Word Classification in Domain on Current Affairs by Feature Similarity based K Nearest Neighbor

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

Abstract—In this research, we propose the modified KNN version as the approach to the word categorization. The words are encoded into numerical vectors whose features are texts, and the similarity metric between them which consider the similarities among features is defined. By adopting the proposed similarity metric, for computing the similarity between the test example and each training example, the KNN is modified. The modified version is applied to the word categorization which is covered in this research. The discriminations among numerical vectors are improved by considering the similarities among features as well as ones among feature values.

In this research, we will validate empirically the proposed approach to the word categorization 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. It potentially possible to reduce the dimension by considering the feature similarity.

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.

I. INTRODUCTION

The word categorization is referred to the process of assigning one or some among the predefined categories to each word. We may consider the two tasks as word categorizations: the lexical classification which is the process of classifying words by their spellings, and the POS (Part of Speech) tagging which is the process of classifying them by their grammatical functions. The scope is restricted to the semantic classification where words are classified based on their topics or meaning. We mention the K Nearest Neighbor as the approach and modify it into the version which considers the feature similarity, as well as the feature value similarity. This section covers the motivation, the idea, and validation of this research.

Let us consider the motivations for doing this research. The fact that we may expect the synergy effect between the word categorization and the text categorization motivates for setting the former as the task of this research. The KNN algorithm is a simple approach to the data classification for starting to modify machine learning algorithms. Texts which are features for encoding words into numerical vectors have their own semantic relations with other. In this research, by introducing the feature similarity as well as the feature value similarity, we expect the discriminations among numerical vectors to be improved.

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 as the standard representations of texts in applying the machine learning algorithms to the text classifications [4]. 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 [5]. In 2004, Lesile et al. used the version of SVM which proposed by Lodhi et al. to the protein classification [6]. In 2004, Kate and Mooney used also the SVM version for classifying sentences by their meanings [7].

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 [8]. In 2008, Jo applied also his proposed approach to the text clustering, as well as the text categorization [10]. In 2011, Jo described as the technique of automatic text classification in his patent document [14]. In 2015, Jo improved the table matching algorithm into its more stable version [15].

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[10]. 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 [11]. In 2010, Jo proposed the unsupervised neural networks, called Neural Text Self Organizer, which receives the string vector as its input data [12]. 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 [13].

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. Texts which are used as features of numerical vectors which represent words have their semantic similarities among them, so the similarities will be used for processing sparse numerical vectors, in this research.

III. PROPOSED APPROACH

This section is concerned with what we propose in this research. Words are encoded into numerical vectors and the feature similarity is considered for computing similarities among them. The KNN algorithm is modified into the version where the feature similarity is computed as well as the feature value one. The modified version is applied to the topic based word categorization. In this section, we describe what is proposed in this research.

Let us explain the process of encoding a word into a numerical vector. The texts in the corpus are given as feature candidates and among them, some are selected by their coverage to given words. The word is given as the input and its TF-IDF (Term Frequency-Inverse Document Frequency) weights or its frequencies to texts which are given as features are computed as the feature values which indicate relationship of the word with the texts. In other words, the word is represented into a numerical vector which consists of weights and frequencies. Numerical vectors which represent words or texts, tend to have their sparse distribution where zero values are dominant.

Figure 1 illustrates the outline of computing the proposed similarity metric between two numerical vectors. \(d_1, d_2, \ldots, d_n\) are the text identifiers which are selected from the corpus as features and the two words, \(t_1\) and \(t_2\) are represented into the two numerical vectors: \(t_1 = [w_{11}, w_{12}, \ldots, w_{1n}]\) and \(t_2 = [w_{21}, w_{22}, \ldots, w_{2n}]\). The similarity between the two features, \(d_i\) and \(d_j\), is computed by Equation (1)

\[
s_{ij} = \text{sim}(d_i, d_j) = \frac{2 \times t(f(d_i), d_j)}{t(f(d_i)) + t(f(d_j))}
\]  

where \(t(f(d_i), d_j)\) is the number of words which are shared by the two texts, \(d_1\) and \(d_2\), and \(t(f(d_i))\) is the number of words which are included in the text, \(d_i\). The similarity between the two numerical vectors which considers the feature similarity, \(s_{ij}\), is computed by Equation (2),

\[
sim(t_1, t_2) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} w_{1i} w_{2j}}{||t_1|| ||t_2||}
\]

where \(||t_1|| = \sqrt{\sum_{i=1}^{n} w_{1i}^2}\) and \(||t_2|| = \sqrt{\sum_{i=1}^{n} w_{2i}^2}\). The complexity in computing the proposed similarity metric is quadratic to the dimension, \(n, O(n^2)\).

