Cogent Engineering

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/oaen20

Development of the information system for the Kazakh language preprocessing

Darkhan Akhmed-Zaki, Madina Mansurova, Gulmira Madiyeva, Nurgali Kadyrbek & Marzhan Kyrgyzbayeva |

To cite this article: Darkhan Akhmed-Zaki, Madina Mansurova, Gulmira Madiyeva, Nurgali Kadyrbek & Marzhan Kyrgyzbayeva | (2021) Development of the information system for the Kazakh language preprocessing, Cogent Engineering, 8:1, 1896418, DOI: 10.1080/23311916.2021.1896418

To link to this article: https://doi.org/10.1080/23311916.2021.1896418

Published online: 10 Mar 2021.

cogent engineering

COMPUTER SCIENCE | RESEARCH ARTICLE

Development of the information system for the Kazakh language preprocessing

Darkhan Akhmed-Zaki[1,2], Madina Mansurova[3*], Gulmira Madiyeva[4], Nurgali Kadyrbek[5] and Marzhan Kyrgyzbayeva[3]

Received: 25 October 2019
Accepted: 31 January 2021

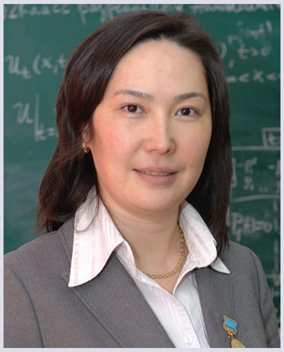
*Corresponding author: Madina Mansurova, Department of Artificial Intelligence and Big Data, Al-Farabi Kazakh National University: Madina Mansurova , Almaty, Kazakhstan
E-mail: mansurova.madina@gmail.com; madina.mansurova@kaznu.kz

Reviewing editor:
Duc Pham, School of Mechanical Engineering, University of Birmingham, Birmingham, UK

Additional information is available at the end of the article

Abstract: The aim of this work is the design and development of linguistic resources and preprocessing tools for the Kazakh language. The media-corpus of the Kazakh language is presented as a linguistic resource, which is available on Al-Farabi Kazakh National University platform. The media-corpus of the Kazakh language consists of texts of news content and is implemented as an information system. The general architecture of an information system for the automatic and reliable collection, storage and analysis of texts in the Kazakh language is described. Three automatic text preprocessing tools for the Kazakh language – word forms generator, morphological analyzer, and morphological disambiguation tool – are presented in the article. The proposed tools can also be applied in the systems of automatic analysis of texts, in creation of other linguistic resources such as thesauri and ontologies.

Darkhan Akhmed-Zaki



Madina Mansurova

ABOUT THE AUTHORS

Darkhan Akhmed-Zaki is the Doctor of Technical Sciences, Professor, Rector of the Astana IT University. He is the author of over 200 scientific papers. His scientific research relates to the organization of distributed and parallel computing, program verification and data mining. He is the scientific supervisor of more than 20 international and Kazakhstan scientific projects.

Madina Mansurova is the Candidate of Physical and Mathematical Sciences, Associate Professor, Head of the Department of Artificial Intelligence and Big Data of Al-Farabi Kazakh National University. The author of over 100 different publications, including "Vector models in Natural Language Processing" monograph. Her research interests include computer linguistics, high-performance computing, data mining.

Gulmira Madiyeva is the Doctor of Philological Sciences, Professor, Head of the Department of General Linguistics and European Languages of Al-Farabi Kazakh National University. The author of more than 200 publications, supervisor of the scientific project "Almaty corpus of the Kazakh language".

PUBLIC INTEREST STATEMENT

Development of the language resources allow to solve wide range of natural language-related problems. Text preprocessing is an important task and essential stage in text mining. The Kazakh language belongs to the class of agglutinative languages of Turkic language family. The Kazakh language is one of the low-resourced languages and the most challenging issue with these languages is the difficulty of obtaining enough language resources. The aim of this work is the design and development of the Kazakh language media-corpus presented as a linguistic resource. The media-corpus of the Kazakh language consists of texts of news content from official news portals and websites of Republic of Kazakhstan and is available on Al-Farabi Kazakh National University platform. Three automatic text preprocessing tools for the Kazakh language media-corpus—word forms generator, morphological analyzer, and morphological disambiguation tool are presented in the article.

Subjects: Computer science; engineering & technology; information science

Keywords: Corpus linguistics; morphological parsing; Kazakh language; preprocessing tool; architecture of information system

1. Introduction

In the age of information technology and exponential growth of digital data, the need to develop linguistic resources and effective tools for natural language processing (NLP) is increasing. There is a large number of different resources and tools for such languages as English and Russian. The situation with the text processing in Kazakh is more complicated. Nowadays, Kazakh refers to the category of less-resourced languages, as there is a small number of linguistic resources accessible to a wide range of users. One of the important and complex linguistic resources is a text corpus or a union of vocabulary resources of the language, such as newspaper articles, literary works, legal documents and other texts electronically stored and processed.

A text corpus is a collection of texts that:

(1) is selected on a definite specific basis (one language, genre, period of creation, etc.);
(2) is marked up in a special way (highlighted in clichés, terminology, sustainable design, etc.).

A corpus contains a special markup, which is additional information about the properties of texts included in it. Markup is the main characteristic of the corpus; it distinguishes the corpus from simple text collections. The richer and more diverse the markup, the higher is scientific and educational value of the corpus. The Russian National Corpus with the volume of more than 600 million words can be taken as an example of a well-developed corpus. The Russian National Corpus covers, first of all, the period from the middle of the eighteenth to the beginning of the twenty-first century (National corpus of the Russian language, 2019).

The Kazakh language refers to the class of agglutinative languages and together with Uzbek, Kyrgyz, Bashkir, Tatar, Azerbaijani, Turkish and other languages forms a Turkic linguistic family. Agglutinative languages are characterized by a consecutive addition of suffixes or endings bearing a grammatical meaning to an unchangeable root or stem having a lexical meaning. The existing corpora of Turkish languages include:

(1) Turkish national corpus with 50 million tokens, which is one of the most widespread and influential structures of modern Turkish. It consists of samples of text data in a wide variety of genres covering a period of 20 years (1990–2009) (Turkish National Corpus, 2019).
(2) Bashkirian poetic corpus with more than 1.8 million tokens. It is the world's second poetry corpus. Its peculiarity is in the fact that it consists of the works of Bashkir poets of the twentieth and beginning of the twenty-first century (Bashkir poetry corpus, 2019).
(3) The written corpus of the Tatar language currently stands at more than 356 million words (430 million tokens), the number of different word forms is about 4.5 million (Corpus of Written Tatar language, 2019).

