UDC 004.912
DOI: 10.15587/1729-4061.2021.238743

GRAMMATICAL CATEGORIES DETERMINATION FOR TURKISH AND KAZAKH LANGUAGES BASED ON MACHINE LEARNING ALGORITHMS AND FULFILLING DICTIONARIES OF LINK GRAMMAR PARSER

Aigerim Yerimbetova
Corresponding author
PhD, Associate Professor, Leading Researcher*
E-mail: aigerian@mail.ru
Madina Tussupova
Master of Science in Applied Mathematics and Informatics,
Data Scientist
ENGIE IT
Bd Simon Bolivar, 34, Bruxelles, Belgium, 1000
Madina Sambetbayeva
PhD, Associate Professor, Senior Researcher*
Mussa Turdalyuly
PhD, Head of Department
Department of Software Engineering
Institute of Automation and Information Technologies
Satbayev University
Satbayev str., 22a, Almaty, Republic of Kazakhstan, 050013
Bakzhan Sakenov
Software-Engineer*
*Institute of Information and Computational Technologies
Shevchenko str., 28, Almaty, Republic of Kazakhstan, 050010

This research is aimed at identifying the parts of speech for the Kazakh and Turkish languages in an information retrieval system. The proposed algorithms are based on machine learning techniques. In this paper, we consider the binary classification of words according to parts of speech. We decided to take the most popular machine learning algorithms. In this paper, the following approaches and well-known machine learning algorithms are studied and considered. We defined 7 dictionaries and tagged 135 million words in Kazakh and 9 dictionaries and 50 million words in the Turkish language.

The main problem considered in the paper is to create algorithms for the execution of dictionaries of the so-called Link Grammar Parser (LGP) system, in particular for the Kazakh and Turkish languages, using machine learning techniques.

The focus of the research is on the review and comparison of machine learning algorithms and methods that have accomplished results on various natural language processing tasks such as grammatical categories determination.

For the operation of the LGP system, a dictionary is created in which a connector for each word is indicated – the type of connection that can be created using this word. The authors considered methods of filling in LGP dictionaries using machine learning.

The complexities of natural language processing, however, do not exclude the possibility of identifying narrower tasks that can already be solved algorithmically: for example, determining parts of speech or splitting texts into logical groups. However, some features of natural languages significantly reduce the effectiveness of these solutions. Thus, taking into account all word forms for each word in the Kazakh and Turkish languages increases the complexity of text processing by an order of magnitude

Keywords: natural language processing, part-of-speech, machine learning algorithms, agglutinative language, Word2vec

Received date 10.08.2021
Accepted date 07.10.2021
Published date 29.10.2021

***How to Cite**: Yerimbetova, A., Tussupova, M., Sambetbayeva, M., Turdalyuly, M., Sakenov, B. (2021). Grammatical categories determination for Turkish and Kazakh languages based on machine learning algorithms and fulfilling dictionaries of link Grammar Parser. Eastern-European Journal of Enterprise Technologies, 5 (2 (113)), 55–65. doi: https://doi.org/10.15587/1729-4061.2021.238743*

1. Introduction

Currently, natural language processing (NLP) is considered a major problem in many areas [1]. Fundamental techniques in NLP include sequence labeling, n-gram patterns, rollback, and scoring. These techniques are useful in many subject areas such as machine translation, named entity recognition, etc., and tagging in turn gives us a simple context to represent them.

The research is aimed at creating a scientific and technical groundwork in the field of information and communication technologies and obtaining new knowledge that allows for semantic analysis of texts in natural languages [2].

Research on the development of a system for determining grammatical categories for the system of relations of the Turkic languages, and the implementation of tools for constructing connection diagrams on the platform of the LGP software system for the Kazakh and Turkish languages are currently insufficient for automatic text processing.

Many researchers are inclined to the need for a deep semantic analysis of texts to create their semantic images, on the basis of which it would be possible to conduct a fine ranking of documents. This approach is undoubtedly the most reasonable, but it requires careful and long work on creating suitable tools for automatic text processing. In particular, a detailed description of various areas of knowledge may be required. Therefore, it also makes sense to search for partial solutions, one of which is presented in this paper.

The disadvantages of known procedures (methods):

– the need to develop new and improve existing algorithms for searching and ranking documents that can take into account the semantics of incoming requests;

– the presence of scientific problems related to the search and analysis of textual information; the variability of vocabulary, homonymy and syntactic synonymy (paraphrasing);

– the need to develop fast search and analysis algorithms used for large text collections, as well as the fact that information search and analysis algorithms are often hidden by developers.

Therefore, studies that are devoted to the problems of information search have always been relevant for the Internet and have scientific relevance [3]. First of all, this is due to the enormous amount of information resources.

Nowadays, the tasks of data mining, searching and extracting information, determining the topics of texts are complex, but relevant. The semantic analysis of textual information plays a particularly important role.

2. Literature review and problem statement

The research paper uses a software tool known as Link Grammar Parser [4]. Although it is called a parser, in fact, it considers a lot of relationships between words that can be called semantic. The number of main links for the English language is more than 100. Taking into account the fact that there are various "derivative" connections, about 200 connections are obtained. Dictionaries are easy to add new links. In fact, the whole theory of [5], the research of [6] on the semantics of verbs, etc. can be put there. The method of matching sentences used in the so-called basic algorithm is quite simple. An attempt was made to find out which set of links is sufficient to get good results when determining relevance. The paper presents this set – 35 links. It also says the following. The minimal variant that gave fairly good results when only 8 connections were taken into account: C, CC, S, SI, SF, SFI, SX, SXI. Connections were found that significantly spoiled the situation.

Further, an attempt was made to take into account the paraphrasing. Actually, about 35 paraphrases were formulated for the English language. Many types of paraphrases are considered for the Russian and Kazakh languages [7]. Experiments have shown that taking into account paraphrases almost does not give any effect. The fact is that the snippets that were analyzed are usually very simply arranged, have a direct word order, etc., and queries are also usually simply arranged. Any artificial variations of requests were not considered.

In the tasks of classifying texts by topic, the topic in the work is a set of small reference texts. In most works, the topic is associated with a set of keywords that are either set initially or formed during the operation of algorithms. Thus, in the work, fragments of the text under study (for example, paragraphs) are compared with standards. In principle, different fragments can be attributed to different topics.

This is done as follows: some graphs are compared to a text fragment and a standard, then they are compared. [9] whose works are used in the work compares the text with a graph such that statistics and word order are "sewn" in it. The paper additionally involves the relations (i.e. connections) generated by the Link Grammar Parser system. It is important that the graph is compared to the text as a whole, and not to a separate sentence.

Thus, two graphs corresponding to the text fragment and the standard are obtained. Next, the vertices with large weights are allocated in a certain way in them. When determining the weights of words, they usually refer to the concept of PageRank, but, in fact, there is another analogy. In mathematical logic, there is a concept of the rank of formulas. If we look from the complexity point of view, then vertices with low weights are discarded, while the complexity of the algorithm decreases.

Summarizing, we can say that in general, the algorithms for determining topics, classification by topic, etc. have some instability (including the algorithm proposed in the paper) in relation to subject areas. If there is a well-established terminology in these subject areas, then the algorithms work quite correctly. Moreover, "playing on the choice of thresholds", you can see how the document classes are divided into subclasses, and the tree-shaped structure is clearly visible. If the terminology is not well-established, then we get incorrect results.

The question arises, why is this happening? Two hypotheses can be made. Hypothesis 1. The set of syntactic relations, if we use the classical theory of syntax, is too simple. The choice of the set of semantic relations used in the Exactus system [9] is not sufficiently investigated. The theory of [10] or the communicative grammar and their own programs are used, all the same, the question remains unexplored, which connections are "useful", which are "harmful", and maybe some connections are missing.

While working on the study, efforts were made to clarify this issue. The main focus of the study is on reviewing and comparing machine learning algorithms and methods that allowed us to obtain results when solving various natural language processing tasks, such as determining grammatical categories.

