Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2018”

Moscow, May 30—June 2, 2018

SEMI-AUTOMATIC INTEGRATION OF A NEW LANGUAGE INTO A MULTILINGUAL NLP MODEL: THE CASE OF JAPANESE

Petrova M. A. (maria_pet@abbyy.com),
Druzhkina A. A. (anna_r@abbyy.com),
Garashchuk R. V. (ruslan_g@abbyy.com),
Yudina M. V. (maria_yu@abbyy.com)
ABBYY, Moscow, Russia

The current paper deals with the integration of the Japanese language in a multilingual NLP model, namely, the Compreno model. The formalism includes morphological, syntactic and semantic patterns, covering all possible semantic and syntactic dependencies a word can attach. The architecture of the model allows us to acquire nearly all semantic links of a word through its proper positioning in a thesaurus-like semantic hierarchy, where words are linked through semantic dependencies. The inheritance principle of the hierarchy simplifies the syntactic description of a newly added language as well. Unlike the traditional approach to Japanese parsing based on chunks, or bunsetsus, we suggest a Japanese parser based on constituents. Special attention is given to the tools that allow us to automatize language description process and significantly speed up the description. The work on the Japanese model is still in progress, therefore, we show the current results we have achieved, and point out problems that remain to be solved.

Keywords: Japanese parsing, multi-lingual parsing, semantic and syntactic analysis, formal language models, information extraction

ПОЛУАВТОМАТИЧЕСКАЯ ИНТЕГРАЦИЯ НОВОГО ЯЗЫКА В МНОГОЯЗЫЧНУЮ NLP-МОДЕЛЬ (НА ПРИМЕРЕ ЯПОНСКОГО ЯЗЫКА)

Петрова М. А. (maria_pet@abbyy.com),
Дружкина А. А. (anna_r@abbyy.com),
Гаращук Р. В. (ruslan_g@abbyy.com),
Юдина М. В. (maria_yu@abbyy.com)
ABBYY, Москва, Россия

1. Introduction and related work

The given paper is devoted to the integration of the Japanese language in the ABBYY Compreno model—NLP model based on morphological, syntactic and semantic text analysis. The model serves as the basis for a dependency parser and helps to apply text mining algorithms to different NLP tasks. Currently, it functions for English, Russian, German and, partly, for French, Spanish and Chinese.

Here we focus on linguistic problems bound with the process of automating language description, and show our experience of introducing new tools, which make the description semi-automatic and, therefore, more effective.

Japanese NLP tools are in high demand now, and there are quite a few works devoted to Japanese parsing ([Uchimoto et al. 2000]; [Kudo and Matsumoto 2002]; [Kurohashi and Nagao 1994]; [Kawahara and Kurohashi 2006]; [Kawahara et al. 2017]; [Tanaka, Nagata 2015], and others).

Traditionally, Japanese parsers differ significantly from the ones for European languages, as they are mostly based on syntactic dependencies modeled in terms of chunks called bunsetsus instead of constituents (the example of bunsetsus-based parsers are CaboCha [Kudo, Matsumoto 2002] and KNP [Kawahara, Kurohashi 2006]).

Taking the specificity of the Japanese language into account (especially, the problem of text division into words), this approach has some benefits. However, the bunsetsus-based models have significant disadvantages bound with the difficulties of setting correspondences between the bunsetsus and constituents. As [Tanaka, Nagata 2015: 237] points out, it “complicates the task of extracting semantic units from bunsetsus-based representations” and makes the analysis of non-tree links such as coordination problematic.

There are different studies aimed at improving parsing quality.

Recently, word-based dependency schemes have been suggested for Japanese, particularly within the UD project ([Nivre et al. 2016]; [Kanayama et al. 2015]; [Tanaka et al. 2016]). Besides, there are studies showing that adding lexical knowledge can significantly improve dependency analysis [Kawahara et al. 2017]. The

use of case frames has also been reported to enhance parsing quality [Kawahara, Kurohashi 2006]; [Kawahara et al. 2017].

