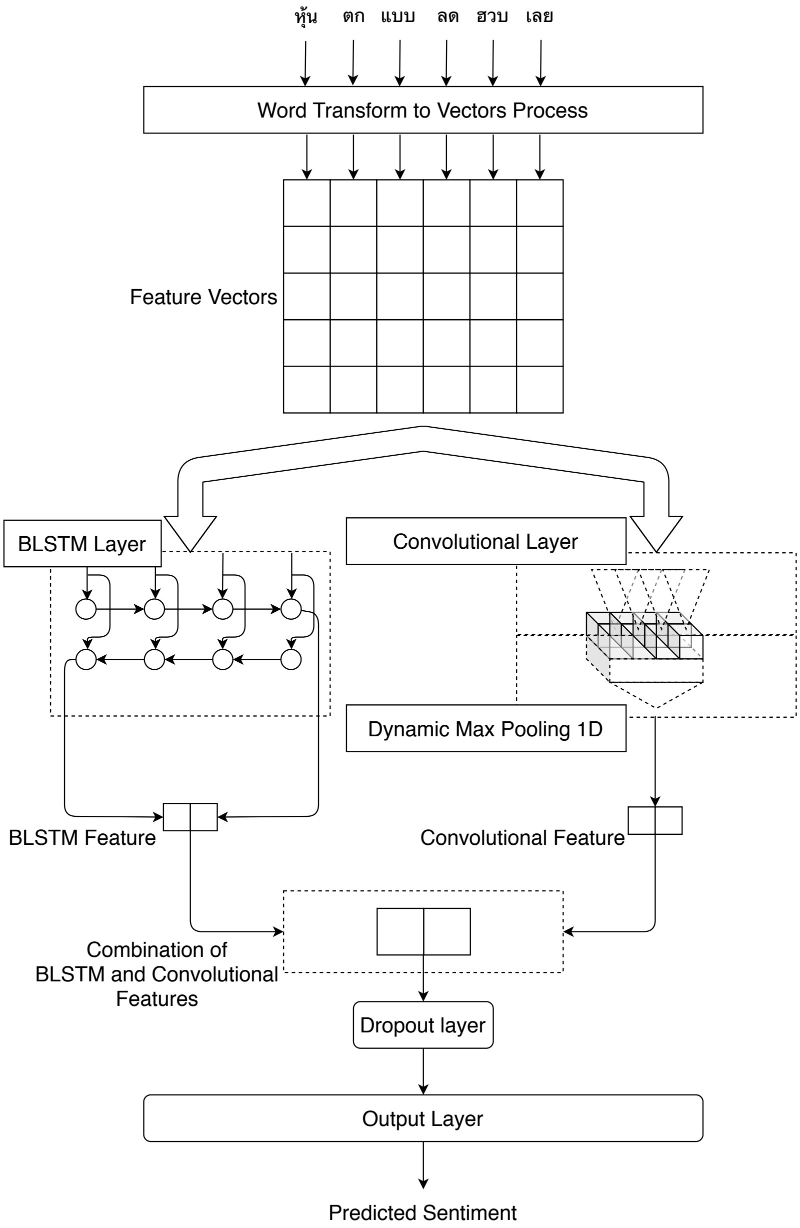


Fig. 7 BLSTM+CNN model for sentiment classification.

and CNN. The final predicted sentiment is that which has the highest probability. The model is shown in Figure 8.

4 Experimental Framework

4.1 Datasets

The proposed hybrid models–BLSTM-CNN, CNN-BLSTM, BLSTM+CNN BLSTM×CNN–were compared with their individual counterparts–CNN and BLSTM–on three datasets.

4.1.1 Wisesight Sentiment Dataset

This set, Wisesight, was collected from public pages in Facebook, Twitter, YouTube, Pantip.com and other web forums between 2016 and early 2019. The majority of the topics in this dataset were consumer

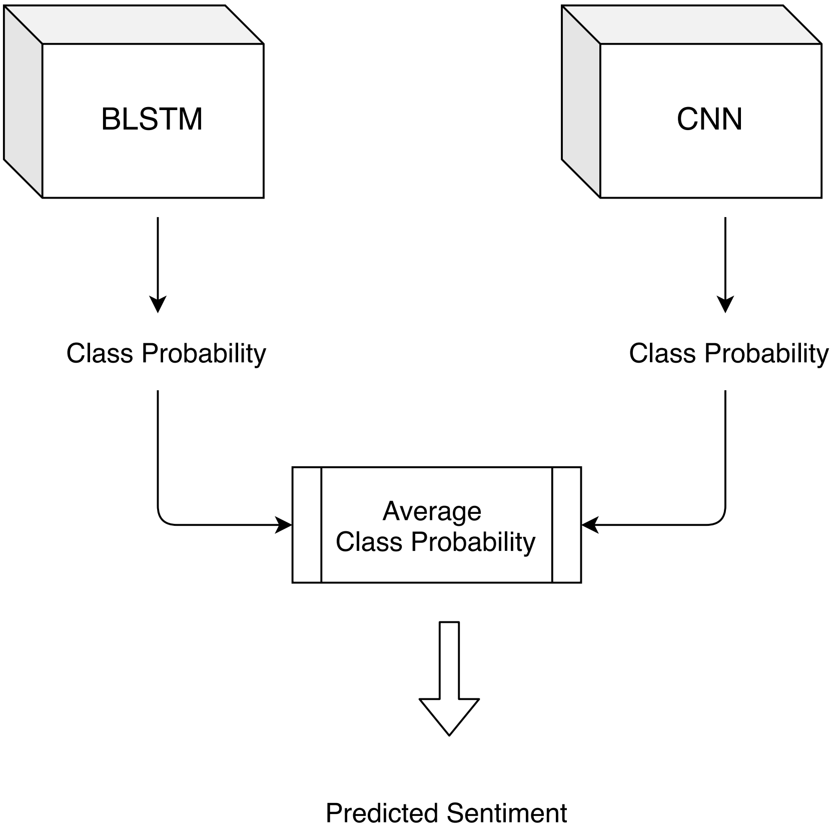


Fig. 8 BLSTM×CNN model for sentiment classification.

products and services. There were 26,740 messages that were divided into four classes–6,800 of negative sentiments, 14,500 of neutral sentiments, 4,700 of positive sentiments, and 500 of queries. The length of each message was between 1–428 words. It should be noted that this dataset was labelled by a group of annotators. Each message was given only one label by an annotator. The dataset is available to download at `https://github.com/PyThaiNLP/wisesight-sentiment`.

4.1.2 Thailand Economy Twitter Dataset

This dataset, ThaiEconTwitter, was proposed by [81]. This set was collected from Twitter. Tweets with two hashtags–#หุ้น (stock) and #เศรษฐกิจ (economic)– between 17 April 2017 and 5 May 2017 were retrieved. It consisted of 2,000 sentences and three classes–positive, neutral, and negative sentiments. Each sentence was given a label by one of three experts. In this work, we selected only a set of sentences that were given the same label from all three experts. Therefore, there were only 1,041 sentences comprising 608 negative sentiments, 84 neutral sentiments, and 349 positive sentiments.

4.1.3 The 40 Thai Chidren Tales Dataset

This dataset, ThaiTales, was first used in [19]. The dataset was collected from 40 Thai tales consisted of 1,964 sentences. Each sentence was labelled as one of three classes, i.e., positive, neutral or negative sentiment by three experts. All of the experts gave the same label to a sentence for only 1,115 sentences consisted of 309 sentences with positive sentiment, 508 sentences with neutral sentiment, and 298 sentences with negative sentiment. The dataset is available for download at `https://github.com/dsmlr/40-Thai-Children-Stories`.

4.2 Experiment Settings

The performance of the proposed hybrid models, i.e., BLSTM-CNN, CNN-BLSTM, BLSTM+CNN and BLSTM×CNN were compared with individual models, i.e., BLSTM and CNN. Three types of features were also compared together with their various combinations, i.e. Word Embedding (F_W), POS-tag (F_P), Sentic (F_S), $F_W + F_P$, $F_W + F_S$, $F_P + F_S$, and $F_W + F_P + F_S$. The experiments were conducted on three datasets–ThaiTales, ThaiEconTwitter and Wisesight–that are explained in the previous subsection. Each dataset was split in a stratified manner into three subsets–training, validation and test sets–at a ratio of 60:20:20. Hence, all of the subsets inherited the some of the same characteristics of the original dataset including class distribution and sentence length distribution. We employed Adam optimiser [82] with a learning rate of 0.001. Every model was trained for 300 epochs on ThaiTales and ThaiEconTwitter datasets and 50 epochs on Wisesight dataset. The reasons behind the setting of 50 epochs on Wisesight dataset were (i) the loss function converged at around the 30th epoch as shown in Figure 9 and (ii) the number of samples was very large leading to high computational cost. We performed grid searches to tune the many parameters of each algorithm. The search settings were as follows:

- BLSTM: the number of hidden nodes in BLSTM layer was either {16, 32, 64, 128, 256 or 512}.
- CNN: the number of hidden nodes in the filter was either {16, 32, 64, 128, 256 or 512} and the filter size was fixed at 3.
- BLSTM-CNN: the number of hidden node in BLSTM layer are {16, 32, 64, 128, 256, 512}, the number of hidden nodes in the filter were either {16, 32, 64, 128, 256 or 512} and the filter size was fixed at 3.
- The cases of CNN-BLSTM, BLSTM+CNN and BLSTM×CNN: were similar to that of BLSTM-CNN.

