KWJA: A Unified Japanese Analyzer Based on Foundation Models

Nobuhiro Ueda[1], **Kazumasa Omura**[1], **Takashi Kodama**[1],
Hirokazu Kiyomaru[1], **Yugo Murawaki**[1], **Daisuke Kawahara**[2], **Sadao Kurohashi**[1]
[1] Kyoto University, Kyoto, Japan,
[2] Waseda University, Tokyo, Japan
{ueda,omura,kodama,kiyomaru,murawaki,kuro}@nlp.ist.i.kyoto-u.ac.jp,
dkw@waseda.jp

Abstract

We present KWJA, a high-performance unified Japanese text analyzer based on foundation models. KWJA supports a wide range of tasks, including typo correction, word segmentation, word normalization, morphological analysis, named entity recognition, linguistic feature tagging, dependency parsing, PAS analysis, bridging reference resolution, coreference resolution, and discourse relation analysis, making it the most versatile among existing Japanese text analyzers. KWJA solves these tasks in a multi-task manner but still achieves competitive or better performance compared to existing analyzers specialized for each task. KWJA is publicly available under the MIT license at https://github.com/ku-nlp/kwja.

1 Introduction

End-to-end neural network-based models have become mainstream for many NLP tasks including machine translation (Sutskever et al., 2014; Luong et al., 2015; Vaswani et al., 2017) and dialogue response generation (Serban et al., 2015; Roller et al., 2020). However, end-to-end models are not always the best means of developing an NLP application because exploratory tasks, such as information analysis and causal analysis, inherently require manual trial-and-error processes. We believe that for such tasks, text analysis still plays an important role.

Text analysis, including morphological analysis, dependency parsing, predicate-argument structure (PAS) analysis, and discourse relation analysis, saw shifts in model architectures. Recent studies demonstrate that foundation models (Bommasani et al., 2021) drastically improve the performance in dependency parsing (Zhou and Zhao, 2019), PAS analysis (Ueda et al., 2020; Umakoshi et al., 2021), and discourse relation analysis (Kishimoto et al., 2020; Kiyomaru and Kurohashi, 2021). Moreover, improvements on foundation models tend to have a greater impact on performance than incremental improvements tailored to individual tasks (Bommasani et al., 2021).

In this study, we design and build a unified Japanese text analyzer, KWJA,[1,2] in view of the fact that recent high-performance text analysis models are all based on foundation models. KWJA supports a wide variety of text analysis tasks: typo correction, word segmentation, word normalization, morphological analysis, named entity recognition, linguistic feature tagging, dependency parsing, PAS analysis, bridging reference resolution, coreference resolution, and discourse relation analysis (Figure 1, Table 1).

Our emphasis is on usability in addition to performance. KWJA provides a single command to perform a variety of text analyses, collapsing the painstaking steps previously needed to obtain the same result, namely, installing and combining multiple text analyzers, one for each task.

The design policy of KWJA is to minimize the amount of code and hand-written rules by maximally exploiting the power of foundation models. This is a drastic departure from the traditional Japanese analysis suite, including the morphological analyzers JUMAN (Kurohashi et al., 1994)[3] and Juman++ (Tolmachev et al., 2018)[4] and the dependency parser KNP (Kurohashi and Nagao, 1994),[5] which rely heavily on manually constructed dictionaries, rules, and features. Such lexical knowledge is context insensitive and suffers from limited coverage. Motivated by the observation that foundation models learn massive knowledge through pre-training on large raw corpora, we narrow our efforts to supervised learning from relatively small annotated corpora. This approach enables us to support a new task just by constructing an annotated

[1] Kyoto-Waseda Japanese Analyzer.
[2] Video demo: https://youtu.be/p2x_IrSmS20
[3] https://github.com/ku-nlp/juman
[4] https://github.com/ku-nlp/jumanpp
[5] https://github.com/ku-nlp/knp

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 538–548

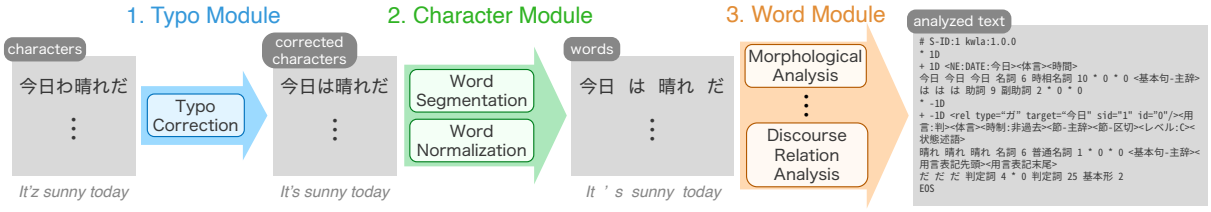

Figure 1: A flowchart of the analysis process in KWJA. KWJA consists of the typo module, character module, and word module. An input text is processed through each module.

