applied sciences

MDPI

Article

Development of Intellectual Web System for Morph Analyzing of Uzbek Words

Davlatyor Mengliev [1,*], Vladimir Barakhnin [1,2] and Nilufar Abdurakhmonova [3]

[1] Department of Information Technologies, Novosibirsk State University, 630090 Novosibirsk, Russia; bar@ict.nsc.ru
[2] Federal Research Center for Information and Computational Technologies, Ac., 630090 Novosibirsk, Russia
[3] Department of Uzbek Linguistics, National University of Uzbekistan, Tashkent 700174, Uzbekistan; abdurahmonova.1987@mail.ru
* Correspondence: d.mengliev@g.nsu.ru; Tel.: +7-998937511616 or +7-9963810069

Abstract: Currently, there is an active development of the Uzbek sector of the Internet. In it, as in other national sectors, the most common form of presentation of textual information is semi-structured documents, work that presupposes the availability of reliable algorithms for text analysis, including its lexical characteristics. The article offers an intelligent web application developed for morphological analysis of words in the Uzbek language. The web application is based on the concept of generation and stem analysis of the Uzbek language word forms. A well-known Porter algorithm was chosen as the basis for stemming. The morphoanalyzer generates word forms of the Uzbek language based on the division of words into certain classes, taking into account the specifics and structure of this language. For example, nouns can be classified by meaning (related, nominal), by quantity (singular and plural), by case, and also, by the endings of belonging (possessive).

Keywords: Uzbek language; morphological analysis; natural language processing (NLP); big data; Turkic language

check for updates

Citation: Mengliev, D.; Barakhnin, V.; Abdurakhmonova, N. Development of Intellectual Web System for Morph Analyzing of Uzbek Words. *Appl. Sci.* **2021**, *11*, 9117. https://doi.org/10.3390/app11199117

Academic Editor: Rafael Valencia-Garcia

Received: 15 August 2021
Accepted: 23 September 2021
Published: 30 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

It is known that Uzbek people for centuries treated the language with great respect, carefully guarded and developed it, considering it one of the important values. After all, language is a carrier of scientific and cultural information, artistry, and expressiveness of the spiritual essence of each nation [1].

In addition, comprehensive work is being carried out to enhance the role of the Uzbek language in society [2]. It should be noted that according to the World Bank data [3], the population of Uzbekistan is over 34 millions people, which makes the country the 2nd largest among the Turkic family countries. Besides, the President of Uzbekistan signed a decree on developing the Uzbek language (2020), which focuses in the enhancement of such research areas as creation electron resources, reliable translation software systems from state language to foreign one and vice versa, and other issues, related to language technology. Taking into account all these facts, it might be said that the importance of development of the Uzbek language technology, particularly in the natural language processing (NLP) sphere, is the most crucial issue in this case.

Relevance of the Paper

Currently, there is accelerated development of the Uzbek Internet due to increasing information paths all over the world. Additionally, this, first of all, implies an increase in the amount of information that is subject to analysis and processing for further work with it. It is well known that the most popular form of presentation of textual information is semi-structured documents, work that requires the availability of reliable algorithms for text analysis, including its lexical characteristics.

In addition, within the framework of the research, a system for generating and stemming of word forms of the Uzbek language was implemented. On the basis of this development, in the near future, it is expected that a more complex and intelligent system capable of performing lexical analysis of entire research works will be created. Some algorithms used are similar for Turkic languages owing to morphologic peculiarities of the words. For instance, [4] used technology for the Kazakh language in the same direction. Comparatively, Uzbek grammar has a unique system of structures; due to the fact that this language is built on the dominance of affixes, all word connections are based on the coincidence of endings, and these endings are unified [5]. All endings of verb forms, participial forms, they are unambiguous and of the same type, and this is a ready-made model in the language [6].

2. Related Works

Nowadays, creating accurate algorithms, which can process the text correctly is considered as one of the most common problems of NLP. In fact, there is a wide variety of algorithms and tools, which are dedicated to work with such text processing tasks as translation, the generation of words, and text recognition for Turkic languages. Although, there is a few works that cover the Uzbek language in the tasks of computer linguistics; most of them provide old research results. As for relevant studies, [7] proposed an ontological model of Uzbek language as one of the most effective ways for processing texts in morphological analysis or rule-based machine translation.

In addition, algorithm synthesis of word forms by distribution on inflexion classes in the Kazakh language was suggested in [8]. Some of the best advantages of the algorithm were simplicity and flexibility in using for generation different kinds of part of speech in the Kazakh language. An algorithm might be modified to generate verbs, nouns, or even, adjectives quite easily. The authors developed a web application within their work; thanks to these applications, any users could test and use this algorithm in action.

Moreover, the basic algorithms of morphological analysis of words were studied within this [9] manuscript. As a result, the authors of the work developed an algorithm, which could translate text in Russian from pre-reform spelling to a modern one according to the morphology of words.

Although there are a few related works, we need to see more research in detail.

