A Universal Dependencies Corpora Maintenance Methodology Using Downstream Application

Ran Iwamoto*, **Hiroshi Kanayama**†, **Alexandre Rademaker**†‡, **Takuya Ohko**†
* Keio University, † IBM Research, ‡ FGV/EMAp
raniwamoto@gmail.com, hkana@jp.ibm.com
alexrad@br.ibm.com, ohkot@jp.ibm.com

Abstract

This paper investigates updates of Universal Dependencies (UD) treebanks in 23 languages and their impact on a downstream application. Numerous people are involved in updating UD's annotation guidelines and treebanks in various languages. However, it is not easy to verify whether the updated resources maintain universality with other language resources. Thus, validity and consistency of multilingual corpora should be tested through application tasks involving syntactic structures with PoS tags, dependency labels, and universal features. We apply the syntactic parsers trained on UD treebanks from multiple versions (2.0 to 2.7) to a clause-level sentiment extractor. We then analyze the relationships between attachment scores of dependency parsers and performance in application tasks. For future UD developments, we show examples of outputs that differ depending on version.

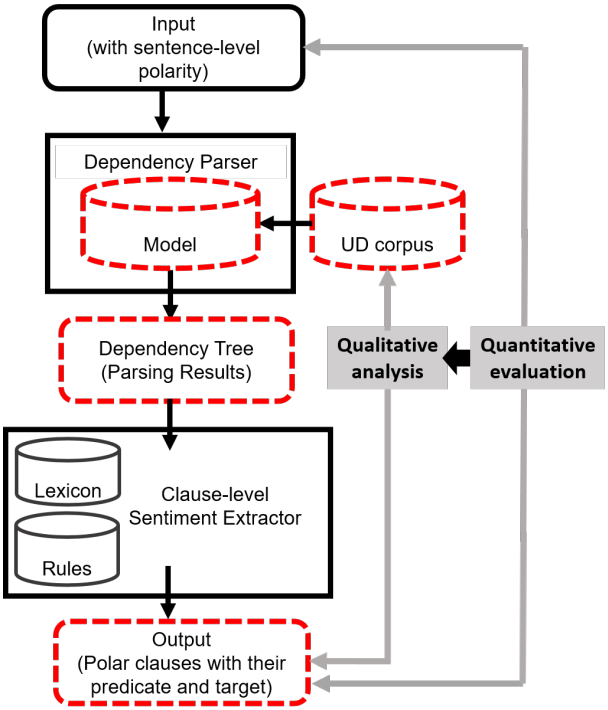


Figure 1: Our methodology to get insights from the difference of the corpora on the flow of the multilingual sentiment annotator. The components in red dashed lines are variable, while solid ones are fixed.

1 Introduction

Universal Dependencies (UD) (Nivre and Fang, 2017; Zeman et al., 2020) is a worldwide project that provides cross-linguistic treebank annotations. UD defined 17 PoS tags and 37 dependency labels to annotate multilingual sentences in a uniform manner, allowing language-specific extension to be represented by features. The resources and documents are updated every six months. The latest version 2.7, as of November 2020, consists of 183 treebanks in 104 languages.

The UD corpora are consistently annotated in multiple languages and are extensively used to train and evaluate taggers and parsers (Zeman et al., 2017). Kondratyuk and Straka (2019) trained a single dependency model for many languages relying on UD corpora. Schwenk and Douze (2017) used universal PoS (UPOS) labels to evaluate multilingual sentence representations. However, few studies have focused on the contributions of syntactic parsers trained by UD corpora to real-world applications.

Extrinsic evaluation of dependency parser has been already studied in a series of shared tasks (Oepen et al., 2017; Fares et al., 2018) using tasks of event extraction, negation detection and opinion analysis for English documents. In addition to extrinsic evaluation of parsers, Kanayama and Iwamoto (2020) established a method for evaluating the universality of UD-based parsers and UD corpora by using a clause-level sentiment extractor, which detects positive and negative predicates and targets on top of UD-based syntactic trees. They showed that language-universal syntactic structures

Proceedings of the Third Workshop on Computational Typology and Multilingual NLP, pages 23–31

lang	name	# sentence
ar	PADT	7,664
ca	AnCora	16,678
cs	PDT	87,913
de	GSD	15,590
en	EWT	16,622
es	Ancora	17,680
fa	Seraji	5,997
fr	GSD	16,341
he	HTB	6,216
hi	HDTB	16,647
hr	SET	9,010
id	GSD	5,593
it	ISDT	14,167
ja	GSD	8,071
ko	GSD	6,339
nl	Alpino	13,578
no	Bokmaal	20,044
pl	LFG	17,246
pt	Bosque	9,364
ru	SynTagRus	61,889
sv	Talbanken	6,026
tr	IMST	5,635
zh	GSD	4,997

Table 1: UD corpora used in this study and their sizes. Sizes are based on sentence numbers in v2.7.

and features are effective in their multilingual systems.