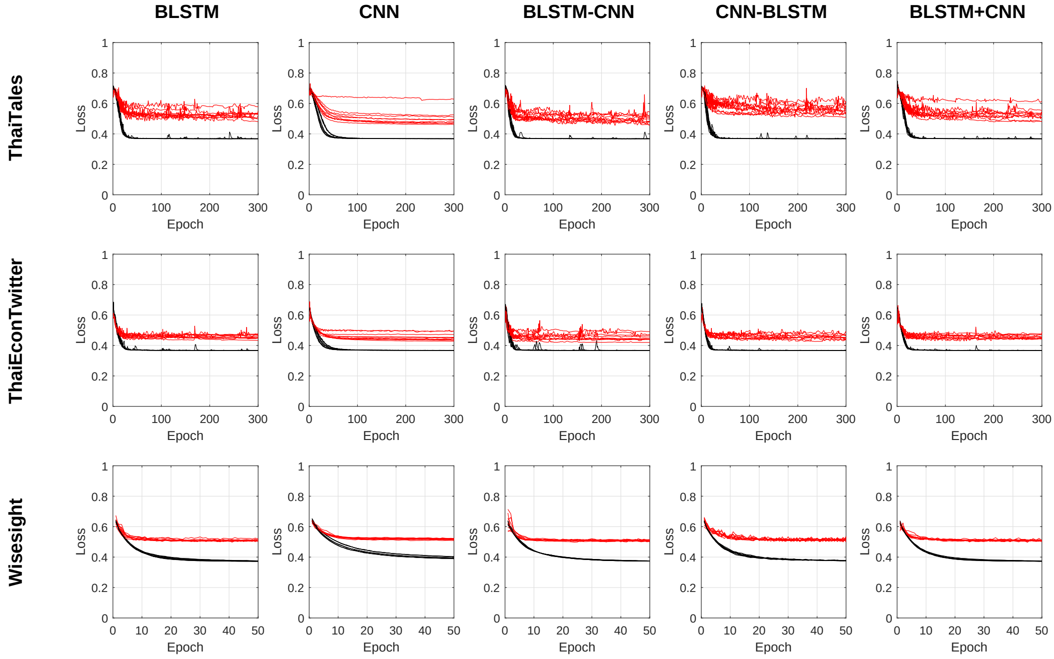


Fig. 9 Convergence curve of the loss function for each algorithm and dataset. Black and red indicate training and validation sets, respectively.

Dropout layers were employed in all models. They were between the embedding layer and output layer, and their dropout value was set to 0.5. Then, we selected optimal parameters based on F_1-score obtained in the validation process and explained in the following subsection. The optimal parameters were used as settings in the optimal model trained with the combined training and validation datasets, then the optimal model was evaluated on the test dataset. The above process was repeated with 10 different random splits.

4.3 Performance Evaluation

As our datasets were mostly imbalanced, we used F_1 as the performance measure. F_1 is a score that seeks to balance between precision and recall. It is calculated from the harmonic mean of precision and recall as follows:

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}, \tag{8}$$

where P is precision and R is recall that can be calculated as (9) and (10), respectively.

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$R = \frac{TP}{TP + FN} \tag{10}$$

where TP, FP, and FN denote true positive, false positive and false negative, respectively.

5 Results and Discussions

The fused deep learning models were evaluated with each of the three different features on the three datasets. Table 3 lists the average F_1-score across ten random splits. We first compared the performance of each individual feature, F_W, F_P and F_S. The best individual feature was F_W that yielded F_1-score of 0.6576 on average across all datasets, models, and ten random splits. This score is followed by F_P (0.4669) and F_S (0.4598). When two or more features were combined, they improved the overall performance. $F_W + F_S$ yielded the best contender at 0.6653, followed by $F_W + F_P + F_S$ (0.6647), $F_W + W_P$ (0.6593). It can be clearly seen that F_W is the most important feature because the top four contenders from all runs by all models on all datasets always included F_W as an individual feature or in combination with other feature(s), while this was not true for any other features. Combining F_P and F_S improved the overall performance to 0.5294 F_1-score, better than using each of them as an individual feature. Thus, combining feature led to improvement of overall performance.

There was variation in the performances shown in Table 3. Eighteen judges ranked the performance of each of the 7 features for every model and dataset (denoted as objects) based on F_1-scores. The ranks are shown in Table 4. The significance of the ranks in the table was tested by using Kendall Coefficient of Concordance (W) which turned out to be 0.8137 ($p < 0.01$ for 6 degrees of freedom). W is particularly useful for testing inter-judge or inter-test reliability [83]. The sum of the rank of every feature indicates the best overall rank of the objects [83] that suggests the following ordering:

$$F_W + F_P + F_S \sim F_W + F_S > F_W + F_P > F_W > F_P + F_S > F_S > F_P.$$

We further employed multiple t-test on the results in Table 3 to test the significance level of the difference between the means of two independent samples [84]. The test shows that each of the possible pairwise combinations is very highly significant ($p < 0.001$) except F_P-*vs*-F_S ($p = 0.5122$), F_W-*vs*-$F_W + F_P$ ($p = 0.3239$) and $F_W + F_S$-*vs*-$F_W + F_P$ ($p = 0.7013$) which are less conclusive.

According to the test, fusing features were able to clearly improve the prediction performance. Findings from two of our previous works support this observation [11,12]. For example, they suggest that it is possible that combining F_W and F_P could improve prediction performance. F_W could capture some syntactic information of a word while W_P could directly capture the grammatical type of a word. Pasupa and his colleagues has shown that intransitive verb (vi), transitive verb (vt), adverb (adv), common noun (n) and adjective (adj) were the most affective words that stimulate strong human emotions [19]. In addition, Pasupa and Seneewong Na Ayutthaya showed that simply including POS information for all words in a sentence could improve the prediction performance [11]. It was even better when POS information was included in RNN model with only some selected words based on the five types of POS (n, vi, vi, adj and adv).

Since Wisesight dataset contained much more samples than the others, we separately evaluated the ranks of the performance that every feature achieved on two different groups of datasets: (i) small-sized dataset group–ThaiTales and ThaiEconTwitter and (ii) large-sized dataset group–Wisesight. In the analysis of the small-sized dataset group, the ranks of all features led to the value of $W = 0.8585$ which was significant at $p < 0.01$. This high value enabled us to report with confidence that the following ranking is valid:

$$F_W + F_S > F_W + F_P + F_S > F_W + F_P > F_W > F_P + F_S > F_P > F_S.$$

In the analysis of the large-sized dataset group, the ranking of all features was:

$$F_W > F_W + F_P > F_W + F_P + F_S > F_W + F_S > F_P + F_S > F_S > F_P.$$

This ranking was significant at $p < 0.01$ with $W = 0.8254$. According to this ranking, any combinations with F_W are in the top ranks. Combining additional information–sentic and POS–was able to improve the performance of F_W on small-sized datasets. However, on the large-sized dataset, F_W was the best contender. Please note that the good outcomes of combined features were applicable to all models.

Moreover, Table 5 shows F_1-score achieved by every model for all datasets (averaged across all feature sets and ten random splits). Focusing on individual models–BLSTM and CNN–CNN outperformed BLSTM on ThaiTales dataset. Moreover, its performance was better than BLSTM for all features as shown in Table 3. However, BLSTM achieved 0.4694 F_1-score, better than CNN did at 0.4183, on Wisesight dataset. In addition, it also outperformed CNN for all features. This might be because the Thai tales were simple in vocabulary and grammatical structure; therefore, focusing on learning neighbouring words–local features–was more relevant than learning the sequences of sentences. On the other hand, users might use some difficult words or complex sentences in social media data. In ThaiEconTwitter, the F_1-score achieved by CNN was 0.6481 on average while that achieved by BLSTM was 0.6502. However, it is inconclusive whether CNN performed worse than BLSTM because its performance was worse than that of BLSTM for only in 4/7 cases.