2.1. UzMor

UzMor is a morphological module for the Uzbek language, which includes two main functions: finding the basic form of a word (root) or all of its word forms. This module can be used in search engines to improve the search for documents with Uzbek text. A dictionary with a base of 30,000 bases is used as the initial data for the generation of word forms, which allows you to generate more than 1,500,000 word forms.

The algorithm of morphology in the process of its work carries out the following sequence of steps:

Step 1. The word is searched in the dictionary of initial forms. If the word is found in the dictionary, then the program goes to step 4.
Step 2. The word is read character by character in reverse order (starting from the end of the word) and truncated with the longest matching affixes until it truncates three affixes. Regardless of the number of truncated morphemes, the program will proceed to the next (3rd) step.
Step 3. The search for the stem of the word is performed, based on the stemming results from the previous step. If the program finds a match for the stem of a word from the stem base, then it goes to step # 4, and if not, it returns to step # 2 *.
Step 4. A dictionary is returned to the user, consisting of the stem of the word and truncated affixes.

* When the program returns to the previous step, those affixes that were used in previous iterations are excluded from the stemmatization process.

For example, the input is the word form *texnologiyalarning*; at the end of the program, we will receive root—*texnologiya*, and two affixes *lar* and *ning*. Based on the morphological class of the stem (noun) and affixes, we calculate the morphological information: *lar* <plural>, *ning* <genus case>. The UzMor interface can be seen in Figure 1.

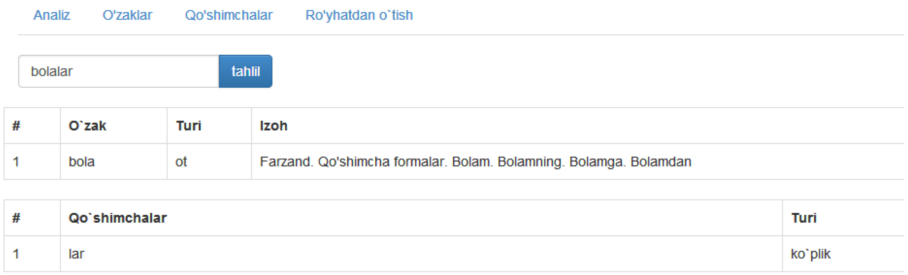

Figure 1. Detailed scheme of working process.

Unfortunately, UzMor was developed within framework of a master's thesis [10] in Uzbek language; therefore, its interface only has an Uzbek version. Summing up, in Table 1, there are the positive and negative aspects of UzMor.

Table 1. Positive and negative aspects of UzMor.

Positives	Negatives
1. The presence of a large base of bases (20,000)	1. Can only recognize and analyze nouns
2. The presence of a base of endings, which allows you to generate ~120 combinations of word forms	2. Limited set of affixes
3. Having loops to choose the best combination of affixes	3. Absence of a base of word forms—exceptions

2.2. Uz-Kaz-Nlp-Tools (UKNT)

Unlike UzMor, UKNT does not have any dictionaries or bases for storing the roots of words, although the system has a very similar stemmatization algorithm. Compared to its counterpart, UKNT has a richer base of affixes that are used in the process of truncating endings. Due to this, UKNT might recognize more parts of speech.

The algorithm of UKNT is as follows:

Step 1. Wordform is fed to the input.
Step 2. The program starts stemming the sent word from right to left (from the end). In this case, the endings are truncated as long as the program sees matches of substrings of a word with affixes from the endings base.
Step 3. The program makes stemmatizations three times, and as a result, there will be three truncated versions of the original word.
Step 4. UKNT selects the largest dictionary according to quantity of truncated morphemes and returns that as the most correct result.

To summarize, Table 2 reflects the pros and cons of the system.

Table 2. Positive and negative sides of UKNT.

Positives	Negatives
1. The presence of a base of endings, which allows you to generate ~665 combinations of word forms	1. Only works for nouns and verbs
2. The presence of 3 cycles to choose the best combination of affixes	2. Lack of a base of basics
3. Having loops to choose the best combination of affixes	3. Absence of a base of word forms—exceptions

It should be noted that UKNT was developed within the framework of a master's thesis [11]. The Uz-Kaz-Nlp-Tools interface is shown in Figure 2.

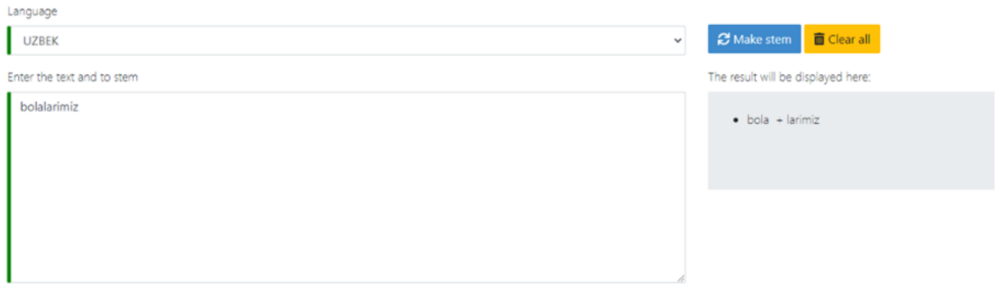

Figure 2. Positive and negative aspects of UzMor.