In this paper we investigate how the UD corpora and underlying guidelines are updated and how they contribute to the parser and sentiment extractor which consumes the output of the parser. We compared UD versions 2.0 to 2.7 [1] in 23 languages from diverse language families.

Figure 1 shows the proposed methodology. The idea is to use corpora with sentence-level sentiment annotations (SA) in two ways: 1) we can compare SA results considering different syntactic models; 2) we can compare the SA annotation with the golden sentiment annotation. The first one is useful for qualitative analysis. The second one is useful for quantitative analysis, given that we can measure the SA efficiency.

First, we trained a dependency parsing model for each UD corpora version (UD release) using a fixed syntactic parser. Using the models, we produce as many syntactic analyses as models for each corpus with sentence-level sentiment annotations. Later, we applied a deterministic rule-based sentiment annotator for each syntactic tree. The advantage of this methodology is that it is much easier to find sentiment annotation errors than syntactic annotation errors, and those errors often show the essential aspects of syntax. Comparing the gold sentiment annotation of input and final output, we can quantitatively estimate the usefulness of parsing models, moreover, a qualitative analysis of system outputs provides practical insights for corpora maintainers. In particular, inspecting the output of a sentiment analysis system for discovering possible annotation inconsistencies is one important additional advantage.

We found examples where improvements in the corpus have led to improvements in the output of the sentiment annotator (reducing the number of uses of the `dep` relation and minimizing the errors reported by the UD validator). But some examples can be also found where change in the corpus had made a negative impact in the sentiment analysis (Section 5). We use a different measure (F_2) to extrinsically evaluate the UD corpora. It is not directly related to the intrinsic UD measures such as star rating for UD corpora and LAS for the dependency parser.

Section 2 summarizes the changes of the UD corpora in versions 2.0–2.7. Section 3 describes the sentiment analysis methodology which is used for benchmarking dependency parsers. In Section 4 we show how to evaluate multilingual systems, and in Section 5, we discuss the differences of multiple versions of UD corpora with multilingual instances of changes in syntactic structures and downstream results.

2 Universal Dependencies

Universal Dependencies is a framework for designing and maintaining consistent syntactic annotations across multiple languages. The UD corpora are updated every six months by numerous contributors.

However, few studies have focused on the changes in outputs of UD-trained parsers used for application tasks. Labeled Attachment Score (LAS) and the UD star rating are two commonly used metrics to evaluate the update of the UD corpora. LAS is a measure of the performance of dependency parsers, where the universal dependency labels are taken into account in the measurement. The star rating is a measure designed by UD organizers, which quantifies the qualities of the corpora themselves, such as usability of corpora and variety of genres. While the UD corpora and the parsers have been evaluated, there is a need for an external evaluation of UD in application tasks.

To explore the impacts caused by updates of UD

[1] In this paper we skipped v2.1 and v2.3 to more focus on the recent releases.

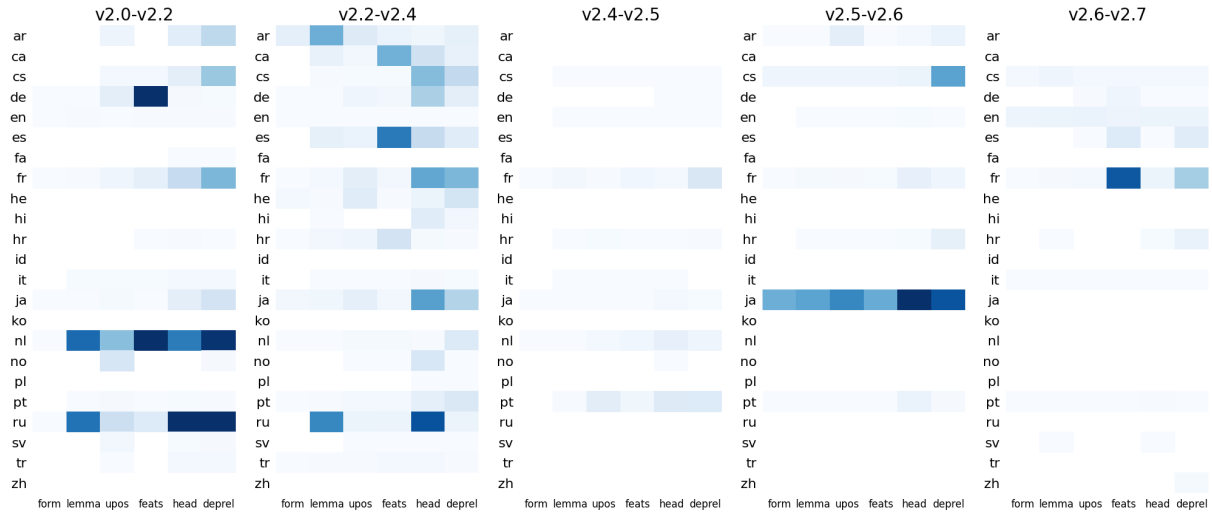


Figure 2: UD corpora version updates. The color of each cell represents the rate of change from the previous version. When a corpus has been significantly updated, the cell is dark in color.