Analysis Component	Input	Output Unit	Output
Typo Correction	characters	character	corrected characters
Word Segmentation	corrected characters	word	words
Word Normalization		word	normalized words
Morphological Analysis	words	word	POS, conjugation, lemma, and reading
Named Entity Recognition		word	named entity spans and categories
Linguistic Feature Tagging		word	word features
		base phrase	base phrases features
Dependency Parsing		word	dependency tree and dependency types
		base phrase	dependency tree and dependency types
PAS Analysis		base phrase	predicates and their arguments
Bridging Reference Resolution		base phrase	anaphors and their referents
Coreference Resolution		base phrase	coreferring mentions
Discourse Relation Analysis		clause	discourse relations

Table 1: Input and output of each analysis component. A base phrase is a unit consisting of a single independent word and its ancillary words (Hangyo et al., 2012). We group the components into three blocks (separated by horizontal lines) for multi-task learning.

corpus for the task.

KWJA reorganizes text analysis tasks into a dozen of components, as shown in Table 1, on the ground of task independence and the units of inputs and outputs. A notable difference from conventional morphological analysis is that while it is usually defined as the joint task of word segmentation, part-of-speech (POS) tagging, and the tagging of lexical information, such as lemmas and readings, we divide the task into word segmentation and the remaining tasks. For convenience, we refer to the latter as morphological analysis.

Each analysis component utilizes a fine-tuned Transformer. As Transformer consumes a considerable amount of computational resources, we resort to multi-task learning to reduce the model parameters. We group the components into three modules: the typo module, character module, and word module. Within each module, analysis components share most of their model parameters and run concurrently. Consequently, KWJA executes all the components in three steps (Figure 1).

Although KWJA is extremely slow for users who only need word segmentation, it is the most practical choice for advanced text analysis, for which, after all, only Transformer achieves state-of-the-art performance. KWJA is publicly available under the MIT license at `https://github.com/ku-nlp/kwja`.

2 Related Work

Traditionally, Japanese text analysis tasks have been tackled individually, with separate analyzers for each task. Juman++ (Tolmachev et al., 2018) and Mecab (Kudo et al., 2004) are examples of morphological analyzers, while KNP (Kurohashi and Nagao, 1994) is a dependency parser. Juman++ and Mecab segment Japanese text into words and assign lexical information to each word using manually constructed dictionaries. KNP assigns dependency relations between phrases using linguistic features obtained from Juman++ and external dictionaries. In addition to dependency parsing, KNP handles named entity recognition, linguistic feature tagging, and PAS analysis. KWJA, however, supports an even broader range of tasks.

The Universal Dependencies (UD) project (Nivre et al., 2020) standardizes the

annotation of dependency structures across languages. While their focus is on dependency relations, the UD guidelines define word units, POS tags, and other linguistic features. The tasks supported by major UD-based analyzers, UDPipe (Straka et al., 2016), spaCy,[6] and `Stanza` (Qi et al., 2020), are sentence segmentation, word segmentation, POS tagging, lemmatization, and dependency parsing.[7] In other words, higher-level tasks such as PAS analysis and discourse relation analysis are out of the scope of these analyzers. A major advantage of UD-based analyzers is that they can handle multiple languages. This is done at the expense of ignoring language-specific features (Kanayama et al., 2018). For the purpose of pioneering task design, it is reasonable to focus on a single language.

GiNZA[8] is also a UD-based analyzer but specializes in Japanese. GiNZA supports morphological analysis, dependency parsing, and named entity recognition. GiNZA v5 improved the dependency parsing performance by utilizing the foundation model ELECTRA.

Kachako (Kano, 2013) and Jigg (Noji and Miyao, 2016) have been proposed as frameworks for combining existing analysis tools to form a pipeline. These works aim to improve the usability of existing analysis tools. In contrast, our goal is to design and build a unified analyzer itself.

