UBMK'23

Bildiriler Kitabı
Proceedings

8. Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı

8th International Conference on Computer Science and Engineering

13-14-15 Eylül (September) 2023 Burdur - Türkiye

Table of Content / İçindekiler

Pos Taging of Uzbek Text Using Hidden Markov Model

Elov Botir Boltayevich
Tashkent State University of Uzbek Language and Literature named Alisher Navo'i
Tashkent, Uzbekistan
elov@navoiy-uni.uz

Sirojiddinov Shuhrat Samariddinovich
Tashkent State University of Uzbek Language and Literature named Alisher Navo'i
Tashkent – Uzbekistan
rector_tsuull@navoiy-uni.uz

Khamroeva Shahlo Mirdjonovna
Tashkent State University of Uzbek Language and Literature named Alisher Navo'i
Tashkent, Uzbekistan
shaxlo.xamrayeva@navoiy-uni.uz

Eşref Adalı
Computer Eng.and Informatics Faculty
Istanbul Technical University
Istanbul – Türkiye
adali@itu.edu.tr

Xusainova Zilola Yuldashevna
Tashkent State University of Uzbek Language and Literature named Alisher Navo'i
Tashkent, Uzbekistan
xusainovazilola@navoiy-uni.uz

***Abstract*– Markov models are one of the most widely used machine learning methods for natural language processing. Markov chain and hidden Markov model is a stochastic (random) method used to model dynamic systems, and the current state of the system is predicted based on previous states. The Markov chain, which correctly generates a sequence of words in the generation of sentences, is widely used in NLP tasks. It is also used for identifying NERs in a sentence and POS tagging based on a hidden Markov model. Based on the Markov model, hidden tags are predicted based on the tagged words in the language corpus. This article presents methods and algorithms for automatic POS tagging of a given sentence based on the tagged Uzbek corpus using a hidden Markov model.**

Keywords: Parts of Speech Tagging, POS tagging, hidden Markov model, Markov chain, Hidden Markov Models, stochastic methods, NLP, word groups, homonymy resolution, transition probability, emission probability, Viterbi algorithm.

I. INTRODUCTION

The purpose of the science of linguistics is to describe and explain linguistic features that exist in oral and written speech in our environment. First of all, explaining the cognitive aspects of language acquisition, understanding and use of human beings, another task is related to our understanding of how language connects with the world. One of the important tasks of linguistics is to study how communication is carried out through the linguistic structure of the language. There is a set of rules that govern linguistic expressions to explore this task. Later, statistical linguistic models were created and effectively used in various fields of NLP. Although being useful in practice is not the same as developing a valid theory, the effectiveness of statistical linguistic models has demonstrated the validity of the original approach [1].

This article presents various solutions for applying Markov models to NLP tasks. A Markov model is a method that studies the probability of something happening in the future by analyzing known probabilities. Natural languages are probabilistic languages that depend on the order (sequence) of words and phrases to get meaning in the context, and it is possible to use stochastic models such as the Markov model.

Associating each word in a sentence with its corresponding POS (part of speechs) is called POS tagging or POS annotation in NLP. Phrases, morphological features or lexical tags can be processed as POS tags. In many cases, nouns, verbs, adjectives, and similar word groups are used as speech tags. For small language corpora, POS tagging can be done manually. However, since the POS tagging process in large language corpora is complex and requires many steps, today this process is performed by automatic means. [10]. POS tags provide important information about a word and its neighbors. The POS tagging process can be used in a variety of tasks such as NLP applications such as data acquisition, parsing, text-to-speech (TTS) and data retrieval, and linguistic research for corpora. [11]. Also, POS tagging is used as an intermediate (initial) step for syntactic parsing, semantic parsing, translation, and many other high-level NLP tasks. This makes POS tagging an important step for advanced NLP applications. This article presents the methods of implementing POS tagging in Uzbek texts using the Hidden Macro model. The article focuses on Markov models as a stochastic approach to text processing in NLP. Most NLP research uses supervised models with improved use of Markov models to reduce dependency on annotation tasks.