We suggest a different approach, in which the Japanese parser is based on a linguistic model, which includes not only lexical knowledge, but a full language description, covering all possible semantic and syntactic dependencies a word can attach. This is possible due to the integration of the Japanese vocabulary into the universal semantic model, which is much broader than the case frames application. Unlike bunsetsus-oriented parsers, our Japanese parser is based on constituents, which provides faster and easier integration of Japanese in a multilingual system.

The structure of the paper is as follows. First, we give a short description of the Compreno linguistic model in general, as far as it is necessary for further understanding (for more details see [Anisimovich et al. 2012; Manicheva et al. 2012; Petrova 2014]). Then we focus on integrating Japanese in the formalism and characterize the semantic and syntactic patterns of our Japanese description, drawing particular attention to the automation methods. Following this, we illustrate the work of the parser based on the given model—both for English and Japanese, and give a short description of the corpora annotation used in the project. Finally, we offer the conclusion, where the results are summarized and further perspectives are given.

2. The Compreno linguistic model

The Compreno model is based on a multilingual lexical database organized in the form of a thesaurus-like hierarchy (Semantic Hierarchy, hereafter as SH)—a hyperhyponymy relation tree built on a universal language, an interlingua. The branches of the tree are the so-called universal semantic classes (SCs)—universal labels, or boxes, which are filled with the contents in different natural languages. For instance, the tree includes the path such as PHYSICAL_OBJECT > BEING > ANIMAL > MYTHOLOGICAL_ANIMAL > DRAGON, and the DRAGON class is filled with the English 'dragon', German 'Drache', Japanese '竜', and so on.

The semantic links between words are provided through the *Deep Slots* (DSs)—semantic roles, under which we understand any semantic dependency a word can attach, like agent in '*[the cat] ran away*', or evaluation characteristic in '*a [nice] house*'.

The basic SH principle is the inheritance principle: all the DSs and other semantic features are introduced as high as possible in the hierarchy, and the lower branches inherit them. Such a strategy minimizes the amount of work necessary for the description of each word's semantic links: that is, when a new word is positioned in the hierarchy, it inherits nearly all possible semantic links it can have.

The SCs and the DSs are universal and do not depend on any definite language. It means that when we add a new language in the model, the semantic part of its description comes to efficient word positioning in the hierarchy, which provides the word with all the necessary DSs at once.

Each DS has a number of syntactic realizations—so called *Surface*, or *Syntactic*, *Slots* (SSs). Unlike the DSs, SSs are not universal. For each SS, we specify its grammar value—define the parts of speech that can fill it, indicate case, prepositions and other grammatical information, set its order in a sentence, and punctuation. For example,

the Agent DS corresponds to the $Subject SS in '*[the boy] reads a book*' and to the $Object_Indirect_By slot—in the passive transformation '*the book is read [by the boy]*'.

The syntactic pattern of a newly added language demands more work, as syntax is special for each language. Although the inheritance principle helps here as well, we still face a great deal of work trying to determine, first, which surface realizations each DS can have, and, second, adding this information to the model.

3. Adding the Japanese language in the model

3.1. The semantic description: word positioning in the hierarchy

As shown above, proper word positioning in the SH is the key point of the semantic pattern, as it provides each word with an entire semantic model.

Previous work with other languages has proven that manual descriptions based on dictionaries are ineffective and take too much time. To overcome this, we developed a semi-automated (or semi-supervised) approach to adding new vocabulary, which was first used for the description of the German language (for details, see [Goncharova et al. 2015]). Using the approach, we created an auxiliary dictionary-like tool: on the one hand, it accumulated all relevant information from dictionaries and corpora—meanings of words, examples, and grammatical features; on the other hand, it automatically suggested a SC for each meaning of the word, and a linguist had only to approve or reject it.

In the current formalism, the number of the word meanings corresponds to the number of the SCs where a unique lexeme is represented. That is, each pair *lexeme—SC* represents one word meaning. We can get a number of such pairs for each language of the model. Moreover, we can get a frequency of each pair parsing parallel corpora with the Compreno parser, and, therefore, we obtain a variety of statistically ranged meanings.

