Journal of Engineering and Applied Sciences 13 (Special Issue 14): 10896-10900, 2018
ISSN: 1816-949X

Comparison of Traditional Version and Graph based k Nearest Neighbor for Keyword Extraction in NewsPage.com

Taeho Jo
School of Game, Hongik University, Sejong, Korea

Abstract: This study proposes the version of k nearest neighbor where words are encoded into graphs, instead of numerical vectors as the approach to the task of keyword extraction. The keyword extraction is mapped into a binary classification task within a domain and the task should be distinguished from the topic based word categorization. In this research, words are encoded into string vectors each of which is represented into a list of edges, the k nearest neighbor algorithm is modified by adopting the proposed similarity metric and it is applied to the keyword extraction which is mapped into a binary classification. It is validated empirically that the proposed k nearest neighbor version is better than the traditional version in extracting keywords from a text which is tagged with its own domain. In future, we will connect the task with the text categorization in order to process texts which are untagged with their domains.

Key words: Keyword extraction, graph, graph similarity, graph based k nearest neighbor, classification, domains

INTRODUCTION

Keyword extraction refers to the process of extracting the important words from an article as its keywords. The keywords are very important indications for performing the tasks involved in information retrieval, so, the researchers and developers of information retrieval systems are interested in developing the schemes of extraction keywords, automatically. In this research, the task is viewed into a binary word classification where each word is classified into a keyword or a non-keyword. We prepare the sample words which are labeled with one of 'keyword' or 'non-keyword' and construct the classification capacity by learning the sample words. In this research, we assume that the supervised learning algorithms are used as the approach to the classification which is derived from the keyword extraction.

We consider some facts which become the motivations for doing this research. Requirement of many features for the robustness in encoding words or texts into numerical vectors causes too much computation time (Jo, 2006). The sparse distribution in each numerical vector as the additional effect of using too many features for encoding words into numerical vectors causes very poor discriminations among vectors (Jo, 2006). Recently, previous researches proposed that knowledge should be transformed into ontologies which are given graphs (Allemang and Hendler, 2011; Noy and Hafner, 1997). Therefore, in this research, motivated by the above facts, we attempt to encode words into graphs and modify the machine learning algorithm into its graph based version.

We consider some points which this research proposes as its ideas. In this research, each word is encoded into a graph with its vertices which indicate text identifiers and with its edges which indicate their semantic relations. In this research, the keyword extraction is viewed into an instance of classification task and a similarity measure between two graphs is defined. We modify the k Nearest Neighbors (kNN) into its graph based version where a graph is given as the input data by itself and use it as the approach to the keyword extraction. Even if the keyword extraction is interpreted into the word classification, it should be distinguished from the task of classifying words into one of the predefined topics.

Literature review: This study explores the previous schemes of encoding texts for using the machine learning algorithms to tasks of text mining. In encoding them into numerical vectors, there are the three main problems, huge dimensionality, sparse distribution and poor transparency. Previous researches have proposed various methods of preprocessing texts in order to solve the problems. This study focuses on encoding texts into alternatives to feature vectors. Therefore, we intend this section to explore previous works on solutions to the problems.

Sebastiani (2002) presented the numerical vectors are the standard representations of texts in applying the machine learning algorithms to the text classifications and the Support Vector Machine (SVM) was the best approach in his survey paper. Lodhi *et al*. (2002) initially proposed the string kernel as raw text kernel function in

applying the SVM to the text classification. Lesile *et al.* (2004) the string kernel version was applied to, instead of text classification, the protein classification. In 2006, the SVM version was applied used also the SVM version for categorizing sentences semantically by Kate and Mooney (2006).

Let us, survey the previous works about the process of encoding texts into tables and the table based versions of classification algorithms. Jo and Cho (2008) initially proposed the simple table based algorithm, called the table matching algorithm as the text classification tool. Jo (2008a, b) also applied the proposed algorithm to text clustering tasks as well as text classification tasks. Afterward, Jo (2011) proposed the automatic text classification system by selecting the classification algorithm as a patent. Jo (2015) the table matching algorithm was upgraded into its more stable version.

Previously, as well as tables, previous works tried to encode texts into string vectors and accordingly, modified some machine learning algorithms so. The k means algorithm was modified into the string vector based version by Jo (2008a, b). He modified the two supervised learning algorithms, the k nearest neighbor and the support vector machine into their string vector based versions as the improve text classification tools (Jo, 2010a). The string vector based neural networks which was called Neural Text Self Organizer was initially proposed by Jo (2010b). At same time, he applied the supervised string vector based neural network, Neural Text Categorizer as the text classification tool (Tae-Ho, 2010).

This research proposes the graph based k nearest neighbors, instead of the string vector based version as the keyword extraction tool. The keyword extraction task is mapped into a binary classification where each word is classified into keyword or non-keyword. Each word is encoded into a graph, instead of numerical vector. The similarity between graphs is defined in this study. This makes this research distinguished from the previous researches which are mentioned above.

MATERIALS AND METHODS

Proposed research: This study is concerned with what we propose in this research. The keyword extraction is mapped into a binary classification task where each word is classified into keyword or non-keyword. We propose the similarity metric between graphs and modify the kNN algorithm into version which computes a similarity between a training example and a test example by the similarity metric. The modified version is applied to the keyword extraction which is viewed as the word

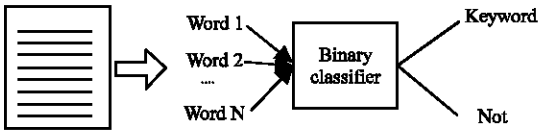

Fig. 1: Mapping keyword extraction into binary classification

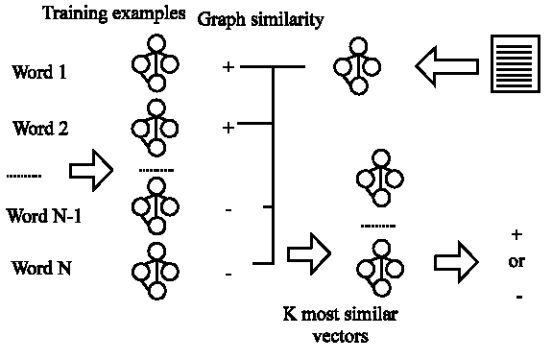

Fig. 2: The proposed version of kNN

classification task but it should be distinguished from the topic based word categorization. In this study, we describe what is proposed in this research.

Figure 1 illustrates the process of mapping the keyword extraction into a binary classification. A text is given as the input, it is indexed into a list of words and they are encoded into graphs. Each word is classified into keyword or non-keyword. Sample words are collected domain by domain as shown in Fig. 2. Instead of classification, the task may be mapped into regression where the word importance degrees are estimated as continuous values.

The proposed version of k nearest neighbor is illustrated in Fig. 2. The proposed keyword extraction system indexes an input text into a list of words and encodes the words into graphs let us assume that a graph which represents a word is given as the input. Its similarity is computed by equation which is proposed by Jo (2006) and the most k similar training words are selected as its nearest neighbors. The label of the novice graph is set by voting ones of the selected ones. The graph is classified into one of the two labels, 'keyword', or 'non-keyword'.

We explain how to apply the proposed kNN algorithm to the keyword extraction task. The sample words which are labeled with keyword or non-keyword are gathered domain by domain as shown in Fig. 3 and they are encoded into graphs. The text which is assumed to be tagged with its own domain is indexed into a list of words and for each word, its similarities with the sample words in the corresponding domain. For each word, its k nearest