Powerful models such as HMM require very large amount of training data and provide less accuracy for unknown (untrained) word recognition. Most of the world's languages do not have enough resources (language corpus) to implement the computation to use for training such models. [12], [11]. In the process of developing NLP applications for such languages, many unknown words are encountered. This leads to low accuracy of the model. Nowadays, increasing accuracy is a pressing problem for low-resource languages. As of Dec. 2022, the tagged educational corpus of the Uzbek language developed by the team of authors contains more than 5,000,000,000 sentences and about 1,200,000 word forms, and the examples in the article are based on the sentences in this corpus [5].

II. MATERIAL AND METHODS

Hidden Markov model. A Markov chain is a model of a random process that represents the probabilities of sequences of random variables, commonly known as states. Each state can take values from a specific set. That is, it can be understood as the probability of the current state depending on the previous state. A Markov chain is used when it is necessary to calculate the probability of a sequence of observable events. However, in most cases, the chain is hidden or invisible, and each state randomly generates 1 out of every k observation that are visible to us. [10]. A situation means certain conditions at a certain time. Let us be given a sequence of variables for states $s_1, s_2, \ldots, s_n$. According to the assumption of the Markov model:

$$(s_i = a|s_1 \dots s_{i-1}) = (s_i = a|s_{i-1}) \quad (1)$$

A formal description of the Markov chain is given in Table I below:

TABLE I. MATHEMATICAL DESCRIPTION OF MARKOV CHAINS

$S = S_1, S_2, \dots, S_n$	A group of N states
$A = a_{11}, a_{12}, \dots, a_{n1}, \dots, a_{nn}$	A - probability transition matrix, it is each one aij – represents the probability of transition from state i to another state j. $\sum_{i=1}^{n} a_{ij} = 1, \quad \forall i$
$P = P_1, P_2, \dots, P_n$	The initial probability distribution for S cases. P_i represents the probability that the Markov chain starts at a given state i. $\sum_{i=1}^{n} p_i = 1$

When analyzing a text, the user usually does not distinguish the tags related to the word group, but infers about the tags from the sequence of words. We call such tags "hidden" because we do not track them directly. A formal description of the hidden Markov model is given in Table II below [15]:

TABLE II. MATHEMATICAL DESCRIPTION OF HMM

$S = S_1, S_2, \dots, S_n$	A group of N states
$A = a_{11}, a_{12}, \dots, a_{n1}, \dots, a_{nn}$	A is a transition matrix of probability, representing the probability of transition from each state $a_{ij} - i$ to another state j. $\sum_{i=1}^{n} a_{ij} = 1, \quad \forall i$
$O = O_1, O_2, \dots, O_n$	T is a sequence of observations (O), all of which are taken from a special dictionary (source). $V = V_1, V_2, \dots, V_t$
$B = b_i(O_i)$	A sequence of observation probabilities (called emission probabilities), all of which represent the probability that observation O_i will occur from state i
$P = P_1, P_2, \dots, P_n$	The initial probability distribution for S cases. P_i represents the probability that the Markov chain starts at a certain state i. $\sum_{i=1}^{n} p_i = 1$

III. REVIEWS

There are different approaches and methods to implement POS tagging [5], as shown in Fig. 1: rule-based, stochastic or statistical, hybrid [14]:

Rules-based POS tagging. A rule-based approach uses a dictionary or lexicon to match a word with a tag. This requires a large number of manual operations, such as dataset annotations or lexicon creation. For example, an adjective word is often followed by a noun word. In some cases, regular expression templates are used to define the keyword. Knowledge-based taggers, powered by expert-compiled vocabularies, provide highly accurate results. But there are a number of limitations to implementing rules-based POS tagging [15].

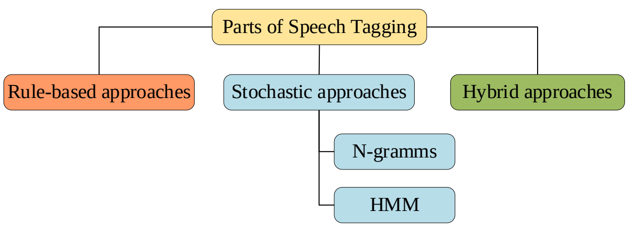

Fig. 1. Approaches of POS tagging

POS tagging using stochastic methods. With this method, the tagging process is performed automatically and does not require data encoding and dictionary creation. It uses statistics, frequency and probability to tag a word. In the process of tagging, the probability of a word being associated with a certain tag or the frequency of consecutive words is determined. In this approach, probabilities are calculated based on labeled data in the training corpus. Stochastically, POS tagging is performed using n-grams or hidden Markov models [11]. There are three common types of n-grams: unigram, bigram, and trigram The lists the n-grams corresponding to the sentence "I read a fiction book" is given in Table III:

TABLE III. N-GRAMS CORRESPONDING TO THE SENTENCE

N-gramm	words/word phrases
Unigram	Men badiiy kitobni o‘qidim
Bigram	Men badiiy badiiy kitobni kitobni o‘qidim
Trigram	Men badiiy kitobni badiiy kitobni o‘qidim

This approach uses a statistical model to calculate the probability for n-grams and assigns a tag corresponding to the specified n-grams. The probability of a unigram word is determined by the following equation:

$$P(g_i \mid d_i) = \frac{freq(d_i|g_i)}{freq\ (d_i)}$$

The probability of a word bigram is determined as:

$$P(g_i \mid d_i) = P(d_i \mid g_i) * P(g_i \mid g_{i-1})$$

The probability of a word trigram is determined as:

$$P(g_i \mid d_i) = P(d_i \mid g_i) * P(g_i \mid g_{i-2}, g_{i-1})$$

Here g is a label and w are a sequence of words. P(di | gi) represents the probability of the current word given the current tag, and P(gi | gi-1) represents the probability of the current tag given the previous tag [24]. In POS tagging, HMM associates each word in the text with a corresponding tag. POS tags are considered latent states and the HMM model tries to predict the tag based on the observed (tagged) words in the corpus. It is necessary to define gin in such a way that the following equality should hold:

$$\prod_{i=1}^{n} (d_i \mid g_i) * P(g_i \mid g_{i-1})$$

POS tagging based on hybrid methods. In the first step, the model is trained using statistical methods. Then, a rule-based approach is also implemented in order to improve the efficiency of the result [26.]. There are different methods that can be used for POS tagging of Uzbek language texts:

Rules-based POS tagging. Rule-based POS tagging models assign POS tags to words in text using a set of handwritten rules. These grammatical rules are often called context frame rules. We will give an example of such rules: "*Agar noaniq/noma'lum so‘z "-di" qo‘shimchasi bilan tugasa uni fe’l deb belgilang*".

Transformation based tagging. Transform-based approaches use a predefined set of manually defined rules as well as automatically applied rules developed during training.

Deep learning models. Different deep learning models are used for POS tags. For example, the Meta-BiLSTM model provides about 97 percent accuracy [27, 28].

Stochastic tagging. A stochastic approach involves frequency, probability, or statistics. The simplest stochastic approach is to identify the most frequently used tag for a given word in the training data and use this information to tag that word in plain text.

IV.RESULTS

POS Tagging via Hidden Markov Model.. HMMs can be applied to NLP applications such as speech, writing, gesture recognition, and bioinformatics. Let's consider an example proposed by Luis Serrano and choosing a suitable sequence of tags for text using a HMM as shown in Fig. 2:

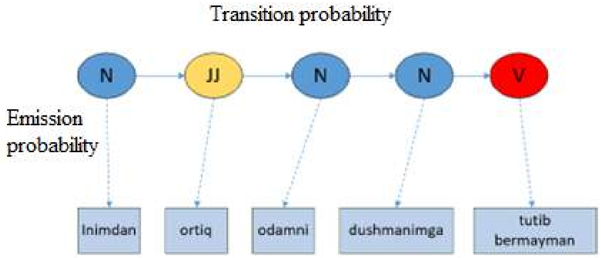

Fig. 2. The HMM method proposed by Luis Serrano

In this example, we are only looking at 3 POS tags, which are nouns, modals, and verbs. The sentence "*Nature is as beautiful as a bride*" is tagged as a noun, predicate, noun, and adjective, and it is necessary to calculate the transition probability and emission probability of this sequence. Transition probability is the probability of a given sequence, such as the probability that a noun is followed by a preposition, and a noun is followed by a preposition, and an adjective is followed by a noun. This type of probability is called transition probability. This probability value must be high for a particular sequence to be true. Now let's calculate the probability that the word "*tabiat*" is a noun, the word "*xuddi*" is a particle, the word "*kelinchak*" is a noun, and the word "*serbezak*" is an adjective. This set of probabilities is called emission probability.