When the work on the Japanese morphological system was completed, we aligned the Japanese-English parallel corpora and dictionaries with our alignment parser, found word pairs, where a certain Japanese word form corresponds to a certain English word form, and counted the frequency for each pair. As we have already had the mapping of the English word forms into SCs, we could obtain some hypotheses, or suggestions, for positioning Japanese words as well.

Therefore, we got a number of suggestions for each Japanese lexeme on where to place it in the SH and ranged them according to their frequency.

When we started using the tool for German, the percent of the correct suggestions in top 5 hypotheses was about 0.6 at first. By the time we started the Japanese description, the tool was significantly improved, and the algorithm switched from the heuristic-based approach to machine learning.

Namely, we evaluated the correctness of the German suggestions after the German vocabulary had been checked by linguists, and taught the system on it, as the classifier estimates the good and the bad features of the hypotheses (the features include word's/suggestion's part of speech, source of the suggestion (dictionary or parallel

texts), distances between meanings in the source and target languages in the SH, depth of the suggestion in the SH, and other). As a machine learning method, gradient boosting over decision trees has been chosen.

Currently the algorithm gives us 0.72 precision within the top 5 results. It increased the speed of the Japanese description about 5 times in comparison with the German one, when the tool was used for the first time (we do not compare it with the speed of English and Russian, as their description was done together with elaborating the SH, DSs, SSs and other universal features of the system, which also took time).

Another option of the word positioning tool is to analyze, which additional semantic and grammatical features a word can have. It suggests not only the proper place for a word, but also other features, such as *semantemes* (universal features, which distinguish antonyms like *bad* and *good*, for instance) and *grammemes* (language-specific features, which describe the syntactic behaviour of a word (mark transitivity or government, for example).

Automatic suggestions like these come from several sources. First, some grammemes are shared within languages, like 'CharacteristicParametric' for parametric nouns; therefore, we assume that if English or Russian descendants of some SC have this grammeme, it is most likely that Japanese descendants can need it, too. Second, some suggestions are calculated from the models of the Japanese lexis already introduced in the hierarchy: if a Japanese word has some semanteme or grammeme, it is likely that its newly added neighbors will need them, too. Third, there are grammemes that are always relevant for some word groups,—for instance, all verbs must have a transitivity marker. Therefore, transitivity grammemes are always suggested when dealing with a verb.

A linguist now only has to test whether the positioning hypothesis is right, and if yes, to choose additional features from the list of the already generated suggestions. If the suggestion is incorrect, which is usually easy to find out from the information provided with the vocabulary tool (definition, different examples from the web, and so on), the correct SC can be chosen manually.

Currently we have more than 35,000 Japanese lexical units in the SH. For comparison, the total number of the universal SCs is about 190,000, and the number of English and Russian lexical units is nearly 270,000 and 247,000, correspondingly.

3.2. The syntactic description

Unlike semantics, syntax is special for every language. Therefore, when we add a new language in the model, we have to make a full description of its syntax. To make the work faster, we use the tools described below and turn to the inheritance principle again.

However, different dependencies demand different strategies. There are DSs that have the same syntactic realizations with every core they can be attached to, and the expression of some DSs depends on the cores they combine with.

That is, the syntactic realization of the adjuncts such as Purpose, Cause, or Condition does not depend on the verb they are bound with. For instance, every verb that can attach the reason slot can have ため-reason adjunct, as in example (1):

(1) 列車　は　[雪　の　ために]　遅れた
ressya-train　wa-Nom　yuki-snow　no-Gen　tameni-Caus　okureta-be delayed-past
The train was delayed [because of snow].

This means that we can indicate just once that the cause DS corresponds to the cause adjunct with the necessary grammatical properties. Therefore, the main task for the surface description of such DSs is to find all possible syntactic realizations for them.

