Headline2Vec: A CNN-based Feature for Thai Clickbait Headlines Classification

Natsuda Kaothanthong[1], Sarawoot Kongyoung[2], and Thanaruk Theeramunkong[3]

[1,3] Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani, Thailand
[2] NECTEC, National Science and Technology Development Agency, Pathum Thani, Thailand
E-mail: natsuda@siit.tu.ac.th, sarawoot.kon@nectec.or.th,thanaruk@siit.tu.ac.th

Received: May 1, 2020 / Revised: June 25, 2020 / Accepted: August 7, 2020

***Abstract*—Clickbait is an article title or a social media post that attracts readers to follow a link to the article's content. It is one of the major contributors to spread fake news. To prevent a wide-spread of fake news, it should be detected as soon as possible. This paper presents a content-based feature called *headline2vec* that is extracted from a concatenation layer of a convolutional neural network (CNN) on the well-known word2vec model for high dimensional word embeddings, to improve an automatic detection of Thai clickbait headlines. A pioneer dataset for Thai clickbait headlines is collected using a top-down strategy. In the experiment, we evaluate the headline2vec feature for Thai clickbait news detection using 132,948 Thai headlines where the CNN features are constructed using a non-static modelling technique with 50 dimensions of word2vec embedding with a window size of two, three, and four with epoch of 5. Using the proposed features, we compare three classifiers, naïve Bayes, support vector machine, and multilayer perceptron. The result shows that the headline2vec with multilayer perceptron achieves up to 93.89% accuracy and it outperform the sequential features that utilize *n*-gram with tf-idf.**

***Index Teams*—Thai Clickbait Detection, Text Feature, Convolutional Neural Network, Headline2Vec**

I. Introduction

In the past, research on natural language processing (NLP) has been focusing on tasks such as information retrieval, information extraction, and text summarization [1]-[3]. An emerging of social networks results in many NLP related tasks such as opinion mining, sentimental, fraud detection, etc. While online contents become more popular, there are many attempts to add advertisements or useless elements into the contents, particularly news headline. Towards this issue, one interesting and impact application is to detect fake news, rumor, and clickbait, particularly news headlines have gained attention from researchers in the past 10 years [4]. Fake news is an intentionally and verifiably false news [5]. Rumor, on the other hand, refers to an unofficially confirm information that has been spread. Clickbait refers to article titles or social media posts are designed to attract readers to click on its hyperlink and then lead the actual article page [6]. Moreover, clickbait headlines are the major contributors to spread fake news [7]. The approaches such as Natural Language.

Processing (NLP), Data Mining (DM), and Social Network Analysis (SNA) have been applied for detecting false information. Two approaches are used to represent the news: (1) content-based and (2) context-based features [4]. The first features rely on the content such as news or headlines. On the other hand, surrounding information such as users characteristic, reactions, and network propagation are included in the context-based features.

Behavior-based analysis is an example of context-based features that remove clickbait posts from a social media website. Facebook analyzed a click-to-share ratio and length of time that the users spent on the post [8], [9]. Similarly, Twitter detects the clickbait tweets using the users' behavior together with a naïve Bayes classifier [10]. A disadvantage of this feature is that it could only detect the previously known clickbait headlines. To capture the news headlines,

machine learning approaches targeting NLP problems were introduced [11]-[17]. Content-based features extract from text. In text mining, terms acts as basic elements for constructing feature space [2]. Besides simple terms, some higher orders of terms, so-called *n*-gram, such as bi30 gram, trigram, etc., can be used. Together with the definition of terms, some weighting schemes, such as Term Frequency (TF), and Inverse Document Frequency (IDF), form the foundation of the processing. Based on the features constructed, a number of classifiers can be applied, such as Random Forest [10], naïve Bayes [y], and Support Vector Machine (SVM) [12], [14], [18] to recognize the clickbait headlines.

Recently, neural networks based on dense vector representations have been introduced. Most works utilize the word embeddings, called word2vec [19] that allow an input sentence to be transformed into a matrix and use it as source for any text processing tasks. Later meaningful or significant features are implicitly discovered or extracted from this matrix by means of a Convolutional Neural Network (CNN) enable multi-level feature extraction and automate discovery of significant features. The classification of the input text can be achieved by using deep learning [20]. In the past, based on this concept, Agrawal [21] applies CNN to represent English news headlines as a feature and classify them using deep learning.

As a real-world application, detection of clickbait from the Thai news headlines is useful and promising. The clickbait detection from Thai news headlines has been studied in [22]. Since it is an early-staged study, there is still much room for further investigation and performance improvement. In the past, the traditional approach usually used dictionary-based word segmentation to tokenize a Thai running text into a bag of words, the performance varied upon several factors, such as definition of words in the dictionary and the segmentation algorithm. This variety of word definition and algorithm affects the counting of term frequency and inverse document frequency [23], [24].

As an alternative to the dictionary-based word segmentation, Sarawoot et al. [22] proposed to apply word2vec [19] to encode each segmented headline as a matrix using conditional probability of the target term and its surrounding terms. In this approach, the text is encoded in the form of term vectors that are standardized and preserve the consistency between distance and similarity. The CNN-based term vectors are acted as features for representing a text and the deep learning algorithm is applied to learn the optimal classification model.