For the LGP system to work, a dictionary is created in which a connector is specified for each word – the type of connection that can be created using this word. The authors consider methods of filling LGP dictionaries using machine learning.

However, the complexity of natural language processing does not exclude the possibility of identifying narrower tasks that can already be solved algorithmically: for example, determining parts of speech or dividing texts into logical groups. However, some features of natural languages significantly reduce the effectiveness of these solutions. Thus, taking into account all word forms for each word in the Kazakh and Turkish languages increases the complexity of text processing by an order of magnitude.

The second hypothesis is that the account of syntax and morphological features in Exactus is too strict [11]. Therefore, the system rejects almost everything, does not find matches between the matched sentences.

As a result, a comparison with a close approach, i.e. with the Exactus system, is omitted in the work and the abstract, because the comparison is not in favor of Exactus. However, it is known from publications that there are subject areas in which Exactus works correctly, and is useful in practice.

In particular, [12] describes the approaches focused on agglutinative languages. In most works, the authors limit themselves to considering the morphological structure of the Kazakh or Turkish languages, carry out their comparative analysis.

The Slavic and East European Language Resource Center of Duke University presented the Kazakh grammar with suffixes in a manner that seeks dividing the suffix from its various surface forms [13].

Like Kazakh, the Turkish language is also highly agglutinative. [14] provided a full and accessible description of the language, concentrating on the real patterns of use in a Comprehensive Grammar for Turkish.

In 2019, a stemming algorithm for the Kazakh language using the rule-based approach was provided by [15]. The stemming algorithm will be useful in the development of sentiment analysis for the Kazakh language, because of its rareness and competency. The stemming algorithm works with the aim to cut ending combinations. The results were checked by Machine Learning algorithms using the annotated dataset, which were retrieved from the Kazcorpus dataset.

Our review has revealed the following disadvantages of known procedures (methods):

– the need to develop new and improve existing algorithms for searching and ranking documents that can take into account the semantics of incoming requests;

– the presence of scientific problems related to the search and analysis of textual information.

Vocabulary variability, homonymy and syntactic synonymy (paraphrasing).

The need to develop fast search and analysis algorithms that can be used for large text collections.

Algorithms of information search and analysis are often hidden by developers.

Therefore, it is necessary to develop an improved method for the grammatical categories determination for the Turkish and Kazakh languages based on machine learning algorithms and fulfilling dictionaries of the link grammar parser.

For this research, we selected the rule-based part of speech tagging. As a rule, 8–9 classes are considered, they are Noun, Verb, Participle, Article, Pronoun, Preposition, Conjunction, Adverb, and Adjective. However, Articles in some cases were not considered because of the Turkic language peculiarities.

3. The aim and objectives of the study

The aim of this study is to develop a machine learning algorithm for determining grammatical categories for the system of relations of the Turkic languages, and the implementation of tools for constructing connection diagrams on the platform of the LGP software system for the Kazakh and Turkish languages.

To achieve this aim, the following objectives were set:

– to create a rule-based algorithm for dataset tagging;

– to test machine learning algorithms for solving the problem of part of speech, case of noun and determination for Kazakh;

– to test machine learning algorithms for solving the problem of part of speech, case of noun and determination for Turkish;

– to create special dictionaries for LGP.

4. Materials and research methods

The studies are partly based on the "Set-theoretic models of languages" theory of S. Marcus. The features and connections of the LGP system are specially encoded by some number vectors. The system for fulfilling dictionaries of LGP essentially uses the developed identifiers of parts of speech of the Kazakh and Turkish languages. The research basically focuses on Kazcorpus, i.e. the Kazakh language corpus and Turkish National Corpus (TNC) of the Turkish language. Kazcorpus contains more than 400,000 documents [16] and TNC contains about 50 million words [17].

When implementing the prototype of the LGP system for the Kazakh and Turkish languages, a communication system for the Turkic languages was proposed. We have identified 7 dictionaries and marked 135 million words in the Kazakh language and 9 dictionaries and 50 million words in the Turkish language.

The main problem considered in the paper is the creation of algorithms for the execution of dictionaries of the so-called link grammar parsing system (LGP), in particular for the Kazakh and Turkish languages, using machine learning methods.

The study used mainly methods related to information technology and used in the processing of texts in natural language, as well as methods from graph theory and mathematical logic. The work involved quite extensive material from classical and mathematical linguistics.

In this paper, two major problems are considered.

The first problem is to develop an algorithm for determining the parts of speech in Kazakh and Turkish. The algorithm is based on machine learning [18]. A deeper consideration of this topic leads us to the theory of [19] that is well-known as "Set-theoretic models of languages", and Word2vec technology [20].

Markus showed that having a collection of texts, it is possible to formally determine grammatical categories of the language using computational procedures: parts of speech, gender, and cases. At the same time, well described in the literature generative grammars of languages do not allow us to do this correctly.

Word2vec is a set of models for analyzing the semantics of natural languages, where the technology is based on distributive semantics and vector representing of words. This tool was developed by a group of Google researchers in 2013 and described in their paper "Distributed Representations of Words and Phrases and their Compositionality" [21]. In order to create a repository of vector representations of the words of the Kazakh language, we had to perform a huge work on analyzing Kazcorpus [16], the Kazakh corpus of texts and Turkish National Corpus (TNC) for the Turkish language [17]. First, we tagged the corpus with the rule-based model and then on this basis, we have tested various machine learning algorithms such as Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, etc.

The second problem is to develop and implement methods based on machine learning for fulfilling the dictionaries of the LGP software system. This parser is based on an original Link Grammar theory of syntax and morphology. It is noteworthy that this theory is totally different from the classical theory of syntax. After receiving a sentence, the system adds a syntactic structure that consists of a set of labeled links that connect pairs of words.

Currently, there are some variants of this system for many other languages. The Institute of Informatics Systems of the Siberian Branch of the Russian Academy of Sciences (IIS SB RAS) carried out advanced studies for the Kazakh and Turkish languages.

The agglutinative languages are characterized by a sufficiently developed system of word-formation and inflectional affixation, grammatical uniqueness of affixes and the absence of alterations. Agglutination is the essence of forming new grammatical forms and words by attaching to the stem the affixes with distinct properties in a way that the boundaries of morphs do not change. Each affix has a single meaning, and each function is expressed by one particular affix. The phenomenon of agglutination is reduced to the assignment without changing the word-forming suffix to the morph of the stem. In this case, each specific suffix or affix has specific semantic, psycholinguistic meanings and functions [22].

Rule-based algorithm for the Kazakh and Turkish languages.

Morphologically and syntactically, the Kazakh language is a nominative and agglutinative language with the presence of polysynthetic language. There are 8 parts of speech in Kazakh: Adverb, Adjective, Verb, Noun, Numeral, Conjunction, Pronoun and Postposition.

Let's name a finite set of words a dictionary as and consider a free semigroup as on, namely, the set of all finite sequences of words are defined with an associative and non-commutative binary operation of concatenation. The sequence of words are also called as the sequence (chain) over. The zero sequence, we denote θ as a sequence of $\theta x = x\theta = x$, where x is for each sequence. In the above example, the specifically stated words are not considered.

We suppose that a language L contains a word w and a dictionary D. Each word has a stem s and an affix a. We define part-of-speech tags as categories C_i. We define predicates $C_i(w)$, $A_i(w)$, $D_i(w)$, where w belongs to category $C_i(w)$, a belongs to category $A_i(w)$, and w belongs to category $D_i(w)$.

We define model m, where $m|=C_i(x)$ means that property $C_i(x)$ is true in the model m, as follows:

$$1.\ m\big|=C_i(w)\leftrightarrow\big(w=s+a\wedge A_i(a)\big)\vee D_i(w),\quad (1\le i\le 5);$$

$$2.\ m\big|=C_i(w)\leftrightarrow D_i(w),\quad (6\le i\le 8).$$

Here are affixes and selected set of tagged words from Kazcorpus and TNC accordingly.

