A STUDY ON TEXT CATEGORIZATION BY USING GRAPH BASED K NEAREST NEIGHBORS APPROACH

[1]Alekhya Bonam, [2]Srinivasa Rao Manchem
[1]Student, [2]Asst. Professor
[1]Department of CSE, Adikavi Nannaya University,
[1] Department of CSE Adikavi Nannaya University, Rajamahendravaram, India

Abstract : Text Categorization is the activity of labeling natural language texts with relevant categories from a predefined set. In this method, a graph is given as input, instead of a numerical vector. In this regard, encode texts into graphs, define the similarity measure between graphs, and modify the K Nearest Neighbor into its graph based version as the text categorization tool.

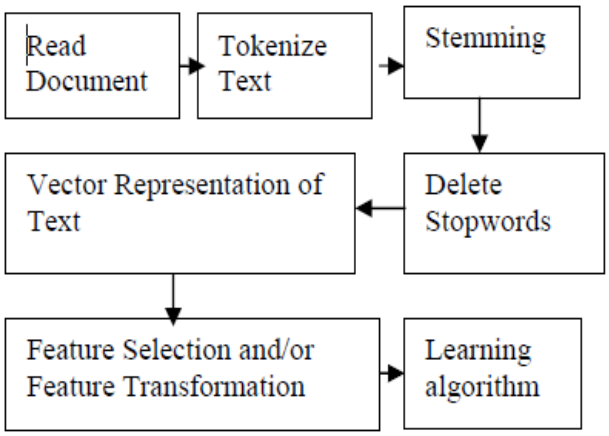
IndexTerms **- Text categorization, Graph, KNN, Text mining**

I. INTRODUCTION

Text mining is the process of examining large collections of written resources to generate new information, and to transform the unstructured text into structured data for further analysis. Text mining identifies facts, relationships and assertions that would otherwise remain buried in the mass of textual big data. These facts are extracted and turned into structured data, for analysis, visualization (e.g. via html tables, mind maps, charts), integration with structured data in databases or warehouses, and further refinement using machine learning (ML) systems.

Text Categorization is the activity of labeling natural language texts with relevant categories from a predefined set. In laymen terms, text classification is a process of extracting generic tags from unstructured text. Text categorization refers to the process of classifying each text into its relevant topics or categories among the predefined ones. As its preliminary tasks, a finite number of categories are predefined and sample texts which are labeled with one or some of the predefined are prepared. As the learning process, using the sample labeled texts, the classification capacity is constructed. Subsequent texts which are given as ones separated from the sample labeled texts are classified as the generalization process. Even if other kinds of approaches such as manual rule based schemes and other heuristic ones are available, in this paper, we assume that the supervised learning algorithms are used as the approach. In general, text classification includes topic based text classification and text genre-based classification. Topic-based text categorization classifies documents according to their topics [33]. Texts can also be written in many genres, for instance: scientific articles, news reports, movie reviews, and advertisements. Genre is defined on the way a text was created, the way it was edited, the register of language it uses, and the kind of audience to whom it is addressed.

The process of text classification as seen from the point of view of automatic text Classification system is as follows.



II. LITERATURE SURVEY

With the rapid growth of online information, text categorization has become most important the key techniques for handling and organizing text data. It is important to apply information processing method. Because most of the internet information is stored in the form of text in various servers' hard disks or in databases, and most of them is semi-structured text data; text categorization has become a hotspot in the research of the modern information processing. Text categorization is defined as determining a category for each document in a collection of documents according to the predefined topic category. Through text categorization,

text can be classified, and thereby the efficiency of information search by end users can be greatly improved. Until now, there are a number of text categorization models have been proposed such as k neighbor algorithm, naive Bayesian algorithm, support vector machine (SVM) algorithm, neural network and decision tree algorithm. Among them, K-NNG algorithm is a famous integrated learning algorithm by taking the decision tree as basic categorizer. The algorithm does not require a priori knowledge, and has high categorization accuracy without over-fitting problem.

The specific details of The K-Nearest Neighbor Graph (K-NNG) for a set of objects V is a directed graph with vertex set V and an edge from each v ∈ V to its K most similar objects in V. K-NNG construction is an important operation with many web related applications: in (user-based) collaborative filtering, a KNNG is constructed by connecting users with similar rating patterns, and used to make recommendations based on the active user's graph neighbors. In content-based search systems, when the dataset is fixed, a K-NNG constructed offline is more desirable than the costly online K-NN search. K-NNG is also a key data structure for many established methods in data mining and machine learning especially manifold learning. Furthermore, an efficient K-NNG construction method will enable the application of a large pool of existing graph and network analysis methods to datasets without an explicit graph structure.

K-NNG construction by brute-force has cost O (n2) and is only practical for small datasets. Substantial effort has been devoted in research related to K-NNG construction and KNN search, and numerous methods have been developed, but existing methods either do not scale, or are specific to certain similarity measures. Paredes et al. [ref19] proposed two methods for K-NNG construction in general metric spaces with low empirical complexity, but both require a global data structure and are hard to parallelize across machines. Efficient methods for l2 distance have been developed based on recursive data partitioning and space filling curves, but they do not naturally generalize to other distance metrics or general similarity measures.

