O‘zMU XABARLARI

ВЕСТНИК НУУз

ACTA NUUz

MIRZO ULUG‘BEK NOMIDAGI O‘ZBEKISTON MILLIY UNIVERSITETI ILMIY JURNALI

JURNAL 1997-YILDAN CHIQA BOSHLAGAN

2023

MAXSUS SON

ТОШКЕНТ – 2023

MUNDARIJA

O‘ZBEKISTON MILLIY UNIVERSITETI XABARLARI, 2023
ISSN 2181-7324

MAXSUS SON
http://journals.nuu.uz

Botir ELOV,
doctor of philosophy (PhD) of technical sciences, associate professor.
Tashkent State University of Uzbek Language and Literature named after Alisher Navoi.
E-mail: elov@navoiy-uni.uz
Shahlo HAMROYEVA,
doctor of philological sciences (DSc), associate professor.
Tashkent State University of Uzbek Language and Literature named after Alisher Navoi.
E-mail: shaxlo.xamrayeva@navoiy-uni.uz
Zilola XUSAINOVA,
PhD student of
Tashkent State University of Uzbek Language and Literature named after Alisher Navoi.
E-mail: xusainovazilola@navoiy-uni.uz
Nizomaddin XUDAYBERGANOV,
Teacher of the Department of Computer Linguistics and Digital Technologies of Tashkent State University of Uzbek Language and Literature named after Alisher Navoi.
E-mail: nizomaddin@navoiy-uni.uz
Tel: 91 422 46 26,
Umid YODGOROV,
Teacher of Tashkent State University of Uzbek Language and Literature named after Alisher Navoi.
e-mail: yodgorov@navoiy-uni.uz
***Aziz YULDASHEV**,*
Teacher of Tashkent State University of Uzbek Language and Literature named after Alisher Navoi.
E-mail: yuldashevaziz@navoiy-uni.uz

POS TAGGING OF UZBEK TEXTS USING HIDDEN MARKOV MODELS (HMM) AND VITERBI ALGORITHM

Annotation

Nowadays one of the popular problems of Natural Language Processing (NLP) is defining the categories of words in a given text. Being able to determine whether words in a sentence belong to such categories as noun, pronoun, verb, adverb, etc. is important and is called the POS tagging task in NLP. Phrase tagging (POS tagging, PoS tagging, or POST) in NLP, is also called grammar tagging or phrase segmentation. The process of assigning a word in a text (corpus) to a certain category is based on the context. This article presents methods and algorithms for tagging Uzbek texts using hidden Markov models and the Viterbi algorithm based on the tagged corpus of the Uzbek language.

Key words: Parts of Speech Tagging, POS tagging, Hidden Markov Model, Markov chain, Hidden Markov Model, HMM, stochastic methods, NLP, transition probability, emission probability, Viterbi Lattice, Viterbi algorithm.

Introduction. Parts of speech (also known as POS) and named objects are important in learning the grammar of any language. Knowing whether a word is a noun or a verb, the ability to determine the syntactic structure of the words next to it may belong to, indicates that the tagging of word groups is one of the main factors. Knowing the meaning of nouns, names of persons, names of places, etc., is significant in performing many tasks of NLP. In this article, we will study tagging of word groups, the probability of word sequences, as well as the detection of named objects (NER), and also tagging words in the form of as a person, a place, an organization.

The world languages have four main categories: nouns (including proper nouns), verbs, adjectives, and adverbs, and a smaller category such as interjections. English has these five categories, but other languages may not have such categories [1; 2].

Nouns are words that refer to a name, place, thing, person, or event. In many languages, including English and Uzbek, common nouns are divided into countable and uncountable nouns. Countable nouns can be singular and plural (goat/goats, relationship/relationships), they can also be counted (one goat, two goats). Uncountable nouns show something as a group. Thus, snow, salt, and justice are not considered countable nouns [3; 4]. Parts of speech also have different subgroups.

Verbs describe actions, situations, and processes. Adjectives often describe characteristics or qualities of a horse, such as its color (white, black), age (old, young), and value (good, bad). Some languages do not have adjectives. For example, in Korean, words that are adjectives in English act do the function of verbs, so the adjective "beautiful" in English is used as a verb meaning "to be beautiful" in Korean. Adverbs express the state, amount, degree, time, place of the action. Pronouns act as stenography for an event. Personal pronouns refer to persons or objects. Possessive pronouns are other forms of pronouns that mainly express possession and indicate the relationship between a person and an object. Conjunctions join two phrases, clauses, or sentences. The auxiliary words in a sentence link the noun to another category. Complements add extra meaning to words and sentences [4].

A simplified form of this (such Grammar) is usually taught to school-aged children to identify words like nouns, verbs, adjectives, adverbs, and other parts of speech. Identifying Parts of Speech tags is a much more complex process than simply comparing Parts of Speech tags. This is because there is no general approach to POS tagging in NLP. One word can belong to different tags in different sentences based on the context [3; 5].

It is not possible to manually index (record) Parts of Speech tags for the given language corpus. Dictionaries are constantly adding new words from different languages, and it’s impossible to expand POS tags by themselves manually. That's why we use a machine-based POS tag.

The main part

POS tagging

POS tagging itself cannot solve any NLP problems. POS tagging is one of the first steps in the NLP pipeline and is a prerequisite for simplifying many NLP tasks.

POS tagging algorithms are divided into two separate groups [6,7]:

POS taggers basedon rules.

Stochastic POS taggers.

The E.Brill POS tagging method is one of the first and most widely used English POS taggers and uses algorithms based on rules [8].

POS tagging based on rules

Automatic POS tagging is one area of NLP where statistical methods are more effective than rule-based methods. Conventional rule-based approaches use context information to tag unknown or ambiguous words. POS is performed by analyzing the linguistic features of a word, the preceding and following words, and other aspects in order to eliminate ambiguity in tagging. For example, if the preceding word is an adjective, then the word in question is most likely to be a noun. This information is coded in the form of rules [9].

An example of a grammar rule:

If the indefinite/unknown X word is preceded by a determiner and followed by a noun, mark it as an adjective.

It is very difficult to manually define a set of rules. So, we need some automated method or system is needed to do this.

The Brill tagger is a rule-based tagger that looks through training data, finds a set of tagging rules that define the data well, and minimize POS tagging errors [8]. The most important thing to note about Brill tagging here is that the rules are not created manually, but instead are defined using the provided corpus. The only requirement for feature development is to define a set of rule templates that the model can use to generate new features.