Thus, using the general rule-based algorithm, it is possible to describe separate causes.

```
Input: Word
Output: Part of Speech
While true:
Take word from the text;
If word has Noun-affix or word in Noun-dictionary:
        Function returns "Noun"
Elif word has Verb-affix or word in Verb-dictionary:
        Function returns "Verb"
Elif word has Adjective-affix or word in Adjective-dictionary:
        Function returns "Adjective"
Elif word has Adverb-affix or word in Adverb-dictionary:
        Function returns "Adverb"
Elif word has Pronoun-affix or word in Pronoun-dictionary:
        Function returns "Pronoun"
Elif word has Preposition-affix or word in Preposition-dictionary:
        Function returns "Preposition"
Elif word has Determiner-affix or word in Determiner-dictionary:
        Function returns "Determiner"
Else:
        None
If press ctrl+C:
Break(close)
```

Description of the rule-based algorithm for the Kazakh language is based on Kazakh Grammar with Affix List proposed by [20]:

*A*1: (. *ша|. *ше|. *дай|. *дей|. *тай|. *тей|. *лай|. *лей|. *дайын|. *дейін|. *тайын|. *тейін|. *шама|. *шеме|. *шалық|. *шелік|. *сын|. *сін|. *дан|. *ден|. *тан|. *тен|. *нан|. *нен|. *де|. *та|. *те|. *нда|. *нде|. *ға|. *ге|. *қа|. *ке|. *на|. *не).

If word $w\subseteq D1$={ертең|бүгін|биыл|таңертең|ерте|қыстай|осыншама| жемейінше|жылдам|жақсы|қиын|оңай|алға|төмен|сонда|ілгері|артқа|сыртта|мұнда|артта|іште|etc.} or

$$w=s+A1\rightarrow w\in C1\ \text{(Adverb)}.$$

*A*2: (. *дық|. *дік|. *тық|. *тік|. *лық|. *лік|. *паз|. *ғыш|. *гіш|. *қыш|. *кіш|. *шек|. *шақ|. *шыл|. *шіл|. *ды|. *ді|. *ты|. *ті|. *лы|. *лі|. *ғы|. *гі|. *қы|. *кі|. *дай|. *дей|. *тай|. *тей|. *и|. *ы).

If word $w\subseteq D2$={абай|ағылшын|адал|айнадай|ақ|көңіл|аққұба|ақсақ| ақшыл|көк|ақылды|ақылы|ақымақ|ала|аласа|алғышқы|бай|байғұс|байсалды|байтақ|бақытсыз|бақытты|басқа|бастауыш|басты|түсті|газдалған|ғажайып|дайын|дана|дарынды|дәмді|дәмсізегде|ежелгі|емдік|ең|еңбекқор|еңкіш|ерекше|ересек|еркін|ерте|ертеңгі|еріншек|есепшіл|ескі|жабайы|жабық|жағымды|жағымсыз|жазба|жазық|жай|жайлы|жақсы|зиялы|зиянды|кәрі|кедей|кез|келген|кекшіл|көне|көнерген|көңілді|көңілсіз|көпшіл|көтеріңкі|күйеуге|шыққан|күлгін|күлдіргі|күлкілі|күміс|күнделікті|күндізгі|күңгірт|күрделі|күткен|күтпегенқатты|қауіпсіз|қауіпті|қаһарман|қисық|қиын|қола|қолайлы|қонақжай|қоңыр|қос|қосымша|қою|қу|қуанышты|қуырылған|құрама|құрғақ|құрмалас|құрметті|құтты|қызба|қызғаншақ|сары|қызық|қызықсыз|қызықты|қызыл|қымбат|қымбатты|қыңыр|қырғыз|қырсық|қысқа|қысқы|қышқыл|лас|майда|мақсатты|мақтаншақ|мемлекеттік|момын|ресми|реттік|риза|рухани|сабырлы|сабырсыз|сақ|сақалды|салқын|салмақты|саңырау|сараң|сары|сау|сауатты|сәлемет|сәнді|сезімтал|семіз|сенгіш|сенімді|сирек|сол|соңғы|сопақ|сөзуар|сөзшең|сулы|суық|сұйық|сұлу|сұр|тегіс|темір|тентек|тең|терең|тиімді|тоқ|тоқыма|толқынды|толы|толық|томпақ|төменгі|туған|турашыл|тұздалған|тұзды|etc.} or

$$w=s+A2\rightarrow w\in C2\ \text{(Adjective)}.$$

*A*3: (. *сыңдар|. *сіңдер|. *сыздар|. *сіздер|. *ңыздар|. *ңіздер|. *тау|. *мын|. *мін|. *бын|. *бін|. *пын|. *пін|. *мыз|. *міз|. *быз|. *біз|. *пыз|. *піз|. *сың|. *сің|. *сыз|. *сіз|. *м|. *к|. *ң|. *ңдар|. *ңдер|. *ңыз|. *ңіз|. *ап|. *ып|. *іп|. *п|. *ды|. *ді|. *ты|. *ті|. *е|. *й|. *са|. *се|. *у|.*р).

If word $w\subseteq D3$={тұр|жүр|отыр|жатыр|көр|аш|же|бол|қу|кел|тұр|жүр|жат| жүгір|көтер|ал|қаш|орал|жыла|күл|жыми|секір|қуан|өт|бер|әкел|сипа|сына|қина|гүлде|ойна|киін|қара|көн|кінәлә|жеркен|той|кеп|сула|etc.} or

$$w=s+A3\rightarrow w\in C3\ \text{(Verb)}.$$

*A*4:(|. *дікі|. *тікі|. *нікі|. *дан|. *ден|. *тан|. *тен|. *нан|. *нен|. *дың|. *дің|. *тың|. *тің|. *нің|. *ның|. *дар|. *дер|. *тар|. *тер|. *лар|. *лер|. *і|. *ы|. *шы|. *ші|. *ба|. *бе|. *па|. *пе|. *ма|. *ме|. *да|. *де|. *та|. *те|. *нда|. *нде|. *ды|. *ді|. *ты|. *ті|. *ны|. *ні|. *н|. *ба|. *бе|. *па|. *пе|. *ма|. *ме|. *ға|. *ге|. *қа|. *ке|. *на|. *не|. *а|. *е).

If word $w \subseteq D4$={қарн|адам|қыз|бала|ит|мысық|алма|гүл|ұл|ата|ана|әке|әже| құмырсқа|жұмыртқа|үстел|орындық|тышқан|қалам|қалампыр|ноутбук|ғаламтор|кітап|пакет|дорба|сызба|қағаз|дәптер|сызғыш|төбе|тақта|тырнақ|шаш|бас|өшіргіш|кетіргіш|картон|қасық|қайшы|шай|тоңазытқыш|шанышқы|кесе|бала|қыз|ұл|etc.} or

$w = s + A4 \rightarrow w \in C4$ (Noun).

*A*5: (. *інші|. *ыншы).

If word $w \subseteq D5$={бір|екі|үш|төрт|бес|алты|жеті|сегіз|тоғыз|он|жиырма|отыз| қырық|елу|алпыс|жетпіс|сексен|тоқсан|жүз|мың} or

$w = s + A5 \rightarrow w$, then $w \in C5$ (Numeral).

*A*6: If word $w \subseteq D6$={және|әрі|да|де|та|те|бен|пен|ал|бірақ|алайда|дегенмен|әйткенмен|әйтпегенде|әйтпесе|болмаса|я|яки|не|немесе|болмаса|әлде|біресе|бірде|яғни|өйткені|себебі|сондықтан|егер|онда|etc.}, then

$w \in C6$ (Conjunction).

*A*7: If word $w \subseteq D7$={мен|сен|сіз|ол|біз|сендер|сіздер|олар|менің|сенің|сіздің| оның|біздің|сендердің|сіздердің|олардың}, then

$w \in C7$ (Pronoun).

