EasyChair Preprint

№ 11048

One-Hot Encoding and Bag-of-Words Methods in Processing the Uzbek Language Corpus Texts

Botir Elov, Shahlo Hamroyeva, Noila Matyakubova and Umidjon Yodgorov

October 9, 2023

Elov Botir Boltayevich
PhD, associate professor.
Tashkent State University of Uzbek
Language and Literature named
after Alisher Navoi.

E-mail: elov@navoiy-uni.uz
Hamroyeva Shahlo Mirdjonovna
DSc, associate professor.
Tashkent State University of Uzbek
Language and Literature named
after Alisher Navoi.
E-mail: *shaxlo.xamrayeva@navoiy-uni.uz*

Matyakubova Noila Shakirjanovna
PhD student of Tashkent State University
of Uzbek Language and Literature named
after Alisher Navoi.
E-mail*: nailya89mm@mail.ru*

Yodgorov Umidjon Saydilla o'g'li
A teacher of Tashkent State University
of Uzbek Language and Literature named
after Alisher Navoi
E-mail: *yodgorov@navoiy-uni.uz*

ONE-HOT ENCODING AND BAG-OF-WORDS METHODS IN PROCESSING THE UZBEK LANGUAGE CORPUS TEXTS

Abstract. Computers are designed to process information in digital or numerical form. But data is not always in numerical form. This article describes how to process data in the form of characters, words, and text, as well as the application of ONE-HOT ENCODING and BAG-OF-WORDS methods to the Uzbek language, among the methods of teaching a computer to process natural language. How do Alexa, Google Home, and many other "smart" assistants understand and respond to our speech today? This article presents the approaches of text processing of the Uzbek language corpus through text processing methods such as Bag-of-words (BOW), ONE-HOT encoding in the field of artificial intelligence called natural language processing.

Keywords: Uzbek language corpus, text processing, Bag-of-words (BOW), ONE-HOT encoding.

Introduction. Natural language processing is a subfield of artificial intelligence that helps machines understand and process human language. For most

natural language processing (NLP) tasks, the most basic step is to convert words into numbers to understand and decode patterns in natural language. In NLP, this stage is called text representation [1, 2, 3].

The “raw” text in the language corpus is pre-processed and converted into a suitable format for the machine learning model. Data is processed through tokenization, de-wording, punctuation removal, stemming, lemmatization, and a number of other primary processing NLP tasks (Figure 1). In this process, existing "noise" in the data is cleaned [4, 5, 6]. This cleaned data is presented in various forms (templates) according to the input requirements of the NLP application and machine learning model. Common terms used in text processing in NLP are:

Corpus (Corpus, C): a collection of data or multiple textual data together interpreted as a corpus.

Vocabulary (V): collection of all unique words in the corpus.

Document (D): A single text record of a dataset.

Word(Word, W): words in the dictionary.