2.3. Suggested Solution

In this paper, a web application *UzMorphoanalyzer (UM)* is suggested as a result of the research. More details about *UM* might be found in paragraphs 5 and 6, although Table 3 shows an application's brief review of pros and cons.

Table 3. Positive and negative aspects of *UM*.

Positives	Negatives
1. The presence of a base of word stems 2. The presence of a base of endings, which allows you to generate more than 400 combinations of word forms 3. Works for all major parts of speech 4. The presence of a base of words—exceptions 5. Availability of a database of words of Oghuz Uzbek	1. Poorly copes with informal word forms

3. Rules of Word Generation

According to the morphological structure of the Uzbek language, it is an agglutinative language [12], which means that new words can be formed through adding affixes to the base (root word).

At the moment, our system is able to recognize the main parts of speech such as nouns, adjectives, verbs, and adverbs; in addition, it should be noted that the system has full rule bases of grammatical suffixes and endings (affixes) of those parts of speech. In this article, we do not study Uzbek grammar in detail, although we briefly show the main rules of each part of speech.

3.1. Form of Nouns

In the Uzbek language, there are three main grammar categories of affixes, which may be merged with root word of noun [13]:

- Quantity (singular or plural form);
- Affiliation (possessive affixes);
- Declension (case affixes).

Basically, the singular form of a noun does not have a special affix, while the plural form has the–lar ending. Take, for example, (Table 4):

Table 4. Singular and plural forms of noun.

Singular	Plural
bola (a boy)	bolalar (boys)
qiz (a girl)	qizlar (girls)
odam (a human)	odamlar (humans)

However, in some cases, the plural is not used with special words:

(1) If there is a number before noun, which indicates its quantity, for instance: o'n talaba (ten students), uch odam (three people).

(2) If a noun defines uncountable thing, such as: ko'p paxta (a lot of cotton), ko'p un (a lot of flour).

Another type of affix is the possessive one; it has 12 kinds of endings (Table 5). Actually, there are three main groups: 1st, 2nd, and 3rd forms of declensions. Additionally each of them has two forms as well—singular and plural. However, we can regroup them in another way: nouns that end with a vowel and nouns that end with a consonant.

Table 5. Twelve kinds of endings, which could be concatenated to a noun, depending on its last letter (vowel or consonant) and form (singular or plural).

The Last Letter of Noun Is Vowel		The Last Letter of Noun Is Consonant	
Singular Affixes	**Plural Affixes**	**Singular Affixes**	**Plural Affixes**
-m, -ng, -si	-miz, -ngiz, -(lar)i	-im, -ing, -i	-imiz, -ingiz, -(lar)i

Additionally, the third category of morphemes consists of case suffixes. There are six cases in the Uzbek language (Table 6), each of which (except for Nominative) has some suffixes.

Table 6. Singular and plural forms of noun.

#	Grammar Cases	Affixes
1.	Nominative	-
2.	Genitive	-ning
3.	Accusative	-ni
4.	Dative	-ga, -ka, -qa
5.	Locative	-da
6.	Ablative	-dan

According to the table, there are 11 affixes overall in grammar cases of Uzbek language. So, now, here is a common model, which is used to form nouns with different affixes (we have seen above). We denoted each kind of affix by P_i, i in this case is an index of affix (i = 1, 2, 3). On the basis of the model, a sequence of word endings was demonstrated in the example below:

Source word: bolalarimizning

Decomposition: $bola + lar(P_1) + imiz(P_2) + ning(P_3)$

As you can see, we have just analyzed a sequence of suffixes, in order to find out root word (noun). According to this model, we can calculate possible combinations of morphemes, which might be added to the root word.

The number of possible placements could be calculated by this formula:

$$S = \sum_{i=1}^{3} \prod_{j=1}^{i} P_j \quad (1)$$

In this expression, P indicates the quantity of endings, and its index i reflects one of the three main grammar categories. So, the value of each Pj looks like this: $P_1 = 2$, $P_2 = 12$, $P_3 = 11$. According to the formula, we will obtain this expression:

$$S = (2) + (2 \times 12) + (2 \times 12 \times 11) = 290 \text{ endings.} \quad (2)$$

Moreover, there some suffixes added to nominal groups (noun, pronoun, adjective, numeral):

Affiliation—*niki*
Place—*dagi*
Restriction—*gacha*
Particles—-mi, -chi, -a, -ya, -u, -yu, -ku, -da

3.2. Form of Verbs

In the Uzbek language, verbs are used in order to show the state and actions of something or someone. Take, for example: *Bola uxladi* (A child was asleep), *bizlar har kun yuguramiz* (we run every day). As you can see, the first example demonstrates a state of the boy, while second one shows an action of group of people. So, if verbs reflect actions, which have appeared due to something or people's physical activity, those verbs are called active verbs. Meanwhile, if verbs represent the inner state of objects or subjects, including changes, we call these verbs state verbs. In addition, in the modern Uzbek language, verbs are usually written at the end of the sentence.

