CLASSIFICATION OF TEXT DATA

Raxmanov Asqar Tajibaevich,
Associate professor at Tashkent University of Information Technologies
e-mail: asqartr1.2.3dipu@gmail.com,

Abduvaliyeva Zebiniso Abdulxamidovna,
Assistant at Tashkent University of Information Technologies
e-mail: zebinisoabduvaliyeva@gmail.com
tel.: +998932327355

Abstract. This article discusses the TF-IDF modeling method for converting given text data into a vector quantity for further classification. Using the TF-IDF modeling method, a cosine classifier for text data is obtained. This classifier can be used not only for classifying text data but also for classifying arbitrary vector data. The use of the obtained modeling and classification results with other methods allows determining the most effective classification method.

Keywords. *Classification, vector, nearest neighbor method, Word2Vec, SVM, Apollon's sphere, BOW, TF-IDF.*

INTRODUCTION

Text classification, as a fundamental problem in natural language processing, has been studied by many authors [1-4]. Currently, text classification algorithms have become more efficient, but there is still much room for improvement in the study of classification and text recognition. Therefore, the research on mathematical modeling of text data and their subsequent classification has received significant attention and is a pressing issue.

The task of text classification is formed as a process of assigning one or more text data, such as articles, reports, messages, etc., to one or more classes. This process involves analyzing the content of the texts and using various methods to determine their affiliation with predefined classes. The main goal of the task is to find an algorithm that can determine the class of a new text.

This approach is used in text classification, including the task of automatic document type determination and other types of text processing. The main task of classification is to develop a model that recognizes text by its features.

Feature formation in text data processing plays a crucial role, as it determines which characteristics of the text will be used for analysis and model building. To apply mathematical classification methods, we need to convert the text into a vector suitable for computer processing in advance. One of the approaches to forming a text model is its features. The features of text data include word frequency in a sentence or document. By frequency, we mean the total number of occurrences of a word in the text. The length of the text in a sentence or document should be considered as the total number of all words or characters used to build the given text. The number of unique words in a sentence is an estimate of the diversity of the text. When converting text into a vector, it is necessary to consider the exclusion of so-called stop words, which are words without special meaning. To form numerical features from text, text processing methods can be used, which convert text data into a vector for subsequent analysis or application of mathematical methods.

METHODS OF TEXT DATA PROCESSING

Text data processing consists of three consecutive stages: error correction, lexical normalization of words, and reduction to the base form. Correcting grammatical and lexical errors is a complex procedure that requires conducting extensive research. It is necessary to consider not only the rules of the language in question but also numerous exceptions. One of the most difficult tasks related to natural language processing is understanding the semantics of words, specifically forming features so that the algorithm can distinguish concepts rather than sets of letters.

Reduction to the base form is the process of finding the base of a word for a given original word. The base of a word does not necessarily coincide with its morphological root. Lexical normalization of a word is the process of determining the lemma of a word, i.e., its basic form. Lexical normalization of a word is a more complex procedure than reduction to the base form, as identifying the lexical base of a word must be based on context.

When converting words into vectors, there are the following methods: "Bag of Words (BoW)". Each unique word in the text is assigned a unique index, and the text is converted into a vector reflecting word frequencies. TF-IDF (Term Frequency-Inverse Document Frequency) is a method based on the frequency of word occurrences in the text and their occurrence in other texts. Vector representations of words (Word Embeddings, such as Word2Vec, GloVe) convert text into numerical vectors, taking into account semantic information.

The TF-IDF Algorithm. Let us delve into the TF-IDF algorithm (Term Frequency-Inverse Document Frequency). The TF-IDF algorithm is a statistical measure used to evaluate the importance of a word in a text relative to a corpus of documents. It consists of two components: a) TF (Term Frequency), which is the ratio of the frequency of a term or word to the total number of words in a document; b) IDF (Inverse Document Frequency), indicating how rarely a word occurs in the corpus of documents.

