Chomsky Was (Almost) Right: Ontology-based Parsing of Texts of a Narrow Domain*

Boris Geltser[1][0000-0002-9250-557X], Tatiana Gorbach[1][0000-0003-4380-6517], Valeria Gribova[2][0000-0001-9393-351X], Olesya Karpik[3][0000-0003-4380-6517], Eduard Klyshinsky[4][0000-0002-4020-488X], Dmitrii Okun'[2][0000-0002-6300-846X], Margarita Petryaeva[2][0000-0002-1693-4508], and Carina Shakhgeldyan[1,5][0000-0002-4539-685X]

[1] Far East Federal University boris.geltser@vvsu.ru
[2] Institute of Automation and Control Processes, FEB, RAS gribova@iacp.dvo.ru
[3] Keldysh Institute of Applied Mathematics, RAS
[4] National Research University Higher School of Economics eklyshinsky@hse.ru
[5] Vladivostok State University of Economics and Service carinashakh@gmail.com

Abstract. The common approach to analysis of natural texts implies that semantic analysis should following the stage of parsing. However, medical texts are know as very complicated and written in very specific language. Traditional parsers are demonstrating relatively small productivity here. In this article we demonstrating an opposite approach: ontology-based entailing of words in combination with simple shallow parsing rules. It allows us increasing UAS metrics from 0.82 for SpaCy to 0.834 for our approach.

Keywords: Parsing · Ontology · patients' complaints.

1 Introduction

There are a lot of theories which explain the human speech as reflection of processes in our mind. N. Chomsky wrote: "Universal grammar, then, constitutes an explanatory theory of a much deeper sort than particular grammar, although the particular grammar of a language can also be regarded as an explanatory theory." [1, p. 24]. Freely interpreting ideas of Chomsky, we can say that semantics should be first, and the syntax just shapes the ideas we want to express.

Speaking about grammars, we imply dependency structures which restoring syntactic (and shallow semantic) connections between head and tail words, as we did it at school time. If we will parse the phrase "The quick brown fox jumps over the lazy dog", we will find out the following facts: that is the fox which is quick and brown, that is the dog which is lazy, and that is the fox which carries out an action over the dog as an object of such action. Parsing of this sentence, we create a list of connections between words (fox is quick, fox is brown, fox

* Supported by RFBR grant 18-29-03131.

jumps, ...) and tag them by semantic relations (quick is a property of fox, fox is the subject of jumping, ...) using both our knowledge about the world (foxes have some specific colors, foxes and dogs are animals and animals have some temper, ...) and rules of grammar (subject precedes predicate, object follows predicate, ...).

The common approach to natural text processing conducts in the following way. At first stages one should tag a text, than parse it, and, finally, conduct semantic processing. Neural networks are trying to make these steps altogether or at least entail them as tough as they can. However, neural networks try to infer such information as verbal government, words co-occurrences, multi-word terms etc., which are statistical on their nature and can be just collected in dictionaries, but dictionaries of huge size.

The aim of this article is to demonstrate that if one have on ontology of a narrow domain, then he or she can construct the dependency structure of a sentence using just simple rules of shallow parsing. For our experiments, we used an ontology of patients' complaints applied to medical records written in the Russian language. We achieved the higher quality in comparison to modern parsers.

2 Review

There are several options for natural text parsing; the choice of an option depends on the purposes of an constructed system. In case of such systems as machine translation, one needs a powerful subsystem providing complete parsing and semantic labeling [19]. However, in some cases, one could use a shallow parsing system which aims to extract a list of connected clauses from a text. Such an approach names chunking [2] or shallow parsing depending on the output of the algorithm in hand. There is no need in reconstruction of dependencies structure for such tasks as terms or named entities recognition [3] and fact extraction [4] because a system just needs to define borders of a clause. Chunking increases the speed of a system by reducing the number of rules and extracted patterns and by using less complicated methods such as Hidden Markov Models [4], Conditional Random Fields [5], context-free grammars [3], and finite automatons [6].

There are two different approaches to creating such a systems: empirical and machine learning ones. Empirical approach is more robust and controlled one; however, it implies a lot of manual job which consists of writing many concrete rules for a variety of language phenomena. Machine learning methods presuppose that one has a corpus, and this corpus is big enough and allows training a model with a high precision. Modern neural networks can successfully solve all the stated tasks [7,8,9], but for some tasks their speed or precision is too low; another option is lack of tagged corpora mentioned above. Empirical approaches could increase the quality of results but in narrow domain or task only.