The size of the minimum context for the Tatar language remains to be investigated. Nevertheless, there is reason to believe that a certain rigor of the syntactic structure will allow us to count on the detection of clear contextual restrictions in the immediate context (Hakimov et al., 2014).

Despite the difficulties, it can be stated that for the English, Russian and Turkish languages, the problem of resolving morphological polysemy has been largely solved. Using various add-ons over algorithms (or increasing the training sample for statistical methods), the accuracy of resolution methods can be brought to a level of at least 97%. The typological and genetic similarity of the

Turkish and Tatar languages gives reason to believe that these methods are capable of giving acceptable results for the Tatar language.

For English, which has a non-poor morphology, the problem of resolving morphological polysemy, as a rule, boils down to resolving polysemy at the level of parts of speech (POS-tagging), which, in turn, significantly complicates the task. In agglutinative languages, such as Turkish, Hungarian and Tatar, morphemes are added to the stem of the word, which, in addition to semantics, determine syntactic relations. Morphological polysemy in these languages is manifested in various forms. In some cases, both syntactic and semantic analysis may be required to resolve morphological polysemy (Gataullin, 2016).

During the research, there were attempts to combine the ideas of the contextual and statistical-probabilistic approach.

The issue of the minimum resolving context was also relevant for the problem under study. In this regard, the results obtained by A. Caplan (Caplan, 1955) in the study of the minimal resolving context deserve attention. The work analyzed 140 polysemantic common English words (mainly lexical homonyms), which were in various contextual conditions. The author has identified the following types of contexts:

combination with the preceding word—P1;
combination with the subsequent word—F1;
combination with the preceding and subsequent words—B1 (both);
combination with two preceding words—P2;
combination with two subsequent words—F2;
combination with two preceding and two subsequent words—B2;
the whole sentence—S (sentence)

To resolve morphological ambiguity based on contextual rules in the Tatar language, the Applied Semiotics Research Institute of the Academy of Sciences of the Republic of Tatarstan has created a software toolkit for developing and testing contextual rules. The first results of experiments on the construction of contextual rules have shown the efficiency of the method, however, additional studies are required for final conclusions (Gataullin & Gil'mullin, 2016).

Almost all of these approaches have been tested in many Turkic languages (Constant et al., 2017; Eryiğit et al., 2019; Eryiğit & Torunoğlu-Selamet, 2017; Sak et al., 2008; Sulubacak & Eryiğit, 2018; Tunali & Bilgin, 2012), but unfortunately, the Kazakh language is scarce and does not have an accessible corpus marked with a resolution of homonymy, so at this stage we will try to get around this problem: the Kazakh language is projective and has a rigid structure of words in sentences, and this will allow us to partially solve the problem using regular patterns. The template implies the construction of a sentence with morphological components, and at this stage we do not rely on semantics.

The experience of developing corpuses of the Turkic languages has positively influenced the development of the corpuses of the Kazakh language (Assylbekov et al., 2016; Makhambetov et al., 2013; Myrzakhmetov & Zh, 2018; Turganbayeva & Tukeyev, 2020). The Kazakh language corpus, which was posted on the official language portal of the Committee for Languages of the Ministry of Culture and Information of the Republic of Kazakhstan, can be referred to the previously developed, but currently inaccessible corpuses of the Kazakh language; an Anglo-Kazakh parallel corpus based on legal texts; Kazakh National Corps.

Thus, the expediency of creating corpus is reinforced by the importance of information that will be stored and processed in it. These data can be useful not only for philologists, but also to carry out statistical analysis and solve NLP problems (Kudo & Richardson, 2018; Kuriyozov et al., 2020; Petrovic & Stankovic, 2019; Said et al., 2009).

The second section consists of two parts, the first one describes the media-corpus architecture, and the second one describes the development of preprocessing tools.

2. Method of research

2.1. Media-corpus information system architecture

There are many tools for processing large amounts of data and the choice of each of them depends on the specific task. In our case, we will work with a large amount of text data with meta-information. For the task of data storage, it is important to find a solution that would allow to accumulate text data for their processing in future, with an increase in data volumes, could maintain stable operation, had simplified access to data and was reliable. The developed storage architecture should take into account such important properties as database structure flexibility, system scalability, system fault tolerance, high data availability.

The work (Pokorný, 2016) is devoted to studying the issue of storing and analyzing large amounts of data. It gives an idea of the modern approach of interaction with data and considers the features of different types of data management systems. The author indicates that in NoSQL database management system the simplification of the data model, the lack of a standard query language, especially the weakening of the ACID semantics are the most relevant in the context of processing large amounts of data. Therefore, for the accumulation task, it is advisable to select the NoSQL database type.

In (Han et al., 2011), the authors detail the NoSQL database management system. The main advantages are highlighted such as easiness of management, providing a flexible data model, high availability, high scalability and fault tolerance. The article discusses various approaches to the organization of non-relational database management systems:

1)Key-value storage;

2)Document-oriented storage;

3)Distributed storage.

The example of a key-value data store is Amazon's Dynamo, Project Voldemort. This type is not suitable for our tasks, because we need to store the whole document. The examples of document-oriented repositories are SimpleDB, CouchDB, and MongoDB. These types are the most appropriate because databases of these types are targeted at storing full-text data. SimpleDB is a service provided by Amazon. The examples of distributed storage are Big Table, HBase, Cassandra. We need an open-source database management system for our purposes to host on your own servers.

The article (Bhardwaj, 2016) is a direct comparison of NoSQL databases CouchDB and MongoDB. The author concludes that in MongoDB the document saving speed is about 10 times faster than in CouchDB under the same conditions. It is also possible to highlight the fact that the CouchDB database can be interacted with via HTTP, which provides high availability of data, but still the standard functionality for interacting with data is not enough for full operation.

The article (Nevzorova et al., 2017) describes the architecture of the data storage system for the national corpus of the Tatar language. Modern methods of data storage and extraction are compared. The authors of the article have done a lot of work, testing the reading and writing time of each instrument. As a result of their work, a bunch of Redis + MySQL was chosen. Thus, MySQL database management system is responsible for data storage, and Redis database management system is used for high-performance interaction with data. The choice was predetermined by their goal to create a system immediately with the ability to use quick queries. This option is not suitable for our purposes, because our main task at the current stage is accumulation of data. The results of this article can be used in future, when our tasks will need to use a high-performance database management system for complex queries among a large amount of data. At the current stage, it is optimal to use NoSQL database MongoDB as a database for the storage system.

Therefore, the architecture you are developing must take into account the important properties described earlier. At the moment, the low-level storage architecture has been developed using NoSQL database management system MongoDB with application programming interface (API). The API allows to interact with databases without connecting directly to them, eliminating problems with low-level interaction with MongoDB database nodes. Fault tolerance and scalability can be achieved using replication. Replication is a mechanism for synchronizing databases and providing read scalability. Scalability on a read operation implies that data can be obtained not only from the main database with a record in it, but also from database replicas. As a result, the total load on data reading is distributed between these databases.

