UDC 004.822

KNOWLEDGE BASES AND DESCRIPTION LOGICS APPLICATIONS TO NATURAL LANGUAGE TEXTS ANALYSIS

H. I. Hoherchak

Т*aras Shevchenko National University of Kyiv*

The article describes some ways of knowledge bases application to natural language texts analysis and solving some of their processing tasks. The basic problems of natural language processing are considered, which are the basis for their semantic analysis: problems of tokenization, parts of speech tagging, dependency parsing, correference resolution. The basic concepts of knowledge bases theory are presented and the approach to their filling based on Universal Dependencies framework and the correference resolution problem is proposed. Examples of applications for knowledge bases filled with natural language texts in practical problems are given, including checking constructed syntactic and semantic models for consistency and question answering.

Keywords: knowledge bases, natural language processing, syntax dependencies, coreference resolution, semantic analysis.

Introduction

The history of the field of natural language processing generally originates in the 1950s. At first, scientists faced the task of machine translation: for the United States government, it was important to have a system that would allow the translation of Russian-language texts into English with high accuracy. Already in 1954, as part of the Georgetown experiment, a primitive machine translation system was first demonstrated.

The modern natural language processing has more than three dozens of tasks, including the tasks of part-of-speech tagging, tokenizing, dependency parsing, syntax tree construction, coreference resolution, lexical normalization, named entities recognition, cloze test solving, natural language inference, relation extraction, speech recognition, machine translation, sentiment analysis, spelling check, etc. The current state of the vast majority of these and similar tasks is described on the NLP-progress[1] portal and on the Natural Language Processing page of the Papers With Code[2] portal, which provides in particular the rankings of models for solving each of the tasks with links to scientific articles that these models describe. Also, for a large number of models the latter provides links to the source codes of the relevant machine learning models.

Among these tasks, the task of open information extraction should be outlined. The purpose of it is to present natural text in a structured form: usually in the form of binary relations or relations of larger dimensions. A qualitative solution to this problem would make it possible to talk about the presence of automated methods of filling the knowledge base with natural language data, which exactly operate with atomic concepts and roles – the relationship between them.

At the time of writing, this task does not have clearly formulated and generally accepted standards of the result. As a consequence, it does not specify what attitudes should be obtained and how they should be formatted. There is also no standard for evaluating models and no corpora of acceptable size for high-quality training of ML models, as is customary for many of the above tasks of the NLP area.

The first steps towards the specification and evaluation of the results of this problem were made in [1], which offers a comparison of OpenIE models based on the precision-recall curve and AUC metrics (area under the curve). Most of the new models for this task [2] utilize its methodic for results evaluation, however several new works [3,4] provide evaluation based on more natural F1 metric.

In general, the models for open information extraction are divided into two subtypes [5]:

- machine learning systems (e.g. Neural Open Information Extraction and OpenIE-5.0);
- rule-based system (e.g. Graphene [2]).

It is worth to mention, that the quality of modern models for solving this problem (even measured using existing F1 and AUC metrics) does not allow us to consider the knowledge bases construction based on natural text information as a qualitatively solved problem at this stage, in particular, given the variety of problem formulations and metrics for results comparison.

Thus, a promising direction of research is the extraction of open (arbitrary) relations from natural texts, in particular, the formalization of the OpenIE problem with respect to its application in knowledge base filling, construction of a metrics apparatus to compare problem solving models in a given formal system and the actual solution of the problem.

Building a knowledge base on text makes it possible to analyze the properties of text using algorithms and methods of knowledge bases, utilizing descriptive logics techniques. In particular, tableau-algorithm for proving

[1] http://nlpprogress.com/

[2] https://paperswithcode.com/area/natural-language-processing/

concept executability within a natural language knowledge base allows you to check the compatibility of the built syntactic and semantic models of the human-readable information. Thus, by operating over some additional knowledge about the subject area, it is possible to identify contradictory and therefore erroneous elements to eventually correct them. The partially solvable problem of executing knowledge base queries, in the case of converting a text question into the appropriate expression of the query language, is also a useful tool for solving the problem of question answering.

Overview of natural language processing tasks

Potentially useful input for the task of open information extraction can be gained from results for parts of speech, named entities, grammatical dependencies and coreferences analysis.

Tokenization in the field of natural language processing is aimed at processing a sequence of symbols (text) and identify individual words or sentences in it. The division of such sequence into words in the first approximation can be done by dividing the input stream of characters into parts by delimiters (eg, spaces, punctuation). Full tokenization should also take into account the features of certain languages, where punctuation marks can be part of complex lexical constructions (for example, in English the sequence of characters *i.e.* corresponds to the phrase *in other words*, and the construction *let's* denote two words: let and us) or abbreviations. Similarly, the sentence division problem cannot be simply reduced to the problem of splitting the text by delimiters, because punctuation marks, as mentioned above, can be part of words and complex speech constructions.

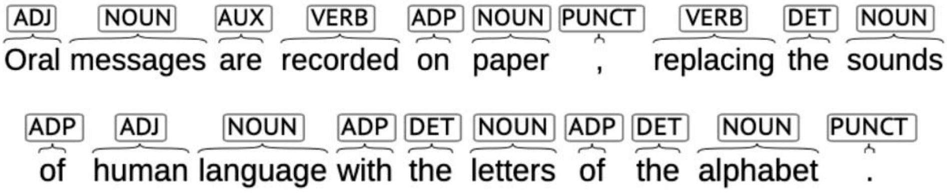
As mentioned above, tokenization algorithms take into account the characteristics of the language, text of which is submitted to the input. Thus, tokenization algorithms are usually built for each language or group of similar languages □□separately. For instance, division of English texts into words and sentences can be done using Stanford Tokenizer, proposed in [6].