The best performer was BLSTM-CNN that achieved 0.6100 F_1-score on average. Combining two models improved the performance in most cases, except for BLSTM×CNN on Wisesight and ThaiTales, and CNN-BLSTM on ThaiTales. However, the overall performances of the fused models were better than individual ones on average.

Considering only the fused models, it is clear that BLSTM-CNN was better than CNN-BLSTM in all cases (all features) on ThaiTales and ThaiEconTwitter, while it outperformed CNN-BLSTM in only 1/7 cases ($F_P + F_S$) on Wisesight as shown in Table 3. Nonetheless, CNN-BLSTM was the worst combination in all cases in ThaiTales and 5/7 cases in ThaiEconTwitter. Learning with concatenated features and with BLSTM and CNN (BLSTM+CNN) yielded better performances than learning by voting scheme with simple ensemble of both models (BLSTM×CNN) in all cases on Wisesight dataset. On the other hand, BLSTM×CNN performed better than BLSTM+CNN in 5/7 cases on ThaiEconTwitter and 6/7 cases on ThaiTales. The best combination on Wisesight was CNN-BLSTM (for 6/7 cases) and the worst was BLSTM×CNN (for all cases).

We further investigated the reason why BLSTM-CNN performed worse than CNN-BLSTM only on Wisesight. Figure 10 shows sentence length (number of words in an item) distributions of all three datasets. The sentence length distributions of Wisesight and ThaiEconTwitter datasets were clearly skewed to the right, i.e., the sentences in these two datasets were mostly short with a mode value of 5. On the other hand, such distribution of ThaiTales was close to normal (unskewed) with the mean of 16.84, the median of 15 and the mode of 15. According to the figure, the length of sentences in Wisesight varied from 1 to 428 words. Its range was much wider than those of the other two datasets because the longest sentence in ThaiEconTwitter was 74 words and 68 words in ThaiTales.

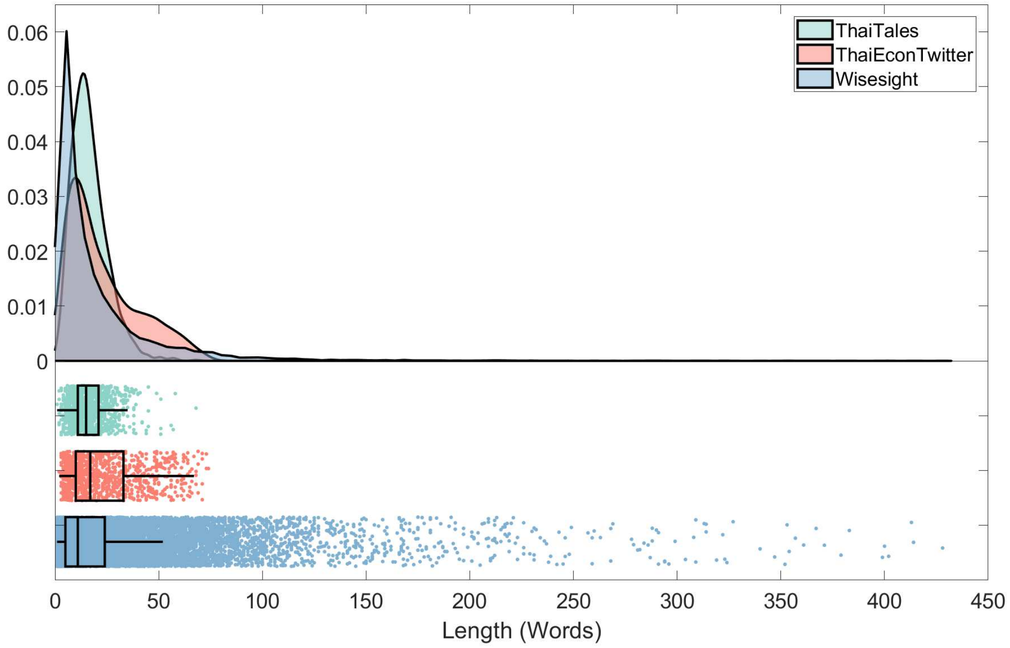


Fig. 10 Sentence length distribution on Wisesight, ThaiEconTwitter, and ThaiTales datasets.

Because of this variation, we divided each range into 10 equal intervals and tested the samples in each interval separately for each dataset. Then, the F_1-score for every interval was reported on all datasets as shown in Figure 11. BLSTM-CNN performances were better than those of CNN-BLSTM in all cases on ThaiTales and ThaiEconTwitter. On Wisesight dataset, CNN-BLSTM achieved better performances than BLSTM-CNN only for three intervals–[0–43], [87–129] and [173–215]–but the percentage of the samples in these three intervals were 88.08 %, 1.94 %, 0.25 %, respectively. Therefore, the overall performance of BLSTM-CNN dropped by a bit, meaning that CNN-BLSTM performed better than BLSTM-CNN for short sentences, while BLSTM-CNN outperformed CNN-BLSTM for long sentences on Wisesight dataset.

As there was a degree of variation in the ranking of the models, the significance test, Kendall Coefficient of Concordance, for evaluation of ranking of the models was again conducted. The six models were assigned ranks by 21 judges (across all features and datasets) as shown in Table 6. The computed W was 0.3263. This value was significant at $p < 0.01$. Given that the significant level of agreement between various rankings of the same set of models has been established, the best overall order of the model was based on the sum ranks. This gives the following ranks:

$$\text{BLSTM-CNN} > \text{BLSTM} + \text{CNN} > \text{BLSTM} \times \text{CNN} > \text{CNN-BLSTM} > \text{CNN} > \text{BLSTM}.$$

Again, a multiple t-test was conducted to test the significance level of the differences between the means of F_1-scores achieved by pairs of models. We tested every possible pairs of all models. The test showed that the differences for all pairwise combinations were very significant ($p < 0.001$), except for those for BLSTM-*vs*-CNN ($p = 0.3354$), CNN-*vs*-CNN-BLSTM ($p = 0.1779$), and BLSTM+CNN-*vs*-BLSTM×CNN ($p = 0.2866$) which were less conclusive.

In addition, the ranking of every model on the small size datasets led to a value of $W = 0.6257$ ($p < 0.01$), giving the following overall ranking,

$$\text{BLSTM-CNN} > \text{BLSTM} \times \text{CNN} > \text{BLSTM} + \text{CNN} > \text{CNN} > \text{BLSTM} > \text{CNN-BLSTM},$$

while the ranking of every feature on the large dataset was

$$\text{CNN-BLSTM} > \text{BLSTM-CNN} > \text{BLSTM} + \text{CNN} > \text{BLSTM} > \text{BLSTM} \times \text{CNN} > \text{CNN}.$$

The computed value of W was 0.9005 on the large dataset–significant at $p < 0.01$.

According to [85], the bigger sentiment lexicon is, the better prediction accuracy is. Therefore, we plotted the average ratios of the number of words with sentic value to the number of words in the whole sentence for all datasets against the average F_1-scores across all models and features for the test sets, shown in Figure 12. There were no significant correlations between such F_1-scores and ratios for each dataset and across all datasets. Such ratios between ThaiEconTwitter and Wisesight datasets were along the same line but the F_1-scores were far different. On the other hand, such ratio for ThaiTales dataset was the biggest among the three datasets but such F_1-scores for it were smaller than those for ThaiEconTwitter which also had a smaller value of such ratio.

In addition, we show the confusion matrices of BLSTM-CNN in conjunction with $F_W + F_P + F_S$ for each dataset as this combination of classifiers was the best contender according to our analysis, shown in Figure 13. Regarding misclassified samples in ThaiTales dataset, the model had a tendency to predict negative samples to neutral rather than positive samples. In addition, positive samples were misclassified as neutral samples. This shows that the classifier tended to predict the model towards the majority class. This observation is also applicable to the remaining datasets. The majority class of ThaiEconTwitter was negative class. Those misclassified samples of positive and neutral samples were classified as negative class. It should be noted that the model performed well in predicting each class on ThaiTales and ThaiEconTwitter datasets, but did not do so on Wisesight dataset. Most misclassified samples were classified as neutral as expected, i.e.,toward the majority class. The model correctly classified 4,741 positive samples from the total of 9,820 positive samples (48.3 %), while it classified 4,165 positive samples as neutral samples (42.4 % of positive samples). Likewise, only 283 question class samples from a total of 1,100 samples were correctly classified (25.5 %), while 617 neutral samples were correctly classified (55.6 % of question class samples). Overall, all tested classifiers tended to bias towards the majority class.