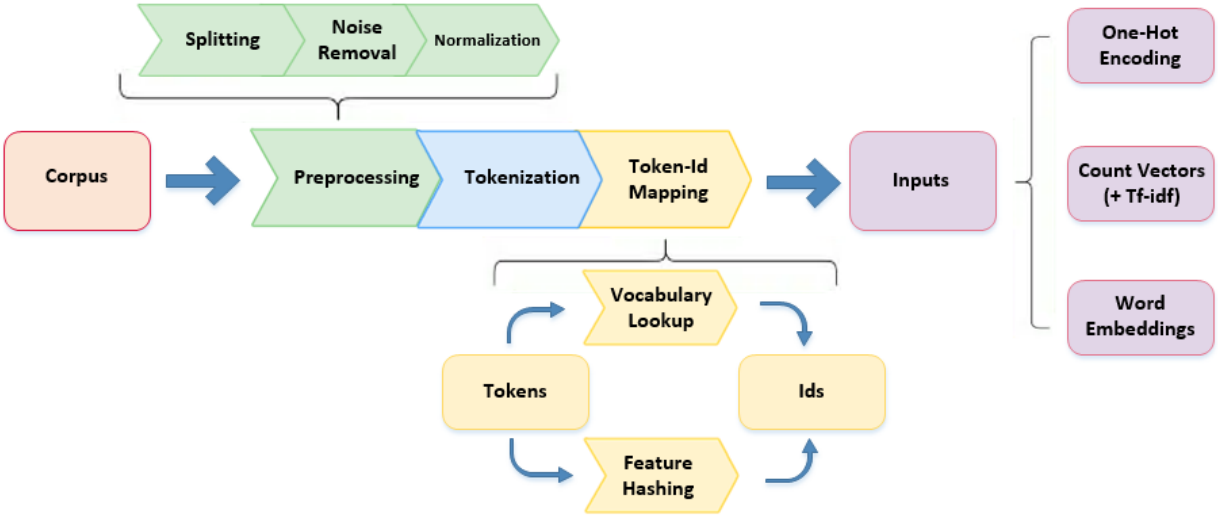

Figure 1. Stages of initial processing of language corpus texts

Figure 1 shows the process of converting the corpus matrix to different input formats for the ML model. Starting from the left, a corpus goes through several steps before obtaining tokens, a set of text building blocks, i.e. words, characters, etc. Since ML models are based on numerical value processing only, the tokens in the sentence are replaced by the corresponding numerical values. In the next step, they are converted to the various input formats shown on the right. Each of these formats has its pros and cons and should be chosen strategically based on the specifics of a given NLP task.

Types of text processing

Although the process of text processing is iterative, it plays an important role for a machine learning model/algorithm. Text views can be divided into two parts [7,8]:

1. Discrete text representations;
2. Distributed/Continuous text representations.

This article focuses on discrete text representations and introduces text processing methods using the Python package Sklearn.

Discrete views of text

In the discrete representation of corpus texts, words in the corpus are represented independently of each other. In this approach, words are represented by indexes corresponding to their position in the vocabulary of the corpus(s). Methods belonging to this category are listed below [1,3,7]:

- One-Hot encoding;
- Bag-of-words (BOW);
- CountVectorizer;
- TF-IDF
- Ngram.

One-Hot encoding method

In the One-Hot encoding method, a vector consisting of 0 and 1 is assigned to each word in the corpus [9]. In the coding of this method, only one element of the vector is assigned - 1, and all other elements - 0. This value represents the element category. The resulting digital vectors are called hot vectors in NLP, and a unique hot vector is assigned to each word in the corpus. This action allows the machine learning model to recognize each word individually by its vector. One-Hot encoding method can be useful when there is a categorical feature in the data set. For example: The vector values corresponding to the sentence I like to read are expressed corresponding to each word in the sentence as follows:

Men → [1 0 0 0], o‘qishni → [0 1 0 0], yaxshi → [0 0 1 0], ko‘raman → [0 0 0 1]
or,

$$\begin{matrix}\textbf{Men}: \\ \textbf{o‘qishni}: \\ \textbf{yaxshi}: \\ \textbf{ko′raman}:\end{matrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

In this case, the sentence is expressed numerically as follows:

sentence = [[1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]]

In One-Hot encoding, each bit represents a possible category, and if a given variable does not belong to more than one category, one bit is sufficient to represent it. By this method, the words "Men" and "men" are matched with different vectors. By applying lowercase to all words in word processing, it is possible to match the same vector to uppercase and lowercase letters. In this method, the size of the one-dimensional vector is equal to the size of the dictionary.

When a corpus is encoded using the One-Hot encoding method, each word or token in the dictionary is converted into a digital vector. So, sentences in the corpus, in turn, become a matrix of size (p, q). In this,

- "p" is the number of tokens in the sentence;
- "q" is the size of the dictionary.

The size of the digital vector corresponding to the word in the One-Hot encoding method is directly proportional to the dictionary size of the corpus. So, with the increase in the size of the case, the size of the vector also increases. This

method is not useful for large corpora, which may contain up to 100,000 or more unique words. We implement the One-Hot encoding method using the Sklearn package:

```
from sklearn.preprocessing import OneHotEncoder
import itertools
# 4 ta namunaviy hujjat
docs = ['Men NLP bilan ishlayman', 'NLP juda ajoyib texnologiya’,
'Tabiiy tilni qayta ishlash', 'Zamonaviy texnologiyalar bilan ishlash']
# hujjatlarni tokenlarga ajratish
tokens_docs = [doc.split(" ") for doc in docs]
# tokenlar ro‘yxatini umumlashtirish va so‘zni  identifikatoriga moslashtiradigan lug‘atni yaratish
all_tokens = itertools.chain.from_iterable(tokens_docs)
word_to_id = {token: idx for idx, token in enumerate(set(all_tokens))}
# tokenlar ro‘yxatini token-id ro‘yxatlariga aylantirish
token_ids = [[word_to_id[token] for token in tokens_doc] for tokens_doc in tokens_docs]
# token-id ro‘yxatlarini umumlashtirish
vec = OneHotEncoder(categories="auto")
X = vec.fit_transform(token_ids)
print(X.toarray())
```

_____

```
[[0. 0. 0. 1. 0. 0. 1. 0. 0. 1. 0. 0. 1. 0.]
 [0. 0. 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 1. 0. 0.]
 [0. 1. 0. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0.]]
```

We show the advantages and disadvantages of this method in the table below:

Advantages	Disadvantages
Easy to understand and implement	if the number of categories is very large, a large amount of memory is required
	the vector representation of words is orthogonal, and the relationship between different words cannot be determined
	the meaning of the word in the sentence cannot be determined
	a large number of computations are required to represent a high-dimensional sparse matrix

Bag-of-words method

In the bag-of-words method, words from the corpus are placed in a "bag of words" and the frequency of each word is calculated. In this method, word order or lexical information is not taken into account to represent the text. In algorithms based

on the BOW method, documents with similar words are returned as similar regardless of word placement.