The TF-IDF algorithm is a method for evaluating the importance of a word in the context of a text or document based on its frequency of occurrence in other documents. This algorithm is widely used in text processing tasks such as text classification, information retrieval, and building search systems. The algorithm consists of two main components:

By definition, the TF(t, d) value approaches unity if the word d occurs very frequently in document d and approaches zero if the word d occurs very rarely in document d. TF is calculated using a simple formula:

$$TF(t, d) = \frac{\text{"The number of occurrences of word t in document d."}}{\text{"The total number of words in document d."}} \quad (1)$$

IDF(t, n) −the Inverse Document Frequency (IDF) is a measure of how rare or unique the term t is within a collection of n documents. It is calculated as:

$$DF(t, n) = \lg\left(\frac{n}{m(t)}\right), \quad (2)$$

where n is the total number of documents in the corpus, and $m(t)$ is the number of documents in which the term t appears. From Equation (2), it is evident that the rarer the word in the documents, the higher its IDF value, with the upper bound $IDF(t, n) \leq \lg(n)$.

The TF-IDF, a combined measure of the importance of a word in a document, is obtained by multiplying the TF(t,d) and IDF(t,n) coefficients:

$$TF - IDF(t, d, n) = TF(t, d) * IDF(t, n). \quad (3)$$

For each term in a document, TF-IDF is calculated, reflecting its significance for a particular document in the context of the entire collection. TF-IDF is employed in search engines to find the most relevant documents, as well as for text classification, where it is often used as a feature in machine learning algorithms. In text analysis, TF-IDF is applied to highlight the most important words in the text, i.e., for keyword extraction. The algorithm is an excellent tool for the practical processing of large volumes of textual data.

Consider the example of converting the sentence "I love programming " into a vector using TF-IDF within a corpus of the following three documents:

1. I love programming.
2. Machine learning in programming.
3. Classification using machine learning.

The sequence of the following 9 words is considered:

"I", "love", "programming" , "machine", "learning", "in", "classification" , "with" , " help".

Now, we calculate TF for each word in each document; that is, for each word in the document, we calculate its frequency of occurrence in the document and divide it by the total number of words in the document. For document 1 ("I love programming "), the total number of words in the document is 3. The word "I" occurs once; therefore, TF("I",1)=1/3. The word "love" occurs once, TF("love",1)=1/3. The word "programming" occurs once, TF("programming ",1)=1/3, and so on for all the other words, we have: TF("machine", 2)=1/4, TF("в",2)=1/4, TF("learning",2)=1/4, TF("programming",2)=1/4, TF("classification",3)=1/5, TF("with",3)=1/5, TF("help ",3)=1/5, TF("machine ",3)=1/5, TF("learning",3)=1/5.

After, we perform the IDF calculation for each word. The total number of documents n=3. Using the IDF formula (2), for each word in the corpus, we calculate the number of documents in which it occurs. The word "I" occurs in one document; therefore,

$IDF(\text{"I"}, n) = \lg\left(\frac{n}{m(t)}\right) = \lg\left(\frac{3}{1}\right)$ $=\lg 3 \approx 0{,}477$. The word "love" occurs in one document; therefore, $IDF(\text{"love"}, n) \approx 0{,}477$. The word "programming" (programming) occurs in two documents; therefore, $IDF(\text{"programming"}, n) = \lg\left(\frac{n}{m(t)}\right) = \lg\left(\frac{3}{2}\right) \approx 0{,}176$, and so on for all words. $IDF(\text{"machine"}, n) \approx 0{,}176$,, $IDF(\text{"learning"}, n) = \lg\left(\frac{3}{2}\right) \approx 0{,}176$, $IDF(\text{"в"}, n) \approx 0{,}477$, $IDF(\text{"classification"}, n) \approx 0{,}477$, $IDF(\text{"c"}, n) \approx 0{,}477$, $IDF(\text{"help "}, n) \approx 0{,}477$. Now, we calculate TF-IDF by multiplying the TF coefficient by the corresponding IDF coefficient using formula (3). $TF - IDF(\text{"I"}, 1{,}3) = TF(\text{"I"}, 1) * IDF(\text{"I"}, 3) \approx \frac{1}{3} * 0477 = 0{,}159$. $TF - IDF(\text{”love”}, 1{,}3) = TF(\text{”love”}, 1) * IDF(\text{”love”}, 3) \approx \frac{1}{3} * 0{,}477 = 0{,}159$, $TF - IDF(\text{"programming"}, 1{,}3)\, TF(\text{programming"}, 1)\, IDF(\text{"programming"}, 3) \approx \frac{1}{3} * 0{,}176 = 0{,}059$, $TF - IDF(\text{"machine"}, 1{,}3) = TF(\text{"machine"}, 1) * IDF(\text{"machine"}, 3) \approx \frac{1}{4}\ 0{,}176 = 0{,}044$, $TF - IDF(\text{"learning"}, 1{,}3) = TF(\text{"learning"}, 1) * IDF(\text{"learning"}, 3) \approx \frac{1}{4}\ 0{,}176 = 0{,}044$,