We further performed an error analysis on what caused the error in the best model–BLSTM-CNN with $F_W + F_P + F_S$ feature. Examples of prediction errors are shown in Figure 14.

The sentence in Example 1 can be divided into two parts: (1) ประเทศญี่ปุ่นบริโภค ของในประเทศมากเว่อร์ and (2) มองมุมหนึ่งก็ชาตินิยม อีกมุมคือสนับสนุนเศรษฐกิจภาย ในประเทศไปอีก. The first part is a sarcastic remark that Japanese consume too much of only Japanese-made products, with a negative sentiment of over-consumption from the Thai word "เว่อร์", which is a shortened pronunciation of the English word "over". The second part, มองมุมหนึ่งก็ชาตินิยม อีกมุมคือสนับสนุน เศรษฐกิจภายในประเทศไปอีก, is a remark with a positive sentiment that supports the idea that nationalism benefits domestic economy. The sentiment of this two-part sentence should be positive, but BLSTM-CNN predicted that it was negative because it focused on the first part and did not comprehend that part was a sarcastic remark.

The sentence in Example 2 also can be divided into two parts: (1) รำคาญ and (2) ถ้าถามว่างานกีฬาอันไหนที่ global สุด ก็คือโอลิมปิกอะค่ะ เกือบทุกชาติเข้าร่วม แต่ละประเทศก็แย่งกันเป็นเจ้าภาพสุด เพื่อแสดงศักยภาพทางเศรษฐกิจ เผยแพร่วัฒนธรรมของตน. The words in the first part clearly convey a negative sentiment (2) of getting annoyed (by), while the words in the second part state that Olympic game is a world-class athletic event that every country wants to host to benefit its own economy, which convey a positive sentiment. The sentiment of the whole sentence should be negative because of the annoyance, but BLSTM-CNN focused on the second part and predicted that the sentiment of the whole sentence was positive.

6 Conclusion

This paper proposes a Thai sentiment analysis framework that includes data preprocessing steps, feature extraction, and model construction to perform the task. In addition, we propose a Thai-SenticNet5 corpus built on SenticNet5 in association with LEXiTRON, Volubilis, and Thai WordNet. Furthermore, three hybrid deep learning models–BLSTM-CNN, CNN-BLSTM, BLSTM+CNN, and BLSTM×CNN–are proposed and evaluated on three datasets: ThaiTales, ThaiEconTwitter, and Wisesight. Three different types of features–Word embedding, POS-tag, and Sentic–were used to represent meaning, POS, and sentiment of a word, respectively. Apart from these three sets of features, we also evaluated all of their combinations. The results show that feature combination was able to improve the overall performance of sentiment analysis. The best candidate feature was a combination of word embedding, POS, and sentic features that led to the highest F_1-score. Moreover, the results demonstrate the enhancement of task performance aided by hybrid deep learning models. The experimental results show that BLSTM-CNN was the overall best contender.

As mentioned, our datasets were mostly imbalanced, but the current models did not consider the class imbalance. This problem cause the model to bias toward the majority class. In future work, we will consider applying focal loss that can handle the class imbalance problem as it was successfully evaluated on image data, e.g. red blood cell classification [86]. Also, a more recent version of SenticNet integrates symbolic models and subsymbolic methods to encode meaning and learn syntactic patterns from data [87]. It can be employed to improve the overall performance of sentiment analysis task.

Acknowledgements This research was supported by the Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.
Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

1. Bright J, Margetts H, Hale S, Yasseri T. The Use of Social Media for Research and Analysis: A Feasibility Study. Oxford Internet Institute, University of Oxford; 2014. 13.
2. Herhold K. Why Digital Marketing Is an Essential Investment; 2014. (Accessed: 01.07.2019). https://www.business2community.com/digital-marketing/why-digital-marketing-is-an-essential-investment-02129487.
3. Cambria E, Grassi M, Hussain A, Havasi C. Sentic Computing for social media marketing. Multimedia Tools and Applications. 2012;59(2):557–577.
4. Cambria E. Affective Computing and Sentiment Analysis. IEEE Intelligent Systems. 2016;31(2):102–107.
5. Kim Y. Convolutional Neural Networks for Sentence Classification. In: Proceedings of the International Conference on Empirical Methods in Natural Language Processing, (EMNLP 2014), 25-29 October 2014, Doha, Qatar; 2014. p. 1746–1751.
6. Wang Y, Huang M, Zhu X, Zhao L. Attention-based LSTM for Aspect-level Sentiment Classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016), 1-4 November 2016, Austin, Texas, USA; 2016. p. 606–615.
7. Plank B, Søgaard A, Goldberg Y. Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), 7-12 August 2016, Berlin, Germany; 2016. p. 412–418.

8. Wang P, Li Z, Hou Y, Li W. Combining convnets with hand-crafted features for action recognition based on an HMM-SVM classifier. arXiv preprint arXiv:160200749. 2016;.
9. Chavaltada C, Pasupa K, Hardoon DR. Combining Multiple Features for Product Categorisation by Multiple Kernel Learning. In: Proceedings of the 14th International Conference on Computing and Information Technology (IC2IT2018), 5-6 July 2018, Chiang Mai, Thailand; 2018. p. 3–12.
10. Pasupa K, Sunhem W, Loo CK. A Hybrid Approach to Build Face Shape Classifier for Hairstyle Recommender System. Expert Systems With Applications. 2019;120:14–32.
11. Pasupa K, Seneewong Na Ayutthaya T. Thai Sentiment Analysis with Deep Learning Techniques: A Comparative Study based on Word Embedding, POS-Tag, and Sentic Features. Sustainable Cities and Society. 2019;50:101615.
12. Seneewong Na Ayutthaya T, Pasupa K. Thai Sentiment Analysis via Bidirectional LSTM-CNN Model with Embedding Vectors and Sentic Features. In: Proceedings of the 13th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP 2018), 15-17 November 2018, Pattaya, Thailand; 2018. p. 84–89.
13. Wang J, Yu LC, Lai KR, Zhang X. Dimensional sentiment analysis using a regional CNN-LSTM model. In: Proceedings of the 54th Annual Meeting of the Association of Computational Linguistics (ACL 2016), 7-12 August 2016, Berlin, Germany; 2016. p. 225–230.
14. Lin S, Xie H, Yu LC, Lai KR. SentiNLP at IJCNLP-2017 task 4: Customer feedback analysis using a Bi-LSTM-CNN model. In: Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP 2017), 27 November-1 December 2017, Taipei, Taiwan; 2017. p. 149–154.
15. Minaee S, Azimi E, Abdolrashidi AA. Deep-Sentiment: Sentiment analysis using ensemble of CNN and Bi-LSTM models. arXiv preprint arXiv:190404206. 2019;.
16. Lertsuksakda R, Netisopakul P, Pasupa K. Thai Sentiment Terms Construction using the Hourglass of Emotions. In: Proceedings of the 6th International Conference on Knowledge and Smart Technology (KST 2014), 30-31 January 2014, Chonburi, Thailand; 2014. p. 46–50.
17. Surin B. Volubilis 9.5 (2019.1)–107,000 Entries; 2019. (Accessed: 01.02.2019). `http://belisan-volubilis.blogspot.com`.
18. Bird S, Klein E, Loper E. Natural Language Processing with Python. 1st ed. O'Reilly Media, Inc.; 2009.
19. Pasupa K, Netisopakul P, Lertsuksakda R. Sentiment Analysis on Thai Children Stories. Artificial Life and Robotics. 2016;21(3):357–364.
20. Ofek N, Poria S, Rokach L, Cambria E, Hussain A, Shabtai A. Unsupervised Commonsense Knowledge Enrichment for Domain-Specific Sentiment Analysis. Cognitive Computation. 2016;8(3):467–477.
21. Oneto L, Bisio F, Cambria E, Anguita D. Semi-supervised Learning for Affective Common-Sense Reasoning. Cognitive Computation. 2017;9(1):18–42.
22. Wang J, Sun C, Li S, Wang J, Si L, Zhang M, et al. Human-Like Decision Making: Document-level Aspect Sentiment Classification via Hierarchical Reinforcement Learning. In: Inui K, Jiang J, Ng V, Wan X, editors. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), 3-7 November 2019, Hong Kong, China. Association for Computational Linguistics; 2019. p. 5580–5589.
23. Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau R. Sentiment analysis of twitter data. In: Proceedings of the Workshop on Language in Social Media (LSM 2011), 23 June 2011, Portland, Oregon, USA; 2011. p. 30–38.
24. Phienthrakul T, Kijsirikul B, Takamura H, Okumura M. Sentiment Classification with Support Vector Machines and Multiple Kernel Functions. In: Proceedings of the 16th International Conference on Neural Information Processing (ICONIP 2009), 1-5 December, 2009, Bangkok, Thailand. vol. 5864 of Lecture Notes in Computer Science; 2009. p. 583–592.
25. Flender M, Gips C. Sentiment Analysis of a German Twitter-Corpus. In: Proceedings of the Lernen, Wissen, Daten, Analysen Conference (LWDA 2017), 11-13 September 2017, Rostock, Germany. vol. 1917 of CEUR Workshop Proceedings; 2017. p. 25.
26. Abdaoui A. French Social Media Mining: Expertise and Sentiment. Université Montpellier; 2016.
27. Peng H, Cambria E, Hussain A. A Review of Sentiment Analysis Research in Chinese Language. Cognitive Computation. 2017;9(4):423–435.
28. Hussein DMEDM. A survey on sentiment analysis challenges. Journal of King Saud University–Engineering Sciences. 2018;30(4):330–338.
29. Vilares D, Peng H, Satapathy R, Cambria E. BabelSenticNet: A Commonsense Reasoning Framework for Multilingual Sentiment Analysis. In: Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI 2018), 18-21 November 2018, Bangalore, India; 2018. p. 1292–1298.
30. Sriphaew K, Takamura H, Okumura M. Sentiment Analysis for Thai Natural Language Processing. In: Proceedings of the 2nd Thailand-Japan International Academic Conference (TJIA 2009), 20 November 2009, Kyoto, Japan; 2009. p. 123–124.
31. Boonkwan P; 2017. Personal Communication.
32. Inrak P, Sinthupinyo S. Applying latent semantic analysis to classify emotions in Thai text. In: Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICCET 2010), 16-18 April 2010, Chengdu, China; 2010. p. 450–454.

