String Vector based Version of K Nearest Neighbor for Index Optimization in Current Affairs

Taeho Jo
School of Game
Hongik University
Sejong, South Korea
tjo018@hongik.ac.kr

***Abstract*—In this research, we propose the index optimization for maximizing both the performance and the efficiency of information retrieval systems and the application of the proposed KNN to the task. The index optimization is mapped into a classification task within a domain, the task should be distinguished from the topic based word categorization, and the previous works about text categorization and clustering show the better text categorization performances from encoding texts into string vectors than into numerical vectors. In the proposed system, a text which is tagged with its own domain is given as the input, the words which are indexed from the text are classified into expansion, inclusion, or removal, by the string vector based KNN version. We validated empirically that the proposed KNN works better than the tradition version in optimizing indexes of news articles. In future, we will connect the task with the text categorization, in order to process texts which are untagged with their domains.**

***Keywords*-Index Optimization; K Nearest Neighbor; String Vector**

I. Introduction

The index optimization is referred to the process of optimizing the text indexes for maximizing the information retrieval performance. The text indexes provide the important reference for deciding relevancy of the text, and words from doing it should be discriminated by their importance degrees. In this research, the index optimization is mapped into the classification task where each word is classified into the three categories: extension, inclusion, and removal. The index optimization system adds more relevant words to those which are classified into extension from external sources, takes words, which are classified into inclusion by themselves without adding any word, and removes the words which are classified into removal. In this research, we describe briefly the motivation, the idea, and validation of this research.

Let us consider the motivations for doing this research. It is necessary to discriminate words in each text by their importance degrees for maximizing the information retrieval performance. The index optimization can be mapped into a classification task where each word is classified into expansion, inclusion, and removal, and the KNN algorithm is a simple approach to the data classification for starting to modify machine learning algorithms. In previous works, the foundation research on semantic operations on strings was progressed [13]. In [12] and [14], the string vector based algorithm was proposed as the approach to the text categorization, and its performance was successful.

In this research, we apply the proposed KNN version to the index optimization task. The task is interpreted into the classification task where each word is classified into expansion, inclusion, or removal. In the proposed system, a text which is given as the input is indexed into a list of words, the words which are classified into expansion are added by their associate ones, and ones which are classified into inclusion are taken by themselves. The KNN algorithm is modified into the version where the similarity between string vectors is computed based on a similarity matrix which is built from a corpus. This research is intended to improve the discriminations among numerical vectors, by encoding words into alternative representations to numerical vectors.

In this research, we will validate empirically the proposed approach to the index optimization as the better version than the traditional KNN version. We extract words which are classified their own topics from the news collection: 20NewsGroups. The traditional KNN version and the proposed version are compared with each other. We observe the better results of the proposed KNN version in classifying words into expansion, inclusion, or removal. It potentially possible to require less dimension in encoding words into string vectors, in addition.

Let us mention the organization of this research. In Section II, we explore the previous works which are relevant to this research. In Section III, we describe in detail what we propose in this research. In Section IV, we validate empirically the proposed approach by comparing it with the traditional one. In Section V, we mention the significances of this research and the remaining tasks as the conclusion.

II. Previous Works

Let us survey the previous cases of encoding texts into structured forms for using the machine learning algorithms to text mining tasks. The three main problems, huge dimensionality, sparse distribution, and poor transparency, have existed inherently in encoding them into numerical vectors. In previous works, various schemes of preprocessing texts have been proposed, in order to solve the problems. In this survey, we focus on the process of encoding texts into alternative structured forms to numerical vectors. In other words, this section is intended to explore previous works on solutions to the problems.

Let us mention the popularity of encoding texts into numerical vectors, and the proposal and the application of string kernels as the solution to the above problems. In 2002, Sebastiani presented the numerical vectors are the standard representations of texts in applying the machine learning algorithms to the text classifications [1]. In 2002, Lodhi et al. proposed the string kernel as a kernel function of raw texts in using the SVM (Support Vector Machine) to the text classification [2]. In 2004, Lesile et al. used the version of SVM which proposed by Lodhi et al. to the protein classification [3]. In 2004, Kate and Mooney used also the SVM version for classifying sentences by their meanings [4].