$TF - IDF(\text{"в"}, 1{,}3) = TF(\text{"в"}, 1) * IDF(\text{"в"}, 3) \approx \frac{1}{4} * 0{,}477 = 0{,}119$, $TF - IDF(\text{"programming"}, 1{,}3) = TF(\text{"programming"}, 1) * IDF(\text{"programming"}, 3) \approx \frac{1}{4} * 0{,}176 = 0{,}044$

$$TF - IDF(\text{"classification"}, 1{,}3) TF(\text{"classification"}, 1)\ \ IDF(\text{classification}, 3) \approx \frac{1}{5} * 0{,}477 = 0{,}095,$$

$$TF - IDF(\text{"c"}, 1{,}3) = TF(\text{"c"}, 1) * IDF(\text{"c"}, 3) \approx \frac{1}{5} * 0{,}477 = 0{,}095,$$

$$TF - IDF(\text{»help»}, 1{,}3) = TF(\text{»help»}, 1) * IDF(\text{»help»}, 3) \approx \frac{1}{5} * 0{,}477 = 0{,}095$$

$TF - IDF(\text{"}\ of\ \text{machine"}, 1{,}3)\, TF(\text{"}of\ \text{machine"}, 1) * IDF(\text{"}of\ \text{machine"}, 3) \approx \frac{1}{5} * 0{,}176 = 0{,}035$,

$TF - IDF(\text{"learning"}, 1{,}3) = TF(\text{"learning"}, 1) * IDF(\text{"learning"}, 3) \approx \frac{1}{5} * 0{,}176 = 0{,}035$.

Now, for this corpus, considering the above calculations, we can write the coordinates of the vector for the sentence " I love programming ". It will look like this: " I "— 0.159, " love " — 0.159, " programming— 0.059, " machine " — 0, " learning " — 0, " in "— 0, " programming "— 0, " classification " — 0, " with " () — 0, " help " (help) — 0, " *of* machine" (machine) – 0, " learning " - 0; that is, the first three coordinates of this vector which correspond to the important terms in the sentence are non-zero, and the rest are zero. Exactly the same vectors with corresponding changes can be written for the other two sentences of this corpus. Then, the three sentences of the corpus correspond to the following vectors:

A1=[0.159, 0.159, 0.059, 0, 0, 0, 0, 0, 0, 0, 0, 0],

A2=[0, 0, 0, 0.044, 0.044, 0.119, 0.044, 0, 0, 0, 0, 0],

A3=[0, 0, 0, 0, 0, 0, 0, 0, 0.095, 0, 0.095, 0, 0.095, 0, 0.035, 0, 0.035]....

It follows that the vector will have several non-zero values that correspond to important terms in the sentence. Each word is represented as a vector, where semantically similar words have similar vectors.

Text Classification. This $TF-IDF$ method is one way to translate text into a vector, converting text into a digital format that implements work with text data for analysis, classification, understanding text, and other tasks. The subsequent result and effectiveness of the models depend on the quality of data processing. We will use this method for classifying text documents. Furthermore, our main task is to determine a classifier using the TF-IDF method that will effectively predict the label y_i (the class of text data c_i)) for a new text t_{new}. Mathematically, the problem can be represented as follows: given a set of texts $T=\{t_1,t_2,t_3,\dots,t_n\}$,, where each text t_i is a sequence of words, sentences, or other units of text. For each text t_i, there is a label y_i that belongs to one of the classes of the set $C=\{c_1,c_2,c_3,\dots.c_k\}$, where k is the number of classes. The classification task is to predict its class $y_{new}$for a new text $\mathrm{t_{new}}$. Consider multi-class classification, in which each text belongs to only one class.