33. Haruechaiyasak C, Kongthon A, Palingoon P, Sangkeettrakarn C. Constructing Thai opinion mining resource: A case study on hotel reviews. In: Proceedings of the 8th Workshop on Asian Language Resources, 21-22 August 2010, Beijing, China; 2010. p. 64–71.
34. Haruechaiyasak C, Kongthon A, Palingoon P, Trakultaweekoon K. S-Sense: A Sentiment Analysis Framework for Social Media Sensing. In: Proceedings of the IJCNLP Workshop on Natural Language Processing for Social Media (SocialNLP 2013), 14-18 October 2013, Nagoya, Japan; 2013. p. 6–13.
35. Damdoung W, Chanlekha H, Kawtrakul A. A context-induced bootstrapping approach for constructing contextual-dependent Thai sentiment lexicon. In: Proceedings of the 10th International Symposium on Natural Language Processing (SNLP 2013), 28 October-30 October 2013, Phuket, Thailand; 2013. p. 225–230.
36. Sarakit P, Theeramunkong T, Haruechaiyasak C, Okumura M. Classifying emotion in Thai youtube comments. In: Proceedings of the 6th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES 2015), 22-24 March 2015, Hua-hin, Thailand; 2015. p. 1–5.
37. Netisopakul P, Chattupan A. Thai Stock News Sentiment Classification using Wordpair Features. In: Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, (PACLIC 2015), 30 October-1 November 2015, Shanghai, China; 2015. p. 188–195.
38. Vateekul P, Koomsubha T. A study of sentiment analysis using deep learning techniques on Thai Twitter data. In: Proceedings of the 13th International Joint Conference on Computer Science and Software Engineering (JCSSE 2016), 13-15 July 2016, Khon Kaen, Thailand; 2016. p. 1–6.
39. Cambria E, Havasi C, Hussain A. SenticNet 2: A semantic and affective resource for opinion mining and sentiment analysis. In: Proceedings of the 25 International Florida Artificial Intelligence Research Society Conference (FLAIRS 2012), 23-25 May 2012, Florida, USA; 2012. p. 202–207.
40. Netisopakul P, Pasupa K, Lertsuksakda R. Hypothesis Testing based on Observation from Thai Sentiment Classification. Artificial Life and Robotics. 2017;22(2):184–190.
41. Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2018;8(4):e1253.
42. Li Y, Pan Q, Yang T, Wang S, Tang J, Cambria E. Learning Word Representations for Sentiment Analysis. Cognitive Computation. 2017;9(6):843–851.
43. Severyn A, Moschitti A. Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2015), 9-13 August 2015, Santiago, Chile; 2015. p. 959–962.
44. Graves A, Fernández S, Schmidhuber J. Bidirectional LSTM networks for improved phoneme classification and recognition. In: Proceedings of the 15th International Conference on Artificial Neural Networks (ICANN 2005), 11-15 September 2005, Warsaw, Poland; 2005. p. 799–804.
45. Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks. 2005;18(5-6):602–610.
46. Xu K, Xie L, Yao K. Investigating LSTM for punctuation prediction. In: Proceedings of the 10th International Symposium on Chinese Spoken Language Processing (ISCSLP 2016), 17-20 October 2016, Tianjin, China; 2016. p. 1–5.
47. Pang B, Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005), 25-30 June 2005, Michigan, USA; 2005. p. 115–124.
48. Ouyang X, Zhou P, Li CH, Liu L. Sentiment analysis using convolutional neural network. In: Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, (CIT/IUCC/DASC/PICom 2015), 26-28 October 2015, Liverpool, UK; 2015. p. 2359–2364.
49. Nowak J, Taspinar A, Scherer R. LSTM recurrent neural networks for short text and sentiment classification. In: Proceedings of the International Conference on Artificial Intelligence and Soft Computing (ICAISC 2017), 11-15 June 2017, Zakopane, Poland; 2017. p. 553–562.
50. Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng A, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP 2013), 18-21 October 2013, Washington, USA; 2013. p. 1631–1642.
51. Yu LC, Lee LH, Hao S, Wang J, He Y, Hu J, et al. Building Chinese affective resources in valence-arousal dimensions. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016), 12-17 June 2016, San Diego California, USA; 2016. p. 540–545.
52. Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011), 19-24 June 2011, Oregon, USA; 2011. p. 142–150.
53. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All you Need. In: Proceedings of the Advances in Neural Information Processing Systems (NIPS 2017), 4-9 December 2017, CA, USA; 2017. p. 5998–6008.
54. Ma D, Li S, Zhang X, Wang H. Interactive attention networks for aspect-level sentiment classification. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), 19-25 August 2017, Melbourne, Australia; 2017. p. 4068–4074.