We calculate the above two probabilities for the following set of sentences: *Nodir Zuhrani yaxshi ko'radi. Komil uzoqdan Nodirni ko'rdi. Zuhra Nodirni so'rab ko'rdimi? Zuhra Komilga nodir kitobni berdi.* Note that Nadir, Zuhra and Kamil are human names. We can see this in Fig. 3:

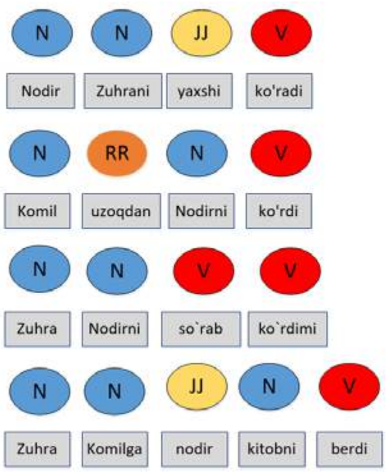

Fig. 3. Sentences tagged with POS

In the above sentences, the word "*Nodir*" appears three times as a noun and once as an adjective. To calculate the probability of emission, we form a calculation table in a similar way (as shown in Table IV):

TABLE. IV. TAG FREQUENCIES CORRESPONDING TO THE GIVEN SENTENCES

Words	Noun	Adjective	Adverb	Verb
Nodir	3	1	0	0
Zuhra	3	0	0	0
Komil	2	0	0	0
yaxshi	0	1	0	0
ko'radi/ ko'rdimi	0	0	0	3
uzoqdan	0	0	1	0
so'rab	0	0	0	1
kitobni	1	0	0	0
berdi	0	0	0	1

In the next step, we divide each column by their total number of occurrences. For example, "noun" occurs 9 times in the above sentences. After this operation, we create the following table (TABLE. V.):

TABLE. V. EMISSION PROBABILITY VALUES

Words	Noun (N)	Adjective (JJ)	Adverb(RR)	Verb (V)
Nodir	3/9	1/2	0	0
Zuhra	3/9	0	0	0
Komil	2/9	0	0	0
Yaxshi	0	1/2	0	0
ko'radi/ko'rdimi	0	0	0	3/5
Uzoqdan	0	0	1	0
so'radimi	0	0	0	1/5
Kitobni	1/9	0	0	0
Berdi	0	0	0	1/5

Based on the above tables, we come to the following conclusions:

– *The probability that the word* ***"Zuhra"*** *is a noun = 3/9*
– *The probability of the word* ***"Zuhra"*** *being ravish = 0*
– *The probability that the word* ***"Nodir"*** *is a noun = 3/9*
– *The probability that the word* ***"Nodir"*** *is an adjective = 1/2*

All remaining probabilities can also be determined by the above method. This is the emission probability. In the next step, it is necessary to calculate the transition probability. Therefore, we define two additional <S> and <E> tags. As shown in Fig. 4, <S> is placed at the beginning of each sentence and <E> at the end:

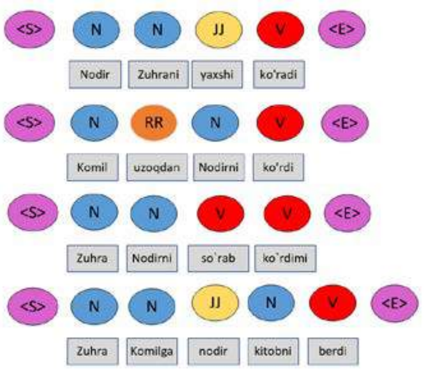

Fig. 4. Pos-tagged sentences (tags matching the beginning and end of the sentence)

In the next step, we will form a table representing the number of decimals of tags:

TABLE VI. STATISTICS OF CONSECUTIVE OCCURRENCE OF WORD GROUPS

	N	JJ	RR	V	<E>
<S>	4	0	0	0	0
N	3	2	1	2	0
JJ	1	0	0	1	0
RR	1	0	0	0	0
V	0	0	0	1	4

In the image above, we can see that the <S> tag is followed by the N tag four times. Adjective tag (JJ) occurs only 1 time after Noun tag (N). Therefore, the second entry is equal to 1. In a similar way, the remaining parts of the table are filled.

