Mathematical Method of Translation into Ukrainian Sign Language Based on Ontologies

Maksym Davydov(✉) and Olga Lozynska(✉)

Information Systems and Networks Department, Lviv Polytechnic National University,
Lviv, Ukraine
{Maksym.V.Davydov,Olha.V.Lozynska}@lpnu.ua

Abstract. This paper introduces the mathematical method for translation into sign language based on ontologies. The modification of affix context-free grammar (AGFL) that adds semantical attribute and a new form of production called the "template production" is discussed. This new form helps to represent ontology-based productions in a short and computationally inexpensive way. The mathematical method utilizes dictionaries, ontology database, weighted affix context-free grammar (WACFG) parser, algorithm for transformation of constituency tree into dependency tree, and an algorithm for synthesis of Ukrainian sign language glosses. The algorithm for selection and convertion of grammatically augmented ontology (GAO) expressions into the set of WACFG productions is suggested. The major increase in percentage of correctly parsed sentences was achieved for Ukrainian sign language (UKL) and Ukrainian spoken language (USpL). All algorithms are components of the translation system for Ukrainian sign language. Simple video sequencing is utilized for sign language synthesis, however any other sign animation tool can be used. Tasks that require further research are defined.

Keywords: Sentence parser · Ontology · Machine translation · Sign language translation

1 Introduction

Development and study of methods for automatic translation into Ukrainian Sign Language is an urgent task today. Due to results of a survey conducted by Lviv Polytechnic National University, more than 90% of people who communicate using sign language would benefit from devices for automatic translation into sign language and more than 60% of them prefer using smartphone for performing this task. Nevertheless there are many known scientific articles on this issue, the problem of translation into Ukrainian Sign Language there are no viable solution for mobile devices yet.

Successful solution of Sign Language translation should be computationally effective, use limited storage, and require minimal bandwidth for communication with a server.

In the proposed solution translation from Ukrainian spoken language into annotated Ukrainian sign language and vice versa is divided into several steps: parsing sentences

N. Shakhovska and V. Stepashko (eds.), *Advances in Intelligent Systems and Computing II*, Advances in Intelligent Systems and Computing 689,
https://doi.org/10.1007/978-3-319-70581-1_7

using weighted affix context free grammar, the transformation of constituency trees into dependency trees using tree transformation algorithm, transformation of the dependency trees into the sequence of UKL glosses using translation rules for sign language translation, animation of glosses.

The use of ontologies and common sense databases play valuable role in parsing and translation of such languages as Ukrainian spoken language that have flexible word order. Grammatically augmented ontology is an ontology extension that links phrases to their meaning. The link is established via special expressions that connect phrase meaning to grammatical and semantical attributes of words that constitute it. The paper discusses an approach to sentence parsing that is based on integration of ontology relations into productions of weighted affix context-free grammar.

2 Related Work

A lot of linguistic problems such as machine translation, text recognition, information retrieval and extraction require the automatic analysis of sentences.

There are two main approaches for machine translation: the rule-based approach and the statistical approach. In the first approach, human experts specify a set of rules to describe the translation process. In another approach, large parallel corpora are used as source of knowledge. Each of these approaches has its own advantages and challenges. One of the steps of rule-based machine translation is the parsing input sentences.

The problem of sentence parsing has been already studied for a long time. There are a lot of approaches for sentence parsing: syntactic sentence parsing based on generative grammars [1], extended affix grammar (EAG) [2], and stochastic context-free grammar [3]; semantic sentence parsing based on predicate logic [4] and sub-domain driven parsing [5]; syntactic sentence parsing based on semantic relations using statistics [6], ontologies [7], tensor factorization [8] or mixed methods [9]; sentence parsing based on ontologies.

The idea of ontology integration to the process sentence parsing is not new. In [10] the generation of productions from ontologies for LTAG grammar parser was studied. In the article by Faten Kharbat [11] the WordNet ontology [12] is utilized to be the syntactic guide along with the Transition Network Grammar that helps to get better translation from English to Arabic language. The approach based on rich ontologies was used by Murat Temizsoy and Ilyas Cicekli [13] for Turkish language parsing. The approach uses ontologies to improve text meaning representation model for parsed sentences.

The work [14] describes a semi-automatic method for associating a Japanese lexicon with a semantic concept taxonomy called an ontology, using a Japanese-English bilingual dictionary. They developed three algorithms to associate a Japanese lexicon with the concepts of the ontology automatically: the equivalent-word match, the argument match, and the example match. These algorithms were tested on a dataset of 980 nouns, 860 verbs and 520 adjectives and can be effective for more than 80% of the words.

