Pindus Journal of Culture, Literature, and ELT
ISSN: 2792 – 1883 | Volume 3 No. 12
https://literature.academicjournal.io

An Approach to Distinguishing Affixes in the Development of the Uzbstemming Algorithm

Khusainova Zilola Yuladashevna
Tashkent state university of uzbek language and literature Doctoral student

Introduction

One of the main areas of NLP is morphological analysis, that dealing with the components of words. Today, more efficient various methods and algorithms have been developed and introduced to make morphological analysis [1,2,3].

At first glance, it seems that a dictionary can store all the flective forms of a word and process the language without any morphological analysis. This approach may be suitable for morphologically simple languages. However, the application of this approach to agglutinative languages, where the word can take many different forms after the combination of affixes, does not give an effective result. The necessary concepts and terms for implementing pos tagging and stemming for agglutinative languages [4] are presented in the study.

Stemming is usually done by removing prefixes and post-stem affixes from a word [5,6]. The stemming can be thought of as a "rough" heuristic process that simply cuts out the affixes of words. According to some authors, in contrast to lemmatization, the stemming process does not use a dictionary or morphological analysis [7]. The stem formed as a result of stemming may not be similar to the real word or its morphological root. In NLP, the method that determines the common form (stem) of various morphological variants of a word is called stemming algorithm [8,9].

Conducted scientific researches: Currently, many NLP researchers have developed stemming algorithms. Modern stemming algorithms are generally divided into three classes: rule-based, statistical, and hybrid algorithms.

Rule-based stemmers aim is to identify stems using non-automatic rules. Popular rule-based stemmers include Lovins [10], Porter [11,12] and Krovetz [13]. Rule-based stemming algorithms are usually supervised.

Statistical stemming algorithms use statistical methods to learn stems. Xu and Croft [14] presented a method that uses a random statistical word to overcome the shortcomings of Porter's stemmer [11]. Based on their random statistics, they implemented a graph partitioning algorithm to reduce the number of classes generated by Porter's stemmer [11].

Hybrid stemming algorithms combine rule-based and statistical methods into a single system. Some hybrid stemming algorithms were developed by Shrivastava [15], Goweder [16] and Adam [17].

A graph-based stemming algorithm was proposed by Bacchin [18]. In the first step, the algorithm splits each word into all possible split points to determine the set of substrings. In the second step, a directed graph is created using a set of substrings. Finally, the prefix and suffix scores are calculated based on the frequency of those under the line in the graph, and the stem is determined.

A stemming algorithm called YASS (Yet Another Suffix Striper) [19] was presented by Majumder, which is based on a clustering algorithm that uses a distance measure between strings. The measure of distance between lines was used to evaluate the morphological similarity between words.

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

A stemmer based on word similarity detection was developed by Peng [20]. This stemmer was used for IR tasks to improve search engine results and gave high performance.

A.Jabbar and S. Iqbal classified stemming algorithms as follows (Figure 1) [21].

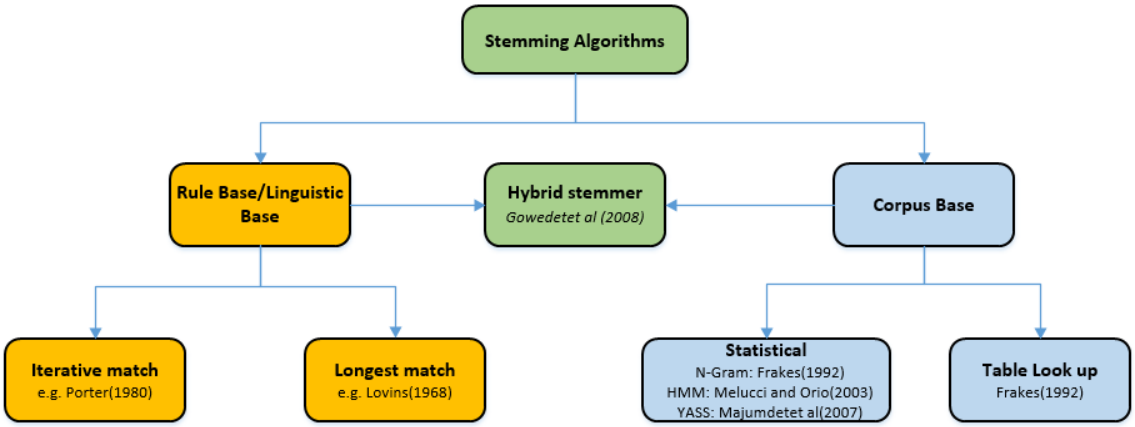

Figure 1. A. Jabbar and S. Iqbal approach to classification of stemming algorithms

