Ngoc Thanh Nguyen · Bao Hung Hoang ·
Cong Phap Huynh · Dosam Hwang ·
Bogdan Trawiński · Gottfried Vossen (Eds.)

LNAI 12496

Computational Collective Intelligence

12th International Conference, ICCCI 2020
Da Nang, Vietnam, November 30 – December 3, 2020
Proceedings

Springer

Lecture Notes in Artificial Intelligence 12496

Subseries of Lecture Notes in Computer Science

More information about this series at http://www.springer.com/series/1244

Ngoc Thanh Nguyen · Bao Hung Hoang ·
Cong Phap Huynh · Dosam Hwang ·
Bogdan Trawiński · Gottfried Vossen (Eds.)

Computational Collective Intelligence

12th International Conference, ICCCI 2020
Da Nang, Vietnam, November 30 – December 3, 2020
Proceedings

Springer

Editors
Ngoc Thanh Nguyen
Department of Applied Informatics
Wrocław University of Science
and Technology
Wroclaw, Poland

Faculty of Information Technology
Nguyen Tat Thanh University
Ho Chi Minh, Vietnam

Cong Phap Huynh
Vietnam - Korea University of Information
and Communication Technology
University of Da Nang
Da Nang, Vietnam

Bogdan Trawiński
Department of Applied Informatics
Wrocław University of Science
and Technology
Wroclaw, Poland

Bao Hung Hoang
Thua Thien Hue Center of Information
Technology
Hue, Vietnam

Dosam Hwang
Department of Computer Engineering
Yeungnam University
Gyeungsan, Korea (Republic of)

Gottfried Vossen
Department of Information Systems
University of Münster
Münster, Germany

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-030-63006-5 ISBN 978-3-030-63007-2 (eBook)
https://doi.org/10.1007/978-3-030-63007-2

LNCS Sublibrary: SL7 – Artificial Intelligence

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

for analysis and decision making for the “logistics-agglomeration”; system in the framework of the digital economy of the Republic of Kazakhstan» (2018–2020).

References

1. Embley, W.D., Campbell, M.D., Smith, D.R.: Ontology-based extraction and structuring of information from data-rich unstructured documents. In: Information and Knowledge Management (1998)
2. Cardie, C.: A case-based approach to knowledge acquisition for domain-specific sentence analysis. In: Eleventh National Conference on Artificial Intelligence, pp. 798–803. AAAI Press (1993)
3. Scheffer, T., Decomain, C., Wrobel, S.: Active hidden Markov models for information extraction. In: International Symposium on Intelligent Data Analysis (2001)
4. Scheffer, T., Wrobel, S., Popov, B., Ognianov, D., Decomain C., Hoche, S.: Learning hidden Markov models for information extraction actively from partially labeled text. K¨unstliche Intelligenz (2) (2002)
5. Skounakis, M., Craven, M., Ray, S.: Hierarchical hidden Markov models for information extraction. In: IJCAI (2003)
6. McCallum, A.K., Freitag, D., Pereira, F.: Maximum entropy Markov models for information extraction and segmentation. In: ICML (2000)
7. McCallum, A.K., Jensen, D.: A note on the unification of information extraction and data mining using conditional-probability, relational models. In: IJCAI’03 Workshop on Learning Statistical Models from Relational Data (2003)
8. Mansurova, M., Barakhnin, V., Khibatkhanuly, Y., Pastushkov, I.: Named entity extraction from semi-structured data using machine learning algorithms. In: Nguyen, N.T., Chbeir, R., Exposito, E., Aniorté, P., Trawiński, B. (eds.) ICCCI 2019. LNCS (LNAI), vol. 11684, pp. 58–69. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28374-2_6
9. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language processing, Computational Linguistics, and Speech Recognition. Pearson Prentice Hall, Upper Saddle River (2009)
10. Peters, M., Neumann, M., Iyyer, M., et. al.: Deep contextualized word representations. In: 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long Papers), pp. 2227–2237 (2018)
11. Devlin, J., Chang, M., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. Computing Research Repository (2018)
12. Chikibayeva D., Mansurova M., Nugumanova A., Kyrgyzbayeva M.: Named entity recognition from news sources based on BI-LSTM. In: 2019 IICT Conference, pp. 519–525 (2019)
13. Luo, G., Huang, X., Lin, C., Nie, Z.: Joint entity recognition and dis-ambiguation. In: 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp 879–888. Association for Computational Linguistics (2015)
14. Rau, L.F.: Extracting company names from text. In: Seventh IEEE Conference on Artificial Intelligence Applications, vol. 1, pp. 29–32. IEEE (1991)
15. Sekine, S., Nobata, C.: Definition, dictionaries and tagger for extended named entity hierarchy. In: LREC, pp. 1977–1980 (2004)
16. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. In: Lingvisticae Investigationes (2007)

17. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: The seventh conference on Natural language learning at HLT-NAACL 2003, vol, 4, pp. 188–191. Association for Computational Linguistics (2003)
18. Kubis, M.: Quantitative analysis of character networks in polish XIX and XX century novels. In: Digital Humanities 2019 Conference, Utrecht, The Netherlands (2019)
19. Bikel, D.M, Miller, S., Schwartz, R., Weischedel, R.: Nymble: a high-performance learning name-finder. In: Fifth Conference on Applied natural language processing, pp. 194–201. Association for Computational Linguistics (1997)
20. Nugumanova, A., Baiburin, Y., Apaev, K.: A new text representation model enriched with semantic relations. In: ICCAS 2015 – 2015 15th International Conference on Control, Automation and Systems, Proceedings (2015)
21. Nugumanova, A.B., Apayev, K.S., Baiburin, Y.M., Mansurova, M.Y.: A contrastive approach to term extraction: case-study for the information retrieval domain using BAWE corpus as an alternative collection. Eurasian J. Math. Comput. Appl. **5**, 73–86 (2017)
22. Asahara, M., Matsumoto, Y.: Japanese named entity extraction with redundant morphological analysis. In: 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 8–15. Association for Computational Linguistics (2003)
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. In: Neural Computation, pp. 1735–1780 (1997)
24. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling (2014)
25. Mikolov, T., Karafi, M., Burget, L., Cernock‘y, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech, vol. 2, p. 3 (2010)
26. Sundermeyer, M., Schl¨uter, R., Ney, H.: LSTM neural networks for language modeling. In: Interspeech, pp. 194–197 (2012)
27. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2014)
28. Kalchbrenner, N., Blunsom, P.: Recurrent convolutional neural networks for discourse compositionality (2013)
29. Tran, Q., Zukerman, I., Haffari, G.: A hierarchical neural model for learning sequences of dialogue acts. In: 15th Conference of the European Chapter of the Association for Computational Linguistics, vol. 1, Long Papers, pp. 428–437. Association for Computational Linguistics, Valencia (2017)
30. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF (2016)
31. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 260–270. Association for Computational Linguistics, San Diego (2016)
32. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research **12**, 2493–2537 (2011)
33. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural Architectures for Named Entity Recognition. CoRR (2016)
34. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging (2015)
35. Jason, P.C., Nichols, E.: Named Entity Recognition with Bidirectional LSTM-CNNs (2016)
36. Rei, M., Crichton, G., Pyysalo, S.: Attending to characters in neural sequence labeling models. In: 26th International Conference on Computational Linguistics, pp. 309–318 (2016)
37. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty oftraining recurrent neural networks (2012)

38. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: 18th International Conference on Machine Learning, pp. 282–289 (2001)
39. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. J. Mach. Learn. Res. **7**(Mar), 551–585 (2006)
40. Okazaki, N.: CRFsuite: a fast implementation of Conditional Random Fields (CRFs) (2007). http://www.chokkan.org/software/crfsuite/. Accessed 14 Feb 2020
41. Tjong, E.F., Sang, K., Meulder, F.D.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: CoNLL-2003, Proceedings, Edmonton, Canada, pp. 142–147 (2003)

Creation of a Dependency Tree for Sentences in the Kazakh Language

Darkhan Akhmed-Zaki[1,2], Madina Mansurova[1](✉), Nurgali Kadyrbek[1], Vladimir Barakhnin[3,4], and Armanbek Misebay[1]

[1] Al-Farabi Kazakh National University, Almaty, Kazakhstan
darhan_a@mail.ru, mansurova.madina@gmail.com, nurgaliqadyrbek@gmail.com, armanbek128@mail.ru
[2] Astana IT University, Nur-Sultan, Kazakhstan
[3] Institute of Computational Technologies, Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russian Federation
bar@ict.nsc.ru
[4] Novosibirsk State University, Novosibirsk, Russian Federation

Abstract. In the semantico-syntactic analysis of great importance is understanding of its formal structure. For this, in the text it is necessary to distinguish units of lexical meaning and designate the types of relations between them. The dependency tree is an indispensable tool for parsing sentences and determination of hierarchical relationships between the main components in it. In this work, an algorithm for constructing a dependency tree for sentences in the Kazakh language using the filter method is proposed. The dependency tree was created on the basis of the spinning tree from the oriented graph constructed according to the rules of syntactic relationship in the Kazakh language.