3 Resources

This section presents the model and data resources used when training the modules in KWJA.

3.1 Foundation Models

As a foundation model, we adopt DeBERTa (He et al., 2021), which has shown high performance in the SuperGLUE language understanding benchmark (Wang et al., 2019). We pre-trained character-level[9] and word-level[10] DeBERTa V2 large models on Japanese texts. The typo and character module employs the character-level model, and the word module employs the word-level model.

[6] `https://spacy.io`

[7] Using extra resources, spaCy and `Stanza` support named entity recognition in some languages.

[8] `https://github.com/megagonlabs/ginza`

[9] `https://huggingface.co/ku-nlp/deberta-v2-large-japanese-char-wwm`

[10] `https://huggingface.co/ku-nlp/deberta-v2-large-japanese`

3.2 Annotated Corpora

We use the Japanese Wikipedia Typo Dataset (JWTD v2, Tanaka et al., 2021) to train a typo correction model. JWTD was created by mining typos from the edit history of Japanese Wikipedia.

We use the Kyoto University Text Corpus (KC, Kurohashi and Nagao, 1998),[11] the Kyoto University Web Document Leads Corpus (KWDLC, Hangyo et al., 2012),[12] and the Annotated Fuman Kaitori Center Corpus (Fuman)[13] to train models for tasks other than typo correction. Note that as for discourse relation analysis, we use only KWDLC because the other two corpora do not have discourse relation annotations.

4 Architecture

Each analysis component of KWJA uses a Transformer-based (Vaswani et al., 2017) foundation model. We add two layers of feed-forward neural networks for each task and fine-tune the whole model.

KWJA formulates the text analysis tasks as a sequence labeling task, word selection task, or word relation classification task. A sequence labeling task assigns a label to each character or word in a text. Figure 2 shows an example of solving named entity recognition as a sequence labeling task. In a word selection task, a word is selected from a text for each given word in the text. Figure 3 shows an example of solving dependency parsing as a word selection task. A word relation classification task assigns a label to each word pair in a text.

To reduce computational time and space, the model parameters of the analyzer are shared as much as possible through multi-task learning. The tasks in each block separated by the horizontal lines in Table 1 are the unit of multi-task learning. Multi-task learning is not possible for the pair of word segmentation and morphological analysis, for example, because the latter's input depends on the former's output. In this study, we perform multi-task learning for tasks in each block. Thus, KWJA consists of three modules corresponding to each block. These modules are referred to as the typo, character, and word modules, respectively.

While morphological analysis and dependency parsing use a sentence as the smallest unit of analysis, PAS analysis and discourse relation analysis

[11] `https://github.com/ku-nlp/KyotoCorpus`

[12] `https://github.com/ku-nlp/KWDLC`

[13] `https://github.com/ku-nlp/AnnotatedFKCCorpus`

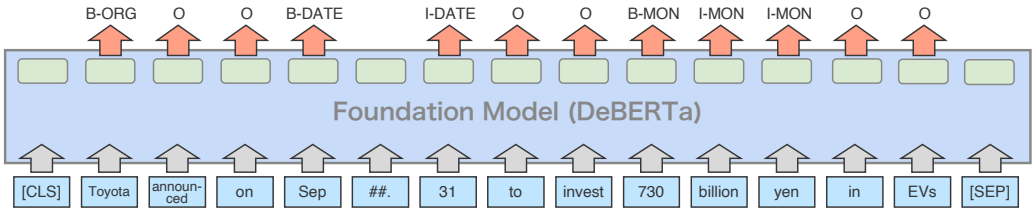

Figure 2: An example of solving named entity recognition as a sequence labeling task.

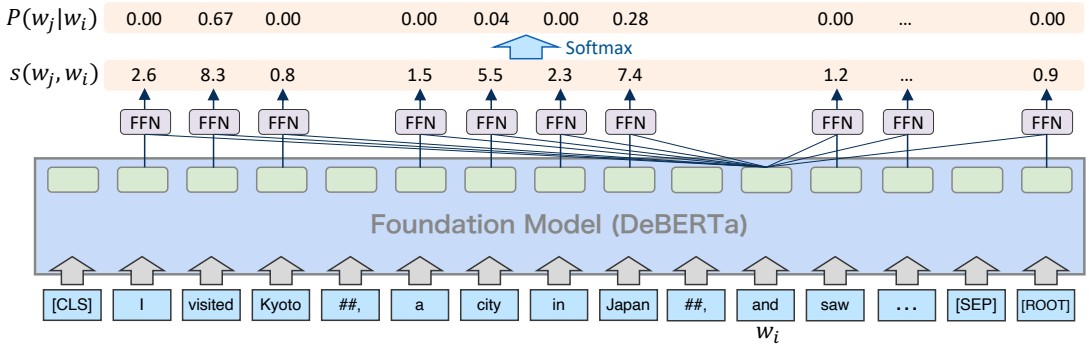

Figure 3: An example of solving dependency parsing as a word selection task.

