Multilingualization of Question Answering Using Universal Dependencies

Hiroshi Kanayama
IBM Research - Tokyo
Tokyo, Japan
hkana@jp.ibm.com

Koichi Takeda
Nagoya University
Nagoya, Japan
takedasu@i.nagoya-u.ac.jp

ABSTRACT

This paper investigates the capability of the syntax representation framework of Universal Dependencies to build multilingual applications. In a multilingual question answering system, the dependency structures are used as the input to the downstream components that can be shared for multiple languages. The experiment on the question answering pipeline demonstrates that the dependency structures commonly designed for multiple languages work better than conventional language-dependent representations, even for the Japanese language which has very different structures from those of English and Spanish.

KEYWORDS

Question answering, Dependency parsing, Universal dependencies

1 INTRODUCTION

Universal Dependencies (UD) [14] project aims to design and provide consistent treebanks for many languages, through the implementation of multilingual dependency parsers, cross-lingual transfer learning, and quantitative comparison of languages from linguistic viewpoints [10, 12]. As of the end of 2016, treebanks of 49 languages have been released [13].

Figure 1 shows the notions of data and process in typical existing studies using Universal Dependencies. First, several works tried to create UD treebanks by converting the existing treebanks of various languages, such as Russian [9], Swedish [1] and Estonian [11].

For low-resource languages, several methods of cross-lingual transfer learning have been studied, relying on richer resources in other languages, such as for part-of-speech tagging [18] and dependency parsing [5, 7, 17]. These studies were evaluated by comparing the accuracy of part-of-speech tagging and parsing with the treebanks based on Universal Dependencies.

However, there has been little work on evaluating the appropriateness of multilingual dependency representation using Universal Dependencies on multilingual downstream applications. Particularly for the Japanese language, it is still an open problem whether the Japanese dependency structures represented by Universal Dependencies are actually useful compared to conventional syntactic frameworks. Another potential issue is that the performance on the UD treebank and its usefulness for application may be different; for

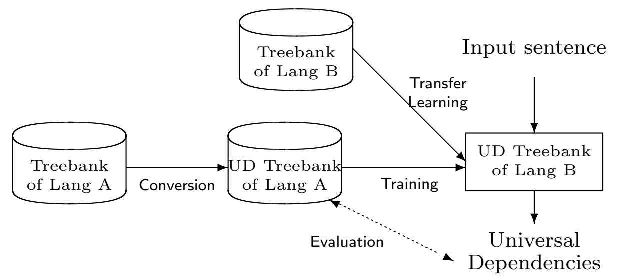


Figure 1: Typical existing studies on Universal Dependencies using languages A and B.

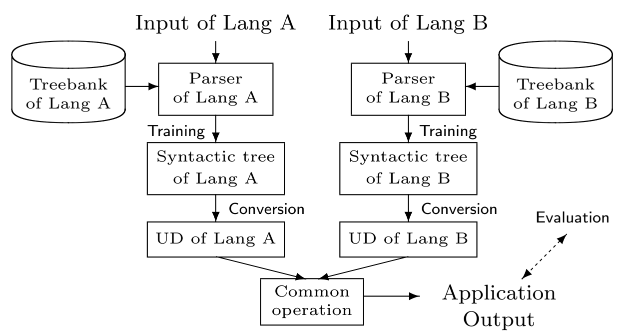


Figure 2: Concepts of a multilingual application that uses UD syntactic structure discussed here.

example, if a parser is tuned for the UD corpus, it may reduce the quality of application.

This paper discusses the advantage of uniform multilingual dependency structures from the viewpoint of applications, rather than evaluating the parsers of many languages themselves. As shown in Figure 2, the effect of using Universal Dependencies as a representation of syntactic structure are evaluated, by converting the multilingual dependency structure into UD representation and examining the output on a multilingual downstream component that takes the UD structure as input and applies common algorithms.

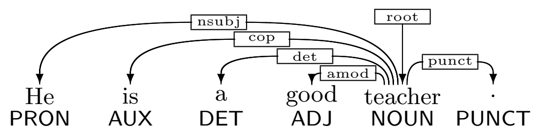
A question answering (QA) system designed for English and Spanish is used as a case study. Multiple types of representation of Japanese syntactic structures will be evaluated on this multilingual QA system, to see whether the UD can be used for a Japanese version of QA without having to implement language-specific downstream components.

