Development of Hybrid Deep Learning in Sentence Classification

Thiptanawat Phongwattana, Praisan Padungweang, and Jonathan H. Chan*
School of Information Technology,
King Mongkut's University of Technology Thonburi,
Bangkok, Thailand
thiptanawat.p@mail.kmutt.ac.th, praisan.pad@sit.kmutt.ac.th, jonathan@sit.kmutt.ac.th

***Abstract*— This paper explores the combination of two deep learning techniques that consists of convolutional neural networks (CNN) and long short-term memory recurrent neural networks (LSTM-RNN) as a hybrid approach to sentence classification. The technique used CNN in feature extraction, followed by LSTM-RNN to build a classifier. In the text mining field, the performance mostly depends on word features that we need to utilize the distinctive point of CNN to re-organize the feature set for training in the LSTM-RNN layer. In feature initialization, we used a pre-trained GloVe dataset of Common Crawl to reproduce our existing dataset that includes research articles, which were extracted into a set of sentences of success factors relationship statements. We hypothesize that using the hybrid deep learning technique, CNN and LSTM-RNN, with word embedding, GloVe, can improve the performance of sentence classification in terms of recall and precision from the prior work that used only a single CNN technique.**

I. Introduction

Currently, machine learning is increasingly being utilized in research and industry fields. Much interest has been developed in various established machine learning techniques as computational means have improved in form of faster parallelized CPUs and GPUs. Convolutional neural networks (CNN) is one of the most popular techniques that is implemented in image recognition problems. An example is a classification work that contributed a CNN model for classifying 1.2 million high-resolution images from the ImageNet dataset into a thousand different classes [1]. Feature selection is a nontrivial part of CNN that can be utilized to extract images by their pixels and color channels into a set of multi-dimensional vectors. Thereafter we use the features set as inputs to build a classifier by using multi-layer perceptron (MLP) neural networks, and the outputs in terms of accuracy, precision, as well as recall, are exceptional. Therefore, we would like to utilize the benefits of CNN in terms of feature extraction in the text mining domain as the classifiers' effectiveness is significantly affected by the proper selection of a set of appropriate features. In text mining, many tools are available, for example GENIA, BANNER, Turku Event Extraction System (TEES), Stanford natural language processing (NLP); however, there haven't been tools utilizing a neural network technique yet.

In this paper, we would like to build a set of word features in each sentence by using a combination of CNN and pre-trained word vectors. Thereafter we use another kind of neural networks, long short-term memory recurrent neural networks (LSTM-RNN) for training a classifier in sentence classification. By doing so, we define both deep learning techniques that are used in the same pipeline as a novel hybrid deep learning technique. As for the research domain, we used the corpus from Krathu et al. [2] as our base line for comparison. According to the previous work, they used many kinds of techniques for feature representation and classification, for example Naïve Bayes, Binary, Term frequency-inverse document frequency (TFIDF), Logistic Regression. None of the techniques produced good performance in both recall and precision. Consequently, we would like to explore the use of deep learning in sentence classification for this prior work in order to improve the recall and precision values simultaneously.

In particular, we utilized glove vector (GloVe) [3], which was developed by Stanford, for word representation. For the GloVe dataset, we used pre-trained word vectors from Common Crawl that includes 840 billion tokens, 2.2 million vocabularies as well as 300 dimensional vectors. The dataset was used as an input to convolutional neural networks (CNN) to build the second layer after the word representation layer in order to extract useful word features. Thereafter, LSTM-RNN was implemented for training a classifier. The corpus that we used was manually exported from a database that consists of many research papers in a business-related domain. Each research paper was already extracted into sentences, and each sentence was assessed by 2 domain experts for identifying the success factor statements within each article.

According to the assessment, the identification was categorized into binary classes comprising positive, which refers to a success factor, and negative, which is not. The corpus is separated into 2 sub-corpora with 80 percent for training and another 20% for testing. We elaborately build the corpora by balancing between 2 classes in order to minimize biases that could be an issue when training a classifier.

II. Methodology

According to Fig. 1, we used the existing corpus from Krathu et al. [2] in order to compare our technique with their results. First, we have to access to Hyporet KMUTT database that belongs to School of Information Technology,

978-616-455-375-0 © 2018 JSCI

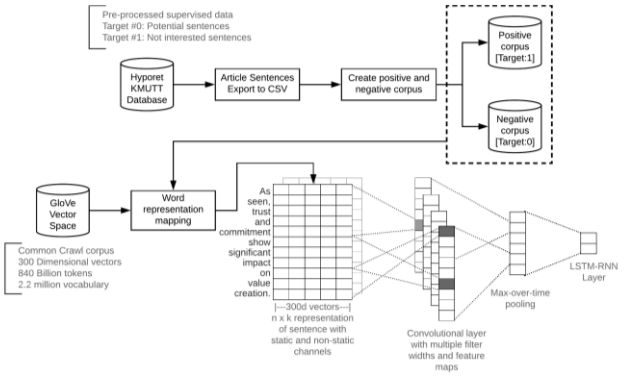

Figure 1. Model architecture with two deep learning techniques in sentence classification