Keywords: Dependency tree · Dependency grammar · Corpus linguistics · Syntactic structure · Phrase structure rule

1 Introduction

When studying syntax which is an integral part of grammar, the main subject of research is the syntactical structure of the language. The syntactical structure is a set of system rules and laws of composing sentences. In its turn, a sentence is a single language unit organized according to grammatical rules in a particular language which is the main means of expressing thoughts, messages. Sentences are characterized by the following features: expressiveness, predicate relation, intonation consisting of separate words, word combinations [1–3]. In the research of the speech syntax in general linguistics, special attention is paid the sentence structure in the approach of a predicative construction which is a formal structure of a sentence. In this work, the authors consider the syntactical structure of a sentence for the description of which dependency trees were chosen as a basis. Let us consider this concept in more detail.

To present a formal structure in language, with a loose word order and case marking, a syntax of dependency trees is used [4]. A dependency tree is the most graphic and

N. T. Nguyen et al. (Eds.): ICCCI 2020, LNAI 12496, pp. 709–718, 2020.
https://doi.org/10.1007/978-3-030-63007-2_55

wide spread method of representing the syntactical structure of a sentence. Formalism of dependency trees presupposes construction of a graph of syntactic relationships between the words of a sentence. However, this graph, unlike the graph of a model tree of direct components, is not hierarchical [5]. The graph which serves as a basis for formalism of dependency trees must meet the following requirements: the graph has directed relationships between the sentence words - from the principal word to the dependent one; each word has only one parent; The graph does not contain cycles, syntax relations may be named termed but it is optional.

In this case, a sentence is presented as a linearly ordered set of elements (derivations forms) in which one can create an oriented tree with nodes from the elements of this set. Each rib connecting a pair of nodes indicates a subordinate relation between the main (subordinating) and dependent (subordinate) word corresponding to the direction of this rib.

Algorithms for creation of dependency trees are usually made up using the rules based on the strategy "it…,then…". These rules allow realizing a free logic concept mechanism, therefore they are widely used when representing knowledge in expert systems.

At present, two main methods of constructing a dependency tree based on the rules the method of fulcrums and the method of filters are used. The method of fulcrums [6] is used to search for the so-called fulcrums, i.e. the roots of syntax trees or some key supporting words according to the established rules. The filter method [7, 8] is based on the extraction of all possible phrases while applying the rules of syntactic relations after which a filter is used to the obtained structure (an oriented graph). It should be noted that the dependency tree like structures are a dispensable tool in solving the problems of syntactic analysts, machine translation, removal of homonymy, formal presentation of a speech construction [9–12]. The aim of this research is to develop an algorithm of building a dependency tree for sentences in the Kazakh language using the filter method. The dependency tree was created on the basis of the spinning tree from the oriented graph constructed according to the rules of syntactic relationships in the Kazakh language. This work is continuation of the authors investigations in the field of NLP for the Kazakh language [13, 14].

2 The Main Types of Syntactic Relationships of Words in Phrases for the Kazakh Language

According [15], there are 5 main types of syntactical connection of the words in the phrases in Kazakh language (see Fig. 1):

- *Kiyisu* – negotiation
- *Menggeru* – domination
- *Matasu* – subordination
- *Kabysu* – adjunction
- *Zhanasu* – convergence

Kiyisu (negotiation). In the Kazakh language, this type of connection is mostly used to connect subject and predicate. Kiyisu is the interconnection of subject and predicate