Table 1: 17 PoS tags used in Universal PoS. ∗ denotes a PoS for content words.

NOUN ∗	ADV ∗	CCONJ	PART	X ∗
PROPN ∗	PRON	SCONJ	PUNCT	
VERB ∗	NUM ∗	DET	SYM	
ADJ ∗	AUX	ADP	INTJ	

2 UNIVERSAL DEPENDENCIES

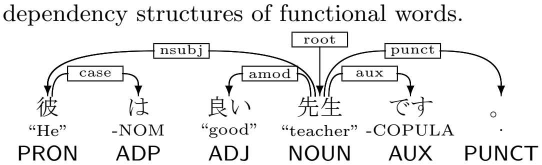
In the Universal Dependencies framework, a dependency structure is represented as in English example (1). Every word except for the root depends on another word, so a whole sentence forms a single tree.

(1)


Representing only the dependencies between two words, with no regard for constituent structures, UD simplifies the tree structure, thus reducing the cost in creating treebanks. It is also robust for informal writing and ellipses.

To make the PoS system uniform across languages, the 17 PoS tags shown in Table 1 based on Google Universal Part-of-Speech Tags [15] are used. Each dependency is classified into 37 labels based on 42 labels originally defined in Universal Stanford Dependencies [4].

Rather than handling classical syntactic relationship such as agreement between a verb and its subject, UD focuses on relationships between content words in order to absorb the syntactic differences in many languages. A typical example is a copula in (1). Unlike most of the classical syntactic frameworks, which regard 'be' as the root of the sentence, and 'he' and 'teacher' as a subject and a complement of 'be' respectively, UD picks up 'teacher' as the root of the sentence, and directly connects 'he' and 'teacher'. This makes it possible to obtain a closer structure between most of the European languages with copula and languages like Russian, which do not have copula. The Japanese UD structure (2) which corresponds to (1) shows that both languages have the same root 'teacher', and two relations between content words are aligned: 'he - nsubj - teacher' and 'good - amod - teacher', even though there are differences in the PoS tags and dependency structures of functional words.

(2)


In spite of the philosophy of UD to give common representation for any language as described in Section 2, there are many open issues in UD design for Japanese [16]. For example, Japanese does not have the syntactic notion of nsubj, obj, iobj, so it is not easy to attach those labels to the arguments of a verb and an adjective. Also it is difficult to draw a line between acl and amod because the attributive adjective behaves like a relative clause in Japanese.

The next two sections discuss how to apply the Japanese UD structure to the common downstream components without having to worry about intrinsic inconsistency of syntactic structures.

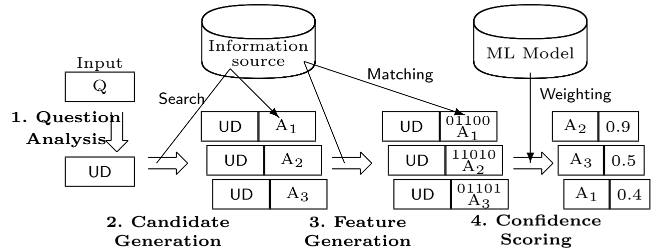


Figure 3: The flow of multilingual factoid QA.

3 MULTILINGUALIZATION OF A QA SYSTEM

As a case study of UD application, a multilingual question answering system is adapted for an additional language. QA is selected because it is one of applications that benefit from multilingual information sources. The open domain factoid question answering system for English, DeepQA [6], has been redesigned to accept Spanish and other European languages [3] based on Universal Dependencies as common syntactic representation.

Figure 3 shows the flow of question answering discussed in this paper. To realize its multilingualization, language dependent operations are consolidated into the question analysis part, and the downstream components will be designed for many languages. Here is the simplified pipeline for multilingual QA:

1. **Question analysis** parses the input question and convert the parse tree into the UD structure (denoted as UD in Figure 3), and then obtains the type of the answer.
2. **Candidate answer generation** searches on the documents stored in the information source using the words extracted from the input question as the query, and then enumerates the titles of the documents and anchor links in the documents as candidate answers. The search query is generated by enumerating the content words (see Table 1) in the UD tree. Using Wikipedia as the information source, it is possible to use the common logic for this component since it has uniform structures for any language.
3. **Feature generation** calculates the multiple similarity values between the information source and question filled by each candidate answer, referring to the passages obtained by secondary search from the information source. Those values are then used as features of each candidate answer. For the calculation, PoS tags

and relation labels of UD structures are used to detect content words and phrases.