It was proposed that texts are encoded into tables instead of numerical vectors, as the solutions to the above problems. In 2008, Jo and Cho proposed the table matching algorithm as the approach to text classification [5]. In 2008, Jo applied also his proposed approach to the text clustering, as well as the text categorization [9]. In 2011, Jo described as the technique of automatic text classification in his patent document [7]. In 2015, Jo improved the table matching algorithm into its more stable version [8].

Previously, it was proposed that texts should be encoded into string vectors as other structured forms. In 2008, Jo modified the k means algorithm into the version which processes string vectors as the approach to the text clustering[9]. In 2010, Jo modified the two supervised learning algorithms, the KNN and the SVM, into the version as the improved approaches to the text classification [10]. In 2010, Jo proposed the unsupervised neural networks, called Neural Text Self Organizer, which receives the string vector as its input data [11]. In 2010, Jo applied the supervised neural networks, called Neural Text Categorizer, which gets a string vector as its input, as the approach to the text classification [12].

The above previous works proposed the string kernel as the kernel function of raw texts in the SVM, and tables and string vectors as representations of texts, in order to solve the problems. Because the string kernel takes very much computation time for computing their values, it was used for processing short strings or sentences rather than texts. In the previous works on encoding texts into tables, only table matching algorithm was proposed; there is no attempt to modify the machine algorithms into their table based version. In the previous works on encoding texts into string vectors, only frequency was considered for defining features of string vectors. In this research, based on [10], we consider the grammatical and posting relations between words and texts as well as the frequencies for defining the features of string vectors, and encode words into string vectors in this research.

III. Proposed Approach

This section is concerned with what we propose in this research. The index optimization is mapped into a classification task where each word is classified into expansion, inclusion, or removal. We propose the semantic similarity between string vectors as the operation on string vectors, 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 index optimization which is viewed as the word classification task, but it should be distinguished from the topic based word categorization. In this section, we describe what is proposed in this research.

Figure 1 illustrates the process of mapping the index optimization into a classification task. A text is given as the input, it is indexed into a list of words, and they are encoded into string vectors. Each word is classified into expansion, inclusion, and removal; to ones which are classified into expansion, their associated words are retrieved and added, and to ones which are classified into inclusion, they are taken without adding their associated ones. Sample words are collected domain by domain as shown in Figure 3. The task may be mapped into a regression task, where word importance degrees are estimated as continuous values.

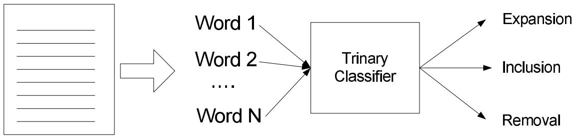

Figure 1. Mapping Index Optimization into Classification Task

Figure 2 presents the proposed KNN algorithm. In advance, the training words are encoded into string vectors. A novice word is encoded into a string vector, its similarities with ones which represent the training ones by equation which is proposed in [14], and the most k similar training words are selected its nearest neighbors. The label of the novice word is decided by voting ones of the nearest neighbors. We may consider the KNN variants which are derived from this version by discriminating the similarities and the attributes.

Let us explain how to apply the proposed KNN version to the index optimization task. The sample words which are labeled with expansion, inclusion, or removal are gathered domain by domain as shown in Figure 3, and they are encoded into string vectors. The text which is assumed to be tagged with its own domain into a list of words, and