corpora on the sentiment analysis task, we first investigate the changes in the UD corpora listed in Table 1 with versions 2.0–2.7. One treebank was selected per language on the basis of the following conditions: texts are included in the corpus, the corpus is sufficiently large, the updates are frequent so a long term comparisons can be made across versions 2.0–2.7.

Figure 2 shows the UD treebanks updates for versions 2.0 (March 2017) to 2.7 (November 2020) in 23 languages. Inspecting the amount of changes between versions for each treebank was done regarding six out of the ten fields in the CoNLL-U files (form, lemma, upostag, feats, head, and deprel). Most languages have been actively updated in versions 2.0–2.7. In versions 2.0–2.4, most of the modifications in the UD corpora focused on fundamental syntactic elements such as PoS tags and dependency labels, and universal features were incrementally appended. On the other hand, in versions 2.4–2.6, the major updates shifted towards language-specific features.

Through discussion across languages, the UD's annotation policy is gradually becoming more consistent among close languages. PoS tags for copulas and auxiliary verbs are one typical examples of this: "be" in "have been" and "will be" were changed from VERB to AUX in English v2.5, as well as "hebben" in Dutch v2.4. In addition, there is a movement to make AUX a closed set. In Portuguese v2.5, many AUX were changed to VERB, *e.g.,* "continuar" ('continue'), "deixar" ('leave'). Similarly, French v2.1 and onward limit AUX to "être", "avoir" and "faire,". "Pouvoir" ('can') and other words in the same category are tagged as VERB, even though English modal verbs are tagged as AUX.

3 Multilingual Clause-level Sentiment Analysis

We investigate changes in UD corpora and their impact on an application task. We use clause-level sentiment analysis designed for fine-grained sentiment detection with high precision. Kanayama and Iwamoto (2020) demonstrated that a system which fully utilizes UD-based syntactic structures can easily handle many languages, making it an effective platform for evaluating UD corpora and parsing models trained on them.

The main objective of clause-level sentiment analysis is to detect polar clauses associated with a predicate and a target. For example, the sentence (1) below conveys two polarities: (1a) a positive polarity regarding the hotel (which is loved) and (1b) a negative polarity about the waiters (who are *not* friendly).

(1) I love the hotel but she said none of the waiters were friendly.
(1a) + love (hotel)
(1b) − not friendly (waiter)

Figure 3 illustrates the top-down process of detecting sentiment clauses in the dependency tree. The main clause is headed by the `root` node of the dependency tree. When the node has child nodes labeled `conj` and `parataxis`, those nodes are

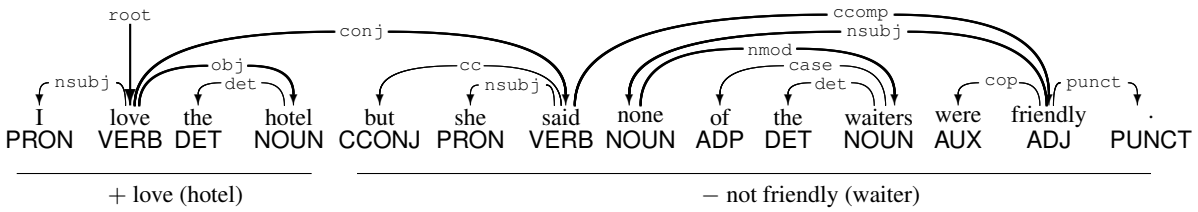


Figure 3: Dependency tree for sentence (1). The dependencies in bold lines from the `root` node are traversed to detect two sentiment clauses (predicates and targets).

	lemma	PoS tag	polarity	case frame
(a)	love	VERB	+	`nsubj`, `obj`
(b)	friendly	ADJ	+	`nsubj`
(c)	unhappy	ADJ	−	`nsubj`, with

Table 2: Examples of lexical entries. '+' is positive and '−' is negative. Underline denotes the target case.

recursively scanned as potential sentiment clauses. When a node is a verb that takes a `ccomp` (clausal compliment) child, *e.g.*, "say", the child node is also examined. In example (1), two clauses, headed by "love" and "friendly" are detected. After detecting the clauses, the predicates are compared with lexical entries associated with a lemma, a PoS tag and its polarity and the case frame, as exemplified in Table 2. Entry (a) is for the verb "love", which is positive and takes a subject and an object; the target (which is positive) is its object. For most adjectives, the target is in the subject, as in (b), but (c) "unhappy" specifies the target as "with" to detect "breakfast" as the target in (2).