Indian scientist G. Anjali divided stemming algorithms into three groups, such as truncation, statistical and mixed methods: [9]. Algorithms for each of these groups determine the appropriate stems for word variants based on specific approaches. The classification of this algorithm and methods is presented in Figure 2:

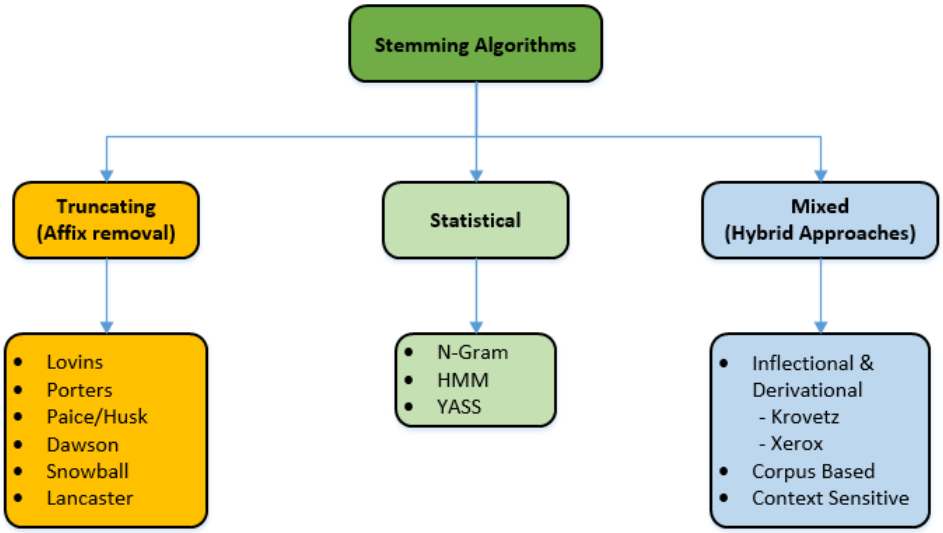

Figure 2. Classification of stemming algorithms.

Methods based on affix separation

These methods are based on truncation of suffixes or prefixes of the word. The simplest stemmer of this type is the **Tuncate(n)** stemmer, which truncates a word to the **n** character. In this case, while keeping the first **n** letters of the word, it removes the rest [22,23,24]. In this method, words shorter than n are kept in their original form. The number of redundant (unnecessary) operations increases when the word length is small. Another approach based on affix removal combines the singular and plural forms of English nouns in the S-stemmer algorithm. This algorithm was proposed by Donna Harman [25]. In the S-stemmer algorithm, rules for converting plural suffixes into singular forms have been developed [23,26].

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

In some approaches, the task of POS tagging is treated as a problem of tagging/sequencing words in a sentence. Algorithms based on such approaches often use Hidden Markov Models (HMMs) [27,28,29,30].

Methodology

In the Uzbek language, sentences are made up of separate words. Morphologically, words are formed by adding some suffixes to the root. In this process, phonetic changes (phonetic harmony) may occur in the word, and this is directly reflected in the text. The root itself can be a word that expresses its own meaning. Although affixes play an important role in the sentence, they do not have an independent meaning.

Affixes are divided into derivational suffixes and inflectional suffixes [31]. Formative adverbs change only the grammatical function of the word. A semantic change can occur in a word by adding word-forming suffixes to the stem. Form-forming adverbs cause syntactic changes in the word. Word-forming suffixes are attached to the stem first, and then form-forming suffixes. However, it is also possible to attach form-forming attachments directly to the stem (Fig. 3).

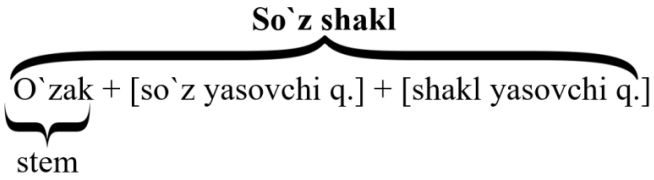

Figure 3. The general morphological structure of a word in Uzbek.

The work done so far in the field of morphological analysis can be grouped into two classes: based on **stemming** and **affix removing**. To perform stem-based morphological analysis, first the stem of the word is determined, and then its affixes. In the affix removal approach, the affixes in the word are first identified. After the affixes are removed, the rest of the word can be assumed to be the stem, or a dictionary can be used to confirm this assumption.

The number of suffixes that can be added to a word and their numerous combinations make the process of root identification in agglutinative languages a complex problem. Because in most agglutinative languages, combinations of affixes form complex word forms. Indicators on the number of additions that form new words or word forms are also different. Textbooks show that there are **228** word-forming adverbs, **69** lexical forms and **41** syntactic forms in the Uzbek language. If we include their options, in fact, this number will increase again.

To determine the stem of words in the Uzbek language, the stem and all types of affixes attached to it are determined. Traditional stemming algorithms are based on additions and some morphological rules, and the stemming process may result in uncertainty in the stem. In the process of stemming, all kinds of suffixes are usually removed from the word. But when stemming is carried out in this way, in some cases the wrong result can be obtained. The following problems may arise when determining the stem of words in the Uzbek language:

- *that the root and suffix are homonymous with one root;*
- *meeting of a word with a sound change;*
- *stemming of neologisms and NERs.*

The stem and the suffix are homonymous with one stem

Determining a polysemous stem is a more complex process, and sentence-level semantic information is not taken into account in the stemming process. Sometimes the POS tag of a word may not be the same as the POS tag of its stem.

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

Pindus Journal of Culture, Literature, and ELT
ISSN: 2792 – 1883 | Volume 3 No. 12
https://literature.academicjournal.io

POS tagging is the task of assigning (tagging) to each word form in a given sentence its category (noun, verb, adjective, number, adverb or pronoun). POS tagging can be done with or without a dictionary. A word-based approach was used in most of the scientific studies on POS tagging [32,33]. Some agglutinative languages use word stems to implement the POS tagging process [34]. Table 1 below shows examples of the lemma, stem and POS tag of a word in the Uzbek language [4,35].

Table 1. Lemma, stem and POS tag of a word in Uzbek

№	words	lemma	POS	stem	POS	root	POS
1	muzladi	muzlamoq	VB	muz	N	muz	N
2	issiqroq	issiq	JJ	isi	VB	isi	VB
3	soddalashtiriladi	soddalashtirmoq	VB	sodda	JJ	sodda	JJ
4	ixtiyoriy	ixtiyoriy	JJ	ixtiyor	N	ixtiyor	N
5	qo‘llaniladigan	qo‘llamoq	VB	qo‘l	N	qo‘l	N
6	yo‘lakda	yo‘lak	N	yo‘l	N	yo‘l	N
7	qishlog‘im	qishloq	N	qishlog‘	?	qishloq	N

In the Uzbek language, there is a phenomenon of homonymy between word-forming and form-forming suffixes [36,37,38]. This creates a problematic situation in the stemming process. Table 2 below provides a sample list of homonymous suffixes[4]:

Table 2. Homonymy between word-forming and form-forming adverbs.

Shakl yasovchi qo‘shimcha	So‘z yasovchi qo‘shimcha
- ay (lug‘aviy shakl yas.) boray	kuchay (fe’l)
- gi (lug‘aviy shakl yas.) borgim	supurgi (ot), yozgi (sifat)
- da (sintaktik shakl yas.) uyda	undamoq (fe’l)
- i (sintaktik shakl yas.) do‘sti	jannati (sifat), boyi (fe’l)
- in (lug‘aviy shakl yas.) ko‘rin	ekin (ot), sog‘in (sifat)
- im (sintaktik shakl yas.) uyim	bilim (ot), ayrim (sifat)
- ir (lug‘aviy shakl yas.) o‘chir	gapir (fe’l)
- iq (lug‘aviy shakl yas.) siniqmoq	yo‘liq (fe’l), ochiq (sifat), chiziq (ot)
- y (lug‘aviy shakl yas.) o‘qiy	qoray (sifat)

Phonetic changes of words

Phonetic changes (insertion, deletion, phonetic harmony, and assimilation) may occur in some cases as a result of adding form-forming suffixes to the last letters of the stem [4]. Uzbek words can undergo three phonetic changes, such as weakening, assimilation (Table 3).

Table 3. Increase, decrease and change of phoneme in Uzbek words.

so‘z	o‘zak	stem	turi
shahrim	shahar	shahr	1
qishlog`im	qishloq	qishlog`	2
huquqim	huquq	huquq	0
ayblov	aybla	ayblo	2
aldov	alda	aldo	2
angla	ong	ang	2
anov	ana	ano	2

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

Type: 0-exception, 1-decrease, 2-replace, 3-increase.

To solve the problem of phonetic change in the first step is stem detection, in the second step, the boundaries of the stem and suffixes are determined, and lemmatization is performed. As a result of lemmatization, the wrong stems are changed to the root in the dictionary.

A proposed solution

To develop the Uzbek language stemmer based on the removal of affixes, the corpus of the language, the base of affixes, the morpholexicon of the Uzbek language, and the root database were used [4]. given that the steps in step 2 below were followed to identify the stem.

Stage I. Preparation of data (dataset).

1. Prefixes and suffixes that can be added to the root are determined (tables 5, 6).

Table 5. Prefix suffixes in the Uzbek language.

Prefix qo‘shimchalar		
ba-	be-	ham-
bo-	no-	bar-
bad-	ser-	be-
bar-	g‘ayri-	xush-

Table 6. Suffix additions in the Uzbek language (according to POS).

POS	**so‘z yasovchi (derevatsion)**	**shakl yasovchi (flektiv)**
Ot (N)	51	17
Son (Num)	1	5
Fe`l (Vb)	30	22+10 (nisbat)
Sifat (JJ)	97	9
Olmosh (P)	-	-
Ravish (R)	18	5

2. Add prefixes and suffixes that can be added to the stems of the given word group by taking several (active) word forms of each word group and deleting their stem from the word form extra set of strings defined.

3. The string of suffixes is a combination of several Uzbek adverbs added to the stem to form this word form (Table 7).

Table 7. A set of combinations of prefixes and suffixes in the Uzbek language determined from the language corpus.

	Ot (N)	**Son (Num)**	**Fe`l (Vb)**	**Sifat (JJ)**	**Olmosh (P)**	**Ravish (R)**
1.	chiligimizning	Inchi	dilarki	almashtirgichda	nga	aydiki
2.	dagilarning	larcha	nganlaridan	bardoshligini	ngacha	aydimi
3.	doshlarimiznikidan	larda	yapsizlarmi	imizning	ngagina	chiligimizning
4.	korlarimizning	ovi	ilmayotganidan	lamaganidan	nikilarning	dangina
5.	laganlaridagina	ovimiz	ishimizdan	lashayotganligini	dek	gilarining
…	…	…	…	…	…	…
Jami:	**1773**	**229**	**15870**	**725**	**509**	**1376**

4. A special software application was used to retrieve the string of all suffixes that could be added to the stem from the language bridge.

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

5. With the help of this program, it was possible to create a database of necessary word forms based on regular expressions from the existing word forms in the Uzbek language corpus.

6. An example of the division of some word forms identified as a result of the search into prefix, stem, and suffix strings is given in Table 8:

Table 8. Separation of word forms into prefix, stem and suffix strings.

So‘z shakl	**prefiks**	**stem**	**suffiks qo‘shimchalar satri**
kitobni		kitob	ning
kitoblarning		kitob	larning
shahringdan		shahr	ingdan
shaharda		shahar	da
badavlat	ba	davlat	
berahm	be	rahm	

A set of values g = 〈p,s,q〉 was determined for all given word forms satisfying the condition $p + s' + q = w$. In this,

w - word form;

s' – stem;

p∈P;

P is a set of prefix suffixes;

s∈S, S is a set of stem word groups;

q∈Q, Q is a set of strings of suffix suffixes;

g∈G, G is a set with three parameters, each element **{p,s,q}.**

An example of determined **{p,s,q}** values is given in Table 9:

Prefiks (p)	Stem so‘z turkumi (s)	Suffiks qo‘shimchalar satri (q)
	ot	ning
	ot	larning
	ot	da
	ot	lariga
	fe’l	di
	fe’l	ysan
	ot	ga
	ot	lari
	ot	iga
	ot	ingdan
	ot	da
ba	ot	
be	ot	

After determining the values in Table 9, the stem of the word form can be determined using the UzbStemming algorithm below.

Stage 2. **UzbStemming algorithm**

The following definitions are used in the algorithm:

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

R is a set of roots;

G– a set of 〈**p,s,q**〉 values;

T – G is a set sorted in descending order by the length of p and q values of the elements of the set;

V - a set of vowel letters;

C is a set of consonant values;

w - word form;

replacelastletter(x,y) – the function to replace the last letter of the x line with the letter y;

removelastvowel(x) - the function to delete the vowel letter x in the 2nd position from the end of the line;

right(x,n) – the function to determine the nth letter from the end of the line x;

left(x,n) – the function to determine the nth letter from the beginning of the line x;

RPOS(r) is a function to determine the list of word groups of the rozak. It is determined based on the base of roots and their word groups;

$QPOS(p,q)$ A function to determine the list of word groups that can be added to the suffix string ***p*** and ***q***. It is determined based on table 9.

Enter the string **w.**

2. For each element of the set **T**

If the string **w** starts with **p** and ends with **q,** go to step **2.2,** otherwise go to step **2.6.**

Let the parts **p** and **q** of the string **w** be cut off, and we denote the rest by **r'.** In this case, the equality **p+r^'+q=w** is valid.

If **r'** contains at least one vowel, go to step **2.4,** otherwise go to step **2.6.**

If **r'∈R,** go to **step 4,** otherwise go to **step 2.5.**

For every element **r** of the set **R**

If **RPOS(r)∩QPOS(p,q)=∅,** go to **step 2.5.19,** otherwise **q1=left(q,1),** go to **step 2.5.2.**

If **right(r,1)="k"** and **q1="i",** go to **step 2.5.3,** otherwise go to **step 2.5.5.**

z=replacelastletter(r,""g").

If **r^'=z,** go to **step 4.**

If **right(r,1)="q**" and **q1="i",** skip to **step 2.5.6,** otherwise skip to **step 2.5.8.**

z=replacelastletter(r,""g'").

If **r^'=z,** go to **step 4.**

l=length(r).

If **q1="i"**and **l>4** and **right(r,1)∈C** and

right(r,2)∈V, go to **step 2.5.10,** otherwise go to **step 2.5.12.**

z=removelastvowel(r).

If **r^'=z,** go to **step 4.**

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

If **"\"Verb\""∈RPOS(r),** skip to **step 2.5.13,** otherwise skip to **step 2.5.19.**

If **right(r,1)="a"** skip to **step 2.5.14,** otherwise skip to **step 2.5.16.**

z=replacelastletter(r,"o").

If **q1∈{"v","q"}** and **r^'=z,** go to **step 4.**

If **right(r,1)="i"** skip to **step 2.5.17,** otherwise skip to **step 2.5.19.**

z=replacelastletter(r,"u").

If **q1∈{"v","q"}** and **r^'=z,** go to **step 4.**

Let **R** be the next element of the set, go to **step 2.5.1.**

Let the next element of the set **T be** taken and go to **step 2.1.**

"Stem not found". Go to **step 5.**

Stem r'.

That's it.

Applying the above UzbStemmer based on removing affixes to 100,000 sentences in the Uzbek educational corpus (http://uzschoolcorpara.uz/uz/Search) gave a result of 97.5% accuracy. In order to increase the efficiency of the UzbStemmer algorithm, I it is desirable to use a hybrid approach based on the language rules for the combination of suffixes that are not present in the group of affixes defined in the 3rd step of the stage (data preparation). There is also an opportunity to increase the size of the language corpus, to develop effective methods for identifying NER objects and neologisms, and to improve the accuracy of the stemming algorithm by eliminating the homonymy problem.

Summary

Implementing POS tagging and stemming through a dictionary is challenging for many natural language processing tasks. Using a language corpus for POS tagging and stemming solves problems with vocabulary. Various experiments on language corpora show that combining stem information with a syntactic task improves the POS tagging result for a morphologically rich language, which improves the solving efficiency of the NLP task. The article presents several different joint models that assume different dependencies in the corpus. The general experimental results are based on language rules and statistical approaches, the hybrid UzbStemmer with POS tagging using the HMM model provides high performance for stemming Uzbek texts. The results show that stemming and POS using semantic information tagging significantly improves algorithm accuracy. The authors are currently conducting research on the production of Uzbek stemmer based on N-grams and neural networks. The approach presented in this article serves as the basis for the development of stemmer based on neural networks.

List of used literature:

1. Bölücü, N., & Can, B. (2019). Unsupervised joint PoS tagging and stemming for agglutinative languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *18*(3). https://doi.org/10.1145/3292398
2. Kışla, T., & Karaoğlan, B. (2016). A hybrid Statistical Approach to Stemming in Turkish: An Agglutinative Language. *Anadolu University Journal of Science and Technology-A Applied Sciences and Engineering*, *17*(2). https://doi.org/10.18038/btda.31812

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

3. 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://link.springer.com/chapter/10.1007/978-3-031-27199-1_4*
4. B.Elov, Sh.Hamroyeva, O.Abdullayeva, Z.Xusainova, N.Xudayberganov. (2023). Agglyutinativ tillar uchun POS teglash va stemming masalasi (turk, uyg‘ur, o‘zbek tillari misolida). *Til va madaniyat, Kompyuter lingvistikasi. Vol.1(6). – B. 35-50.*
5. Eiman Tamah Al-Shammari, “Towards An ErrorFree Stemming”, in Proceedings of ADIS European Conference Data Mining 2008, pp. 160-163
6. Paice, C. D. (1990). Another Stemmer. *ACM SIGIR Forum*, *24*(3). https://doi.org/10.1145/101306.101310
7. B.B.Elov, Sh.M.Khamroeva, Z.Y.Xusainova. NLP (tabiiy tilga ishlov berish)ning pipeline konveyeri (2022).
8. Paice, C. D. (1994). An evaluation method for stemming algorithms. *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*. https://doi.org/10.1007/978-1-4471-2099-5_5
9. Anjali, M., & Jivani, G. (2011). A Comparative Study of Stemming Algorithms. *Int. J. Comp. Tech. Appl.*, *2*(6).
10. Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, *11*(June).
11. Porter, M. F. (2006). An algorithm for suffix stripping. *Program*, *40*(3). https://doi.org/10.1108/00330330610681286
12. Porter M.F. “Snowball: A language for stemming algorithms”. 2001.
13. Krovetz, R. (2000). Viewing morphology as an inference process. *Artificial Intelligence*, *118*(1–2). https://doi.org/10.1016/S0004-3702(99)00101-0
14. Xu, J., & Croft, W. B. (1998). Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems*, *16*(1). https://doi.org/10.1145/267954.267957
15. Manish Sh., Nitin A., Bibhuti M. Morphology based natural language processing tools for indian languages. *In Proceedings of the 4th Annual Inter Research Institute Student Seminar in Computer Science, IIT, Kanpur, India, April. Citeseer.*
16. A.Goweder, H.Alhami, T.Rashed, and A.Al-Musrati. *A hybrid method for stemming Arabic text. Journal of computer Science*
17. Adam, G., Asimakis, K., Bouras, C., & Poulopoulos, V. (2010). An efficient mechanism for stemming and tagging: The case of Greek language. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6278 LNAI*(PART 3). https://doi.org/10.1007/978-3-642-15393-8_44
18. Bacchin, M., Ferro, N., & Melucci, M. (2002). The effectiveness of a graph-based algorithm for stemming. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2555*. https://doi.org/10.1007/3-540-36227-4_12
19. Majumder, P., Mitra, M., Parui, S. K., Kole, G., Mitra, P., & Datta, K. (2007). YASS: Yet another suffix stripper. *ACM Transactions on Information Systems*, *25*(4). https://doi.org/10.1145/1281485.1281489

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

20. Peng, F., Ahmed, N., Li, X., & Lu, Y. (2007). Context sensitive stemming for web search. *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'07*. https://doi.org/10.1145/1277741.1277851

21. Jabbar, A., Iqbal, S., Khan, M. U. G., & Hussain, S. (2018). A survey on Urdu and Urdu like language stemmers and stemming techniques. *Artificial Intelligence Review*, *49*(3). https://doi.org/10.1007/s10462-016-9527-1

22. B.Elov, Sh.Hamroeva, Z.Xusainova, M.Samatboyeva. *Tabiiy tilni qayta ishlash (NLP)da stemming jarayoni tavsifi (2023).*

23. Frakes, W. B., & Fox, C. J. (2003). Strength and similarity of affix removal stemming algorithms. *ACM SIGIR Forum*, *37*(1). https://doi.org/10.1145/945546.945548

24. [24] Sirsat, S. R., Chavan, V., & Mahalle, H. S. (2013). Strength and Accuracy Analysis of Affix Removal Stemming Algorithms. *International Journal of Computer Science and Information Technologies*, *4*(2).

25. Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science*, *42*(1). https://doi.org/10.1002/(SICI)1097-4571(199101)42:1<7::AID-ASI2>3.0.CO;2-P

26. Sharma, A., Kumar, R., & Mansotra, V. (2016). Proposed Stemming Algorithm for Hindi Information Retrieval. *International Journal of Innovative Research in Computer and Communication Engineering (An ISO Certified Organization)*, *3297*(6). https://doi.org/10.15680/IJIRCCE.2016

27. Gao, J., & Johnson, M. (2008). A comparison of Bayesian estimators for unsupervised Hidden Markov Model POS taggers. *EMNLP 2008 - 2008 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference: A Meeting of SIGDAT, a Special Interest Group of the ACL*. https://doi.org/10.3115/1613715.1613761

28. Merialdo, B. (1994). Tagging English text with a probabilistic model. *Computational Linguistics*, *20*(2).

29. Banko, M., & Moore, R. C. (2004). Part of speech tagging in context. *COLING 2004 - Proceedings of the 20th International Conference on Computational Linguistics*. https://doi.org/10.3115/1220355.1220435

30. Haghighi, A., & Klein, D. (2006). Prototype-driven learning for sequence models. *HLT-NAACL 2006 - Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings of the Main Conference*. https://doi.org/10.3115/1220835.1220876

31. Ҳожиев А. Ўзбек тилида сўз ясалиши. − Ташкент, 2005.

32. Goldwater, S., & Griffiths, T. L. (2007). A fully Bayesian approach to unsupervised part-of-speech tagging. *ACL 2007 - Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

33. van Gael, J., Vlachos, A., & Ghahramani, Z. (2009). The infinite HMM for unsupervised PoS tagging. *EMNLP 2009 - Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: A Meeting of SIGDAT, a Special Interest Group of ACL, Held in Conjunction with ACL-IJCNLP 2009*. https://doi.org/10.3115/1699571.1699601

34. Taner Dinçer, B., & Karaoğlan, B. (2003). Stemming in agglutinative languages: A probabilistic stemmer for Turkish. *Lecture Notes in Computer Science (Including Subseries*

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), *2869*. https://doi.org/10.1007/978-3-540-39737-3_31

35. B.Elov, Sh.Hamroyeva, O.Abdullayeva, M.Uzoqova. (2022). O‘zbek tilida pos tegging masalasi: muammo va takliflar. *Til va madaniyat (Amaliy filologiya). Vol.5(4). − B. 45-63.*
36. B.Elov, K.Akhmedova. A Mathematical Model That Semantically Analyzes Polysemantic Words. (2021). *Journal of Pedagogical Inventions and Practices. P.* V3. 119-122
37. B.Elov & X.Akhmedova (2022). Business Process Modeling That Distinguishes Homonymy Within Three Parts of Speechs in The Uzbek Language. *Proceedings - 7th International Conference on Computer Science and Engineering, UBMK 2022*. https://doi.org/10.1109/UBMK55850.2022.9919453

ISSN 2792-1883 (online), Published in Vol: 3 No: 12 for the month of Dec-2023
Copyright (c) 2023 Author (s). This is an open-access article distributed under the terms of Creative Commons Attribution License (CC BY).To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/