공학석사학위논문

Comparative Analysis on Encoding Methods for Convolutional Neural Network in Korean Text Classification

한국어 문장 분류에서 컨볼루션 신경망의 인코딩 방법에 따른 비교분석

2018년 8월

서울대학교 대학원
협동과정 인지과학전공
송이구

Comparative Analysis on Encoding Methods for Convolutional Neural Network in Korean Text Classification

지도교수 장 병 탁

이 논문을 공학석사학위논문으로 제출함

2018년 7월

서울대학교 대학원
협동과정 인지과학 전공

송이구

송이구의 공학석사 학위논문을 인준함
2018년 7월

위 원 장 _____ 김 홍 기 (인)

부위원장 _____ 장 병 탁 (인)

위 원 _____ 김 청 택 (인)

Abstract

Text classification is an essential task for natural language processing. Among those classifiers, Convolutional Neural Network(CNN) has recently shown strong performance in text classification. However, those researches are based on words and word-level CNN requires vast word vectors and morphological analyzers in Korean language. This study excluded the use of morphological analyzers and compared the results of classifying Korean internet news articles among different input levels of CNN. The experiment result shows that syllable-level CNN performs as well as word-level CNN, while character-level CNN shows weak performance.

Keyword: Text classification, Convolutional Neural Network, text analysis, morphological analyzer

Student Number: 2015-20105

Table of Contents

Contents of Figures

Contents of Tables

Introduction

Study Background

The classification problem in the field of Natural Language Processing has been studied for a long period of time. In natural language processing, text classification is a task of assigning documents to one or more topics according to their contents (Zhang et al., 2015). These days, massive amount of data are generated and transmitted through variety sources of media, such as mobile messengers, and the internet. Anyone could easily access to those data and that being so, text classification became an important part (Aggarwal, 2012).

Nowadays one can easily encounter large-scale data. As the internet has developed, massive information is easily accessible in all kinds of formats such as videos, music, texts. In the process of encountering more information, we are being able to access and deal with more data, and therefore making data mining crucial. Relating to that, classification problem is also becoming a crucial part, as it is helpful to sort out information from massive data. As the importance of text classification problems has grown, it has been widely studied to improve the performance of text classification (Jo et al., 2015).

Text classification is usually considered to be divided into two areas, which are sentiment analysis and categorization. In a sentiment analysis problem, one is to determine if the given document either connote positive or negative opinion towards a specific subject. Sentiment analysis can be seen in areas such as review documents. People could predict the sentiment of reviews of a movie by classifying the reviews as positive or negative based on the ratings.

On the other hand, the categorization problem is to assign documents to corresponding classes or categories which are predefined (Joachims, 1998). Categorization or classification is seen in areas such as assigning emails to correct labels or classifying news articles. In this study, we focused on improving the performance text categorization, in other words, assigning to correct predefined classes. Main assignment was to classify news articles to correct predefined categories.

Research in text classification

Text classification problem is approached through different methods. Most of the studies on text classification problem are focused on using the supervised classification, which is to train a model based on a predefined category. In unsupervised classification, no particular set of categories or answers are given. Instead the model attempts to determine the category by the similarity measure (Zechner, 2013). Unsupervised classification is applied in areas where unknown texts are used.

Traditional text classification approaches focused on feature representations. From the given documents, by selecting features one is able to assign documents to corresponding predefined categories. Setting the appropriate feature to select is what defines each category and divides one from one another (Basu 2003). At first, researchers attempted to select discriminative features and classify the documents. Feature representations are commonly based on frequency of each word in documents. Common models, such as bag-of-words (BOW) model were used to extract features. Though this kind of attempt had flaws, as they did not consider any semantics of the words and therefore ignored some contextual information (Lai, 2015).

Approaching to text classification problem, many people have

attempted different methods. Text classification studies focused on setting and retrieving the best features and choosing the best machine learning classifiers (Zhang et al., 2015).

As improvement in machine learning techniques, given documents as training sets, they have shown impressive performances. Machine learning methods, such as Support Vector Machine (SVM) and Naïve Bayes has shown great results. Mechanism of SVM is to find an optimal hyper-plane to linearly separate data. While maximizing the boundary, SVM performs well with document vectors that are sparse (Joachims, 1998). One important trait of text data is that it is sparse. For example, we collected hundreds of news articles for this study, and as we collected more articles it meant that more features were created. Being so, it would result in document vector to contain only few non-zero entries (Joachims, 1998).

Naïve Bayes assumes that all words are independent. It is based on Bayes' theorem and assumes that events are independent from each other (Jo et al., 2017).

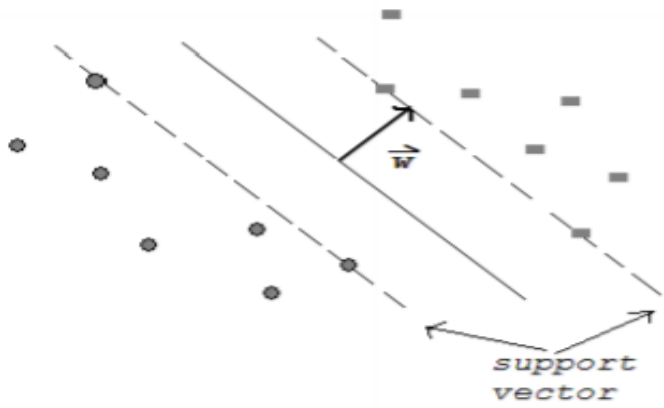

Figure 1. Illustration of SVM (Basu 2003)

As mentioned above, previous feature representation methods could not capture the contextual information and therefore not able to distinguish between homonym. However, with the emergence and rapid growth of deep neural network, researchers had attempted various methods (Lai, 2015). There have been notable studies on natural language processing, such as Collobert and Weston (2008), (Bengio et al. 2001), interest in natural language processing has grown (Schwenk et al., 2017).

Resolving the feature representation Socher et al. (2011) proposed Recurrent Neural Network. Recurrent Neural Network receives input data as a sequence (Lai, 2015). Recurrent Neural Network is effective dealing with hand writing, speech recognition, and time series data. As effective dealing with sequential data,

Recurrent Neural Network is applied in text classification tasks and has shown better results in capturing contextual information. However, Recurrent Neural Network could not perfectly deal with semantics of sentences and also had a problem in capturing contextual information. Classifiers based on Recurrent Neural Network model shown to be biased as words appearing at the end being more effective and dominant than the words that appeared earlier (Lai, 2015). To deal with this problem, research on text classification with different methods is ongoing as Convolution Neural Network being one of the methods.