![Figure 1. The Combination of Feature and Feature Value Similarity](image)

Figure 2 presents the proposed KNN algorithm. In advance, the training words are encoded into numerical vectors. A novice word is encoded into a numerical vector,
its similarities with the numerical vectors which represent the training ones by Equation (2), 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.

\[ \left[ a_{i_1}, \ldots, a_{i_N} \right] + \left[ a'_{i_1}, \ldots, a'_{i_N} \right] \]

\[ \left[ a_{i_1}, \ldots, a_{i_N} \right] - \left[ a'_{i_1}, \ldots, a'_{i_N} \right] \]

Figure 2. The Proposed Version of KNN

Let us make some remarks on what is proposed in this research. Even if the KNN is a very simple machine learning algorithm, it is useful for implementing a light version of classification system. Even if it takes much time for computing the proposed similarity metric, it tackled against the poor discriminations from the sparse distribution of numerical vectors. We may use words which are called contexts as features for representing words into numerical vectors, as well as text identifiers. The proposed KNN is described in more detail in [16].

IV. EXPERIMENTS

This section is concerned with one more set of experiments where the better performance of the proposed version is validated empirically on the text collection: 20NewsGroups I. In this set of experiments, we predefined the four general categories, and gather words from the collection category by category as the classified ones. Each word is classified exclusively into one of the four categories. We apply the KNN algorithms directly to the given task without decomposing it into binary classification, and use the accuracy as the evaluation measure. Therefore, in this section, we observe the performance of the both versions of KNN algorithm, with the different input sizes.

In Table I, we specify the general version of 20NewsGroups which is used for evaluating the two versions of KNN algorithm. In 20NewsGroup, the hierarchical classification system is defined with the two levels; in the first level, the six categories, alt, comp, rec, sci, talk, and misc, are defined, and among them, the four categories are selected, as shown in Table I. In each category, we select 1000 texts at random, and extract 375 important words from them as the labeled words. The 375 words are partitioned into the 300 words as the training examples and the 75 words as the test ones, as shown in Table I. In the process of gathering the classified words, they are selected by their frequencies which are concentrated in their corresponding categories.

<table>
<thead>
<tr>
<th>Category</th>
<th>#Texts</th>
<th>#Training Words</th>
<th>#Test Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comp</td>
<td>1000</td>
<td>300</td>
<td>75</td>
</tr>
<tr>
<td>Rec</td>
<td>1000</td>
<td>300</td>
<td>75</td>
</tr>
<tr>
<td>Sci</td>
<td>1000</td>
<td>300</td>
<td>75</td>
</tr>
<tr>
<td>Talk</td>
<td>1000</td>
<td>300</td>
<td>75</td>
</tr>
<tr>
<td>Total</td>
<td>4000</td>
<td>1200</td>
<td>300</td>
</tr>
</tbody>
</table>

The experimental process is identical to that in the previous sets of experiments. In each category, we extract the 375 important words and encode them into numerical vectors with the input sizes, 10, 50, 100, and 200. For each test example, we compute its similarities with the 1200 training examples, and select the three similar ones as its nearest neighbors. The versions of KNN algorithm classify each of the 300 test examples into one of the four categories: comp, rec, sci, and talk, by voting the labels of its nearest neighbors. We also use the classification accuracy as the evaluation measure in this set of experiments.

In Figure 3, we illustrate the experimental results from categorizing words using the both versions of 20NewsGroups. Figure 3 has the identical frame of presenting the results to those of Figure 1 and 2. In each group, the gray bar and the black bar indicates the achievements of the traditional version and the proposed version of KNN algorithm, respectively. The performance is expressed as the accuracy of classifying words into one of the four categories. In this set of experiments, the classification task is not decomposed into binary classifications.
shows its better performances in the three of the four cases. In the input size, 200, it is competitive with the traditional version. From this set of experiments, we conclude that the proposed version wins over the traditional one, in averaging over their four achievements.

V. CONCLUSION

Let us mention the remaining tasks for doing the further research. We need to validate the proposed approach in specific domains such as medicine, engineering, and economics, as well as in generic domains such as ones of news articles. We may consider the computation of similarities among some main features rather than among all features for reducing the computation time. We try to modify other machine learning algorithms such as Naive Bayes, Perceptrons, and SVM (Support Vector Machine) based on both kinds of similarities. By adopting the proposed approach, we may implement the word categorization system as a real program.

REFERENCES