(2) [−] I was **unhappy** with the breakfast.

In all of the languages, detecting negation is the key to detecting polarities with high precision. The basic types of negation are direct negation of the verb and noun in (3) and (4).

(3) [−] The hotel was *not* **good**.
(4) [+] It was *no* **problem**.

To multilingualize the clause-level sentiment detector, the English polarity lexicon shown in Table 2 was transferred to other languages as described in previous paper (Kanayama and Iwamoto, 2020).

4 Experimental Settings

To extrinsically evaluate the UD corpora, we combine a UD-compliant dependency parser trained with multiple versions of UD corpora to the sentiment extractor. Since the syntactic structure is the only factor that changes the output of sentiment detection, we can easily find the effects of parsing to the downstream application.

4.1 Dependency parser

In our experiments, we have used two UD-compliant dependency parsers: UDPipe and Stanza. UDPipe (Straka and Straková, 2017) is the standard pipeline which performs sentence segmentation, tokenization, PoS tagging, lemmatization and dependency parsing, can be trained given annotated CoNLL-U format. Though a prototype of UDPipe 2.0 is released with improved morphosyntactic performance compared to UDPipe 1.2, we use UDPipe 1.2 because the resources for training UDPipe 2.0 was not available at this moment.

Since pretrained models are provided for most of treebanks, we used the distributed models trained on UD versions 2.0, 2.2, 2.4, and 2.5[2]. For UD v2.6 and v2.7, we trained the models using the same parameters and word embeddings as those of v2.5. The models for Chinese v2.0 and v2.2 were not included since a simplified Chinese corpus was not available in those versions[3], and Polish for v2.0 is missing as well.

We also used Stanza, an open-source Python natural language processing toolkit that supports 66 languages. In this study we trained the Stanza models using each version of the UD corpora.

4.2 Datasets

To our knowledge, there is no *multilingual* clause-level sentiment annotation such as the Stanford Sentiment Treebank (Socher et al., 2013) for English. To compare output of various languages under the

[2]Downloaded from http://ufal.mff.cuni.cz/udpipe.

[3]UD_Chinese-GSDSimp did not exist in the official version of UD2.4 thus we trained the model for Chinese v2.4 by picking up the pre-release corpus from the development branch as of September 9, 2019.

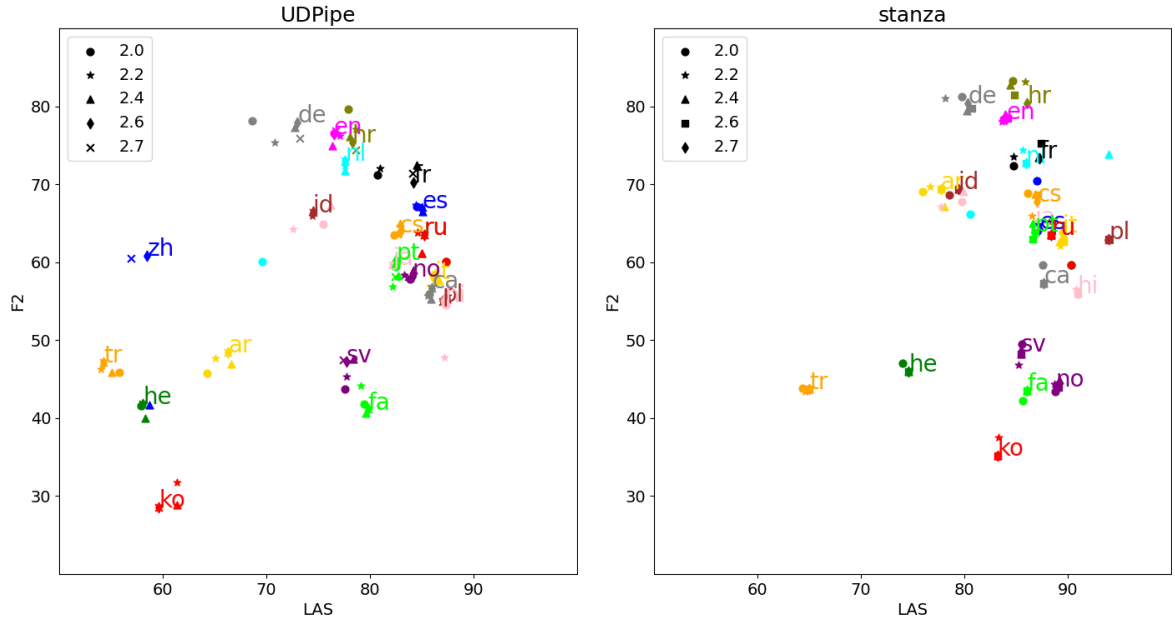


Figure 4: Relationship between parsing score (LAS) and sentiment detection performance (F_2) for each version in UDPipe and Stanza.

same conditions as possible, existing sentiment analysis datasets with clause-, aspect- or sentence-level annotations are simplified to sentence-level annotations by Kanayama and Iwamoto (2020). The reformatted dataset in each language consists of about 500 sentences, each with a positive or negative label. The percentage of those labels is equal and a sentence with a label does not contain a clause of the opposite polarity. Refer to the paper for more details.