Figure 1 shows the general concept of the data warehouse core in the Kazakh language.

Figure 2 shows the data warehouse architecture. Interaction with the data occurs with the help of the Application Programming Interface (API). This means that it is not necessary to connect directly to databases to retrieve, delete, or modify data. It is also shown how the API can be used, it is seen that the data warehouse can interact with many user interfaces and web applications, depending on the tasks.

It is worth noting that the programming languages and frameworks that can be used in web applications and user interfaces can be different. This is done by reducing the data format to a unified view and exchanging information in JavaScript Object Notation (JSON) form—a text-based data exchange format. Thus, the data model must be defined in the data warehouse itself and in the client using the storage.

We should also mention the NoSQL database management system MongoDB, which was used in the data warehouse. Figure 2 shows that the API interacts with two MongoDB database nodes: Primary and Secondary. This is necessary for a replication mechanism, which ensures data safety. Moreover, an additional configuration was made: data are written to the Primary node, and reading occurs with the secondary one. This solution reduces the load on the primary database node.

The API itself is implemented using the popular framework module – a set of ready-made libraries, Spring Boot in conjunction with the Data Rest module. The standard functionality includes the addition, deletion and modification of data using HTTP Protocol, data exchange Protocol. You can create services that will be designed for various purposes in the API as well. In addition to the standard functionality of adding, saving, updating, it is also possible to create services. In comparison with the standard functionality, all the same, except that the services can be written for any specific tasks. For example, you can create a service that will display the last saved document or

Figure 1. General model of data transferring.

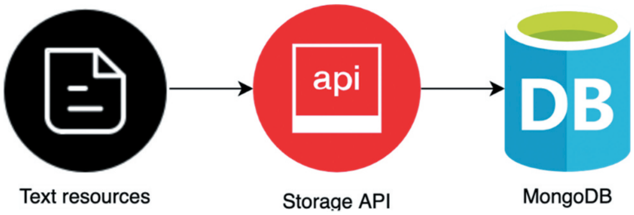
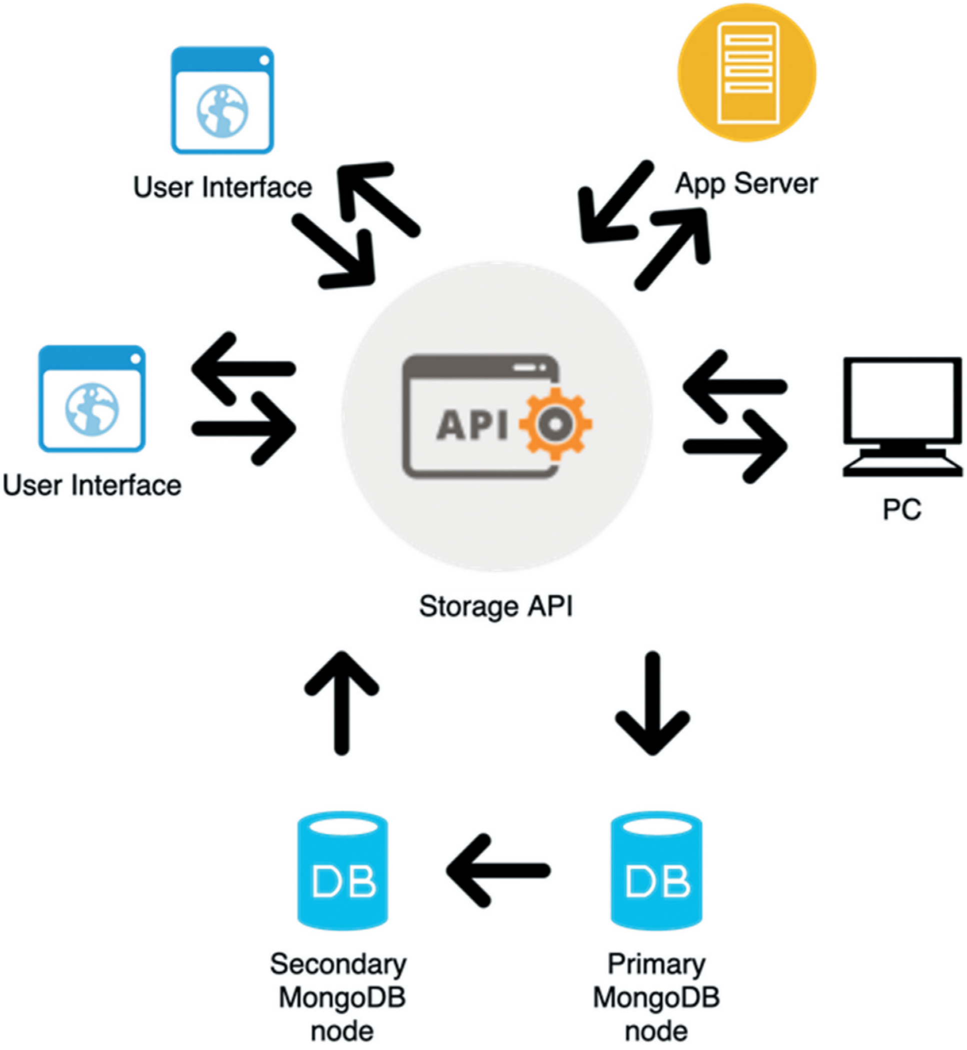


Figure 2. Main architecture of corpus prototype.



the last five documents, it all depends on the task. API is implemented in Java programming language.

The reason that the user interface is separated from the web application is in the fact that the web application receives and interacts with data less frequently than it does with the user interface. For example, for a certain analytics that is run only once, the data will be downloaded from the storage, and a whole service will not be created in the API, which will then no longer be used. This approach minimizes the risk of creating unnecessary functionality.

Figure 3. Example of transferring JSON to UI.

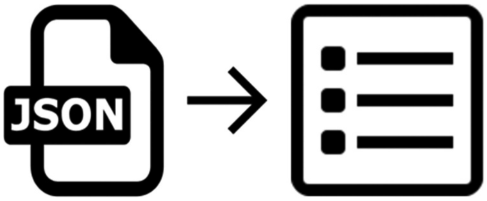


Figure 3 shows how data in JSON format is converted to the user interface using the AlpacaJS library. This reduces development time and allows to focus on other tasks.

Thus, we chose to create our own tools for working with texts using ready-made libraries. This gives you more flexibility and independence from third-party tools for working with cases. The analysis was carried out on the choice of database for storing text data. The final result was a modular architecture for storage and moderation of text data for solving problems in computational linguistics. With this approach, you do not need to interact with the data at the lowest level, but still have the possibility to solve their local problems. The purpose of the created system is accumulation of data and their intermediate processing. The modularity of the architecture and system was designed with understanding that there would always be easy access to data. This is solved by the created API using the HTTP Protocol. The implemented system uses the JSON data format, in any case, conversion to XML will be a trivial task when using ready-made tools.