It is known that for the scalar product of vectors A and B the following formula holds:

$$(\mathrm{A}\cdot\mathrm{B}) = |\mathrm{A}|\cdot|\mathrm{B}|\cos(\mathrm{A},\mathrm{B}) \qquad (4)$$

$(\mathrm{A}\cdot\mathrm{B})$ is the scalar product, (A,B)-is the angle between vectors A, B, $|\mathrm{A}|, |\mathrm{B}|$ −are the lengths of vectors A, B respectively. From (4), the angle between vectors A, B can be determined:

$$\cos(\mathrm{A},\mathrm{B}) = (\mathrm{A}\cdot\mathrm{B})/(|\mathrm{A}|\cdot|\mathrm{B}|) \qquad (5)$$

If the angle between vectors is acute, then they have almost the same direction. The direction of a vector, to some extent, indicates their similarity. Therefore, this fact can be used to measure the similarity of vectors; that is, if there are two vectors $A\ and\ B$ and the angle between them is acute, then the direction of vectors $A\ and\ B$ can be considered close, although their lengths may differ significantly. From the point of view of the meaning of the text, the length of the vector obtained from the transformation of text into a vector is not important. Therefore, formula (5) can be used as a definition of the similarity of vectors $A\ and\ B$ (or the corresponding texts). Thus, using (5) it can be stated that, if $\cos(A,B)$ ≥∝, where ∝ - is a sufficiently small positive number, dependent on the training sample, then the direction of vectors A and B are close and the texts corresponding to them are similar. Thus, the following inequality

$$\cos(A,B) \geqq \propto \qquad (6)$$

can be used as a classifier of text data. Since the resulting formula (6) does not directly depend on the text, it can be used not only for the classification of text data, but also for the classification of arbitrary vector data in which the angle between object-vectors can be determined.

From the example above it follows that $TF-IDF$ is one of the good methods in the field of Natural Language Processing (NLP) and recognition of text documents. This method allows us to evaluate the participation and importance of terms in a document relative to the entire corpus of texts, as well as to determine the keywords of a certain class. In the future, we will apply the TF-IDF method and its classifier (6) to the study of documents of various enterprises.

REFERENCES:

1. Rakhmanov A.T., Abduvalieva Z.A. Management of Document Flow Based on Mathematical Modeling. Muhammad al-Khwarizmiy avlodlari Scientific and Analytical Journal. ISSN-2181-9211.3(29)/2024 C 224-227.

2. Abduvalieva Z.A , Marisheva L.T, Latipova N.X, Sheyna N.E. Structure and functional features of document management systems on the example of the department. Journal of Northeastern University, Volume 25 Issue 04, 2022.

3. Marysheva L.T. Rakhmonov A.T. Latipova N.Kh. Abduvalieva Z.A. METHODS FOR IMPLEMENTING DOCUMENT FLOW BASED ON THE USE. «Science and innovation» Научный республиканский журнал. №MG-2024-06-7438

4. Isaeva, M., Yoon, H., Y.: Paperless university — How we can make it work?. In: 15th International Conference on Information Technology Based Higher Education and Training (ITHET). pp. 1–8 (2016). Luo, H., Fan, Y., Wu, C.: Overview of Workflow Technology. J. Softw. 11, 78-82

5. Fan, Yusun: Base on Workflow Management Technology. Beijin:Tsinghua University Press, 32, (2001)

6. DerryJatnikaa, Moch Arif Bijaksanaa ,ArieArdiyanti Suryania. International Conferenceon,12-13September 2019 Word2Vec Model Analysis for Semantic Similarities in English Words. Computer Science and Computational Intelligence 2019 (ICCSCI)

7. Chen, Hong-na, Zu, Xu, Zhou, Feng: On the Developing Situation, Research Content and Trend of Workflow Technology. Journal of Chongqing Instiute of Technology. 20(2), 65-69 (2006)

8. Li, Zhao, Qing, Li, Farong, Zhong: A Visual Modeling Framework of Workflow Systems Based on CCS. Semantics, Knowledge and Grid. Fifth International Conference. pp. 200-207 (2009).