55. Pontiki M, Galanis D, Pavlopoulos J, Papageorgiou H, Androutsopoulos I, Manandhar S. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), August 2014, Dublin, Ireland; 2014. p. 27–35.
56. PyThaiNLP. PyThaiNLP: Thai Natural Language Processing in Python. GitHub; 2019. (Accessed: 01.02.2019). `https://github.com/PyThaiNLP/pythainlp`.
57. Meknavin S, Charoenpornsawat P, Kijsirikul B. Feature-based Thai Word Segmentation. In: Proceedings of the Natural Language Processing Pacific Rim Symposium (NLPRS 1997), 2-4 December 1997, Phuket, Thailand; 1997. p. 1–6.
58. Aroonmanakun W. Collocation and Thai word segmentation. In: Proceedings of the 5th Symposium on Natural Language Processing (SNLP) & 5th Oriental COCOSDA Workshop, 9-11 May 2002, Huahin, Thailand; 2002. p. 68–75.
59. Haruechaiyasak C, Kongyoung S, Dailey M. A comparative study on Thai word segmentation approaches. In: Proceedings of the 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2008), 14-17 May 2008, Krabi, Thailand; 2008. p. 125–128.
60. Theeramunkong T, Sornlertlamvanich V, Tanhermhong T, Chinnan W. Character Cluster Based Thai Information Retrieval. In: Proceedings of the 5th International Workshop Information Retrieval with Asian Languages (IRAL 2002), 30 September-1 October 2000, Hong Kong, China; 2000. p. 75–80.
61. Norvig P. How to Write a Spelling Corrector; 2016. (Accessed: 01.02.2019). `https://norvig.com/spell-correct.html`.
62. Sornlertlamvanich V, Takahashi N, Isahara H. Thai part-of-speech tagged corpus: ORCHID. In: Proceedings of the Oriental COCOSDA Workshop; 1998. p. 131–138.
63. Nivre J, De Marneffe MC, Ginter F, Goldberg Y, Hajic J, Manning CD, et al. Universal dependencies v1: A multilingual treebank collection. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016); 2016. p. 1659–1666.
64. Eisner B, Rocktäschel T, Augenstein I, Bošnjak M, Riedel S. emoji2vec: Learning Emoji Representations from their Description. In: Proceedings of the 4th International Workshop on Natural Language Processing for Social Media, Austin, TX, USA; 2016. p. 48–54.
65. Kim T, Wurster K. emoji terminal output for Python. GitHub; 2015. (Accessed: 01.02.2019). `https://github.com/carpedm20/emoji`.
66. Cambria E, Speer R, Havasi C, Hussain A. Senticnet: A publicly available semantic resource for opinion mining. In: Proceedings of the 2010 AAAI Fall Symposium: Commonsense Knowledge, 11-13 November 2010, Arlington, Virginia, USA. vol. FS-10-02 of AAAI Technical Report; 2010. p. 14–18.
67. Wang J, Oard DW. Combining bidirectional translation and synonymy for cross-language information retrieval. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2006); 2006. p. 202–209.
68. Thoongsup S, Robkop K, Mokarat C, Sinthurahat T, Charoenporn T, Sornlertlamvanich V, et al. Thai wordnet construction. In: Proceedings of the 7th Workshop on Asian Language Resources (ALR 2009), 6-7 August 2009, Singapore; 2009. p. 139–144.
69. Netisopakul P, Thong-iad K. Thai sentiment resource using Thai WordNet. In: Proceedings of the 12th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS 2018), 4-6 July 2018, Matsue, Japan; 2018. p. 329–340.
70. Cambria E, Poria S, Hazarika D, Kwok K. SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18), 2-7 February 2018, Louisiana, USA; 2018. p. 1795–1802.
71. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781. 2013;.
72. Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. In: Proceedings of the International Conference on Empirical Methods in Natural Language Processing, (EMNLP 2014), 25-29 October 2014, Doha, Qatar; 2014. p. 1532–1543.
73. Howard J, Ruder S. Universal language model fine-tuning for text classification. arXiv preprint arXiv:180106146. 2018;.
74. Polpanumas C. ULMFit Language Modeling, Text Feature Extraction, and Text Classification in Thai Language. GitHub; 2019. (Accessed: 01.02.2019). `https://github.com/cstorm125/thai2fit`.
75. Merity S, Keskar NS, Socher R. Regularizing and optimizing LSTM language models. arXiv preprint arXiv:170802182. 2017;.
76. Cambria E, Livingstone A, Hussain A. The hourglass of emotions. In: Cognitive behavioural systems. Springer; 2012. p. 144–157.
77. Susanto Y, Livingstone A, Ng BC, Cambria E. The hourglass model revisited. IEEE Intelligent Systems. 2020;35(5).
78. Sundermeyer M, Schlüter R, Ney H. LSTM neural networks for language modeling. In: Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH 2012); 2012. p. 1–4.
79. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014;15(1):1929–1958.
80. Agarap AF. Deep learning using rectified linear units (ReLU). arXiv preprint arXiv:180308375. 2018;.

81. Khunkwang P. A dictionary-based sentiment classification approach for business news on Twitter [B.Sc. Thesis]. Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang. Bangkok, Thailand; 2017.

82. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014;.

83. Siegel S, Castellian NJ. Nonparametric statistics for the behavioral sciences. 2nd ed. Singapore: McGraw-Hill; 1988.

84. Siegel AF. Multiple t Tests: Some Practical considerations. TESOL Quarterly. 1990;24(4):773–775.

85. Abdi A, Shamsuddin SM, Hasan S, Piran J. Deep learning-based sentiment classification of evaluative text based on multi-feature fusion. Information Processing and Management. 2019;56:1245–1259.

86. Pasupa K, Vatathanavaro S, Tungjitnob S. Convolutional Neural Networks based Focal Loss for Class Imbalance Problem: A Case Study of Canine Red Blood Cells Morphology Classification. Journal of Ambient Intelligence and Humanized Computing. 2020;.

87. Cambria E, Li Y, Xing FZ, Poria S, Kwok K. SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In: Proceedings of the 29th ACM Conference on Information and Knowledge Management (CIKM 2020), 19-23 October 2020, Virtual Event, Ireland. Association for Computing Machinery; in press. p. 1–9.

Table 1 POS Mapping between ORCHID and UD POS-Tags [56].

The 47 ORCHID POS-tag	Description	The 17 UD POS-tag	Description
NCMN	Common noun	NOUN	Noun
NTTL	Title noun		
CNIT	Unit classifier		
CLTV	Collective classifier		
CMTR	Measurement classifier		
CFQC	Frequency classifier		
CVBL	Verbal classifier		
VACT	Active verb	VERB	Verb
VSTA	Stative verb		
NPRP	Proper noun	PROPN	Proper noun
NONM	Ordinal number	ADJ	Adjective
VATT	Attributive verb		
DONM	Determiner, ordinal number expression		
ADVN	Adverb with normal form	ADV	Adverb
ADVI	Adverb with iterative form		
ADVP	Adverb with prefixed form		
ADVS	Sentential adverb		
INT	Interjection	INTJ	Interjection
PPRS	Personal pronoun	PRON	Pronoun
PDMN	Demonstrative pronoun		
PNTR	Interrogative pronoun		
DDAN	Definite determiner, after noun without classifier in between	DET	Determiner
DDAC	Definite determiner, allowing classifier in between		
DDBQ	Definite determiner, between noun and classifier or preceding quantitative expression		
DDAQ	Definite determiner, following quantitative expression		
DIAC	Indefinite determiner, following noun; allowing classifier in between		
DIBQ	Indefinite determiner, between noun and classifier or preceding quantitative expression		
DIAQ	Indefinite determiner, following quantitative expression		
NCNM	Cardinal number	NUM	Numeral
NLBL	Label noun		
DCNM	Determiner, cardinal number expression		
XVBM	Pre-verb auxiliary, before negator	AUX	Auxiliary
XVAM	Pre-verb auxiliary, after negator		
XVMM	Pre-verb, before or after negator		
XVBB	Pre-verb auxiliary, in imperative mood		
XVAE	Post-verb auxiliary		
RPRE	Preposition	ADP	Adposition
JCRG	Coordinating conjunction	CCONJ	Coordinating conjunction
PREL	Relative pronoun	SCONJ	Subordinating conjunction
JSBR	Subordinating conjunction		
JCMP	Comparative conjunction		
FIXN	Nominal prefix	PART	Particle
FIXV	Adverbial prefix		
EAFF	Ending for affirmative sentence		
EITT	Ending for interrogative sentence		
NEG	Negator		
PUNC	Punctuation	PUNCT	Punctuation
		SYM	Symbol
		X	Other

Table 2 Number of verified words and the number of words that had a sentic vector in each corpuses.

Corpus	#Verified Words	#Words with Sentic Vector
LEXiTRON-Bi	2,871	2,871
Volubilis-Bi	10,732	6,832
LEXiTRON-Volubilis-Bi	11,820	7,952
Thai-WordNet	17,597	12,387
LEXiTRON-Volubilis-Bi + Thai-WordNet	22,952	15,146
LEXiTRON-Volubilis-Bi + Thai-WordNet (With words with no stop word)	23,093	15,247

Table 3 Performance comparison of all models with different combinations of features on three datasets. Performance was evaluated based on F_1-score averaged across ten random splits. Bold text represents the best results.