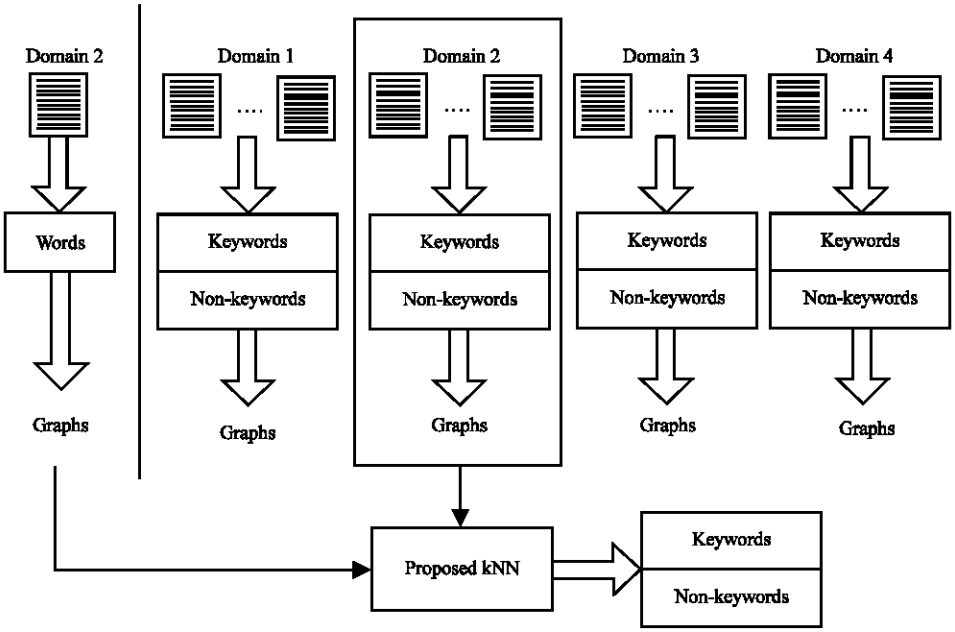

Fig. 3: Keyword extraction: domain dependent classification

sample words are selected and its label is decided by voting the labels of its nearest neighbors. The words which are classified into keyword are extracted as the keywords of the text.

The task in this research should be distinguished from the topic based word categorization, even if both tasks belong to the classification task. In the topic based word categorization, sample words are collected independently of domain whereas in the keyword extraction, sample words should be done domain by domain. In the former, a topic or a category is absolutely assigned to each word whereas in the latter, one of the two categories is assigned to word, depending on the domain. In the word categorization, a word is classified, depending on its meaning which is related with a category or a topic whereas in the keyword extraction, it is classified, depending on its semantic importance degree to the given text. The process of applying the proposed kNN to the keyword extraction is described by Jo (2006).

RESULTS AND DISCUSSION

Experiments: This study is concerned with the experiments for validating the better performance of the proposed version on the collection: NewsPage.com. We interpret the keyword extraction into the binary classification where each word is classified into keyword or non-keyword and gather words which are labeled with one of the two categories, from the collection, topic by topic. Each word is allowed to be classified into one of the two labels, exclusively. We fix the input size as 50 of numerical vectors and graphs and use the accuracy as the evaluation measure. Therefore, this study is intended to observe the performance of the both versions of kNN in the four different domains.

Table 1: The number of texts and words in NewsPage.com

Category	#Texts	#Training words	#Test words
Business	500	300	75
Health	500	300	75
Internet	500	300	75
Sports	500	300	75
Total	2,000	1,200	300

In Table 1, we specify NewsPage.com which is used as the source for extracting the classified words, in this set of experiments. The text collection was used for evaluating approaches to text categorization in previous researches by Jo (2015). In each topic, we extracted 125 words labeled with keyword and 125 words labeled with non-keyword. The set of 250 words in each topic is partitioned into the 200 words as training ones and the 50 words as the test ones, keeping the complete balanced distribution over the two labels as shown in Table 1. In building the test collection of words, we decide whether each word is a keyword or not, depending on its frequency concentrated in the given category combining with the subjectivity in scanning articles.

We mention the experimental process of validating empirically the proposed approach to the task of keyword extraction. We collect sample words which are labeled with keyword or non-keyword in each of the four domains: business, sports, internet and health, depending

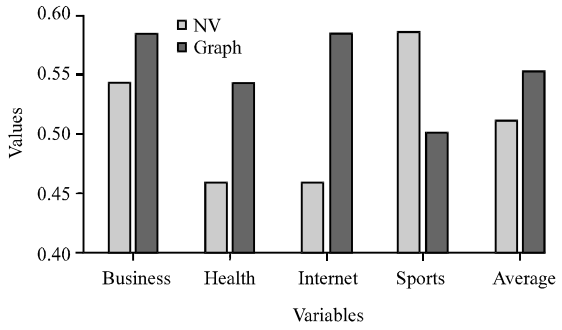

Fig. 4: Experimental results

on subjectivities and concentrated frequencies of words and encode them into numerical vectors. In each domain, for each of the 50 test examples, the kNN computes its similarities with the 200 training examples and select the three most similar training examples as its nearest neighbours. Independently, we perform the four experiments each of which classifies each word into keyword or non-keyword by the two versions of kNN algorithm. For evaluating the both versions of kNN in the classification which is mapped from the keyword extraction, we compute the classification accuracy by dividing the number of correctly classified test examples by the number of test examples.