While many machine learning techniques has risen, recent studies has shown impressive performance using Convolutional Neural Networks (CNN) and showing possibilities. Convolutional Neural Networks are known to process well in extracting raw signals and be effective in fields such as computer vision and speech recognition (LeCun 1989). Applying convolutional neural network to text classification has also risen. Yoon Kim (2014) has claimed that convolutional neural network has shown impressive results in sentence classification, though it has not reached the same level as those in computer vision and speech recognition. Also, Santos (2014) has shown impressive results in sentiment analysis using convolutional neural network.

.

Using text data as an input data, it means that the text data are computable. In order to successfully precede natural language processing, text data have to be represented in computers. Unlike input data in computer vision, which are presented in numbers (LeCun 1989), in text classification field, it is presented in text. As being so, one should make the text data into a computable format. This process is represented as word embedding or word representation. One possible way to treat text data to computable format was to create a dictionary of words and give corresponding id number for each word. For example, the word 'food' could have an id number 5, while the word 'school' with a number of 10. An example of sentence converted into numbers is illustrated in 1.

1. I went to school on a school bus ⟶ (1, 2, 3, 4, 5, 6, 4, 7)

Another way to convert words into numbers is using the One-Hot encoding method. An example of one-hot encoding method is illustrated in 2. However, problem with one-hot encoding method is that as more words come in, the vector becomes sparser.

2. I like dogs ⟶ (1, 0, 0), (0, 1, 0), (0, 0, 1)

Problem with the above methods is that if one processes in that way one needs to create a dictionary for every single word, which could result in a massive dictionary. Also, in this way it cannot distinguish repetitive words. For instance, there needs to be a dictionary for ‘dog’ and ‘puppy’, however, those two words meaning the same. To resolve this kind of problem methods called word2vec have come out. Another flaw is that it cannot consider the contextual information and therefore cannot distinguish between homonyms.

One possible solution was to convert words into a vector space and giving them placement information. The process holds on to context information. The method concentrates on the placement of each word and holds the information, as words that has similar meaning being close and on the other case being far apart (Mikolov 2013).

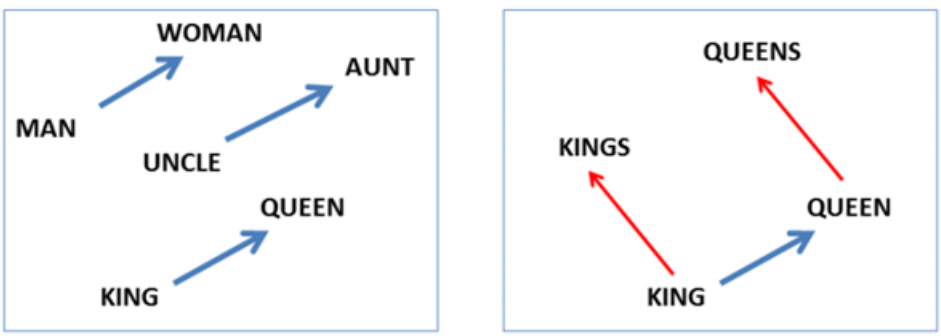

Figure 2. Illustration of word2vec (Mikolov, 2013)

.

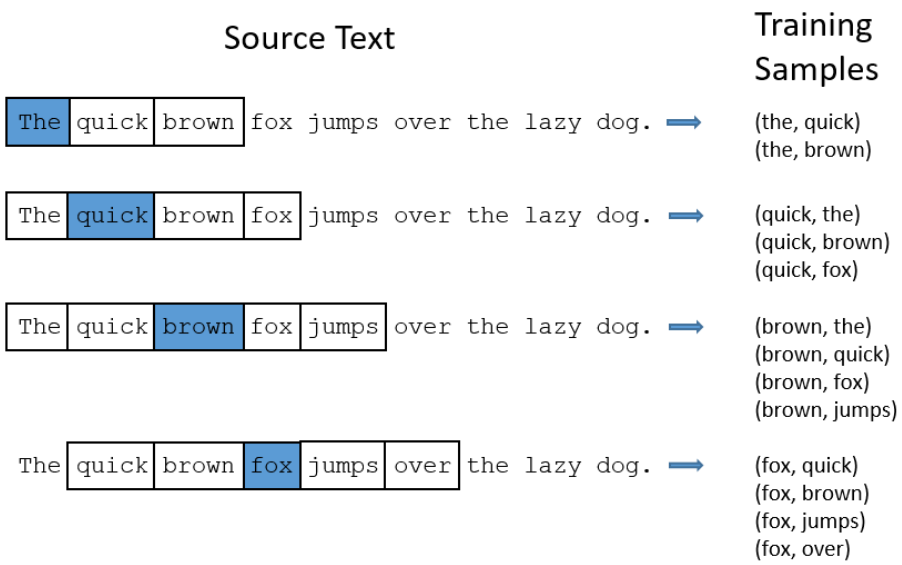

Figure 3. Example of a sentence in skip-gram model. Figure shows the target and context words and has a window size of 2 (McCormick, 2016).

There exists many word embedding methods, such as Latent Semantic Analysis, Latent Dirichlet Allocation, Word2Vec. Out of those methods, Word2Vec is widely used. Word2Vec has two types being continuous bag-of-words (CBOW) and Skip-Gram. CBOW estimates the target word based on the surrounding words. For example, in a sentence "I went to school", CBOW would predict 'went' based on 'I' and 'to school'. Skip-Gram operates the opposite way. It predicts the surrounding words based on the target word. As in figure 3, the vector holds the information for the words .

surrounding them. Holding the information of the surrounding and given that into a vector, it holds context information. Then it will be possible to represent the words consisting its meaning such as in figure 2 (Mikolov, 2013). It is essential to implement the word2vec method before processing the training stage in machine learning methods.

As there are numerous languages in the world, it consists of different language characteristics. Thus, different languages should be treated in a different way as they could give different data structures. To extract information, it is important to correctly classify the Parts-of-Speech (POS).

Korean language is somewhat different from English language. Unlike English language, Korean language has a different sentence structure. Korean language is more centered with verbs whereas English is more concentrated on the nouns. The parts of speech differ from each other, so Korean language should be dealt differently in the process of text classification. For example, English language and Korean language are different in dealing with compound nouns. As in English language it may be treated as two-word counts, while in Korean language it is only treated as one. Also, in English language space usually what divides the compound nouns, though it is different in Korean language. Different sentence

structure results in different part of speech tagger. In English language, python NLTK library (Bird et al, 2009) is used for natural language processing. In Korean language, there is a python KoNLPY library (Park & Cho, 2014) which allows us to perform the Part of Speech tagger in Korean. As part of speech tagging process defines the corresponding morphemes based on its definitions and contexts (Park & Cho, 2014). KoNLPY package provides several Part of Speech taggers, which have their own pros and cons, 'KOMORAN' and 'Twitter' was used in this study.