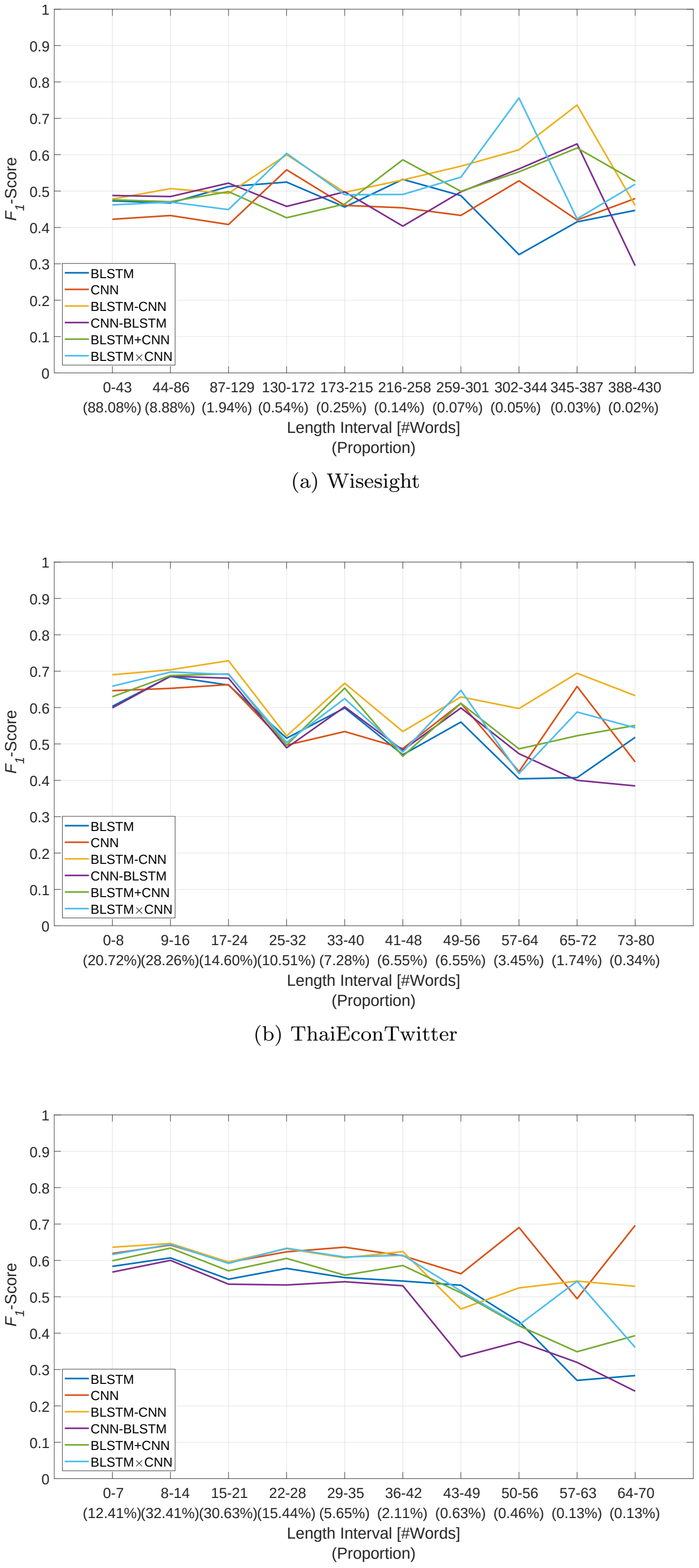
Dataset	Model	Feature						
		F_W	F_P	F_S	$F_W + F_P$	$F_W + F_S$	$F_P + F_S$	$F_W + F_P + F_S$
WiseSight	BLSTM	0.5515	0.3464	0.3511	0.5452	0.5483	0.3965	0.5470
	CNN	0.5045	0.2990	0.2545	0.5054	0.5088	0.3486	0.5074
	BLSTM-CNN	0.5503	0.3566	0.3675	0.5506	0.5483	0.4264	0.5521
	CNN-BLSTM	0.5587	0.3724	0.3864	**0.5609**	0.5580	0.4247	0.5574
	BLSTM+CNN	0.5533	0.3490	0.3526	0.5517	0.5498	0.4042	0.5499
	BLSTM×CNN	0.5461	0.3244	0.3169	0.5453	0.5412	0.3883	0.5437
ThaiEconTwitter	BLSTM	0.7291	0.5603	0.5304	0.7282	0.7294	0.5549	0.7190
	CNN	0.7087	0.5963	0.4506	0.6980	0.7226	0.6398	0.7209
	BLSTM-CNN	0.7667	0.6193	0.5939	**0.7775**	0.7685	0.6345	0.7707
	CNN-BLSTM	0.7138	0.5892	0.5748	0.7138	0.7083	0.6238	0.7147
	BLSTM+CNN	0.7308	0.5972	0.5516	0.7424	0.7453	0.6460	0.7422
	BLSTM×CNN	0.7513	0.5949	0.5527	0.7569	0.7625	0.6320	0.7456
ThaiTales	BLSTM	0.6770	0.4393	0.4717	0.6779	0.6872	0.5255	0.6980
	CNN	0.7208	0.4875	0.4920	0.7261	0.7393	0.6038	0.7272
	BLSTM-CNN	0.7297	0.4880	0.5202	0.7182	0.7416	0.5854	**0.7436**
	CNN-BLSTM	0.6324	0.4327	0.4944	0.6556	0.6690	0.5427	0.6768
	BLSTM+CNN	0.6898	0.4785	0.5044	0.6865	0.7124	0.5717	0.7124
	BLSTM×CNN	0.7215	0.4737	0.5106	0.7270	0.7345	0.5800	0.7357
	Average	0.6576	0.4669	0.4598	0.6593	0.6653	0.5294	0.6647

Table 4 Ranks assigned to 7 features by 18 judges (for each of the models and datasets) from Table 3 were based on F_1-score. Rank 1 indicates the best candidate according to the judges while rank 7 indicates the worst.

Dataset	Model	Feature						
		F_W	F_P	F_S	$F_W + F_P$	$F_W + F_S$	$F_P + F_S$	$F_W + F_P + F_S$
WiseSight	BLSTM	1	7	6	4	2	5	3
	CNN	4	6	7	3	1	5	2
	BLSTM-CNN	3	7	6	2	4	5	1
	CNN-BLSTM	2	7	6	1	3	5	4
	BLSTM+CNN	1	7	6	2	4	5	3
	BLSTM×CNN	1	6	7	2	4	5	3
ThaiEconTwitter	BLSTM	2	5	7	3	1	6	4
	CNN	3	6	7	4	1	5	2
	BLSTM-CNN	4	6	7	1	3	5	2
	CNN-BLSTM	2.5	6	7	2.5	4	5	1
	BLSTM+CNN	4	6	7	2	1	5	3
	BLSTM×CNN	3	6	7	2	1	5	4
ThaiTales	BLSTM	4	7	6	3	2	5	1
	CNN	4	7	6	3	1	5	2
	BLSTM-CNN	3	7	6	4	2	5	1
	CNN-BLSTM	4	7	6	3	2	5	1
	BLSTM+CNN	3	7	6	4	1.5	5	1.5
	BLSTM×CNN	4	7	6	3	2	5	1
	$\sum R$	52.5	117	116	48.5	39.5	91	39.5

Table 5 Average F_1 scores of every model on three datasets, averaged across all features and ten random splits. Bold–best.

Algorithm	Dataset			Average
	WiseSight	ThaiEconTwitter	ThaiTales	
BLSTM	0.4694	0.6502	0.5967	0.5721
CNN	0.4183	0.6481	0.6424	0.5696
BLSTM-CNN	0.4788	**0.7044**	**0.6467**	**0.6100**
CNN-BLSTM	**0.4884**	0.6626	0.5862	0.5791
BLSTM+CNN	0.4729	0.6794	0.6222	0.5915
BLSTM×CNN	0.4580	0.6851	0.6404	0.5945



(a) Wisesight

(b) ThaiEconTwitter

(c) ThaiTales

Fig. 11 F_1-scores for every sentence length interval in 10 test sets on all datasets, averaged across 10 runs and all features.

Table 6 Ranks assigned to 6 models by 21 judges (across all features and datasets)according to their F1 scores listed in Table 3.

Dataset	Feature	Model					
		BLSTM	CNN	BLSTM-CNN	CNN-BLSTM	BLSTM+CNN	BLSTM×CNN
WiseSight	F_W	3	6	4	1	2	5
	F_P	4	6	2	1	3	5
	F_S	4	6	2	1	3	5
	$F_W + F_P$	5	6	3	1	2	4
	$F_W + F_S$	3.5	6	3.5	1	2	5
	$F_P + F_S$	4	6	1	2	3	5
	$F_W + F_P + F_S$	4	6	2	1	3	5
ThaiEconTwitter	F_W	4	6	1	5	3	2
	F_P	6	3	1	5	2	4
	F_S	5	6	1	2	4	3
	$F_W + F_P$	4	6	1	5	3	2
	$F_W + F_S$	4	5	1	6	3	2
	$F_P + F_S$	6	2	3	5	1	4
	$F_W + F_P + F_S$	5	4	1	6	3	2
ThaiTales	F_W	5	3	1	6	4	2
	F_P	5	2	1	6	3	4
	F_S	6	5	1	4	3	2
	$F_W + F_P$	5	2	3	6	4	1
	$F_W + F_S$	5	2	1	6	4	3
	$F_P + F_S$	6	1	2	5	4	3
	$F_W + F_P + F_S$	5	3	1	6	4	2
	$\sum R$	98.5	92	36.5	81	63	70

