A neural parser as a direct classifier for head-final languages

Hiroshi Kanayama    Masayasu Muraoka    Ryosuke Kohita
IBM Research
Tokyo, Japan
{hkana, mmuraoka}@jp.ibm.com, Ryosuke.Kohita1@ibm.com

Abstract

This paper demonstrates a neural parser implementation suitable for consistently head-final languages such as Japanese. Unlike the transition- and graph-based algorithms in most state-of-the-art parsers, our parser directly selects the head word of a dependent from a limited number of candidates. This method drastically simplifies the model so that we can easily interpret the output of the neural model. Moreover, by exploiting grammatical knowledge to restrict possible modification types, we can control the output of the parser to reduce specific errors without adding annotated corpora. The neural parser performed well both on conventional Japanese corpora and the Japanese version of Universal Dependency corpus, and the advantages of distributed representations were observed in the comparison with the non-neural conventional model.

1 Introduction

Dependency parsing helps a lot to give intuitive relationships between words such as noun-verb and adjective-noun combinations. Those outputs are consumed in text mining systems (Nasukawa and Nagano, 2001) and rule-based approaches such as in fine-grained sentiment extractors (Kanayama et al., 2004), though some of recent end-to-end systems do not require intermediate parsing structures.

Many recent dependency parsers have been implemented with neural net (NN) methods with (typically bidirectional) LSTM and distributed word vectors (Dozat et al., 2017; Shi et al., 2017), as we can see in the 2017 shared task on dependency parsing from raw text for 49 languages (Zeman et al., 2017) based on the multilingual corpora of Universal Dependencies (UD) (Nivre et al., 2015).

Most of such dependency parsers exploit a transition-based algorithm (Nivre et al., 2007), a graph-based algorithm (McDonald et al., 2005) or a combination of both (Nivre and McDonald, 2008). Those algorithms addressed several problems in multilingual dependency analysis such as bidirectional dependency relationships and non-projective sentences. However, it is hard to intuitively interpret the actions to be trained on the transition-based parser. Though it can handle the history of past parsing actions, the output may violate syntactic constraints due to the limitation of visible histories. The graph-based approach captures global information in a sentence, but the difficulty in reflecting the interaction of attachment decisions causes contradictory labels in a tree.

The parsing results from the participants in the 2017 shared task show low scores on Japanese (67 to 82% in the UAS scores, excluding the team that provided the data) in particular, which shows that the language-universal approaches do not work effectively for Japanese.¹

¹Note that another factor in the low score was the inconsistent tokenization (84 to 93% F1 value).

The syntactic structures in the Japanese version of Universal Dependencies (Tanaka et al., 2016; Asahara et al., 2018) have dependencies in both directions, as well as in other languages, since it is based on the word level annotations and the content-head dependency schema. However, when the syntactic structures are expressed with the dependencies between phrasal units (so-called bunsetsus in Japanese; ‘PU’ hereafter in this paper), the head element always comes in the right position, since Japanese is a consistently head-final language. This allows us to apply a method for such languages to simplify the model. We con-
structured a neural parsing model to directly select the head word among the limited candidates. The model works as a classifier that outputs intuitive and consistent results while exploiting grammatical knowledge.

Section 2 reviews the head-final property of Japanese and the Triplet/Quadruplet Model (Kanayama et al., 2000) to exploit syntactic knowledge in a machine-learning parser. Section 3 designs our neural model relying on the grammatical knowledge, and its experimental results are reported in Section 4. Other head-final languages are discussed in Section 5 and some related approaches are discussed in Section 6. Section 7 concludes this paper.

2 Background

First, Section 2.1 shows the head-final property of the Japanese language. and Sections 2.2 and 2.3 explain the main ideas in the Triplet/Quadruplet Model: the methods of simplification of dependency parsing task using the linguistic knowledge.

2.1 Head-final structure in Japanese

Figure 1 shows an example of a word-level dependency structure of a Japanese sentence in the representation of Universal Dependencies. Traditionally Japanese dependency parsers have been evaluated on the unit of bunsetsu, a phrasal unit (PU), as performed in Kyoto University Text Corpus (Kawahara et al., 2002). A PU consists of a content word and optional functional words and prefixes and suffixes. Figure 2 depicts the dependency structure represented in PUs, where the all dependencies are in a single direction. The head PU is always in the right and the rightmost PU is always the root, as long as exceptional inversion cases are not cared. In this paper we exploit this property to apply a simplified parsing algorithm: the parsing can be regarded as the selection of the head PU from the limited number of candidates PUs located to the right of the dependent in question. For example, the second PU “赤い” (“red”) in Figure 2 must modify one of the four PUs from the third “ボールを” (“ball’-ACC) to the sixth PU “遊んでいる” (“play’-PROG). The correct head is “ボールを” (“ball’-ACC).

2.2 Restriction of modification candidates

The head-final feature can further simplify the dependency parsing by adding syntactic constraints. The Triplet/Quadruplet Model (Kanayama et al., 2000) has been proposed to achieve the statistical dependency parsing making most of the linguistic knowledge and heuristics. In their work, a small number (about 50) of hand-crafted grammar rules determine whether a PU can modify each PU to its right in a sentence as shown in Table 1. In the rules, the modified PUs are determined on the conditions of the rightmost morpheme in the modifier PU. In addition to PoS-level relationships,

---


d2In case of compound nouns and verbs, multiple content words may be included in a single PU.

be simplified by restricting the candidates to these
by the grammar rules. Therefore, the model can
Table 2: Percentages of the position of the correct
modified PU among the candidate PUs selected
by the initial grammar rules. The column ‘Sum’
shows the coverage of the 1st, 2nd and last (the far-
nearest PU, the second nearest PU, or
propositional “から” kara (‘from’) proper noun + propositional “から”
postpositional “れた” no (genitive, nominative) postpositional “と” to (conjunctive) postpositional “と” to (conjunctive)
appearance of those attributes. Second, the probabil-
detailed condition with specific functional words
and exceptional content words are covered in the
rules. Even with these simplified rules, 98.5% of
the modifications between PUs are covered.

The role of the grammar rules is to maximize
the coverage, and the rules are simply describing
high-level syntactic dependencies so that the rules
can be created easily without worrying about pre-
cision or contradictory rules. The statistical infor-
mation is later used to select the rules necessary
for a given sentence to produce an accurate pars-
ing result.

Furthermore, an analysis of the EDR corpus
shows that 98.6% of the correct dependencies are
either the nearest PU, the second nearest PU, or
the farthest PU from the modifier (more details in
Table 2) among the modifiable PUs enumerated
by the grammar rules. Therefore, the model can
be simplified by restricting the candidates to these

| # of candidates | Ratio | 1st | 2nd | Last | Sum  |
|-----------------|-------|-----|-----|------|------|
| 1               | 32.7  | 100.0 | –   | –    | 100.0 |
| 2               | 28.1  | 74.3 | 26.7 | –    | 100.0 |
| 3               | 17.5  | 70.6 | 12.6 | 16.8 | 100.0 |
| 4               | 9.9   | 70.4 | 11.1 | 13.8 | 95.3  |
| ≥5              | 11.8  | 70.2 | 11.1 | 10.5 | 91.9  |
| Total           | 100   | –   | –   | –    | 98.6  |

Table 1: The excerpt of Japanese grammar rules. The left side is the condition of the modifier PU
specified with the rightmost morpheme (except for punctuation) with optional preceding morphemes,
and the right side is the list of the condition for the modifiable PUs specified with the head word and
optional functional words.

detailed condition with specific functional words
and exceptional content words are covered in the
rules. Even with these simplified rules, 98.5% of
the modifications between PUs are covered.

The role of the grammar rules is to maximize
the coverage, and the rules are simply describing
high-level syntactic dependencies so that the rules
can be created easily without worrying about pre-
cision or contradictory rules. The statistical infor-
mation is later used to select the rules necessary
for a given sentence to produce an accurate pars-
ing result.

Furthermore, an analysis of the EDR corpus
shows that 98.6% of the correct dependencies are
either the nearest PU, the second nearest PU, or
the farthest PU from the modifier (more details in
Table 2) among the modifiable PUs enumerated
by the grammar rules. Therefore, the model can
be simplified by restricting the candidates to these
two or three candidates and by ignoring the other
PUs with a small sacrifice (1.4%) of parsing ac-
curacy. Retaining the farthest modifiable PU from
the modifier, the long distance dependencies are captured.

2.3 Calculation of modification probabilities
Let \( u \) be a modifier PU in question, \( c_{un} \) the \( u \)’s \( n \)th modification candidate PU, \( \Phi_u \) and \( \Psi_{cu_n} \) the
respective attributes of \( u \) and \( c_{un} \). Then the probability
of \( u \) modifying its \( n \)th candidate is calculated
by the triplet equation (1) when \( u \) has two candidates
or the quadruplet equation (2) when \( u \) has
three candidates. \(^3\) These two equations are known
as the Triplet and Quadruplet Model.

\[
P(u \leftarrow c_{un}) = P(n | \Phi_u, \Psi_{cu_1}, \Psi_{cu_2}) \tag{1}
\]

\[
P(u \leftarrow c_{un}) = P(n | \Phi_u, \Psi_{cu_1}, \Phi_{cu_2}, \Psi_{cu_3}) \tag{2}
\]

Assuming the independence of those modifications,
the probability of the dependency tree for
an entire sentence \( P(T) \) is calculated as the product
of the probabilities of all of the dependencies
in the sentence using beam search from the right-
most PU to the left, to maximize \( P(T) \) under the
constraints of the projected structure.

\[
P(T) \approx \prod_u P(u \leftarrow c_{un}) \tag{3}
\]

Equations (1) and (2) have two major advan-
tages. First, all the attributes of the modifier and
its candidates can be handled simultaneously – the
model expresses the context through the combi-
nation of those attributes. Second, the probab-
ity of each modification is calculated based on the

\(^3\)It is trivial to show that \( P(u \leftarrow cu_1) = 1 \), when \( u \) has only one candidate.
Figure 3: The neural net for the quadruplet model to select the head of the PU “社長を” (‘president’-ACC) from three modification candidates in an example sentence “…以前の社長を1買収した企業に紹介2事業を開始した3” (‘… introduced2 the previous president0 to the acquired1 company, and launched3 a … business’). The attributes of the modifier PU and three modification candidates, and the features between the modifier and each candidate are input as distributed vectors.

relative positions of the candidates, instead of the distance from the modifier PU in the surface sentence, making the model more robust.

3 NN parsing model

In the past implementation of the parser by the Triplet/Quadruplet model (Kanayama et al., 2000), the equations (1) and (2) were calculated with logistic regression (maximum entropy method) in which many binary features represent the attributes of each PU. We designed the NN model using distributed representation of words and parts-of-speech as Chen and Manning (2014) did.

Figure 3 shows the neural net model that directly selects the head PU and an example sentence. Here, the head of “社長を” (‘president-ACC’) is predicted among three modification candidates selected by the method described in Section 2.2. The second candidate PU “紹介、” (‘introduced-ADV’) is the correct head.

To make the prediction, the attributes for each PU are extracted, and we focus on two words in a PU: the head word – the rightmost content word in the PU – and the form word – the rightmost functional word in the PU except for a punctuation. First, the surface form and the PoS of the head word and the form word are converted into vector representations. That is, two vectors are used for 6 PUs in the triplet model and 8 PUs are used in the quadruplet model. Furthermore, the following attributes between two PUs are added.

- the number of a postpositional “は ha” between two PUs (0, 1, 2, 3, 4, or 5+)
- the number of commas between two PUs (0, 1, 2, 3, 4, or 5+)
- the distance between two PUs (1, 2, 3, ..., 9, or 10+)

These features expressed as vectors are concatenated to form a single layer, and the final output is given as the softmax of two or three values (1, 2, or 3).

The above calculation computes the probabilities of the modification to candidate PUs, but it does so independently for each modifier PU; therefore, there may be crossing of modification in a whole sentence. Since the Japanese dependency structures are fully projective, the optimal tree for the sentence is constructed using a beam search to maximize the Equation (3) in Section 2.3, excluding modification pairs that violate the projective constraint in each step of the beam search. More specifically, combinations of dependencies which violate projective constraints (e.g. 5 ← 7 and 6 ← 8) are excluded from the beam, then the projective tree structure is guaranteed.

4 Experiments

4.1 Experimental settings

We used EDR Japanese Corpus (EDR, 1996) for the initial training and evaluation. After remov-
| Training method     | Training size | Accuracy   |
|--------------------|---------------|------------|
| nearest baseline   | none          | 62.03% (13581/21894) |
| logistic regression| 190k          | 88.92% (19468/21894) |
| NN                 | 40k           | 88.15% (19300/21894) |
| NN                 | 80k           | 88.49% (19373/21894) |
| NN                 | 120k          | 89.17% (19522/21894) |
| NN                 | 160k          | 89.31% (19554/21894) |

Table 3: The accuracy of PU dependencies tested on EDR corpus. The ‘nearest baseline’ denotes the ratio of the case where the head is the right next PU.

| Content words | Commas | Accuracy |
|---------------|--------|----------|
| Yes           | Yes    | **89.31%** |
| No            | Yes    | 88.95%   |
| Yes           | No     | 87.40%   |
| No            | No     | 87.24%   |

Table 4: Ablation studies to remove content word vocabularies and commas.

4.2 Experimental results

Table 3 shows the accuracies of dependency parsing by the conventional model trained with logistic regression and our proposed neural net model. Both models used the very similar grammar rules and the features are used. While the logistic regression method required manual selection of combination of features to get optimal accuracy, the neural net model outperformed the others when the training corpus was more than 120k sentences, by 0.4 points when the training corpus was 160k sentences. The maximum number of the training data in neural net model (160k) is less than that used in the logistic regression method (190k) because the development set needed to be kept for the neural model, and some sentences were dropped as described in Section 4.1.

Only words in the modifier PU and the candidate PUs were used in these methods, and other surrounding context and other dependencies were not considered. By capturing appropriate contexts of candidate PUs selected by the grammar rules and heuristics, our method successfully predicted the dependencies with relatively small pieces of information compared to the initial transition-based neural parser (Chen and Manning, 2014) that used a maximum of 18 words in the buffer, stack and modifiers.

There was a huge difference in the training speed. The logistic regression method took 4 to 20 hours on a CPU, but the neural net model converged in 5 to 15 seconds on a GPU.

We conducted an ablation study to see the contribution of attributes. We focused on the vocabulary of content words that can be better captured using distributed representation rather than the conventional method, and commas that played an important role in suggesting long-distance attachments. According to the results in Table 4, the contribution of the content words (the vocabulary size was 11,362) was not very big; even if the content words were ignored, the loss of accuracy was only 0.36 points. On the other hand, the model ignoring commas (where all of the features regarding commas was removed) downgraded the
accuracy by nearly 2 points, which suggests that commas are important in parsing.

The example dependency in Figure 3 (‘president’ ← ‘introduced’) was correctly solved by our neural model. Though many PUs followed the modifier PU in question, the model selected the head word from only three candidates restricted by the grammar rules, and the known dependency relationship between two PUs is guaranteed to be associated with the parsing result. The conventional model without neural net wrongly selected the first candidate (‘acquired’) as the head. The ablation of the content words also made the prediction wrong, that clarified that it was because the conventional model did not capture the content words and the attributes in the distance were stronger. On the other hand, the neural model with the content word embeddings appropriately captured the relationship between the functional word in the modifier PU (accusative case) and the content word of the correct candidate PU (‘introduced’).

4.3 Comparison on Japanese UD

To compare the performance of our parser with the results in the 2017 Shared Task (Zeman et al., 2017), we apply the trained model to UD Japanese-GSD \(^5\) test data. As shown in Figure 4, the word-based dependency in UD Japanese and PU-based dependencies are interchangeable with a strict rule to detect PU boundaries and the head word in a PU. In UD Japanese-GSD data, the attachment direction between head words in PUs is always the same after converting to the PU-based structure. Also the non-head words in a PU always depends on the head word of the PU.

The first section of Table 5 shows comparisons with the shared task results. Here we evaluate them by UAS (unlabeled attachment score) rather than by LAS (labeled attachment score) because most of the Japanese labels can be deterministically assigned with the combination of the head and the dependent, and assignment with the rules can reproduce the labels in UD Japanese-GSD corpus, thus it is not fair to compare LAS with other machine learning methods. The scores are associated with tokenization accuracies because they highly affect Japanese parsing accuracies in the shared task to handle raw text inputs. Our model performed better than any other results in the shared task, though the comparison is not completely fair since we rely on the segmentation and functional word attachment based on the consistent rules with the UD data creation.

In the 2017 Shared Task, the raw text was used as the input, thus the performance of sentence splitting and tokenization highly affected the parsing result.\(^6\) To make more direct comparisons in parsing, our results were mapped with the baseline tokenzation by UDPipe (Straka et al., 2016) which many task participants have used. That is, the parsing score was intentionally downgraded, but it outperformed any other results which used UDPipe tokenization as it was, as shown in the second section of Table 5.

Also we compared our parser when the gold tokens are given, with UDPipe UDv2.0 model (Straka and Straková, 2017), and RBG Parser (Lei et al., 2014) which was trained with UD Japanese-GSD training set. Our model had 9% and 20% less errors than UDPipe and RBG Parser, respectively.

5 Application to other languages

Our approach relies on the head-final feature of the languages. In addition to Japanese, Korean and Tamil are categorized as rigid head-final languages (Polinsky, 2012). Table 6 shows the ratio of head-final dependencies by languages in the Universal Dependencies version 2.0 development data. Though the word-level dependencies in UD do not reflect the head finalness as only 45% of Japanese dependencies have the head in the right side, but when it comes to content words (the list of functional PoSs are shown in the caption of Table 6) without functional labels and exceptional labels such as conjunction (see the caption again),

\(^5\)It was formerly known as UD Japanese until 2017.

\(^6\)The mismatched tokens are always regarded as parsing errors in the calculation of UAS/LAS.
Table 5: F1 scores of tokenization and UAS on the UD Japanese-GSD test set. The top section shows the systems which used their own tokenizers. The second section is a comparison with the systems relying on the default settings of UDPipe, and the bottom section is the situation to ru parsers using the gold PoS as input.

| Models                  | Tokens | UAS  |
|-------------------------|--------|------|
| **Own tokenizers**      |        |      |
| Our model               | 98.61  | 94.03|
| TRL (Kanayama et al., 2017) | 98.59  | 91.14|
| HIT-SCIR (Che et al., 2017) | 92.95  | 81.94|
| **UDPipe default tokenization** |        |      |
| Our model - UDPipe aligned | 89.41  | 75.88|
| C2L2 (Shi et al., 2017)  | 89.68  | 75.46|
| Stanford (Dozat et al., 2017) | 89.68  | 75.42|
| **Gold tokenization**   |        |      |
| Our model - with gold tokenization | 100.0  | 95.97|
| UDPipe UDv2.0 model (Straka and Straková, 2017) | 100.0  | 95.48|
| RBG Parser (Lei et al., 2014) | 100.0  | 94.94|

Japanese has the complete head-final structures, and Korean and Tamil have high ratios supporting the linguistic theory.

However, the UD Korean corpus has so many coordination structures under the UD’s general constraint that the left coordinate should be the head, because many subordinating structures are represented as coordination while corresponding Japanese ones are not, that it is difficult to convert the corpus to the strictly head-final structure. That is the reason why we could not evaluate our method on Korean, but the Triplet/Quadruplet Model has been applied to Korean with similar grammatical rules and it has been shown that the transfer learning from Japanese worked (Kanayama et al., 2014), thus our neural classifier approach to Korean parsing is expected to work well.

Also UD Tamil data has exceptional cases in proper nouns and other phenomena, and the relatively small corpus made the further investigation difficult. We are leaving it to future work.

6 Related work

The parsing approach to select heads of dependents has been proposed by Zhang et al. (2017). They applied bidirectional RNN to select the probability that each word chooses another word or the ROOT node as its head. They reported comparable results for four languages. Their method required a maximum spanning tree algorithm to generate valid trees. On the other hand, our approach straightforwardly outputs the projective tree by exploiting the head-final feature in Japanese.

Martínez-Alonso et al. (2017) shares the similar
motivation with ours. They tackled multilingual parsing by using a small set of attachment rules determined with Universal POS, and achieved 55 UAS value with predicted PoS as input. Our method applied a neural model on top of the grammatical restriction to achieve higher accuracy for a specific language.

García et al. (2017) tackled the multilingual shared task with the rule-based approach. The rules are simplified with the almost delexicalized PoS-level constraints and created with a small effort by an expert. Though the performance was limited compared to other supervised approaches, it is meaningful for the comparison of linguistic features, and the combination with machine learning methods can be useful as we are aiming at.

7 Conclusion

In this paper we implemented a neural net parsing model as the direct classifier to predict the attachment of phrasal units in a intuitive manner by exploiting grammatical knowledge and heuristics, and confirmed that the neural net model outperformed the conventional machine learning method, and our method worked better than the shared task results. Unlike the most of neural parsing methods in which interpretation of the model output and control of the model without data supervision are difficult, our method is simple enough to understand the behavior of the model, and the grammatical knowledge can be reflected in the restriction of modification candidates. Moreover, the neural net with distributed vector representations enabled us to handle more vocabularies than the logistic regression with distinct word features in which only the limited number of content words and their combination with other features could be distinguished in the parsing model.

Our experiments showed that a limited number of words are seen as able to predict attachments. For further improvement, we can integrate the LSTM model, which handles more contextual information with simplification (Cross and Huang, 2016). We did not handle coordination relationships explicitly in this work, but we will intend to address coordination with more lexical knowledge and a broader context.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kadlec, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Masayuki Asahara, Hiroshi Kanayama, Takaaki Tanaka, Yusuke Miyao, Sumire Uematsu, Shinsuke Mori, Yuji Matsumoto, Mai Omura, and Yugo Murawaki. 2018. Universal Dependencies version 2 for Japanese. In Proceedings of LREC 2018.

Wanxiang Che, Jiang Guo, Yuxuan Wang, Bo Zheng, Huaipeng Zhao, Yang Liu, Dechuan Teng, and Ting Liu. 2017. The hit-scri system for end-to-end parsing of universal dependencies. Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies pages 52–62.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pages 740–750.

James Cross and Liang Huang. 2016. Incremental parsing with minimal features using bi-directional lstm. In The 54th Annual Meeting of the Association for Computational Linguistics. page 32.

Timothy Dozat, Peng Qi, and Christopher D Manning. 2017. Stanford’s graph-based neural dependency parser at the conll 2017 shared task. Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies pages 20–30.

EDR. 1996. EDR (Japan Electronic Dictionary Research Institute, Ltd.) electronic dictionary version 1.5 technical guide.

Marcos García and Pablo Gamallo. 2017. A rule-based system for cross-lingual parsing of romance languages with universal dependencies. In CoNLL Shared Task.

Hiroshi Kanayama, Masayasu Muraoka, and Katsumasa Yoshikawa. 2017. A semi-universal pipelined approach to the conll 2017 ud shared task. Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies pages 265–273.

Hiroshi Kanayama, Tetsuya Nasukawa, and Hideo Watanabe. 2004. Deeper sentiment analysis using machine translation technology. In COLING 2004. pages 494–500.
Hiroshi Kanayama, Youngja Park, Yuta Tsuboi, and Dongmook Yi. 2014. Learning from a neighbor: Adapting a Japanese parser for Korean through feature transfer learning. In Proceedings of the EMNLP ’2014 Workshop on Language Technology for Closely Related Languages and Language Variants. pages 2–12.

Hiroshi Kanayama, Kentaro Torisawa, Yutaka Matsuishi, and Jun’ichi Tsujii. 2000. A hybrid Japanese parser with hand-crafted grammar and statistics. In Proceedings of the 18th International Conference on Computational Linguistics. pages 411–417.

Daisuke Kawahara, Sadao Kurohashi, and Kōiti Hasida. 2002. Construction of a Japanese relevance-tagged corpus. In LREC.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Tao Lei, Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2014. Low-rank tensors for scoring dependency structures. Association for Computational Linguistics.

Héctor Martínez Alonso, Željko Agić, Barbara Plank, and Anders Søgaard. 2017. Parsing universal dependencies without training. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. Valencia, Spain, pages 230–240.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pages 523–530.

Tetsuya Nasukawa and Tohru Nagano. 2001. Text analysis and knowledge mining system. IBM systems journal 40(4):967–984.

Joakim Nivre, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richard Farkas, Jennifer Foster, Filip Ginter, Yoav Goldberg, Jan Hajič, Jenna Kanerva, Veronika Laippala, Alessandro Lenci, Teresa Lynn, Christopher Manning, Ryan McDonald, Anna Missila, Simonetta Montemagni, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Maria Simi, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. 2015. Universal dependencies 1.0.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chaney, Gülşen Erýiýit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. Natural Language Engineering 13(2):95–135.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. Proceedings of ACL-08 pages 950–958.

Maria Polinsky. 2012. Headness, again. UCLA Working Papers in Linguistics, Theories of Everything 17:348–359.

Tianze Shi, Felix G Wu, Xilun Chen, and Yao Cheng. 2017. Combining global models for parsing universal dependencies. Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies pages 31–39.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In LREC.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Association for Computational Linguistics, pages 88–99.

Takaaki Tanaka, Yusuke Miyao, Masayuki Asahara, Sumire Uematsu, Hiroshi Kanayama, Shinsuke Mori, and Yuji Matsumoto. 2016. Universal dependencies for Japanese. In Proceedings of LREC 2016.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajíc, Joakim Nivre, Filip Ginter, Juhan Liutolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gökkırmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajíč jr., Jaroslava Hlavačová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stína Ojala, Anna Missilä, Christopher Manning, Sebastian Schuster, Siva Reddy, Dina Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadova, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisoraj, and Josie Li. 2017. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Association for Computational Linguistics.

Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. 2017. Dependency parsing as head selection. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 665–676.