In Fig. 4, we illustrate the experimental results from decoding whether each word is a keyword or not, using the both versions of kNN algorithm. The Y-axis indicates the accuracy which is the rate of the correctly classified examples in the test set. Each group in the X-axis indicates the domain within which the keyword extraction which is viewed into a binary classification is performed, independently. In each group, the grey bar and the black bar indicate the performance of the traditional version and the proposed version of kNN algorithm, respectively. The most right group in Fig. 4 indicates the average over accuracies over the left four groups and set the input size which is the dimension of numerical vectors and the number of edges of graphs as 50.

We make the discussions on the results from doing the keyword extraction using the both versions of kNN algorithm as shown in Fig. 4. The accuracy which is the performance measure of the classified task is in the range between 0.45 and 0.58. The proposed version of the kNN algorithm works better in the two domains: health and internet. It matches with the traditional version in the domain, business but is lost in the domain, sports. However, from this set of experiments, we conclude the proposed version works better than the traditional one, in averaging over the four cases.

CONCLUSION

As the conclusion of this research, we consider some significances. This research proposed that texts are encoded into graphs and defined the similarity metric between graphs. Based on the similarity metric, the kNN was modified as the improved classification tool. The keyword extraction was interpreted into a binary classification and the improved version was applied.

RECOMMENDATION

We mention some remaining tasks for doing the further research. We need to validate more the proposed approach in extracting keywords in specific domains such as medicine, engineering and economics and customize it correspondingly. We need to consider other schemes of encoding words into graphs and other similarity measures between graphs. We modify other machine learning algorithms into their graph based versions where a graph is given by itself as the input data. We implement a keyword extraction system by adopting the proposed approach.

ACKNOWELDGEMENT

This research is supported by 2017 Hongik University Research Fund.

REFERENCES

Allemang, D. and J. Hendler, 2011. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. 2nd Edn., Morgan Kaufmann, Burlington, Massachusetts, USA., ISBN:978-0-12-385965-5, Pages: 353.

Jo, T. and D. Cho, 2008. Index based approach for text categorization. Intl. J. Math. Comput. Simul., 1: 127-132.

Jo, T., 2006. The implementation of dynamic document organization using text categorization and text clustering. Ph.D Thesis, University of Ottawa, Ottawa, Canada.

Jo, T., 2008a. Single pass algorithm for text clustering by encoding documents into tables. J. Korea Multimedia Soc., 11: 1749-1757.

Jo, T., 2010b. NTC (Neural Text Categorizer): Neural network for text categorization. Intl. J. Inf. Stud., 2: 83-96.

Jo, T., 2010a. NTSO (Neural Text Self Organizer): A new neural network for text clustering. J. Network Technol., 1: 31-43.

Jo, T., 2011. Device and method for categorizing electronic document automatically. Korean Intellectual Property Office, Korea.

Jo, T., 2015. Normalized table-matching algorithm as approach to text categorization. Soft Comput., 19: 839-849.

Jo, T.H., 2008a. Inverted index based modified version of K-means algorithm for text clustering. J. Inf. Process. Syst., 4: 67-76.

Kate, R.J. and R.J. Mooney, 2006. Using string-kernels for learning semantic parsers. Proceedings of the Joint 21st International Conference and 44th Annual Meeting on Computational Linguistics and the Association for Computational Linguistics, July 17-18, 2006, Association for Computational Linguistics, Stroudsburg, Pennsylvania, USA.,-pp: 913-920.

Leslie, C., E. Eskin, J. Weston and W. Noble, 2004. Mismatch string kernels for discriminative protein classification. Bioinformatics, 20: 467-476.

Lodhi, H., C. Saunders, J. Shawe-Taylor, N. Cristianini and C. Watkins, 2002. Text classification using string kernels. J. Mach. Learn. Res., 2: 419-444.

Noy, N.F. and C.D. Hafner, 1997. The state of the art in ontology design: A survey and comparative review. AI. Mag., 18: 84-94.

Sebastiani, F., 2002. Machine learning in automated text categorization. ACM Comput. Surveys, 34: 1-47.

Tae-Ho, J., 2010. Representation of texts into string vectors for text categorization. J. Comput. Sci. Eng., 4: 110-127.