The BOW method converts a text fragment into vectors of fixed length. Word frequency detection helps to compare documents. The BOW method can be used in a variety of NLP applications, such as thematic modeling, document classification, and email spam detection. Below is the BOW vector corresponding to 2 Uzbek sentences.

1-sentence	**2-sentence**
“Adirlar ham bahorda lola bilan go‘zal, chunki lola – bahorning erka guli”.	*“Lola ham shifokorlik kasbini tanladi”.*

	Adirlar	**bahorda**	**lola**	**go‘zal**	**bahorning**	**erka**	**guli**	**shifokorlik**	**kasbini**	**tanladi**
1-gap	1	1	2	1	1	1	1	0	0	0
2-gap	0	0	1	0	0	0	0	1	1	1

The article "Using bag of words algorithm in natural language processing" written by B.Elov, N.Khudaiberganov and Z.Khusainova presents methods of converting Uzbek texts into digital form using the BoW algorithm [10].

Conclusion. Through Discrete Text Representation methods, each word in the corpus is considered unique and converted into a numerical form based on the various methods discussed above. The article presents several advantages and disadvantages of the different methods. We summarize them as a whole. Methods that generate discrete numerical values of text are easy to understand, implement, and interpret. Discrete representations of text are widely used in classical machine learning techniques and deep learning applications to solve NLP tasks such as document similarity, sentiment classification, spam classification, and topic modeling.

References

1. Naseem, U., Razzak, I., Khan, S. K., & Prasad, M. (2021). A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *20*(5). https://doi.org/10.1145/3434237

2. Chai, C. P. (2023). Comparison of text preprocessing methods. *Natural Language Engineering*, *29*(3). https://doi.org/10.1017/S1351324922000213

3. Probierz, B., Hrabia, A., & Kozak, J. (2023). A New Method for Graph-Based Representation of Text in Natural Language Processing. *Electronics*, *12*(13). https://doi.org/10.3390/electronics12132846
4. B.ELov, E.Adalı, Sh.Khamroeva, O.Abdullayeva, Z.Xusainova, N.Xudayberganov (2023). The Problem of Pos Tagging and Stemming for Agglutinative Languages. *8 th International Conference on Computer Science and Engineering UBMK 2023, Mehmet Akif Ersoy University, Burdur – Turkey.*
5. B.ELov, Sh.Khamroeva, Z.Xusainova (2023). The pipeline processing of NLP. *E3S Web of Conferences 413, 03011, INTERAGROMASH 2023.* https://doi.org/10.1051/e3sconf/202341303011
6. B.Elov, Sh.Hamroyeva, X.Axmedova. Methods for creating a morphological analyzer. *14th International Conference on Intellegent Human Computer Interaction, IHCI 2022, 19-23 October 2022, Tashkent.* https://dx.doi.org/10.1007/978-3-031-27199-1_4
7. Siebers, P., Janiesch, C., & Zschech, P. (2022). A Survey of Text Representation Methods and Their Genealogy. *IEEE Access*, *10*. https://doi.org/10.1109/ACCESS.2022.3205719
8. B.Elov, Z.Xusainova, N.Xudayberganov. Tabiiy tilni qayta ishlashda Bag of Words algoritmidan foydalanish. *O‘zbekiston: til va madaniyat (Amaliy filologiya), 2022, 5(4).* http://aphil.tsuull.uz/index.php/language-and-culture/article/download/32/29
9. B.Elov, Z.Xusainova, N.Xudayberganov. O‘zbek tili korpusi matnlari uchun TF-IDF statistik ko‘rsatkichni hisoblash. *SCIENCE AND INNOVATION INTERNATIONAL SCIENTIFIC JOURNAL VOLUME 1 ISSUE 8 UIF-2022: 8.2 | ISSN: 2181-3337*
https://www.academia.edu/105829396/OZBEK_TILI_KORPUSI_MATNLARI_UCHUN_TF_IDF_STATISTIK_KORSATKICHNI_HISOBLASH
10. Fu, Y., & Yu, Y. (2020). Research on text representation method based on improved TF-IDF. *Journal of Physics: Conference Series*, *1486*(7). https://doi.org/10.1088/1742-6596/1486/7/072032