Ontologies are one of the solution for the task of semantic processing of texts. We will mention here some of alternatives used for processing of medical texts,

the domain which was selected for our experiments. The biggest medical ontology here is Unified Medical Language System (UMLS) [10] which consists of Metathesaurus (hierarchy of terms collected from many vocabularies), Semantic Network (relationships among these terms and their categories), and SPECIALIST Lexicon and Lexical Tools (a large syntactic lexicon of biomedical and general English combined with natural language processing tools). Metathesaurus vocabulary (Medical Subject Headings - MeSH) was translated into 15 languages including Russian [11]. MeSH was used in such projects as MetaMap - a program for information extraction from medical texts [12]. MetaMap algorithm consists of two stages: 1) processing of a medical text and fact extraction, and 2) notions refinement. The first stage starts with tokenization and finishes with a syntactic analysis. It includes an acronym/abbreviation identification, multi-word terms extraction, and their identification in dictionaries. The result of the medical text processing is a tagged text with links to Metathesaurus. Exactus system [13] used UMLS dictionary translated into Russian. The main purpose of this project is a logical inference for diagnosis of chronic diseases. Using of machine learning algorithms allows the authors to increase the precision of fact extraction up to 82% for a severity of disease and 99% for a flow of disease.

Note that both of the mentioned systems are using parsing of medical texts before entailing extracted terms to articles in their thesauri. Medical texts should be considered as texts written in a very specific language. Some parts of such texts do not have any verbs at all, but consist of long sequences of homogeneous parts: lists of complaints, diseases, pharmacy etc. That is why the real performance of parsers falls down to 80-85%. Ambiguity of terms decreases performance as well. A combined solution [14], which tries to correct the structure of dependency tree according to the information from an ontology, also demonstrated small performance.

Despite of its notorious quality, parsing of medical texts keeps its practical and theoretical importance. Creating a precise dependency structure of a medical text allows drawing a correct picture of disease and its flow. Such results could be used, for example, for quantitative research of historical data, inference of a diagnosis and in other tasks.

Note that most of the modern medical text processing systems are not so sensitive to the quality of parsing. For example, the paper [20] introduce a method for processing of medical records. However, this method is devoted to extraction of such facts as medication, diseases, indications, and some other and relations among them. The output of the method is list of such facts and their tags. Thus, the need of deep parsing in such system is very doubtful. The same is true for the system described in [21] which extracts named entities.

In our project, we investigate an approach which is opposite to the mentioned above. Our goal was creating of an algorithms which will infer the structure of a medical text mainly using information from an ontology with help of just a few simple shallow syntactical rules. We hope that this approach allows gather all the advantages and increase the productivity of the system.

3 Database of Terms and Observations

The main part of our system is the Database of Terms and Observations [15]. It is formed on the basis of an ontology with the same name. This ontology contains definitions of all concept classes and consists of two main types of medical terms descriptions symptoms and factors. Symptoms characterize the current functional state of a patient, and factors are used to describe the risks of various diseases. Symptoms and factors can be combined into logically related groups to make them easier to navigate. A symptom can be simple or composite. The first ones are described by name and a set of qualitative, numeric, or interval values. A composite symptom has a name and a set of characteristics. Each characteristic is also described by its name and a set of possible values (qualitative, numerical or interval). Thus, the Database of Terms and Observations have the form of a tree, where values are leafs which are subordinated to a name of a characteristic, a name of a characteristic subordinated to a name of feature, and a name of feature is subordinated to the Complaints node, e.g. Notalgia→Localization→Dorsal Spine. Each medical term could have several synonyms.

The "Symptom" section of this database contains several groups of symptoms: Complaints, Objective examination, Laboratory and instrumental examination. In this article, we use only the Complaint group of symptoms which describes the subjective feelings of the patient, characterizes his or her current functional status and the state of individual systems: digestive, respiratory, circulatory, nervous system, etc. This group contains a subgroup General complaints, which includes those that occur in many diseases (dizziness, weakness, nausea, sweating, etc.). Subgroup Pain is a part of the subgroup General complaints; it includes such symptoms as headache, back pain, neck pain, sore throat, etc. The most of composite symptoms in the Complaints group have such characteristics as localization, severity, cause", time of occurrence, intensity, frequency, etc. Characteristics of the Pain group also include an additional characteristics: irradiation, increasing, increasing, etc.