In this work, a feature called *Headline2Vec* is retrieved from convolutional neural network (CNN) architecture, originally proposed by Kim [25]. A number of experiments are conducted to investigate the performance of our proposed method, which aims to optimize the hyper-parameters for clickbait classification from Thai news headlines. The experiment compares the classification performance of Headline2Vec features retrieved from CNN and the simple sequential feature, i.e. tf-idf. The experiments are conducted using three classification schemes: the SVM, naïve Bayes, and Multilayer Perceptron on a collected dataset of preprocessed Thai clickbait headlines, which is available online. This paper is organized as follows. Section 2 presents a number of works related to clickbait classification, word2vec feature extraction, and convolutional neural networks. Implementation details of our CNN architecture and our headline2vec method are described in Section 3. In Section 4, experimental settings related to hyperparameter selection are described and the experimental results on comparison of our headline2vec with tf-idf (the baseline) is discussed on four machine learning algorithms; naïve Bayes, decision tree, support vector machine, and convolutional neural network. Error analysis is made in Section 5, to identify the issue of our method. Finally, Section 6 provides the concluding remarks and future works.

II. Related Work

A. Clickbait Classification from News Headlines

The machine learning approach for a clickbait detection was firstly introduced by Potthast et al. [10]. The work used the dataset of 2,992 tweets, including 767 clickbait tweets. To obtain an unbiased choice of publishers, this corpus was created by randomly selecting Twitters from a several social media platform by many content publishers; including Business Insider, the Huffington Post, and BuzzFeed; BuzzFeed. Each tweet was annotated independently by three assessors who rated them being clickbait or not. Judgments were made only based on the tweet's plain text and image but not by clicking on links. With majority vote as ground truth, a total of 767 tweets (26 %) out of 2,992 tweets are considered as clickbait. More details can be found from [10].

The clickbait tweets were classified into three categories: (1) the teaser message, (2) the linked web page, and (3) the meta information. The clickbait detection model was constructed based on 215 features of the tweets such as linked web page, word *n*-gram, and a sender's name. The result showed 76% precision and recall using Random Forest, which outperformed Logistic Regression and naïve Bayes.

Later, Chakraborty et al. [12] collected 7,500 non-clickbait headlines from Wiki-news and 7,500 clickbait from the clickbait websites. The Stand ford Core NLP tools [2] as applied to extract 14 features from the clickbait headlines such as sentence structure, or patterns, clickbait language, *n*-gram, etc. The experimental result shows that SVM achieved

93% accuracy, which outperformed Decision Tree and Random Forest.

Instead of using the set of manually selected features as in [12], [10], Agrawal [21] applied CNN to automatically discover the features of the dataset. The 814 clickbait and 1574 non-clickbait headlines were collected from Reddit, Facebook, and Twitter. The experimental result reported that the CNN model with word embedding from Word2Vec achieves 90.00% accuracy at 85.00% precision and 88.00% recall.

B. Feature Extraction

Shu et al. [26] classified the features for fake news detection into two categories: (1) Content-based and (2) Context-based. The content-based feature relies on the information that can be extracted from text. Text mining techniques such as number of content words or POS [27], [28]. Such features have also been applied with machine learning and deep learning approach for detecting rumors. The context-based considers the 110 surrounding information such as user's information [29].

1) Vectorized Word-based Features: Word2vec

Word embedding is a method that transforms text into a numerical representation. A common fixed-length features of text is bag-of-words. The limitation of bag-of-words is its inherited issue on lacking word orders and phrase structure. Word2vec is a word embedding that considers a continuous bag-of-word (CBOW) and a skip-gram models [29]. Given the context words surrounding by a target word across a window of size c, CBOW computes the conditional probability of a target word. On the other hand, the skip-gram model predicts the surrounding context words of the given target word.

In this way, the words with similar meanings tend to occur in similar context. Thus, these vectors capture the characteristics of the surroundings of the term. The advantage of this transformation is that the similarity between words can be captured [30] and computed using measurement such as cosine similarity. In our preliminary study in [22], a publicly available library called Gensim [31] was applied to compute a Word2Vec of the Thai headlines. Examples of the words in the same context are shown in Table I. The meaning of the words is similar.

A limitation of utilizing Word2Vec for a sentence or headlines classification in widely used classifier is its variable number of dimensions. To cope with this issue, the word embedding must be transformed in such a way that a headline is represented as a vector. The simplest method is to concatenate the word embedding of each word in the sentence [32].

2) Data Collection Strategies

To collect the dataset, two strategies are considered: (1) top-down and (2) bottom-up [33]. The first strategy requires a set of keywords and tag of the rumor or fake news for collection. The second strategy requires to manually evaluate by a human. The major drawback for this method is the requirement of the human resource. On the other hand, the fake news or rumor can be found when they were spread to collect using top-down approach.

Due to the small number of data such as rumors, fake news, or clickbait, there are not many datasets available for fake news, rumors, and clickbait headlines [4]. Moreover, the dataset of clickbait headlines in Thai language was initially be collected in our companion paper [22].

3) Convolution Neural Network

The convolutional neural network (CNN) has been utilized in computer vision application [34], [35].

It allows many filters to be used and automatically select the features that are able to represent the salient points in an image. In addition, the features can be refined using the feedback from the classification layer.

Recently, CNN has been applied to problems in Natural Language Processing (NLP) [25], [36], [37]. CNN enables us to extract salient n-gram features from the input sentence [25] to create an informative latent semantic representation for downstream tasks.

Given a word embedding of a sentence, a few convolutional filters, also called kernels, of different window sizes slide over the entire word embedding matrix to extract a specific pattern of n-gram. A convolution layer maps the feature into a vector by selecting the maximum weight of each window. The outcome is a fixed-length feature vector that keeps the salient n-gram of the whole sentence.

TABLE I
AN EXAMPLE OF WORDS IN THE SAME CONTEXT.