*A*8: If word $w \subseteq D8$={үшін|туралы|жайлы|жайында|жөнінде|бойынша| бойында|бойы|сайын|арқылы|сияқты|тәрізді|сықылды|секілді|ғұрлы|құрлы|ғұрлым|шамалы|шақты|қаралы|түгіл|кейін|соң|бері|бастап|басқа|бұрын|әрі|артық|бетер|астам|аса|тыс|гөрі|көрі|дейін|шейін|қарай|салым|жуық|таяу|таман|тарта|сәйкес|орай|қарсы|бола|бірге|қоса|қатар|қабат|ма|ме|ба|бе|па|пе|мы|мі|бы|бі|пы|пі|etc.}, then

$w \in C8$ (Postposition).

For example, "Адам аулап, сыпыра саулап, байды жаулап жетісер". (Kunanbayev, 1893). First, we divide the sentence into words, so we have:

1) "адам" – there are no affixes so the algorithm checked the dictionary and found it in *D*4;

2) "аулап" – there is an affix *ап|. and it could be found in *A*3;

3) "сыпыра" – there is an affix*а|. and it could be found in *A*3;

4) "саулап" – there is an affix *ап|. and it could be found in *A*3;

5) "байды" – there is an affix *ды|. and it could be found in *A*4 and *A*2, the stem "бай" as well could be found in *D4* and *D2*;

6) "жетісер" – there are two affixes *р|. and *се|. and they could be found in *A*3.

Description of the rule-based algorithm for the Turkish language is based on a Comprehensive Grammar proposed by [14]:

*A*1: (. *sal|. *ıt|. *cağız|. *cık|. *cik|. *cuk|. *cük).

If word $w \subseteq D1$={hakkında|yukarısında|karşısında|sonra|karşısında|arasında| etrafında|gibi|önce|önünde|arkasında|aşağısında|altında|yanında|arasında|ötesinde|ama|tarafından|rağmen|aşağı|sırasında|dışında|için|itibaren|içinde|içinde|içine|yakın|yakınında|sonra|sonrasında|üzerinde|karşısında|dışarı|dışında|dışarısında|üzerinde|başına|artı|ek|olarak|etrafında|beri|dolayı|göre|sayesinde|kadar|doğru|altında|aksine|kadar|yukarı|üzerinden|aracılığıyla|ile|içinden|olmadan|iki|kelime|göre|nedeniyle|yakın|dolayı|hariç|uzak|içinde|yerine|yakın|yanındaki|dışında|önce|öncesinde|üç|kelime|kadarıyla|olabildiğince|uzak|hem|de|ek|olarak|önünde|rağmen|adına|üzerinde|belirteç|edatları|bu|o|bunlar|Bu|nın|etc.} or

$w = s + A1 \rightarrow w \in C1$ (Adverb).

*A*2: (. *sız|. *siz|. *suz|. *süz|. *lı|. * li|. *lu |. *lü|. *ca |. *ce |. *ça |. *ç|. *cil|. *cıl|. *şın).

If word $w \subseteq D2$={orgeneral|edildiği|siyah|mavi|kahverengi|gri|yeşil|portakal| rengi|mor|kırmızı|beyaz|sarı|büyük|derin|uzun|dar|kısa|küçük|uzun|yüksek|kalın|ince|geniş|dairesel|yuvarlak|düz|kare|üçgen|şeklinde|acı|taze|tuzlu|ekşi|baharatlı|tatlı|kötü|temiz|karanlık|zor|kirli|kuru|kolay|boş|pahalı|hızlı|yabancı|tam|noksansız|dolu|iyi|sert|ağır|ucuz|hafif|yerel|yeni|gürültülü|eski|güçlü|sessiz|doğru|yavaş|yumuşak|çok|zayıf|ıslak|yanlış|genç|az|küçük|çok|çok|bölüm|bazı|birkaç|bütün|etc.} or

$w = s + A2 \rightarrow w \in C2$ (Adjective).

*A*3: (. *lamak|. *lemek|. *almak|. *l|. *e|. *damak|. *demek|. *atmak|. *etmek|. *ıkmak|. *ikmek|. *ımsamak|. *imsemek|. *kırmak|. *lanmak|. *lenmak|. *laşmak|. *leşmak|. *samak|. *semek|. *ala|. *ele|. *ımsa|. *imse|. *in|. *un|. *ün|. *ş|. *t|. *ıl|. *il).

If word $w \subseteq D3$={açmak|açar|açtırmak|akmak|akar|akıtmak|almak|alıştım|alır| aldırmak|anmak|anar|andırmak|artmak|artar|artırmak|asmak|asar|astırmak|aşmak|aşar|aşırmak|atmak|atar|attırmak|banmak|banar|bandırmak|basmak|basar|bastırmak|bıkmak|bikar|bıktırmak|bozmak|bozar|bozdurmak|bulmak|bulur|buldurmak|caymak|cayar|caydırmak|coşmak|coşar|coşturmak|çakmak|çakar|çaktırmak|çalmak|çalar|çaldırmak|çalmak|etc.} or

$w = s + A3 \rightarrow w \in C3$ (Verb).

*A*4: (. *|. *le|. *ci|. *cı|. *cu|. *cü|. *çi|. *çı|. *çu|. *çü. *tı|. *ti|. *tu|. *tü|. *ca|. *ça|. *ce|. *çe|. *acak|. *ecek|. *ak|. *ek|. *ga|. *ge|. *da|. *de|. *gan|. *kan|. *gen|. *ken|. *gı|. *gi|. *gıç|. *giç|. *gın|. *kın|. *gin|. *kin|. *gün|. *kün|. *gun|. *kun|. *ı|. *i|. *e|. *u|. *ü|. *ıcı|. *ici|. *ucu|. *ücü|. *ık|. *ik|. *uk|. *ük|. *ım|. *im|. *um|. *üm|. *in|. *un|. *ün|. *inç|. *inç|. *unç|. *ünç|. *ıntı|. *inti|. *untu|. *üntü|. *ar|. *er|. *ır|. *ir|. *ur|. *ür|. *r|. *ış|. *iş|. *uş|. *üş|. *ıt|. *it|. *ut|. *üt|. *tı|. *ti|. *tu|. *tü |.*dum|. *dun|. *dunuz).

If word $w \subseteq D4$={kol|geri|yanaklar|göğüs|çene|kulak|dirsek|göz|yüz|parmak| parmaklar|ayak|saç|el|kafa|kalp|diz|bacak|dudak|ağız|boyun|burun|omuz|mide|dişler|uyluk|boğaz|başparmak|ayak|parmağı|dil|diş|kuvvet|idare|denetleme|incelemelerde|tatbikat|incelemelerde|denetleme|idare|savunma|etc.} or

$w = s + A4 \rightarrow w \in C4$ (Noun).

*A*5: (. *uncu|. *inci).

If word $w \subseteq D5$={bir|iki|üç|dört|beş|altı|yedi|sekiz|on|dokuz|yirmi|yüz|bin| milyon|birinci|ilk|ikinci|üçüncü|dördüncü|beşinci|altıncı|yedinci|sekizinci|dokuzuncu|onuncu|yirminci|etc.} or

$$w = s + A5 \rightarrow w, \text{ then } w \in C5 \text{ (Numeral).}$$

$A6$: If word $w \subseteq D6$={ama|de|da|ise|ile|ki|madem|fakat|hatta|ya da|yahut|etc.}, then $w \in C6$ (Conjunction).

$A7$: If word $w \subseteq D7$={ben|sen|o|biz|siz|onlar|bana|sana|ona|bize|size|onlara|bende|sende|onda|bizde|sizde|onlarda|bu|şu|o|bunlar|şunlar|onlar}, then $w \in C7$ (Pronoun).

$A8$: If word $w \subseteq D8$={gibi|için|ile|kadar|doğru|göre|kadar|karşı|önce|sonra| beri|itibaren|dolayı|bakımdan|hakkında|tarafından|yüzünden|etc.}, then $w \in C8$ (Postposition).

$A9$: If word $w \subseteq D9$={şarj|saniyede|kontrol|konferans}, then $w \in C9$ (Foreign Words).

As an example, we took the sentence "Tütüne böyle havada alıştım Böyle havada aşık oldum;" (Orhan, 1951). As in the previous example, we divide the sentence into words and apply the algorithm on them.