Stochastic POS tagging

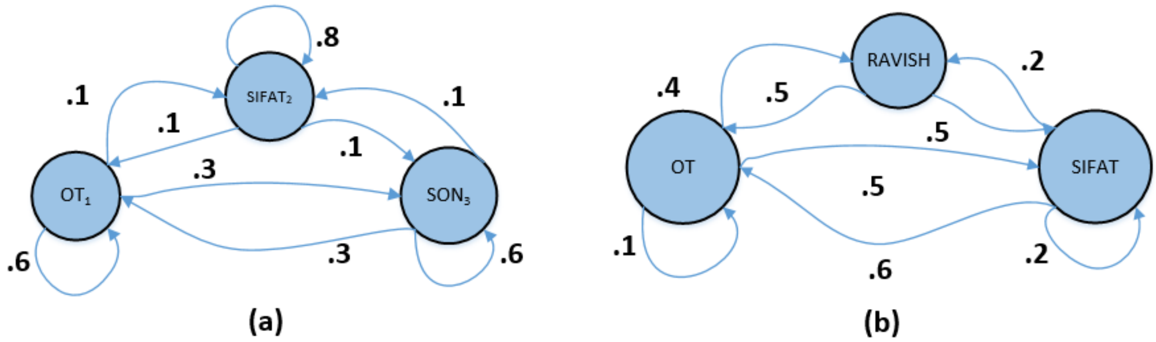
The term "stochastic tagger" can refer to different approaches of the POS tagging problem. Any model that involves frequency or probability in some way can be called stochastic. The simplest stochastic taggers identify words based on the probability of the words occurring with a given tag. In other words, the most frequent tag and training dataset with a word is the tag assigned to the ambiguous word pattern [10; eleven]. The problem with this approach is that while it may return the correct tag for a given word, it may return the wrong sequence of tags.

An alternative approach to word frequency is to calculate the probability of occurrence of a sequence of tags. This approach is sometimes referred to as the n-gram approach and is based on determining the best tag for a given word by the probability of its occurrence with the previous n tags. This approach considers tags for individual words depending on the context. An approach using tag sequence probabilities and word frequency measures is known as the Hidden Markov Model (HMM) [12;13;14].

Hidden Markov Models (HMMs)

The Markov chain is a model that describes the probability of a sequence of random events/variables. In a Markov chain, only the current state is used to predict the next element in the sequence. All conditions before the current state do not affect the next steps [15,16].

We will study the Markov chain model for predicting the next word from the current word in a sentence to be generated in Uzbek. Words can belong to the noun/adjective/number group of words. The word sequence of the current word can be used to predict the next word. In this case, the previous word is not used in the prediction. The following examples illustrate this process:



Picture. Predicting parts of speech using Markov chain.

In the first Markov chain (a), there are states of NOUN, ADJECTIVE and NUMERALS, and the numbers in decimal format on the edges connecting the vertices of the graph indicate the probability of transition from one state to another state (State1→State2). In this graph, the probability that a NOUN word group will be followed by group of words belong to an ADJECT 0.1. Table 1 below shows all the components of a Markov chain:

Table 1. All components of Markov chain

Q = Q1,Q2,…Qn	A group of N cases
A = a11,a12,…an1…ann	*A* is a transition matrix of probability, representing the probability of transition from each state $aij - i$ to another state j.$\sum_{i=1}^{n} a_{ij} = 1, \quad \forall i$
π = π1, π2,… πn	The initial probability distribution for S cases. pi – i represents the probability that the Markov chain starts in a certain state.$\sum_{i=1}^{n} \pi_i = 1$

HMM

HMM is a very powerful statistical modeling tool whi is used for speech recognition, handwriting recognition and other NLP tasks. Hidden Markov Model (HMM) is an unsupervised machine learning algorithm that is part of graphical models. If the training data is available, the HMM is trained using a supervised learning method [17;18]. It is important to understand where the

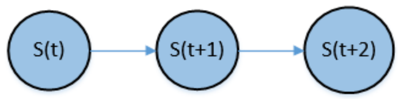
Hidden Markov Model algorithm actually fits or is used. Today, the HMM graphical model is used to predict (hidden) states using sequential data such as weather, text, speech, etc. [19;20;21].

In some cases, it was necessary to predict the sequence of events that cannot be directly observed in the environment. But while we are given a sequence of other observable states in the environment, these hidden states depend on the observable states. That is, the most important point established by HMM is that the future state/event depends only on the current state/event and not on other past states.

Arithmetically, the probability of a state at time t depends only on the time step t-1. In other words, the probability of s(t) given s(t-1) is p(s(t) ┤| s(t-1)). This is used as a first order Markov model. If the probability of state s at time t depends on time steps t-1 and t-2, this is called a 2nd order Markov model. That is, when dependence on past events increases, order increases accordingly. A second-order Markov model can be written as p(s(t)┤|s(t-1), s(t-2)). The probability of successive occurrence of events s1, s2 and s3 is calculated as follows [22]:

$$\begin{aligned} p(s_3, s_2, s_1) &= p(s_3|s_2, s_1)p(s_2, s_1) \\ &= p(s_3|s_2, s_1)p(s_2|s_1)p(s_1) \\ &= p(s_3|s_2)p(s_2|s_1)p(s_1) \end{aligned}$$

The figure below is a schematic of the simple Markov model that we have defined in the above equation.



Picture 2. Diagram of a simple Markov model

Probability of transition

It is defined as the probability of transition from one state to another. Thus, if there are 3 states (Q1, Q2, Q3), there will be a total of 9 possible transitions. In the diagram below, all transition probabilities for states A, B, C are determined. The transition probability is usually denoted by A. It is interpreted as the probability of transition of the system from state i to state j in time step t+1.

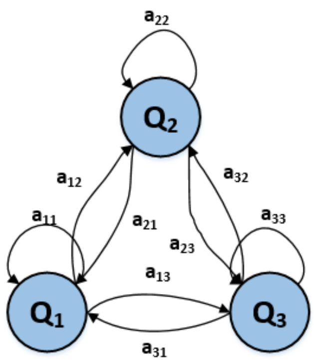


Figure 3. 3-state transition probability diagram

Mathematically,

a_ij=p(s(t+1) =j ┤| s(t)=i)

For example, in the state diagram above, the transition probability from state Q1 to state Q2 is defined as α12. If we observe state Q1 for two days in a row, then the probability of transition from state Q1 to state Q1 in step t+1 is a11. It is usually defined using the transition probability matrix (M x M). For our example above, the transition probability matrix can be determined as follows:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{23} & a_{33} \end{pmatrix}$$

One important property to note is that when the machine transitions to another state, the sum of all transition probabilities, given the current state, must be equal to 1. In our example, a11+a12+a13=1 yoki:

$$\sum_{i=1}^{n} a_{ij} = 1, \quad \forall i$$

The initial state of the Markov model (when the time step is t = 0) is denoted by p, which is a vector of size M. The sum of all probabilities must be equal to 1, that is:

$$\sum_{i=1}^{n} \pi_i = 1$$

During initialization, all cases can be assigned the same probability. In our example, we set the initial state to $\pi = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$. In some cases, we can have $\pi_i = 0$because they (i) cannot be the initial state.