2.2. Media-corpus data modeling

During the study, we developed the structure of text resources. Modeling the structure of stored data is an important part because design errors can make interaction with the data difficult. This is another reason why the MongoDB database management system is great for our purposes. We can easily change data structures. Figure 4 shows a fragment of a lemma entity.

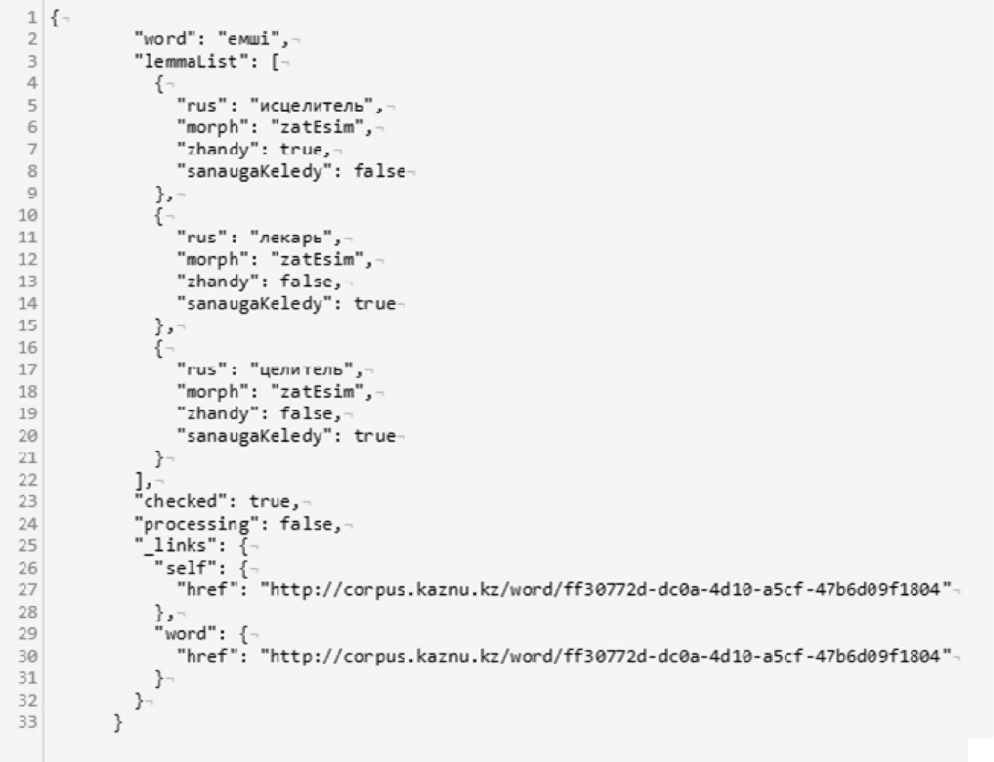
word—word/lemma

rus—equivalent lemma in Russian

morph—part of speech

zhandy—animate or inanimate

sanaugaKeledi—countable and uncountable



```
{
        "word": "емші",
        "lemmaList": [
          {
            "rus": "исцелитель",
            "morph": "zatEsim",
            "zhandy": true,
            "sanaugaKeledy": false
          },
          {
            "rus": "лекарь",
            "morph": "zatEsim",
            "zhandy": false,
            "sanaugaKeledy": true
          },
          {
            "rus": "целитель",
            "morph": "zatEsim",
            "zhandy": false,
            "sanaugaKeledy": true
          }
        ],
        "checked": true,
        "processing": false,
        "_links": {
          "self": {
            "href": "http://corpus.kaznu.kz/word/ff30772d-dc0a-4d10-a5cf-47b6d09f1804"
          },
          "word": {
            "href": "http://corpus.kaznu.kz/word/ff30772d-dc0a-4d10-a5cf-47b6d09f1804"
          }
        }
      }
```

Figure 4. Fragment of lemma entity.

checked—checked, valid lemma

processing—in the process of checking

By the way, the last two attributes appeared in the process of manual validation of lemmas by philologists using the API, they determine the status of the state in the current moment.

If the lemma has several Russian translations, then all options will be placed in the form of a list with the corresponding parts of speech and attributes.

Analysis tools allow us to add meta-information to our data. Meta-information will be very useful in further research. Figure 5 shows how the marked sentence should look like.

Examples of resources obtained in the course of operation can serve as an example of effective use of the system. The opportunities provided by the prototype of the Kazakh language corpus cover the basic requirements for the corpus. This is all the result of a study of the data format and tools with which efficient data storage is organized. The results obtained in the course of the study show that the use of technology has a positive impact on the speed of creation and support of the prototype of the Kazakh language corpus.

Figure 6 shows part of the interface for manual addition of a new word to the corpus. The user interface is created using the Alpaca JS library.

2.3. Development of the preprocessing tools

The first task is creation of an automatic Kazakh word forms generator tool. To solve the task, the lexical and grammatical features of the Kazakh language were analyzed, on the basis of which the composition of the fundamental rules for changing the words of Kazakh was determined. Based on the use of Kazakh grammar and the examination of philologists, the following results were obtained: up to 75 word forms can be generated for nouns, 96 forms for adjectives, 319 forms for verbs. The results of research concerning these tools are presented in (Mansurova et al., 2019).