King Mongkut's University of Technology Thonburi in order to retrieve the raw data of business success factor statements as a CSV file. As the raw data was already pre-processed as binary classes, we were able to classify the sentences by using their attributes. For the target equals to "1", we define those sentences as the positive sentences, whereas the negative sentences are assigned the target of "0". Thereafter, we separated the dataset into two files comprising the positives and the negatives as potential statements and non-interested statements respectively. Second, an input of convolutional neural networks must be vectors. Normally if we build a text mining model from scratch, any texts or words cannot be used directly for training. Consequently, we have to convert each token in each sentence to a dimensional vector. By doing so, we use a word embedding technique that it is called GloVe, which stands for Global Vector, which was developed by a Stanford team. There are several ways for building a set of word embedding as a vector space, for example Word2Vec, CBOWs, but we found that in a variety of research domains that use word embedding, GloVe is one of the most popular and it can provide a better result. The difference of GloVe from others is it builds a co-occurrence matrix for a corpus as the prior step to factorize it to yield matrices for word vectors. According to a prior work [4] that also used CNN, Word2Vec was used as the word representation for the work; however, we proposed a change from Word2Vec to GloVe in order to improve score of the accuracy without any hyper-parameters tuning. The pre-trained GloVe dataset that we use is Common Crawl corpus, which consists of 840 billion tokens and 2.2 million vocabularies. The dataset was trained and extracted its features to 300 dimensional vectors. In this step, we use the GloVe dataset to convert the input corpora to vectors before inputting the vector space to the CNN layer afterwards. Third, convolutional layer is implemented with multiple filter widths and feature maps that are built by feature extraction, which is the distinctive point of CNN. Fourth, max-over-time pooling is enabled for the architecture. By doing so can capture the most important feature, which is one with the highest value for each feature map. Hence the pooling scheme will be utilized for dealing with variable sentence lengths. Finally, in the classification stage, we use long short-term memory (LSTM) neural networks technique in training a classifier since the technique is the-state-of-the art in text mining, which out-performs, particularly, a dataset that relates to time series such as sentences. In practical use, we also prefer Dropout technique in the part of regularization that can improve the classifier for predicting unseen data by guarding against over-fitting.

Hyper-parameter	Value	Hyper-parameter	Value
ALLOW_SOFT_PLACEMENT	True	EVALUATE_EVERY	100
BATCH_SIZE	64	FILTER_SIZES	3, 4, 5
CHECKPOINT	100	L2_REG_LAMBDA	0.0
DROPOUT_KEEP_PROB	0.5	NUM_EPOCHS	200
EMBEDDING_DIM	128	NUM_FILTERS	128

Table 1. A Set of Hyper-Parameters of CNN Architecture

According to the training corpus information, the vocabulary size is 20,982, and the hyper-parameters are listed in Table 1, along with the tuned values.

III. Results

Fig. 2 shows the preliminary result that we evaluated by using a loss function. With the first 100 steps of training, the model was optimized and dropped by 50%. Hence, given more steps and epochs can help to define its global optimum that leads the model to be more satisfactory in sentence classification.

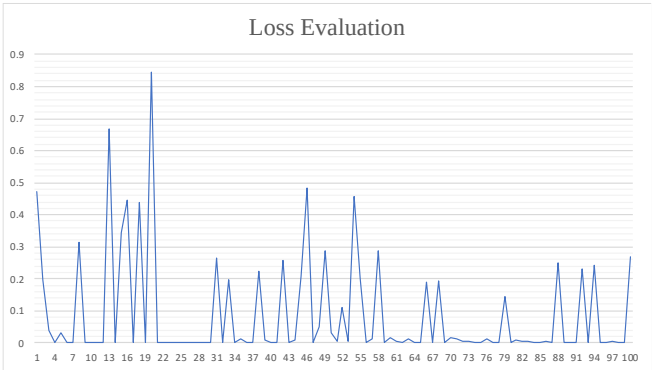

Figure 2. Loss evaluation within 100 steps.

References

[1] Alex Krizhevsky et al., "ImageNet Classification with Deep Convolutional Neural Networks", NIPS proceedings, 2012.

[2] Worarat Krathu et al., "Data Mining Approach for Automatic Discovering Success Factors Relationship Statements in Full Text Articles" 8th International Conference on Advanced Computational Intelligence, Chiang Mai, Thailand; February 14-16, 2016.

[3] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014.

[4] Yoon Kim, "Convolutional Neural Networks for Sentence Classification" New York University, 2014.