Markov chain

There are 4 main types of Markov models [12;13;19]. When a system is completely observable and autonomous, it is called the Markov chain. What we have learned so far is an example of a Markov chain. Thus, we can conclude that the Markov chain consists of the following parameters:

M is the set of cases;

A is the matrix of transition probabilities;

p is the initial probability distribution.

If the probability of transition from any step to other steps is zero, this is fixed as a final state. Thus, when the system enters the final state, it never moves to the next step.

In a hidden Markov model, the state of the system is hidden (unknown), but at each time step t, the system in the state s(t) determines the observed/visible value v(t). The diagram below shows the general outline of a Hidden Markov Model:

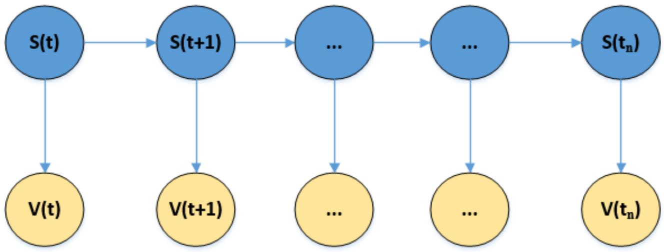


Figure 4. General scheme of Hidden Markov model

The following considerations apply to HMM:

We can define a certain sequence of visible/observable states as V^T={v(1),v(2)…v(T)};

Let's denote our model as θ. Therefore, for any state of s(t) there is a probability of the state vk(t).

Since we only have visible states, s(t)s is not observable, and such model is called the Hidden Markov model.

Such network is called the Limited state machine.

If state machines are associated with transition probabilities, this is called the Markov network.

Biz ko‘rinadigan/kuzatiladigan holatlarning ma’lum bir ketma-ketligini $\boldsymbol{V^T} = \{\mathbf{v(1)}, \mathbf{v(2)} \dots \mathbf{v(T)}\}$ sifatida belgilashimiz mumkin;

Probability of emission

Let's redefine our previous example. Suppose that depending on any 3 states (Q1, Q2, Q3), there are visible/observable symbols of V1 and V2. An HMM model containing the latent state and characters in the model is as follows:

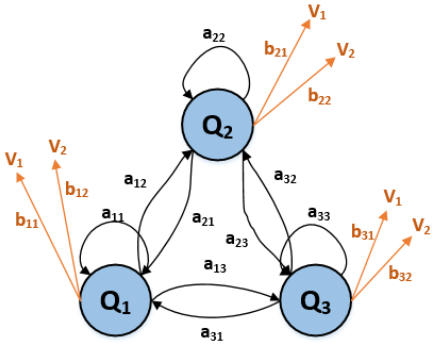


Figure 5. Emission probability in a hidden Markov model

The observables in the hidden Markov models are (V1,V2), and one of them must be extracted from each case. The probability of emitting any character is known as the emission probability, commonly denoted bjk. Mathematically, the probability of producing a symbol k given state j is defined as:

b_jk=p(v_k (t)|s_j (t))

The emission probability matrix is defined using an MxC matrix:

$$B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{23} \end{pmatrix}$$

Like the transition probability, the sum of the emission probabilities is 1. That is,

$$\sum_{k=1}^{C} b_{jk} = 1, \quad \forall j$$

Various attributes/properties of Hidden Markov Model have been identified in the above comments. Prediction of subsequent states is the ultimate goal for any statistical model/algorithm. However, before making predictions, we need to solve two main problems in HMM.

Assessment problem;

A learning problem.

Decoding problem.

Assessment problem

First, we define the model (θ) as follows:

θ→s,v,a_ik,b_jk

Considering the model θ and the sequence of observed signs (V^T) we need to determine θ probability of the generation of a certain sequence of states / signs generated (determined) based on the θ model.

There can be many {θ1,θ2…θn} models. We need to find such $p(V^T|\theta_i)$ and correctly classify the sequence V^T using Bayes rule V^T. In this case, the following equality is relevant:

$$p(\theta|V^T) = \frac{p(V^T|\theta)p(\theta)}{p(V^T)}$$

Learning problem

In general, HMM is an unsupervised machine learning process where different types of visible characters are known. But the number of hidden cases is unknown. The idea behind the HMM is to try different options and require more computation and processing time. Therefore, HMM uses study data and a certain number of hidden states to make faster, better predictions.

After determining the high-level structure of the model (the number of hidden and visible states), it is necessary to estimate transition (a_ij) and emission (b_jk) probabilities using training sequences. This is noted as the learning problem in science.

We use the evaluation problem to solve the learning problem. Therefore, it is important to understand how the estimation problem works. Also, in many cases, the Expectation Maximization (EM) algorithm is used to estimate the probability of transition (a_{ij}) and emission (b_{jk}) [23]. The learning problem is solved by the forward-backward or Baum-welch algorithm [24;25].

Decoding problem

Once the transition (a_{ij}) and emission (b_{jk}) probability estimates are determined, we can use the (θ) model to predict the WT hidden states that produced the VT visible sequence. The decoding problem is also known in science as the Viterbi algorithm [26]. After defining the concept of hidden Markov model and the main steps, it is possible to consider the issue of POS tagging of sentences in Uzbek using this model.

In the HMM above, the parts of speeches like NOUN, NUMERALS and VERBS are given as observable cases. But we are more interested in observing the sequence of words (hidden cases) of Uzbek sentences, mainly belonging to the parts of speech as NOUN and ADJECTIVE.

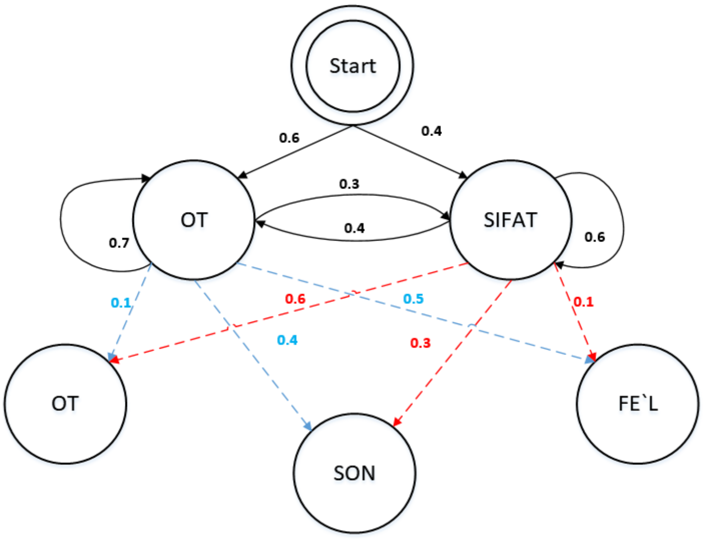


Figure 6. Prediction of POS using Hidden Markov Model (HMM).

B. Elov, Sh. Hamroyeva, O. Abdullayeva and M. Uzokova developed an alternative tagset based on the laws of the Uzbek language [27]. Table 1 below lists the Uzbek language POS tags in short form:

Table 2. List of Uzbek language POS tags (short version)

№	So‘z turkumi	POS
1.	Ot	N
1.1.	Mavhum ot	NAbs
1.2.	Aniq ot	NCnc
1.3.	Atoqli ot	NP
1.4.	Turdosh ot	NC
1.5.	Sodda ot	ND
1.6.	Qo‘shma ot	NCmp
1.7.	Juf ot	NCpl
1.8.	Takroriy ot	NRep
1.9.	Tub ot	ND0
1.10.	Yasama ot	ND1
2.	Sifat	JJ
2.1.	Sodda sifat	JJD
2.2.	Qo‘shma sifat	JJCmp
2.3.	Juft sifat	JJCpl
2.4.	Takror sifat	JJRep
2.5.	Tub sifat	JJD0
2.6.	Yasama sifat	JJD1
3.	Fe’l	VB
3.1.	Mustaqil fe’l	VBI
3.2.	O‘timli fe’l	VBTran (Transitive)

3.3.	O‘timsiz fe’l	VBItran (Intransitive)
3.4.	Sodda fe’l	VBD
3.5.	Qo‘shma fe’l	VBComp
3.6.	Juft fe’l	VBCpl
3.7.	Takror fe’l	VBRep
3.8.	Tub fe’l	VBD0
3.9.	Yasama fe’l	VBD1
3.10.	Ko‘makchi fe’l	VBH
3.11.	Yetakchi fe’l	VBL
3.12.	To‘liqsiz fe’l	VBNotF
4.	Son	NUM
4.1.	Sodda son	NUMD
4.2.	Murakkab son	NUMCmp
5.	Olmosh	P
5.1.	Kishilik olmosh	PP
5.2.	Ko‘rsatish olmosh	PDem (Demosntrative)
5.3.	O‘zlik olmosh	PRef (Reflexive)
5.4.	Gumon olmosh	PPred (Prediction)
5.5.	So‘roq olmosh	PQues (Question)
5.6.	Sodda olmosh	PD
5.7.	Qo‘shma olmosh	PCmp
5.8.	Juft olmosh	PCpl
5.9.	Takror olmosh	PRep
5.10.	Tub olmosh	PD0
5.11.	Yasama olmosh	PD1
6.	Ravish	RR
6.1.	Payt ravishi	RT
6.2.	O‘rin ravishi	RL
6.3.	Holat ravishi	RCon (Condition)
6.4.	Miqdor-daraja ravishi	RMes (Measurment)
6.5.	Maqsad ravishi	RPur (Purpose)
6.6.	Sabab ravishi	RRs (Reason)
6.7.	Sodda ravish	RRD
6.8.	Qo‘shma ravish	RRComp
6.9.	Juft ravish	RRCpl
6.10.	Takror ravish	RRRep
6.11.	Tub ravish	RRD0
6.12.	Yasama ravish	RRD1
7.	Bog‘lovchi	C
7.1.	Sof bog‘lovchi	C0
7.2.	Vazifadosh bog‘lovchi	C1
7.3.	Teng bog‘lovchi	CEq (Equeal)
7.4.	Ergashtiruvchi bog‘lovchi	CS
8.	Ko‘makchi	II
8.1.	Sof ko‘makchi	II0
8.2.	Vazifadosh ko‘makchi	II1
9.	Yuklama	Prt (Particle)
9.1.	So‘z yuklama	PrtD
9.2.	Qo‘shimchasimon yuklama	PrtComp
10.	Modal so‘z	MD
11.	Undov so‘z	UH
11.1.	His-hayajon undovlari	UHEm (Emotion)
11.2	Buyruq-xitob undovlari	UHImp (Imperative)
12.	Taqlid so‘zlar	IM (Imitative)

Using HMM to develop POS Tagger

The words in this Uzbek language can be understood as observable cases (given to us in the data). POS tags that match words can be written as hidden states, so HMM can be used to evaluate POS tags. Note that we refer to observed states as "observations" and hidden states as "states". The Hidden Markov Model has the following components:

Table 3. Components of HMM

Q = Q1,Q2,…Qn	A group of N cases (Unnnoun)
A = a11,a12,…an1…ann	is a transition matrix of probability, representing the probability of transition from each state $aij - i$ to another state j.

	$$\sum_{i=1}^{n} a_{ij} = 1, \quad \forall i$$
O = O1,O2,…Ot	T is a sequence of observations (O), all of which are taken from a special dictionary (source). V = V1,V2,…Vt
B = bi(Ot)	A sequence of observation of probabilities (called emission probabilities), all of which represent the probability that observation Ot will occur from state i.
π = π1, π2,… πn	The initial probability distribution for S cases. πi means the probability that the Markov chain starts at a certain state i. $$\sum_{i=1}^{n} \pi_i = 1$$

In that,

Q is the set of possible labels;

A-A matrix represents the probability that the current tag will be formed from previous tags, keeping the tag transition probability P(ti|ti−1) Example: Calculation of A[Verb] [Ot]:

P (Noun |Verb): Count (Noun and Verb) / Count (Verb)

O - the sequence of observation (for the words in the sentence);

B – B is the emission probability representing the probability of P(wi|ti) that the given tag (supposing, a verb) is associated with a given word. For example, the outlier probability B[Verb][boiler] is calculated using the following formula:

P (pot | Verb): Count (pot and Verb) / pot (Verb)

It should be noted that the values of the Count () function mentioned above are taken from the tagged corpus (data) of the Uzbek language to study the HMM model [16]. Matrices "A" and "B" in the HMM model for the sentence “Bu yoqdagi odamlar menga juda yoqdi” ("I really liked the people here”) look like this:

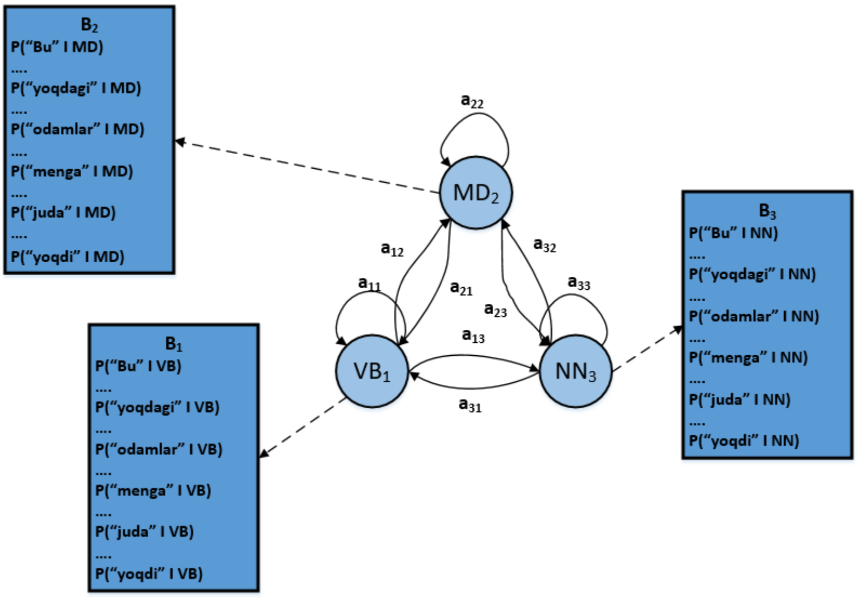


Figure 7. Transition and emission matrices in the HMM model

Here, the solid black lines in “A" represent the values of transition matrix, and the dotted black lines in "B" represent the emission matrix for a system with Q: {MD, VB, NN}.

Decoding using HMM

Suppose, an NMM be given, consisting of a transition and emission matrix and a sequence of observations O=O1,O2, …,Ot (words in corpus sentences). Given these values, we have to to determine the maximum possible sequence of states Q=Q1,Q2, …,Qt (POS tags). In order to decode a sequence of tags using HMM, two main assumptions are made:

the probability of the appearance of the current word depends only on its own tag and does not depend on neighboring (3-gram) words and tags;

the probability of occurrence of a tag does not depend on the sequence of previous tags, but only on the previous tag (gram 2).

The pseudocode of the HMM algorithm is given below:

Hidden Markov Model Algorithm
Initialize:
σ ← k x N array
For s = 1 … k: σ[1, s] ← π(s)Pr[O1|s]
X ← k x N array
Populate σ and X
For i = 2 … N:

For s = 1 … k:
xprev ← argmax {σ [i-1, x] Pr[x → s]}
x
X[i, s] ← xprev
σ[i, s] ← σ[i -1, xprev] Pr[x → s]Pr[Oi|s]
Reconstruct OptPath:
s ← argmax{σ[N, x]}
x
Optpath ← EmptyList
For j = N … 1:
Optpath ← s :: Optpath
if j > 1: s ← X[j, s]
Return OptPath

Viterbi algorithm

The decoding process used for HMM is called Viterbi algorithm [26]. First, it is necessary to form a probability matrix called a grid. The columns in this matrix are the sequence of words in a sentence; lines: represent hidden states (all possible POS tags). The Viterbi grid, (Viterbi Algorithm) corresponding to the saying “Shirin nodir toshni o‘tga otdi” ("Shirin threw the rare stone into the fire"), looks like this:

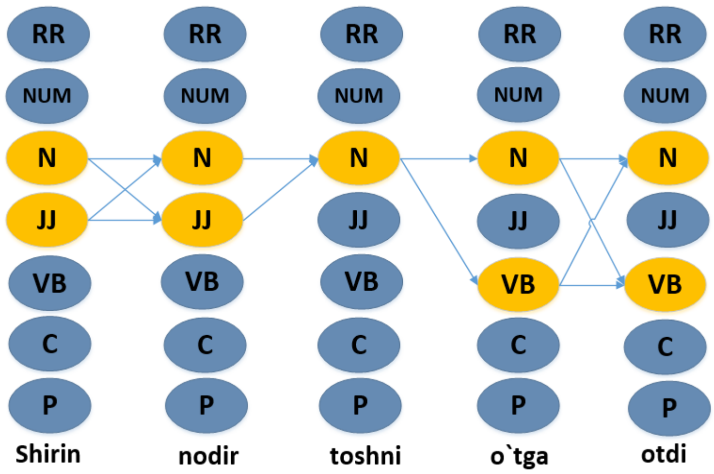


Figure 8. Viterbi grid/algorithm corresponding to this sentence

In the Viterbi grid/algorithm above, one can observe the columns corresponding to the words in the given sentence (Shirin nodir toshni o‘tga otdi) (Shirin threw the rare stone into the fire) and the rows representing all known POS tags (RR, NUM, N, JJ, VB, C, P). The data in this grid can be interpreted as follows:

Each cell (cell) of the grid is represented as V V(t,j) (“t” represents a column and j represents a row, called the Viterbi path probability);

After the first t observations, the HMM calculates the probability of occurrence of state j (current POS tag) and determines/identifies the sequence of states with the highest probability;

The V(t,j) value is calculated based on the following formula:

$$V(t,j) = \ max: V_{t-1} \cdot a(i,j) \cdot b(j,O_t)$$

Here:

$\boldsymbol{V_{t-1}}$ – the probability of the Viterbi path corresponding to the previous steps.

$\boldsymbol{a(i,j)}$ – the probability of transition from the previous state qi to the current state qj. ;

$\boldsymbol{b(j,O_t)}$ – the probability of observation state j taking into account of the current state of Ot.

Using the Viterbi algorithm, it is necessary to calculate the values of the transition matrix “A” and the emission matrix “B” of the HMM discussed above. In our example, using the bigram HMM model, the POS tags depend only on the previous tag. For the sentence “Shirin nodir toshni o‘tga otdi” ("Shirin threw the rare stone into the fire"), it is necessary to have the result in the following format:

Sweet/N; rare/JJ; stone/N; into the fire/N; shot/VB

Here, characters like N, JJ and VB are POS tags. To use HMM, it is necessary to form the necessary matrices calculated using the corpus of the Uzbek language (tables 4.1, 4.2) [28;29]:

Table 4.1. Statistical transition matrix (partial)

	N	JJ	P	RR	NUM	VB	C	PRT	MD	UH	IM	II
<S>	12	2	6	3	1	0	0	0	0	0	0	0
N	21	3	1	3	1	13	0	0	0	0	0	2
JJ	3	0	0	0	0	3	0	0	0	0	0	0
P	3	1	2	2	0	1	0	0	0	0	0	1
RR	3	1	1	1	0	2	0	0	0	0	0	0
NUM	2	0	0	0	0	0	0	0	0	0	0	0

VB	1	0	0	0	0	1	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0
PRT	0	0	0	0	0	0	0	0	0	0	0	0
MD	0	0	0	0	0	0	0	0	0	0	0	0
UH	0	0	0	0	0	0	0	0	0	0	0	0
IM	0	0	0	0	0	0	0	0	0	0	0	0
II	2	0	0	0	0	0	0	0	1	0	0	1

Table 4.2. A - transition matrix (partial)

	N	JJ	P	RR	NUM	VB	C	PRT	MD	UH	IM	II
<S>	0.2553	0.2857	0.6	0.34	0.5	0	0	0	0	0	0	0
N	0.4468	0.4286	0.1	0.34	0.5	0.65	0	0	0	0	0	0.5
JJ	0.0638	0	0	0	0	0.15	0	0	0	0	0	0
P	0.0638	0.1429	0.2	0.2222	0	0.05	0	0	0	0	0	0.25
RR	0.0638	0.1429	0.1	0.1111	0	0.1	0	0	0	0	0	0
NUM	0.0425	0	0	0	0	0	0	0	0	0	0	0
VB	0.0213	0	0	0	0	0.05	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0
PRT	0	0	0	0	0	0	0	0	0	0	0	0
MD	0	0	0	0	0	0	0	0	0	0	0	0
UH	0	0	0	0	0	0	0	0	0	0	0	0
IM	0	0	0	0	0	0	0	0	0	0	0	0
II	0.0425	0	0	0	0	0	0	0	1	0	0	0.25

Table 5. B - emission matrix

	Shirin	nodir	toshni	o‘tga	otdi
N	0.72	0.32	0.12	0.31	0.11
JJ	0.47	0.07	0	0	0
P	0	0	0	0	0
RR	0	0	0	0	0
VB	0	0	0	0.29	0.28
C	0	0	0	0	0
PRT	0	0	0	0	0
MD	0	0	0	0	0
UH	0	0	0	0	0
IM	0	0	0	0	0
II	0	0	0	0	0

There are 5 columns equivalent to our example. The first step is to determine the meanings of the word "Shirin”. To do this, use P (POS tag | start) with the help of the transition matrix "A" (first row, initial_probabilities).

V(1,1) = P(P | Start) * P('Shirin' | P) = 0,6 * 0 = 0;
V(1,2) = P(C | Start) * P('Shirin' | C) = 0 * 0 = 0;
V(1,3) = P(VB | Start) * P('Shirin' | VB) = 0 * 0 = 0;
V(1,4) = P(JJ | Start) * P('Shirin' | JJ) = 0.2857 * 0.47 = 0.1343;
V(1,5) = P(N | Start) * P('Shirin' | N) = 0.2553 * 0.72 = 0.1838;
V(1,6) = P(NUM | Start) * P('Shirin' | NUM) = 0.5 * 0 = 0;
V(1,7) = P(RR | Start) * P('Shirin' | RR) = 0.34 * 0 = 0;

For the word "nodir" ("rare") we use the following formula:

$$V(t,j) = max: V_{t-1} \cdot a(i,j) \cdot b(j,O_t)$$

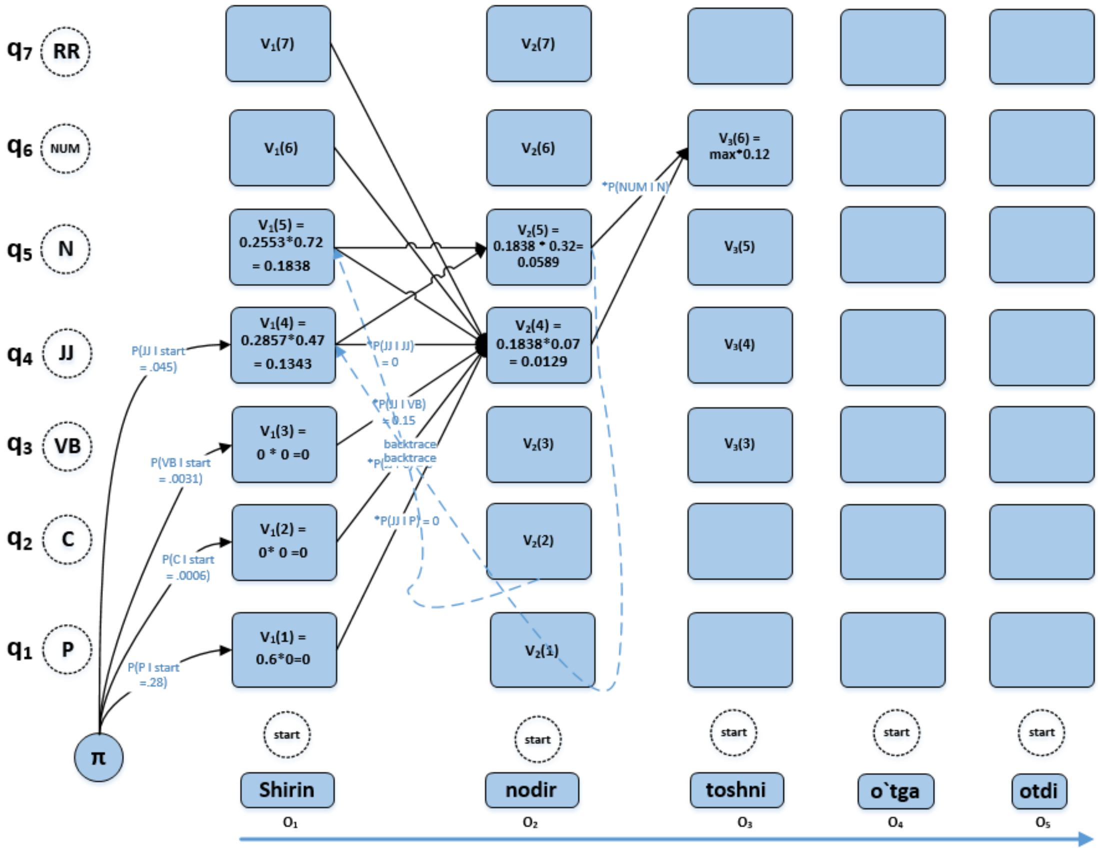


Figure 9. HMM model

However, the value of $b(j, O_t)$ will have the same constant (invariant) for all calculations for that cell. So, to calculate value of $max: V_{t-1} \cdot a(i,j) \cdot b(j,O_t)$ we have to multiply the $V_{t-1} \cdot a(i,j)$ and $b(j,O_t)$. Here,

i: previous tag;

j: current tag.

Relying to the considerations above, it is necessary to calculate the value max: V_(t-1) · a(i,j). where j represents the current cell of the row (POS tag) in the column corresponding to the word "unique". Also, to avoid confusion, the value of V[j,t] can be interpreted as the value corresponding to the jth row and t-column in the Viterbi matrix.

Biroq $b(j, O_t)$ qiymat ushbu katak uchun barcha hisoblar uchun doimiy bir xil qitmat (o‘zgarmas)ga ega bo‘ladi. Demak, $max: V_{t-1} \cdot a(i,j) \cdot b(j,O_t)$ qiymatni hisoblash uchun $V_{t-1} \cdot a(i,j)$ va $b(j,O_t)$ qymatni ko‘paytirish lozim.