The new architecture for the translation (Italian – Italian Sign Language) that performs syntactic analysis, semantic interpretation and generation are proposed in [15].

They present some general issues of the ontological semantic interpretation that is based on a syntactic analysis that is a dependency tree.

However, the problem of integrating hypernymy/hyponymy relations into the process of sentence parsing was not previously studied. This article introduces a new method that extends the system of productions using ontology relations. These relations are expressions of GAO and hypernymy/hyponymy relations.

3 Main Part

3.1 Architecture and Components of the Developed System

The developed system consists of dictionaries, ontology database, weighted affix context-free grammar parser, algorithm for transformation of constituency tree into dependency tree, and an algorithm for synthesis of Ukrainian sign language glosses.

There are two kinds of used dictionaries: Ukrainian morphology dictionary and Ukrainian Sign Language dictionary.

There are two open-source dictionaries that are suitable for parsing Ukrainian morphology: Spell-uk dictionary maintained by Andriy Rysin and a dictionary supported by Mariana Romanyshyn and others from Grammarly. The first dictionary is widely known as it is used in OpenOffice spell checker. It is very memory effective and requires only 3 MB uncompressed and 500 KB when compressed. Unfortunately it has no direct method to obtain word tags that are required for further parsing of sentences. The second dictionary has more words and contains all necessary tags, but its size is 150 MB uncompressed and 17 MB in compressed state. Thus, the first dictionary is preferred for usage in mobile devices due to its compactness. In the developed system both dictionaries were used: the Spell-uk dictionary was extended with special word tagging rules, and words that were missing or were incorrectly tagged by spell-uk are used from the second dictionary.

The result of tagging a word is a set of hypotheses for its base form and possible grammatical attributes. Some mutually exclusive attributes can be included into the same hypothesis in order to decrease possible search space. This attributes can be refined later in the syntax parser.

For example parsing of Ukrainian word "мати" (mother) leads to the following productions that where added to the set of WACFG productions:

noun[gf c1 n1] → <мати>[gf c1 n1] → мати[r],
noun[gm c1 c4 n*] → <мат>[gm c1 c4 n*] → мати[r],
verb[i n1 m-] → <мати>[i n1 m-] → <мати>[r],

where tags "gm" and "gf" mean male and female gender, "n1" and "n*" mean singular and multiple number, "c1" and "c4" mean nominative and accusative cases respectively, tag "i" means infinitive, and tag "r" means word as it was written in the sentence (i.e. terminal symbol of the grammar). All of these productions have the same structure:

part_of_speech[TAGS] → <base_form>[TAGS] → <word>[r].

Base form of the word can be used later when ontology hyponymy and hypernymy relations are used.

The sentence parsing is done be means of WACFG parser that turned out to be very effective. Grammar productions are written in compact "template form" that assures low memory usage and computationally effective parsing. The system utilizes 230 productions for parsing of Ukrainian language. Two examples of these rules with small description are given below.

For example, production

$$\mathrm{NG}[=] \rightarrow \mathrm{adj}[=]?\ \mathrm{AN}(*)[=]\ \mathrm{NG}[\mathrm{c2}]?$$

is used to describe a noun group that can be a noun with optional general adjective and adjective of place. One of possible phrases that can be handled by this production is "розумний студент Львівської політехніки" (smart student of Lviv Polytechnic University). In the given production NG stands for Noun Group, AN stands for Annotated Noun, "c2" means genitive case, symbol "?" means optional symbol, "=" means all standard attributes of this part of speech, and "*" denotes head word of the phrase that is used later to obtain dependency tree.

Production

$$\mathrm{NG}\left[\mathrm{C\ n} * \mathrm{p3\ gm\ gn\ gf}\right] \rightarrow \mathrm{NG}(*)[\mathrm{C}]\mathrm{conj\ NG}[\mathrm{C}]$$

is used to describe pair of nouns, for example "мило і рушник" (a soap and a towel). Here "C" means that all words should have the same case and the result phrase should have the same case, "p3" means the third person, "conj" means conjunction.