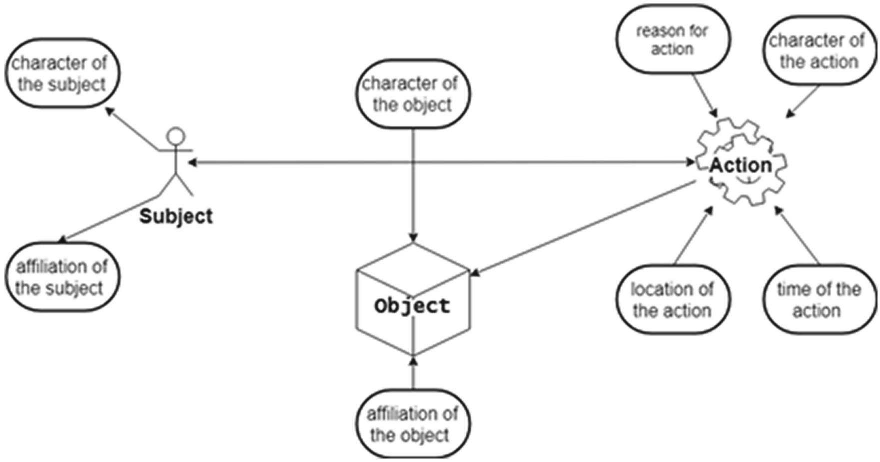

Fig. 1. Scheme of sentence and relations between the words.

in the level of person and plurality. In the word combination the subject initial and the predicative is narrative. Therefore, predicative would be the same form as subject. For example: *Men keldim, sen aittyn, biz erten baramyz.* The personal endings help words connect in the meaning of numeric meanings, and grammatical representation [16].

Mengeru (domination). Menggeru is the one of the main types of syntactical connection of the word phrases in which words are divided into initial and narrative. In this type of connection the form of the narrative depends on the ending of the initial word. But there is a big difference between kiyisu and menggeru. The interconnection of the initial word in the phrase with the narrative part in the level of case ending is menggeru. For example: *Kalammen zhazdy, dapterge zhazdy, kokzhiekten koterildi.* Narrative words with the case endings mostly are object or the object of preposition in the sentence.

Matasu (subordination). One of the forms of syntactic connection of words is matasu. It comes among the determined combinations of words, pronounced on 1 side (usually 3rd party) connection. For example: *Kunnin zhyluy, agamnyn balasy, ertennin isi.*

Kabysu (adjunction). The interconnection of the words in the phrase without any ending is called kabysu. Most conjugated words do not change the word order. A number of adjacent words change their place in the sentence and are removed from the words under which they obey. Such remote communication is subject to contact. It is not allowed to give other words between related combinations. At the same time, the syntactic relations of word combinations are changing. Adjacent phrases are words defined by the directory, worked out refined, supplemented. The first word, the connecting link, is the noun, the adjective, the demonstrative pronouns, the verbs, the adverbs, the adverbs of the verbs, the leader, and the second is the noun or verb. For example: Adal adam, zhauapty tulga, beibit el, aitylgan soz.

Zhanasu (convergence). The interconnection of the initial word and the narrative word without any ending despite its position in the sentence is called zhanasu. In this type of connection, the words can be objects of the preposition objects of the preposition. They are used in such samples below:

Table 1. Morphological features of the types of syntactic connection

Kind of syntactic connection	Phrase structure rules	Morphological features of syntactic connection
Kiyisu negotiation	<SP>–<VP>	The connection of words using personal and plural endings
Matasu subordination	<N>–<SP>	The connection of the word in the genitive case and the word with the possessive ending <word + genitive case> <word + possessive ending> <word > < word + possessive ending> <verb + participle + genitive case> <word + possessive ending>
Menggeru domination	<OP>–<VP>	The connection of words using the endings of the cases except the genitive <word + accusative case ending><verb> <word + dative case ending><verb> <word + instrumental case ending><verb> <word + locative case ending><verb> <word + nominative case ending><verb>
Kabysu adjunction	<ADJ>–<SP> <ADJ>–<OP>	The connection of words without any postpositions only due to word order. Usually no other words between them. <noun><noun> <adjective><noun> <numeral><noun> <locative><noun> <noun + dative case ending><participle> <noun + accusative/locative/nominative/ instrumental case ending><participle> <adverb><noun/adjective/numeral/pronoun>
Zhanasu convergence	<ADV>–<VP>	The connection of words without any postpositions. Basically, the combination of adverb and verb is formed by participation < adverb><verb> < participle><verb>

Here SP – subject predicative phrase structure, a description/qualification of the sub-ject. Adjectival or nominal;
OP – object predicative phrase structure, a description/qualification of the direct ob-ject. Adjectival or nominal;
VP – verb phrase structure;
N – noun; NUM – numeral; ADJ – adjective; ADV – adverb; CONN – conjunction; AUX – auxiliary verb.