Part-of-speech tagging. The task of defining parts of speech (POS tagging) aims to mark each word in the text with the part of speech, which the word belongs to. Modern research in the field of natural language processing mostly use the morphological designations defined in Universal Dependencies [7] - a framework for a single system of grammar annotation of different natural languages. This framework allows you to work with the morphological and grammatical structure of the sentence, abstracting from a particular language and operating only with the appropriate universal symbols.

Consider the following sentence:

Oral messages are recorded on paper, replacing the sounds of human language with the letters of the alphabet..

The corresponding result of the morphological analysis in the format of Universal Dependencies is presented in Fig. 1.



ADJ NOUN AUX VERB ADP NOUN PUNCT VERB DET NOUN
Oral messages are recorded on paper , replacing the sounds
ADP ADJ NOUN ADP DET NOUN ADP DET NOUN PUNCT
of human language with the letters of the alphabet .

Fig. 1. The result of POS tagging

Here *ADJ* denoted adjectives, *NOUN*– nouns, *VERB* – verbs, *ADP* – prepositions, *PUNCT* – punctuation marks.

Modern models for determining parts of speech are mostly based on the machine learning approach. That said, to solve this problem for the English language, a standard set of data - part of Penn Treebank, associated with the Wall Street Journal, which contains 45 different POS-tags, - is used for training. At the time of writing, the best accuracy at 97.96% is reached by the Meta BiLSTM model proposed in [8]. This model is based on two recurrent neural networks with a sentence-level context. Its results are combined using a meta-model, so that the output is a unified representation of each word, which is then used for notation.

Solving a similar problem for multiple languages □□at the same time, using tags from the Universal Dependencies framework and corresponding corpora for different languages, is a more difficult task. Currently, several models, including Uppsala and HIT-SCIR, show the best results for a large number of languages □□(the average F1 score for all languages □□for both models exceeds 0.9 for this task). In particular, the HIT-SCIR and Stanford models give an F1 score above 0.97 for English, Ukrainian and Russian.

Dependency parsing is aimed to identify the dependencies that represent the grammatical structure of a sentence and determine the connections between the "main" words and the words that modify them.

General principles for denoting syntactic dependencies are also presented in the Universal Dependencies framework, which defines more than 30 different types of dependencies and some extensions for them, depending on the group of languages □□under consideration. In the basic version of these dependencies, the syntactic structure of the sentence is presented in the form of a tree, i.e. each word of the sentence (except the main one - the root) has exactly one ancestor. Each branch of the tree is marked with a special label that categorizes the relationship between the ancestor word and the descendant word according to one of 36 different types of dependencies.

An example of the result of such parsing for the above mentioned sentence is given in Fig. 2.

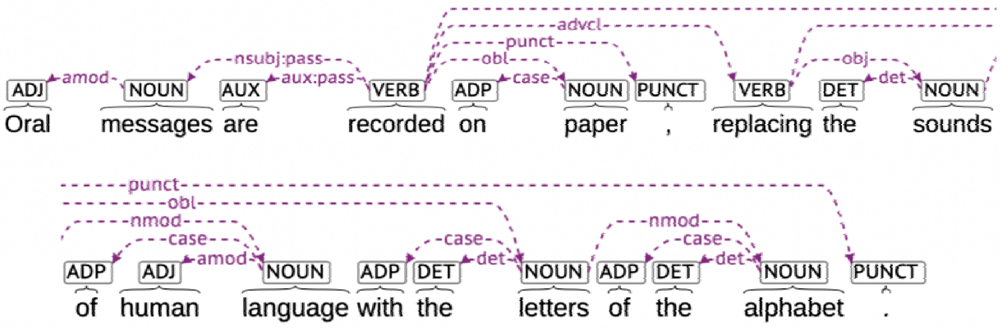

ADJ NOUN AUX VERB ADP NOUN PUNCT VERB DET NOUN
Oral messages are recorded on paper , replacing the sounds
amod
nsubj:pass
aux:pass
advcl
punct
obl
case
obj
det
ADP ADJ NOUN ADP DET NOUN ADP DET NOUN PUNCT
of human language with the letters of the alphabet .
punct
obl
nmod
case
amod
case
det
nmod
case
det

Fig. 2. Dependency parsing results

Here the *amod* indicates the adjective, *obj* – the object of action, *obl* – circumstance, *advcl* – adverb, *case – auxiliary word, nmod* – noun, *punct* – punctuation marks.

Models for solving this problem for English are mostly compared on the basis of the Penn Treebank dataset with the provided markings of parts of speech. For their comparison, the following metrics are used:

- UAS (unlabeled attachment score), which does not take into account the dependency labels, but compares only the correctness of the ancestor attachment for each word in the sentence;
- LAS (labeled attachment score), which denotes the proportion of correctly parsed words (correctly marked with both ancestor and dependency label).

At the time of writing, the best values of the above metrics are demonstrated by the Label Attention Layer + HPSG + XLNet model, proposed in November 2019 in [9]. This model is also based on the neural network approach and reaches UAS 97.33% and LAS 96.29%. However, recent scientific conferences have focused on building unified parsing models for a large number of languages. Thus, the HIT-SCIR model allows to achieve LAS of 0.92, 0.88 and 0.87 for Russian, Ukrainian and English, respectively.

Consider a more complex sentence: *Cats usually catch and eat mice and rats*.