Besides regular productions that are used to describe the grammar of language being parsed, grammatically augmented ontology productions are used. These productions help to identify language constructions specific for particular subject area usage. For example, in phrases "PLAY THE PIANO" and "PLAY FOOTBAL" the word PLAY means completely different actions that are translated into sign language differently. Although such a colocation can be effectively solved using statistical approach it can't be applied right now due to the lack of large parallel corpuses for sign language. Instead of that the approach based on grammatically augmented ontologies is adopted. This approach requires creation of ontology dictionaries for specific target areas that can be effectively achieved by means of the developed GAODL language described earlier in [16].

The ontology is used to incorporate word abstraction rules into the grammar. For example, productions that were generated for subject area education had the following form:

<людина> → <школяр> (human → pupil)
<може-вчитись> → <людина> (can-learn → human)
<наука> → <математика> (science → math)
<містить-знання> → <наука> (contains-knowledge → science)
<вчити-навчати> → <вчити> (teach-proc → teach)
<вчити-навчати> → <вчити-навчати>(*) <може-вчитись>[c4] (teach-smb → teach-proc can-learn)

<вчити-навчати> → <вчити-навчати> <містить-знання>[c3] (teach-smth → teach-proc contains-knowledge)

These rules can be effectively used for semantical parsing of phrases like "вчити школярів математики" (teach math to pupils). More information about use of grammatically augmented ontologies for sentence parsing can be found in [17].

3.2 The Algorithm for Parsing Sentences

The problem of sentence parsing is formulated as a problem of finding a sequence of productions that has the maximum weight and can be applied sequentially to some starting attributed symbol (S, A_s) to produce a given sequence of terminals $t_1 t_2 \ldots t_n$. The weight of the sequence is calculated as a multiplication of weights of all contained productions.

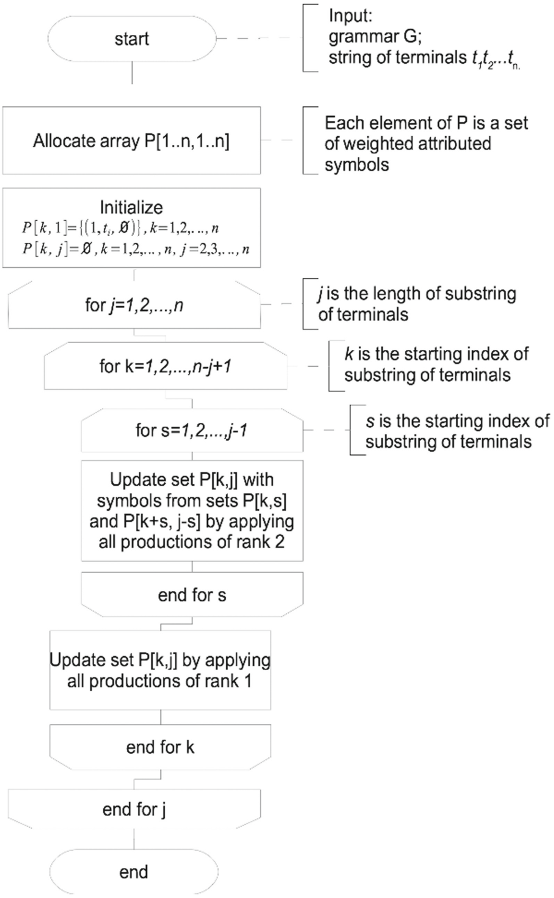

start
Input:
grammar G;
string of terminals $t_1 t_2 \ldots t_n$.
Allocate array P[1..n,1..n]
Each element of P is a set of weighted attributed symbols
Initialize
$P[k,1]=\{(1,t_i,\varnothing)\}, k=1,2,\ldots,n$
$P[k,j]=\varnothing, k=1,2,\ldots,n, j=2,3,\ldots,n$
for $j=1,2,\ldots,n$
j is the length of substring of terminals
for k=$1,2,\ldots,n-j+1$
k is the starting index of substring of terminals
for s=$1,2,\ldots,j-1$
s is the starting index of substring of terminals
Update set P[k,j] with symbols from sets P[k,s] and P[k+s, j-s] by applying all productions of rank 2
end for s
Update set P[k,j] by applying all productions of rank 1
end for k
end for j
end

Fig. 1. The block scheme of sentence parsing algorithm.

The block scheme of parsing algorithm is shown on Fig. 1.

The algorithm provided above uses internal procedure for updating set of weighted attributes symbols Q with set of possible left symbols L and set of possible right symbols R by applying of rank 2. The block scheme of this procedure is depicted on Fig. 2.