Word	Top 5 similar words
เหลือเชื่อ (incredible)	ทึ่ง (surprise), มหัศจรรย์ (wonderful), น่ากลัว (awful), แปลก (strange), สะพรึง (horrible)
ช็อค (shock)	ช็อก (shock), ขนหัวลุก (thrilling), ผงะ (flinch), หลอน (haunt), ตกใจ (shock)
โซเชี่ยล (social)	โซเชียล (social), เม้นท์ (comment), สนั่น (loudly), คอมเม้นท์ (comment), โซเชียล (social)
แชร์ (share)	ลงโซเชียล (share to social), ไลค์ (like), ชื่นชม (admire), ว่อน (in a cloud), กด (push)
เฟี้ยว (magnificently)	แจ่ม(good), ชิค (chic), จ๊าบ (prominent), อาร์ต (art), อลัง (magnificent)

TABLE II
AN EXAMPLE OF THE NEWS HEADLINE IN THE PREPROCESSING STEP

Collected headline	Output of text cleaner	Preprocessed Headline
ทูตอินเดียเผยทางเชื่อม อินเดีย-เมียนมา-ไทย ใกล้ เสร็จ http://tnn24.tv/2qMf 3n8 #TNN24 #ช่อง16 @tnnthailand **Original Headline** India Ambassador says; the India-Myanmar-Thailand road connectivity is almost done. **Translation**	ทูตอินเดียเผยทางเชื่อม อินเดียเมียนมาไทยใกล้ เสร็จ	ทูตเอินเดียเผยทางเชื่อม อินเดียเมียนมาไทย ใกล้ เสร็จ

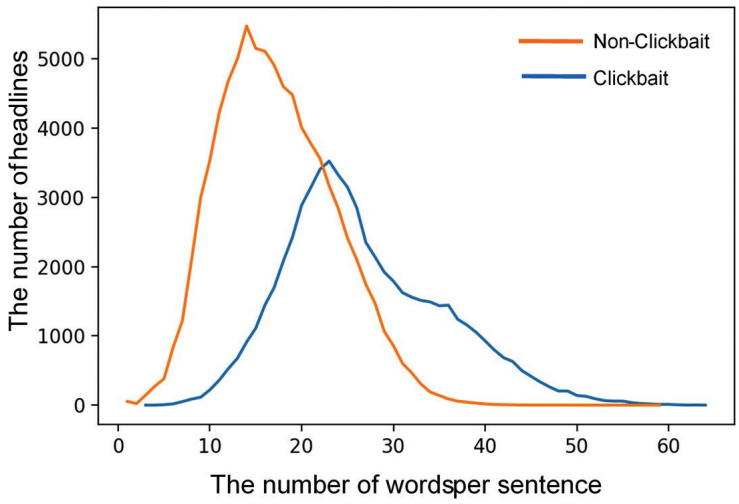

Fig. 1. A comparison of the number of words per sentence of clickbait and non-clickbait headlines.

III. THAI CLICKBAIT HEADLINES CORPUS

A Thai headline news corpus for a natural language processing task has initially been gather in [22]. The headline news is gathered using top-down strategy. An online content crawler is used for retrieving headlines from known clickbait website. Then, text preprocessing is applied to remove any unwanted data from the collected headlines.

The sources of clickbait and non-clickbait headlines are identified by human. For the clickbait websites, the sources are selected from "Jobkaow" Facebook page, which shows the summary of the clickbait news. The sources of the news in this page are considered as a clickbait website. On the other hand, the non-clickbait websites are selected from the Twitter account of the news agency in Thailand. Both clickbait and non-clickbait headlines are collected using an online content crawler.

The collected headlines as shown in the first column of Table II contains unwanted information such as a hyperlink, hashtag, and the source of the tweet. A regular expression called a text cleaner is used during the preprocessing to remove that unwanted information. The outcome of the text cleaner of the headline of the collected headline is shown in the second column of Table II.

To allow the text to be processed, each headline is segmented into a list of words using a software called LextoPlus [38], which is a segmentation for Thai text. An example of the segmented list of words is shown in third column of Table II. The total number of the collected headlines is 132,938, in which 11.96% of words are in out of vocabulary set (OOV). The number of clickbait and non-clickbait headlines are 60,393 and 72,545, respectively. The comparison of the number of words of the collected clickbait and non-clickbait headlines are shown in Fig. 1.

IV. CNN-Based Features for Clickbait Detection

This section describes how to implement clickbait detection from news headlines, by extending the headline2vec concept proposed in our previous work [22]. Two issues; (1) feature extraction and (2) hyperparameter tuning in CNN are described.

A. Feature Extraction

To allow a computer to understand the text, the list of the segmented words of the headlines are transformed into a numerical representation, which is called a *word embedding*. In this work, two features are investigated to represent each segmented headline: 1) a simple sequential feature called tf-idf and 2) CNN feature extracted from Word2vec embedding.

1) Headline to Vector

To extract the patterns of the training set, each headline is presented in a form of a vector. Unlike a numerical data type, a word can be converted to a vector called a word embedding. A model called *Word2Vec* [19] allows a word to be represented as a vector in such a way that the relationship among words that frequently occurred in the same context is maintained. It is possible to transform a word into a vector and then use such vector for classification, such as naïve Bayes, support vector machine, and multi-layer perceptron classifiers. Therefore, the word embedding must be transformed in such a way that a headline is represented as a vector.

Let $h \in H$ be a headline in the form of bag of words in the headline corpus H and m be the length of the longest headline in H. Let $W = \{w_1, w_2, \ldots, w_{|W|}\}$ be the entire vocabulary of words in the headline corpus H. A segmented headline h is embedded on a Word2Vec model of size $m \times d$, where d is the word vector dimension. It is transformed into a feature vector using a Convolutional Neural Network (CNN) [25]. Two main parts are feature extraction and classification. In this work, a Word2Vec of the segmented headline is transformed into a vector feature at the concatenation layer of CNN architecture. See Fig. 2 for illustration. The detail of the feature extraction is described below. A word embedding of a headline h is given in the first layer as shown in Fig. 2. For each headline, let $w_i \in W^d$ represent the word embedding of the i^{th} word in the headline, where d is the dimension of the word embedding. The headline h is represented as an embedding matrix $W \in W^{m \times d}$, where m is the number of words of the longest headline in the corpus. Then, the word embedding is filtered using different window size and filter. It works correspondingly with n-gram. Let $w_{i:i+j}$ be the concatenation of vectors $w_i, w_{i+1}, \ldots w_j$. Convolution is performed on this input embedding layer by applying a filter $k \in W^{nd}$ to a window of size n to produce a feature vector of each window size. Then, a feature c_i is generated using the window of words $w_{i:i+n-1}$ in h of a filter as follow:

$$c_i = f(w_i{:}i{+}n{-}1.k^T + b)$$

where, b is the bias term and f is a non-linear activation function, such as the hyperbolic tangent.