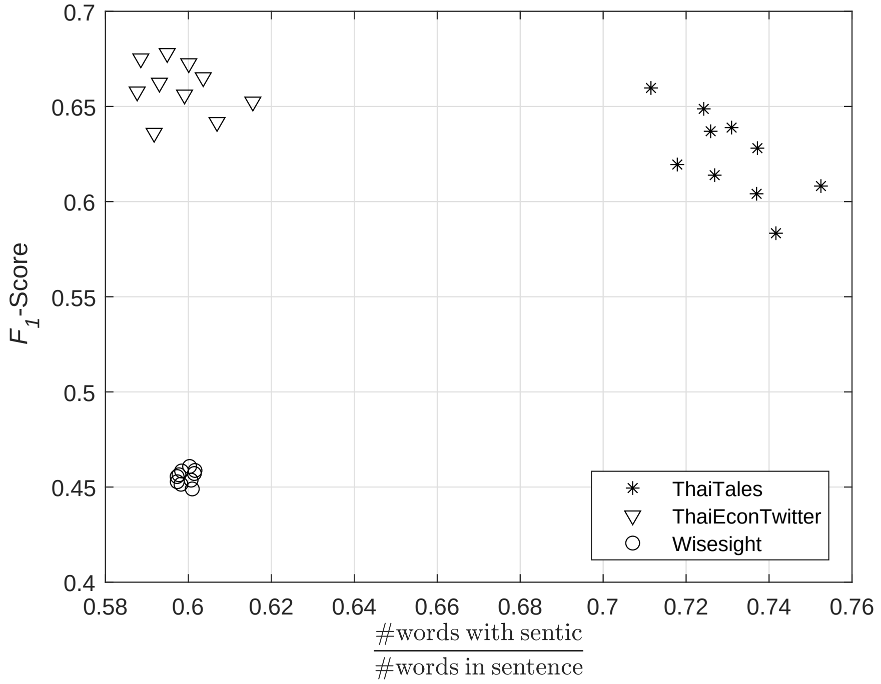
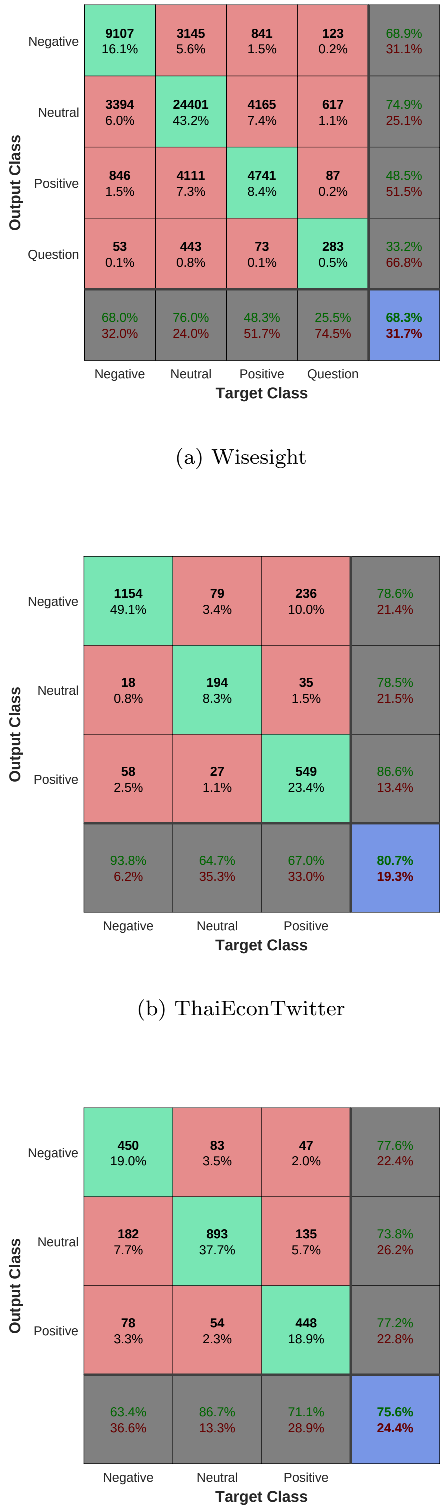


Fig. 12 Plot of average ratios of the number of words with sentic values to the number of words in a sentence against average F_1-scores achieved by every model and feature across all three datasets.



Output Class	Negative	Neutral	Positive	Question	
Negative	**9107** 16.1%	**3145** 5.6%	**841** 1.5%	**123** 0.2%	68.9% 31.1%
Neutral	**3394** 6.0%	**24401** 43.2%	**4165** 7.4%	**617** 1.1%	74.9% 25.1%
Positive	**846** 1.5%	**4111** 7.3%	**4741** 8.4%	**87** 0.2%	48.5% 51.5%
Question	**53** 0.1%	**443** 0.8%	**73** 0.1%	**283** 0.5%	33.2% 66.8%
	68.0% 32.0%	76.0% 24.0%	48.3% 51.7%	25.5% 74.5%	**68.3%** **31.7%**

Target Class

(a) Wisesight

Output Class	Negative	Neutral	Positive	
Negative	**1154** 49.1%	**79** 3.4%	**236** 10.0%	78.6% 21.4%
Neutral	**18** 0.8%	**194** 8.3%	**35** 1.5%	78.5% 21.5%
Positive	**58** 2.5%	**27** 1.1%	**549** 23.4%	86.6% 13.4%
	93.8% 6.2%	64.7% 35.3%	67.0% 33.0%	**80.7%** **19.3%**

Target Class

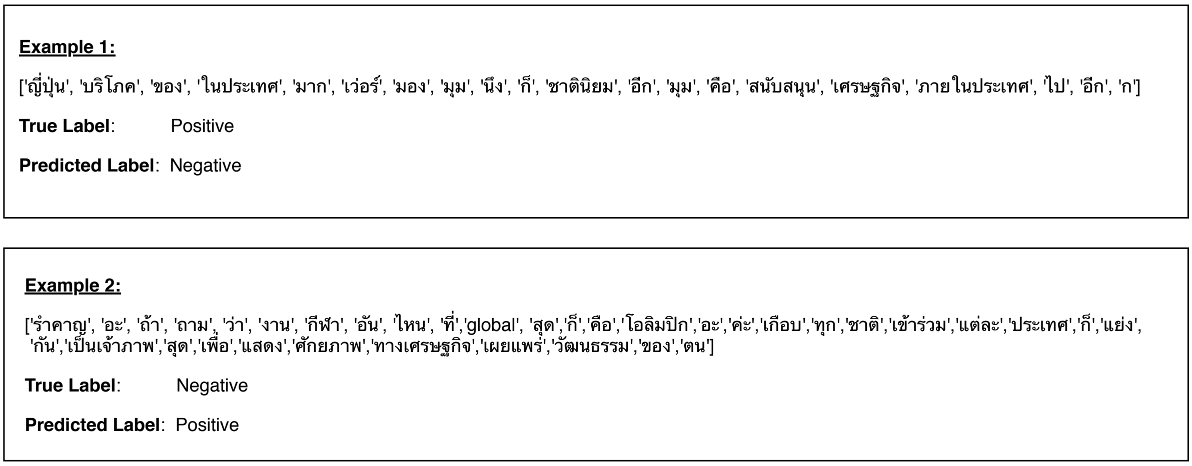
(b) ThaiEconTwitter

Output Class	Negative	Neutral	Positive	
Negative	**450** 19.0%	**83** 3.5%	**47** 2.0%	77.6% 22.4%
Neutral	**182** 7.7%	**893** 37.7%	**135** 5.7%	73.8% 26.2%
Positive	**78** 3.3%	**54** 2.3%	**448** 18.9%	77.2% 22.8%
	63.4% 36.6%	86.7% 13.3%	71.1% 28.9%	**75.6%** **24.4%**

Target Class

(c) ThaiTales

Fig. 13 Confusion matrices of BLSTM-CNN with $F_W + F_P + F_S$ features on test sets–from a total of 10 runs with different random splits.



Example 1:

['ญี่ปุ่น', 'บริโภค', 'ของ', ''ในประเทศ', 'มาก', 'เว่อร์', 'มอง', 'มุม', 'นึง', 'ก็', 'ชาตินิยม', 'อีก', 'มุม', 'คือ', 'สนับสนุน', 'เศรษฐกิจ', 'ภายในประเทศ', 'ไป', 'อีก', 'ก']

True Label: Positive

Predicted Label: Negative

Example 2:

['รำคาญ', 'อะ', 'ถ้า', 'ถาม', 'ว่า', 'งาน', 'กีฬา', 'อัน', 'ไหน', 'ที่','global', 'สุด','ก็','คือ','โอลิมปิก','อะ','ค่ะ','เกือบ','ทุก','ชาติ','เข้าร่วม','แต่ละ','ประเทศ','ก็','แย่ง', 'กัน','เป็นเจ้าภาพ','สุด','เพื่อ','แสดง','ศักยภาพ','ทางเศรษฐกิจ','เผยแพร่','วัฒนธรรม','ของ','ตน']

True Label: Negative

Predicted Label: Positive

Fig. 14 Examples of sentences that BLSTM-CNN predicted a wrong sentiment.