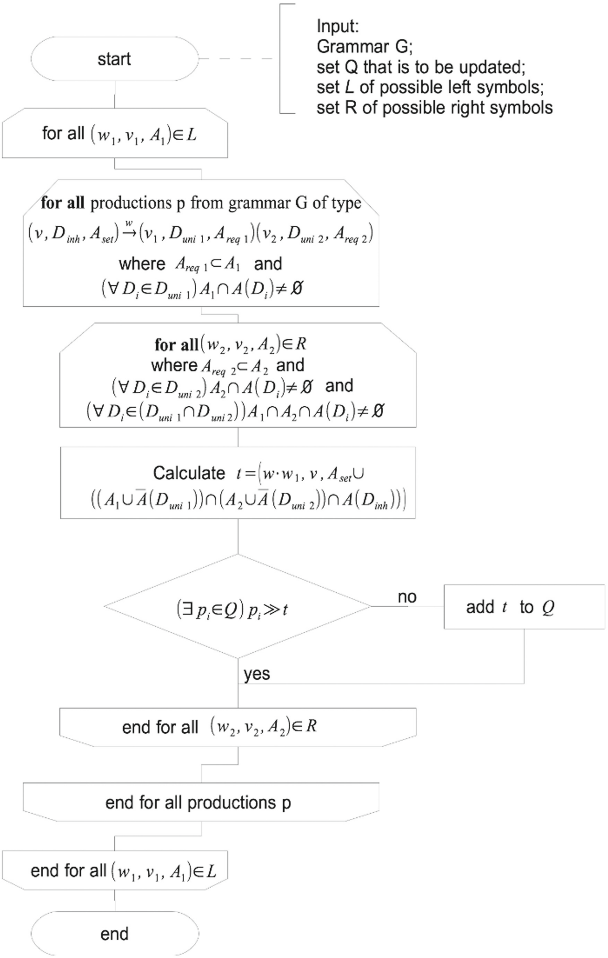

start
Input:
Grammar G;
set Q that is to be updated;
set L of possible left symbols;
set R of possible right symbols
for all $(w_1, v_1, A_1) \in L$
for all productions p from grammar G of type
$(v, D_{inh}, A_{set}) \xrightarrow{w} (v_1, D_{uni\ 1}, A_{req\ 1})(v_2, D_{uni\ 2}, A_{req\ 2})$
where $A_{req\ 1} \subset A_1$ and
$(\forall D_i \in D_{uni\ 1}) A_1 \cap A(D_i) \neq \emptyset$
for all $(w_2, v_2, A_2) \in R$
where $A_{req\ 2} \subset A_2$ and
$(\forall D_i \in D_{uni\ 2}) A_2 \cap A(D_i) \neq \emptyset$ and
$(\forall D_i \in (D_{uni\ 1} \cap D_{uni\ 2})) A_1 \cap A_2 \cap A(D_i) \neq \emptyset$
Calculate $t = \langle w \cdot w_1, v, A_{set} \cup$
$((A_1 \cup \bar{A}(D_{uni\ 1})) \cap (A_2 \cup \bar{A}(D_{uni\ 2})) \cap A(D_{inh}))\rangle$
$(\exists p_i \in Q)\, p_i \gg t$
no
add t to Q
yes
end for all $(w_2, v_2, A_2) \in R$
end for all productions p
end for all $(w_1, v_1, A_1) \in L$
end

Fig. 2. The block scheme of procedure for updating set of weighted attributes symbols Q.

If the right part of a production contains only one symbol, the weight of the production should not exceed 1 in order to avoid cyclic productions that increase weight of non-terminal symbols during the bottom-up parsing procedure.

3.3 Extending the Set of Productions with Ontology Relations

The grammar augmented ontology was introduced in [18]. Along with relations that are common to ontology databases (hyponymy, hypernymy, meronymy, holonymy) GAO contains relations that link synsets to expressions with associated grammatical attributes.

In order to benefit from ontology knowledge new productions were added to generative grammar. The addition of ontology productions into the generative grammar extends the set of semantic attributes. Each synset of ontology was treated as semantic attribute. For the purpose of efficiency semantic attributes and corresponding productions were added only for hierarchies that contained words that were present in the sentence being parsed.

The algorithm that adds new productions to syntactic parser is shown on Fig. 3. A more detailed algorithm presented in [19].