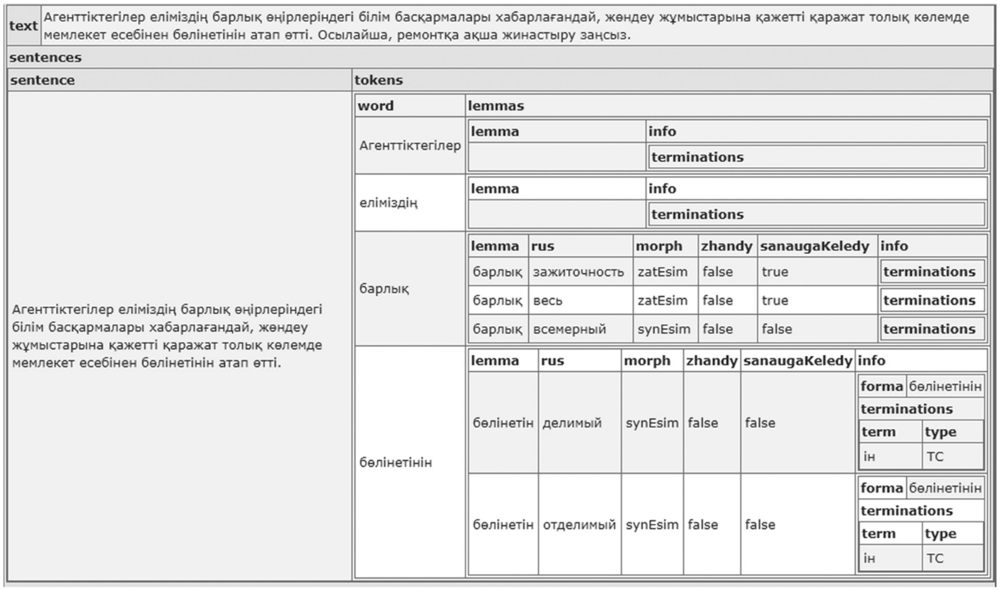
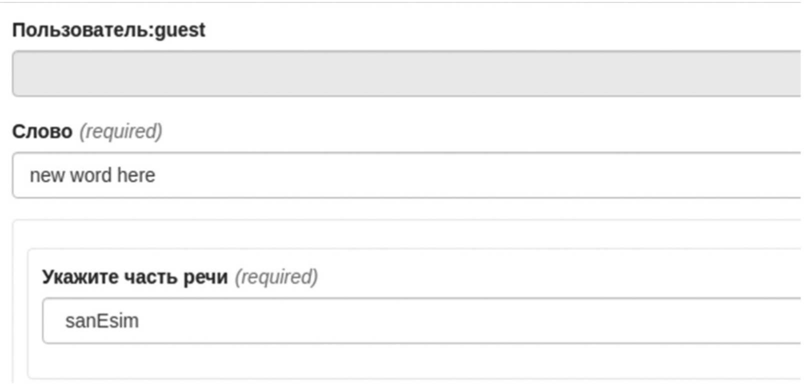


Figure 5. Fragment of a sentence with meta-information.

text Агенттіктегілер еліміздің барлық өңірлеріндегі білім басқармалары хабарлағандай, жөндеу жұмыстарына қажетті қаражат толық көлемде мемлекет есебінен бөлінетінін атап өтті. Осылайша, ремонтқа ақша жинастыру заңсыз.

sentences

sentence	**tokens**						
	word	**lemmas**					
		lemma	**rus**	**morph**	**zhandy**	**sanaugaKeledy**	**info**
Агенттіктегілер еліміздің барлық өңірлеріндегі білім басқармалары хабарлағандай, жөндеу жұмыстарына қажетті қаражат толық көлемде мемлекет есебінен бөлінетінін атап өтті.	Агенттіктегілер						**terminations**
	еліміздің						**terminations**
	барлық	барлық	зажиточность	zatEsim	false	true	**terminations**
		барлық	весь	zatEsim	false	true	**terminations**
		барлық	всемерный	synEsim	false	false	**terminations**
	бөлінетінін	бөлінетін	делимый	synEsim	false	false	**forma** бөлінетінін **terminations** **term** **type** ін ТС
		бөлінетін	отделимый	synEsim	false	false	**forma** бөлінетінін **terminations** **term** **type** ін ТС



Пользователь:guest

Слово *(required)*

new word here

Укажите часть речи *(required)*

sanEsim

Figure 6. Fragment of UI generated via Alpaca JS.

The second task is creation of an automatic morphological analyzer which allows to identify of parts of speech and extract of lemma. For automatic extraction of lemmas from word forms and a deeper analysis of the composition of words, two approaches were implemented:

(1) The direct approach is the task of normalizing words based on Porter's algorithm according to the grammatical rules of the Kazakh language.

(2) The reverse approach is developed using a dictionary of word forms.

The direct approach is the task of normalizing words. Based on the Porter algorithm, according to the grammatical rules of Kazakh, affixes are sequentially cut off from the end of the word: that is how stemming is performed. To do this, the help of philologists, dictionaries of Kazakh affixes, namely, plural endings, endings of the possessive form, case and personal endings for nouns, which are also used for adjectives, in addition to specific adjectives for various adjectives and personal endings for verbs, suffixes and participles affixes are used. Furthermore, normalization is performed on the basis of results of stemming, synthesis of normal form is carried out. This stage consists of research of the lemma dictionary, meant to find the closest lemma to the derived word basis. The system can return several options of lemmas. At the moment, there is no algorithm that would allow to determine the lemma uniquely. Therefore, this work is carried out with the assistance of philologists. An online interface has been developed for convenience and productivity (http://corpus.kaznu.kz/). The reverse approach is carried out using a dictionary of word forms. For each lemma from the created dictionary of lemmas, various word forms were generated and stored in a database together with the corresponding meta-information (morphological features of the word).

The third task is creation of a morphological disambiguation tool. The theoretical aspect of the work was previously determined by analyzing the works of specialists in general linguistics and using recommendations of specialists of the Kazakh language (Assylbekov, Washington, Tyers et al., 2016; Bekmanova & Sharipbay, 2017; Koibagarov et al., 2013, 2014; Mansurova et al., 2016, 2017). Approaches to the morphological disambiguation are divided into deterministic, based on syntactic analysis and syntactic dictionaries, and probabilistic, which use statistics of the co-occurrence of grammatical signs of words in large corpora with morphological disambiguation.

In this paper, the first approach was applied. Table 1 provides a system of numeric indices and unified tags for parts of speech and morphological features. Numeric indices are assigned heuristically and are identifiers of various parts of speech and their forms. Using Table 1, the syntactic construction of a sentence can be represented in a vector form, in which the numerical index of an individual word is formed as the sum of index of the part of speech and the index of corresponding form of the word.

Table 1. Numeric indices and unified tags for morphological characters

Numeric index	Title in English	Unified tag
10	Noun	NOUN
20	Adjective	ADJ
30	Adverb	ADVB
40	Pronoun	PRON
50	Numeral	NUMR
50	Verb	VERB
−50	Preposition	PREP
Personal endings		
7	Personal 1 singular	PERS.1SG
	Personal 2 singular	PERS.2SG
	Personal 3 singular	PERS.3SG
	Personal 1 plural	PERS.1PL
	Personal 2 plural	PERS.2PL
	Personal 3 plural	PERS.3PL
Possessive endings		
8	Possessive 1 singular	POSS.1SG
	Possessive 2 singular	POSS.2SG
	Possessive 3 singular	POSS.3SG
	Possessive 1 plural	POSS.1PL
	Possessive 1 plural	POSS.2PL
	Possessive 1 plural	POSS.3PL
	Case endings	
0	Nominative	NOM
1	Genitive (whose?)	GEN
2	Dative	DAT
3	Ablative	ABL
4	Locative (where?)	LOC
5	Accusative (whom?)	ACC
6	Instrumental	INS
Participles and gerunds		
1	Converbs	Conv
5	Participles	Parti
0	Unknown word	X

The idea of the method used in the study is to subordinate the structure of sentences to certain general laws. Despite the large variety of semantic parts of sentences, the structure of sentences is limited and can be represented as a set of mathematical patterns. Consequently, when analyzing sentences, it becomes possible to access the tool, which is a mathematical pattern of the sentence structure. We tried to cope with the morphology of the Kazakh language by considering the structure of simple sentences. As an advantage of the method, various statistical indicators can be easily integrated as a determining parameter and show the possibility of improving the quality of results.