4.3 Metrics

We evaluated the performance of the sentiment extractor using sentence-based metrics. Given a sentence, which is labeled either positive or negative in the datasets, our system detects an arbitrary number of sentiment clauses.

We calculate *recall* as the ratio of sentences for which the system detects one or more sentiment clauses that have the same polarity as the sentence-based polarity labeled in the gold data. *Precision* is the ratio of polarity coincidence between the system output (for a clause) and the gold label (for a sentence) for polar clauses detected by the system. A sentence that is labeled either positive or negative may have multiple clauses of opposite polarities, but for simplicity we just consider the sentence polarity in the gold data because we found that a simple evaluation is sufficient for relative comparison of parsers and syntactic operations.

To give precision more weight than recall for practical evaluation, we use the F_2 score in Equation (5), setting $\beta = 2$,

$$F_\beta = (1+\beta^2)\frac{\text{prec} \cdot \text{rec}}{\text{prec} + \beta^2 \cdot \text{rec}} \qquad (5)$$

We do not measure our system using the F_1 score because a naive word-spotting approach may result in a higher F_1 score where every sentence is classified positive or negative. The system is not for polarity classification, but to detect clauses that certainly express polarity. Therefore, non-syntactic sentiment clues (*e.g.* hashtags) or polar clauses with uncertain polarity to the target (*e.g.* subjunctive) are basically undetected.

5 Results and Analysis

5.1 Overall Quantitative Results

Figure 4 shows an overview of the relationship between dependency parsing and sentiment detection. The F_2 values calculated by switching the dependency parsing models trained on UD versions 2.0–2.7 in 23 languages and keeping the rest of the process (sentiment lexicon and tree-screening algorithms) consistent described in Section 3.

The figure shows that within a language, F_2 tend to increase as the LAS improves, and the latest version (v2.7) achieves better LAS and F_2 scores than the oldest one (v2.0) in many languages. The removing of bugs using UD validator and a variety

of annotation changes in the corpus contributes to the improvement of both the LAS and the F_2, but other changes do not always improve the scores. For example, when annotations with complex and correct dependency relations are added, the learning of parsers become difficult and the LAS may decrease. No clear correlations between LAS and F_2 scores can be observed, and that is precisely the motivation for the qualitative analysis presented in next section. It means, F_2 (or our system, namely, evaluation in an application task) works as a different measure from the LAS or star rating.

Note that the F_2 score is difficult to compare in different languages because of diversity in complexity of datasets. The performance of the dependency parsers are determined not only by the training corpora but also by the parameter settings and external resources (*e.g.* word embeddings).[4]

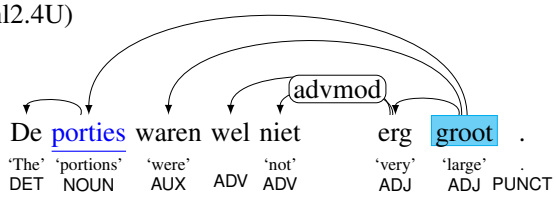
5.2 Analysis of each language

We illustrate sentence pairs where dependency parsers changed the outputs of the sentiment extractor correspondingly. A label such as "(nl2.4U)" denotes the language, UD version and dependency parser (UDPipe or Stanza). For example, (nl2.4U) denotes a Dutch result parsed by UDPipe which was trained on UD v2.4. A highlighted box shows the predicate and an underlined word shows its target, with blue color for positive and red for negative.

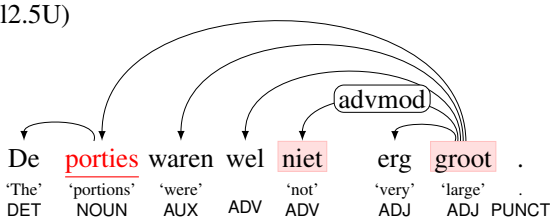
First, we show the differences in parsing that can significantly affect the results of sentiment clause detection, although we cannot guarantee whether they are caused by changes in corpora. Correct detection of negation is important for a downstream task, especially polarity detection. The sentiment extractor detected a polar expression "groot" ('large') from Dutch sentences (nl2.4/2.5U). In (nl2.5U), the adjective "groot" is correctly negated by the adverb "niet" due to the direct link between two words and resulted in the correct extraction of negative sentiment, while in (nl2.4U) the system failed to change the polarity because "niet" was not directly attached to "groot".