To achieve it, we take parallel English-Japanese texts and analyze their English part with our parser in order to get all possible constituents corresponding to the necessary DS. In this way, we get parent-child pairs for each DS we need. After this, we find all possible Japanese correspondences for the lexemes that fill the DS. As Japanese is a left branching language, we check additionally that the supposed child node precedes the supposed parent node, and find the postposition closely following the child. Therefore, we get a table-like catalogue of possible grammar realizations of each DS.

All grammar realizations are grouped into separate files according to realization markers. The files are ranged by the frequency of each realization: the larger the file, the more examples were found for this particular marker in the current search.

Each file contains a table, which provides: a) a source language instance with the parent node in red and the child node in blue (based on the default syntactic analysis by Compreno); b) a corresponding target language instance with the same colour code plus the marker in green; c) the vocabulary form for both nodes and the marker in the target language.

For instance, see Table 1—a small fragment of the file for the Cause DS expressed through the から postposition:

Table 1. The Cause DS expressed through the から postposition

source language instance	target language instance	the vocabulary form of parent and child nodes and the marker in the target language
I don't say that just because of your circumstances.	あなたの境遇から 言った訳ではない	言う から 境遇
He was called 'Eiki no oyakata' (Master in Eiki) because of his address.	住所から「永木の親方」と呼ばれた。	呼ぶ から 住所
For some reason, he grew up in a fatherless family.	家庭的な事情から、母子家庭で育つ。	育つ から 事情
Inventions are born, so to speak, of necessity.	発明はいわば必要から生まれるのだ。	生む から 必要
It is also called akoyamochi (lit 'oyster mochi') because of its shape.	その形からあこや餅とも呼ばれることもある。	呼ぶ から 形
Because of the importance of their role, they were allowed to adopt surnames and wear pairs of swords.	その役目の重要性から苗字帯刀を許されていた。	許す から 重要性

Nevertheless, the expression of some DSs depends on the core predicate. Mainly, this concerns the actant DSs, such as agent, object, experiencer, or alike. For example, in sentences (2)

(2) *He touched* [the water] *with his foot.*
I gave [a present] *to my friend.*

the bracketed constituent corresponds to the [Object] DS. The Japanese verbs 触れる [fureru] 'to touch' and 上げる [ageru] 'to give' have different government: 触れる has *ni*-Object (Dative Object) and 上げる demands *wo*-Object (Accusative Object). This means we must not only indicate that the [Object] DS can be expressed through *wo*-groups and *ni*-groups, but also indicate that different cores demand different syntactic realizations of the object-slot.

We describe this information in a semi-automated manner. First, we assign all possible realizations for the DSs like [Object]. Then we add grammemes for each type of the object-government, and provide the verbs with the necessary grammemes, namely, 触れる acquires the <NiObject>-grammeme and 上げる—the <AccusativeObject>. All possible surface realizations of the [Object] DS are introduced high in the hierarchy, but the core of each correspondence is marked with the necessary grammeme: the *ni*-realization demands that the core verb should have the <NiObject>-grammeme, and the *wo*-realization demands the accusative grammeme.

When a verb is placed in the SH, relevant grammemes are suggested within the procedure described above.

This means that the syntactic description of most of the DSs, such as adjuncts and characteristics, is universal, so to say, as their syntactic realizations are introduced only once in the SH. The syntactic description of the DSs, which have lexicalized realization, is done in a semi-automated manner.

The model includes about 330 DSs. Currently, more than 70% of them are provided with Japanese surface realizations. However, the fullness of this part is still being checked. The number of DSs that demand partly lexicalized description is less than 10%. All the rest can be described universally.

3.3. Cross-language asymmetry

Of course, there are a lot of cases of asymmetry between Japanese and other languages of the model, which concern lexicon, voice system, serial verb constructions, copula absence in complement constructions with predicative adjectives, classifiers, or counter words, and a number of other things. The description of these cases is problematic for automation and demands manual work. Different kinds of asymmetry demand different solutions. Due to the lack of space, we cannot provide their detailed description here, and will have to restrict ourselves with a few examples.