Verbs of the Uzbek language might be changed by person, tense, quantity, and mood forms (by person, tense, number). Moreover, there are three mood forms in Uzbek:

(1) Indicative;
(2) Imperative;
(3) Conditional.

Apart from that, the indicative mood has three tenses: present, past, and future. It should be noted that in the Uzbek language, there are two kinds of present tense:

(1) Present–future tense (PF tense);
(2) Present continuous (PC tense).

Speaking of PF tense, when a verb ends with a consonant, we add an –a affix, while in case of a vowel, we concatenate an additional suffix, –*y*. The table below shows examples of two verbs in the PF tense (Table 7):

Table 7. Example of grammar rules of verb construction in the present–future tense.

#	Grammar Cases			Affixes		
			Singular Form			
	Root Word + y	**Personal Affix**	**Result**	**Root Word + a**	**Personal Affix**	**Result**
1.	yasha + y	man	yashayman *(I live\I will live)*	o'rgan + a	man	o'rganaman *(I study\I will study)*
2.	yasha + y	san	yashaysan *(You live\you will live)*	o'rgan + a	san	o'rganasan *(You study\You will study)*
3.	yasha + y	di	yashaydi *(he lives\she will live)*	o'rgan + a	di	o'rganadi *(He studies\She will study)*
			Plural Form			
	Root Word + y	**Personal Affix**	**Result**	**Root Word + a**	**Personal Affix**	**Result**
1.	yasha + y	miz	yashaymiz *(We live\We will live)*	o'rgan + a	miz	o'rganamiz *(We study\We will study)*
2.	yasha + y	siz	yashaysiz *(You live\You will live)*	o'rgan + a	siz	o'rganasiz *(You study\You will study)*
3.	yasha + y	dilar	yashaydilar *(They live\They will live)*	o'rgan + a	dilar	o'rganadilar *(They study\They will study)*

In terms of negative form, we just add –ma affix after root verb. For instance: *yasha* + *ma* + *y* + *man* = *yashamayman* (I do not live\I will not live).

So, this time, we have these values $P_1 = 2$, $P_2 = 2$ $P_3 = 6$.

$$S = (2) + (2 \times 2) + (2 \times 2 \times 6) = 30 \text{ endings.} \quad (3)$$

As for PC tense (Table 8), it is almost the same present continuous tense as in English. In addition, we can create (affirmative) verbs in this tense in three ways:

(1) Root verb (RV) + yap + personal affix (PA) = *yoz* + *yap* + *man* = *yozyapman* (I am writing).
(2) RV + moqda + PA = *o'qi* + *moqda* + *san* = *o'qimoqdasan* (You are reading).
(3) RV + a/y + yotir + PA = *yoz* + *a* + *yotir* + *man* = *yozayotirman* (I am writing).

Table 8. Example of grammar rules of verb construction in present continuous tense.

	Root Verb	**-di Affix**	**Personal Affix**	**Result**	**Translation**
			Singular form		
1.			-m	Ishladim	I worked
2.	ishla	-di	-ng	Ishlading	You worked
3.			-	Ishladi	She\He worked
			plural form		
1.			-k	Ishladik	We worked
2.	ishla	-di	-ngiz	Ishladingiz	You worked
3.			-lar	Ishladilar	They worked

In case you want to write verbs in the negative form (PC tense), the rules are the same as for PF tense. Additionally, now, we have $P_1 = 2$, $P_2 = 3$ $P_3 = 6$.

$$S = (2) + (2 \times 3) + (2 \times 3 \times 6) = 44 \text{ endings.} \quad (4)$$

The past tense of verbs is used in order to describe finished action(s).

We make negative forms in the same way. The only thing we need to do is to add a –ma affix right after the root verb. For instance, ishla + ma + di + m = ishlamadim (I did not work).

Additionally, the sum of endings in verbs consists of $P_1 = 2$, $P_2 = 1$ $P_3 = 6$.

$$S = (2) + (2 \times 1) + (2 \times 1 \times 6) = 14 \text{ endings.} \quad (5)$$

3.3. Form of Adjectives

Basically, adjectives (Table 9) describe objects or subjects, and they can be formed:

- From nouns (by adding such affixes as: *xush-*, *bad-*, *ser-*, *ba-*, *be-*, *bar*, e.g.);
- From verbs (by adding such affixes as: *-choq*, *-chak*, *-chiq*, *-gir*, e.g.);
- From merge of two words, which belong to different parts of speech:
    - Noun + noun: *sher* (lion) + *yurak* (heart) = *sheryurak* (brave);
    - Adjective + noun: *qimmat* (expensive) + *baho* (valuable) = *qimmatbaho* (precious);
    - Adverb + verb + {*-ar*}: *tez* (fast) + *oqar* (to flow) = *tezoqar* (fast-flowing);
    - Adverb + noun: *yarim* (half) + *avtomat* (automatic) = *yarimavtomat* (semiautomatic).