A fragment of the Database of Terms is presented at Fig. 1. Group of symptoms Pains includes a symptom Back Pain that has synonyms Spinal Pain and Lumbodynia. The symptom Back Pain has such characteristics as Localization (possible values are Lumbar Region and Lumbar Spine) and Amplification (possible values are Deep Breath and In a Strong Position).

The section of the Database in neurology domain was created according to 3000 anonymized medical records from the Department of Neurosurgery of Far Eastern Federal University. The information resources described above are stored in a heterogeneous repository developed by the authors [15]. The Database was created by a team consisting of both doctors in neurosurgical domain and ontology creation specialists.

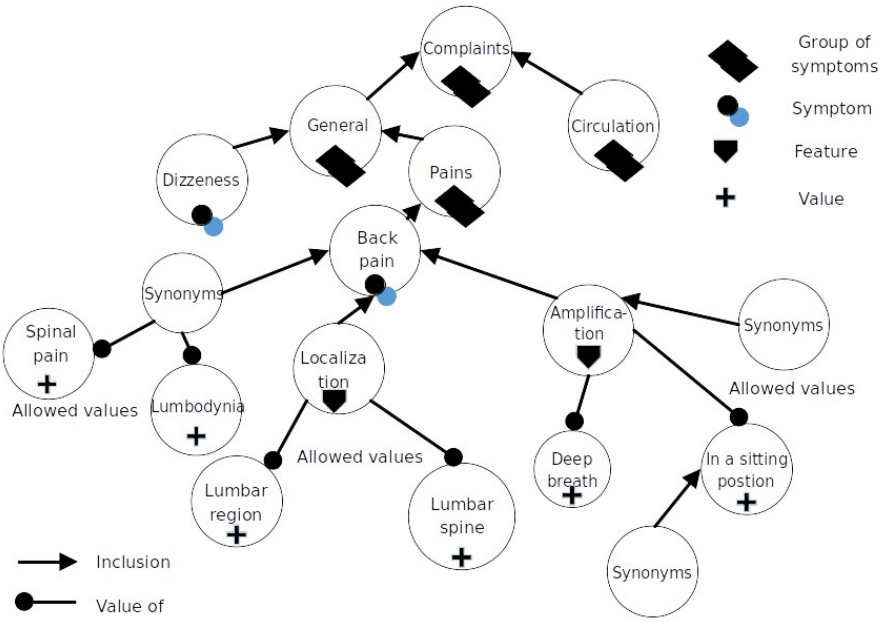

Fig. 1. A fragment of the Database of Terms and Observations

4 The Algorithm of Parsing of Patients' Complaints

As it was mentioned above, patients' complains are written in a very specific language. We used the following presuppositions for parsing such texts.

- A feature should be introduced before its characteristics; a characteristic should be introduced before its values: Complains to a notalgia localized in dorsal spine.
- In a text of a complaint, a value can be syntactically subordinated to a feature or characteristic; a characteristic can be subordinated to a feature: Complains to a notalgia ($\rightarrow$) localized in ($\rightarrow$) dorsal spine VS Complains to a notalgia in ($\rightarrow$) dorsal spine.
- A sequence of medical terms with the same first or last words (packed form of terms usage) can be joined using a conjunction or without it: dorsal and lumbar spine, or dorsal, lumbar spine, or dorsal aorta and artery.
- In a Russian text, relations between nouns should be governed by defined prepositions and grammatical cases, an adjective should be governed by a noun : a notalgia localized in dorsal spine$_{instr.case}$.

Two last presuppositions was implemented in a form of syntactical rules. Every rule can be presented as a tuple $R =< t_h, p_h, v_h, t_t, p_t, v_t, prep, case >$, where t_h is a token of a head term, p_h - a path in the Database of Terms and

Observations that defines a feature or a characteristic to which belongs the head term, v_h - type of head term (value, characteristics and feature), *prep* and *case* - a preposition (if any) and a grammatical case by which governs a tail token, t_t, p_t, and v_t - the same variables defined for a tail word.