Nevertheless, there are many asymmetry cases between the languages that have already been integrated in the model, and the basic principles for dealing with language asymmetry (such as using transformational rules and collocations) are the same for all languages of the model, including Japanese. A detailed description of the methods we use for it is given in [Petrova 2014]. Some instances from Chinese, German and French are suggested in [Manicheva et al. 2012] as well.

As a Japanese instance, let us take lexical asymmetry. There are many concepts in Japanese that are special for Japan, so we do not have SCs for them, like 温泉 [onsen] 'hot spring', 炬燵 [kotatsu] 'Japanese table', 先輩 [sempai] 'senior', and so on.

In addition, Japanese abounds with ‘compound’ words, like 陳述書 ‘written declaration’, 魔界 ‘world of spirits’, or 遅咲き ‘late blooming’.

In cases like these, we usually have to add new SCs to the hierarchy, put the required Japanese concepts there and provide their correspondences in other languages of the model. If the equivalent is not a word but a collocation (like ‘hot spring’), we fill the SCs with *terms*—collocations that can be put in particular places of the SH like lexical classes.

The particular cases are concepts composed of antonyms (or somehow opposed notions), or the notions that usually come together: 男女 ‘men and women’, 和戦 ‘war and peace’, 花鳥 ‘flowers and birds‘. Since ‘flower’ and ‘bird’ are definitely two different notions, the corresponding words are “situated” in different SCs, so it is not quite clear where we should place ‘花鳥’. Anyway, in most cases, we have to add such words to the SH as well, as it facilitates lexical analysis: otherwise, each time the model would have to choose between analyzing the hieroglyph separately, or as a part of a compound word.

4. The Compreno parser and its application

The semantic and syntactic patterns discussed above serve as the basis for the Compreno parser, which includes several other patterns, such as morphology, non-tree links like conjunction, anaphora, and others as well (for detailed analysis, see [Anisimovich et al. 2012]; [Bogdanov, Leontyev 2013]).

Analyzing a sentence, the parser finds a set of syntactic structures that can be matched to it, and then ranges them according to their evaluation. As a result, the parser builds syntactic-semantic structures with the following nodes: SSs and DSs, lexical and semantic classes, semantic and grammatical value, non-tree links—for example, it finds a host for each pronoun, and so on (non-tree links are of great importance for parsing, however, due to the lack of space, we have to omit their description here and refer to the papers mentioned above).

An illustration of the parser’s work is figure 1, where the output tree for the English sentence (3) is given:

(3) *I gave a present to my friend.*

Fig. 1. English output tree

In other words, our parser builds a representation of a sentence or a text. Although the current model-based approach demands relatively higher costs, the model

shows good evaluation results within the tests such as the "Dialogue evaluation competitions" (http://www.dialog-21.ru/evaluation/), which include a wide range of tasks: morphological analysis, anaphora, entity and fact extraction, machine translation and others (for details, see [Anastasyev et al. 2017]; [Stepanova et al. 2016]; [Bogdanov et al. 2014]; [Zuev et al. 2013]). Moreover, automation methods help to reduce the costs significantly.

As Japanese description is still in progress, we have not made comparative evaluations with other Japanese parsers yet. However, the important idea is that application of the parser to different NLP tasks is to a large extent based on the universal model. For instance, our data extraction mechanism used for English and Russian now relies on the information it gets from the SCs, DSs and tree dependencies—namely, universal objects, which are not language-specific. It means that when we add a new language to the parser, we still use the same universal structures that we referred to when working with English or Russian. Therefore, most of the IE rules would work for the Japanese IE as well, which helps us to avoid significant work when starting to use the Japanese parser for the tasks like this one.