In the next step, we divide each term in the table row by the total number of the tag under consideration. For example, the quality tag (JJ) is followed by another tag 2 times. Therefore, we divide each element as shown in Fig. 5:

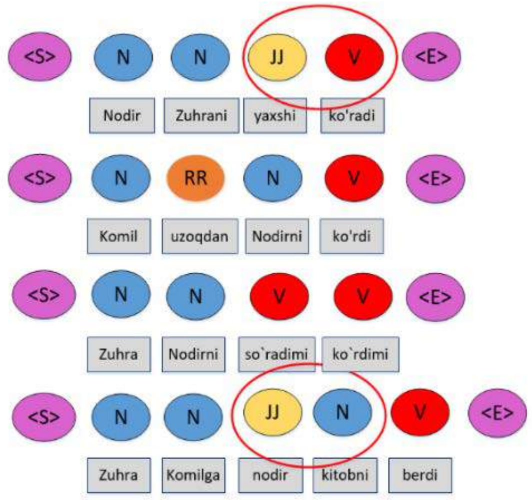

Fig. 5. Sentences tagged with pos

TABLE VII. EMISSION PROBABILITY VALUES

	N	JJ	RR	V	<E>
<S>	4/4	0	0	0	0
N	3/8	2/8	1/8	2/8	0
JJ	½	0	0	1/2	0
RR	½	0	0	0	0
V	0	0	0	1/5	4/5

The values in Table-VII are the transition probability values for the given four sentences. How does a HMM determine the appropriate sequence of tags for a given sentence from the above tables? Let's take a new sentence and label it with incorrect tags: We tag the sentence "*Zuhra Nodirni so‘rab ko‘rdi*" as follows (wrongly): Zuhra – noun; Nodirni - adjective; so‘rab – noun; ko‘rdi – verb. In the next step, we calculate the probability of this sequence being correct as shown in Fig. 6:

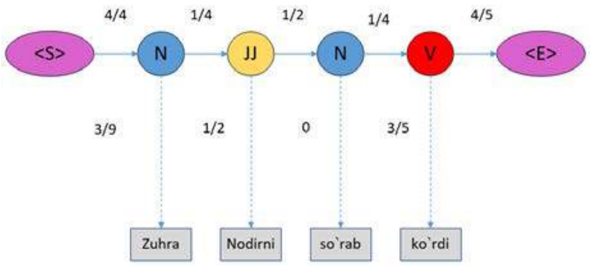

Fig. 6. Probability of consecutive occurrence of word groups

The probability of an Adjective (JJ) tag is 1/4 after a Noun (N) tag, as shown in the table. Also, the probability of the word Venus being Horse is 3/9. In the same way, we calculate each probability in the graph. Now the product of these probabilities determines the probability that the sequence is correct. The multiplier is zero because the tags are incorrect (not formatted):

$$\frac{4}{4}\cdot\frac{3}{9}\cdot\frac{1}{4}\cdot\frac{1}{2}\cdot\frac{1}{1}\cdot 0\cdot\frac{1}{4}\cdot\frac{3}{5}\cdot\frac{4}{5}=0$$

If the words in the given sentence are correctly tagged, we generate a probability greater than zero, as shown in Fig. 7:

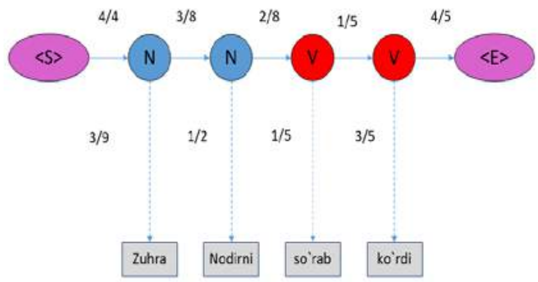

Fig. 7. Probability of consecutive occurrence of word groups

We calculate the product of values of transition and emission probabilities corresponding to the sequence of words in the given sentence:

$$\frac{4}{4}\cdot\frac{3}{9}\cdot\frac{3}{8}\cdot\frac{1}{2}\cdot\frac{2}{8}\cdot\frac{1}{5}\cdot\frac{1}{5}\cdot\frac{3}{5}\cdot\frac{4}{5}=0.0003$$