Totally, we have written 44 such rules; 41 of them describing verbal government for the used terms. For example, a rule $<$ "","$.+/Pains$","","",$.+/Pains/.+/Localization/.+$","","$in$","$Case = Loc$" $>$ means that any term which belongs to pains can be connected with any term which belongs to localization of pains (both of them are placed on any level of the Database) using the preposition "in", tail word should be in prepositional case.

The overall algorithm of parsing of a patient's complaint is following.

1. Tag the text of a patient's complaint.
2. Find terms in the tagged text. Every multi-word term should be joined in one token and subscribed by grammatical parameters of its head word. Sequence of terms in packed form should be restored to a sequence of full terms.
3. Connect words according to the set of rules.
   (a) Create an empty list of candidate heads.
   (b) Iterating over words in the tagged text.
   (c) Starting from the tail of the list of candidate heads, find the first head with the "active" tag which can be applied to the current word. If such a rule was found, remove all the rules after it; create a new connection between head and tail word; a preposition, if any in the rule, connected to the tail word.
   (d) Put into the end of the list of candidate heads all the rules which can be successfully applied to the current word. If a rule needs a preposition, mark this rule by the "suspend" tag, otherwise mark it by the "active" tag.
   (e) In case of the current word is a preposition, mark all the candidate heads which are waiting for this preposition by the "active" tag; mark all the candidate heads which needs no preposition by the "suspend" tag until the next noun.
4. Iterate over all adjectives in the tagged text. Every adjective should be connected to a neighboring noun which coordinates it or to a noun in the right position if there is no coordination to any neighboring noun.
5. Create a dependency tree using the list of connections between words.

By removing rules in the item 3c, we try to keep the projectivity of the resulting tree. Connecting adjectives to nouns, we follow the statistics of such connections in medical records. We try to "reuse" a prepositions since we found out that doctors tends to write just one preposition followed by several terms. The same is true for a sequence of terms restored from packed form. Note that we ignore all the punctuation marks because of a variety of mistakes found in real texts. The same is true for terms; that is why we use fuzzy algorithms for comparison of words.

Let us consider an example of parsing taken from one of the patient's complaint.

Complaints on the significant levels of pain in lumbar and dorsal spine, increased pain on exertion, ambulation disorder.

At the first stage we tag the text and process terms, including multi-word ones. Let us denote the extracted terms by square braces, non-term tokens by curly braces. Note that English "significant levels of" corresponds to one Russian adjective. The result should be as following.

[Complaints] {on} {the} [significant levels of] [pain] {in} [lumbar spine] {and} [dorsal spine], [increased pain] {on} [exertion], [ambulation disorder].

After analyzing the first two words (Complaints on), we have one active rule in the list of candidate heads which corresponds to the connection between a feature "Complaints" to its characteristics. Than we pass over the text "the significant levels of". The word "pain" can be connected to "complaints on"; it also adds a set of rules to the list of candidate rules: "pain + in + localization in prepositional case" and "feature + characteristic". The former rule will be activated after the preposition "in". This rule will be applied on terms "lumbar spine" and "dorsal spine" and will create proper connections. The term "increased pain" creates connection between "pain" and "increased pain" and removes the rule "pain in". It also adds the rule "increased pain + on + value in prepositional case" which will be activated by "on" and applied to the next word "exertion". Finally the term "ambulation disorder" is connected to "complaints on" and removes proper rules from the list.

The result of our algorithm is the following list of connections:
[$complaints \rightarrow pain \rightarrow on$,
$pain \rightarrow lumbarspine \rightarrow in$,
$pain \rightarrow dorsalspine \rightarrow in$,
$pain \rightarrow increasedpain$,
$increasedpain \rightarrow exertion \rightarrow on$,
$complaints \rightarrow ambulationdisorder$]

On the next stage, we connect to the term "pain" the term corresponding to "significant levels of". After that we construct the resulting dependency tree.

As it was mentioned above, our aim was the comparison of such simple ontology-based approach to the traditional parsing of a natural language sentence. However, we make some steps in our algorithm which make impossible the direct comparison of results. That is why we created a procedure which aligns the resulting trees.

First of all, we created a "gold standard" containing manually parsed trees consisting of terms instead of sole words. We compared just these connections which was found in this "gold standard", instead of comparison of resulting trees. For trees provided by traditional parsing algorithms, we reduced terms into one node and restore terms written in packed form. Since we always add a preposition to a term, including situation when we process terms in packed form, we excluded prepositions from the resulting list of connections. These steps allows us comparing our results to results provided by other parsing algorithms.