Bu yerda

i: oldingi teg;

j: joriy teg.

Yuqoridagi mulohazalardan, $\boldsymbol{max: V_{t-1} \cdot a(i,j)}$ qiymatni hisoblash kerak bo‘ladi. Bu yerda j "nodir" so‘ziga mos ustunidagi joriy qator katakchasi (POS tegi)ni ifodalaydi. Shuningdek, chalkashmaslik uchun V[j,t] qiymatni Viterbi matritsasidagi j-satr va t-ustunga mos qiymat sifatida talqin qilinish mumkin.

Let’s study j = 2 (the case where POS is MD). To do this, we need to calculate the value of V(2,2). $max: V_1$= V(1,5) = 0.1838

V(2,1) = 0; V(2,2) = 0; V(2,3) = 0; V(2,6) = 0; V(2,7) = 0;

V(2,4) = V(1,5) * P (P | JJ) = 0.1838 * 0.07= 0.0129;

V(2,5) = V(1,5) * P (P | N) = 0.1838 * 0.32= 0.0589;

It is necessary to calculate all values of V(i,j) according to the above method.

After all the cells of the matrix V are filled, we select the appropriate label for the column (word) to determine the maximum values in the grid. For example, N - POS is selected as a tag for the word "Shirin". And The Viterbi algorithm is given below:

Yuqorida keltirilgan usul bo‘yicha barcha V(i,j) qiymatlarni hisoblash lozim.

Viterbi algorithm

function VITERBI (observations of len T, state-graph of len N) return best-path, path-prob

create a path probability matrix viterbi[N,T]

for each state s from 1 to N do ishga tushurish bosqichi

viterbi[s,1] ← πs * bs(o1)

backpointer[s,1] ← 0

for each time step t from 2 to T do

for each state s from 1 to N do

viterbi[s,1] ← $\overset{N}{\max}$ viterbi[s1, t-1] * $a_{s^1,s} * bs(o_t)$

s1=1
N
backpointer[s,1] ← argmax viterbi[s1, t – 1] * $a_{s^1,s} * bs(o_t)$
s1=1
N
bestpathprob ← max viterbi[s, T] tugatish bosqichi
s=1
N
bestpathpointer ← argmax viterbi[s, T] tugatish bosqichi
s=1
bestpath ← the path starting at state bestpathpointer, that follows backpointer[] to states back in time
return bestpath, bestpathprob

Summary

This article studied the problems of POS labeling in NLP using Hidden Markov Model and Markov Chains. POS tagging algorithms had been defined based on rules and stochastic POS tags. In addition, it was noted that the language corpus (data) should be automatically or manually POS-tagged for the POS tags of a given suggestion.

As an alternative approach to the frequency of words in a language corpus, context labels for individual words were cogitated, based on the n-gram approach to calculate the probability of occurrence of a sequence of labels. Using tag sequence probability and word frequency measurements the components of the hidden Markov model were described on the example of Uzbek lexical units. Moreover, calculations related to evident and latent states for calculating the necessary probabilities for POS tagging of Uzbek sentences using HMM were given with examples. The components of the hidden Markov model are described, using measurements of the probability of a sequence of tags and the frequency of words. The HMM matrix was decoded using the Viterbi algorithm, POS tagging of the Uzbek sentence was achieved, and the results were analyzed. Many NLP problems can be solved by executing POS tags using HMM and the Viterbi algorithm presented in this article.

REFERENCES

1. Daniel Jurafsky, James H. Martin. Speech and Language Processing. − 2023. Draft of January 7, 2023. // https://web.stanford.edu/~jurafsky/slp3/8.pdf
2. Atmakuri, S., Shahi, B., Ashwath Rao, B., & Muralikrishna, S. N. (2018). A comparison of features for POS tagging in Kannada. International Journal of Engineering and Technology(UAE), 7(4). https://doi.org/10.14419/ijet.v7i4.14900
3. Evans N. 2000. Word classes in the world’s languages. In G.Booij, C.Lehmann, J.Mugdan. Morphology: A Handbook on Inflection and Word Formation, pages 708-732. Mouton.
4. Garside, R., Leech G., McEnery A. Corpus Annotation. Longman, 1997.
5. Kumar, A., Katiyar, V., & Kumar, P. (2021). A Comparative Analysis of Pre-Processing Time in Summary of Hindi Language using Stanza and Spacy. IOP Conference Series: Materials Science and Engineering, 1110(1). https://doi.org/10.1088/1757-899x/1110/1/012019
6. Rajasekar, M., & Udhayakumar, A. (2020). POS tagging using naïve bayes algorithm for tamil. International Journal of Scientific and Technology Research, 9(2).
7. Kumawat, D., & Jain, V. (2015). POS Tagging Approaches: A Comparison. International Journal of Computer Applications, 118(6). https://doi.org/10.5120/20752-3148
8. Megyesi, B. (1998). Brill’s rule-based PoS tagger. In Change (Issue section 3).
9. Bulusu, A., & Sucharita, V. (2019). Research on machine learning techniques for POS tagging in NLP. International Journal of Recent Technology and Engineering, 8(1 Special Issue 4).
10. Cing, D. L., & Soe, K. M. (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. Jassim, A. K., & Al_Bayaty, B. F. Z. (2021). A Stochastic Approach to Identify POS in Iraqi National Song using N-Iterative HMM using Agile Approach. IOP Conference Series: Materials Science and Engineering, 1094(1). https://doi.org/10.1088/1757-899x/1094/1/012019
12. Pietrzykowski, M., & Sałabun, W. (2014). Applications of Hidden Markov Model: state-of-the-art. International Journal of Computer Technology and Applications, 5(4).
13. Stratos, K., Collins, M., & Hsu, D. (2016). Unsupervised Part-Of-Speech Tagging with Anchor Hidden Markov Models. Transactions of the Association for Computational Linguistics, 4. https://doi.org/10.1162/tacl_a_00096
14. Gao, X., & Zhu, N. (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
15. Azeraf, E., Monfrini, E., Vignon, E., & Pieczynski, W. (2021). Introducing the hidden neural markov chain framework. ICAART 2021 - Proceedings of the 13th International Conference on Agents and Artificial Intelligence, 2. https://doi.org/10.5220/0010303310131020
16. Baum, L. E., & Petrie, T. (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
17. S. Al-Anzi, F., & DiaAbuZeina. (2017). Statistical Markovian Data Modeling for Natural Language Processing. International Journal of Data Mining & Knowledge Management Process, 7(1). https://doi.org/10.5121/ijdkp.2017.7103
18. Baishya, D., & Baruah, R. (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
19. Suleiman, D., Awajan, A., & al Etaiwi, W. (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
20. Chiche, A., Kadi, H., & Bekele, T. (2021). A Hidden Markov Model-based Part of Speech Tagger for Shekki’noono Language. International Journal of Computing, 20(4). https://doi.org/10.47839/ijc.20.4.2448