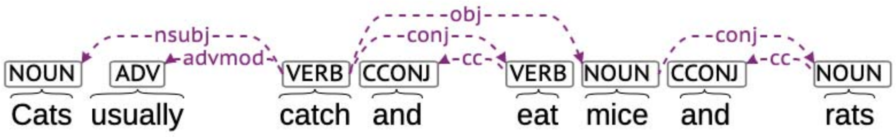

NOUN ADV VERB CCONJ VERB NOUN CCONJ NOUN
Cats usually catch and eat mice and rats
nsubj
advmod
obj
conj
cc
conj
cc

Fig. 3. Dependency parsing results

From the dependency tree above, one may notice that the basic set of dependencies does not allow sufficient analysis of syntax links and extraction of its semantics. Thus, the conjunctive objects of the action *mice* and *rats* here are connected by the *conj* bond, meaning that the word *rats* is found to be related to the action only indirectly, although semantically it is also its object.

This and other problems are solved by expanding the dependency tree with additional arcs (Fig. 4) - of course, with the loss of the tree structure. Transforming the base dependency tree into an extended dependency graph requires, in particular, solving the following problems:

- recovery of missed words by creating fictitious tokens;
- propagation of conjuctions (objects, subjects, definitions) through conjunction;
- distribution of subjects to subordinate verbs of a complex predicate;
- processing a subordinate clause that specifies an object as an action performed by that object (may result in loops);
- adding case information to the dependency name.

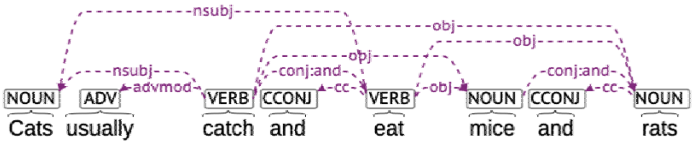

NOUN ADV VERB CCONJ VERB NOUN CCONJ NOUN
Cats usually catch and eat mice and rats
nsubj
nsubj
advmod
obj
obj
obj
obj
conj:and
cc
conj:and
cc

Fig. 4. Enhanced graph of syntax dependencies

The CoreNLP[3] natural language processing package allows you to achieve a value of 0.92 for the LAS for English. The presence of a corpus for Ukrainian allows us to talk about the potential possibility of solving this problem also for this language, but, at the time of writing, no such models were found in the free access.

Along with universal dependencies, there are also a number of language-specific dependency formats. An example of an alternative format for describing the syntactic structure of a sentence in the Ukrainian language, proposed in [10], is given in comparison with the universal dependencies in Figs. 5, 6.

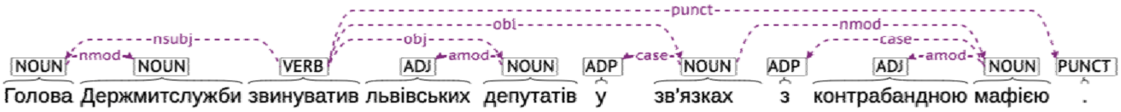

NOUN
NOUN
VERB
ADJ
NOUN
ADP
NOUN
ADP
ADJ
NOUN
PUNCT
nmod
nsubj
obj
obl
amod
case
punct
nmod
case
amod
Голова Держмитслужби звинуватив львівських депутатів у зв'язках з контрабандною мафією .

Fig. 5. An example of a universal tree of dependencies for a Ukrainian-language sentence

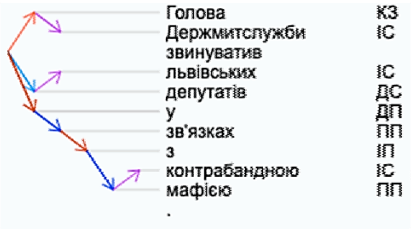


Fig. 6. An example of an alternative dependency tree for a Ukrainian-language sentence

Here the symbol *К3* denotes the subject-predicate compound, *ІС* is the noun prepositional compound, *ДС* is the verb prepositional compound, *ДП* is the verb prepositional compound, *ПП* is the prepositional compound, *ІП* is the noun prepositional compound.

Combining results of different parsing formats allows to achieve a better aggregate result and correct errors that occur in each of the resulting trees.

Coreference resolution aims to cluster references in the text that relate to the same entity of the real world.

Consider the following sentence: *"I voted for Barack Obama, because his beliefs are closest to my own values," she said.*

Analysis of its syntactic dependencies (Fig. 7) does not allow to fully determine which objects of the real world: the same or different, – refer to the pronouns given in the sentence. A similar problem, but extended to references throughout the text, not just within a sentence, cannot be solved at all with either basic or extended dependencies, because they represent connections within a single sentence.