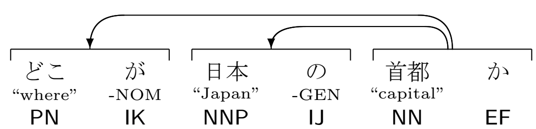
4. **Confidence scoring** applies logistic regression to weight the features generated in the previous component, using the training data consisting of pairs of a question and correct and wrong answers. Then the confidence value is calculated for each candidate answer as the inner product of a feature vector and weights of a feature, and the candidate answer with highest confidence will be selected as the output of the system. This process is completely language independent.

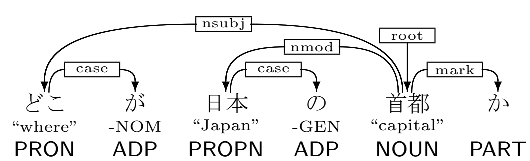
This approach has enabled the question answering for English and Spanish. Now the question is whether the same approach is valid for Japanese which has very different syntactic structures and units of words. Section 4 discusses the capability of multilingualization using the UD framework.

4 USAGE OF JAPANESE UD

4.1 Conversion of Parse Tree

To obtain a dependency structure compatible with the Japanese UD definition [16], we convert the phrase-level output of the Japanese syntactic parser [8] into the word-level dependency structures. For instance, a dependency structure (3) ("What is the capital of Japan?") is converted into the UD format (4).

(3)


(4)


The UD structure is used as the output of the question analysis component of the QA system in Figure 3, and it is consumed in the downstream multilingual components.

4.2 Experiment

To determine the effectiveness of the UD-compliant structure as an input to multilingual components, we ran the whole pipeline of the QA system with varying dependency structures. For the evaluation and training, an existing set of open domain factoid questions were translated into Japanese as in Table 2. Japanese Wikipedia articles were used as the information source. For simplicity, language dependent features have been removed from the feature generation part, though there may be improved quality for each language.

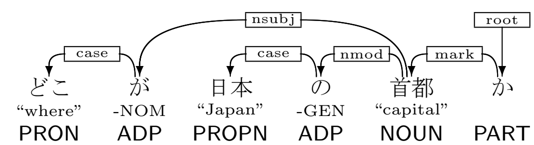
As the output of the question analysis component, we considered, by artificial conversion of labels and dependency structures, the following syntactic structures.

Table 3: QA performance with 220 test questions.

	Recall	Accuracy
(A) UD Compliant	67.3%	15.0%
(B) Conventional dependency	62.7%	10.5%
(C) Without relation labels	62.7%	10.0%
(D) Without PoS tags	58.6%	14.5%
(E) Randomized PoS	34.1%	2.7%
(F) Search only	41.3%	3.6%

(A) UD compliant. A syntactic structure compatible with UD definition as exemplified by (4). It is converted from the original Japanese parsing structure as (3).

(B) Conventional dependency structure. A word-level dependency structure in which all dependencies have right-head direction as (5). UD-style labels are assigned to relations, though some of them have opposite dependency direction from the UD definition.

(5)


(C) Without relation labels. Use only dep (default value) as the relation labels for all dependencies.

(D) Without PoS tags. Use only X (default value) as the PoS tags for all words.

(E) Randomized PoS. Randomly assign the 17 PoS tags in UD definition.

Table 3 shows recall (the ratio at which the correct answer appeared in the top 100 candidates) and accuracy (the ratio at which the answer with the highest confidence value was correct) in the QA system, with using (A) to (E) above in the question analysis component. 'Search only' is the baseline method that naively searches the Wikipedia articles and outputs the title of the most relevant document. Its low accuracy indicates that there are few questions that can be solved trivially. The following discussion focuses on relative performance among various syntactic representations rather than absolute value of the quality, since it highly depends on the complexity of the questions and coverage of the information source.