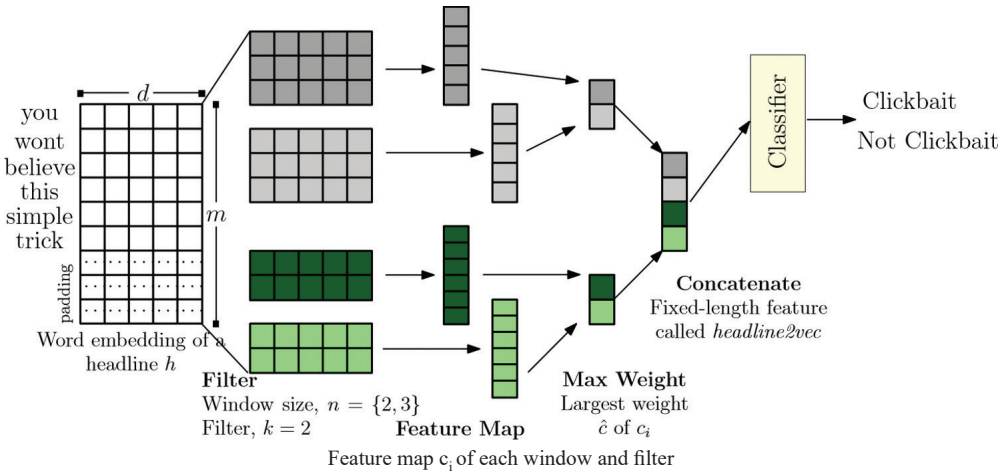

Fig. 1. The headline feature extraction using Convolutional Neural Network (CNN) architecture for clickbait Detection.

The filter k is applied to all possible windows using the same weights to create the feature map as shown in Fig. 2. It can be defined as followed:

$$c = [c_1, c_2, ..., c_{v-n+1}]$$

The convoluted word vector is passed to a maximum-pooling layer to adjust an appropriate weight for representing a vector of each word. A limitation of the feature map is the dimension of c is depended on the length of the headline and the filter k. To cope with the problem, the input is subsampled by applying a max operation on each filter in such a way that $\hat{c} = \max\{c\}$. In this way, the output is mapped to a fixed dimension while keeping the salient n-gram features of the whole headline.

The concatenation of the weight $\hat{c}$ of the word features of the headline h using n windows and k filters results in a vector of size $n \times k$. This vector is a Headline2Vec of an input headline h. It is used for classifying into 'clickbait' or 'not clickbait' using the SoftMax function.

The Headline2Vec is a feature vector of a headline h, whose size is $n \times k$, where n is a window size and k is a filter. See, concatenation layer in Fig. 2. In this way, the machine learning techniques such as naïve Bayes, Support Vector Machine (SVM), and Multilayer Perceptron can utilize the feature vector for a headline classification task.

2) TF-IDF

Term frequency (tf) is the frequency of a word occurs in the document. If a certain word occurs frequently, the term has the great effect on the document. Given a segmented headline h where $h \in H$ of the headline corpus, let $f_{wi,h}$ be the raw count of the word w_i in the headline h, where it is equal 1 if the word w_i occurs in h and 0 otherwise. The term frequency is defined as the ratio of $f_{wi,h}$ to the size of the words in h. Inverse document frequency (idf) measures how often a word occurs in the documents. Let $idf(w_i,H)$ be an inverse document frequency of the word w_i of the corpus H. It can be defined as $idf(w_i,H) = \log \frac{|H|}{|\{h \in H : w_i \in W\}|}$. In this way, tf-idf of the word w_i in a headline h of the corpus H can be defined as $tfidf(w_i,h,H) = fw_{i,h} \times idfw_{i,H}$. The tf-idf feature of h is a vector $tfidf(w_i,h,H)$ of size $|W|$.

A limitation of representing each headline using tf-idf feature is the exclusion of the relationship among the surrounding words, which results in the different vector representation of the wrong segmentation words.

B. Hyperparameters for CNN

To achieve a good classification result, the four hyperparameters: (1) numbers of epoch, (2) word vector dimension, (3) window size (n-gram), and (4) modelling technique are studied.

The epoch is the numbers of times that the entire dataset is passed through a neural network for updating the weights. On the other hand, the small number of epoch may make model underfit and overfit. The word vector dimension is the vector size d that represents each word in a dictionary of millions of words into a small dense vector. It relates to the efficiency of the features because the high number of the dimension could lead to a sparseness and results in the loss of its semantic. The window size for surrounding words is considered as an n-gram. There are three model variations of CNN, i.e., static, non-static, and random. The static is the model that uses the weight in the first layer, i.e. Word2Vec, to represent all word in dataset and kept its static. On the other hand, the non-static technique initially uses the weight from the Word2Vec. Then, it is adjusted during training of the model. The random variation initially assigns a random weight to every word and modified during model training.

V. Experiment

The preprocessed and segmented [38] headlines from Sarawoot et al. [22] are used in the experiment. The data set contains 132,938 headlines. The total number of the headlines in the training and the testing sets are 106,350 and 26,588, respectively. See Table III for the detail.

Two features are extracted from the dataset: (1) Headline2Vec and (2) tf-idf. The features are separated into training and testing set, where the number of each set can be found in Table III. The experiments of the classification performance of each feature are conducted using the separated training and testing set. The detail of each step is explained in the following subsections.

Two experiments were conducted in this work. The first one is to find the value of the hyperparameters that enable the CNN to achieve the best performance. The hyperparameters are (1) numbers of epoch, (2) word vector dimension, (3) window size (n- gram), and (4) modelling technique. The second experiment compares the performance of the feature vectors, i.e., Word2Vec which are extracted from CNN, and tf-idf. The classification methods being used are naïve Bayes (NB), support vector machine (SVM), multilayer perceptron (MLP), and deep learning. The machine learning libraries of Python are used for NB, SVM and MLP. The CNN feature extraction and classification implementation of Kim [25] is applied.

The performance of the classification is evaluated using four measures precision, recall, F1-score, and accuracy. Precision is the percentage of the correctly predicted headlines for a clickbait by the size of the test set. Recall is the percentage of correctly predicted headlines for a clickbait divided by the total number of clickbait headlines over the test set. F-measure is the weight of precision and recall. Lastly, accuracy is the percentage of the retrieved headlines that are

correctly in the same class as the query by the number of headlines in test set.

Hyperparameters Tuning of CNN

To study the effects of the hyperparameters of CNN, a number of experiments were conducted. Four hyperparameters: (1) Modelling technique (2) *n*-gram (3) Epoch, and (4) Dimension of Word2Vec are being focused.

For the modeling technique, we applied *static*, *non-static*, and *random* weight adjustment techniques. The experimental result in Table IV shows that the non-static modeling technique achieves the highest accuracy of 95.28%.

To find an appropriate window size for the word embedding, the numbers of *n*-gram are {2,3,4,5,6,7} are applied. The experimental result can be found in Table V. The result shows that the most appropriate window size is n = {2,3,4}, which gives 95.284% accuracy. Combining the three modelling techniques and the different window size, the result can be found in Fig. 3. The graph shows that the *non-static* modeling technique 305 with the window size n = {2,3,4,5,6} gives the highest accuracy of 94.37%.

The number of epochs equals to {1,5,10} are used in the experiment. The result in Fig. 4 shows that 5 passes gives the highest accuracy of 95.28%. The comparison of the accuracy of the combination of the three modeling techniques and the different number of epoch can be found in Fig. 4. From the figure, the *non-static* model 310 and epoch equals to 4 achieves the highest accuracy of 94.82%.

Lastly, the dimensions of Word2Vec are set as {25,50,100}. The comparison of the result in Table VI shows that using 25 dimensions gives the highest accuracy of 95.28%. Applying the non-static modeling technique using 50 dimension gives the highest accuracy of 93.36%. Combining the parameters that individually achieves the best result, i.e. *n*-gram equals {2,3,4}, 50 dimensions of word2vec, epoch equals to 5, and non-static model, it achieves 94.373%. The comparison to the other models can be found in Table VII.

TABLE III
THE NUMBER OF COLLECTED HEADLINES.

	Headlines	**Training Set**	**Testing Set**
Clickbait	60,393	48,314	12,079
Non-Clickbait	72,545	58,036	14,509
Total	132,938	106,350	26,588

TABLE IV
RESULT OF APPLYING THE THREE DIFFERENT MODELING TECHNIQUE

Setting	**Precision**	**Recall**	**F1**	**Accuracy**
Static	95.11%	94.76%	94.93%	94.46%
Non-Static	95.15%	95.34%	95.24%	**95.28%**
Random	93.28%	92.74%	93.01%	92.80%

TABLE V
RESULT OF APPLYING THE THREE DIFFERENT MODELING TECHNIQUE

***n*-gram**	**Precision**	**Recall**	**F1**	**Accuracy**
2	94.85%	94.79%	94.82%	94.86%
2,3	95.06%	95.16%	95.11%	95.15%
2,3,4	95.15%	95.34%	95.24%	**95.28%**
2,3,4,5	94.58%	95.08%	94.83%	94.84%
2,3,5	95.25%	95.22%	95.23%	95.27%
2,3,5,7	94.27%	94.98%	94.62%	94.61%

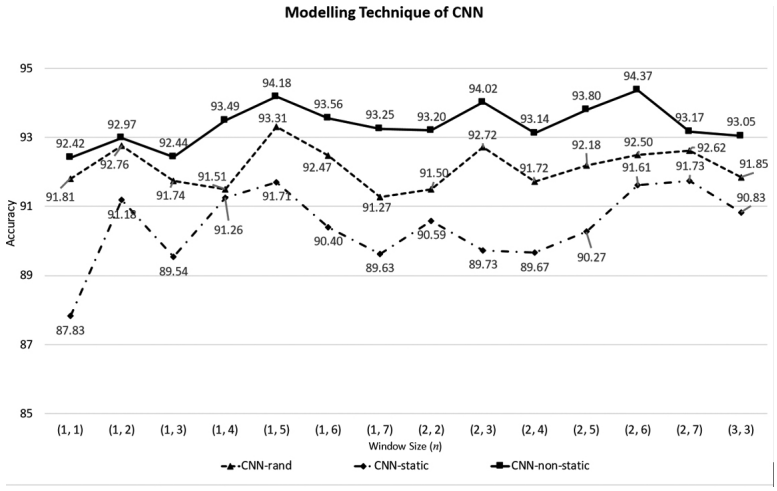

Fig. 3. Experiment result of the three modelling techniques using the different number of window size.

TABLE VI
RESULT OF THE DIFFERENT WORD2VEC DIMENSION.

Dimension	Precision	Recall	F1	Accuracy
25	95.45%	95.09%	95.27%	94.90%
50	95.15%	95.34%	95.24%	**95.28%**
100	95.00%	95.13%	94.07%	95.11%

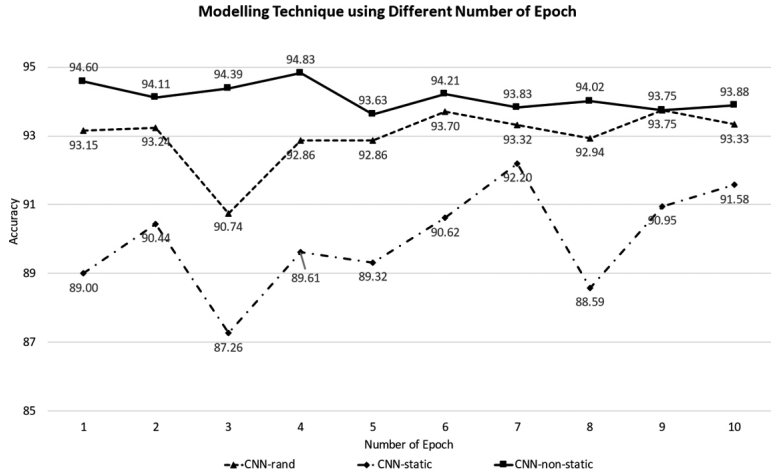

Fig. 4: Experiment result of the three modeling techniques using the different number of epoch.

TABLE VII
RESULT WHEN APPLYING THE DIFFERENT MODELLING TECHNIQUES ON N-GRAM EQUALS {2, 3, 4, 5, 6}, 50 DIMENSIONS OF WORD2VEC, AND 5 EPOCH.

Modelling Technique	Precision	Recall	F1	Accuracy
Non-static	93.526%	94.055%	93.79%	**94.37%**
Static	87.532%	93.591%	90.46%	91.61%
Random	94.644%	89.466%	91.982%	92.50%

1) TF-IDF Features

The classification results of tf-idf feature are reported in Table VIII. The accuracy of NB, SVM, and MLP are 89.69%, 92.05%, 90.88%, respectively. Comparing to the accuracy of the most appropriate hyperparameter setting of CNN, it achieves higher accuracy of 94.37%. Fig. 5 shows the comparison to the CNN and tf-idf features using different window sizes. The CNN feature achieves the highest accuracy for every window size.

2) Headline to Vector

The Headline2Vec of every headline is extracted using the hyperparameters of the highest accuracy as reported in Table VII. The headlines are embedded on a 50 dimension of word2vec. The non-static modelling technique with *n*-gram of {2,3,4,5,6} and 5 epoch of CNN framework is used to extract the headline2vec feature.

The comparison of the classifiers can be found in Table IX. Among the classifiers, Multilayer Perceptron achieves the highest result of 93.89% which outperform SVM and naïve Bayes. Comparing the result using tf-idf feature in Table VIII, the Headline2Vec achieves higher accuracy for every classifier. However, the accuracy of Multilayer Perceptron classifier using Headline 2Vec in Table 9 is still lower than the classification of the CNN architecture, which achieves 94.37%.

3) Sentiment Dataset

The proposed framework for extracting headline2vec is applied with a dataset for a sentiment analysis from the comment of the restaurants in Thailand. The dataset can be found on the web site. The experimental result is shown in Table X. The classification using MLP achieve 98.49%, which is lower than SVM.

4) BERT Feature Extraction

Bidirectional Encoder Representations from Transformers (BERT) is a new language representation model [39]. In this experiment, BERT is applied for extracting features from the headlines. The same classifier as shown in Fig. 2 is used to predict the result. In this work, an implementation from [40] was utilized. The experimental result shows 58.34% accuracy.

VI. Analysis

The experimental results in Table VIII and Table IX show that the Headline2Vec feature extracted from CNN achieves a higher accuracy than tf-idf for every classifier. The reason is that the Word2Vec can capture the similarity between words and CNN preserves *n*-gram feature of the whole headline. With the characteristic of Word2Vec and CNN, it allows the proposed Headline2Vec to classify the text even if the keywords are not existed in the headline. Comparing to the classification result using the Headline2Vec feature and CNN architecture [25], the Headline2Vec using Multilayer Perceptron achieves lower accuracy. However, the different is 0.48%. One reason is because the CNN architecture allows the weight to be adjusted to achieve the best accuracy of the headline classification in the last layer.

We also applied BERT [39] model to extract feature from the headline. The result shows that BERT achieves lower accuracy than the proposed method. The reason is the segmentation outcome of BERT is finer than the word level. Since the length of the headline is not long as compared to the text in the paragraph, the extracted feature becomes sparse.

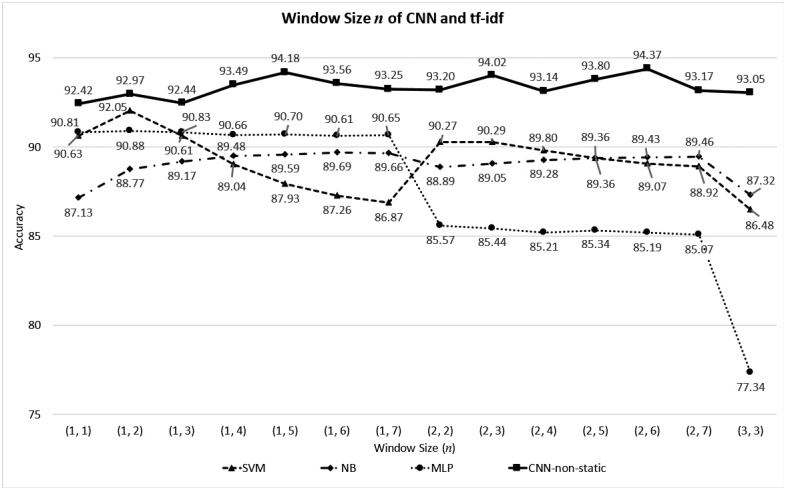

Fig 5. Experiment result of the CNN and tf-idf features using the different number of window size.