1) "tütüne" – there is an affix *e|. and it could be found in $A4$;

2) "böyle" – there are no affixes, it could be found in extended dictionaries $D1$ and $D2$, it could be both Adjective and Adverb;

3) "havada" – there is an affix *da|. and it could be found in $A4$;

4) "alıştım" – there are two affixes *im|. and *t| and they could be found in $A3$ and $A4$ accordingly, so we check extended dictionaries and can find it in $D3$;

5) "aşık" – there is no affix, however, it could be found in extended dictionary $D2$;

6) "oldum" – there is an affix *dum| and it could be found in $A3$.

Subsequently, the following options of well-known machine learning algorithms were applied: Logistic Regression, K-nearest neighborhood, Decision Tree Classifier, Random Forest, Support Vector Machine. Computer Area Under the Receiver Operating Characteristic Curve (ROC AUC) was taken for prediction score. The dataset was divided into two parts: train (67 %) and test (33 %) with random state 42. Parameters for Logistic regression were stopping criteria 1e-4, with maximum iterations of 100. For the K-nearest neighborhood, we determined parameters as 2 neighbors and 'ball_tree' algorithm. The Decision Tree and Random Forest algorithms had quite similar parameters, the only difference was in the numbers of estimates in the Decision Tree equal to 100 and for SVM we took C-Support Vector Classification with default parameters.

5. Results of the grammatical categories determination for the Turkish and Kazakh languages

5. 1. Word2Vec Algorithm for Parts of Speech Determination

Due to the fact that when processing text, words of texts are discrete and categorical features, because most algorithms process only numerical values that are necessary for using some vectorizers. After some experiments with TF-IDF, Word2Vec and one hot encoder algorithms, we decided to use Word2Vec because of its higher accuracy in results.

Word2Vec is a group of related models for the word's occurrences and connections analysis, created by Google. There are two general algorithms in Word2Vec such as Continuous Bag of Words (CBOW) and Skip-gram. Vector model is the algebraic model for representing text documents in the form of vectors. In the vector model, the document is considered as a set of terms, i.e. selected words or word combinations. Every component of a vector of features corresponds to a separate term. The numerical value of a component is named as a weight of the term that characterizes the importance of the term for representing the given document. If the term is not met in the document, then its weight in this document is equal to zero.

Furthermore, all terms that are met in documents of a processed collection may be ordered. If we want to write out the weight of all terms for some documents, even the terms that are not entered in the document, the vector will be obtained in any event. In any case, the vector of the given document will be represented in the vector space. The dimension of this vector, as well as the dimension of the whole space, is equal to the number of various terms in all collections, and it is identical to all documents.

Word2Vec takes a huge corpus of content as its input and usually produces a vector space having a dimension of several hundred. Given a text corpus, the Word2Vec tool learns a vector for every word in the vocabulary using the Continuous Bag-of-Words or the Skip-Gram neural network architectures.

Kazcorpus Kazakh language corpus exceeds 135 million words [23, 24] and it contains more than 400.000 documents classified into five major genres:

1) literary genre comprises Kazakh literary works of art, including novels, stories, poems, and others, published in the range from the beginning of the XX century to the present;

2) official genre includes mainly official statutes, orders, acts and other legal documents produced by the governmental organizations within 2009 and 2012;

3) scientific genre includes academic books, monographs, theses, research papers and essays from various subject areas, such as computer science, biology, chemistry, and others;

4) publicistic genre (mass media) comprises periodicals and articles from online sources, i.e. newspapers and magazines published over the last ten years;

5) informal genre includes documents with colloquial Kazakh texts extracted from the popular blog platforms starting from 2009. The vector model for the Kazakh language was built after experiences in using Kazcorpus. Afterwards this model was used for parts of speech determination. For each tag, 1,000 words were taken. The results are represented below (Table 1).

Table 1

Machine Learning algorithms for the Kazakh language

Algorithms Part of speech	LogReg	KNN	Des. Tree	Rand. For	SVM
Noun	0.6777	0.6317	0.5910	0.6450	0.7098
Adjective	0.6542	0.6206	0.5594	0.6231	0.6962
Verb	0.8559	0.7046	0.6548	0.8315	0.8601
Adverb	0.8388	0.6857	0.6478	0.7879	0.8031
Pronoun	0.9072	0.5295	0.5318	0.8651	0.8221
Preposition	0.7729	0.5241	0.5126	0.6414	0.5748
Determiner	0.7641	0.5589	0.4964	0.6832	0.6475

We can see that the best predictor for nouns, adjectives and verbs is SVM, for adverbs, pronouns, prepositions and determiners is Logistic Regression.

Turkish National Corpus (TNC) for the Turkish language is a corpus of contemporary Turkish language with

50 million words in it. TNC comprises texts of a wide range covering 24 years (1990–2013).

Let's define model *m*, where (*m*)|=*C*(*x*) means that property *C*(*x*) is true in model *m*.

1. $m \vDash C_i(w) \Leftrightarrow w = s + a \,\&\, A_i(a)$, or $D_i(w)\forall i,\ i \in \{1,5\}$;

2. $m \vDash C_i(w) \Leftrightarrow D_i(w)\forall i,\ i \in \{6,8\}$.

Using this model system, we build the rule-based part of speech algorithm. Based on rule-based and machine learning algorithms, it can be concluded that algorithms are working successfully. Algorithms of Machine Learning and Vectorization are working successfully. They may be used to fill up various dictionaries such as dictionaries of LGP. The main research question has been concisely answered. Fig. 1 shows the best predictors for each part of speech.

In Fig. 1, for nouns, adjectives, verbs, conjunctions and postpositions, the best algorithm is Logistic regression with results of 0.7004, 0.7167, 0.6373, 0.781 and 0.6585, respectively. For numerals, the best algorithm is Random Forest Classifier. For pronouns and adverbs, the best algorithm is Support Vector Machines Classifier. Using this information, we could predict everything faster. Overall time spent on calculations is 3.3033 seconds.

In Fig. 2, the results of machine learning algorithms work for the Turkish language are illustrated. Overall time spent is 4.7138 seconds.

The best predictable algorithm for Turkish Nouns, Verbs, Postposition and Conjunctions is K-nearest neighborhoods, SVM is best for Pronouns, Numerals and Adverbs. For Adjectives with a cross validation score equal to 0.9086, the best is Logistic regression and for Foreign Words, the best machine learning algorithm is Decision Tree (0.8973). The meanest results are shown by random forest algorithm.

Dictionaries for both languages were created. To better understand the implications of these results, future studies could address the determination of relations between words.

5. 2. Results of machine learning algorithms for the Kazakh language

Part-of-speech determination for the Kazakh language using machine learning algorithms. The first step in any machine learning task is the determination of the baseline solution. Fig. 3 shows the results of Zero-rule algorithm.

The results, i.e. the quality of the parts of speech determination, are represented below (Table 2).

Thus, the best predictors are determined for each part of speech in the Kazakh language. For nouns, adjectives, conjunctions and postpositions, the best algorithm is Logistic regression. Random Forest Classifier is the best algorithm for numerals. For verbs, pronouns and adverbs, the best algorithm is Support Vector Machines Classifier.

