Presented at Research Congress 2012
De La Salle University
March 7-9, 2013

NEURAL NETWORK BASED KEYWORD EXTRACTION USING WORD FREQUENCY, POSITION, USAGE AND FORMAT FEATURES

Juan Paolo Tensuan[1], Arnulfo Azcarraga[1]
[1] De La Salle University

Abstract: A Backpropagation Neural Network (BNN) is known to be useful in extracting keywords from journal articles. The journal articles were used to generate a corpora of words that were categorized as either keyword or non-keyword depending on whether the word appears in the title of the journal article. Each word is encoded as a set of features, including the number of times the word appears in the article, the frequency by which a word appears in the different parts of the document, the use of the word as part of the abstract, and figure/table captions, and various word formatting options such as bold-faced, italicized and large fonts. Unlike a similar previous work, the Inverse Document Frequency (IDF), which is a common text processing feature used for keyword extraction, was not included due to the fact that the IDF requires the entire document corpus to be analyzed, unlike the other features which can be computed solely with the information in a single document. To fine-tune the resulting BNN, a calibration phase was done against a separate validation set. Finally, features deemed useful by the BNN were inferred through the C4.5 Tree generation algorithm. Results show that the BNN can predict keywords with a precision of 90.11%, a recall of 59.50%, and an F-Measure of 0.717 with only the features previously mentioned, without the inclusion of IDF.

Key Words: Keyword Extraction, Backpropagation Neural Network, Rule Extraction, Machine Learning

1. INTRODUCTION

Information has become a commodity - so much so that the process of information retrieval has become a tedious process. Pieces of information over the web, for example, have become too disorganized and cluttered for efficient utilization. One way to alleviate this problem is to make use of keywords in documents; however, many documents lack manually annotated keywords. Hence, it is ideal to develop automatic keyword extraction methods.

There are various statistical models that are commonly used to automating keyword extraction. These models include Bayesian, K-Nearest Neighbor, TF-IDF, and Expectation Maximization (Jin, R. and Hauptmann, A., 2001). Another approach that can be employed is the use of expert systems. This approach, however, requires a manually constructed system of inference rules. These can prove to be very accurate given their specific task (Weiner et al., 1995); however, the systems are usually not flexible to other document types and can be very time-consuming to develop. Machine learning algorithms like Support Vector Machines, Genetic Algorithms, and Decision Trees have also been used to do keyword extraction (Wu et al., 2009; Zhang et al., 2009; Chou et al., 2007; Zhang et al., 2006).

This work was built from the study by Azcarraga, Liu and Setiono (2012) that investigated the use of a Bckpropagation Neural Network (BNN), a Machine Learning algorithm, in extracting keywords from text

corpora. In the earlier study, a BNN was used to be able to predict whether a given word in a document is a keyword.

In this study, we used the same Computer Science journal articles and BNN algorithm used in the previous work by Azcarraga, Liu and Setiono (2012), while extracting additional features that have to do with how a word is formatted (italicized, bold faced, larger font size). Specifically, we looked at the feasibility of dropping the Term Frequency – Inverse Document Frequency (TF-IDF) feature while trying to achieve comparable prediction accuracy rates. The TF-IDF as a feature entails the extraction of features from the entire document corpus, unlike all the other features that can be extracted based solely on the information contained in a single document. Indeed, this was the weak point of the earlier study.

To fine-tune the BNN produced, a calibration phase was employed wherein the trained model is tested against a validation set, which was separate from both the training and testing set. The aim of this phase was to get the optimum threshold for the Neural Network's output. After calibration, we finally employed the C4.5 algorithm to generate a tree that can describe how the Neural Network actually generalized the extracted features to achieve its goal of classifying keywords.

2. WORD FEATURES AND THE DATASET

A carefully selected set of criteria to define keywords must be defined to do automated keyword extraction with a BNN. These criteria, we refer to as features. Some features can prove to be useful on their own, whereas sometimes, some of these features may come hand in hand with other features. Although ideally, all features used would be useful for selecting candidate keywords, it is difficult and computationally expensive to actually have a definite subset of features that we can truly say are the best features to be used for selecting keywords. Because of this, the study simply selected features on the basis of whether these features are logical and quantifiable. By quantifiable, the feature must be easily measurable and computed. (Azcarraga et al., 2012)

The following are the features that we used for the BNN:

Word Position and Frequency

Term Frequency refers to the frequency of the word in the document.