5 Results of Experiments and Discussion

As it was mentioned above, we had manually tagged a "gold standard" consisting of 80 texts of patient's complaints and 1036 connections. All these patient's complaints was extracted from medical records written by different doctors and describing a flow of neurological diseases for different patients. All texts was collected at the same hospital. Since our task was correctly connect terms which are already tagged by our ontology, we evaluated our results using unlabeled attachment score (UAS, percentage of correctly connected pairs of words) instead of labeled attachment score (LAS, percentage of correctly connected and labeled pairs of words). We used for our comparison UDPipe version 2.5 (released in December 2019) [16] and SpaCy 3.0. (released in March 2021) [17]. Both of them built using LSTM neural networks. For training, UDPipe used Universal Dependencies corpora [18], while SpaCy trained on a bunch of different corpora and provides a lot of models for different purposes. In our experiments, we used "ru_core_news_sm" SpaCy model which provides more precise results.

The results of our experiments was following. UDPipe correctly parsed 11 out of 80 sentences and creates 780 correct connections out of 1036; SpaCy correctly parsed 23 sentences and created 850 correct connections; our algorithm correctly parsed 26 sentences and created 864 correct connections. Thus, the resulting UAS of the systems is 0.753 for UDPipe, 0.82 for SpaCy, and 0.834 for our approach.

Note that authors of SpaCy claims for 0.96 UAS on regular texts, while for patients' complaints their system demonstrated results as low as 0.83. The reason of such breakdown was already discussed above: that is lack of verbs, formal listing of claims and symptoms with sudden transitions among different groups of terms, introducing just one preposition for a sequence, some grammatical mistakes. However, such text have very strict inner logic: any subordinated term should be introduced after its main term only, convolution of several terms with a common prefix or postfix, etc. Note that some of drawbacks of such texts can be simply described in very short rules; that is why their analysis can be easily implemented, but only in case if you know what you should wait as an input. Thus, some of drawbacks can be considered as hints, but in case of parsing common texts such peculiarities could lead to over-specification and redundant complexity. That is why we can say that parsers compete on uneven played field - a common parser VS a specialized one in case of very special texts.

Note again that modern parsers, which are based on neural network, can be easily trained on existing syntactically tagged corpora. Creating such corpora takes long years; but once created and published, they can be re-used for a long. Creating an ontology takes much more time. Putting the same efforts, one is able to create an ontology for just a small domain. Creating a complete ontology takes decades. That is why ontology-based approach is much more expensive than a traditional one. However, if one already has an ontology which reflects relations in a domain, than it can be used for parsing as well.

If we will consider such syntactic structure as gerundial group, we will found that it reflects such relations as "owner-property" (hand of a human), "object of a process" (publishing of orders), and so on. That is why the opposite process

can help us describe syntactical relations between terms using semantic relations from an ontology; but currently, for a narrow domain only.

Thus, we can state that idea of using of semantics as a syntactic instrumentation could lead to success. However, this way is very expensive and still unproved on a wide domain or common knowledge texts.

6 Conclusion

In this paper, we use an old idea that semantics could play the first role in text parsing. Modern parsers are demonstrating pretty high accuracy, but their productivity falls down from 0.96 to 0.85 UAS in case of medical texts, more precisely - in case of patient's complaints. We introduce an ontology-based method of text parsing which uses connections in an ontology in order to restore connections in a text. However, the used ontology has a structure which differs from a traditional one, consisting synonyms, associations etc. The used ontology contains the same logical relations among terms one can see in a text: locations, reasons, etc. In order to take into account syntactical relations, we introduce a small set of simple rules. Most of these rules are describing verbal governance combined with semantic tags.

Our system demonstrates the same productivity as modern parsers, but its aim was demonstrating a new approach to parsing special text. Now we can state that such an approach can be successfully applied to short text parsing, in case, if one already has a big and complete ontology of a domain. However, our approach needs a new automatic method for extraction of verbal governance.