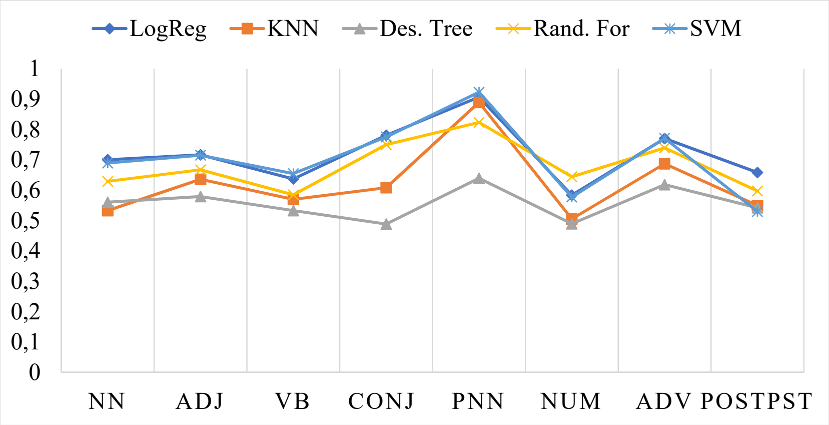


Fig. 1. Machine Learning algorithms for prediction of parts-of-speech of the Kazakh language

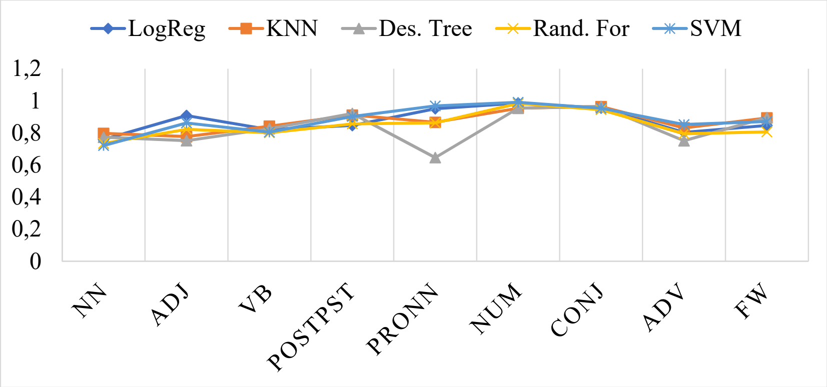
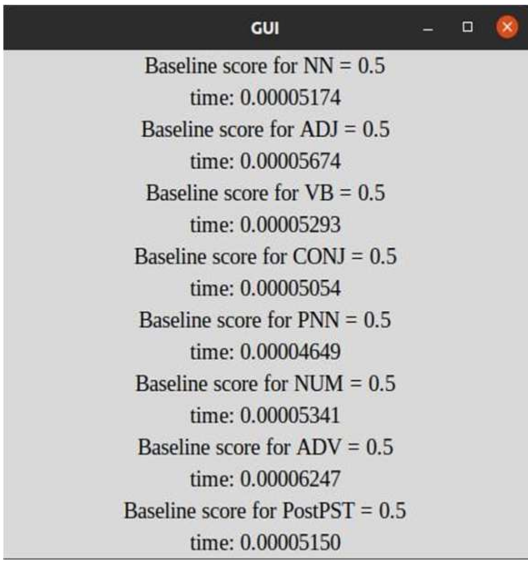
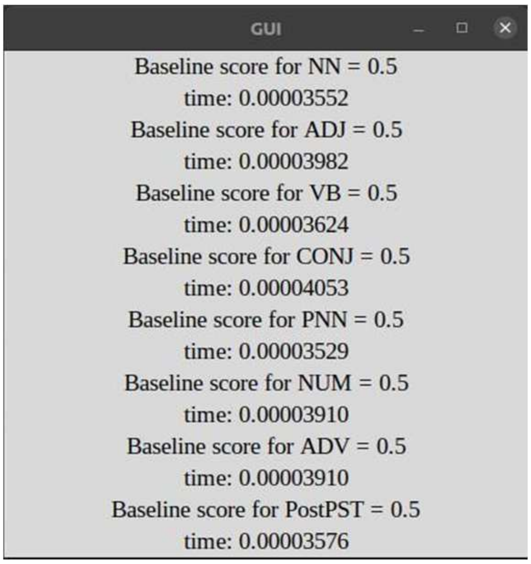


Fig. 2. Machine Learning algorithms for prediction of parts-of-speech of the Turkish language



```
Baseline score for NN = 0.5
time: 0.00005174
Baseline score for ADJ = 0.5
time: 0.00005674
Baseline score for VB = 0.5
time: 0.00005293
Baseline score for CONJ = 0.5
time: 0.00005054
Baseline score for PNN = 0.5
time: 0.00004649
Baseline score for NUM = 0.5
time: 0.00005341
Baseline score for ADV = 0.5
time: 0.00006247
Baseline score for PostPST = 0.5
time: 0.00005150
```

Fig. 3. Baseline solutions for part-of-speech tagging of the Kazakh language



```
Baseline score for NN = 0.5
time: 0.00003552
Baseline score for ADJ = 0.5
time: 0.00003982
Baseline score for VB = 0.5
time: 0.00003624
Baseline score for CONJ = 0.5
time: 0.00004053
Baseline score for PNN = 0.5
time: 0.00003529
Baseline score for NUM = 0.5
time: 0.00003910
Baseline score for ADV = 0.5
time: 0.00003910
Baseline score for PostPST = 0.5
time: 0.00003576
```

Fig. 4. Baseline score for part-of-speech determination task of the Turkish language

Table 2

Machine Learning algorithms for the Kazakh language

Algorithms Part of speech	LogReg	KNN	Des. Tree	Rand. For	SVM
Noun	0.7004	0.5330	0.5605	0.6292	0.6896
Adjective	0.7167	0.6360	0.5793	0.6675	0.7153
Verb	0.6372	0.5700	0.5329	0.5851	0.6549
Conjunction	0.7810	0.6084	0.4887	0.7501	0.7753
Pronoun	0.9067	0.8895	0.6395	0.8233	0.9231
Numeral	0.5838	0.5055	0.4895	0.6444	0.5776
Adverb	0.7704	0.6873	0.6185	0.7397	0.7711
PostPST	0.6585	0.5496	0.5423	0.5975	0.5299

Table 3

Machine Learning algorithms for the Turkish language

Algorithms Part of speech	LogReg	KNN	Des. Tree	Rand. For	SVM
Noun	0.7594	0.7972	0.7733	0.7347	0.7216
Adjective	0.9086	0.7770	0.7517	0.8220	0.8623
Verb	0.821	0.8423	0.8266	0.8011	0.8051
PostPST	0.8464	0.9113	0.9215	0.8566	0.9027
Pronoun	0.9510	0.8663	0.6462	0.8630	0.9685
Numeral	0.9856	0.9542	0.9537	0.9835	0.9908
Conjunction	0.9553	0.9643	0.9655	0.9451	0.9538
Adverb	0.8038	0.8315	0.7512	0.7947	0.8523
FW	0.8467	0.8936	0.8958	0.8064	0.8709

5. 3. Results of machine learning algorithms for the Turkish language

Turkish morphology is characterized by a high degree of stability and an almost complete absence of exceptions. In Turkish, there are no nominal classes and gender category. There are 9 parts of speech: Adverb, Adjective, Verb, Noun, Numeral, Conjunction, Pronoun, Postposition and Foreign words.

Part-of-speech determination for the Turkish language using machine learning algorithms. The first step in any machine learning task is the determination of the baseline solution. Fig. 4 shows the results of Zero-rule algorithm.

The results, i. e. the quality of the parts of speech determination, are represented below (Table 3).

Similarly, the best predictors for each part of speech in Turkish are determined. Datasets for the Turkish language were parsed from newspapers such as Akşam and Hürriyet Daily News. The best predictable algorithm for Turkish nouns and verbs is K-nearest neighborhoods, SVM is best suited for pronouns, numerals and adverbs. The best algorithm for adjectives is Logistic regression. For other parts of speech, the best machine learning algorithm is Decision Tree.

5. 4. Fulfilling Dictionaries of Link Grammar Parser

The methodological basis for the study of various objects arising in computational linguistics and the verification of their logical properties are the concepts and constructions of mathematical logic: the calculus of first-order predicates, the model, the concept of the truth of a formula on a model, etc. Of great interest are constructions and concepts of mathematical logic such as: Genkin's construction, implementation and omission of types, model completeness, forcing, as well as a number of non-classical logics.

To solve the tasks set, it is proposed to use a method for representing semantic and syntactic relations between semantic units of a sentence based on the diagrams of the LGP software system. The diagrams obtained by the LGP analyzer are graphs. This is followed by their preliminary preparation for comparison and the comparison itself. The comparison of graphs will be carried out not only at the lexical level, but also at the level of connections between words. Of particular interest are algorithms based on checking a number of logical properties.