– Objects of time adverbs.
– Objects with the case endings.
– Objects of preposition.

For example: *Biyl bitirdi, erten keledi, aptyga soiledi.*

At present, syntactic dependencies in word combinations for the Kazakh language are determined according to the main rules using phrase structure elements (Table 1) to describe a given languages syntax and are closely associated with the early stages of transformational grammar proposed by Noam Chomsky [17].

Phrasal structures of the Kazakh language can take the following forms:

<SP>::=<N>|<ADJ><SP>|<NUM><SP>|<N><SP>|<SP><CONN><SP>
<OP>::=<N>|<ADJ<OP>|<NUM><OP>|<OP><CONN><OP>
<VP>::=<VP>|<AUX><VP>|<ADV><VP>

3 Construction Dependency Trees for Simple Sentences of the Kazakh Language

The dependency grammar deals with taxanomic units. All relationships in the dependency grammar are considered subordinate. Analysis of syntactic relationships in word combinations and simple sentences in the Kazakh language is presented in [18]. The author notes that in the Kazakh language the arrangement of tokens in a sentence obeys a strict law.

The process of a dependency tree construction consists of several stages (see Fig. 2).

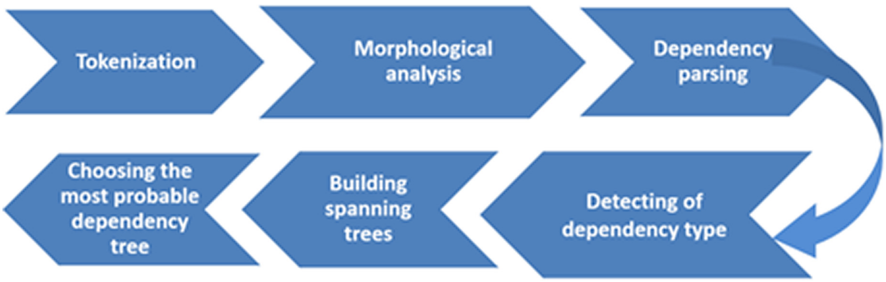

Fig. 2. The process of constructing a dependency tree.

For example: Korshiles turgyndar gimarattyn tobesinde ornalaskan agashtan zhasalgan kyzyl shatyry ortengenin anyktady. Figure 3 shows the dependency tree of this sentence, the vertices are tokens with their indices in the sentence.

A linear order of words in the tree is not reblected and one and the same tree may correspond to several orders. However, the ratio of syntactic dependencies and the order of words is not arbitrary. One to the property of sentences in a natural language- their projectivity was discovered [19]. A sentence is called projective if all dependency arrows are drawn on one side of the line on which the sentence is written and:

- none of the arrows crosses any other arrow,
- none of the arrows covers the root node (the principle of not projectivity is also preserved for the word order in phrases).

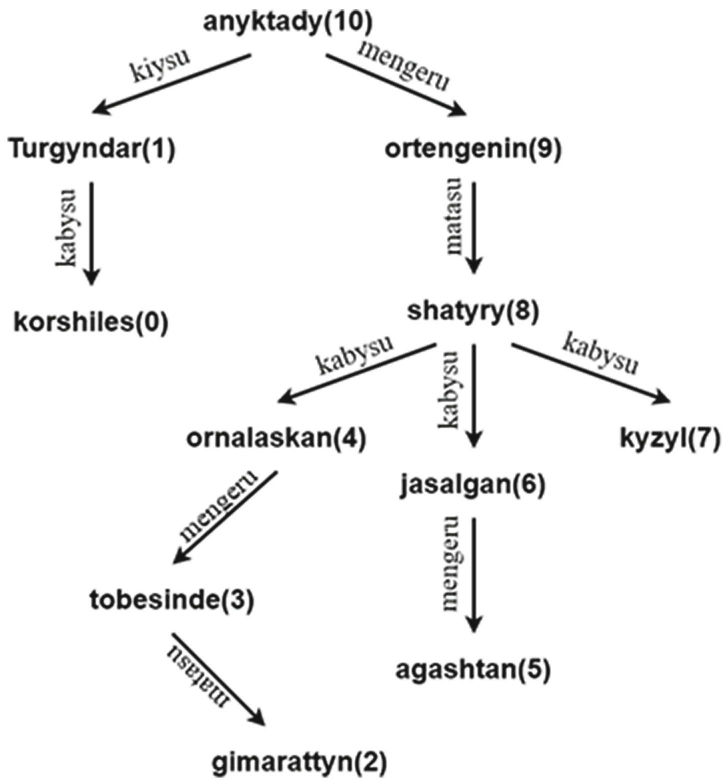

Fig. 3. An example of a constructed dependency tree.