Using Table 1, the syntactic construction of a given sentence can be represented in a vector form.

Let Ψ = {Ψ_1, Ψ_2, …, Ψ_n} be a finite set of sentence constructions, where $\Psi_i, i = \underline{1, n}$ is the i-th sentence construction. For example, for the sentence {"*oқushylar*", "*Kargany*", "*keshe*", "*korgen*", "*edi*"} the correct construction is presented in the form {noun, noun + ending of accusative case, adverb, verb + ending of participle, auxiliary verb}. This design corresponds to a large number of proposals, for example, *Adamdar awıldı qattı sağıңğan edi. Ustaz oqwşını jaqsı oqıtqan edi*. etc.

Let $\Theta = \{\Theta_1, \Theta_2, \ldots, \Theta_m\}$ be possible variants of the syntactic structure of a given sentence, presented in a vector form based on the notation from Table 1. The j^{th} vector $\Theta_j = \underline{1, m}$ may have one or more undefined coordinates.

Comparison of vectors Ψ_i and Θ_j, each of which consists of N elements, allows us to find the most probable version of the correct construction of the sentence. We used a measure of cosine similarity between two vectors to estimate the proximity. Figure 7 shows a scheme of the disambiguation method.

The pattern $\Psi_i$satisfying the condition $Max(cos(\Psi_i, \Theta_j))$, $i = \underline{1, m}$, $j = \underline{1, n}$, is the desired pattern corresponding to the construction *j*.

For the sentence {"*okushylar*", "*kargany*", "*keshe*", "*korgen*", "*edi*"} according to the scheme (Figure 7), *n* structures of length *N* are extracted from the structures built on the basis of the rules and the cosine similarity measure between them and the rows of the matrix of possible structures are calculated (Figure 8).

From the data obtained it follows that the vector {10 13 60 55 − 50} shows the maximum value. Therefore, the correct syntax is the following:

FullInfo{word = ' *okushylar* ', lem = "*okushy*", affix = [Affix{content = "*lar*", type = "Plural"}], morph = ze}

FullInfo{word = " *kargany*", lem = "*karga*", affix = [Affix{content = "*ny*", type = "TC"}], morph = ze}

FullInfo{word = ' *keshe* ', lem = ' *keshe* ', affix = [], morph = usteu}

FullInfo{word = ' *korgen* ', lem = " *kor*", affix = [Affix{content = "*gen*", type = "Parti"}], morph = et}

FullInfo{word = ' *edi* ', lem = ' *edi* ', affix = [], morph = komekshi}

Figure 7. Scheme of the disambiguation method.

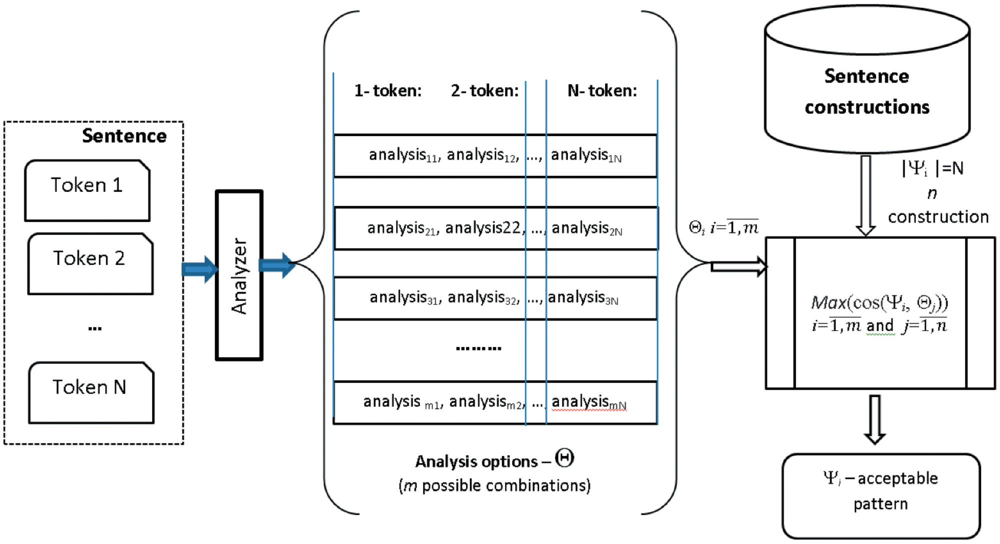
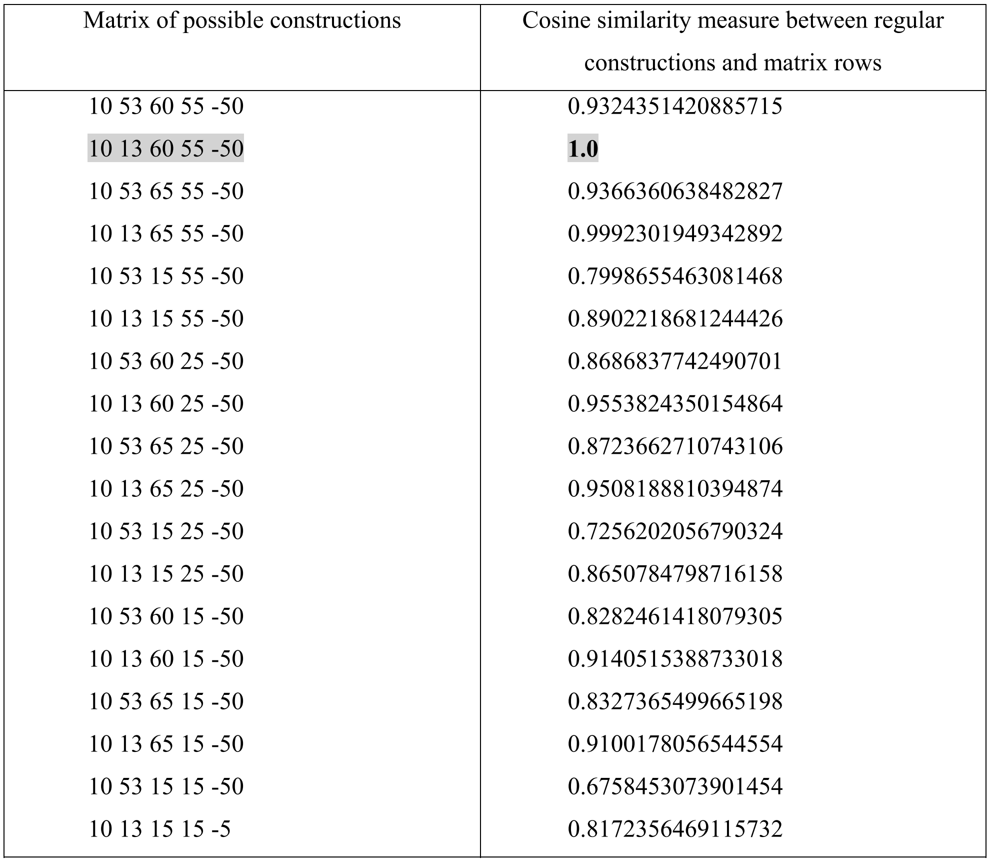