[4]In some languages, different LAS scores were reported in different versions even when two corpora were identical.

(nl2.4U)

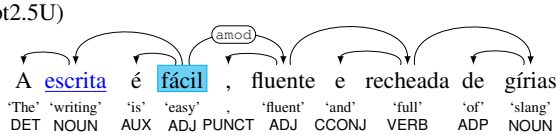


(nl2.5U)

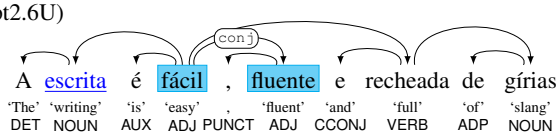


As shown in Section 3, a dependency label `conj` is heavily used for multiple clause detection; thus, it is the factor that significantly impacts the recall of the sentiment detector. For example, let us see (pt2.5U) and (pt2.6U) where the root node is "fácil" ('easy'). In (pt2.6U), the system correctly detected "fluente" as positive predicate, while it is regarded a conjunct of "fácil". In (pt2.5U), a predicate "fluente" ('fluent') is modifying the root node with a wrong label `amod` thus only one positive predicate "fácil" is detected.

(pt2.5U)

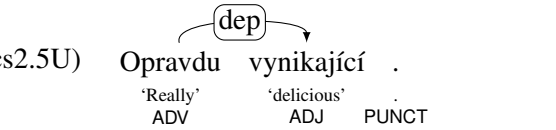


(pt2.6U)

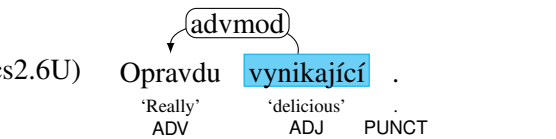


Giving correct annotations and removing inconsistencies within a corpora improve the performance of parsing, and output of the sentiment extractor as well. Reduction of unspecified labels, namely `dep` label, is still a challenge in a variety of UD corpora. In (cs2.5U), the parsing result was not correct with a `dep` label, but the parsing result was improved in (cs2.6U) and thus a positive predicate was extracted.

(cs2.5U)

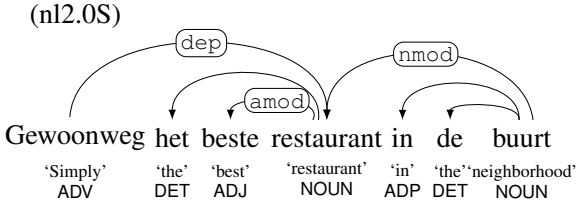


(cs2.6U)

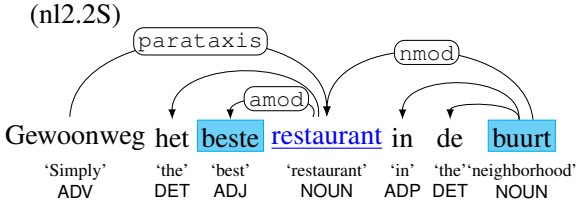


A similar change in parsing results was observed in Dutch. There were 1,471 `dep` labels in UD Dutch v2.0, but they were explicitly labeled in v2.2. That makes it possible to extract the polarity of the sentence in (nl2.2S) with the correct dependency labels.

(nl2.0S)

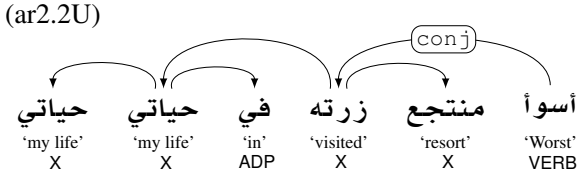


(nl2.2S)

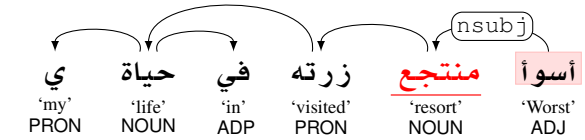


In the update of UD Arabic v2.4, various bugs were fixed which found by the new UD validation tool. In (ar2.2U)[5], the tokenizer did not correctly split a token "حياتي" ('my life'), and thus the parser wrongly duplicate the token. In addition, many words had been tagged as X in (ar2.2U). The usage of X tag should be limited to special cases such as foreign words. In (ar2.4U), all words were correctly tagged and helped the detection of negative polarity. These are typical examples where refinements of corpus improved the output of the sentiment extractor.