Example 1.
Korshiles (0) turgyndar (1) gimarattyn (2) tobesinde (3) ornalaskan (4) agashtan (5) zhasalgan (6) kyzyl (7) shatyry (8) ortengenin (9) anyktady (10).
Phrase model of the sentence: <SP><OP><VP> (see Fig. 4).

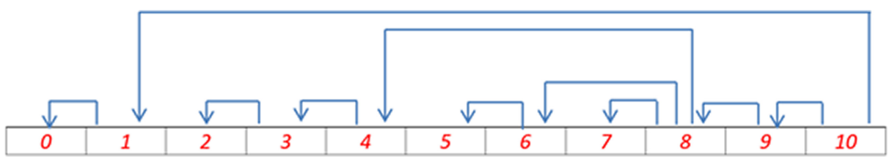

Fig. 4. Phrasal model of the sentence indicating the direction of syntactic connection for example 1.

Example 2.
Gimarattyn (0) tobesinde (1) ornalaskan (2) agashtan (3) zhasalgan (4) kyzyl(5) shatyry (6) ortengenin (7) korshiles (8) turgyndar (9) anyktady (10).
Phrase model of the sentence: <OP><SP><VP> (see Fig. 5).

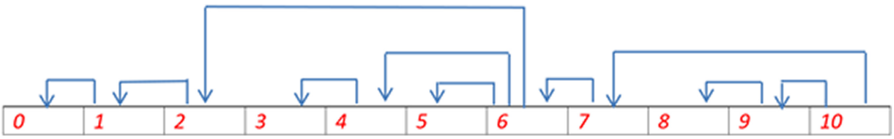

Fig. 5. Phrasal model of the sentence indicating the direction of syntactic connection for example 2.

The property of projectivity may be used in problems of automatic detection of syntactic relationships. It allows avoiding the detection of false relationships. Besides, the property of syntactic relationships. The conditions of projectivity and poor projectivity allow predicting the relationships between words or exclude impossible relationships [20].

There are two approaches to construction of a dependency tree: via building a spinning tree and using the formal model of Backus [21]. In the work, the approach of building a spinning tree was used.

4 The Algorithm of Building a Spinning Tree

The spinning tree is a tree covering all vertices of graph G<V,E>, where V – vertices of the graph a set of tokens in the sentence, E – ribs of the graph, and indicating the dependency between word combinations. The authors of [22] shows the method of obtaining the spinning tree from the oriented graph as well as obtaining several spinning trees from one graph. This multiplicity is due to the fact that the path to some vertices of the graph can be constructed in several ways.

The spinning tree algorithm [23]:

1) from graph G, we choose vertex u_1 which can become a tree forming an inner graph, for example, let us suppose that $i = 1$ (usually the main pair of word combinations forming the relationships is chosen);
2) if $i = n(G)$, the found spinning tree. In the other case, we pass to step 3;
3) let us suppose that graph G_i covering vertices $u_1, u_2, \ldots u_i$ is created which will be an inner graph of G where $1 \leq i \leq n-1$. Then adding a new vertex $u_{i+1} \in V$ adjacent to the vertex u_i of the graph G_i, create the graph G_{i+1}, thus adding an edge $\{u_{i+1}, u_i\}$. In addition, G_{i+1} is a tree, since the graph G_i did not cover the cycle. We perform $i := i + 1$ and continue further, that is, go to step 2.

Dependency search rules:

1. Search for relationships kosymsha: identify all words adjacent to the prepositions;
2. To search for relations kiyisu: determine the root of the tree - the word at the beginning of the sentence, the verb;
3. Perform a kabysu relationship search;
4. Perform a matasu relationship search;
5. Perform a zhanasu relationship search.