Sentence
Token 1
Token 2
...
Token N
Analyzer
1- token:
2- token:
N- token:
analysis$_{11}$, analysis$_{12}$, ..., analysis$_{1N}$
analysis$_{21}$, analysis22, ..., analysis$_{2N}$
analysis$_{31}$, analysis$_{32}$, ..., analysis$_{3N}$
………
analysis $_{m1}$, analysis$_{m2}$, ..., analysis$_{mN}$
Analysis options – Θ
(*m* possible combinations)
Θ_i $i=\overline{1,m}$
Sentence constructions
$|\Psi_i|$=N
n
construction
$Max(\cos(\Psi_i, \Theta_j))$
$i=\overline{1,m}$ and $j=\overline{1,n}$
Ψ_i – acceptable pattern

Figure 8. Cosine similarity measure between regular constructions and matrix rows.



Matrix of possible constructions	Cosine similarity measure between regular constructions and matrix rows
10 53 60 55 -50	0.9324351420885715
10 13 60 55 -50	**1.0**
10 53 65 55 -50	0.9366360638482827
10 13 65 55 -50	0.9992301949342892
10 53 15 55 -50	0.7998655463081468
10 13 15 55 -50	0.8902218681244426
10 53 60 25 -50	0.8686837742490701
10 13 60 25 -50	0.9553824350154864
10 53 65 25 -50	0.8723662710743106
10 13 65 25 -50	0.9508188810394874
10 53 15 25 -50	0.7256202056790324
10 13 15 25 -50	0.8650784798716158
10 53 60 15 -50	0.8282461418079305
10 13 60 15 -50	0.9140515388733018
10 53 65 15 -50	0.8327365499665198
10 13 65 15 -50	0.9100178056544554
10 53 15 15 -50	0.6758453073901454
10 13 15 15 -5	0.8172356469115732

2.4. Analysis of text resources

The Kazakh language belongs to the class of agglutinative languages, which are characterized by the sequential attachment of various formative suffixes or endings carrying grammatical meaning to an unchangeable root or stem that are carriers of lexical meaning. The order of adding affixes is strictly defined. When developing the algorithm of the morphological analyzer, the rules of word formation of the Kazakh language were applied. The morphological analyzer currently analyzes nouns, adjectives and verbs. The information system has a base of affixes. Any word form is fed to the input of the morphological analyzer. The work of the morphological analyzer is being tested by expert linguists. After checking, the experts pointed out several minor errors that were successfully eliminated. Currently, the following statistics are available for checking the functionality of the crawler, morphological analyzer, and morphological disambiguation tool:

- Source: 140,000 news in Kazakh;
- Lemmas: 28, 983;
- Word forms based on new words: 4, 992, 509;
- From the dictionary of word forms and the lemmas themselves 261, 208 are unique, which are found in the news texts;
- Total unique words in these texts are: 1, 269, 549;
- Of these unique words 261, 208 are in the dictionary;
- Total non-unique words: 9, 654, 405;
- Total non-unique words identified: 5, 137,783.

On the basis of 140, 000 news in the Kazakh language word forms were generated and checked. This statistics shows that with the help of generated word forms with meta-information we can consider whole sentences for further analysis as in Figure 5.

3. Conclusion

When working on the research tasks, the following results were obtained:

1) an effective architecture of the text data storage was developed and applied in practice: the structure, which contains many useful attributes such as lemma, parts of speech, equivalent in Russian, allows you to display an effective semantic load, since the user without context can correctly understand the meaning.

2) three preprocessing tools were developed and implemented: word form generator, morphological analyzer and morphological ambiguity resolution tool.

At this stage, the approach to disambiguation is experimental. Due to the lack of a marked-up corpus of the Kazakh language (so far), we cannot use standard approaches. Therefore, we tried to generalize the constructive structure of the sentence in the Kazakh language. Our experiments have shown that this approach will handle morphological ambiguity quite well, although the computational complexity requires good optimization.

Research in the field of corpus linguistics allowed to create a media-corpus of the Kazakh language. The foundation that can accelerate the deployment of the text data corpus has been laid. The prototype of the Kazakh language corpus is used to solve the problems of computational linguistics and is placed on corpus.kaznu.kz. Operational experience has shown the usability of the prototype, as there is a possibility of modifying the data structure.

Funding

This research was supported by grants of Ministry of Education and Science of the Republic of Kazakhstan "Creation of high-performance intelligent analysis and decision making technologies for the «logistics-agglomeration» system within formation of the Republic of Kazakhstan digital economics" [grant number BR05236340] and "Development of methods for automatic extraction of spatial objects from heterogeneous sources for information support of geographic information systems" [grant number AP09261344].

Author details

Darkhan Akhmed-Zaki[1,2]
Madina Mansurova[3]
E-mail: mansurova.madina@gmail.com
Gulmira Madiyeva[4]
Nurgali Kadyrbek[5]
Marzhan Kyrgyzbayeva[3]
[1] Department of Computer Science, Al-Farabi Kazakh National University, Almaty, Kazakhstan.
[2] Astana IT University, Nur-Sultan, Kazakhstan.
[3] Department of Artificial Intelligence and Big Data, Al-Farabi Kazakh National University, Almaty, Kazakhstan.
[4] Department of General Linguistics and European Languages, Al-Farabi Kazakh National University, Almaty, Kazakhstan.
[5] Department of Artificial Intelligence and Big DataAl-Farabi Kazakh National University, Almaty, Kazakhstan.

Citation information

Cite this article as: Development of the information system for the Kazakh language preprocessing, Darkhan Akhmed-Zaki, Madina Mansurova, Gulmira Madiyeva, Nurgali Kadyrbek & Marzhan Kyrgyzbayeva, *Cogent Engineering* (2021), 8: 1896418.

References

Assylbekov, Z., Washington, J., & Tyers, F. (2016). A free/open-source hybrid morphological disambiguation tool for Kazakh. The First International Conference on Turkic Computational Linguistics, 18–26.