References

1. Chomsky, N.: Language and Mind. 3rd edn. Cambridge University Press, Cambridge (2006)
2. Abney, S.: Parsing by chunks. In: Berwick, R., Abney, S., Tenny, C., (eds.) Principle-based Parsing. Kluwer Academic Publishers (1991)
3. Bolshakova, E. I., Baeva, N. V., Bordachenkova, E. A., Vasilyeva, N. E., Morozov, S.S.: Lexicosyntactic Patterns for Automatic Text Processing. (In Russ.) In: Proc. of International Conference on Computational Linguistics and Intellectual Technologies "Dialog-2007", pp. 70–75 (2007)
4. Molina A., Pla F.: Shallow Parsing using Specialized HMMs. Journal of Machine Learning Research, 2, 595–613 (2002)
5. Sha, F., Pereira, F.: Shallow Parsing with Conditional Random Fields. In: Proc. of HLT-NAACL, pp. 134–141. Edmonton (2003)
6. Nozhov, I. M.: Implementation of an Automatic Syntactical Segmentation of a Russian Sentence. PhD Thesis. RSUH, Moscow (2003)
7. Anastasyev, D. G.: Exploring Pretrained Models for Joint Morpho-syntactic Parsing of Russian. In: Proc of International Conference on Computational Linguistics and Intellectual Technologies "Dialog-2020", pp. 1–12 (2020)
8. Korzun V. A.: R-BERT for Relationship Extraction on Russian Business Documents. In: Proc.of International Conference on Computational Linguistics and Intellectual Technologies "Dialog-2020", pp. 467–463 (2020)

9. Lyashevskaya, O. N., Shavrina, T. O., Trofimov, I. V., Vlasova, N. A.: GRAMEVAL 2020 Shared Task: Russian Full Morphology and Universal Dependencies Parsing. In: Proc of International Conference on Computational Linguistics and Intellectual Technologies "Dialog-2020", pp. 553–569 (2020)
10. Current Bibliographies in Medicine, https://www.nlm.nih.gov/archive/20040831/pubs/cbm/umlscbm.html. Last accessed 20 Apr 2021
11. MSHRUS (MeSH Russian) - Statistics, https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/MSHRUS/stats.html. Last accessed 20 Apr 2021
12. Aronson, A. R., Lang, F. M.: An overview of MetaMap: historical perspective and recent advances. Journal of the American Medical Informatics Association 17(3), 229–236 (2010)
13. Shelmanov, A.O., Smirnov, I.V., Vishneva, E.A.: Information Extraction from Clinical Texts in Russian. In: Proc of International Conference on Computational Linguistics and Intellectual Technologies "Dialog-2015", pp. 560–572 (2015)
14. Klyshinsky, E.,Gribova, V., Shakhgeldyan, C., et al.: Formalization of Medical Records Using an Ontology: Patient Complaints. In: Proc. of Analysis of Images, Social Networks and Texts "AIST-2019" (2019)
15. Gribova, V.V., Moskalenko, Ph. M., Shahgeldyan, C.I., Gmar', D.V., Geltser, B.I.: A Concept for a Heterogeneous Biomedical Information Warehouse (In Russ.). In: Information Technologies, 2, (25), 97–106 (2019)
16. Straka, M., Strakovб, J., Haji, J.: UDPipe at SIGMORPHON 2019: Contextualized Embeddings, Regularization with Morphological Categories, Corpora Merging. In: Proc. of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pp. 95–103 (2019)
17. Whats New in v3.0 https://spacy.io/usage/v3. Last accessed 20 Apr 2021
18. Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Haji, J., Manning, C., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D.: Universal Dependencies v1: A multilingualtreebank collection. In Proc. of the 10th International Conference on Language Resources andEvaluation (LREC 2016), pp. 1659–1666 (2016)
19. Apresjan, Ju., Boguslavskij, I., Iomdin, L., Lazurskij, A., Sannikov, V., Sizov, V., Tsinman, L.: ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT. In Proc. of the First International Conference on Meaning-Text Theory, Paris, Йcole Normale Supйrieure, pp. 279–288 (2003)
20. Jagannatha, A., Yu., H.: Bidirectional RNN for Medical Event Detection in Electronic Health Records. In Proc. of Association for Computational Linguistics. North American Chapter, pp. 473-482 (2016)
21. Miftahutdinov, Z., Alimova, I., Tutubalina, E.: On biomedical named entity recognition: Experiments in interlingual transfer for clinical and social media texts. In: Lecture Notes in Computer Science, (12036), 281–288 (2020)