5 Experiment Results

In the constructed adjacency matrix, the rows and columns corespondent to the words in a sentence. The value of matrix element a_{ij} equal to 0 indicates the absence of a syntactic relationships between the word and the i^{th} word and j^{th} word in which i^{th} word a principal. The value of matrix element a_{ij} which is not equal to 0 indicates the fact that the i^{th} word is principal in the syntactic relationships with the j^{th} word and the value of a_{ij}, indicates the type of relationships (see Fig. 6):

$$
A = \begin{matrix}
0&0&0&0&0&0&0&0&0&0&0\\
5&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0\\
0&0&3&0&0&0&0&0&0&0&0\\
0&0&0&2&0&0&0&0&0&0&0\\
0&0&0&0&5&0&0&0&0&0&0\\
0&0&0&2&0&2&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&5&3&0&0&0\\
0&0&0&0&0&2&0&0&2&0&0\\
0&0&0&0&0&0&0&0&0&2&0
\end{matrix}
$$

1 – kiysu
2 – mengeru
3 – matasu
4 – zhanasu
5 – kabysu
6 – kosymsha

Fig. 6. Adjacency matrix (on the left), types of syntactic connections (on the right).

The analyzer returns 4 types of spanning tree for the considered example 1 (see Fig. 7):

1)	2)	3)	4)
the graph is:	the graph is:	the graph is:	the graph is:
vertex 0 : []	vertex 0 : []	vertex 0 : []	vertex 0 : []
vertex 1 : [10, 0]	vertex 1 : [10, 0]	vertex 1 : [10, 0]	vertex 1 : [10, 0]
vertex 2 : []	vertex 2 : []	vertex 2 : []	vertex 2 : []
vertex 3 : [2]	vertex 3 : [2]	vertex 3 : [2]	vertex 3 : [2]
vertex 4 : [3]	vertex 4 : [3]	vertex 4 : []	vertex 4 : []
vertex 5 : []	vertex 5 : []	vertex 5 : []	vertex 5 : []
vertex 6 : [5]	vertex 6 : []	vertex 6 : [5, 3]	vertex 6 : [3]
vertex 7 : []	vertex 7 : []	vertex 7 : []	vertex 7 : []
vertex 8 : [4, 6, 7]	vertex 8 : [4, 6, 7]	vertex 8 : [4, 6, 7]	vertex 8 : [4, 6, 7]
vertex 9 : [8]	vertex 9 : [8, 5]	vertex 9 : [8]	vertex 9 : [8, 5]
vertex 10 : [9]	vertex 10 : [9]	vertex 10 : [9]	vertex 10 : [9]

Fig. 7. Spanning trees generated for example 1.

Variant 1 correspondents to the expected results: coincidence – 100%, variant2: coincidence – 90%, variant 3: coincidence – 90%, variant 4: coincidence – 80%. The shortcoming at the given moment in the system of presentation of syntactic structures in the form of a dependency tree are:

1. a strict requirement to consider each token (parenthesis, isolated part of a sentence, idioms) as a separate element of a sentence. The cases of automatic representation of relationships between the following components of a sentence: idioms, parentheses, verbose expressions are not considered. For example: *Adam balasy garyshka zhyyrmasynshy gasyrda ayak basti.*
2. all relation in word combinations are considered as subordinate.

6 Conclusion

The work presents an algorithm for construction a dependency tree for sentences in the Kazakh language using the filter method and the dependency tree is created on the basis of the spinning tree from the oriented graph built according to the rules of syntactic relationships in the Kazakh language. Dependency trees are an indispensable tool in solution of the problems of syntactic analysis, machine translation, removal of homonymy, formal representation of a speech construction. The authors will go on with the investigations in this field including the task of detecting parenthetical words, isolated parts of a sentence, idioms.

Acknowledgements. This work was supported in part under grant of Foundation of Ministry of Education and Science of the Republic of Kazakhstan BR05236340 – «Creation of high-performance intelligent technologies for analysis and decision making for the "logistics-agglomeration" system in the framework of the digital economy of the Republic of Kazakhstan» (2018-2020).