Indexing data for K-NN search is a closely related open problem that has been extensively studied. A K-NNG can be constructed simply by repetitively invoking K-NN search for each object in the dataset. Various tree-based data structures are designed for both general metric space and Euclidean space. However, they all have the scalability problem mentioned above. Locality Sensitive Hashing (LSH) is a promising method for approximate KNN search. Such hash functions have been designed for a range of different similarity measures, including hamming distance, lp with p, cosine similarity, etc. However, the computational cost remains high for achieving accurate approximation, and designing an effective hash function for a new similarity measure is non-trivial.

III. PROPOSED SYSTEM

In this paper we suggest three mandatory steps includes

Encoding texts into graphs.

Compute similarities between two graphs

Implementation of KNNG (K Nearest Neighbour- Graph based).

3.1 Text Encoding

Graph divided into two sets i.e.,vertex set and edge set. In the graph vertices taken from words and identified edges based on their semantic relationships.

Here graph is defined as weighted and undirected

In this process, first construct an index list where each text is linked to a list of words. Next, Generate list of texts from Corpus and each text is indexed into a list of words.

Later each word has its weights and posting information its relationship with a text.

Text can be indexed as follows.

i)tokenization ii)stemming iii)stop word removal

In the Vertex set contains list of words included in the text (or)weights of words are as follows:

D(v)={Vi1,Vi2,……Vim}

Fixed number of texts is selected by ranking them and score is based on the text weights which are greater than or equal to threshold. A set of vertices which indicate words is extracted through indexing. Edges are getting by compute similarities of all possible pairs of vertices/words. Construct similarity matrix that identified entries are similarities among words from corpus. Select word pairs whose similarities are more than threshold

D(e)={ei1,ei2,…….,ein}

The word similarities were identified in the following possible ways.

i) Adjacency Matrix : Vertices → rows, Edges→ columns and entries.

ii) Linked List: Vertices→Nodes, Edges→pointers.

iii) Graph: List of edges → pairs of vertex identifiers.

3.2 Similarity Matrix

Consider rows and columns in a word corpus. Identify similarities of all possible words based on normalized values between 0 & 1. Here we can take NxN square matrix and diagonal elements as 1. Each entry of similarity matrix is identified as similar between two corresponding words.

The following are the similarity function as defined as

$$\text{Similarity(cm,cn)} = \frac{2|Cn \cap Cm|}{|Cn|+|Cm|} \qquad (1)$$

Equation.1 contains Cn and Cm are set of texts |cn|,|cm|

The result of similarity(cm,cn) will be 1,it indicates those two documents are similar or otherwise it is 0,indicates not similar.

Equation.1 , can generate a NxN square matrix as Tnm=similarity(cm,cn) of sample matrix

$$T=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} \quad (2)$$

Expression.2 indicates an example, T is a symmetry matrix
This similarity matrix is constructed automatically from a corpus.
Similarities between Graphs:
It is computed by averaging similarities among edges.
Two graphs G1 and G2 are expressed as
$G1=\{e_{11},e_{12},\ldots\ldots\ldots\ldots\ldots,e_{1i}\}$
$G2=\{e_{21},e_{22},\ldots\ldots\ldots\ldots\ldots,e_{2i}\}$
Now Similarities between edges as follows equations solves :

$$\text{Similarity}(G1,G2)=\frac{1}{i}\sum_{n=0}^{i}\text{similarity}(e_{1n}+G_2) \quad \text{(eqn.2)}$$

$$\text{Similarity}(e_{1n},G2)=\max\xrightarrow[\text{m=1 to p}]{}\text{similarity}(e_{1n}+e_{2m}) \text{ (eqn.3)}$$

$$\text{Similarity}(e_{1n},e_{2n},)=\frac{1}{2}(w(e_{1n})+w(e_{2n}))\text{(eqn.4)}$$

3.3 Graph vector based KNN

This section is concerned with the proposed KNN version as the approach to the text categorization. Words are encoded into graphs by the process which was described above. In this section, we attempt to the traditional KNN into the version where a graph is given as the input data. The version is intended to improve the classification performance by avoiding problems from encoding texts into numerical vectors. Therefore, in this section, we describe the proposed KNN version in detail, together with the traditional version. The sample words which are labeled with the positive class or the negative class are encoded into numerical vectors. The similarities of the numerical vector which represents a novice word with those representing sample words are computed using the Euclidean distance or the cosine similarity. The k most similar sample words are selected as the k nearest neighbors and the label of the novice entity is decided by voting their labels. However, note that the traditional KNN version is very fragile in computing the similarity between very sparse numerical vectors.

IV. CONCLUSION

Let us mention the remaining tasks for doing the further research. We apply and validate the proposed research in classifying technical documents in specific domains such as medicine or engineering rather than news articles in various domains. We define and characterize more advanced operations mathematically on graphs which represent texts. We modify more advanced machine learning algorithms into their graph based version, using the more sophisticated operations. We implement the text categorization system as a system module or an independent software by adopting the proposed approach.

V. REFERENCES

1. Text Classification Using Machine Learning Techniques by M. Ikonomakis, V. Tampakas, S. Kotsiantis.
2. Machine learning approach for text and document mining by Vishwanath Bijalwan, Pinki Kumari, Jordan Pascual and Vijay Bhaskar Semwal.
3. Graph based KNN for Text Categorization by Taeho Jo
4. Text Classification Method Review by Aigars Mahinovs and Ashutosh Tiwari
5. Algorithms for Graph Similarity and Subgraph Matching by Danai Koutra, Ankur Parikh
6. Measuring Similarity between Graphs Based on the Levenshtein Distance by Bin Cao, Ying Li and Jianwei Yin