Table 9. Result of stemming adjectives.

#	Grammar Cases	Affixes
1.	Nominative	-
2.	Genitive	-ning

Thus, an order of endings for forming adjectives is shown in the table below.

In terms of the quantity of adjective endings, we cannot use Formula (1) in order to compute possible numbers of affixes, due to its structure. However, we manually calculated possible endings, and morphemes' quantity is around 46.

3.4. Form of Adverbs

The next part of speech, which was implemented in our morphological tool, is adverbs. Adverbs are used in order to describe actions' features. In the Uzbek language, adverbs could be formed from other words (derivative words) or be the root one. Here are some root adverbs: *ko'p* (a lot), *kech* (late), *erta* (early), *juda* (very), and so on. In English, there are regular and irregular adverbs; in the Uzbek language, it is similar, and adverbs formed from a root are irregular ones. Though, let us look at the formation rules of regular adverbs (derivative). Basically, we form these kinds of adverbs by concatenating particular affixes (*-cha*, *-chasiga*, *-dek*/*-day*, *-lab*).

The table below (Table 10) shows examples of formation adverbs by those affixes:

Table 10. Different forms of adverbs.

	Root	Affix	Result	Translation
1.	do'stlar (*friends*)	-cha	do'stlarcha	friendly
2.	qahramon (*hero*)	-chasiga	qahramonchasiga	heroically
3.	kecha (yesterday)	-dek	kechadek	as yesterday
	sen (you)	-day	senday	as you
4.	tonna (ton)	-lab	tonnalab	in tons

Adverbs also have quite a similar property of forming as adjectives. So, it is difficult to calculate numbers of all affixes. However, according to our (manual) calculation, it is around 68.

3.5. Exceptions

In 2007, A. Khajiev investigated some ways of creating new words in the Uzbek language. [14] shows that there might be some occasional words or phrases that do not follow the rules of word formation. In addition, [15] discussed different ways of creating

(new) words in the Uzbek language, though their influence has not been researched enough yet. So, there are some words that cannot be formed by adding affixes, as it was shown above. Additionally, in order to solve this problem, we made a decision to create special word sets (technically, they are implemented as dictionaries), each of which contains exceptional cases for every part of speech. Take, for example, one set that consists of only adjectives, another—(exceptional) nouns. These dictionaries contain morphologically analyzed words, so it really helps to increase the precision of our system.

Table 11 shows an example of the noun exception word set, while Table 12 shows the same with adjective.

Table 11. Noun from exception word set according to grammatical morphotactics.

Input	**Output**		
Dadamlar	**Dada**	**M**	**lar**
Word	root—noun	possessive affix	affix of plural form

Table 12. Noun from exception word set according to derivative possibility.

Input	**Output**			
Muzlatgich	**muz-**	**-la**	**-t**	**-gich**
Word	root—noun	root—verb	voice	ending, which forms noun from verb

All ready-formed (exception) words were located by order in dictionaries. It was done in order to be able use binary search algorithm, which is considered as one of the quickest search algorithms in the world [16]. Thanks to using this algorithm, we can store a large number of exception words due to its running time ($log_2 n$). Assume there are two million words in our word set. In a linear search, we would have to compare words one by one, so the system will make 2,000,000 guesses (n—guesses). While in case of a binary search, we make only 21 guesses ($log_2 n$—guesses), which will significantly save our time.

In addition, we have not included base of recognition comparative forms of adverbs and adjectives in our system yet. In the very near future, we are planning to implement a new piece of knowledge for our system in order to make it analyze more widely than now.

4. Algorithm Implementation and Testing

4.1. System Description

Taking into account the grammar rules of forming words above, we have decided to use the popular stemming algorithm (Stemmer Porter). However, the raw version of this algorithm will not work for the Uzbek language as well as for the Kazakh language [8], so we came up with the idea of using Porter's way of stemming [17] as a foundation for our own (software) system of stemming and generation of words.

Now, our system works like this:

Step 1. To start the ball rolling, a word is sent to the server.
Step 2. Firstly, the word is checked in the exception word sets, if the result is positive, the system returns the prepared answer (morphologically analyzed word) to the user. If the search's result is negative, then the system starts the stemming of the word.
Step 3. When stemming starts, the algorithm first searches for endings from right to left and, then, at the end, checks the prefixed endings (that is, from left to right).
Step 4. In this stage, the system forms dictionaries of truncated endings from each part of speech and, then, sends them to the decisive class.
Step 5. The decisive class compares the composition of the dictionaries of truncated endings. Depending on which dictionary with endings will be larger (quantity of truncated endings), the sent word will belong to that part of speech.

For example, we analyze the word *uylarimizga* ((Dative: where?) to our home).