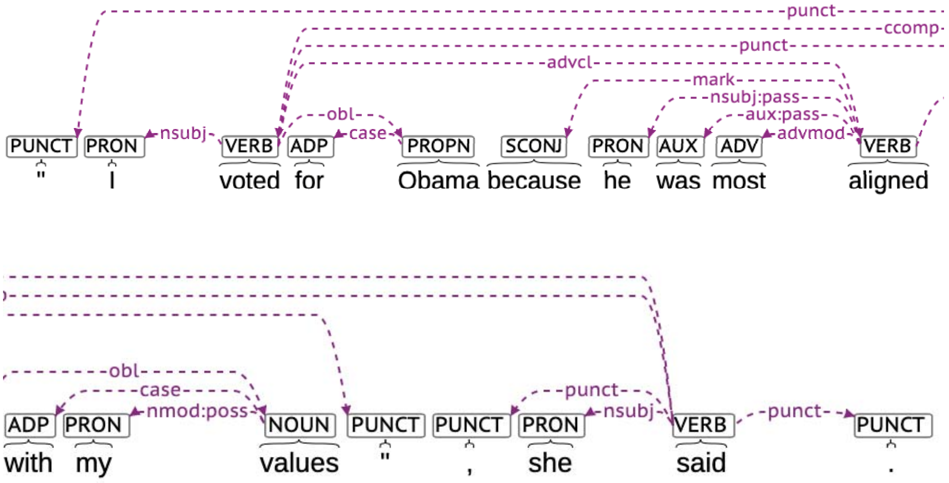

punct
ccomp
punct
advcl
mark
nsubj:pass
aux:pass
advmod
nsubj
obl
case
PUNCT PRON VERB ADP PROPN SCONJ PRON AUX ADV VERB
" I voted for Obama because he was most aligned
obl
case
nmod:poss
punct
nsubj
punct
ADP PRON NOUN PUNCT PUNCT PRON VERB PUNCT
with my values " , she said .

Fig. 7. Sentence dependency tree with many correlated words and phrases

The above mentioned induced a separate problem of processing natural languages, the solution of which would allow to collect equivalent entities in a certain text and analyze all relations for them more rigorously.

[3] https://nlp.stanford.edu/pubs/schuster2016enhanced.pdf

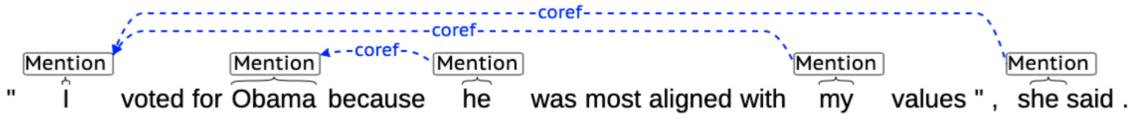

coref
coref
coref
Mention
Mention
Mention
Mention
Mention
" I voted for Obama because he was most aligned with my values " , she said .

Fig. 8. Coreference forest

The set of corefering words and phrases is usually presented in the form of a forest (Fig. 8) - a set of trees, each denoting a set of coreferent nodes. The coreferential arc is usually directed to the most specific notation of a real-world object.

The comparison of coreference resolution models is currently carried out on the OntoNotes[4] corpus, which contains English-language texts of various genres (news, spoken telephone, blogs, talk shows, etc.) with annotated coreferentialities.

Currently, the best performance for this task is shown by modifications of the BERT model [11] based on the machine learning approach developed by the Google AI Language team. BERT proposes a common model for presenting natural language information for a set of word processing tasks and introduces bidireactional context of the word, as opposed to the use of left-handed or right-handed contexts in previous effective models.

Basic concepts of the knowledge base theory

The basic principles of ontology systems and knowledge bases of natural texts are presented in [12, 13, 14, 15, 16]. We will introduce some concepts of the knowledge bases theory that will be used further.

Concepts are considered to be the useful tool for recording knowledge about the subject area to which they relate. This knowledge is divided into general knowledge about concepts and their relationships and knowledge about individual objects, their properties and relationships with other objects. According to this division, knowledge recorded using the descriptive logic language is divided into a set TBox of terminal axioms and a set ABox of facts about individuals.

Definition 1. The terminology axiom is an expression C ф D *(inclusion of concept* C *in concept* D*) or* $C \equiv D$ *(equivalence of concepts* C *and* D*), where* C *and* D *are arbitrary concepts.*

Definition 2. The terminology (TBox) is an arbitrary finite set of terminological axioms.

Definition 3. Axiom C ф D $(C \equiv D)$ *is true in the interpretation* I *if* $C^I \subseteq D^I$ $(C^I = D^I)$*. In this case* I *is called the model of this axiom, noted as* I ‘ C ф D*. Interpretation* I *is called a terminology model* (I ‘ T)*if it is a model for all axioms of* T *.*

Definition 4. Terminology is called compatible or executable if it has non-empty model. Concept C *is executable with regards to* T *if there exists a model* I *of terminology* T*, such as* $C^I \neq \varnothing$ *.*

Terminology makes it possible to write down general knowledge about concepts and roles. But it often needs to record knowledge about specific individuals: which class an individual belongs to, what relationships (roles) there are between induviduals etc.

Definition 5. Factual system (ABox) is a finite set A *of facts like* $a:C$ *or* aRb*, where* a,b *are individuals,* C *– an arbitrary concept,* R *– role.*

Dependency-based approach to the knowledge base filling

Some approaches to the natural language analysis for the knowledge extraction were presented earlier in [17, 18]. The following is a conceptually different approach to filling the knowledge base using universal dependencies and coreferences.

The tree (or graph) of syntactic dependencies considered above is a powerful source for extracting knowledge from it in the form of open relations. Consider the following text:

'La La Land' is the third film by young director Damien Chazelle. His previous work, 'Whiplash', won many prestigious film awards, including three Academy Awards. This year nominees are already known, and the 'La La Land' is the undisputed leader: has 14 nominations. This picture has already won all the most prestigious nominations of Golden Globe Awards.

The main sources of relations, that is, the facts of a kind aRb, are the verbs along with the words, connected by *nsubj* (nominal subject) and *obj* (object) dependencies. Consider the second sentence of the above text, the dependency tree for which is given in Fig. 9.