Position in Document refers to the frequency of the word in different portions of the document.

$$F(w,i) = \frac{w_i}{w_{max}} \quad \text{(Eq. 1)}$$

Suppose the document is divided into 5 equal portions, with each portion i numerically represented, consecutively from 1 to 5. $F(w,i)$ is the function relating to the frequency of word w in portion i in relation to its frequency of the word in the portion it frequents the most. w_i is the frequency of word w in portion i whereas w_{max} is the value of the frequency of this word in the portion it most frequently appears in.

Position in Sentence refers to the average position of the term in a sentence.

$$PS = \frac{k}{N_S} \quad \text{(Eq. 2)}$$

We represent the number of words in a sentence as N_S. If the word is k^{th} word in a sentence, then its position PS is equal to k divided by the number of words in the sentence. Hence, the closer PS is to 0, the closer to the beginning of the sentence it is. The average PS is calculated from all occurrences of the word in sentences.

Position in Paragraph refers to the average position of the term in a paragraph.

$$PP = \frac{k}{N_P} \quad \text{(Eq. 3)}$$

We represent the number of words in a paragraph as N_P. If the word is k^{th} word in a paragraph, then its position PP is equal to k divided by the number of words in the paragraph. Hence, the closer PP is to 0, the closer to the beginning of the paragraph it is. The average PP is calculated from all occurrences of the word in paragraphs.

Word Usage

Used in Caption refers to whether the word was used in a table or figure caption. If the word is ever used in a caption, the value used is 1, otherwise, the value used is 0.

Used in Abstract refers to whether the word was used in the abstract. If the word is ever used in the abstract, the value used is 1, otherwise, the value used is 0.

Word Formatting

Bold-Faced refers to whether the word was used with bold-faced styling. If the word is ever used with bold-faced styling, the value used is 1, otherwise, the value used is 0.

Italicized refers to whether the word was used with italicized styling. If the word is ever used with italicized styling, the value used is 1, otherwise, the value used is 0.

Used in Larger Font Size refers to whether the word was used with a larger font size styling. If the word is ever used with larger font size styling, the value used is 1, otherwise, the value used is 0.

For this research work, Computer Science Journal articles were taken from the IEEE website. The same training set was used as with the previous work by Azcarraga, Liu and Setiono (2012); however, test set is composed of more articles than the said work. Specifically, the dataset consists of 150 journal articles for training and 227 journal articles for testing. Journal articles were procured in PDF format and were converted to HTML format, while retaining formatting features, using Adobe Acrobat. Journal articles followed the IEEE conference paper format. The format starts with the title, author, abstract, key index terms, and then the body of the research work which is formatted into two columns. For training and testing, this research work considered both title words and key index terms as keywords. Both training and testing corpora were balanced.

3. USING THE NEURAL NETWORK

Before training the dataset, the training dataset had to be balanced first. Initially, only 1% of the dataset consisted of keywords. This is natural with the dataset since in any document, there would be thousands of words but only a select few are labelled as keywords, or are in the title. Because of the nature of the BNN, the dataset was modified and balanced to achieve near a 1:1 ratio of keywords to non-keywords. This minimizes the chance of over-specializing the BNN model towards the non-keywords. Each document was represented by vectors

containing the words and their features. 13 features were used, which included both continuous and discrete information. These features were then used as the input nodes for the BNN.

The BNN consisted of three layers: the input layer, the hidden layer, and the output layer. The input layer is composed of 14 nodes, 13 of which represent the features of each data input as mentioned earlier, and 1 of which was the threshold node. The second layer, meanwhile, has 10 nodes, 1 of which, again is a threshold node. Finally, the last layer has two output nodes representing the confidence of the network for each possible prediction: keyword and non-keyword.

The learning rate used for the BNN was fixed at 0.1, while the momentum was fixed at 0.2. The allowed error was at 1.0E-5. In order to optimize the results of the BNN, a separate dataset – the validation set – was used to check for the optimal threshold for the output. The difference between the two output nodes were tested against different threshold levels starting from 0 to 1 at intervals of .01. This is unlike the conventional means that would simply look at whichever output node has the greater value. Since the dataset requires two classifications, wherein the positive class represents that the word is a keyword, a certain threshold of the difference between the positive class and negative class giving the best F-Measure for the validation set was used.

In order to further understand the BNN, a tree was generated from the resulting test set predictions using the C4.5 algorithm. By doing so, this research work attempts to see how the BNN actually utilized the attributes.