require document-level processing. In this study, we apply document-level processing for all tasks to perform multi-task learning. With this formulation, obtaining sentence boundaries becomes less important. Thus, we only use simple rules based on regular expressions for sentence segmentation.

5 Analysis Components

KWJA consists of eleven analysis components belonging to three modules. In this section, we describe the details of each component, including its definition, formulation, and evaluation.

5.1 Typo Module

5.1.1 Typo Correction

Typo correction is the task of detecting and correcting typos in a text. Tanaka et al. (2021) used a pre-trained seq2seq model to convert input text to typo-corrected text. While seq2seq models enable flexible typo correction, they are at risk of grossly deviating from the input text.

To reduce the risk, we take a conservative approach; we formulate the task as a sequence labeling task where two edit operations (Malmi et al., 2019) are assigned to each character. Specifically, for each character in an input text, the model (1) chooses one from {`KEEP`, `DELETE`, `REPLACE:`x} and (2) predicts `INSERT:`x. `REPLACE:`x is an operation of replacing the character with x. `INSERT:`x inserts x before the character (x can be null). Following Tanaka et al. (2021), we use the F1 score of character-level minimum edits for evaluation.

5.2 Character Module

5.2.1 Word Segmentation

Word segmentation is the task of splitting a text into words. We formulate the task as a character-level sequence labeling task. This task assigns a B (Begin) or I (Inside) label to each character in a text. The evaluation metric is the F1 score for word spans.

5.2.2 Word Normalization

Word normalization is the task of normalizing non-standard notations such as "Thank youuuuu" in place of "Thank you." As with typo correction, we formulate the task as a sequence labeling task where a normalization operation is assigned to each character. The list of normalization operations is shown in Appendix A. The evaluation metric is the micro-averaged F1 score over labels other than `KEEP`, given that `KEEP` overwhelms the others.

5.3 Word Module

5.3.1 Morphological Analysis

In this study, we refer to morphological analysis as the task of assigning a POS, a sub-POS, a conjugation type, a conjugation form, a lemma, and a reading to each word. We formulate the first four tasks as word-level sequence labeling tasks. A post-

processing code is used to generate the lemma of a word by looking up its normalized surface form, conjugation type, and conjugation form in the conjugation table. We also formulate the task of assigning readings as a sequence labeling task, where the label set is the subword vocabulary defined in a pre-trained model. This is somewhat complicated because we have to preprocess the training data to split a reading into two or more if the corresponding word is split into multiple subwords. To do this, we perform alignment of character sequences of words and readings in accordance with subwords.

5.3.2 Named Entity Recognition

Named entity recognition is the task of identifying named entities in a text. We formulate the task as a word-level sequence labeling task. Following Akbik et al. (2018), we add a CRF layer on top of a foundation model. Named entities have the eight categories defined in the IREX CRL named entity data (Sekine and Isahara, 2000). The evaluation metric is the micro-averaged F1 score over the named entity categories.

5.3.3 Linguistic Feature Tagging

Linguistic feature tagging is the task of assigning various linguistic features to each word or base phrase.[14] We formulate the task as a word-level sequence labeling task. Base phrase linguistic features are assigned to the head word of each base phrase. The evaluation metric is the macro-averaged F1 score over the features.

As the existing corpora do not have manually annotated linguistic features, we assign silver features using a rule-based Japanese linguistic feature tagger, KNP (Kurohashi and Nagao, 1994). In the future, we will manually correct some of the features and use them as gold data. All the features we used are listed in Appendix B.

5.3.4 Dependency Parsing

In this study, dependency parsing consists of two sub-tasks; one recognizes syntactic dependencies between words, and the other identifies their dependency types. We formulate the former task as a word selection task, following Zhang et al. (2017), and the latter task as a word relation classification task. As the evaluation metric, we use the Labeled Attachment Score (LAS) for base phrases.