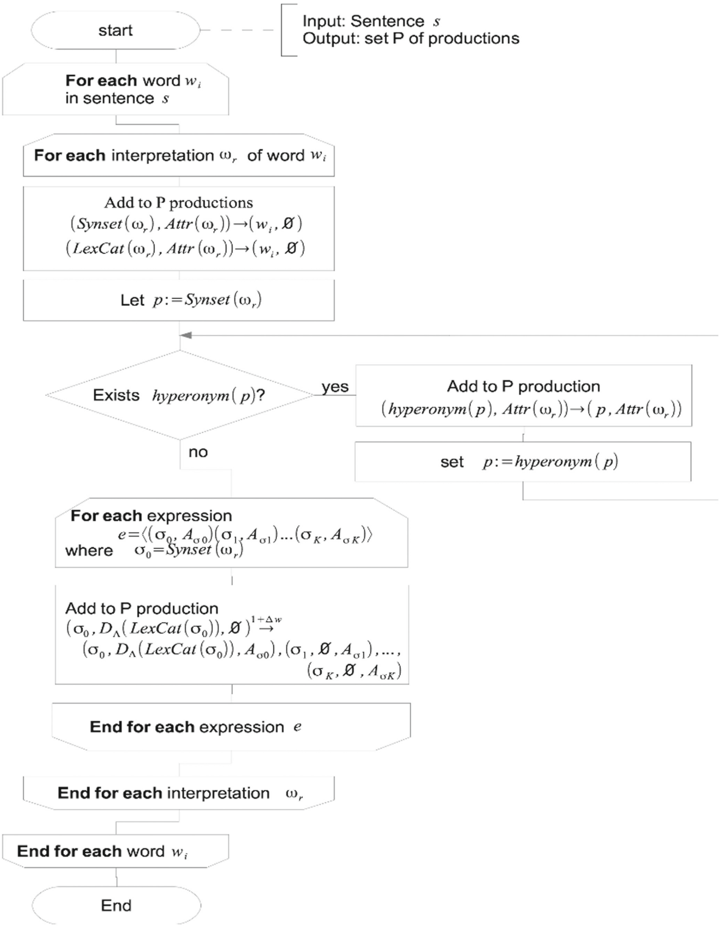

start
Input: Sentence s
Output: set P of productions
For each word w_i in sentence s
For each interpretation ω_r of word w_i
Add to P productions
$(Synset(\omega_r), Attr(\omega_r)) \to (w_i, \varnothing)$
$(LexCat(\omega_r), Attr(\omega_r)) \to (w_i, \varnothing)$
Let $p := Synset(\omega_r)$
Exists $hyperonym(p)$?
yes
Add to P production
$(hyperonym(p), Attr(\omega_r)) \to (p, Attr(\omega_r))$
set $p := hyperonym(p)$
no
For each expression
$e = \langle(\sigma_0, A_{\sigma 0})(\sigma_1, A_{\sigma 1})\ldots(\sigma_K, A_{\sigma K})\rangle$
where $\sigma_0 = Synset(\omega_r)$
Add to P production
$(\sigma_0, D_\Lambda(LexCat(\sigma_0)), \varnothing) \overset{1+\Delta w}{\to}$
$(\sigma_0, D_\Lambda(LexCat(\sigma_0)), A_{\sigma 0}), (\sigma_1, \varnothing, A_{\sigma 1}), \ldots,$
$(\sigma_K, \varnothing, A_{\sigma K})$
End for each expression e
End for each interpretation ω_r
End for each word w_i
End

Fig. 3. The algorithm that adds new productions to syntactic parser.

Each word w in the sentence can have several interpretations $\omega_1, \omega_2, \ldots, \omega_r \in \Omega$, where Ω is a dictionary of all interpretations. Each interpretation ω uniquely defines its semantic attribute *SemAttr*(ω), lexical category *LexCat*(ω) and the set of grammatical attributes *GrAttr*(ω). A tuple $e = \left\langle \left(\sigma_1, A_{\sigma 1}\right)\left(\sigma_2, A_{\sigma 2}\right) \ldots \left(\sigma^*, A_\sigma^*\right) \ldots \left(\sigma_K, A_{\sigma K}\right) \right\rangle$ – each expression in GAO, where σ_i is a synset that narrows the set of words that can appear in the given position of the expression e and $A_{\sigma i}$ is a set of grammar attributes the word is required to possess; σ^* is a head word of the expression. Let $\Lambda = \{noun, verb, adjective, adverb, \ldots\}$ be a set of all lexical categories and $D_\Lambda : \Lambda \to 2^D$ be a mapping from lexical category to the set of its attribute domains.