By converting into the common syntax structure as (A), the whole system worked well enough without implementing any language-specific components in the pipeline. When a different form of dependency structures was used as (B), or when relation labels were missing as (C), recall was decreased because the selection or weighting of the words and phrases for search query were not optimized. Also the accuracy became lower in (B) and (C) because the coincidence of dependencies between two content words in the feature generation was not captured [1].

[1]For example, the relationship between 'Japan' and 'capital' can be obtained in (4), but not in (5).

Table 2: Example of questions and answers.

en	Which country was admitted into the World Trade Organization in August 2012?	Russia (Vanuatu)
ja	2012 年 8 月に世界貿易機関への加盟が承認された国はどこか？	ロシア (バヌアツ)

When all PoS tags were replaced by X in (D), all words were regarded as content words and the recall was reduced due to the noises in the query, but as long as the correct dependency structures were captured, the correct answer could obtain a higher confidence value, so the loss of the accuracy was limited. When the PoS tag was randomized, the content words to build the search query were not correctly obtained, so both the recall and accuracy were drastically reduced.

5 CONCLUSION

This paper examined the contribution of Universal Dependencies to the design of multilingual application. Simply by providing UD based syntactic structures in each language, whole QA pipeline worked, since the downstream component was appropriately generalized to use the syntactic structure to generate search queries and to compare the question and search results within the language.

In this study only language-independent features are used with separated information source by languages. By combining deeper common structure such as universal semantic role label [2], the QA is expected to be enhanced using cross-lingual information source.

If more applications to be evaluated on multiple languages are identified the effectiveness of the universal syntactic structure can be estimated quantitatively. This will enormously help the design of Universal Dependencies, which will be of great benefit to multilingual applications.

REFERENCES

[1] Lars Ahrenberg. 2015. Converting an English-Swedish Parallel Treebank to Universal Dependencies. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*. 10–19.

[2] Alan Akbik and Yunyao Li. 2016. Polyglot: Multilingual semantic role labeling with unified labels. In *Proceedings of ACL 2016*.

[3] Keith Cortis, Urvesh Bhowan, Ronan Mac an tSaoir, D.J. McCloskey, Mikhail Sogrin, and Ross Cadogan. 2014. What or Who is Multilingual Watson?. In *Proceedings of COLING 2014: System Demonstrations*. 95–99.

[4] Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of LREC*. 4585–4592.

[5] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. 845–850.

[6] D. A. Ferrucci. 2012. Introduction to "This is Watson". *IBM Journal of Research and Development* 56, 3.4 (2012), 1:1–1:15.

[7] Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2016. A Universal Framework for Inductive Transfer Parsing across Multi-typed Treebanks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 12–22.

[8] Hiroshi Kanayama, Kentaro Torisawa, Yutaka Mitsuishi, 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*. 411–417.

[9] Janna Lipenkova and Milan Souček. 2014. Converting Russian Dependency Treebank to Stanford Typed Dependencies Representation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics(EACL)*. 143–147.

[10] Ryan T McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal Dependency Annotation for Multilingual Parsing.. In *ACL (2)*. 92–97.

[11] Kadri Muischnek, Kaili Müürisep, and Tiina Puolakainen. 2016. Estonian Dependency Treebank: from Constraint Grammar tagset to Universal Dependencies. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France.

[12] Joakim Nivre. 2015. Towards a Universal Grammar for Natural Language Processing. In *Computational Linguistics and Intelligent Text Processing*. Springer, 3–16.

[13] Joakim Nivre, Željko Agić, Lars Ahrenberg, et al. 2017. Universal Dependencies 2.0 — CoNLL 2017 Shared Task Development and Test Data. (2017). http://hdl.handle.net/11234/1-2184 LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

[14] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, 1659–1666.

[15] Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of LREC*.

[16] Takaaki Tanaka, Yusuke Miyao, Masayuki Asahara, Sumire Uematsu, Hiroshi Kanayama, Shinsuke Mori, and Yuji Matsumoto. 2016. Universal Dependencies for Japanese. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016*.

[17] Jörg Tiedemann. 2015. Cross-Lingual Dependency Parsing with Universal Dependencies and Predicted PoS Labels. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*. 340–349.

[18] Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. 2014. Cross-Lingual Part-of-Speech Tagging through Ambiguous Learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.