[14] A unit consisting of a single independent word and its ancillary words. One or more base phrases make up a phrase.

5.3.5 PAS Analysis, Bridging Reference Resolution, and Coreference Resolution

PAS analysis, bridging reference resolution, and coreference resolution are the tasks of recognizing semantic relations between base phrases. PAS analysis finds arguments corresponding to *who* did/does *what* to *whom* for a predicate. Bridging reference resolution finds nouns that complement the essential information of another noun. Coreference resolution finds a set of nouns that refer to the same real-world entity.

Following Ueda et al. (2020), we formulate all the tasks as a word selection task. In PAS analysis, we focus on four cases: nominative, accusative, dative, and nominative-2.[15]

5.3.6 Discourse Relation Analysis

Discourse relation analysis is the task of recognizing discourse relations between clauses. Following Kawahara et al. (2014), we assign a label to each clause pair in a text. Note that clauses are identified with linguistic feature tagging.

We target the following discourse relations: CAUSE/REASON, PURPOSE, CONDITION, JUSTIFICATION, CONTRAST, and CONCESSION. In addition, we introduce a special relation NORELATION, which indicates none of the above relations is applicable, and formulate the task as a seven-way word relation classification task. We use the micro-averaged F1 score of the labels other than NORELATION as the evaluation metric.

6 Experiments and Discussion

In this section, we investigate the performance of each analysis component through fine-tuning foundation models.

6.1 Experimental Settings

We trained models on all the training data of KC, KWDLC, and Fuman, and evaluated the performance per corpus. The details of the experimental settings are described in Appendix C.

6.2 Results

The result of each task is shown in Table 2. Overall, the performance of KWJA was comparable to SOTA and was sufficient for practical use. However, the F1 score of word normalization was 33.3, which was remarkably lower than those of the other

[15] Nominative-2 is used for a common Japanese construction in which a predicate has two nominative arguments.

Task		Corpus	Metric	Reference	SOTA	KWJA
Typo Correction		JWTD	F1	Tanaka et al. (2021)	77.6	83.1±0.3
Word Segmentation		KC	F1	Tolmachev et al. (2020)	99.5	99.3±0.1
Word Normalization		all	F1	—	—	33.3±0.0
Morphological Analysis	POS	KC	F1	Tolmachev et al. (2020)	99.1	99.7±0.1
	sub-POS	KC	F1	Tolmachev et al. (2020)	97.8	99.0±0.1
	conjugation type	all	F1	—	—	99.3±0.3
	conjugation form	all	F1	—	—	99.5±0.2
	reading	all	Accuracy	—	—	95.8±0.7
Named Entity Recognition		all	F1	—	—	84.3±4.0
Linguistic Feature Tagging	word	all	F1	—	—	98.6±0.1
	base phrase	all	F1	—	—	88.3±3.1
Dependency Parsing		KC	LAS	Kawahara and Kurohashi (2006)	90.4	92.7±0.4
PAS Analysis		all	F1	Ueda et al. (2020)	77.4	75.9±1.5
Bridging Reference Resolution		all	F1	Ueda et al. (2020)	64.3	65.8±1.6
Coreference Resolution		all	F1	Ueda et al. (2020)	67.4	77.7±0.9
Discourse Relation Analysis		KWDLC	F1	Omura and Kurohashi (2022)	51.9	41.7±0.9

Table 2: The performance of KWJA on each task in comparison to the state-of-the-art (SOTA). We fine-tuned KWJA with 3 different random seeds and report the mean and standard deviation of the performance. "—" indicates that no previous studies reported the performance on the corpora we used. "all" indicates KC, KWDLC, and Fuman corpus, and the metric is the macro-average of them.

tasks. Moreover, PAS analysis and discourse relation analysis scores were more than 1 point lower than SOTA.

6.3 Discussion

We discuss word normalization, PAS analysis, and discourse relation analysis in the following sections. Section 6.3.3 compares the analysis speed of KWJA with existing Japanese analyzers.

6.3.1 Word Normalization

The F1 score of word normalization, 33.3, was strinkingly low. We attribute the poor performance to the highly unbalanced label distribution. Word normalization mainly targets informal texts, and there were very few examples with labels other than `KEEP` in the annotated corpora. We generated pseudo training data by applying denormalization rules to randomly selected words. Even with the pseudo-data, the percentage of labels other than `KEEP` was less than 0.1%, however. We plan to expand training data by specifically targeting low-frequency phenomena.