Assylbekov, Z., Washington, J. N., Tyers, F., Nurkas, A., Sundetova, A., Karibayeva, A., Abduali, B., & Amirova, D. (2016). A free/open-source hybrid morphological disambiguation tool for Kazakh. *Proceedings of TurCLing*.

Bashkir poetry corpus. 2019. http://web-corpora.net/bashcorpus/search/?interface_language=ru, Accessed 2019 July 7

Bekmanova, G., & Sharipbay, A. A. (2017). Uniform Morphological Analyzer for the Kazakh and Turkish Languages. Proceedings of the Sixth International Conference on Analysis of Images, Social Networks and Texts, Moscow, Russia, 20–30.

Bhardwaj, N. D. (2016). Comparative Study of CouchDB and MongoDB – NoSQL Document Oriented Databases. *International Journal of Computer Applications*, *136*(3), 975–8887.

Caplan, A. (1955). An experimental study of ambiguity and context. *Mechanical Translation*, *2*(2), 39–46.

Constant, M., Eryiğit, G., Monti, J., Van Der Plas, L., Ramisch, C., & Rosner, M. (2017). Multiword Expression Processing: A Survey. Computational Linguistics, 43(4), 837–892.

Corpus of Written Tatar language. 2019. http://corpus.tatar/, Accessed 2019 July 7

Eryiğit, G., Eryiğit, C., Karabüklü, S., Kelepir, M., Özkul, A., Pamay, T., Torunoğlu-Selamet, D., & Köse, H. (2019). Building the first comprehensive machine-readable Turkish sign language resource: Methods, challenges and solutions. Language Resources and Evaluation, 54, 97–121.

Eryiğit, G., & Torunoğlu-Selamet, D. (2017). Social media text normalization for Turkish. Natural Language Engineering, 1–41. https://doi.org/10.1017/S1351324917000134

Gataullin, R. R., & Gil'mullin, R. A. (2016). Contextual rules for resolving morphological polysemy in the Tatar corpus. OpenSemantic Technologies for Intelligent Systems OSTIS-2016, Minsk, 389–392.

Gataullin, R. R. (2016). Analytical review of methods for resolving morphological ambiguity. Russian Digital Libraries Journal, 19(2), 98–114.

Hakimov, B. J., Gil'mullin, R. A., & Gataullin, R. R. (2014). Resolution of grammatical polysemy in the Tatar corpus. Uchenye zapiski Kazanskogo universiteta [Scientific notes of Kazan University]. *Humanities Series*, *156*(5), 236–244.

Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on NoSQL database, Pervasive computing and applications (ICPCA), 6th international conference, IEEE, 363–366.

Koibagarov, K., Amirgaliyev, Y., & Musabayev, T. (2013). Software implementation of recognition of Kazakh speech commands based on the Markov model. Proceedings of the 9th International Asian School-Seminar "Problems of the optimization of complex systems", Almaty, Kazakhstan, 12–17.

Koibagarov, K., Musabayev, R., & Kalimoldayev, M. (2014). Development of a linguistic processor of texts in the Kazakh language. *Journal of Problems of the Informatics*, *24*(3), 64–72.

Kudo, T., & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Brussels, Belgium, 66–71.

Kuriyozov, E., Doval, Y., & Gómez-Rodríguez, C. (2020). Cross-Lingual Word Embeddings for Turkic Languages. Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 4047–4055.

Makhambetov, O., Makazhanov, A., Yessenbayev, Z., Matkarimov, B., Sabyrgaliyev, I., & Sharafudinov, A. (2013). Assembling the Kazakh Language Corpus. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, 1022–1031.

Mansurova, M., Koybagarov, K., Barakhnin, V., Soltangeldinova, M., & Berdibekov, S. (2016). Application of the morphological analyzer of the Kazakh language for the automated filling of the ontology of the factographic search system. *Bulletin of the Kyrgyz State Technical University*, *38*(2), 61–66.

Mansurova, M., Madiyeva, G., Aubakirov, S., Yermekov, Z., & Alimzhanov, Y. (2017). Design and Development of Media-Corpus of the Kazakh Language. Computational Collective Intelligence Technologies and Applications: ICCCI 2017, Nicosia, Cyprus, 509–518.

Mansurova, M., Madiyeva, G., Kadyrbek, N., & Yermekov, Z. (2019). Design and development of preprocessing tools for media-corpus of the Kazakh language. Proceedings of the 9th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics May 17–19, Poznań, Poland, 25–31.

Myrzakhmetov, B., & Kozhirbayev, Zh. (2018). Extended Language Modeling Experiments for Kazakh. International Workshop on Computational Models in Language and Speech, 2303, 35–43.

National corpus of the Russian language. 2019. http://ruscorpora.ru/corpora-intro.html, Accessed 2019 July 7

Nevzorova, O., Mukhamedshin, D., & Gataullin, R. (2017). Developing Corpus Management System: Architecture of System and Database. Int'l Conf. Information and Knowledge Engineering, Las Vegas, Nevada, United States, 108–112.

Petrovic, D., & Stankovic, M. (2019). The influence of text preprocessing methods and tools on calculating text similarity. *Facta Universitatis*, *34*(5), 973–994. https://doi.org/10.22190/FUMI1905973D

Pokorný, J. (2016). How to Store and Process Big Data: Are Today's Databases Sufficient, 13th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM), Ho Chi Minh City, Vietnam, 5–10.

Said, D. A., Wanas, N. M., Darwish, N. M., & Hegazy, N. H. A. (2009). Study of Text Preprocessing Tools for Arabic Text Categorization. The Second International Conference on Arabic Language, Cairo, Egypt, 230–236.

Sak, H., Gungor, T., & Saraclar, M. (2008). Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus. GoTAL: International Conference on Natural Language Processing, Gothenburg, Sweden, Springer-Verlag Berlin Heidelberg, 417–427.

Sulubacak, U., & Eryiğit, G. (2018). Implementing universal dependency, morphology and multiword expression annotation standards for Turkish language processing. *Turkish Journal of Electrical Engineering & Computer Sciences*, 1–23. https://doi.org/10.3906/elk-1706-81

Tunali, V., & Bilgin, T. T. (2012). PRETO: A high-performance text mining tool for preprocessing Turkish texts. CompSysTech '12: Proceedings of the 13th International Conference on Computer Systems and Technologies, Bulgaria, 134–140.

Turganbayeva, A., & Tukeyev, U. (2020).The Solution of the Problem of Unknown Words Under Neural Machine Translation of the Kazakh Language. Communications in Computer and Information Science book series volume 1178, 319–328.

Turkish National Corpus. 2019. http://www.tnc.org.tr/index.php/en/, Accessed 2019 July 7