4. RESULTS AND DISCUSSION

Using the features mentioned previously, and after balancing of the dataset is done, the BNN was able to garner the results shown in Table 1.

Table 1. Results of BNN without calibration

	Predicted as Non-keyword	**Predicted as Keyword**	**Total**
Non-Keywords	1,417	99	1,516
Keywords	614	902	1,516
Total	2,031	1,001	3,032

Even without the Inverse IDF feature, a precision of 90.11% for predicting keywords was achieved. A less stellar, but still satisfactory precision of 69.77% was achieved for predicting non-keywords. The recall for non-keywords was 93.47%, while for keywords it was 59.50%. While IDF has been key to many previous research works because of its ability to statistically differentiate words that uniquely have high frequencies in only certain texts in a given corpora as opposed to words that are simply frequent in all texts in a given corpora. The BNN was able to predict keywords with an F-Measure of 0.717.

Using calibration and a separate validation set, an optimal threshold that serves as the required difference between the confidences of both output nodes was retrieved. The optimal threshold acquired was 0.99. This meant that in order for a word to be considered as a keyword, the confidence level of the output node classifying it as a keyword must be at least .99 more than the confidence of the output node classifying the word as a non-keyword.

Essentially, this means that the confidence of the output node classifying it as a keyword must be greater than or equal to 0.99.

Table 2. Results of BNN after calibration

	Predicted as Non-keyword	Predicted as Keyword	Total
Non-Keywords	1,487	29	1,516
Keywords	1,005	511	1,516
Total	2,492	540	3,032

Shown in Table 2, the BNN was able to garner a slightly better precision in classifying keywords at 94.63% after doing calibration on the threshold. However, the recall for keywords became significantly lower at 33.71%. For non-keywords, a precision of 59.67% and recall of 98.09% were achieved. The overall F-Measure for the class relating to keywords decreased to 0.497.

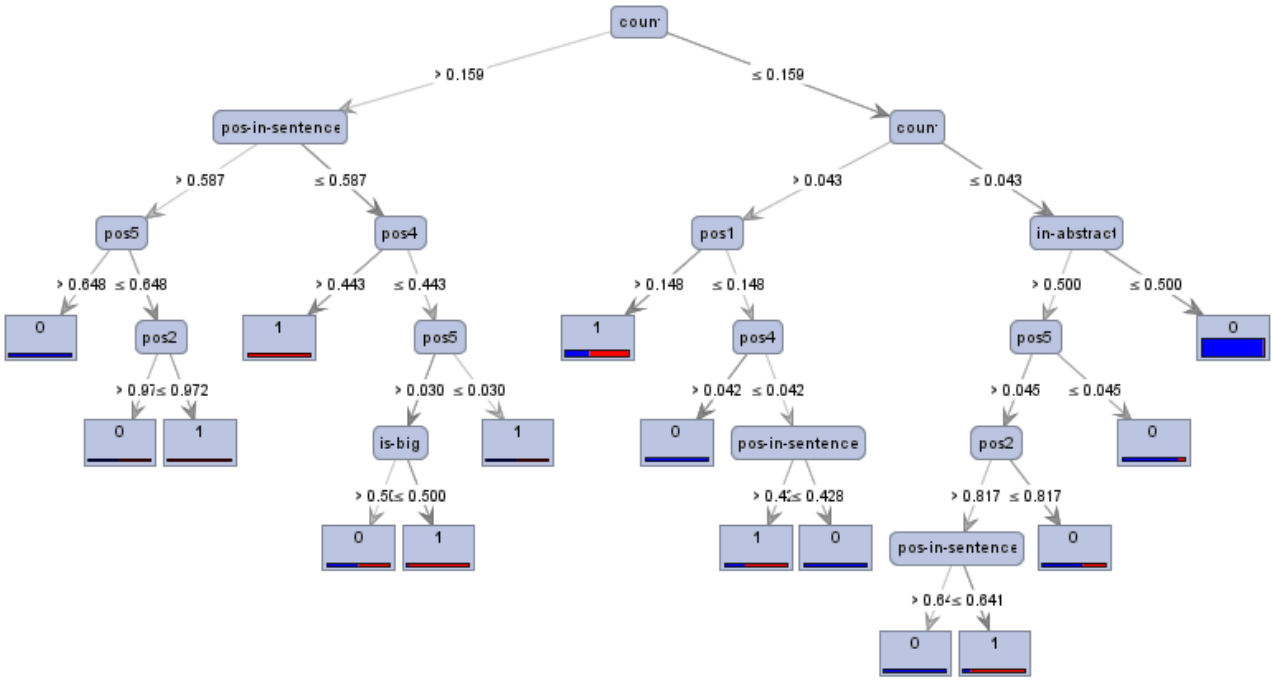

Figure 1. Decision Tree derived from Neural Network.