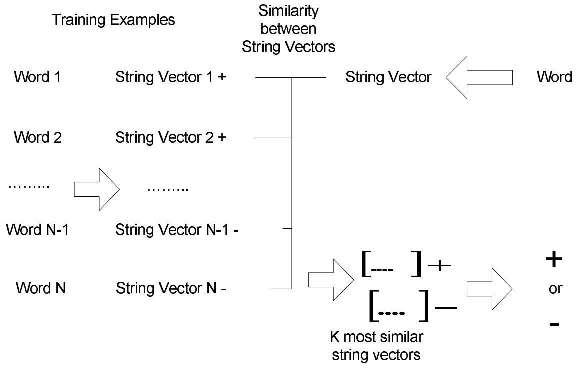

Figure 2. The Proposed Version of KNN

for each word, its similarities with the sample words in the corresponding domain. For each word, its k nearest sample words are selected and its label is decided by voting the labels of its nearest neighbors. The words which are labeled with expansion are attached with their associated words from external sources and ones which are labeled with inclusion are extracted without their associated ones.

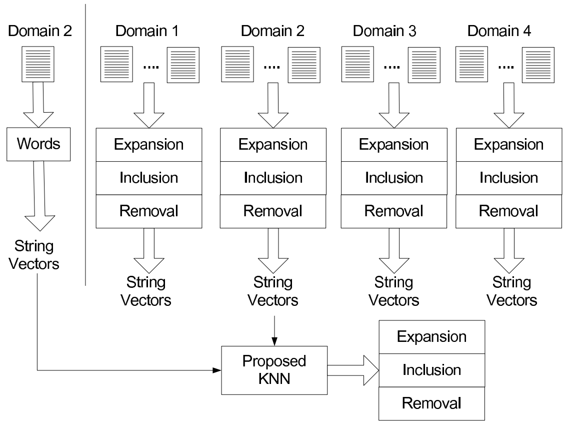

Figure 3. Index Optimization: Domain Dependent Classification

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 index optimization, 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 index optimization, it is classified, depending on its relevancy degree to the given text. The process of applying the proposed KNN to the index optimization is described in [15].

IV. Experiments

This section is concerned with one more set of experiments where the better performance of the proposed version is validated on another version of 20NewsGroups. We gather the words which are labeled with 'expansion', 'inclusion', or 'removal'. We map the index optimization into a binary classification, and carry out the independent four binary classification tasks as many as topics, in this set of experiments. We fix the input size in representing words to 50, and use the accuracy as the evaluation metric. Therefore, in this section, we observe the performances of the both versions of the KNN with the four different domains.

In Table I, we specify the second version of 20NewsGroups which is used in this set of experiments. Within the general category, sci, the four categories, electro, medicine, script, and space, are predefined. In each specific category as a domain, we build the collection of labeled words by extracting 375 important words from approximately 1000 texts. We label manually the words with 'expansion', 'inclusion' or 'removal', maintaining the complete balance. In each domain, the set of 375 words is partitioned with the training set of 300 words and the test set of 75 words, as shown in Table I.

Table I
The Number of Texts and Words in 20NewsGroups II

Category	#Texts	#Training Words	#Test Words
Electro	1000	300(100+100+100)	75(25+25+25)
Medicine	1000	300(100+100+100)	75(25+25+25)
Script	1000	300(100+100+100)	75(25+25+25)
Space	1000	300(100+100+100)	75(25+25+25)

The process of doing this set of experiments is same to that in the previous sets of experiments. We collect the sample words which are labeled with 'expansion', 'inclusion', or 'removal', in each of the four domains: 'electro', 'medicine', 'script', and 'space, and encode them, fixing the in input size to 50. We use the two versions of KNN algorithm for their comparisons. Each example is classified into one of the three categories, by the both versions. We use the classification accuracy as the evaluation metric.

We present the experimental results from classifying the words using the both versions of KNN algorithm on the specific version of 20NewsGroups. The frame of illustrating the classification results is identical to the previous ones. In each group, the gray bar and the black bar stand for the achievements of the traditional version and the proposed version, respectively. The y-axis in Figure 4, indicates the classification accuracy which is used as the performance metric. In this set of experiments, we execute the four independent classification tasks which correspond to their own domains, where each word is classified into 'expansion', 'inclusion', or 'removal'.