For example, given the three POS tags we mentioned, 16 different combinations of tags can be created. In this case, it seems possible to calculate the probabilities of all 16 combinations. But when the task is to define larger sentences, and all the POS tags in the Penn Treebank project are considered, the number of possible combinations grows exponentially and the task seems impossible. Let's think of these 16 combinations as paths and assign transition and emission probabilities to each of the vertices and edges of the graph as shown in Fig. 8:

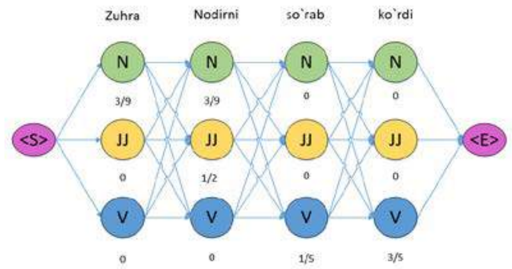

Fig. 8. A (public) graph consisting of all POS tags matching a given sentence

In the next step, all vertices and edges of the graph with zero probability should be removed from the graph. Also, vertices that do not lead to the endpoint are removed as shown in Fig.9:

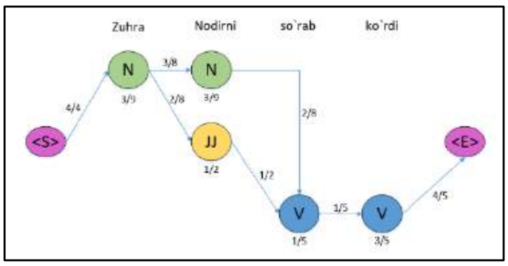

Fig. 9. A graph of POS tags corresponding to a given sentence

There are only two paths from the start point to the end point, and we calculate the probability associated with each path:

$$< S >\to N\to N\to V\to V\to < E > = \frac{4}{4}\cdot\frac{3}{9}\cdot\frac{3}{8}\cdot\frac{3}{9}\cdot\frac{2}{8}\cdot\frac{1}{5}\cdot\frac{1}{5}\cdot\frac{3}{5}\cdot\frac{4}{5}=0.0002$$

$$< S >\to N\to JJ\to V\to V\to < E > = \frac{4}{4}\cdot\frac{3}{9}\cdot\frac{2}{8}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{5}\cdot\frac{1}{5}\cdot\frac{3}{5}\cdot\frac{4}{5}=0.0004$$

In the calculations above, the probability of the second sequence is much higher, and therefore the HMM labels each word in the sentence according to this sequence. But the result is not what we expected, that is, the word "Nodirni" in the sentence "Zuhra Nodirni koʻrdi" was tagged as an adjective. Optimizing the Markov model using the Viterbi algorithm can achieve the desired result. The pseudocode of the HMM algorithm is given below:

σ ← k x N array
For s = 1 … k: $\sigma[1,s] \leftarrow \pi(s)\Pr[O^1|s]$
X ← kxN array
Populate σ and X
For i = 2 … N:
    For s = 1 … k:
     $x_{prev} \leftarrow \underset{x}{argmax}\{\sigma[i-1,x]\Pr[x \rightarrow s]\}$
     $X[i,s] \leftarrow x_{prev}$
     $\sigma[i,s] \leftarrow \sigma[i-1,x_{prev}]\Pr[x \rightarrow s]\Pr[O^1|s]$
Reconstruct OptPath:
     $s \leftarrow \underset{x}{\operatorname{argmax}}\{\sigma[N,x]\}$

Optpath ← EmptyList
For j = N … 1:
    Optpath ← s :: *Optpath*
     if j > 1: $s \leftarrow X[j,s]$
Return OptPath

The Viterbi Algorithm is a dynamic programming algorithm to identify a sequence of hidden states, called a Viterbi path, that uses observed sequences of events, specifically Markov data sources and Hidden Markov Models (HMMs).