As stated above, the number of Japanese lexical units added to the system is currently rather modest in comparison with English and Russian. Moreover, so far the syntactic description of Japanese is not complete either. Nevertheless, the Japanese parser functions already and fulfils the semantic and syntactic analysis on limited text collections (the reference to the treebank is given below in section 5). As an instance of Japanese parsing, see example (4) and figure 2 with its output tree (the sentence comes from the open treebank below):

(4) 日本にはたくさんの美しい場所がある。— *There are many lovely places in Japan.*

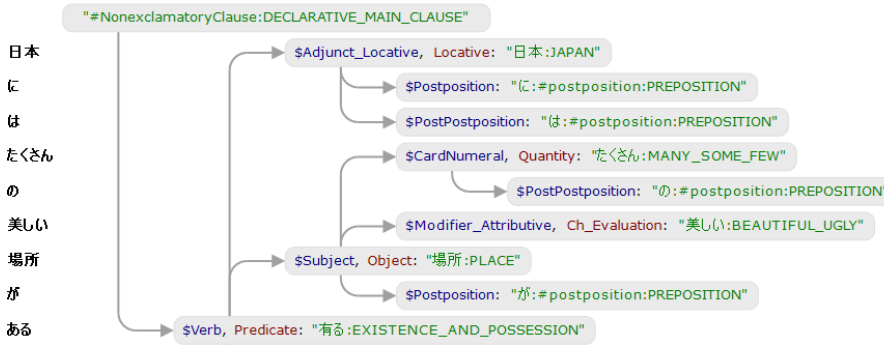

Fig. 2. Japanese output tree

5. Text annotation and Japanese treebank

To evaluate quality change of the parser, we use manually annotated text collections for all languages integrated in the model. Usually, annotation includes DSs and SSs for all constituents.

Our annotation standards have quite a lot in common with the UD principles. Yet, there are significant differences as our annotation is aimed at our project needs and correlates with different opportunities of the model. Unlike the UD, we reconstruct ellipted constituents, which is important for correct information extraction. We treat coordination differently, as in Compreno, all conjuncts are attached to one parent, while in the UD, the first conjunct attaches the other ones. In the UD, punctuators are linked to other constituents, while in Compreno, punctuator is an attribute of a SS, and so on. In general, our annotation demands more competence from the annotator, but gives more precision for our needs.

In future, we plan to open access to some of our annotated corpora, therefore, the opportunity to convert our annotation in the UD standard is in question now.

At the moment, our Japanese treebank consisting of 1,500 sentences is available here: https://github.com/ComprenoData/JapaneseTreebank. The original texts come from the Tatoeba project (https://tatoeba.org/eng), and these are annotated with shallow constituent borders by means of our parser. In addition to the treebank, we suggest the annotation manual at the treebank website, where the annotation syntax and principles are described in greater detail.

6. Conclusion

Integrating Japanese in a formal multilingual model is a challenging task, which faces quite a few difficulties. Nevertheless, the Compreno model proved to be an effective tool for dealing with languages of different groups.

First, the universal SH suits well for word positioning of the lexicon of different languages, including the asymmetry cases. Second, the system of the universal DSs and the inheritance principle allow us to provide each word with all possible semantic links at once purely through the word's positioning in the SH. Third, the architecture of the model reduces significantly the amount of work on the syntactic pattern as well, as the syntactic realizations of the DSs can be introduced high in the SH and be inherited by the SC-descendants.

Though such a model-based approach is rather costly, application of the auxiliary tools and machine learning methods helps to reduce the costs significantly, facilitates and speeds up the description process, and allows us to avoid a number of mistakes inevitable in manual work.

Currently we are continuing to enlarge the Japanese vocabulary added to the SH, progress with the work on the Japanese syntax and start testing the Japanese part of the model on larger text corpora in order to evaluate the current level of the description and to track its progress. At the same time, we plan to focus on the practical application of the Japanese parser and use it for solving different NLP tasks, in particular, for information extraction.