[4] https://catalog.ldc.upenn.edu/LDC2013T19

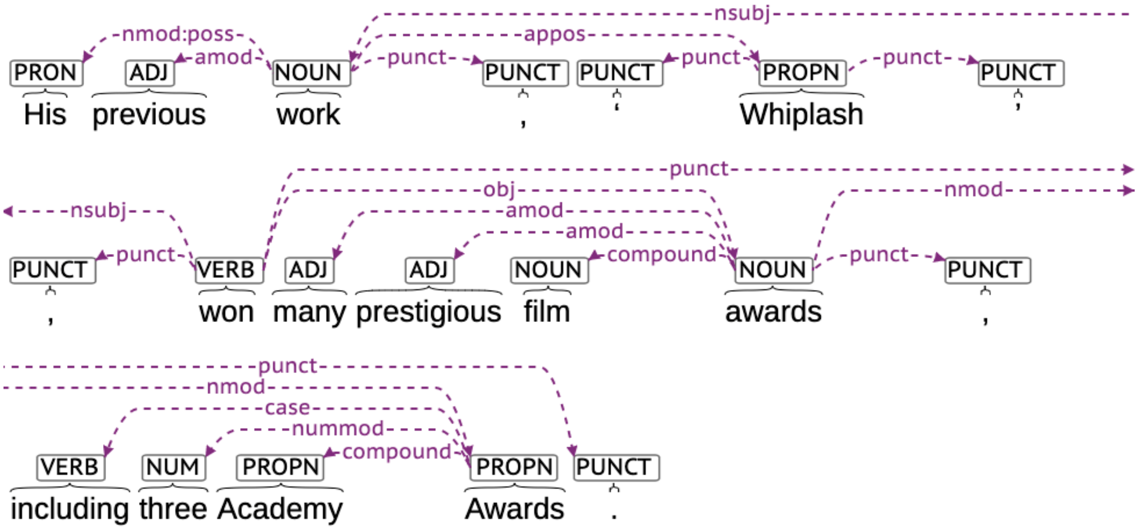


Fig. 9. Dependency tree

From this dependency tree, we can extract a relation triple *(work; win; film award)*. Obviously, such a relation itself does not carry enough substantive load – all that because the individuals involved here need additional specification.

Using *flat* and *amod* dependencies, let's construct the following chain of concepts for the subject of this action:

work_previous ф *work,*

work_Whiplash ф *work_previous*

The fact that individual a_1 belongs to the generated concepts can be written as: a_1 :*work_Whiplash*. We shall note that the affiliation of this individual to other concepts follows from the essence of the concept inclusion relation.

In the same way for the object of the action we can write the following terminological axiom:

film_award_prestigious ф *film_award*

On this stage we shall notice that here we will pay reader's attention only to population of a TBox. ABox facts can be constructed in accordance to the TBox terminology considered below by deterministic algorithm of semantic table, which will be considered below.

Since the object is in plural, the following concepts should be included:

work_Whiplash ф $\geq 2R_{win}$.*film_award_prestigious*

Another R_{win} role subject is hidden in the basic dependency tree behind *conj* dependency. After similar operations, we can obtain the following knowledge base:

TBox = { work_previous ф *work, work_Whiplash* ф *work_previous, film_award_prestigious* ф *film_award,*
Academy_Award ф *award, work_Whiplash* ф $\geq 2R_{win}$.*film_award_prestigious,*
work_Whiplash ф $= 3R_{win}$. *Academy_Award }*

ABox = { a_1 *: work_Whiplash,* a_2 *: film_award_prestigious,* a_3^1 *: Academy_Award,* a_3^2 *: Academy_Award,*
a_3^3 *: Academy_Award,* $a_1R_{win}a_2$, $a_1R_{win}a_3^1$, $a_1R_{win}a_3^2$, $a_1R_{win}a_3^3$ }

Except *flat*, *nmod*, and *amod* dependencies, new terminological axioms can also be formed from *obj* dependencies in the case of an elided predicate. Yes, in the first sentence of the text above (Fig. 10) the verb *is* is considered as a copula.

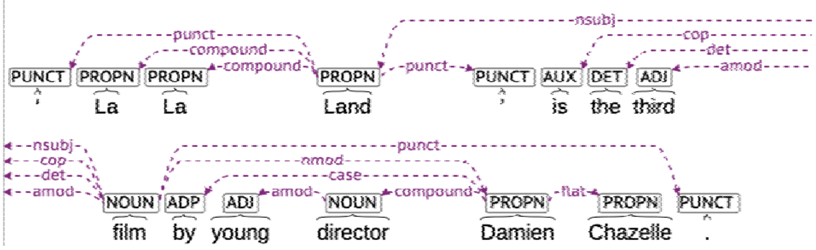


Fig. 10. Dependency tree

Thus, since the dependency tree root is a noun, dependencies *obj* and *subj* here semantically denote the inclusion of concepts. Similarly, we produce concepts and terminological axioms for object and the root of this tree as follows:

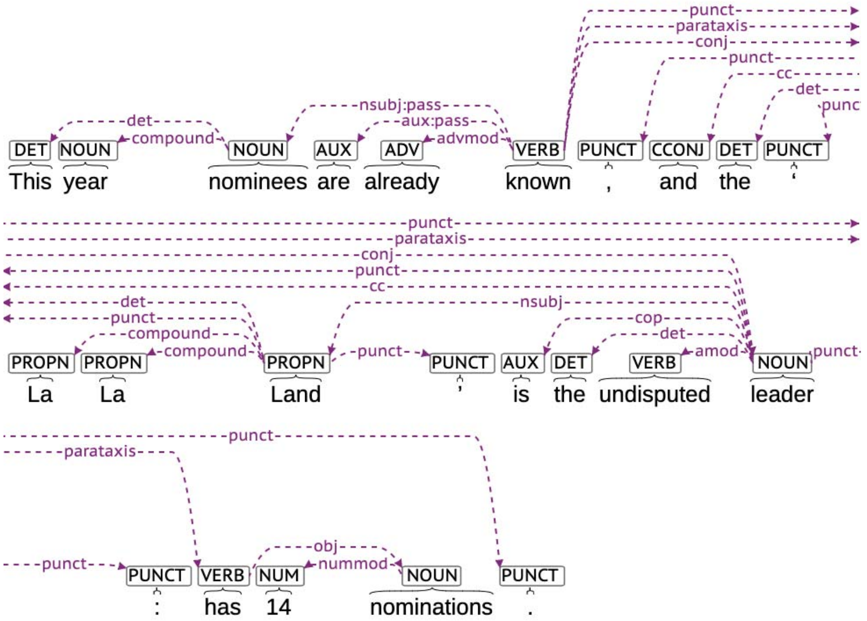
La_La_Land ф *film, director_film* ф *film*

young_director_film ф *director_film,*

Damien_Chazelle_film ф *young_director_film,* *Damien_Chazelle_film_third* ф *Damien_Chazelle_film*

Additionally, the following will be added to the list of terminological axioms for the above sentence:

La_La_Land ф *young_director_Damien_Chazelle_film_third*



DET NOUN NOUN AUX ADV VERB PUNCT CCONJ DET PUNCT
This year nominees are already known , and the ‘
det compound nsubj:pass aux:pass advmod punct parataxis conj punct cc det punc
PROPN PROPN PROPN PUNCT AUX DET VERB NOUN
La La Land ’ is the undisputed leader
punct parataxis conj punct cc det punct compound compound nsubj cop det amod punct
PUNCT VERB NUM NOUN PUNCT
: has 14 nominations .
punct parataxis punct obj nummod punct

Fig. 11. Dependency tree

In cases when the root of a sentence is an adjective or a past participle, it also means an inclusion of concepts. Yes, the first part of the third sentence (Fig. 11) produces the following terminological axioms:

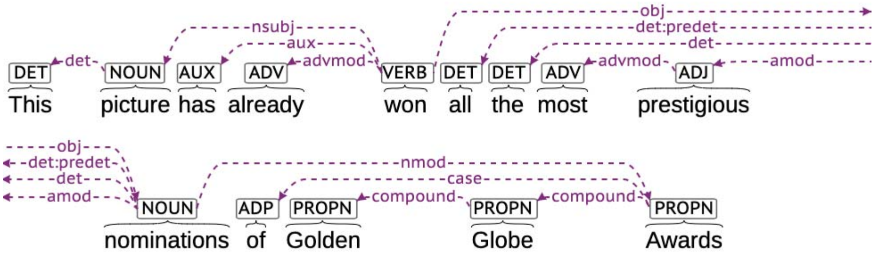
this_year_nominee ф *nominee, this_year_nominee* ф *known*

In cases when the root of a sentence has *conj* dependents associated with it, a each such dependent is considered and processed as the root of its subtree. Since the root of the subtree is a noun, like the case considered earlier, the knowledge base is supplemented by the following terminological axioms:

undisputed_leader ф *leader, La_La_Land* ф *undisputed_leader*

Parataxis-dependent subtrees should be treated in the same way. Thus, consider the expression *has 14 nominations*, which adds to the knowledge base the only concept *nomination*. Since the predicate has no subject, here comes a problem of defining it within context. In this case, the definition is simple enough: we use the subject of the ancestor, that is, the phrase *La La Land*. Thus, the following facts should be added to the knowledge base:

$$a_4 : La\_La\_Land,\ a_5 : nomination,\ a_4 R_{has} a_5$$



DET NOUN AUX ADV VERB DET DET ADV ADJ
This picture has already won all the most prestigious
det nsubj aux advmod obj det:predet det advmod amod
NOUN ADP PROPN PROPN PROPN
nominations of Golden Globe Awards
obj det:predet det amod nmod case compound compound

Fig. 12. Dependency tree

Last sentence (Fig. 12) will produce the following terminological axioms:

Golden_Globe_Awards_nomination ф *nomination,*

Golden_Globe_Awards_nomination_prestigious ф *Golden_Globe_Awards_nomination,*

Golden_Globe_Awards_nomination_most_prestigious ф *Golden_Globe_Awards_nomination_prestigious*

Considering coreference resolution task as solved, the subject *This picture* corresponds to the concept *La_La_Land*. Thus, the knowledge base is also supplemented by the following facts:

$$a_6 : Golden\_Globe\_Awards\_nomination\_most\_prestigious,\ a_4 R_{win\_in} a_6$$

The final knowledge base for the given text fragment will look like this:

CN = { *picture_previous, picture, picture_obsession,* film *award_prestigious, film award, academy_award, award, movie_La La Land, film, director_job, work_director_young, work_director_Damien_Chazell, work_director_Damien_Chazell_third, film_La_Land, this year_nominee, nominee, famous, film_La La Land, film, leader_unquestioned, leader, film_La La Land* }

$$\text{RN} = \{\ R_{win},\ R_{win\_in},\ R_{has}\ \} \qquad \text{IN} = \{\ a_1, a_2, a_3^1, a_3^2, a_3^3, a_4, a_5\ \}$$