In the comments above, the Hidden Markov model was optimized and the computations were reduced from 16 to just two. In the next step, methods of further optimization of the Hidden Markov model using the Viterbi algorithm are presented. Using the example we used earlier, we apply the Viterbi algorithm to it as shown in Fig.10:

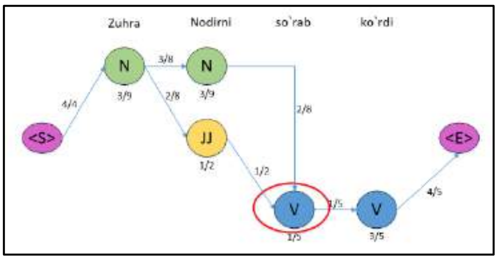

Fig. 10. A graph of POS tags compatible with the Viterbi algorithm

In the example above, we consider a split graph vertex. As shown below, there are two paths leading to this peak as shown in Fig.11:

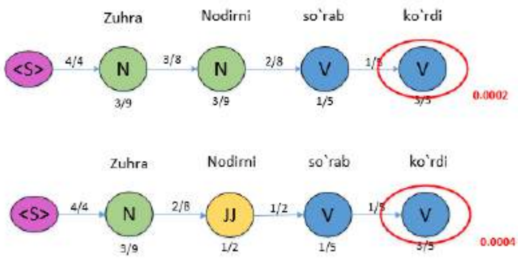

Fig. 11. Paths consisting of POS tags corresponding to a given sentence

In the next step, we analyze the path with the lowest probability. The same method is followed for all states in the graph as shown in the Fig.12:

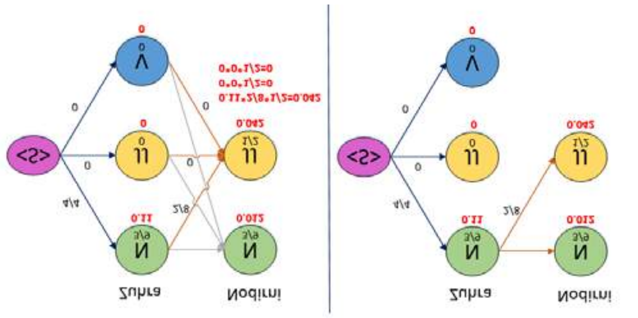

Fig. 12. A graph of POS labels is the probability values of the paths

In the next step, after calculating the probabilities of all the paths leading to the edge of the graph, it is necessary to remove the edges or the path with a lower probability value. It can also be seen from Fig.11 that some vertices have zero probability. After calculating the probabilities of all paths leading to the end of the graph, we form the following graph as shown in Fig.13:

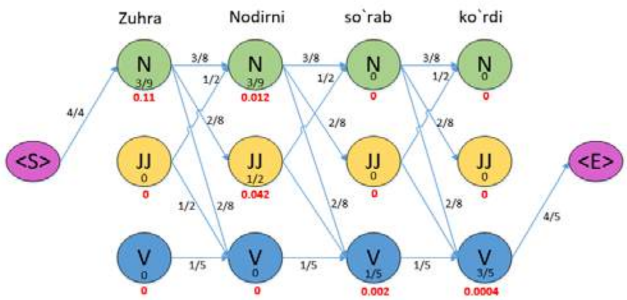

Fig. 13. Probability values of paths in a graph of POS labels (general)

This algorithm returns only one path, compared to the previous method, which offered two paths. Thus, fewer calculations were performed using this algorithm. After applying the Viterbi algorithm, the model tags the sentence as follows: Zuhra – noun; Nodirni – noun; soʻrab – verb; koʻrdi – verb.

These are valid tags and it can be concluded that the HMM model can successfully tag the words with the appropriate POS tags.

V. CONCLUSION

The hidden Markov model is a graphical model designed to analyze the probability of an event. The algorithms used in this model are used to study and infer random processes. Based on the observation data in the tagged national corpus of the Uzbek language, the conditional distribution of the given sentence according to the hidden cases can be determined, and Pos tagging can be carried out according to the value with the greatest probability. However, in order to improve the effectiveness of the Hidden Markov model algorithm, it is necessary to determine the sequence of hidden states in the form of a Viterbi path using the Viterbi algorithm. In this article, on the example of 4 tagged sentences in the Uzbek language corpus, the process of automatic tagging of the given sentence was carried out on the basis of successive steps, the results of the analysis were presented by means of graphs and tables. To increase the quality of POS tagging, it is desirable to increase the amount of observation data (tagged sentences in the corpus).It is also recommended to use the Baum-Welch algorithm to further improve the quality and efficiency of the analysis.