21. Malik, M. K., & Sarwar, S. M. (2017). Urdu named entity recognition system using Hidden Markov Model. Pakistan Journal of Engineering and Applied Sciences, 21.
22. Mor, B., Garhwal, S., & Kumar, A. (2021). A Systematic Review of Hidden Markov Models and Their Applications. Archives of Computational Methods in Engineering, 28(3). https://doi.org/10.1007/s11831-020-09422-4
23. Samdani, R., Chang, M. W., & Roth, D. (2012). Unified expectation maximization. NAACL HLT 2012 - 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference.
24. Zhang, J., Wang, F., Zhong, Z., & Wang, S. (2018). Continuous phase modulation classification via baum-welch algorithm. IEEE Communications Letters, 22(7). https://doi.org/10.1109/LCOMM.2018.2821171
25. Wang, C., Li, K., & He, X. (2021). Network Risk Assessment Based on Baum Welch Algorithm and HMM. Mobile Networks and Applications, 26(4). https://doi.org/10.1007/s11036-019-01500-7
26. Tabar, V. R., Plewczynski, D., & Fathipour, H. (2018). Generalized Baum-Welch and Viterbi algorithms based on the direct dependency among observations. Journal of the Iranian Statistical Society, 17(2). https://doi.org/10.29252/jirss.17.2.10
27. Elov B., Sh.Hamroyeva, O.Abdullayeva, M.Uzoqova. O‘zbek tilida pos tegging masalasi: muammo va takliflar // O ‘zbekiston: til va madaniyat. Amaliy filologiya masalalari. 2022-yil 2(5) son. − B. 51-69.
28. Elov B., 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.
29. Elov B., X.Axmedova (2022). Business Process Modeling That Distinguishes Homonymy Within Three Parts of Speechs in Uzbek Language. 7th International Conference on Computer Science and Engieering, 14-16 September 2022, Turkey.

O‘ZBEKISTON MILLIY UNIVERSITETI XABARLARI, 2023
ISSN 2181-7324

MAXSUS SON
http://journals.nuu.uz

Aziza ERGASHEVA,
ESL teacher, Private School "Future Education",
Methodology in teaching English
E-mail: azizaergasheva7718@gmail.com

PRINCIPLES AND METHODS OF TEACHING ENGLISH AS A FOREIGN LANGUAGE

Abstract

This conceptual paper presents diverse approaches and strategies for preparing competent teachers who work with either English Language Learners (ELLs) or students who speak English as a Second Language (ESL). The pedagogical approaches discussed herein include practical and hands-on activities for teachers at any level. Bilingual learning improves ELL's cognitive development as well as their self-esteem. The paper outlines underlying principles for the best practices with an emphasis on ESL students and also to other learning situations and students. Teachers can modify their instructional methods to adjust ELL's learning needs. Specifically, even though the discussion is framed in the context of ESL students in U.S. classrooms, it is applicable to TEFL (Teaching English as a Foreign Language) environments in schools and other centers of learning.

Key words: English as Second Language (ESL), English Language Learner (ELL), Teaching English to Speakers of Other Languages (TESOL), bilingual learners, teaching strategies

Introduction . 1.1 Issues in Teaching English as a Second Language The increasingly diverse environment of today's classrooms provides a rich opportunity for teachers and students to engage in effective learning. With a growing number of English Language Learners worldwide, there is a critical need for general education and resource teachers to know how to effectively build and implement literacy programs that are inclusive of students' language and culture. Understanding that culture goes beyond the knowledge of ethnic attire, music, food, and language; it includes the total being, comprised of the totality of the student's background, heritage, ancestry, educational, political, and life. The importance of teaching ESL students is critical in the current climate with increasing accountability by way of student performance on standardized tests. ESL students are expected to be on grade level proficiency within three years and teachers are held accountable for their learning (Curtin, 2005).

There exist a variety of terms that have been used for non-native English speakers, ranging from LEP (Limited English Proficient), ESL students (English as a Second Language), Bilingual students and English Language Learners (ELL). For practical purposes, we use the term ESL students for a student whose mother tongue is not English. As educators, we understand that ESL students have to double their efforts in school, to not only learn new information but also learn the academic language of the school. Freeman and Freeman (2011, p.19) state, "ESLs face double the work of native English speakers. They must learn English, and they must learn academic content through English. In addition, they often live in neighborhoods where the schools are underfunded and are staffed by inexperienced teachers." On the same note, teachers of ESL students face double work of teaching core competencies enlisted in the curriculum to meet the benchmarks and teach English to non-native speakers. It is a double-whammy. While we realize that no two students are alike and that no two students have the same needs, there are commonalities among learners that help us approach our teaching in a more informed way. The paper proposes foundational principles and practices for teachers who work with ESL students in their classrooms.

1.2 School Culture and Educational Environment

It is important to consider how the culture of the school eases when a new ESL student enters into the classroom to create a sense of belonging. Using a framework of compare and contrast can be instructional and useful in learning about two cultures. There are commonalities and differences in comparing different cultures. Reaching out to parents by using a few phrases in their native language while greeting them can instantly break down the social barriers between the teachers and the parents. Now with Google Translate, it can be easily done. Creating a ies.ccsenet.org International Education Studies Vol. 12, No. 7; 2019 50 welcoming climate for students new to the country and culture provides the first step in easing into a learning situation. Seating students next to another student who has a similar background can ease the jitters caused by an alien culture and language.

Method. This conceptual paper focuses on the description of pedagogical strategies stemming from a theoretical framework that has evolved out of second language learning research. Research on ESL/ELL strategies is based on the findings that building on learners' background by providing comprehensible input and multiple opportunities for interaction is the key to second language proficiency. These findings lead to the development of a set of strategies built on the framework of principles of learning the second language outlined below. Seven principles of second language learning have been identified as critical to successfully teaching ESL students.

1) Know your student and motivation to learn the second language

2) Create a welcoming classroom environment

3) Build Background Knowledge

4) Provide Comprehensible Input by building vocabulary

5) Include frequent opportunities for Interaction and Discussion

6) Use Multiple Modalities during instruction

7) Conduct ongoing review and assessment

These principles provide a basis for developing a broader theory for second language learning. Cummins (1980) discusses the context-embedded language and its effectiveness with ESL learners. For instance, repetition of classroom routines provides non-English speakers with meaningful language learning opportunities because the words and phrases that accompany such routines are constantly repeated within a concrete context. For instance, a word like 'lavatory' will become a part of their lexicon, if used by a teacher on a routine basis every time for a bathroom break. Using synonyms or rephrasing keywords differently reinforces meaning. Creating a low-stress environment necessary for students to feel ready to participate in a larger group setting