References

1. *Anastasyev D. G., Andrianov A. I., Indenbom E. M.* (2017), Part-of-speech Tagging with Rich Language Description.In Computational Linguistics and Intellectual Technologies. Proceedings of the International Conference "Dialogue 2017", vol. 1, pp. 2–13.
2. *Anisimovich K., Druzhkin K., Minlos F., Petrova M., Selegey V., and Zuev K.* (2012), Syntactic and semantic parser based on ABBYY Compreno linguistic technologies. In Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue", vol. 11, pp. 91–103.
3. *Bogdanov A. V., Dzhumaev S. S., Skorinkin D. A., and Starostin A. S.* (2014), Anaphora analysis based on ABBYY Compreno linguistic technologies. In Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue", pp. 89–101.
4. *Bogdanov A. V., Leontyev A. P.* (2013), Description of the Russian External Possessor Construction in a Natural Language Processing System. In Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue". [Komp'iuternaia Lingvistika i Intellektual'nye Tehnologii: Trudy Mezhdunarodnoj Konferentsii "Dialog 2013] Bekasovo, pp. 110–118.
5. *Goncharova M., Kozlova E., Pasyukov A., Garashchuk R., and Selegey, V.* (2015), Model-based WSA as means of new language integration into a multilingual lexical-semantic database with interlingua. In Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue", vol. 1, pp. 169–182.
6. *Kanayama H., Miyao Y., Tanaka T., Mori S., Asahara M., Uematsu S.* (2015), A draft of universal dependencies for Japanese. In the 21st annual meeting of the Association for Natural Language Processing, pp. 505–508.
7. *Kawahara D., Kurohashi S.* (2006), A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL 2006), pp. 176–183.
8. *Kawahara D., Hayashibe Y., Morita H., Kurohashi S.* (2017), Automatically Acquired Lexical Knowledge Improves Japanese Joint Morphological and Dependency Analysis. In Proceedings of the 15th International Conference on Parsing Technologies, Pisa, pp. 1–10.
9. *Kudo T., Matsumoto Y.* (2002), Japanese dependency analysis using cascaded chunking. In Proceedings of the 6th Conference on Natural Language Learning (CoNLL-2002), Vol. 20, pp. 1–7.
10. *Kurohashi S., Nagao M.* (1994), A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. Computational Linguistics, 20(4), pp. 507–534.
11. *Manicheva E., Petrova M., Kozlova E., and Popova T.* (2012), The Compreno Semantic Model as Integral Framework for Multilingual Lexical Database. In Zock, M. and R. Rapp (eds), Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon (CogALex-III), COLING. Mumbai, pp. 215–229.

12. *Nivre J., de Marneffe M.-C., Ginter F., Goldberg Y., Hajic J., Manning C. D., McDonald R., Petrov S., Pyysalo S., Silveira N., Tsarfaty R., and Zeman D.* (2016), Universal dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari (Conference Chair) et al., editors, Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Paris, France, may. European Language Resources Association (ELRA), pp. 1659–1666.
13. *Petrova M. A.* (2014), The Compreno Semantic Model: The Universality Problem. In International Journal of Lexicography, Volume 27, Issue 2, pp. 105–129.
14. *Tanaka T., Nagata M.* (2015), Word-based Japanese typed dependency parsing with grammatical function analysis. ACL (2), pp. 237–242.
15. *Tanaka T., Miyao Y., Asahara M., Uematsu S., Kanayama H., Shinsuke M., and Matsumoto Y.* (2016), Universal dependencies for Japanese. In Proceedings of the 10th International Conference on Language Resources and Evaluation. LREC, pp. 1651–1658.
16. *Stepanova M. E., Budnikov E. A., Chelombeeva A. N., Matavina P. V., and Skorinkin D. A.* (2016), Information Extraction Based on Deep Syntactic-Semantic Analysis. In Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue", pp. 721–732.
17. *Uchimoto K., Murata M., Sekine S, and Isahara H.* (2000), Dependency model using posterior context. In Proceedings of the 6th International Workshop on Parsing Technology, pp. 321–322.
18. *Zuyev K. A., Indenbom E. M., and Yudina M. V.* (2013), Statistical machine translation with linguistic language model. In Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue", vol. 2, pp. 175–183.