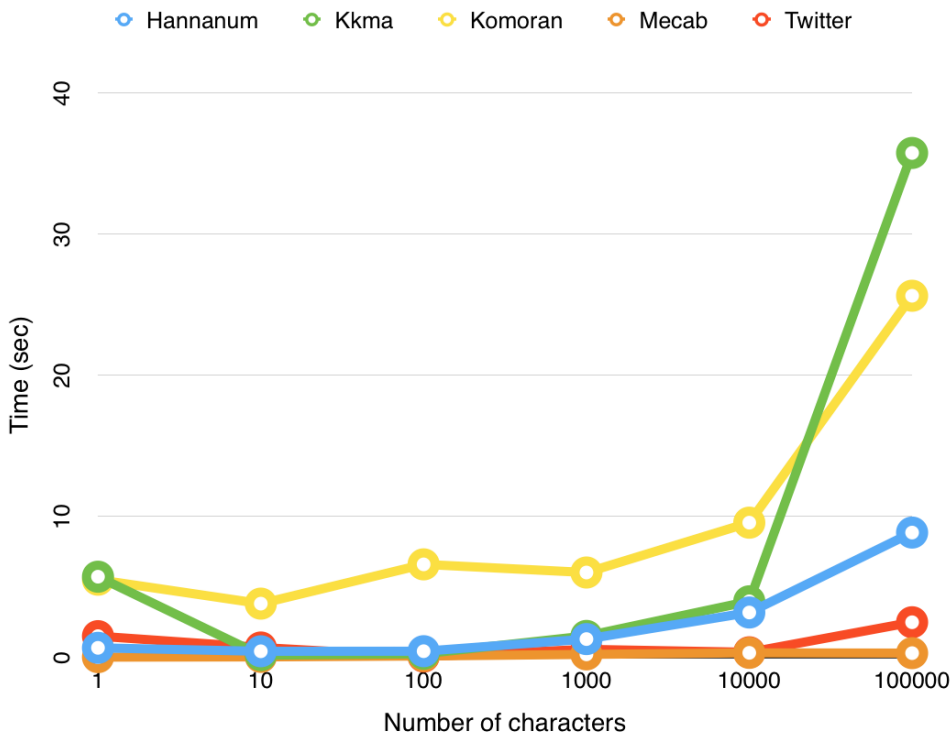

Figure 4. POS tagging classes and their transition of execution time as number of characters increase. ①

Deep Learning models, such as Recurrent Convolutional Neural Network (Lai, 2015), Convolutional Neural Network (CNN) (Kim, 2014), were used to capture the contextual information. Unlike other fields such as computer vision and speech recognition, that have achieved outstanding results through deep learning models, text classification problem has been rarely focused until Yoon Kim's attempt with Convolutional Neural Network.

① Graph provided by KoNLPY hompage (http://konlpy.org/en/v0.4.4/morph/)

Convolutional Neural Networks have been widely used to solve problems in the field of vision (e.g. face recognition, handwriting recognition), but recently it is tried on Natural Language Processing (NLP). Convolutional Neural Networks on NLP, the first layer makes word vectors by referencing a lookup table (Kim, 2014).

Many researchers have found that convolutional neural networks are useful in extracting information from raw signals and saving its contextual information, ranging from computer vision applications to speech recognition and others. Convolutional Neural Network is a hierarchical model, based on the human neural network. Basic Convolutional Neural Network has a convolutional layer, and a subsampling layer or max-pooling layer. Then it is structured with a fully connected layer before the output layer (Lai et al., 2015)

Korean Text Classification

Researches on text classification started from finding the best features. Then based on those features, one would choose the best performing classifier. There have been many attempts to introduce effective text classifiers. There were attempts approached to text classification problems relating to learning word vector representations through natural language models (Bengio et al., 2003; Mikolov et al., 2013).

Recent studies of text classification in Korean text showed impressive results. Jo (2017), Kim(2016), Shin (2017) have used Convolutional Neural Network in order to classify Korean text. In other cases, such as Kim (2017), have used Recurrent Neural Network and shown impressive performance.

In the process of Korean text classification one has to go through a pre-processing process to modify the data to fit the chosen classifier. The most frequently selected classifiers are Multinomial Naïve Bayes Classifier, Support Vector Machine, and Neural Networks including deep architectures such as Convolutional Neural Networks and Recurrent Neural Networks as mentioned above (Jo et al., 2015)(Aggarwal, 2012).

As claimed above, in Korean text classification, the classifying process goes through a preprocessing-process of using

a morphological analyzer. Most of the studies relating with Korean text classification have used morphological analyzers. There have been many morphological analyzers in Korean, though there still remain some limitations. Since, there are many occasions in the Korean language the analyzer is not able to cover all of the cases, resulting an error in Parts-of-Speech.

In Korean Text classification, due to its difficulty in morphological analysis, unlike other languages such as English, there have not been many attempts to break down into the methodology. Previous researches have been used in many methodologies, such as Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM), and Convolutional Neural Networks.

Limitation of Korean in text classification

Though the existence of several morphological analyzers, there still are some limitations to them and creating some problems. One big issue for Korean text classification is the performance of the morphological analyzers. As in Korean there are many ambiguous words and sentence structure and it cannot be said that the morphological analyzers will work perfectly. Unlike English language, whereas it is important to figure out the space in between,

Korean language is different. Also, there are issues relating to ambiguous sentences. As there are ambiguous words, one needs to know the contexts. For example, the phrase '눈을 보며 말하다' could either mean speaking with the eye or speaking while watching the snow. In this example one can understand by understanding the following or previous context.

Another problem with morphological analyzers is update issue. Morphological analyzers use already created algorithm and preset. However, as new words are created, morphological analyzers are not able to tag the parts of speech correctly unless the morphological analyzers are updated. For example, the recent issue of Alpha-GO is not recognized in parts-of-speech tagging process and is defined as a wrong morpheme as illustrated in 3, which is an example of an news article title that contains the word "Alpha-Go". However, the word is divided into 'Alpha' and 'Go', instead of being just one word.

It is difficult to update the analyzer every time a new word is created. Also, symbols not included in the word level dictionaries might appear. Furthermore, there exists the possibility of having a typo.

3. '다시 생각한 알파고... 과연 인류의 희망 될까'

= ('다시', 'MAG'),
('생각', 'NNG'),
('하', 'XSV'),
('ㄴ', 'ETM'),
('알파', 'NNG'),
('이', 'VCP'),
('고', 'EC'),
('...', 'SE'),
('과연', 'MAG'),
('인류', 'NNG'),
('의', 'JKG'),
('희망', 'NNG'),
('되', 'VV'),
('ㄹ까', 'EC')

Research Question

In the case of training the classifier with attributes, such as morphemes or n-gram results, being the input, it is necessary to have a dictionary to convert the words into a word vector. However, these word dictionaries are supposed to be in a limited number to be used in Convolutional Neural Network training. As more words are used, the word dictionary would become bigger.