(ar2.2U)



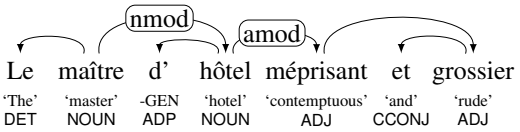
(ar2.4U)



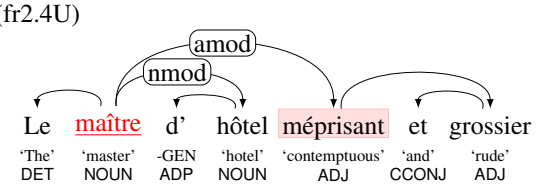
A lot of dependency labels and PoS tags were updated in UD French v2.4. In a noun phrase (fr2.4U), a negative adjective "méprisant" ('contemptuous') was successfully detected because its was correctly attached to the head noun "maître" ('master').

[5]Arabic tokens are written from right to left, based on the actual order of the sentences.

(fr2.2U)

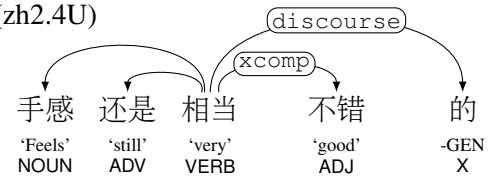


(fr2.4U)

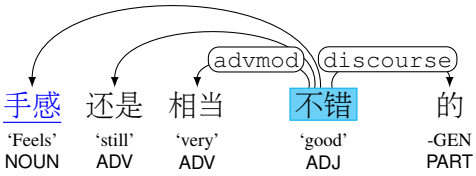


The PoS tagging error of "相当" ('very') in (zh2.4U) was fixed in (zh2.5U). Then the dependency structure was improved and a positive polarity was correctly detected.

(zh2.4U)

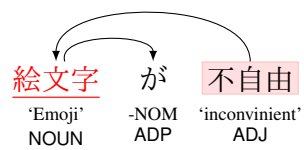


(zh2.5U)

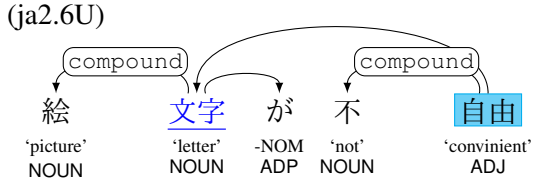


Major changes of tokenization policy or lemmatization significantly affect syntactic structures. The adjective "不自由" ('inconvenience') was regarded as a single word in (ja2.5U), which matched the sentiment lexicon, so the negative polarity was correctly detected. However, since UD Japanese v2.6 adopted short word units in its tokenization policy, "不自由" is divided into two words "不" + "自由" ('in-' + 'convenience') in (ja2.6U). The polarity was wrongly inversed because the system did not handle this type of negation. Meanwhile, this error can be easily fixed in future, by adding a rule to handle the negation by "不" to the sentiment extractor.

(ja2.5U)



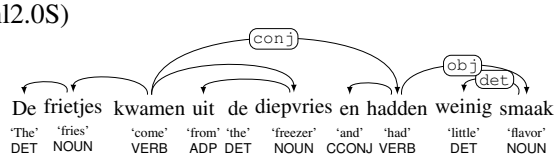
(ja2.6U)



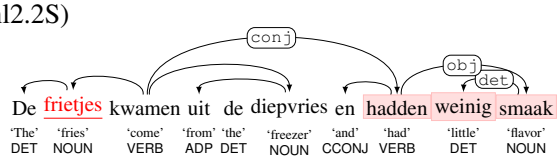
A similar example can be found in Dutch. In (nl2.0S), the lemma of "hadden" ('had') was "heb", but in (nl2.2S) the lemma was changed to "hebben". Since our system is based on the lemma of UD Dutch v2.4 (*e.g.*, for making dictionaries), parsers trained on corpora with different annotation policies result in worse performance.

If a dependency parser trained on UD is to be used for an application task, the user may consider whether the parser should be retrained for the new UD corpus. Detailed change logs of a corpus will help the system catch up the updated corpus.

(nl2.0S)

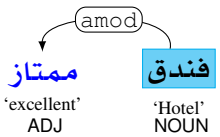


(nl2.2S)

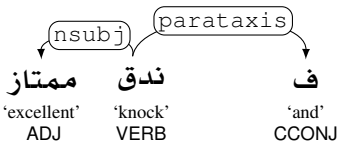


Next, we show an example where the updates in UD corpora have affected unintended parts of parsing results. In Arabic, the improvement of the MWT (multi-word token) labels has influenced other parts of the system. In UD Arabic v2.4, the labeling of MWTs containing "ف" ('and') has been improved. However, it caused overfitting; the model learned that words containing "ف" are always MWTs and increased parsing errors in the word "فندق" ('hotel') as shown in (ar2.4S).