This article is part of the innovative project №. L-402104209 on the topic of "Creating an automatic processing tool for information search systems (Google, Yandex, Google translate) - a morpholexicon and a morphological analyzer

software tool of the Uzbek language" that is being implemented on the basis of the state order for scientific research works.

REFERENCES

[1] X. Gao, N. Zhu,(2013). Hidden Markov model and its application in natural language processing. Information Technology Journal, 12(17). https://doi.org/10.3923/itj.2013.4256-4261

[2] Z. Y. Xusainova, NLP: tokenizatsiya, stemming, lemmatizatsiya va nutq qismlarini teglash. [NLP-tokenization, lemmatization and part of speechs tegging] O‘zbek amaliy filologiyasi istiqbollari / Respublika ilmiy-amaliy konferensiya to‘plami. Elektron nashr / ebook. – Toshkent: ToshDO‘TAU, 26.10.2022. 159-163 b.

[3] D. Baishya & R: Baruah,(2021). Resource Languages with Improved Hidden Markov Model and Deep Learning. International Journal of Advanced Computer Science and Applications, 12(10).

https://doi.org/10.14569/IJACSA.2021.0121011

[4] B. Elov, Sh.Hamroyeva, X.Axmedova. (2022). Methods for creating a morphological analyzer, 14th International Conference on Intellegent Human Computer Interaction, IHCI 2022, 19-23 October 2022, Tashkent.

[5] J. Assunção, P. Fernandes & L. Lopes (2019). Language independent POS-tagging using automatically generated Markov chains. Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE, 2019-July. https://doi.org/10.18293/SEKE2019-097

[6] L.E. Baum, & T. Petrie (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. The Annals of Mathematical Statistics, 37(6).

https://doi.org/10.1214/aoms/1177699147

[7] L.R. Rabiner, (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, 77(2). https://doi.org/10.1109/5.18626

[8] D.E. Cahyani, W. Mustikaningtyas,(2022). Indonesian part of speech tagging using maximum entropy markov model on Indonesian manually tagged corpus. IAES International Journal of Artificial Intelligence, 11(1).

https://doi.org/10.11591/ijai.v11.i1.pp336-344

[9] D. Jurafsky, J. Martin, (2014). Speech and Language Processing. In Speech and Language Processing. (Vol. 3).

[10] D. L. Cing, & K. M. Soe, (2020). Improving accuracy of part-of-speech (POS) tagging using hidden markov model and morphological analysis for Myanmar Language. International Journal of Electrical and Computer Engineering, 10(2). https://doi.org/10.11591/ijece.v10i2.pp2023-2030

[11] D. Suleiman, A. Awajan, W. Etaiwi, (2017). The Use of Hidden Markov Model in Natural ARABIC Language Processing: A survey. Procedia Computer Science, 113. https://doi.org/10.1016/j.procs.2017.08.363

[12] D. Kumawat, V. Jain, (2015). POS Tagging Approaches: A Comparison. International Journal of Computer Applications, 118(6). https://doi.org/10.5120/20752-3148

[13] M. Hadni, S. Alaoui Ouatik, A. Lachkar, M. Meknassi, (2013). Hybrid Part-Of-Speech Tagger for Non-Vocalized Arabic Text. International Journal on Natural Language Computing, 2(6). https://doi.org/10.5121/ijnlc.2013.2601

[14] B. Bohnet, R. McDonald, G. Simões, D. Andor, E. Pitler, J. Maynez, (2018). Morphosyntactic tagging with a meta-bilSTM model over context sensitive token encodings. ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), 1. https://doi.org/10.18653/v1/p18-1246

[15] Y. H. Khoe. (2020). Reproducing a morphosyntactic tagger with a meta-BiLSTM model over context sensitive token encodings. LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings.