Vietnamese Keyword Extraction Using Hybrid Deep Learning Methods

Bui Thanh Hung

Data Analytics & Artificial Intelligence Laboratory
Engineering - Technology Faculty
Thu Dau Mot University
6 Tran Van On street, Phu Hoa district, Thu Dau Mot city, Binh Duong province
hungbt.cntt@tdmu.edu.vn

Abstract – **Keywords provide a short way of reflecting a main idea of the document, making it easier for the readers in reading. Extracting keyword is the main task in natural language processing. Since it is not only time consuming but also requires lots of efforts to extract the keywords manually, it arises the need for the automated approaches. This paper has proposed a solution for the automatic keyword extraction in Vietnamese language using hybrid deep learning approaches. Every existing deep learning approach has its own advantages; and the hybrid deep learning model we are introducing is the combination of the superior features of CNN and LSTM models. The proposed model shows enhanced accuracy and f1-score over another approach.**

Keyword- **keyword extraction, CNN, LSTM, hybrid deep learning.**

I. INTRODUCTION

Nowadays, extracting keywords from documents plays a significant role. With a huge amount of information booming and exponentially increasing on the Internet, it is nearly impossible for a human to perform this task manually. A variety of practical problems could be solved now by identifying keywords from the documents such as: Searching for information, summarizing text, mining text, browsing website,... Many people need to synthesize and summarize the information to facilitate the synthesis of such information for after. The automatic keyword extraction approach plays an important role in many core natural language processing tasks.

In particular, keywords allow people to search faster, easier and more effectively. Through keyword analysis, researchers could grab the key information from a large dataset quickly.

Upon more in-depth investigation on the area, we found out that automatic keyword extraction on all domains and topics may take lots of time just for data entry and training. To come up with the best result, we only focus on the topic of health care.

Keyword Extraction can be called as a binary classification problem [1]. In the exploration of artificial intelligence, there are many methods based on the machine learning techniques for example Naïve Bayes, Decision tree, SVM algorithm [2] [3] etc. has achieved remarkably positive results. Especially in recent years, Deep Learning approaches have proved significant benefits for the task of natural language processing [4]; other researches applied Deep Learning algorithms to extract the keywords could be listed here including CNN, RNN, LSTM, [7][8][9]…and they have achieved outstanding results; and proved to be practical to apply the real life.

In this paper, we present a hybrid deep learning approach by combining the CNN and LSTM deep learning method. Firstly, we build up 5-dimensional matrix features for the hybrid deep learning model from five concepts: Word Embedding, Named-Entity Recognition, Frequency-based, Position Embedding, Phrase Length and Word Length. Text vector is built by the CNN for the given texts being predicted based on the above feature vectors. The useful affective information in different regions can be extracted and weighted according to their contribution to the keyword prediction through a convolutional layer and max pooling layer of CNN model. This feature is integrated across regions using LSTM for keyword prediction.

The rest of this paper is organized as follows. Section II presents related work. In section III, the proposed keyword extraction by the hybrid deep learning approach has been discussed. Section IV presents the experiment. Moreover, the last V presents the conclusion.

II. RELATED WORK

Various approaches to keyphrase extraction have been explored in the past, which can be divided into unsupervised methods and supervised methods.

Unsupervised methods for keyword extraction can be categorized into ranking and clustering candidate keywords to a scoring function.