TABLE VIII
RECALL, PRECISION, F1 SCORE, AND ACCURACY USING TF-IDF FEATURE ON THAI CLICKBAIT DATASET.

Classifier	Precision	Recall	F1	Accuracy
Naive Bayes	86.78%	91.21%	88.94%	89.69%
Support Vector Machine	89.37%	93.63%	91.45%	**92.05**%
Multilayer Perceptron	87.76%	92.88%	90.25%	90.88%

TABLE IX
RECALL, PRECISION, F1, AND ACCURACY USING HEADLINE2VEC THAI CLICKBAIT DATASET.

Classifier	Precision	Recall	F1	Accuracy
Naive Bayes	98.77%	71.81%	86.14%	86.78%
Support Vector Machine	92.70%	93.47%	93.64%	93.69%
Multilayer Perceptron	94.32%	92.11%	93.83%	93.89%

TABLE X
RESULT OF THE SENTIMENT ANALYSIS DATASET.

Classifier	Precision	Recall	F1	Accuracy
Naive Bayes	49.03%	100.00%	51.27%	55.60%
Support Vector Machine	100.00%	99.37%	99.72%	99.73%
Multilayer Perceptron	97.34%	99.16%	98.46%	98.49%

VII. Conclusion

A comparison of the traditional sequential feature and Headline2Vec extracted from the two-dimensional word embedding for Thai clickbait classification using CNN is presented. The two-dimensional word embedding allows a widely used automatic feature extraction architecture called CNN to be used as an input. However, it cannot be used with the machine learning techniques such as SVM, naïve Bayes, and Multilayer Perceptron. In this way, a feature vector called Headline2Vec is retrieved from the last layer of the feature extraction of the CNN architecture. The Headline2Vec allows a learning techniques to utilize a high dimension Word2Vec embedding, which is a word embedding that includes the relationship among the surrounding words.

A number of experiments were conducted to find the most suitable hypermeters settings for the CNN configuration. Moreover, the experiments of applying the Headline2Vec feature and the traditional tf-idf feature are conducted with SVM, naïve Bayes, and Multilayer Perceptron. Comparing to the traditional feature, the Headline2Vec achieves higher accuracy for every classifier. It can be concluded that relationship among words of Word2Vec improve the accuracy for classifying the clickbait headline for Thai.

VIII. Acknowledgement

This work partially supported by Royal Society of Thailand under the contract number 36/2561 and 226/2461. Moreover, the first author gratefully acknowledges the financial support provided by Thammasat University Research Fund under the TU Research Scholar Contract No. 1/27/2561 and SIIT Young Researcher Grant, under contract no. SIIT 2017-YRG-NK04. In addition, this work is also financially supported by Thailand Research Fund under grant number RTA6080013. We would like to thank Prof. Dr. Manabu Okumura and Dr. Anocha Rugchatjaroen for the supports and valuable comments. We would like to also thanks Natnicha Wongsap, Lisha Lou, Sasiwimol Jumun, and Tastanya Prapphan for the data preparationa.

References

[1] C. Erik and B. White, "Jumping NLP curves: A review of natural language processing research," *IEEE Computational intelligence magazine*, Vol. 9, no. 2, pp. 48-57. 2014.

[2] C. D. Manning, M. Surdeanu, J. Bauer et al., "The Stanford corenlp natural language processing toolkit," in *Proc. The 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55-60.

[3] F. Provost and T. Fawcett, *Data science for business what you need to know about data mining and data-analytic thinking. Boston,* USA: O'Reilly Media, 2013, pp. 1-409.

[4] A. Bondielli and F. Marcelloni, (2020, April 10). A survey on fake news and rumour detection techniques, *Information Sciences*. [Online]. *497*, pp. 38-55. Available: https://doi.org/10.1016/j.ins.2019.05.035

[5] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Technical Report, National Bureau of Economic Research,* vol. 31, no. 2, pp. 211-236, Spring. 2017.

[6] Y. Chen, N. J. Conroy, and V. L. Rubin, "Misleading online content: recognizing clickbait's false news," in *Proc. ACM Workshop on Multimodal Deception Detection*, 2015, pp. 15-19.

[7] C. Silverman, "Lies, damn lies, and viral content. How news websites spread (and debunk) online rumors, unverified claims, and misinformation," *Tow Center for Digital Journalism*, vol. 168, pp. 1-155, Feb. 2015.

[8] K. El-Arini and J. Tang. (2020, April 10). *Click-Baiting: Facebook Newsroom*. [Online]. Available https://newsroom.fb.com/news/2014/08 /news-feed-fyi-click-baiting

[9] A. Peysakhovich and K. Hendrix. (2020, April 10). *News Feed FYI: Further Reducing Clickbait in Feed, In Facebook newsroom*. [Online]. available http://newsroom.fb.com/news/2016/08/news-feed-fyi-further-reducing-click bait-in-feed/

[10] M. Potthast, S. Kopsel, B. Stein, and M. Hagen, "Clickbait Detection, in Proc," in *Proc. The 38th European Conference on Machine Learning,* 2016, pp. 810-817.

[11] A. Anand and T. Chakraborty, and N. Park, "We used neural networks to detect clickbait's: you won't believe what happened next," in *Proc. European Conference on Information Retrieval*, 2017, pp. 541-547.

[12] A. Chakraborty, B. Paranjape, S. Kakarla, and N. Ganguly, "Stop clickbait: Detecting and preventing clickbait's in online news media," in *Proc. 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2016, pp. 9-16.