TBox = { *work_previous* ф *work, work_Whiplash* ф *work_previous, film_award_prestigious* ф *film_award,*
Academy_Award ф *award, work_Whiplash* ф $\geq 2R_{win}$ *.film_award_prestigious,*
work_Whiplash ф $= 3R_{win}$ *. Academy_Award, _La_Land* ф *film, director_film* ф *film*
young_director_film ф *director_film, Damien_Chazelle_film* ф *young_director_film,*
Damien_Chazelle_film_third ф *Damien_Chazelle_film, La_La_Land* ф
young_director_Damien_Chazelle_film_third, this_year_nominee ф *nominee, this_year_nominee* ф *known,*
undisputed_leader ф *leader, La_La_Land* ф *undisputed_leader,*
Golden_Globe_Awards_nomination ф *nomination,*
Golden_Globe_Awards_nomination_prestigious ф *Golden_Globe_Awards_nomination,*
Golden_Globe_Awards_nomination_most_prestigious ф *Golden_Globe_Awards_nomination_prestigious}*

ABox = { a_1 *: work_Whiplash,* a_2 *: film_award_prestigious,* a_3^1 *: Academy_Award,* a_3^2 *: Academy_Award,*
a_3^3 *: Academy_Award,* a_4 *: La_La_Land,* a_5 *: nomination,*
a_6 *: Golden_Globe_Awards_nomination_most_prestigious,*
$a_1 R_{win} a_2$, $a_1 R_{win} a_3^1$, $a_1 R_{win} a_3^2$, $a_1 R_{win} a_3^3$, $a_4 R_{has} a_5$, $a_4 R_{win\_in} a_6$ }

Usage of WordNet's lexical database during semantic analysis one can also add the following auxiliary terminological axioms:

{*film* ф *work, film_award* ф *award*}

Search for content contradictions using the tableau-algorithm

Definition 6. The algorithm U *solves the problem of conceptuality in terminology* T *for descriptive logic* L *if the following conditions are met:*

1. *Finiteness: For arbitrary* (C,T) *algorithm* U *generates a response* $U(C,T)$ *in finite time.*
2. *Correctness: for arbitrary* (C,T)*, if* C *is executable in terminology* T *, then* $U(C,T)=1$.
3. *Completeness: for arbitrary* (C,T)*, if* $U(C,T)=1$*, then the concept* C *is executable in terminology* T .

The tableau-algorithm for checking the concept executability is defined by the rules in Table 1.

Table 1: Rules of the tableau-algorithm for logic ALC +TBox

Rule	Terms of application	Action
y -rule	point x – active; $x:(C\text{ y } D) \in \mathrm{A}$	$\mathrm{A}' = \mathrm{A} \cup \{x:C, x:D\}$
т -rule	point x – active; $x:(C\text{ т } D) \in \mathrm{A}$	$\mathrm{A}' = \mathrm{A} \cup \{x:C\}, \mathrm{A}'' = \mathrm{A} \cup \{x:D\}$
$\exists$ -rule	point x – active; $x:\exists R.C \in \mathrm{A}$ yy$:\{xRy, y:C\} \subseteq \mathrm{A}$	y – descendant of x ; $\mathrm{A}' = \mathrm{A} \cup \{xRy, y:C\}$
$\forall$ -rule	point x – active; $x:\forall R.C \in \mathrm{A}$ $\exists y : xRy \in \mathrm{A} \wedge y:C \notin \mathrm{A}$	$\mathrm{A}' = \mathrm{A} \cup \{y:C\}$
T -rule	point x – active; $x:E \notin \mathrm{A}$ $E \sqsubseteq \bullet \in T$	$\mathrm{A}' = \mathrm{A} \cup \{x:E\}$

From the initial Abox A_0 by applying these rules the search tree is being constructed with A_0 as root. Each ABox has 0, 1 or 2 descendants. The application of the rules is terminated if none of the rules can be applied to the next Abox A , or if there is a clear contradiction in A (i.e. for some individual x and concept C, $\{x:C, x:\neg\mathrm{C}\} \subseteq \mathrm{A}$ or $\{x:\perp\} \subseteq \mathrm{A}$).

To fulfill the condition of terminality of the algorithm, the concept of an active point is introduced so that $\exists$-rule and T -rule together do not lead to an infinite generation of individuals with the same set of concepts to which they belong.

Definition 7. The point x *blocks the point* y *if* x *is ancestor of* y *and* $L(x) \supseteq L(y)$*, where* $L(x) = \{C | x:C \in \mathrm{A}\}$ *. The point y is called to be blocked if it is blocked by any point x. An active point is one that is neither a blocked point nor a descendant of some blocked point.*

The tableau-algorithm for the above text does not lead to contradictions, which means text is consistent.

On the other hand, if we supply the knowledge base with the fact *work_Whiplash* y R_{win}*. Academy_Award* $\subseteq \perp$, which means that movie “Whiplash” did not win any Oscars, the algorithm will stop at a clear contradiction.

Question answering

To formulate queries, we will introduce a new variety of characters – the finite set of individual variables $Var = \{x_0, x_1, \ldots\}$. An atomic query is the expressions of the form $u:C$ or uRv , where is a concept, R is a role, $u,v \in IN \cup Var$.