In this study, we compared the results among three different input levels. First input was text documents parsed by morpheme, and then building a lookup table, in other words, using the parts-of-speech taggers. The second part was parsing the text document by syllables. The last level was parsing the text document by character level. Syllable-level and character-level input have a fixed number of word dictionary and less words in the dictionary compared to the word-level. Also, word-level input is affected by the performance of morphological analyzers, whereas syllable-level and character-level input data are irrelevant with morphological analyzers.

Table 1. Example for each input level

Input level	Example for each input level
Word-level	'사과'
Syllable-level	'사', '과'
Character-level	'ㅅ', 'ㅏ', 'ㄱ', 'ㅗ', 'ㅏ'

In this study, we used 'Twitter' and 'Komoran' morphological analyzers POS taggers. 'Twitter' analyzers are useful when doing quick analysis. On the other hand, 'Komoran' morphological analyzer gave slightly but better results than 'Twitter' and while doing so, was quicker than 'Kkma'.

Methods

Data Preparation

Text data were gathered from online news websites. News articles were selected randomly from Korean Newspaper Company 'JoongAng Ilbo' and internet portal site Naver's news page. This study has selected two categories to conduct the classification task. The two categories are 'Entertainment' and 'Politics'. The 'Entertainment' category was especially concentrated on a subcategory 'Celebrities/Entertainer', while the other category is concentrated on subcategory 'Politicians'. There were 1,533 articles assigned as 'Entertainment', 870 articles assigned as 'Politics' and 5 articles that were assigned as both categories.

'Entertainment' related articles had an average count of characters of 682, while 'Politics' related articles had an average of 645 count of characters. Articles that were assigned both 'Entertainment' and 'Politics' consisted of an average count of 500 characters.

Parameter Setting

The main task was to classify between the two topics. Morphological Analyzers were used in the preprocessing process.

'Komoran' and 'Twitter' were used in the preprocessing process. Each article consists of different length of words. In this study, the maximum length of each article was set equal. For word-level it was set as 500. In other words, the part of speech tagging for each article was maximum 500. If an article was less than 500, the rest was set as 0. For input that was parsed by letter-level was set as 1000. As syllable-level input needs more length for more information it was set longer. For character input level, it was set as 3000.

Vocabulary size for word-level input is 30715. In other words, after going through the morphological analyzer, we had 30715 total words. The vocabulary size could be bigger in this case. If more articles were added, the vocabulary size would have increased. On the other hand, for syllable input level and character input level, the numbers are fixed. Vocabulary size is 11172 for syllable-level input, which is the number of every possible single syllable in Korean Language. This number will never and actually could be reduced as we omit the letters such as '뀑' that are not used in any of the Korean words. In this research we did not omit the non-used syllables. Character-level has vocabulary size of 68. As illustrated in 4, in Korean language every syllable can be divided into three parts being, onset, nucleus, and coda. The whole 68 list

can be seen in table 2.

4. 가 = ㄱ, ㅏ, 방 = ㅂ, ㅏ, ㅇ

Table 2. List of Korean language characters

Category	Symbols
Onset (초성: cho-sung)	ㄱㄲㄴㄷㄸㄹㅁㅂㅃㅅㅆㅇㅈㅉㅊㅋㅌㅍㅎ
Nucleus (중성: Joong-sung)	ㅏㅐㅑㅒㅓㅔㅕㅖㅗㅘㅙㅚㅛㅜㅝㅞㅟㅠㅡㅢㅣ
Coda (종성: Jong-sung)	ㄱㄲㄳㄴㄵㄶㄷㄹㄺㄻㄼㄽㄾㄿㅀㅁㅂㅄㅅㅆㅇㅈㅊㅋㅌㅍㅎ (종성이 없는 경우)

Unlike word-level or syllable-level input data, character-level was encoded as 'one-hot' encoding as in figure 5. As using one-hot encoding method, it would increase the size of the data and therefore requiring more training time.

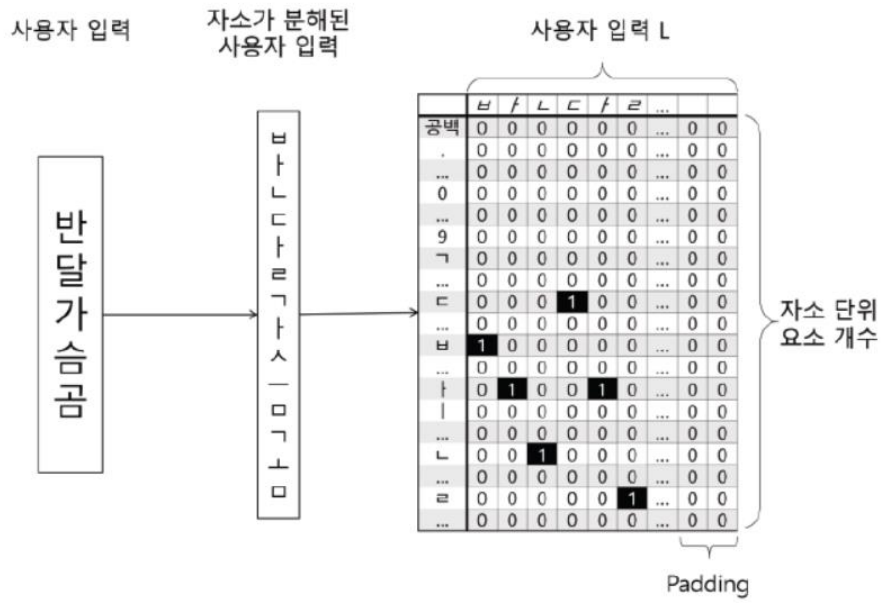

Figure 5. Diagram of vectorization for character level (Shin et al., 2017)

Another crucial parameter is dictionary size, or vocabulary size. This parameter is needed to create the look-up table and transform the text into set of ids. Figure 6 illustrates how the word is converted into a vector. First we created a word dictionary, or lookup table just as in the blue box on the right of figure 6. For each word we converted it into a vector corresponding to the same embedding size that we assigned and in this research it would be 128. After creating the lookup table, it would change each word in the sentence into vectors corresponding to the lookup table therefore creating the green box on the left in figure 6.