Productions that are generated from ontology expression have larger weight than simple syntactic productions in order to dominate over them. Additional weight Δw in expression is devised from the admissibility of the expression in the given context or text topic.

The result of paring the sentences “My father bought several candies at the table” and “My father bought several candies at the shop” using ontologies is depicted on Fig. 4.

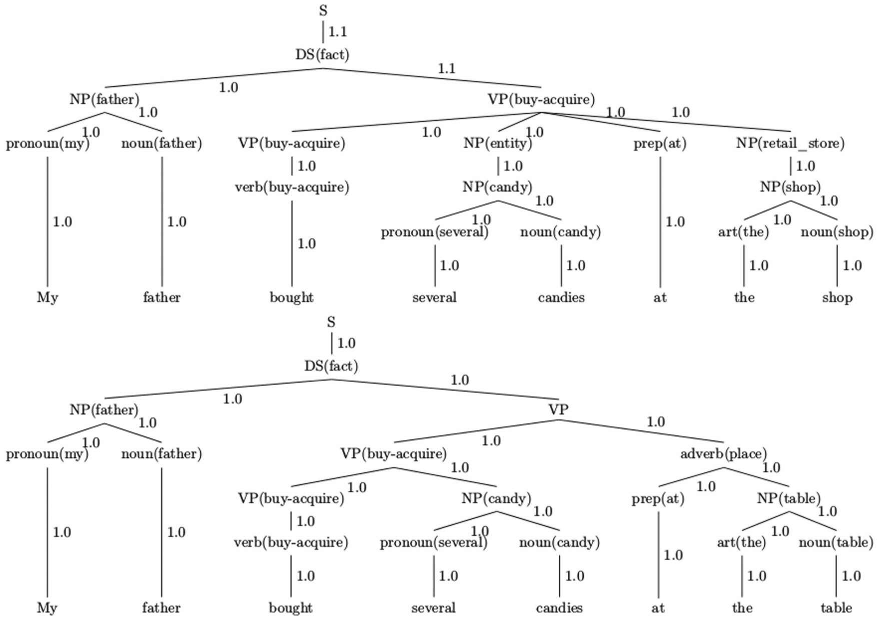


Fig. 4. The result of parsing sentences “My father bought several candies at the table” and “My father bought several candies at the shop”. The weight of the first parse is higher because one ontology expression was used while the second parsing tree is based only on syntactic productions.

The next step after parsing a sentence is to convert constituency tree to dependency tree. The algorithm of conversion takes constituency tree as input. The head of each phrase is used to determine main word of the sentence and main word of each sub-phrase. The core of the algorithm is to identify the head of each phrase in the constituency tree and establish its relation with the head of its parent node.

The transformation algorithm comprises of the following steps (Fig. 5):

1. Find word that is a head of the entire tree and mark all vertices in the tree that have the same head.
2. Join vertices obtained in step 1 into a single vertex of the dependency tree.
3. Apply steps 1–2 recursively for every sub-tree of marked vertices.
4. Verify dependencies in the obtained dependency tree and mark edges where words could not be in parent-child relationship as improper.
5. For all improper relations find better correspondence by width-first search in the dependency tree for other possible parent word.