Step 1. The word *uylarimizga* is sent to the server.
Step 2. Since there are no such words in the exception dictionaries, the system will pass it on to the next stage.
Step 3. Stemming of the word begins; the system uses specific libraries of endings, in order to perform truncation of the word. So, it finds those endings from:

- Noun endings library: uy (N (home)- root) + lar (plural) + imiz (possessive) + ga (Dative-directional case).
- Adjective endings library: nothing will be found.
- Verb endings library: uy (verb (to gather) − root) + lar (plural) + i (unknown) + miz (personal affix of plural form) + ga (unknown).
- Adverb endings library: nothing will be found.

Step 4. At this stage, the system forms a large dictionary, which contains other smaller dictionaries obtained from the previous stage (noun, adjective, verb, and adverb).
Step 5. The system starts iterating over the large dictionary obtained from stage 4. Four dictionaries with truncated endings and a word root in each one is searched. According to the calculation, the verbs' dictionary has more key-value pairs of endings, but the program only counts known endings. So, the system will choose the dictionary, which contains the biggest number of known morphemes. As a result, the number of truncated endings in the noun dictionary significantly exceeds the number of other dictionaries' endings. Thus, the dictionary of nouns is taken as the most appropriate, and the system accepts the original word *uylarimizga* as a noun with its endings.

4.2. Algorithm Implementation and Testing

We chose .NET technology in order to implement the algorithm above.

Since the grammatical rules of creation words in different types of speech are not subject to change, it was decided to write them as static strings in the project itself (inside of *.cs* files), and not as an external file (*.txt*, *.csv*, or others).

As a test, 200 words were selected (50 words from each part of speech—noun, verb, adjective, and adverb), and as a result, all 200 words were correctly processed. Some of the examples are shown in tables below (Tables 13–15).

Table 13. Morphological analysis of words (words will be stemmed).

id	Input	Output
1.	uylarimizga	"ga": "DIR", "imiz": " POSS.1PL", "lar": "PL", "uy": "root"
2.	telefoningizga	"ga": "Dative-directional case (To whom? What? Why? Where?): bolaga, kitobga", "ingiz": "2nd person, consonant endings, plural: uy-ingiz, kitob-ingiz" ,"telefon": "Root word (noun)"
3.	xushhavo	"xush": "Endings that form adjectives from nouns: xushfe'l, xushhavo", "havo": "Root word (noun)"

Table 14. Morphological analysis of words (words will not be stemmed, because they are stored in one of the exception word sets).

id	Input	Output
1.	bahorgi	“bahor”: “Root word (n. spring)”, “gi”: “Endings that form adjectives from nouns: bahorgi (adj. spring)”
2.	kelinchak	“kelin”: “ root word (bride)”, “chak”: “ this ending is used as an affectionate”
3.	dadamlar	“dada”: “root word (father)”, “m”: “ auxiliary particle”, “lar”: “ this ending is used as an affectionate, not as an plural ending”

Table 15. Morphological analysis of words. Words will not be stemmed, because there are not any endings for doing stemming of these words.

id	Input	Output
1	asal	“asal”: “Root word”“
2	David	“david”: “Root word”

To send a word to the system you need to open the weblink (http://uzmorphoanalyzer.ru/gen-of-words, accessed on 22 September 2021) and type a word you want to analyze; after that, you need to click <Analyze>. The system should show you the result of analyzing quite quickly. The result might consist of a list of morphemes with their definitions and the sent word itself.

In addition, a scheme of the system’s working process is demonstrated below (Figure 3).

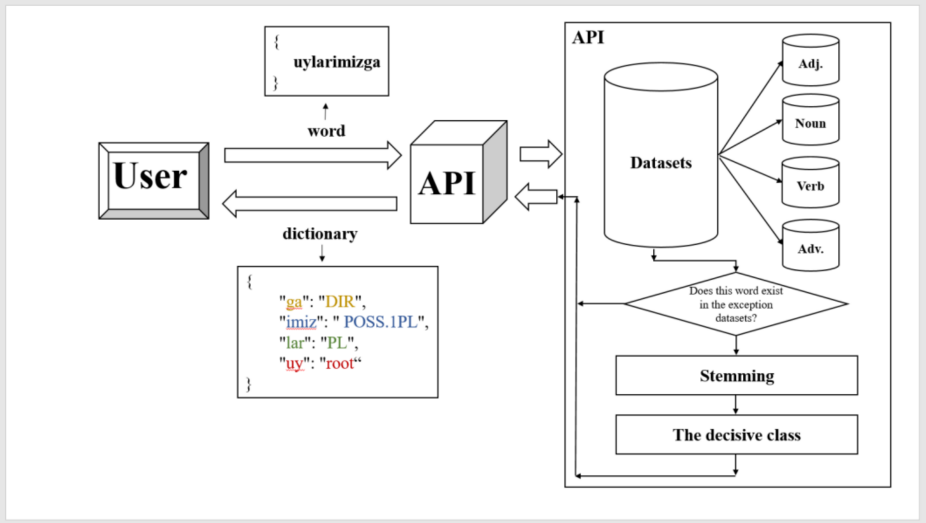

Figure 3. Detailed scheme of working process.