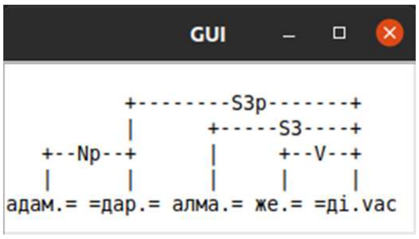
The methodological basis for creating algorithms for determining the topics of texts will be the article by Niraj Kumar, which describes a method that allows taking into account the word order and quite effectively solves the problem of determining topics and abstracting.

LGP is a syntactic analyzer of natural languages developed at the Carnegie Mellon University, USA. It is noteworthy that, in general, the underlying theory differs from the classical theory of syntax. Having received a sentence, the system attributes it with a syntactic structure, which consists of a set of marked connectors (links) connecting the pairs of words. A detailed description of the system can be found in [17, 25]. Currently, there are LGP variants for English, Russian, German, Arabic, Persian, etc.

LGP makes parsing using prepared dictionaries filled manually. To make this work faster, we decided to use all previously explained algorithms.

Within the framework of a joint project, in which the authors of the paper participated, a representative system of links for the Turkic languages was developed, on the basis of which prototypes of the LGP software complex for the Kazakh and Turkish languages were implemented [17, 22].

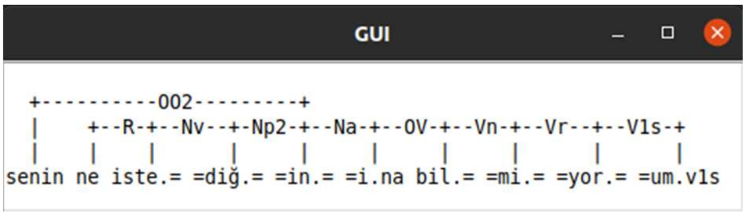
During the automatic analysis of sentences, LGP identifies both morphological and syntactic links simultaneously. For example, by parsing the sentence "Адамдар алма жеді" [in English: People ate an apple], the analyzer identified two syntactic (S3p, OV) and two morphological (Np, Va3p) connections (Fig. 5).



```
          +--------S3p-------+
          |      +-----S3----+
   +--Np--+      |     +--V--+
   |      |      |     |     |
адам.= =дар.= алма.= же.= =дi.vac
```

Fig. 5. Example of parsing the sentence in the Kazakh language

Below is an example of parsing a sentence with a possessive pronoun in Turkish: "Senin ne istedigini bilmiyorum" [in English: I do not know what do you want] (Fig. 6).



```
  +----------002---------+
  |    +--R-+--Nv--+-Np2-+--Na-+--OV-+--Vn-+--Vr--+--V1s-+
  |    |    |      |     |     |     |     |      |      |
senin ne iste.= =diğ.= =in.= =i.na bil.= =mi.= =yor.= =um.v1s
```

Fig. 6. Example of parsing the sentence in the Turkish language

The system for fulfilling dictionaries of LGP essentially uses the identifiers of parts of speech. The experiments have shown that the algorithms are working successfully.

6. Discussion of the results of the rule-based part of speech algorithm

This paper is devoted to the study of algorithms that, penetrating into the structure of the text, can deduce an adequate assessment of the relevance of the text to the search query, determine the topics presented in the text, form an abstract based on the text. It is important that such algorithms are based on the use of contexts and are not limited only to keywords, their proximity or frequency. The proposed approach is based on the use of the link grammar, built on its basis by the LGP software system and the methods of mathematical logic.

Development of a system of connections (morphological and syntactic) for the Turkic languages, and implementation of tools for constructing connection diagrams on the platform of the LGP software system for the Kazakh and Turkish languages.

In this paper, the author focuses on methods that are not statistical at all. An agglutinative language is a language that has a structure in which the dominant type of inflection is agglutination ("gluing") of various formants (suffixes or prefixes), each of which carries only one meaning. Kazakh and Turkish belong to the type of synthetic agglutinative languages of the Turkic group of the Altai family. They have a complex and rich morphology.

Our results (Tables 1, 2) showed that algorithms are based on machine learning techniques.

In this paper, we consider the binary classification of words according to parts of speech. We decided to take the most popular machine learning algorithms. In this paper, the following approaches and well-known machine learning algorithms are studied and considered. We defined 7 dictionaries and tagged 135 million words in Kazakh and 9 dictionaries and 50 million words in the Turkish language.

Many teams are currently working on the creation of systems for morphological and syntactic analysis. In most of the works, the authors limit themselves to considering the morphological structure of the Kazakh or Turkish languages, carry out their comparative analysis. There are a small number of studies on syntax and semantics. Our main task is to demonstrate the connectivity of different levels of analysis: morphological, syntactic and semantic. On the example of the Turkic languages, in some cases this is easier to do than for the Russian language. At the same time, this choice is due to the active spread of the Turkic culture and the fact that texts in these languages are widely represented on the Internet.

When processing text in an agglutinative language, it is wrong to try to isolate semantic analysis into a separate stage. This is directly related to the peculiarities of word formation in languages of such a structure. Already at the stages of morphological and syntactic analysis, semantic relations arise. In this paper, we restrict ourselves to considering the semantic relationships within a single sentence. In the future, it is planned to expand their list for marking up texts (sequences of related sentences).

LGP was chosen as a tool that allows for syntactic and semantic analysis of sentences.

The analysis is carried out by analogy with the assembly of a puzzle (puzzles correspond to the analyzed sentence) from its pieces (separate words). The language is represented by a dictionary or vocabulary, which consists of words and a set of allowed "puzzle forms" that words can have. Usually, words in them consist of a base and affixes added to it (suffix+ending), of which there are at least two or three.

Currently, plug-in dictionaries have been developed for English, Russian, Persian, Arabic, German, Lithuanian, Vietnamese, and Indonesian. We have developed dictionaries for the Kazakh and Turkish languages:

1. The methods of improving the quality of information search based on the grammar of relations, including taking into account the paraphrasing of sentences, are proposed. The methods are based on the use of diagrams generated by the LGP.

2. The analysis of works on agglutinative languages was carried out, and as a result, a representative system of links for the Turkic languages was developed, on the basis of which the prototypes of the LGP software system for the Kazakh and Turkish languages were implemented.

3. The models of determining the topics of texts in natural language, the graphs associated with them, the corresponding concepts and quality assessments are studied. The basis was the work of Niraj Kumar et al. The texts in Russian, English, Kazakh and Turkish were considered.

4. A software toolkit for analyzing texts in natural language has been implemented, including various algorithms: determining the degree of proximity of sentences, constructing graphs by sentences, calculating word weights, centralities and other characteristics. The created toolkit allows for large-scale testing and improvement of information retrieval algorithms in natural language, including the Kazakh and Turkish languages, giving a high degree of relevance of the result to the query. As a general conclusion, we can state that as soon as we move away from simple metrics and such factors as, for example, frequency and proximity, and try to work with grammatical structures, the complexity of algorithms increases significantly, and they also become much less resistant to various modifications. However, in some cases, significantly better results can be obtained. Comprehensive testing and participation of experts are becoming particularly important. The proposed method seems promising, and variants of the Niraj Kumar algorithm that take into account syntactic connections can even be represented as limiting cases of generalization. In order to move further, new ideas are needed.

During the implementation of the LGP software system for the Kazakh and Turkish languages, technical difficulties arose associated with the ordering of linguistic material, with encodings and programs used. All questions have been successfully resolved.

It should be noted that this work has development prospects. It is likely that the proposed approach could be further refined. Usually, heuristic algorithms are sensitive to small changes in their constituent parts. The paper describes possible variations of the discussed algorithm. To find the best configuration in which the algorithm will provide better results, you should test various possible combinations of variations of this algorithm on different data. It is also advisable to involve experts to assess the quality of the modifications of this algorithm, because it is people who can assess how accurately the algorithm works. All this constitutes a huge amount of work. It is also possible that while studying the obtained test results, new opportunities for improving the described algorithm will become visible.