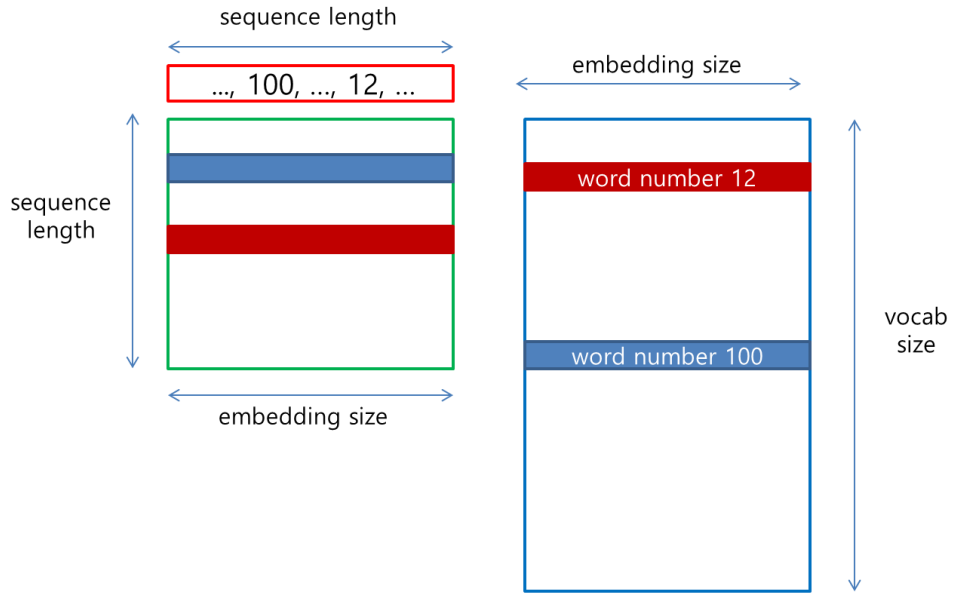

Figure 6. Structure of lookup table and its function[②]

Convolutional Neural Network Architecture

Convolutional Neural Network architecture follows that of Yoon Kim's Model (2014) as in figure 7. Deciding the filter size for word-level input, it decides the n-gram model. If the filter size is 2, it is a bigram model, and if it is 3, it becomes a trigram model. In this model, we set the filter size different for each input-level as their size is different. For example, for the word '밥을 먹다', word-level would only need 2 filters, while syllable-level needs 5 including the space, and 12 for character-level. The model contains a convolutional layer, then a max-pooling layer and finally a fully

② https://ratsgo.github.io/natural%20language%20processing/2017/03/19/CNN

connected layer.

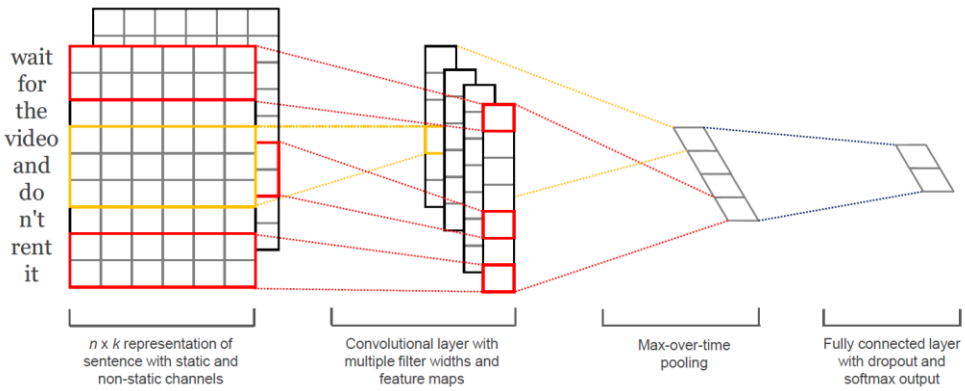

Figure 7. CNN Model architecture with two channels (Kim, 2014)

Specific parameters used in this study are seen in Table 3, 4. In the case of text data, it works similar as image data. Similar to processing image data, Convolutional Neural Network takes converted word vectors just as pixels in image data. After converting sentences to word vectors, the input data goes through convolutional layer. The size would be the number of words per document (Jo, 2017)(Kim, 2014). Embedding size, which would be the dimension of each word vector were set as 128. Filter sizes were set different as syllable-level and character-level input required more data to form one word.

Table 3. Common paramters for all input levels

Optimizer	Objective function	Train / Test ratio	Embedding size
AdamOptimizer	Softmax cross-entropy	80 / 20	128

Table 4. Parameters set for the each input level

	Vocab Size	Filter Size	Number of Filters	Max length
Word	30715	3, 4, 5	128	500
Syllable	11172	5, 10, 15	128	1000
Character	68	10, 15, 20	128	3000

Results and Discussion

Table 5. Classification accuracy and loss for each input level

Epoch number	500	1000	1500	2000	Accuracy	Mean cross-entropy loss
Word - Komoran	0.77	0.82	0.93	1	0.99	0.15
Word - Twitter	0.84	0.89	0.88	0.94	0.99	0.13
Syllable	0.88	0.92	0.97	0.92	0.98	0.23
Character	0.69	0.72	0.63	0.73	0.67	0.69

After training, it was seen that actually syllable-level input data were the followed by word-level with Twitter tagger. Training with word-level input data and syllable-level has shown great performance. Those two results are not significantly different and given impressive results. However, character had the worst performance and shown that the model was not trained well. Learning time was longest for character-level as its dimension of input data were bigger than the other two levels.

Though character-level should be taken concern that there are still possibilities of improvement. Some previous researches

have shown that character-level input data has shown impressive performances (Shin, 2017). Also, adjusting parameter values could give improvement in results. Another noticeable part is that syllable-level reached over 0.9 in accuracy faster than word-level. It seemed that syllable-level reached its maximum point the fastest by taking only around 1000 epochs to reach its maximum while the word-level needed around 1500 to 2000 epochs.