(ar2.2S)

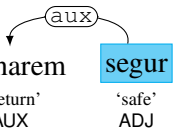


(ar2.4S)

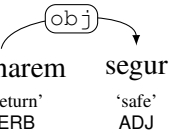


The change in the labels attached to verbs had an effect on the application task. In Catalan, many occurrences of AUX tag were changed to VERB in version 2.4. A PoS tag of "tornar" ('return') is changed from AUX to VERB, making the polar expression "segur" ('safe') being missed, because "segur" is the root node in (ca2.2S) but not in (ca2.4S).

(ca2.2S)

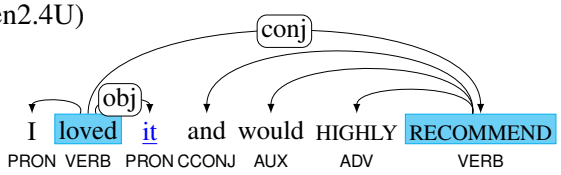


(ca2.4S)

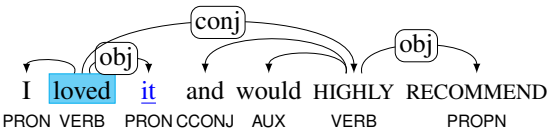


To utilize UD-trained parsers in application tasks, it is expected to be robust to a variety of inputs. In (en2.5U), PoS tagging was not robust enough for an upparcased writing "HIGHLY RECOMMEND", and the PoS tagging error was propagated to dependency parsing and sentiment extraction.

(en2.4U)

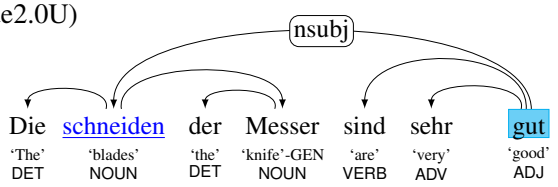


(en2.5U)

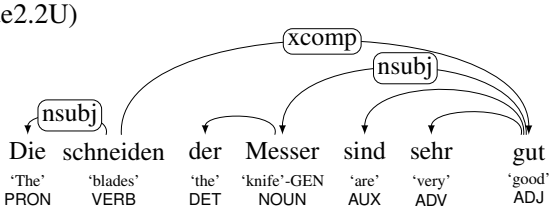


A similar issue can be found in German. Nouns should be always capitalized regardless of its position in a German sentence. In (de2.2U), a noun "Schneiden" ('blades') was wrongly tagged as VERB because it was not capitalized. Since real-world inputs such as reviews may contain such capitalizing errors and misspellings, robust PoS taggers to those errors are desired. It is important to use UD treebanks that is sufficiently large for the parser training and suitable genres for the downstream tasks.

(de2.0U)



(de2.2U)



6 Conclusion

We observed updates of the UD corpora versions 2.0–2.7 in 23 languages and extrinsically evaluated the parsing models trained by the corpora in a real-world scenario. The evaluation using the sentiment extractor with UD-trained parsers do not correlate clearly with existing evaluations such as LAS and star rating, indicating that evaluation using an application task is useful to measure UD corpus from a new perspective.

We showed examples where the updates of UD corpora have either adversely or positively affected the output of dependency parsing and sentiment clause detection. Our methodology is easier to find the changes of sentiment detection. Those changes often show the important aspects of syntax.

We identified issues in multilingual applications of the UD platform. For example, some corpora have less diverse writing styles for informal sentences which are more common in review documents. In some languages, UD corpora updates have been slowed down after version 2.4 and shifted towards language-specific features and augmented dependencies, but there are still open problems in fundamental syntactic structures. We anticipate continuous improvements to multilingual corpora for UD communities worldwide. We hope the emergence of other applications that utilize UD's syntactic structures will lead to further discussions and enhancements of multilingual corpora.

References

Murhaf Fares, Stephan Oepen, Lilja Øvrelid, Jari Björne, and Richard Johansson. 2018. The 2018 shared task on extrinsic parser evaluation: On the downstream utility of English Universal Dependency parsers. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 22–33.

Hiroshi Kanayama and Ran Iwamoto. 2020. How Universal are Universal Dependencies? Exploiting Syntax for Multilingual Clause-level Sentiment Detection. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4063–4073.

Daniel Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2779–2795.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies*, 135, pages 86–95.

Stephan Oepen, Lilja Øvrelid, Jari Björne, Richard Jo-hansson, Emanuele Lapponi, Filip Ginter, and ErikVelldal. 2017. The 2017 shared task on extrinsic parser evaluation. towards a reusable community infrastructure. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*, pages 1–16.

Holger Schwenk and Matthijs Douze. 2017. Learning Joint Multilingual Sentence Representations with Neural Machine Translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

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*, pages 88–99.

Daniel Zeman et al. 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*, pages 1–19.

Daniel Zeman et al. 2020. Universal Dependencies 2.6. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.