Using C4.5, a decision tree was generated from the predictions the BNN made. Doing so, an analysis of how the BNN generalized the features was made possible. Deduced from Figure 1, the most important features for predicting keywords were the following: in-abstract, pos1, pos2, pos5, in-caption, count and is-big, as shown in Figure 1. Pos1, pos2, and pos5, represented the frequency of the term in different positions in the document, as described earlier. Is-big indicates if the word was ever used in a font size larger than the usual font size. This meant that the words that appear in the first portion of the document and the latter two portions are more likely to be classified as keywords. This is similar to the previous study done by Azcarraga, Liu and Setiono (2012) except there is no IDF in this particular research work.

5. CONCLUSIONS AND RECOMMENDATIONS

With results showing an F-Measure of 0.717 in classifying keywords, we were able to show that for the IEEE document dataset, the use of some format features may be helpful as it may be on other datasets. We also demonstrated that the removal of IDF can actually be done while maintaining a good performance for the BNN in differentiating keywords from non-keywords. The paper was also able to affirm some of the important features indicated in the paper of Azcarraga, Liu and Setiono (2012) with regards to keyword extraction using the BNN.

Aside from showing a perspective on how the use of different features for the BNN in Keyword Extraction contributed to the prediction of keywords, we were also able to show how a calibration of the Neural Network through a validation set can help improve some aspects of the BNN's ability to classify keywords.

For future work, we recommend that the use of the same features be tested on other document datasets, and see if results, especially in extracting rules, might be drastically different. For example, heavily formatted documents like Wiki pages can be examined. Finally, we also recommend looking into developing more viable alternatives to IDF that wouldn't require the whole test corpus for feature extraction.

6. REFERENCES

Azcarraga, A., Liu, M. and Setiono, R., *Keyword Extraction Using BNNs and Rule Extraction. in Keyword Extraction Using Backpropagation Neural Networks and Rule Extraction*. In International Joint Conferenceon Neural Networks, (Brisbane, Australia, 2012), 1-7

Chih-Hsun Chou, Chin-Chuan Han, and Ya-Hui Chen. 2007. *GA based optimal keyword extraction in an automatic chinese web document classification system.* In Proceedings of the 2007 international conference on Frontiers of High Performance Computing and Networking (ISPA'07), Parimala Thulasiraman, Xubin He, Tony Li Xu, Mieso K. Denko, and Ruppa K. Thulasiram (Eds.). Springer-Verlag, Berlin, Heidelberg, 224-234.

Chunguo Wu, Maurizio Marchese, Yufei Wang, Mikalai Krapivin, Chaoyong Wang, Xitong Li, and Yanchun Liang. 2009. *Data Preprocessing in SVM-Based Keywords Extraction from Scientific Documents.* In Proceedings of the 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC '09). IEEE Computer Society, Washington, DC, USA, 810-813.

Feng Zhang, Guang Qiu, Jiajun Bu, Mingcheng Qu, and Chun Chen. 2009. *A Novel Approach to Keyword Extraction for Contextual Advertising.* In Proceedings of the 2009 First Asian Conference on Intelligent Information and Database Systems (ACIIDS '09). IEEE Computer Society, Washington, DC, USA, 51-56.

Kuo Zhang, Hui Xu, Jie Tang, and Juanzi Li. 2006. *Keyword extraction using support vector machine.* In Proceedings of the 7th international conference on Advances in Web-Age Information Management (WAIM '06), Jeffrey Xu Yu, Masaru Kitsuregawa, and Hong Va Leong (Eds.). Springer-Verlag, Berlin, Heidelberg, 85-96.

Rong Jin and Alexander G. Hauptmann. 2001. *Title generation for machine-translated documents.* In Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2 (IJCAI'01), Vol. 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1229-1234.

Wiener, E.D., Pefersen, J.O. and Weigend, A.S 1995. *A neural network approach to topic spotting.* In SDAIR-95 Proc. 4th Annual Symposium on Document Analysis and Information Retrieval (1995) 317-322.