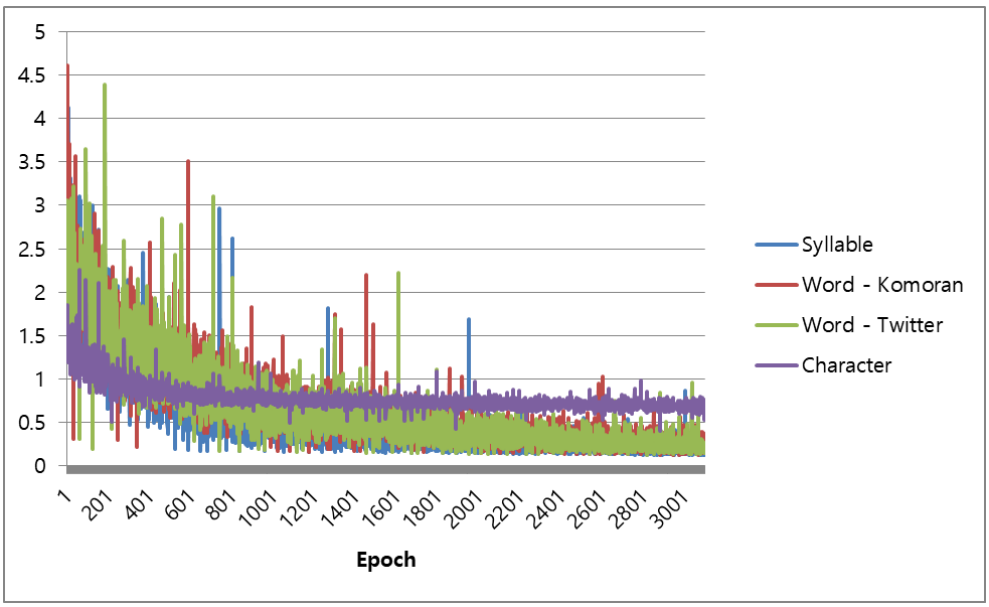

Figure 8. Mean cross-entropy loss for each input level

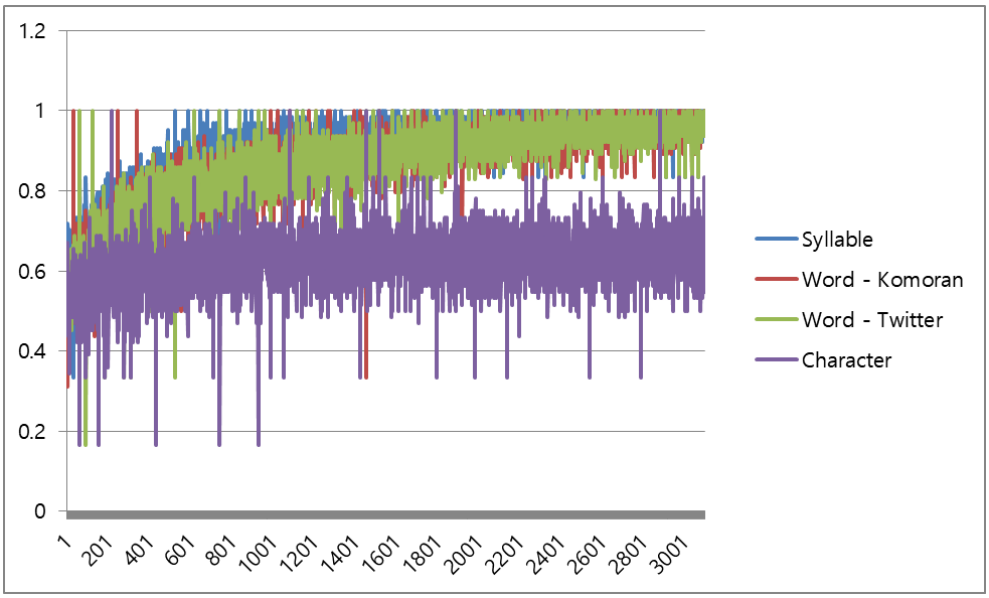

Figure 9. Accuracy score for each input level

Conclusion

Convolutional Neural Network has characteristic of obtaining crucial features of each surroundings therefore able to capture the contextual information. As Convolutional Neural Network has a process of pooling layer, without the process of defining important features, it finds important features and its surrounding features. Depending on the window size, it captures the surrounding contextual information. In this case, the mechanism is somewhat similar to those of Skip-Gram. It could be an explanation for the impressive performance in syllable-level input data. However, for character-level input data, it could not be the case as it needs bigger window size compared to syllable-level and word-level input data.

As the result above, syllable-level reached its maximum the fastest around 1000 epochs, while word-level took around 1500 to 2000 epochs. For future studies, it could be considered which input level reaches its maximum the fastest and could reduce the training time. It could be taken into notice that syllable-level might be faster while the word-level has a better result.

Though actual classification done by human might seem more effective and right, there are several difficulties. First, it is

not easy to read well. In other words, for people to read an article and assigning it to the correct topic is not an easy task. Correctly assigning topics depends on how one categorizes and decides to set the classes. For example, it would be much easier to classify basketball related articles if there is only on category for basketball, such as 'basketball', instead of 'NBA', 'Euro-basketball', 'Olympics basketball', all separated. Also, if a category is divided into too many topics (sub-categories) it would be too specific and takes more time to classify, which becomes an overfitting problem. As the categories are set, it would take much more time to read the whole text. Also, it cannot be said for sure that trained people would perfectly assign articles or text to the correct and corresponding topics.

Results with Syllable-level input data has shown impressive performance. As being so, we could reduce the reliability on morphological analyzers. Though syllable-level input data has shown impressive performance as well as word-level input data, it should be taken into concern that the classification problem was not that difficult. Entertainment news article and politics news article differ in categories using different words and have few words in common. However, classifying news articles between the category of 'finance' and 'business' would be a more difficult task as

news articles in those two categories have more words in common. Also, as further research, having more than two categories should be considered. Obviously, classifying among more than three categories would be more difficult task than distinguishing between two categories. In this study the character-level have not shown impressive results; however, in other studies such as Shin (2017) has shown that character-level input data give results as remarkable as word-level input data. Though the data used in the previous study is different, it has shown that character-level input data works. Also, while using only around 1000 news articles as training data, increasing the number of news articles could result in a better performance. As further research, it should be taken into concern that adjusting the parameters and preprocessing process, could result in a better performance in character-level.

References

권순재, 김주애, 강상우, 서정연. (2017). 문서의 감정 분류를 위한 주목 방법 기반의 딥러닝 인코더. 정보과학회 컴퓨팅의 실제 논문지, 23(4), 268-273.

김건영, 이창기. (2016). Convolutional Neural Network를 이용한 한국어 영화평 감성 분석. 한국정보과학회 학술발표논문집, 747-749.

김도우, 구명완 (2016), Doc2Vec을 활용한 CNN기반 한국어 신문기사 분류에 관한 연구. 『한글 및 한국어 정보처리 학술대회 논문집』, 제 28회

김병희, 장병탁. (2017). 순환신경망을 이용한 한글 필기체 인식. 정보과학회 컴퓨팅의 실제 논문지, 23(5), 316-321.

김정미, 이주홍. (2017). Word2vec을 활용한 RNN기반의 문서 분류에 관한 연구. 한국지능시스템학회 논문지, 27(6), 560-565.

박재홍, 윤성로. (2016). Word, Subword, Character 수준의 한국어-영어 인공신경망 기계번역 성능 비교. 한국통신학회 학술대회논문집, , 60-61.

배수현, 최인규, 이준엽, 이현승, 김남수. (2017). Recurrent Convolution Network 를 이용한 음향 환경 분류에 관한 연구. 한국통신학회 학술대회논문집, , 562-563.