5. Problems and Solution

(1) Despite the fact that we made a significant effort in the framework of the research, our system has a few disadvantages. First of all, the system uses exception word sets during analyzing words; due to it is planned to add new words by experts, it is highly possible that we might not include some words, which could vitally important for certain users. For instance, our dictionaries of exception words will mostly contain words of scientific branch, while the quantity of words in such areas as literature or cooking will be noticeably less. So, cooks or writers may not be interested in our system.

As for the solution of the problem, it might be emphasized that we will gradually expand the capacity of the exception word sets. Although, this will take a long time.

(2) In the meanwhile, there is another problem of the language. Taking into account the fact that there are three families of dialects in the Uzbek language [18], some users might have made a mistake during the use of our API. The main point is that such dialects as Oguz, Kypchak, and Chagatay made a significant contribution to the formation of the modern Uzbek language. However, current official Uzbek language mostly belongs to Chagatay's dialect. Although, other groups of dialects are still being used among local people in different regions. Take, for example, in the west part of Uzbekistan (Khorezm region), people use the Oguz dialect of the Uzbek language (Khorezmian dialect [18]), while in the east part of the country, only the Chagatay dialect is used. Additionally, in the middle-north area of the state, people speak in the Kypchak dialect. In addition, the Khorezmian language was formed thanks to the influence of a wide variety of non-Turkic languages, such as Persian, Arabic, and others.

So, assume that a user from the Khorezm region sent a word to the system in order to morphologically analyze it. That user will likely make a mistake during typing a word. For instance, in Khorezm, people say *vagi'rdi*, which means "noise", while in the Tashkent dialect (official Uzbek), people would say *shovqin*. Additionally as an additional example, the word "ice" in the Khorezmian dialect is spoken as *buz* [bu:z], while in the other two dialects of the Uzbek language, it is spoken as *muz*. It should be noted that Khorezmian *buz* in the official Uzbek means "to destroy". Although Oguz dialect's vocabulary has some differences compared with the other two dialects, native speakers of all dialects can understand each other by the context of their conversation.

The only solution that we came up with is to create additional word sets in our system, which will be connected to the existing exception word sets by words themselves. So, it is supposed to be as follows (Figure 4):

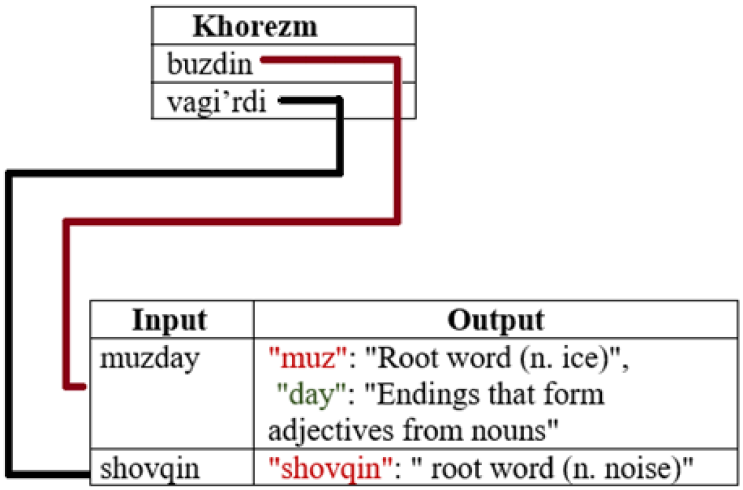

Figure 4. Final version of additional word sets.

(3) Last, but not the least, let us imagine that the user during his interaction with our system sent a word that was written grammatically incorrectly. For instance, instead of *kelinchak* ("bride"), the user typed *kalinchak*. Actually, in this case, our API is not able to recognize this word as *kelinchak*; however, we managed to find out solution! There is an algorithm that helps to analyze the string by comparing the user's word and words from the word's set, and this algorithm is called Levenshtein [19].

So, assume we sent a word, *kalinchak*, to the API, it will return to us an analyzed word *kalinchak* and a key-value pair from the dictionary, such as "Perhaps, you meant: *kelinchak*". So, in this case, we know that the system has decided to offer similar words because of Levenshtein algorithm's result.

6. Conclusions

In this paper, we studied one of the ways to process text in the Uzbek language by stemming words. While in the original stemming algorithm, words are stemmed from the end to the start, in our case, this algorithm works on both sides. Besides, we took into account specific words and created exception word sets of ready-formed words in order to increase the accuracy of our system.

The results of our research can be used as one of the most relevant contents in such university courses of Uzbekistan as computer linguistics, intellectual processing of data, natural language processing, and others. In addition, our system might be included in a module of a bigger software system, which might work for morphological or grammatical fixing sentences of the Uzbek language. Finally, our study results can be used on information search systems by service users from one of the three main Uzbek language dialects instead of the one official, as is the case now.

Finally, our project's source code was uploaded in the Github repository [20].