References

1. Chomsky, N.: Aspects of the Theory of Syntax. The MIT Press, Cambridge (1965)
2. Chomsky, N.: Some Concepts and Consequences of the Theory of Government and Binding. MIT Press, Cambridge (1982)
3. Belletti, A.: Structures and Beyond: The Cartography of Syntactic Structures, vol. 3. Oxford University Press, Oxford (2004)
4. Mel'čuk, I.: Dependency Syntax: Theory and Practice. State University of New York Press, New York (1988)
5. Carnie, A.: Syntax: A Generative Introduction. Blackwell Publishing, Malden (2007)
6. McNamara, T.: Applied linguistics: the challenge of theory. Appl. Linguist. **36**(4), 466–477 (2015)
7. Leserf, I.: Application of the program and model of a specific situation to automatic syntactic analysis. NTI, no. 11, pp. S. 42–50. VINITI, Moscow (1963)
8. Fundel, K., Küffner, R., Zimmer, R.: RelEx – relation extraction using dependency parse trees. Bioinformatics **23**(3), 365–371 (2007)
9. Baiburin, Y., Zhantassova, Z., Nugumanova, A., Syzdykpayeva, A., Bessmertny, I.: The case study approach to learning Text Mining. In: AICT 2016 - Conference Proceedings on Application of Information and Communication Technologies (2016)
10. Nugumanova, A., Baiburin, Y., Apaev, K.: A new text representation model enriched with semantic relations. In: ICCAS 2015 – 2015 15th International Conference on Control, Automation and Systems, Proceedings (2015)

11. Mansurova, M., Barakhnin, V., Khibatkhanuly, Y., Pastushkov, I.: Named entity extraction from semi-structured data using machine learning algorithms. In: Nguyen, N.T., Chbeir, R., Exposito, E., Aniorté, P., Trawiński, B. (eds.) ICCCI 2019. LNCS (LNAI), vol. 11684, pp. 58–69. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28374-2_6
12. Nugumanova, A.B., Apayev, K.S., Baiburin, Y.M., Mansurova, M.Y.: A contrastive approach to term extraction: Case-study for the information retrieval domain using BAWE corpus as an alternative collection. Eurasian J. Math. Comput. Appl. (2017)
13. Alimzhanov, Y., Mansurova, M.: An approach of automatic extraction of domain keywords from the kazakh text. In: Nguyen, N.-T., Manolopoulos, Y., Iliadis, L., Trawiński, B. (eds.) ICCCI 2016. LNCS (LNAI), vol. 9876, pp. 555–562. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45246-3_53
14. Mansurova, M., Madiyeva, G., Aubakirov, S., Yermekov, Z., Alimzhanov, Y.: Design and development of media-corpus of the kazakh language. In: Nguyen, N.T., Papadopoulos, G.A., Jędrzejowicz, P., Trawiński, B., Vossen, G. (eds.) ICCCI 2017. LNCS (LNAI), vol. 10449, pp. 509–518. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67077-5_49
15. Ashirova, A.T.: Approaches of word combinations and connection forms. In: Proceedings of International scientific-methodical online conference «Modern Kazakh language in the conditions of competitiveness of civilization: research paradigms and mobile technologies of teaching» dedicated to the 80th anniversary of the Doctor of Philological Sciences, Professor Talgat Sairambaev Almaty, 28–29 April 2017 (2017)
16. Grammar, K.: Phonetics. Wordable Word. Morphology. Syntax.In: Zhanpeisov, O.E. (ed.) Astana, p. 784 (2002)
17. Chomsky, N.: Syntactic Structures. Mouton, The Hague/Paris (1957)
18. Sairanbaev, T.: Problems of Kazakh linguistics. Almaty, Abzal ay, p. 640 (2014)
19. Testelets, Y.G.: Introduction to General Syntax. Russian State University for the Humanities (2001)
20. Batura, T.V.: Mathematical linguistics and automatic word processing: textbook. In: Batura, T.V. (ed.) Novosib. RIC NSU, state un-t - Novosibirsk, p. 166 (2016)
21. Mehler, A., Lücking, A., Banisch, S., Blanchard, P., Job, B. (eds.): Towards a Theoretical Framework for Analyzing Complex Linguistic Networks. Springer-Verlag, Heidelberg (2016). https://doi.org/10.1007/978-3-662-47238-5
22. Uno, T.: An algorithm for enumerating all directed spanning trees in a directed graph. In: ISAAC'96: Proceedings of the 7th International Symposium on Algorithms and Computation, pp. 166–173 (1996)
23. Chakraborty, M., Chowdhury, S., Chakraborty, J., Mehera, R., Pal, R.K.: Algorithms for generating all possible spanning trees of a simple undirected connected graph: an extensive review. Complex Intell. Syst. **5**(3), 265–281 (2019)