신동원, 이연수, 장정선, 임해창. (2016). CNN-LSTM을 이용한 대화 문맥 반영과 감정 분류, 제28회 한글 및 한국어 정보처리 학술대회 논문집, 141-146.

신준수, 김학수. (2010). 강건한 한국어 상품평의 감정 분류를 위한 패턴 기반 자질 추출 방법. 정보과학회논문지 : 소프트웨어 및

응용, 37(12), 946-950.

신해빈, 서민관, 변형진. (2017). 한국어 자모 단위 기반의 Convolution Neural Network를 이용한 텍스트 분류. 한국정보과학회 학술발표논문집, , 587-589.

임좌상, 김진만. (2014). 한국어 트위터의 감정 분류를 위한 기계학습의 실증적 비교. 멀티미디어학회논문지, 17(2), 232-239.

조휘열, 김진화, 윤상웅, 김경민, 장병탁. (2015). 컨볼루션 신경망 기반 대용량 텍스트 데이터 분류 기술. 한국정보과학회 학술발표논문집, , 792-794.

최상혁, 설진석, 이상구. (2016), 한국어에 적합한 단어 임베딩 모델 및 파라미터 튜닝에 관한 연구. 한글 및 한국어 정보처리 학술대회 논문집』, 제 28회

황재원, 고영중. (2008). 감정 자질을 이용한 한국어 문장 및 문서 감정 분류 시스템. 정보과학회논문지 : 컴퓨팅의 실제 및 레터, 14(3), 336-340.

황재원, 고영중. (2007). 효과적인 감정 자질을 이용한 한국어 문서 감정 분류 시스템. 한국정보과학회 학술발표논문집, 34(1A), 60-61.

Aggarwal, C.C., & Zhai, C. (2012). A Survey of Text Classification Algorithms. Mining Text Data.

Basu, A., Watters, C.R., & Shepherd, M.A. (2003). Support Vector Machines for Text Categorization. HICSS.

Berger, M.J. (2015). Large Scale Multi-label Text Classification with Semantic Word Vectors.

Cho, K., Merrienboer, B.V., Gülçehre, Ç., Bahdanau, D., Bougares, F.,

Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. EMNLP.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. ICML, pp.160-167.

Goldberg, Y., & Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. CoRR, abs/1402.3722.

Grave, E., Mikolov, T., Joulin, A., & Bojanowski, P. (2017). Bag of Tricks for Efficient Text Classification. EACL.

Graves, A. (2008). Supervised sequence labelling with recurrent neural networks. Studies in Computational Intelligence.

Huang, W., & Wang, J. (2016). Character-level Convolutional Network for Text Classification Applied to Chinese Corpus. CoRR, abs/1611.04358.

Hwiyeol Jo, Jin-Hwa Kim, Kyung-Min Kim, Jeong-Ho Chang, Jae-Hong Eom, Byoung-Tak Zhang. (2017). Large-Scale Text Classification with Deep Neural Networks. 정보과학회 컴퓨팅의 실제 논문지, 23(5), 322-327.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. ECML.

Johnson, R., & Zhang, T. (2016). Convolutional Neural Networks for Text Categorization: Shallow Word-level vs. Deep Character-level. CoRR, abs/1609.00718.

Johnson, R., & Zhang, T. (2015). Effective Use of Word Order for

Text Categorization with Convolutional Neural Networks. HLT-NAACL.

Johnson, R., & Zhang, T. (2015). Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding. Advances in neural information processing systems, 28, 919-927.

Kalogiras, V., Karlgren, J., & Kann, V. (2017). Sentiment Classification with Deep Neural Networks.

Kanaris, I., Kanaris, K., & Houvardas, I. (2006). Words vs. Character N-grams for Anti-spam Filtering.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. EMNLP.

Kim, Y., Jernite, Y., Sontag, D., & Rush, A.M. (2016). Character-Aware Neural Language Models. AAAI.

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent Convolutional Neural Networks for Text Classification. AAAI.

Le, H.T., Cerisara, C., & Denis, A. (2017). Do Convolutional Networks need to be Deep for Text Classification ? CoRR, abs/1707.04108.

LeCun, Y. (1998). Gradient-based Learning Applied to Document Recognition.

LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., & Jackel, L.D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation, 1, 541-551.

Lee, J.Y., & Dernoncourt, F. (2016). Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. HLT-NAACL.

Li, C.H., & Park, S.C. (2007). Neural Network for Text Classification Based on Singular Value Decomposition. 7th IEEE International Conference on Computer and Information Technology (CIT 2007), 47-52.

Mandelbaum, A., & Shalev, A. (2016). Word Embeddings and Their Use In Sentence Classification Tasks. CoRR, abs/1610.08229.

McCormick, C. (2016). Word2vec tutorial – the skip-gram model.

Mikolov, T., Chen, K., Corrado, G.S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. CoRR, abs/1301.3781.

Nam, J., Kim, J., Mencía, E.L., Gurevych, I., & Fürnkranz, J. (2014). Large-Scale Multi-label Text Classification - Revisiting Neural Networks. ECML/PKDD.

Santos, C.N., & Gatti, M.A. (2014). Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. COLING.

Santos, C.N., & Zadrozny, B. (2014). Learning Character-level Representations for Part-of-Speech Tagging. ICML.

Schwenk, H., Barrault, L., Conneau, A., & LeCun, Y. (2017). Very Deep Convolutional Networks for Text Classification. EACL.

Semwal, T., Yenigalla, P., Mathur, G., & Nair, S.B. (2018). A Practitioners' Guide to Transfer Learning for Text Classification using Convolutional Neural Networks. SDM.

Sheikh, I.A., Illina, I., Fohr, D., & Linarès, G. (2016). Learning Word Importance with the Neural Bag-of-Words Model. Rep4NLP@ACL.

Singhal, P., & Bhattacharyya, P. (2016). Sentiment Analysis and Deep Learning: A Survey.

Socher, R., Pennington, J., Huang, E.H., Ng, A.Y., & Manning, C.D. (2011). Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. EMNLP.

Waibel, A.H., Hanazawa, T., Hinton, G.E., Shikano, K., & Lang, K.J. (1989). Phoneme recognition using time-delay neural networks. IEEE Trans. Acoustics, Speech, and Signal Processing, 37, 328-339.

Wang, S.I., & Manning, C.D. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. ACL.

Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., & Yan, S. (2014). CNN: Single-label to Multi-label. CoRR, abs/1406.5726.

Yang, X., MacDonald, C., & Ounis, I. (2017). Using word embeddings in Twitter election classification. Information Retrieval Journal, 21, 183-207.

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative Study of CNN and RNN for Natural Language Processing. CoRR, abs/1702.01923.

Zechner, N. (2013). The Past, Present and Future of Text Classification. 2013 European Intelligence and Security Informatics Conference, 230-230.

Zhang, Y., & Wallace, B.C. (2017). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. IJCNLP.

Zhang, X., Zhao, J.J., & LeCun, Y. (2015). Character-level Convolutional Networks for Text Classification. NIPS.

Zhang, X., & LeCun, Y. (2017). Which Encoding is the Best for Text Classification in Chinese, English, Japanese and Korean? CoRR, abs/1708.02657.

국문 초록

방대한 양의 텍스트 데이터를 쉽게 접근할 수 있게 되면서 텍스트 분류에 대한 연구가 활발하게 이루어지고 있다. 그 중 하나로 합성곱 신경망(Convolutional Neural Network)을 이용한 연구들이 좋은 성능을 보이며 최근에 많은 연구가 이뤄지고 있다. 그러나 대다수의 연구는 단어단위로 끊어서 학습이 되는데, 이를 위해서는 방대한 단어 사전이 필요하다. 또한 한국어의 경우 형태소 분석기를 사용하게 되는데, 이 또한 업데이트 문제나 성능에 대한 한계점들이 있다. 본 연구에서는 대량의 인터넷 신문기사를 단어 단위가 아닌 글자 단위와 자모 단위 기반의 합성곱 신경망 학습을 진행하여, 형태소 분석기에 대한 의존도를 줄였으며, 단어 단위에 비하여 더 적은 수의 고정된 단어 사전을 사용하였다. 기존 단어 단위의 학습에 비교하여 글자 단위는 큰 차이를 보이지 않은 반면, 자모 단위는 텍스트 분류를 못 하였다.

주요어: 텍스트 분류, 합성곱 신경망, 텍스트 분석, 형태소 분석기

학 번: 2015-20105

Appendix

Example of News Articles

- Entertainment News

버스커버스커, 톡식, 예리밴드(위부터)엠넷 `슈퍼스타K3`가 11일 최종 결승전을 앞둔 가운데 주요 오디션프로그램 `톱`들과 예리밴드의 묘한 인연이 눈길을 끈다. 버스커버스커(장범준, 브래드, 김형태)는 `슈퍼스타K3`의 생방송 관문을 헤쳐 나가며 울라라 세션과 최종 우승을 가리게 됐지만 사실 `운`이 크게 작용했다. 알려졌다시피 버스커버스커는 `슈퍼위크`에서 탈락, 생방송 진출이 좌절됐던 팀. 하지만 생방송 경연에 진출했던 예리밴드(한승오, 유예리, 김하늘, 김선재)가 제작진의 편집에 불만을 나타내며 자진 하차하면서 추가 합격의 행운을 누릴 수 있었다. 당시 예리밴드의 하차로 버스커버스커와 헤이즈가 추가로 생방송에 진출하면서 톱11에 올랐지만 헤이즈는 첫 생방송에서 탈락하는 아픔을 맛봤다. 반면 버스커버스커는 승승장구, 우승을 눈앞에 두게 됐다. 예리밴드로서는 하차가 아픈 추억이지만 버스커버스커에게는 `천우신조`였던 셈이다. 최근 KBS 2TV 밴드서바이벌 `톱밴드`에서 우승한 톡식(김정우, 김슬옹)은 예리밴드와 좀 더 직접적인 인연을 갖고 있다. 예리밴드와 톡식은 `톱밴드` 이전부터 음악활동을 통해 안면이 있었고, 특히 예리밴드가 톡식의 음악에 여러모로 많은 도움을 주고 있는 것으로 알려졌다. 특히 예리밴드 리더 한승오와 톡식의 인연은 더욱 긴밀하다. 톡식에서 보컬과 기타를 맡고 있는 김정우는 우승 직후 인터뷰에서 19살 때부터 맺은 인연을 소개하며 "톡식의 시작 이전부터 항상 (한)승오 형에게 많은 것을 배웠다. 인간적으로나 음악적으로 너무 감사한 분"이라고 말하기도 했다. 이들은 최근 공동레이블을 내고 함께 활동에 나섰다. 11일 `슈퍼스타K3`에서 버스커버스커가 최종 우승하면 예리밴드는 `슈퍼스타K3`와 `톱밴드`등 두 오디션프로그램 우승자의 우승에 힘을 보탠, 오묘한 운명의 팀으로 기록될 전망이다.

- Politics News

오전에는 당내 화합에 주력]박근혜 새누리당 대선 후보는 7일 오전 9시 서울 여의도동 당사에서 전직 장·차관, 국회의원 등으로 구성된 '국책자문위원회' 임명장 수여식에 참석한다. 이어 10시 30분에는 영등포 타임스퀘어 아모리스홀에서 열리는 '제1차 전국위원회'에 들려 대선 승리를 위한 당내 화합과 결속을 다진다.오후에는 서울 중구 프레스센터에서 열리는 '한국여성유권자연맹 해피바이러스 콘서트' 에 참석, 이웃 사랑을 실천하는 청년들을 격려할 예정이다. '해피바이러스 사업'은 청년들이 돌봄가정 아이들에게 공부를 가르쳐주는 등 멘토 역할을 하는 프로그램이다.박 후보는 또 김성주 공동선대위원장과 함께 서울 노원구에 위치한 서울여자대학교 학생누리관 소극장에서 '박근혜-김성주의 걸투(Girl Two) 콘서트'를 개최한다.이 자리에서 박 후보는 여대생들이 미리 제출한 질문을 즉석에서 뽑아 답변을 할 계획이다. 또 허심탄회한 대화를 통해 학생들과 소통하는 시간도 가질 예정이다.

- News tagged both ‘Entertainment’ and ‘Politics’

오전 서울 여의도동 국회의원회관에서 '2015 국가브랜드컨퍼런스'가 열렸다.배우 김수현이 컨퍼런스를 바라보고 있다.'국가브랜드컨퍼런스'는 이에리사 의원, 전순옥 의원, 국가브랜드진흥원이 주최하며 MattiHeimonen 주한 핀란드 대사와 남경필 경기도지사의 기조연설과 함께 스포츠, 문화, 기업 등 각 분야 별로 선정된 국가브랜드대상의 선정 결과를 발표하고 시상한다.올해 처음으로 시행되는 '국가브랜드대상'은 개인 및 기업 브랜드가 국가 이미지에 미치는 영향에 대하여 그 공로를 인정하고, 앞으로의 국가 브랜드 활동을 장려하여 전 세계에 한국에 대한 긍정적인 국가 이미지를 확산시키고 국가 브랜드 가치를 상승시키기 위하여 마련 되었다. 2015년 국가브랜드대상 문화 부문에는 배우 김수현, 스포츠 부분에는 김연아 선수가 선정되었고, 국가 브랜드 가치 향상에 가장 크게 공헌한 기업 부문에는 아모레퍼시픽그룹이 선정되었다.