Analyzer	Time
Juman++ & KNP	1.1min
Juman++ & KNP (w/ PAS analysis)	18.4min
KWJA (ours)	2.7min

Table 3: Time spent by KWJA to analyze 1k sentences, with comparison to Juman++ (Tolmachev et al., 2018) and KNP (Kurohashi and Nagao, 1994).

6.3.2 PAS Analysis and Discourse Relation Analysis

We hypothesized that the low performance of PAS analysis and discourse relation analysis was due to multi-task learning, in which the model's capability was allocated to the other tasks. To test this hypothesis, we trained the model separately for each task using single-task learning. The F1 score of PAS analysis was 79.3±1.0, and that of discourse relation analysis was 55.3±3.6. Both scores were significantly higher, confirming the hypothesis. We plan to adjust the loss weights and provide an option to use models trained with single-task learning. Appendix D shows the results of single-task learning for all tasks in the word module.

6.3.3 Speed of Analysis

We compared KWJA with the existing Japanese analyzers, Juman++ (Tolmachev et al., 2018) and KNP (Kurohashi and Nagao, 1994), in terms of speed of analysis. For KWJA, we performed all the tasks it supports. Juman++ supports word segmentation and morphological analysis while KNP supports named entity recognition, linguistic feature tagging, dependency parsing, and optionally, PAS analysis. We used 1k sentences randomly sampled from the Japanese portion of the CC-100 corpus (Wenzek et al., 2020). We used an NVIDIA TITAN V 12GB GPU to run KWJA.

Table 3 shows the results. We can see that KWJA was considerably faster than Juman++ and KNP even though KWJA performed a larger number of tasks.

7 Conclusion

In this study, we designed and built a unified Japanese text analyzer, KWJA, on top of foundation models. KWJA supports typo correction, word segmentation, word normalization, morphological analysis, named entity recognition, linguistic feature tagging, dependency parsing, PAS analysis, bridging reference resolution, coreference resolution, and discourse relation analysis in a unified framework. Users can quickly obtain analysis results by inputting a text and specifying the desired level of analysis.

One of the advantages of KWJA is its simplified design, thanks to the use of foundation models. Various analysis tasks, previously solved separately, are now performed only with three modules. For further simplification, we plan to solve all the analysis tasks with a character-level foundation model.

Limitations

As KWJA is based on a large Transformer model, the analysis in an environment without GPUs is expected to be slow. Even in environments with GPUs, when we need only a specific task (e.g., word segmentation), existing analyzers might be faster with little difference in accuracy.

The experiments showed that multi-task learning decreased accuracy in PAS analysis and discourse relation analysis. This fact may be true for other tasks as well. Therefore, when very high analysis accuracy is required for a particular task, using a model trained only on that task is recommended instead of KWJA.

Acknowledgements

The pre-training of foundation models used in this work was supported by the Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures (JHPCN) through General Collaboration Project no. jh221004 and jh231006, "Developing a Platform for Constructing and Sharing of Large-Scale Japanese Language Models." As for the computing environment, we used the mdx: a platform for the data-driven future.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the Opportunities and Risks of Foundation Models.

Masatsugu Hangyo, Daisuke Kawahara, and Sadao Kurohashi. 2012. Building a Diverse Document

Leads Corpus Annotated with Semantic Relations. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 535–544, Bali, Indonesia. Faculty of Computer Science, Universitas Indonesia.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-Enhanced BERT with Disentangled Attention. In *International Conference on Learning Representations*.

Hiroshi Kanayama, Na-Rae Han, Masayuki Asahara, Jena D. Hwang, Yusuke Miyao, Jinho D. Choi, and Yuji Matsumoto. 2018. Coordinate Structures in Universal Dependencies for Head-final Languages. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 75–84, Brussels, Belgium. Association for Computational Linguistics.

Yoshinobu Kano. 2013. Kachako: A Hybrid-Cloud Unstructured Information Platform for Full Automation of Service Composition, Scalable Deployment and Evaluation - Natural Language Processing as an Example. In *Service-Oriented Computing - ICSOC 2012 Workshops - ICSOC 2012, International Workshops ASC, DISA, PAASC, SCEB, SeMaPS, WESOA, and Satellite Events, Shanghai, China, November 12-15, 2012, Revised Selected Papers*, volume 7759 of *Lecture Notes in Computer Science*, pages 72–84. Springer.

Daisuke Kawahara and Sadao Kurohashi. 2006. A Fully-Lexicalized Probabilistic Model for Japanese Syntactic and Case Structure Analysis. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 176–183, New York City, USA. Association for Computational Linguistics.

Daisuke Kawahara, Yuichiro Machida, Tomohide Shibata, Sadao Kurohashi, Hayato Kobayashi, and Manabu Sassano. 2014. Rapid Development of a Corpus with Discourse Annotations using Two-stage Crowdsourcing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 269–278, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Yudai Kishimoto, Yugo Murawaki, and Sadao Kurohashi. 2020. Adapting BERT to Implicit Discourse Relation Classification with a Focus on Discourse Connectives. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1152–1158, Marseille, France. European Language Resources Association.

Hirokazu Kiyomaru and Sadao Kurohashi. 2021. Contextualized and Generalized Sentence Representations by Contrastive Self-Supervised Learning: A Case Study on Discourse Relation Analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5578–5584, Online. Association for Computational Linguistics.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237, Barcelona, Spain. Association for Computational Linguistics.

Sadao Kurohashi and Makoto Nagao. 1994. KN Parser: Japanese Dependency/Case Structure Analyzer. In *Proceedings of the International Workshop on Sharable Natural Language Resources*, pages 48–55.

Sadao Kurohashi and Makoto Nagao. 1998. Building a Japanese Parsed Corpus while Improving the Parsing System. In *Proceedings of the NLPRS*, pages 719–724.

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese Morphological Analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language Resources*, pages 22–38.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *ArXiv*, abs/1508.04025.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, Tag, Realize: High-Precision Text Editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Hiroshi Noji and Yusuke Miyao. 2016. Jigg: A Framework for an Easy Natural Language Processing Pipeline. In *Proceedings of ACL-2016 System Demonstrations*, pages 103–108, Berlin, Germany. Association for Computational Linguistics.

Kazumasa Omura and Sadao Kurohashi. 2022. Improving Commonsense Contingent Reasoning by Pseudo-data and Its Application to the Related Tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 812–823, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics: System Demonstrations, pages 101–108, Online. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric Michael Smith, Y.-Lan Boureau, and Jason Weston. 2020. Recipes for Building an Open-Domain Chatbot. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Satoshi Sekine and Hitoshi Isahara. 2000. IREX: IR & IE Evaluation Project in Japanese. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2015. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *AAAI Conference on Artificial Intelligence*.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. *Advances in neural information processing systems*, 27.

Yu Tanaka, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. 2021. Building a Japanese Typo Dataset and Typo Correction System Based on Wikipedia's Revision History. *Journal of Natural Language Processing*, 28(4):995–1033. (in Japanese).

Arseny Tolmachev, Daisuke Kawahara, and Sadao Kurohashi. 2018. Juman++: A Morphological Analysis Toolkit for Scriptio Continua. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 54–59, Brussels, Belgium. Association for Computational Linguistics.

Arseny Tolmachev, Daisuke Kawahara, and Sadao Kurohashi. 2020. Design and Structure of The Juman++ Morphological Analyzer Toolkit. *Journal of Natural Language Processing*, 27(1):89–132.

Nobuhiro Ueda, Daisuke Kawahara, and Sadao Kurohashi. 2020. BERT-based Cohesion Analysis of Japanese Texts. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1323–1333, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Masato Umakoshi, Yugo Murawaki, and Sadao Kurohashi. 2021. Japanese Zero Anaphora Resolution Can Benefit from Parallel Texts Through Neural Transfer Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1920–1934, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. 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: Volume 1, Long Papers*, pages 665–676, Valencia, Spain. Association for Computational Linguistics.

Junru Zhou and Hai Zhao. 2019. Head-Driven Phrase Structure Grammar Parsing on Penn Treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408, Florence, Italy. Association for Computational Linguistics.

A Word Normalization Operations

We define six types of normalization operations as follows:

KEEP Keep the original character.

DELETE Delete the character.

VOICED Replace voiced or semi-voiced character with voiceless character (e.g., “ が ” (*ga*) → “ か ” (*ka*), “ ぱ ” (*pa*) → “ は ” (*ha*)). This reverts *rendaku*, the voicing of the initial consonant of a non-initial word of a compound.