Definition 8. Conjunctival query is an expression of the form $\exists\overline{v}(t_1 \wedge \ldots \wedge t_k)$*, where* t_i *are atoms,* $\overline{v} = \{v_1,\ldots,v_l\}$ *is a list of some variables included in* t_i*. Variables* v_i *are called related, and the remaining variables are called free. If* $\overline{v} = \{v_1,\ldots,v_l\}$ *is a list of free variables of query* q*, we will write* $q(\overline{v})$*.*

Consider the above mentioned knowledge base. Natural language question *Which films won an Academy Award?* can be written as the following query: $q(x) = x : \exists R_{win}.Oscar$. The answer to the request is a set of individuals that meet these conditions. For the above example, the answer may be presented in the form $\{a_1, a_4\}$. It should be noted, that the theory of knowledge bases is based on the belief of the world openness: the knowledge base is a set of all models in which the axioms given in it are valid. Therefore, the answer to the query to the knowledge base is always a subset of the complete answer to the natural language question, in contrast to the query to the database, which is always the exact set of the complete answer to the question.

Conclusions

The current state of solving the natural language processing problems provides high-quality input data for the task of the natural language texts knowledge base population. Thus, the dependency tree, built according to the Universal Dependencies framework, allows to separate terminological axioms and facts of the knowledge base, including numerical constraints. However, the unresolved problem of finding coreferences for the Ukrainian language does not allow us to speak about a sufficiently high-quality state of solving the problem of populating knowledge bases with Ukrainian-language texts, which confirms the need to work on a corpus of coreferences for the Ukrainian language.

The described approach to population of knowledge bases can be extended to the cases of conditional sentences, causal expressions and adapted to different temporal contexts of statements made in the text. Accordingly, the analysis of knowledge bases containing such information requires the use of an extended apparatus of descriptive logics, including their combination with temporal logics and the use of an additional system of factual axioms.

References

1. G. Stanovsky, I. Dagan. Creating a Large Benchmark for Open Information Extraction. – Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. – Austin, Texas, US. – 2016. – P. 2300–2305.
2. M. Cetto, C. Niklaus, A. Freitas та S. Handschuh. Graphene: A Context-Preserving Open Information Extraction System. – Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations. – Santa Fe, New Mexico. – 2018. – P. 94–98.
3. J. Zhan та H. Zhao. Span Based Open Information Extraction. – 2019.
4. W. Léchelle, F. Gotti та P. Langlais. WiRe57: A Fine-Grained Benchmark for Open Information Extraction. – 2019. – P. 6–15.
5. C. Niklaus, M. Cetto, A. Freitas та S. Handschuh. A Survey on Open Information Extraction. – Proceedings of the 27th International Conference on Computational Linguistics. – Santa Fe, New Mexico, USA. – 2018. – P. 3866–3878.
6. C. Manning, T. Grow, T. Grenager, J. Finkel та J. Bauer. Stanford Tokenizer. – 2002.
7. R. McDonald, J. Nivre, Y. Quirmbach-Brundage, Y. Goldberg, D. Das, K. Ganchev, K. Hall, S. Petrov, H. Zhang, O. Täckström, C. Bedini, N. B. Castelló, and J. Lee. Universal Dependency Annotation for Multilingual Parsing. – In Proceedings of ACL. – 2003. – P. 92–97.
8. B B. Bohnet, R. McDonald, G. Simões, D. Andor, E. Pitler та J. Maynez. Morphosyntactic Tagging with a Meta-BiLSTM Model over Context Sensitive Token Encodings. – Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. – 2018. – P. 2642-2652.
9. K. Mrini, F. Dernoncourt, T. Bui, W. Chang та N. Nakashole. Rethinking Self-Attention: An Interpretable Self-Attentive Encoder-Decoder Parser. – 2019.
10. N. Darchuk Automated syntax analysis of texts from Ukrainian language corpus. – Ukrainian linguistics. – 2013. - № 43. – P. 11-19. (In Ukrainian)
11. J. Devlin, M.-W. Chang, K. Lee та K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. – NAACL-HLT. – 2019. – P. 4171–4186.
12. F. Baader, D. Calvanese, D. McGuinness, D. Nardi та P. Patel-Schneider. The Description Logic Handbook. – Cambridge University Press. – 2007. – P. 578.
13. Serhiienko I., S. Kryvyi, O. Provotar. Algebraic aspects of information technology. – Scientific thought. – 2011. – 399 P. (In Ukrainian)
14. S. Kryvyi, N. Darchuk, O. Provotar. Onlogoly-based systems of natural language analysis. – Problems of programming. – 2018. – № 2-3. – P. 132-139. (In Ukrainian)
15. S. Kryvyi, N. Darchuk, I. Yasenova, O. Holovina, A. Soliar. Methods and means of knowledge representation systems. – Publisher: ITHEA. – Inter. journ. «Information Content and Processing». – 2017. – v. 4 – № 1. – P. 62–99. (In Russian)
16. O. Palagin, S. Kryvyi, N. Petrenko. Knowledge-oriented information systems with the processing of natural language objects: the basis of methodology, architectural and structural organization. – Control Systems and Computers. – 2009. – №3. – P. 42–55. (In Russian)
17. O. Palagin, S. Kryvyi, N. Petrenko. On the automation of the process of extracting knowledge from natural language texts. – Natural and Artificial Intelligence Intern. Book Series. – Inteligent Processing. – ITHEA. – Sofia. – N 9. – 2012. – P. 44–52. (In Russian)
18. O. Palagin, S. Kryvyi, D. Bibikov. Processing natural language sentences using dictionaries and words frequency. – Natural and Artificial Intelligence Intern. Book Series. – Inteligent Processing. – ITHEA. – Sofia. – N 9. – 2010. – P. 44–52. (In Russian)

About authors:
Hryhorii Hoherchak,
1st year PhD student.

Place of work:
Computer Science and Cybernetics Faculty,
Taras Shevchenko National University of Kyiv,

e-mail: gogerchak.g@gmail.com, тел.: +38 (066) 348-55-47.