[13] J. Han, M. Kamber, and J. Pei, *Classification: Basic Concepts*, Massachusetts, Massachusetts. USA: The Morgan Kaufmann, 2012, pp 327-391.

[14] T. Joachims, "Text Categorization with Support Vector Machines:Learning with Many Relevant Features," in *Proc. The 10th European Conference on Machine Learning*, 1998, pp. 137-142.

[15] D. Pandey, G. Verma, and S. Nagpal, *Clickbait Detection Using Swarm Intelligence*, Singapore, SG: Springer, 2019, pp. 64-76.

[16] M. Potthast, T. Gollub, M. Hagen, and B. Stein. (2020, Sep 10). *The Clickbait Challenge 2017: Towards a Regression Model for Clickbait Strength*. [Online]. Available: https://arxiv.org/abs/1812.10847

[17] W. Wei and X. Wan, "Learning to identify ambiguous and misleading news headline," in *Proc. The 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 4172-4178.

[18] B. D. Horne and S. Adali. (2020, Mar 20). *This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to a tire than real news.* [Online]. Available: https://arxiv.org/abs/1703 .09398]

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. The International Conference on Learning Representations*, 2013, pp. 1-15.

[20] S. Richard, P. Alex, W. Jean, et.al., "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in *Proc. The 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1631-1642.

[21] A. Agrawal, "Clickbait detection using deep learning," in *Proc. 2016 2nd International Conference on Next Generation Computing Technologies*, 2016, pp. 268-272.

[22] S. Kongyoung, A. Rugchatjaroen, and N. Kaothanthong, *Automatic Feature extraction and Classification model for detecting Thai clickbait headline using convolutional Neural Network.* Amsterdam, Netherland:IOS Press, 1991, pp. 184-194.
[23] K. Kosawat, "BEST 2009: Thai Word Segmentation Software Contest," in *Proc. The 8th International Symposium on Natural Language Processing,* 2009, pp. 83-89.
[24] T. Suwanapong, T. Theeramunkong, and E. Nantajeewarawat, "Name-alias relationship identification in Thai news articles: A comparison of co-occurrence matrix construction methods," *Chiang Mai Journal of Science*, vol. 44, no. 4, pp. 1805-1821, 2017.
[25] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proc. The 2014 Conference on Empirical Methods in Natural Language Processing,* 2014, pp. 1746-1751.
[26] K. Shu, A. Sliva, S. Wang et al., "Fake news detection on social media: a data mining perspective," *ACM SIGKDD Explor. Newslett*, vol. 19, no. 1, pp. 22-36. Sep. 2017.
[27] Y. Qin, D. Wurzer, V. Lavrenko, and C. Tang. (2020, April 10). *Spotting rumors via novelty detection*. [Online]. Available: https://www.semanticscholar.org/paper/Spotting-Rumors-via-Novelty-Detection-QinWurzer/739d05c6ed0fdb92226924c5cb9866a5c7c9a50
[28] A. Zubiaga, M. Liakata, and R. Procter. (2020, April 20). *Learning reporwting dynamics during breaking news for rumour detection in social media. Researchgate.* [Online]. Available: https://www.researchgate.net/publication/309402969_Learning_Reporting_Dynamics_during_Breaking_News_for_Rumour_Detection_in_Social_Media
[29] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proc. 2013 International Conference on Social Computing*, 2013, pp. 675-684.
[30] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment Analysis Is a Big Suitcase," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 74-80. Dec. 2017.
[31] R. Rehurek, and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. The 7th International Conference on Language Resources and Evaluation*, 2010, pp. 46-50.
[32] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. The 31st International Conference on International Conferenceon Machine Learning*, 2014, pp. 1188-1196.
[33] A. Zubiaga, A. Aker, B. Bontcheva et al., "Detection and resolution of rumours in social media: a survey," *ACM Comput. Surv*, vol. 5, no. 2, pp. 1-36, Apr. 2018.
[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. The 25th International Conference on Neural Information Processing Systems*, 2012, pp. 1097-1105.
[35] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806-813.
[36] C. N. Dos Santos and M. Gatti, "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts." in *Proc. The 25th International Conference on Computational Linguistics*, 2014, pp. 69-78.
[37] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. "A Convolutional Neural Network for Modelling Sentences," in *Proc. The 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 655-665.
[38] C. Haruechaiyasak and A. Kongthon, "LexToPlus: A Thai Lexeme Tokenization and Normalization Tool", in *Proc. The 4th Workshop on South and Southeast Asia Natural Language Processing*, 2013, pp. 9-16.
[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Computer Science - arXiv*, vol. 2, pp. 1-16, May. 2019.
[40] T. Wolf, L. Debut, V. Sanh et al. (2020, April 20). *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. [online]. Available: https://www.semanticsecholar

Natsuda Kaothanthong received a Ph.D. in Information Science from Graduate School of Information Sciences, Tohoku University. She is an Assistant Professor in School of Management Technology at Sirindhorn International Institute of Technology, Thammasat University. Her research interests are machine learning, artificial intelligence, image processing, and medical images processing.

Sarawoot Kongyoung was a research assistant in Human Computer Communication Research Unit at National Electronics and Computer Technology Center, Thailand. He received a master degree in engineering from Thailand Advanced Institute of Science and Technology and Tokyo Institute of Technology. He is now a Ph.D. candidate at School of Computing Science, University of Glasgow.

Thanaruk Theeramunk-ong received a Ph.D. in Computer Science from Tokyo Institute of Technology. He is a professor in School of Computer and Communication Technology at Sirindhorn International Institute of Technology, Thammasat University. His research interests are natural language processing, artificial intelligence, knowledge data discovery, information retrieval, data mining, machine learning, and intelligent information systems.