7. Conclusions

1. In this paper, we created a rule-based algorithm for dataset tagging. The proposed approach was successfully evaluated for two agglutinative languages: Kazakh and Turkish. Languages with a rich morphology open up much more opportunities for labeling. The rule-based system passes several tests after the segmentation and function extraction stage. Analyzing affixes and word patterns, a set of grammatical rules is used. Some examples in the paper were shown when only the rules are applied. A rule-based system is fairly easy to expand, maintain, and change. The proposed approach can also be applied to other NLP processing tasks.

2. In this paper, we conducted a comprehensive review of machine learning algorithms for determining the part of speech of the Kazakh language. The machine learning algorithms for solving the problem of part of speech were tested. We defined 7 dictionaries and tagged 135 million words in the Kazakh language.

3. According to the test results, the best machine learning algorithms for parts of the Turkish language speech were determined. In accordance with our goal, we defined 9 dictionaries and 50 million words in the Turkish language.

4. Special dictionaries of the Kazakh and Turkish languages for LGP were created. The analysis of works on agglutinative languages was carried out, and as a result, a representative system of links for the Turkic languages was developed, on the basis of which prototypes of the LGP software system for the Kazakh and Turkish languages were implemented. On the basis of tagged datasets, we built machine learning models for faster future data labeling. For the Kazakh language, logistic regression is the best approach for nouns, adjectives, verbs, conjunctions, and postpositions, with values of 0.7004, 0.7167, 0.6373, 0.781, and 0.6585, respectively. Random Forest Classifier is the best method for numerals and SVM Classifier is the best method for pronouns and adverbs. However, the result for Turkish is not the same. K-nearest neighborhoods is the most predictable method for nouns, verbs, postpositions, and conjunctions, whereas SVM is best for pronouns, numerals, and adverbs. Logistic regression is the best machine learning method for adjectives with a cross validation score of 0.9086, and Decision Tree is the best machine learning algorithm for foreign words (0.8973). The random forest method displays the meanest results.

Acknowledgments

Firstly, we would like to offer special thanks to Dr. Feodor Murzin who, although no longer with us, continues to inspire by his example and dedication to the students he served over the course of his career.

This research has been funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant No. AP08857179).

References

1. StanfordNLP v0.2.0. python 3.6 | 3.7. Available at: https://stanfordnlp.github.io/stanfordnlp/performance.html
2. Batura, T. V., Murzin, F. A. (2008). Mashinno-orientirovannye logicheskie metody otobrazheniya semantiki teksta na estestvennom yazyke. Novosibirsk: Izd. NGTU, 248.

3. Yerimbetova, A. S., Sagnayeva, S. K., Murzin, F. A., Tussupov, J. A. (2018). Creation of Tools and Algorithms for Assessing the Relevance of Documents. 2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC). doi: https://doi.org/10.1109/rpc.2018.8482202
4. Index to Link Grammar Documentation. Available at: https://www.link.cs.cmu.edu/link/dict/index.html
5. Mel'chuk, I. A. (1974). Opyt teorii lingvisticheskih modeley «Smysl ↔ Tekst». Moscow: Nauka.
6. Paducheva, E. V. (2010). Semanticheskie issledovaniya: Semantika vremeni i vida v russkom yazyke. Semantika narrativa. Moscow: Yazyki slavyanskoy kul'tury, 480.
7. Kasekeyeva, A. B., Batura, T. V., Efimova, L. V., Murzin, F. A., Tussupov, J. A., Yerimbetova, A. S., Doshtayev, K. Zh. (2020). Link grammar and formal analysis of paraphrased sentences in a natural language. Journal of Theoretical and Applied Information Technology, 98 (10), 1724–1736. Available at: http://www.jatit.org/volumes/Vol98No10/10Vol98No10.pdf
8. Kumar, N., Srinathan, K., Varma, V. (2012). Using Graph Based Mapping of Co-occurring Words and Closeness Centrality Score for Summarization Evaluation. Lecture Notes in Computer Science, 353–365. doi: https://doi.org/10.1007/978-3-642-28601-8_30
9. Exactus. Available at: http://www.exactus.ru/
10. Avtomaticheskaya Obrabotka Teksta. Available at: http://www.aot.ru/
11. Sochenkov, I. V. (2013). Metod sravneniya tekstov dlya resheniya poiskovo-analiticheskih zadach. Iskusstvennyy intellekt i prinyatie resheniy, 2, 32–43. Available at: http://www.isa.ru/aidt/images/documents/2013-02/32_43.pdf
12. Batura, T. V., Murzin, F. A., Semich, D. F., Sagnayeva, S. K., Tazhibayeva, S. Z., Bakiyev, M. N. et. al. (2016). Using the link grammar parser in the study of turkic languages. Eurasian Journal of Mathematical and Computer Applications, 4 (2), 14–22. doi: https://doi.org/10.32523/2306-6172-2016-4-2-14-22
13. Zura, D., Doyle, W. J. (2018). A Grammar of Kazakh. Durhame: Duke University, Duke Center for Slavic, Eurasian, and East European Studies, 69. Available at: https://www.twirpx.com/file/2587861/
14. Göksel, A. (2015). Phrasal compounds in Turkish: Distinguishing citations from quotations. STUF - Language Typology and Universals, 68 (3), 359–394. doi: https://doi.org/10.1515/stuf-2015-0017
15. Sultanova, N., Kozhakhmet, K., Jantayev, R., Botbayeva, A. (2019). Stemming algorithm for Kazakh Language using rule-based approach. 2019 15th International Conference on Electronics, Computer and Computation (ICECCO). doi: https://doi.org/10.1109/icecco48375.2019.9043253
16. Makhambetov, O., Makazhanov, A., Yessenbayev, Z., Matkarimov, B., Sabyrgaliyev, I., Sharafudinov, A. (2013). Assembling the Kazakh Language Corpus. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 1022–1031. Available at: https://aclanthology.org/D13-1104.pdf
17. Aksan, Y., Aksan, M., Koltuksuz, A., Sezer, T., Mersinli, Ü., Demirhan, U. U. et. al. (2012). Construction of the Turkish National Corpus (TNC). Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), 3223–3227. Available at: http://www.lrec-conf.org/proceedings/lrec2012/pdf/991_Paper.pdf
18. Smola, A., Vishwanathan, S. V. N. (2008). Introduction to machine learning. Cambridge University Press, 234. Available at: https://alex.smola.org/drafts/thebook.pdf
19. Markus, S. (1970). Teoretiko-mnozhestvennye modeli yazykov. Moscow: Nauka, 332.
20. Murzin, F. A., Tussupova, M. J., Yerimbetova, A. S. (2018). Filling up Link Grammar Parser dictionaries by using Word2Vec techniques. Joint issue of the International Conference, Computational and Information Technologies in Science, Engineering and Education (CITech-2018). Ust-Kamenogorsk-Novosibirsk, 169–176. Available at: http://www.ict.nsc.ru/jct/getfile.php?id=1920
21. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. arXiv.org. Available at: https://arxiv.org/abs/1310.4546
22. Batura, T. V., Bakieva, A. M., Erimbetova, A. S., Murzin, F. A., Sagnaeva, S. K. (2018). Grammatika svyazey, relevantnost' i opredelenie tem tekstov. Novosibirsk: Izd-vo SO RAN, 91. Available at: http://lib.iis.nsk.su/node/277940
23. Krippes, K. A. (1996). Kazakh Grammar with Affix List. Dunwoody Press, 84. Available at: http://www-lib.tufs.ac.jp/opac/en/recordID/catalog.bib/BA36636430
24. Makazhanov, A., Yessenbayev, Z., Sabyrgaliyev, I., Sharafudinov, A., Makhambetov, O. (2014). On certain aspects of Kazakh part-of-speech tagging. 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT). doi: https://doi.org/10.1109/icaict.2014.7035953
25. The CMU Link Grammar natural language parser. Available at: https://github.com/opencog/link-grammar