SMALL Replace a small character with the large character (e.g., “ なぁ ” → “ なあ ”).

PROLONG Replace prolonged sound mark with its equivalent hiragana or katakana (e.g., “ もーれつ ” (*moRretsu*) → “ もうれつ ” (*mouretsu*)).

PROLONG-E Replace a prolonged sound mark with “ え ” (*e*) (e.g., “ ねー ” (*neR*) → “ ねえ ” (*nee*)).

B Word and Base Phrase Features

KWJA assigns the following linguistic features.[16,17] † indicates that the feature is to be corrected manually in the future.

Word Features

- base phrase head†
- base phrase end†, phrase end†
- declinable head or end

Base Phrase Features

- verbal (verb, adjective, copula)†, nominal†
- stative predicate, active predicate
- nominal predicate (verb, adjective)†
- modality†, tense†, negation†, potential expression, honorific, time
- modification
- SM-subject
- verbal level
- dependency:genitive
- clause head†, clause end†,
  clause end:adnominal,
  clause end:complement, clause functional

[16] https://github.com/ku-nlp/knp/blob/master/doc/knp_feature.pdf

[17] https://github.com/ku-nlp/KWDLC/blob/master/doc/clause_feature_manual.pdf

C Training Details

We trained each module with hyper-parameters shown in Table 4. During training, we evaluated a score averaged over tasks on the validation set at the end of each epoch and picked the model with the highest score. When training the word module, the ground-truth word segmentation was used as input. We trained each module three times with different random seeds. Single training runs of the typo, character, and word modules took 38 hours, 2.5 hours, and 5.8 hours on four Tesla V100-SXM2-32GB GPUs, four TITAN X 12GB GPUs, and two Tesla V100-SXM2-32GB GPUs, respectively. The `transformers` package (Wolf et al., 2020) was used for implementation.

D Single-task Learning Results

Table 5 shows the results of single-task learning. We trained each task in the word module separately in a single-task manner. Note that the training of *POS*, *sub-POS*, *conjugation type*, and *conjugation form* tasks was performed in a multi-task manner as before because these tasks had already achieved enough performance.

Hyper-parameter	Typo Module	Character Module	Word Module
Maximum Sequence Length	256	512	256
Dropout	0.1	0.1	0.1
Batch Size	352	32	16
Maximum Training Epochs	20	20	20
Early Stopping Patience	3	3	3
Warmup Steps	1k	2k	100
Maximum Learning Rate	2e-5	2e-5	1e-4
Learning Rate Decay	Cosine	Cosine	Cosine
Optimizer	AdamW	AdamW	AdamW
AdamW ϵ	1e-6	1e-6	1e-6
AdamW β_1	0.9	0.9	0.9
AdamW β_2	0.99	0.99	0.99
Weight Decay	0.01	0.01	0.01
Gradient Clipping	0.5	0.5	0.5

Table 4: Hyper-parameters for training each module.

Task		Corpus	Metric	KWJA (multi)	KWJA (single)
Morphological Analysis	POS	all	F1	99.4±0.1	99.4±0.1
	sub-POS	all	F1	98.7±0.1	98.7±0.0
	conjugation type	all	F1	99.3±0.3	99.5±0.0
	conjugation form	all	F1	99.5±0.2	99.6±0.0
	reading	all	Accuracy	95.8±0.7	96.2±0.0
Named Entity Recognition		all	F1	84.3±4.0	77.9±4.2
Linguistic Feature Tagging	word	all	F1	98.6±0.1	98.5±0.1
	base phrase	all	F1	88.3±3.1	92.4±0.1
Dependency Parsing		all	LAS	93.6±0.3	93.5±0.3
PAS Analysis		all	F1	75.9±1.5	79.3±1.0
Bridging Reference Resolution		all	F1	65.8±1.6	65.2±1.6
Coreference Resolution		all	F1	77.7±0.9	77.6±1.2
Discourse Relation Analysis		KWDLC	F1	41.7±0.9	55.3±3.6

Table 5: The performance of KWJA on each task in single-task learning (**single**) compared to that in multi-task learning (**multi**). We fine-tuned KWJA with three different random seeds. We report the mean and standard deviation of the performance. "all" indicates KC, KWDLC, and Fuman corpus, and the metric is the macro-average of them.