Let us discuss on the results from doing the index optimization on the specific version of 20NewsGroups, as

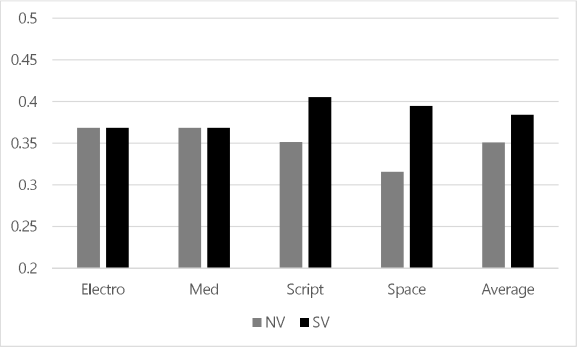

Figure 4. Results from Index Optimization in Text Collection: 20NewsGroup II

shown in Figure 4. The accuracies of both versions of KNN algorithm range between 0.31 and 0.41. The proposed version shows its better results in two of the four domains. However, it is comparable in the others. In spite of that, from this set of experiments, it is concluded that the proposed version wins over the traditional one, according to the average over the four accuracies.

V. Conclusion

Let us mention the remaining tasks for doing the further research. The proposed approach should be validated and specialized in the specific domains: medicine, engineering and economics. Other features such as grammatical and posting features may be considered for encoding words into string vectors as well as text identifiers. Other machine learning algorithms as well as the KNN may be modified into their string vector based versions. By adopting the proposed version of the KNN, we may implement the index optimization system as a real program.

References

[1] F. Sebastiani, “Machine Learning in Automated Text Categorization”, pp1-47, ACM Computing Survey, Vol 34, No 1, 2002.

[2] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text Classification with String Kernels”, pp419-444, Journal of Machine Learning Research, Vol 2, No 2, 2002.

[3] C. S. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble, “Mismatch String Kernels for Discriminative Protein Classification”, pp467-476, Bioinformatics, Vol 20, No 4, 2004.

[4] R. J. Kate and R. J. Mooney, “Using String Kernels for Learning Semantic Parsers”, pp913-920, Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, 2006.

[5] T. Jo and D. Cho, “Index based Approach for Text Categorization”,International Journal of Mathematics and Computers in Simulation, Vol 2, No 1, 2008.

[6] T. Jo, “Single Pass Algorithm for Text Clustering by Encoding Documents into Tables”, pp1749-1757, Journal of Korea Multimedia Society, Vol 11, No 12, 2008.

[7] T. Jo, “Device and Method for Categorizing Electronic Document Automatically”, Patent Document, 10-2009-0041272, 10-1071495, 2011.

[8] T. Jo, “Normalized Table Matching Algorithm as Approach to Text Categorization”, pp839-849, Soft Computing, Vol 19, No 4, 2015.

[9] T. Jo, “Inverted Index based Modified Version of K-Means Algorithm for Text Clustering”, pp67-76, Journal of Information Processing Systems, Vol 4, No 2, 2008.

[10] T. Jo, “Representationof Texts into String Vectors for Text Categorization”, pp110-127, Journal of Computing Science and Engineering, Vol 4, No 2, 2010.

[11] T. Jo, “NTSO (Neural Text Self Organizer): A New Neural Network for Text Clustering”, pp31-43, Journal of Network Technology, Vol 1, No 1, 2010.

[12] T. Jo, “NTC (Neural Text Categorizer): Neural Network for Text Categorization”, pp83-96, International Journal of Information Studies, Vol 2, No 2, 2010.

[13] T. Jo, “Simulation of Numerical Semantic Operations on String in Text Collection”, pp45585-45591, International Journal of Applied Engineering Research, Vol 10, No 24, 2015.

[14] T. Jo, “Encoding Words into String Vectors for Word Categorization”, pp271-276, The Proceedings of 18th International Conference on Artificial Intelligence, 2016.

[15] T. Jo, “String Vector based KNN for Index Optimization”, pp31-36, The Proceedings of 3rd International Conference on Advances in Big Data Analysis, 2016.