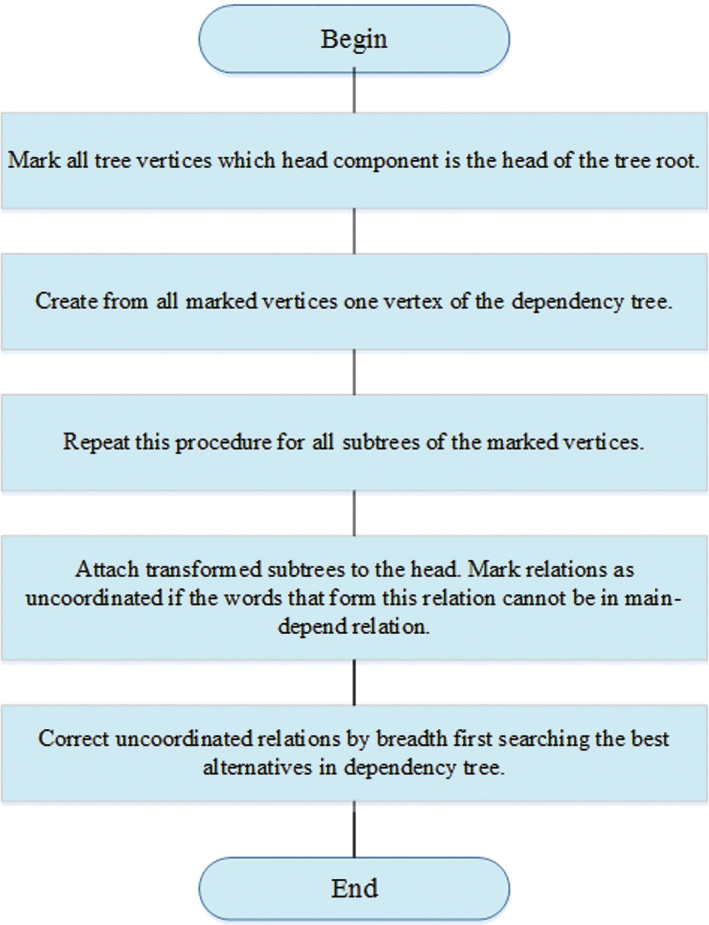

Begin
Mark all tree vertices which head component is the head of the tree root.
Create from all marked vertices one vertex of the dependency tree.
Repeat this procedure for all subtrees of the marked vertices.
Attach transformed subtrees to the head. Mark relations as uncoordinated if the words that form this relation cannot be in main-depend relation.
Correct uncoordinated relations by breadth first searching the best alternatives in dependency tree.
End

Fig. 5. Algorithm for transformation of constituency tree to dependency tree.

4 Results

The result of the transformation step is a dependency tree that can be used to produce translation. An example of a tree for sentence "My father bought several candies at the shop" is shown in Fig. 6.

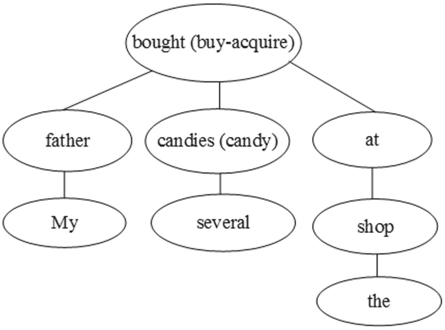
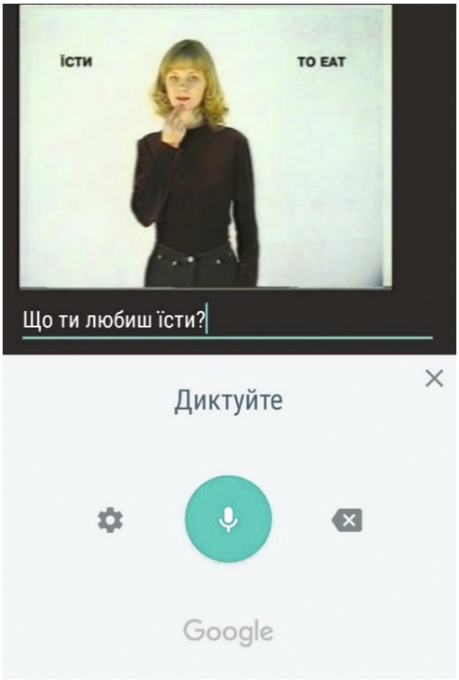


Fig. 6. Dependency tree obtained for sentence "My father bought several candies at the shop".



Що ти любиш їсти?

Fig. 7. Mobile app interface for entering text and translating into sign language.

The translation is produced using transformation rules described in [20].

The information system of translation takes an input text of Ukrainian spoken language. The input can be obtained by typing word with a screen keyboard or by means of speech recognition (Fig. 7).

After entering the text, it is split into sentences and their topics are determined. Productions from grammatically augmented ontology are added when possible and sentences are parsed and translated. The translation is performed involving transfer rules base, which consists of rules for sign language translation and reordering rules that are used to generate the text in UKL.

Evaluation of translation results was performed by comparing sentences with translations available in the database of test sentences. The database of WACFG productions and the database of grammatically augmented ontology were updated by adding new rules and new synsets respectively.

5 Conclusion

The mathematical method for translation into sign language based on ontologies are described in the paper.