In supervised methods, it focuses on two issues: task reformulation and feature design. In task reformulation, supervised approaches were considered as a binary classification problem [1]. Approaches using different learning algorithms such as Naïve Bayes [2], Decision Tree [3],… with a determined keyword to train a classifier which predicts whether a noun phrase is a

978-1-5386-7983-8/18/$31.00 ©2018 IEEE

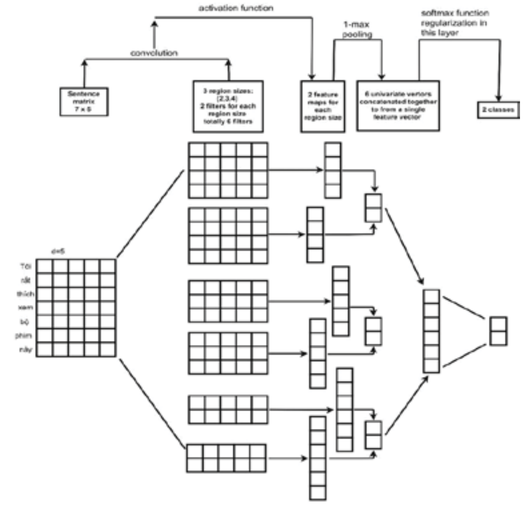
activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 x 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to from a single feature vector
2 classes
d=5
Tôi
rất
thích
xem
bộ
phim
này

Fig 1. CNN model

keyword or not. Later, Jiang et al. [5] proposed a ranking approach to the key phrase extraction.

Recently, word embedding [6] and deep neural networks (NN) such as convolutional neural networks (CNN) [7], recurrent neural networks (RNN) [8] and long short term memory (LSTM) [9][10] have been successfully employed for natural language processing.

Zhang et al. proposed a joint-layer recurrent neural network model to extract key phrases from tweets, which is another application of deep neural networks in the context of key phrase extraction [11].

Rui Meng et al proposed an RNN-based generative model for predicting key phrases in scientific text. This is an encoder-decoder model for key phrase prediction task [12].

In this research we combine the advantages of CNN and LSTM with supervised classifications. The CNN will extract useful affective information from 5-dimensional matrix features concatenated by real-valued vector from word embedding, Named-Entity Recognition, Frequency-based, Position embedding and Phrase Length and Word Length. These features are integrated across regions using LSTM for keyword prediction. By combining those advantages of CNN and LSTM, the hybrid deep learning model proves to be much more effectively.

III. PROPOSED METHOD

Different with task-specific algorithms, deep learning is a member of wider family of machine learning methods which is based on learning data representations, . Learning could be classified into three categories including supervised, semi-supervised or unsupervised.

There are two most well-known types of deep neural networks which are convolutional neural networks (CNNs) and Recurrent Neural Networks (RNNs).

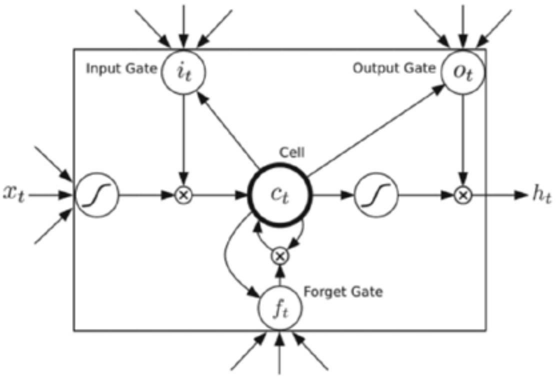

Fig. 2: The Long Short Term Memory cell

In this section, we will first describe the convolutional neural networks (CNN) and Long short term memory (LSTM). Then, we will discuss the hybrid deep learning approach based on combining CNN and LSTM with supervised classifications.

A. CNN

CNN includes many major architectural components which are paired with each other in multi-story structure that is: Convolution, Pooling, Activation function (ReLu, Sigmoid, Tanh), and Fully connected. This model is shown in Fig 1.

The above diagram has an input matrix for the clause "Tôi rất thích xem bộ phim này" ("I like watching this movie very much") It will be a matrix with a width of 5 and a length of 7 lines, followed by a description of which will have areas for stools. The size and dimensions will be width-7, customized height (2,3,4) and we will apply 2 filters for each region, after the first convolution we will have 6 output, followed by 1-max-pooling, In order to find the highest specification for each region, the output of layer 2 will be 3, and we will convert these three single vectors into the same vector for the next analysis function (also the next input layer). The final softmax layer to output the last two classes output.

B. LSTM

Generally, a recurrent neural network is a type of advanced artificial neural network. RNNs can use their internal state (memory) to process sequences of inputs. RNNs have shown great successes in many NLP tasks. They connection previous information to present task, such as predict the next word in the sentence "Mây nằm trên *trời*" ("clouds on the sky"). The gap between the relevant information and the place that it needs is small. So, RNNs can learn to use the past information.

However, in terms of Long-Term dependencies, RNNs has not yet worked well. For example, if there is a sentence like "Tôi cảm thấy đau bụng và nhức đầu, có lẽ tôi phải đi bệnh viện để khám bệnh" and want to predict the last word "khám bệnh", RNNs seem to fail to connect the information since there is a big gap between the relevant information and the words we expect it.

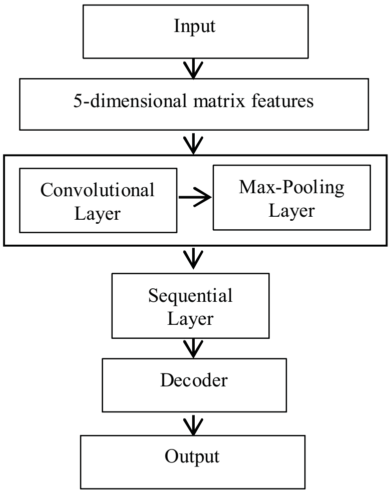

Fig. 3: The proposed model

Long Short Term Memory [13] is a modification of the Recurrent Neural Networks (RNN). The main highlight of the LSTM when compared with the regular feed forward neural network is that they are able to retain the knowledge about previous outputs. The retention part is due to the feedback loop present in their architecture.

Fig. 2 shows a diagram of a simple LSTM cell. Individual cells are combined together to form a large network thereby befitting the term deep neural networks. The cell unit represents the memory. This cell is composed of five main elements: an input gate *i*, a forget gate *f*, an output gate *o*, a recurring cell state *c* and hidden state output *h*.

C. Hybrid deep learning approaches

CNN is capable of extracting local information but may fail to capture long-distance dependency. LSTM can address this limitation by sequentially modeling. In this research we combine advantage features of CNN and LSTM with supervised classifications. When text vector is input of CNN, it will extract features. These features are integrated across regions using LSTM for keyword prediction. By combining advantage features of CNN and LSTM, the hybrid deep learning model does more effectively. The proposed model is shown in Figure 3.

The proposed model consists of five primary components: 5-dimensional matrix features, Convolution layer, Max-pooling Layer, Sequential Layer and Decoder.

For each one 5-dimensional matrix concatenated by real-valued vector from word embedding, Named-Entity Recognition, Frequency-based, Position embedding and Phrase Length and Word Length, CNN model extract useful affective features through a convolutional layer and max pooling layer.

Such local features are then sequentially integrated using LSTM to build a text vector for keyword prediction. We will discuss more detail each components as follows.

5-dimensional matrix features

We extract 5-dimensional matrix features for candidate keywords from five concepts:

- Named-Entity Recognition
- Frequency-based
- Position embedding
- Phrase Length and Word Length
- Word Embedding

Named-entity recognition

We realized that a noun phrase containing a named-entity that is considered a keyword is higher than others. We combine both Number of words in Noun Phrase (nNP) and Number of named-entity words in Noun Phrase (nNER) in single value and the following formula to have a feature value (F_{ner}) as described in the following equation.

$$F_{ner} = \frac{nNER}{nNP} \quad (1)$$

Frequency-based

If a noun phrase is occurring more frequently in a document, the phrase is assumed to be more important than the others in the document.

Two features Phrase Frequency (PF- Number of times a noun phrase occurs independently in a document) and Phrase Link Count (PLC- Number of times a noun phrase appears in full as a part of other noun phrases) are combined to have a single feature value using the following measure:

$$F_{freq} = \sqrt{\frac{1}{2} * PF * PF + PLC} \quad (2)$$

Inverse document frequency (IDF) is a useful measure to determine the commonness of a term in a corpus.

$$IDF = \log\frac{N}{df} \quad (3)$$

N = total number of documents in a corpus.

df (document frequency) = the number of documents in which a term occurs.

We combine F_{freq} and IDF in the following formula to have a variant of the Edmundsonian thematic feature:

$$F_{thematic} = F_{freq} * IDF \quad (4)$$

The value of this feature is normalized by dividing the value by the maximum the $f_{thematic}$ score in a collection of $F_{thematic}$ scores obtained by the phrases corresponding to a document.

Position embedding

If a noun phrase occurs in the title or abstract of a document, it should be given more scores. So, we consider the position of the first occurrence of a noun phrase in a document as a feature. We used the following formula.

$$F_{pos} = \frac{1}{\sqrt{i}} \quad (5)$$

With this feature, it is much easier for us to predict a keyword in the title of a document where F_{pos} is near to 1 and in summary of a document where F_{pos} is near to 0

Phrase Length and Word Length

These two features can be considered as the structural features of a noun phrase. Noun phrase length becomes an important feature in the keyword extraction task because the length of the noun phrase usually varies from 1 word to 5 words. We found out that noun phrase consisting of 6 or more words are relatively rare in our corpus.

Length of the words in a noun phrase can be considered as a feature. We realized that a maximum length of single words in Vietnamese is 7 and 80% words in Vietnamese is a compound word, that is not the same as English. We decided to choose the length of characters in the noun phrase as a feature.

- Length of a noun phrase: PL
- Length of characters in the noun phrase: WL

We combined both single value and the following formula to create a feature.

$$F_{PL*WL} = \sqrt{\log(1+PL) * \ \log(1+WL)} \qquad (6)$$

Word Embedding

Every word has reflected the structure of the word regarding the semantical/morphological/context/ hierarchical/ etc. information. The idea of Word Embedding is to capture with them as much as possible and convert it to vectors. We applied Word Embedding to represent a keyword – can be combined by 2 or more words – as a vector by plus vectors to each word in the dictionary we build.

We concatenate real-valued vector from word embedding, Named-Entity Recognition, Frequency-based, Position embedding and Phrase Length and Word Length into one 5-dimensional matrix. We then fed this input matrix into convolutional layer.

Convolutional Layer

In this layer, vector matrix will be convoluted by filter with some window size k. This filter is composed by weight vectors and bias, and slides over the input matrix. After operating non-linear function, we obtained some valuable features with lower dimension in feature map.

Max-pooling Layer

The pooling units perform max operation to capture the most important features (max value) within each feature map. These features are fed to a sequential layer.

Sequential Layer

To capture long-distance dependency, the sequential layer sequentially integrates each region vector into a vector. After the LSTM memory cell sequentially traverses through all regions, the last hidden state of the sequential layer is regarded as the text representation for keyword prediction.

Decoder

A linear activation function (also known as a linear decoder) is used in the output layer. With the text vector learned from the sequential layer, the linear decoder is used in the output layer, defined as

$$y=(W_d x_t + b_d) \qquad (7)$$

where x_t is the text vector learned from the sequential layer, y is the degree of keyword of the target text, and W_d and b_d respectively denote the weight and bias associated with the linear decoder.

Name	Zing	Thanhnien	Suckhoedoisong	Dantri
Number	100	174	100	150

Tabel 1: The amount of collected data

Data type	**Size**	**Number of documents**
Test	10%	52
Training	80%	420
Validate	10%	52

Table 2: Overview of the test, training and validate set used in our experiments.

Type of Data	**Training and validating**	**Testing**
Document	472	52
Number of most candidate keywords	229	173
Number of less candidate keywords	13	25
Most words in document	1133	1473
Fewer words in document	135	65
Average number of words	419.9	444.8

Table 3: Data analysis

The hybrid deep learning model is trained by minimizing the mean squared error between the predicted keyword and actual keyword. Given a training set of text matrix X = $\{x^1, x^1, x^2, \ldots, x^m\}$, and their keyword ratings set y = $\{y^1, y^2, \ldots, y^m\}$, the loss function is defined as

$$L(X,y) \ = \frac{1}{2n}\sum_{k=1}^{n} \binom{n}{k} \left\| h(x^i) - y^i \right\|^2 \qquad (8)$$

To learn model parameters in the training phase we use back propagation (BP) algorithm with stochastic gradient descent (SGD).

IV. EXPERIMENTS

A. Data Construction

It is extremely challenging to prepare the raw data especially for Vietnamese. In this research, data will be collected from Vietnamese newspapers using Selenium, and we extract a keyword from every newspaper by human. The extracted keywords are revised by lectures of Linguistics Department, Ho Chi Minh University of Social Sciences and Humanities.

The data collected is focused on the topic of "health" and picked up from 4 websites with total files of 524 as shown in Table 1.

With raw data, we have handled punctuation and tokenize data to have standard data sets.

Before training, we divided the collection into three parts: training, validation and test set to avoid overestimating the performance of learned combinations. The data is shown in Table 2.

In the experiment process, we analyzed our training and validated dataset and test dataset as shown in Table 3.

Candidate Keyword Identification

A set of noun phrases and words from a document text is typically extracted as candidate keywords using heuristic rule. To select candidate noun phrases for a given document, we take the full-text content of the document, preprocessed as described above and apply our PoS Pattern (Part of Speech pattern):

(<N.*>+ <A>* <E>)? <N.*>+ <A>*

<N.*>+: One or more of any type nouns

<A>*: Any number of adjectives

<E>: One or zero of preposition

Therefore, this pattern meaning is a noun phrase, which can be combined with two other noun phrases with a preposition word in between.

Using the above grammar and chunking, we have created a result tree, from which we can extract a noun phrase. Calling a noun phrase we need is candidate keyword. In the experimental process, we found that a noun phrase consisting of 6 or more words are rarely in our corpus. So we filter out all noun phrases that contain more than 6 words in the extracting process.

We collected data by Selenium (3.12.0), used chunking of NLTK toolkits (3.3), Vietnamese Tokenize by Pyvi (0.0.0.9 - Tran Viet Trung 2016), Vietnamese Named-Entity Recognition by underthesea (1.1.8), Word2vector pre-train by streetcodevn (Hung Le 2018).

B. Experiment Configurations

After preparation of the training dataset, a hybrid deep learning model is trained on the training set to classify the noun phrase as one of two categories: “Positive” or “Negative”. Positive category indicates that a noun phrase is a keyword and the negative category indicates that it is not a keyword.

In this research, input is a feature’s vector of candidate keyword. Under the influence of the sigmoid activation, the output will be in range 0 to 1. The output determine which candidate keyword is a keyword (1) or not (0).

Because of our group's data, the difference between the keyword and non-keyword terms is huge. That is the main reason for the imbalance problems. We decided to use an over-sampling approach SMOTE [14].

For our training models, we use Keras deep learning tools which provide many useful layer and parameter. The model of the Keras suite has been trained with the following values of its parameters:

CNN: number of convolution layer nodes: 128, Kernel size:5, Max pool: 4D

LSTM: Number of hidden nodes: 128

We used Drop out: 0.4, Activate function in the output layer: Sigmoid, Training iteration (epochs): 1000, Batch size: 500, Optimization function: Adam, Loss function: Binary cross entropy

C. Experiment Results

We evaluated our proposed model – hybrid deep learning – HDL base on Precision, Recall and F1-score with test dataset. To evaluate our proposed model, we trained CNN, LSTM model separately and compare the result of our model with other models. The result shows that our proposed model got better than all another models both on F1-score and Precision as shown in Table 4.

When classifying a keyword, the output is between 0 and 1. In normal, the threshold of classification is 0.5. It means when a score of a new candidate keyword is higher than 0.5, it is a keyword.

Model	Precision	Recall	F1-Score
CNN	0.2914	0.6618	0.3766
LSTM	0.3113	0.6414	0.3879
HDL	**0.3466**	0.6014	**0.4051**

Table 4: Comparison of the proposed model with others

Threshold	F1-Score
0.3	0.4327
0.4	0.4296
0.5	0.4324
0.6	0.4201
0.7	0.3968

Table 5: Comparison of Thresholds

Choosing threshold is a part of the decision component. We tried five thresholds in the range [0.3 – 0.7] to decide which is a keyword with our test data. We used F1 score to evaluate our best model’s Threshold as shown in Table 5.

V. CONCLUSION

The automatic keyword extraction from Vietnamese texts plays an important role for many applications including mobile browsing website, search engines, sales key words, … It also is the subtask of other natural language processing like text summarization, text categorization, document clustering.

This paper has presented a solution for automatic extract keyword in Vietnamese texts. In our experiments, testing with topic about health care is positive; the hybrid deep learning approach shows the acceptable result. We believe that larger data about other topic may show the better predictable results. Our proposed model has also showed that hybrid deep learning approach by combining CNN and LSTM will give the best result for our corpus and solution.

The negative consequence of this approach is the requirement of large data, which seems a time-consuming process for human. We will improve our corpus in the future. Moreover, this task also requires other criteria in NLP such as text tokenize, Pos tagging and name entity recognition for better accuracy. Those tasks in Vietnamese text are also not yet familiar and challenging.

REFERENCE

[1] J. Wang, H. Peng, J.-S. Hu. “Automatic Keyphrases Extraction from Document Using Neural Network”. ICML, 633-641, 2005.

[2] Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C. & Nevill-Manning, C.G. Domain-specific keyphrase extraction. In Proceeding of 16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, pp.668-673, 1999.

[3] P. Turney. “Learning Algorithms for Keyphrase Extraction”. Information Retrieval 2, pp. 303–336, 2000.

[4] Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria. “Recent Trends in Deep Learning Based Natural Language Processing”. IEEE Computational Intelligence Magazine, 2018.

[5] X. Jiang, Y. Hu and H. Li. “A ranking approach to keyphrase extraction”. In: SIGIR, pp. 756–757, 2009.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. “Distributed representations of words and phrases and their compositionality”. In Proceedings of NIPS. 2013

[7] Kim Y. “Convolutional Neural Networks for Sentence Classification”. Proc.Conf. EMNLP (Doha) pp 1746–51, 2014

[8] Ozan Irsoy and Claire Cardie. “Opinion mining with with deep recurrent neural networks”. In Proceedings of EMNLP, pages 720–728, 2014.

[9] Wang P, Qian Y, Soong F K, He L, Zhao H. “Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network”. Cornell University. 2015

[10] Sundermeyer M, Ney H and Schluter R. “From Feedforward to Recurrent LSTM Neural Networks for Language Modelling J. IEEE/ACM Trans. Audio Speech Lang. Process. Issue 3, pp 517–29, 2015

[11] Qi Zhang, Yang Wang, Yeyun Gong, Xuanjing Huang. “Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter”. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 836–845, 2015.

[12] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky and Yu Chi. “Deep Keyphrase Generation”. 55th Annual Meeting of Association for Computational Linguistics, 2017.

[13] Hochreiter, Sepp; Schmidhuber, Jürgen. "Long Short-Term Memory". Neural Computation. 9 (8): 1735–1780, 1997.

[14] Chawla, N.; Bowyer, K.; Hall, L.; and Kegelmeyer, P. “Smote: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence 16:321–357, 2002.