Author Contributions: Conceptualization, V.B. and D.M.; methodology, N.A.; software, D.M.; validation, V.B., D.M. and N.A.; formal analysis, V.B. and N.A.; investigation, N.A. and D.M.; resources, D.M.; data curation, V.B. and N.A.; writing—original draft preparation, D.M.; writing—review and editing, V.B. and N.A.; visualization, D.M.; supervision, V.B.; project administration, V.B.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jurayeva, N.; Sultanov, R.; Abdullayeva, S.; Rakhimjonova, V. Systematization of word combinations in the uzbek language. *Sci. World* **2020**, *6*, 65–68.
2. Durdona, E. Official style of uzbek language. *Int. J. Word Art* **2019**, *1*, 3–11.
3. Population, Total—Uzbekistan, Azerbaijan, Turkey, Turkmenistan, Kazakhstan, Tajikistan, Kyrgyz Republic. Available online: https://data.worldbank.org/indicator/SP.POP.TOTL?locations=UZ-AZ-TR-TM-KZ-TJ-KG&view=map (accessed on 20 September 2021).
4. Barakhnin, V.B.; Fedotov, A.M.; Bakiyeva, A.M.; Bakiyev, M.N.; Tazhibayeva, S.Z.; Batura, T.V.; Lukpanova, L.K. The software system for the study the morphology of the Kazakh language. In Proceedings of the ICPE 2017 International Conference on Psychology and Education, Moscow, Russia, 8–9 June 2017; pp. 18–27.
5. Ismatullayev, X. Samouchitel Uzbekskogo Yazyika. Tashkent: Tashkent, Uzbekistan, 1991; pp. 129–131.
6. Abdurakhmonova, N. Modeling analitic forms of verb in Uzbek as stage of morphological analysis in machine translation. *Iran. J. Soc. Sci. Humanit. Res.* **2017**, *5*, 3–5.
7. Abdurakhmonova, N. Ontological model of uzbek language (as example morphology). In Proceedings of the XV International Conference on Computational and Cognitive Linguistics TEL-2018, Kazan, Russia, 31 October–3 November 2020; pp. 5–11.
8. Barakhnin, V.B.; Fedotov, A.M.; Bakieva, A.M.; Bakiev, M.N.; Tazhibaeva, S.Z.; Batura, T.V.; Kozhemyakina, O.Y.; Tusupov, D.A.; Sambetbaeva, M.A.; Lukpanova, L.K. Algoritmi generatsii i stemmatizatsii slovoform Kazakhskogo yazyka. *Cloud Sci.* **2017**, *4*, 434–449.
9. Kozhemyakina, O.Y.; Tagirova, E.P. The translation algorithm from pre-reform spelling into modern spelling, taking into account the morphology of words. *J. Phys. Conf. Ser.* **2019**, *1405*, 1–8. [CrossRef]
10. Dusmukhamedov, U.S. Razrabotka Slovarya Fonemi i Morfem Uzbekskogo Yazyka na Osnove Informasii v Uznet (Dlya Dalneyshego Vnedrenya v Google Translate). Master's Thesis, Tashkent University of Information Technologies, Tashkent, Uzbekistan, 2018.
11. Matlatipov S.; Tukeyev U.; Aripov M. Towards the Uzbek Language Endings as a Language Resource. In *Advances in Computational Collective Intelligence*; Hernes M., Wojtkiewicz K., Szczerbicki E., Eds.; Springer: Cham, Switzerland, 2020. [CrossRef]
12. Abdurakhmonova, N.; Urdishev, K. Corpus Based Teaching Uzbek as A Foreign Language. *J. Foreign Lang. Teach. Appl. Linguist.* **2019**, *6*, 131–136.
13. Hushmurodova, S.H. Structural discordances of english and uzbek set expressions. *J. Crit. Rev.* **2020**, *7*, 383–385.
14. Turaeva, R. Linguistic Ambiguities of Uzbek and Classification of Uzbek Dialects. *Anthr. Int. Rev. Anthropol. Linguist.* **2015**, *110*, 463–475. [CrossRef]
15. Anarbaev, O. Some Aspects of Periodization of Word Formation in the Uzbek Language. Available online: http://www.rusnauka.com/pdf/280795.docx (accessed on 22 September 2021).

16. Seidl, T.; Enderle, J. Binary Search. In *Algorithms Unplugged*; Vöcking, B., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 5–11.
17. Porter, M. The Porter Stemming Algorithm. 1980. Available online: http://tartarus.org/martin/PorterStemmer/ (accessed on 22 September 2021).
18. Madrahimov, O. *Ozbek Tili Oğuz Lakhjasining Khiva Shevasi (Khiva Sub-dialect of Oguz Dialect of the Uzbek Language)*; Obdolov Regional Press: Urgench, Uzbekistan, 1999; Available online: http://www.rusnauka.com/pdf/275155.pdf (accessed on 22 September 2021).
19. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*; Russian Academy of Sciences: Moscow, Russia, 1966; Volume 10, pp. 707–710.
20. Mengliev, D.; Barakhnin, V.; Abdurakhmonova, N. Morphoanalyzer. Available online: https://github.com/shogunuz/Morphoanalyzer (accessed on 12 August 2021).