The developed information system consists of dictionaries, ontology database, weighted affix context-free grammar parser, algorithm for transformation of constituency tree into dependency tree, and an algorithm for synthesis of Ukrainian sign language glosses. The use of WACFG parser for sentence parsing has allowed to increase the percentage of correctly translated sentences. The obtained sentence parsing trees are more semantically rich than the parsing trees obtained by means of regular syntactic parser. The system utilizes 230 productions for parsing of Ukrainian language. The transformation algorithm from constituency tree into dependency tree has shown high efficiency (89% correct sentences converted) and the possibility of its use in machine translation systems.

Further research can be focused on improving the quality of the information system for translation, adding new rules into WACFG parser, extending the set of synsets in grammatically augmented ontology and devising new rules for the transformation algorithm.

References

1. Chomsky, N.: Three models for the description of language. IRE Trans. Inf. Theor. **2**(3), 113–124 (1956)
2. Oostdijk, N.: An extended affix grammar for the english noun phrase. In: Aarts, J., Meijs, W. (eds.) Corpus Linguistics. Recent Developments in the Use of Computer Corpora in English Language Research, pp. 95–122. Rodopi, Amsterdam (1984)
3. Eddy, S.R., Durbin, R.: RNA sequence analysis using covariance models. Nucleic Acids Res. **22**(11), 2079–2088 (1994)
4. Blackburn, P., Bos, J.: Representation and Inference for Natural Language: A First Course in Computational Semantics. CSLI Publications, Stanford (2005)

5. Plank, B., Sima'an, K.: Subdomain sensitive statistical parsing using raw corpora. In: Proceedings Sixth International Conference on Language Resources and Evaluation, pp. 465–469. European Language Resources Association, Marrakech (2008)
6. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of the International Conference on Research in Computational Linguistics, Taiwan, pp. 19–33 (1997)
7. Rhee, S.K., Lee, J., Park, M.-W.: Ontology-based semantic relevance measure. In: Proceedings of the First International Workshop on Semantic Web and Web 2.0 in Architectural, Product and Engineering Design, Busan, Korea, pp. 63–68 (2007)
8. Anisimov, A., Marchenko, O., Vozniuk, T.: Determining semantic valences of ontology concepts by means of nonnegative factorization of tensors of large text corpora. Cybern. Syst. Anal. **50**(3), 327–337 (2014)
9. Nagarajan, M., Sheth, A.P., Aguilera, M., Keeton, K., Merchant, A., Uysal, M.: Altering document term vectors for classification - ontologies as expectations of co-occurrence. In: Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada, pp. 1225–1226 (2007)
10. Unger, C., Hieber, F., Cimiano, P.: Generating LTAG grammars from a lexicon-ontology interface. In: Proceedings of the 10th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+10), Yale University, pp. 61–68 (2010)
11. Kharbat, F.: A new architecure for translation engine using ontology: one step ahead. In: Proceedings of The International Arab Conference on Information Technology (ACIT 2011), Saudi Arabia, pp. 169–173 (2011)
12. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
13. Temizsoy, M., Cicekli, I.: An ontology based approach to parsing Turkish sentences. In: Proceedings of AMTA 1998-Conference, pp. 124–135. Springer, Langhorne (1998)
14. Okumura, A., Hovy, E.H.: Building Japanese-English dictionary based on ontology for machine translation. In: HLT 1994 Proceedings of the Workshop on Human Language Technology, Plainsboro, pp. 141–146 (1994)
15. Lesmo, L., Mazzei, R., Radicioni, D.P.: An ontology based architecture for translation. In: Proceedings of the Ninth International Conference on Computational Semantics, Oxford, UK, pp. 345–349 (2011)
16. Lozynska, O.V., Davydov, M.V.: Domain-specific language for describing grammatically augmented ontology. Control Syst. Mach. **4**, 31–40 (2015)
17. Lozynska, O., Davydov, M.: Information technology for Ukrainian Sign Language translation based on ontologies. Econtechmod Int. Q. J. **4**(2), 13–18 (2015)
18. Davydov, M., Lozynska, O.: Spoken and sign language processing using grammatically augmented ontology. Appl. Comput. Sci. **11**(2), 29–42 (2015)
19. Davydov, M., Lozynska, O., Pasichnyk, V.: Partial semantic parsing of sentences by means of grammatically augmented ontology and weighted affix context-free grammar. Econtechmod Int. Q. J. **6**(2), 27–32 (2017)
20. Lozynska, O.V., Davydov, M.V., Pasichnyk, V.V.: Rule-based machine translation into Ukrainian Sign Language. Inf. Technol. Comput. Eng. Sci. J. VNTU **1**(29), 11–17 (2014)