Automatic Manipulation of Training Corpora to Make Parsers Accept Real-world Text

Hiroshi Kanayama, Ran Iwamoto, Masayasu Muraoka, Takuya Ohko, Kohtaroh Miyamoto
IBM Research
{hkana@jp., ran.iwamoto1@, mmuraoka@jp., ohkot@jp., kmiya@jp.}ibm.com

Abstract

This paper discusses how to build a practical syntactic analyzer, and addresses the distributional differences between existing corpora and actual documents in applications. As a case study we focus on noun phrases that are not headed by a main verb and sentences without punctuation at the end, which are rare in a number of Universal Dependencies corpora but frequently appear in the real-world use cases of syntactic parsers. We converted the training corpora so that their distribution is closer to that in realistic inputs, and obtained better scores both in general syntax benchmarking and a sentiment detection task, a typical application of dependency analysis.

Keywords: syntax, parsing, Universal Dependencies

1. Introduction

In text processing applications that handle documents such as user reviews and contract documents, accurate syntax parsing is desired for semantic analysis and information extraction. The emerging generative approach also requires the analysis of given utterances to make systems reliable and explainable, such as in retrieval augmented generation (Lewis et al., 2020), and the language models can be improved by incorporating syntactic knowledge (Iwamoto et al., 2023).

Multilingual corpora in Universal Dependencies (UD) (Nivre et al., 2016, 2020) are easily available, and they are used for training and evaluation of syntactic analysis components including tokenizers, part-of-speech (PoS) taggers, and dependency parsers, such as Stanza (Qi et al., 2020), UDPipe (Straka, 2018), spaCy (Honnibal et al., 2020) and Trankit (Nguyen et al., 2021).

However, we found a gap between the standardized UD corpora and the real-world application scenarios. There are many noun phrases (NPs) in reviews such as hotel ones written by a customer as (1), instead of a formal sentence typically with a finite verb in a root node of the syntax tree such as in (2).

(1) A very good hotel close to the park!
(2) I think the hotel is very good because it is close to the park.

Another example of noun phrases is a description in a contract document, such as in (3).

(3) total cost of the services

These noun phrases can appear in many kinds of text documents as the title of a document or section, items in enumeration, a header line of a table, and so on. In addition, in many cases, such strings do not have a period or other punctuation marks at the end.

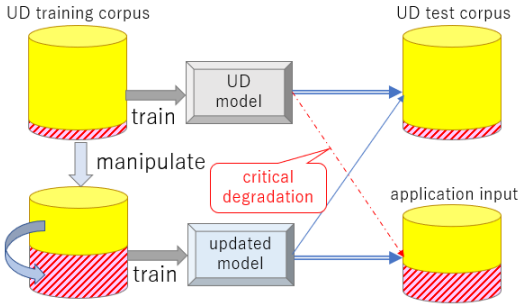
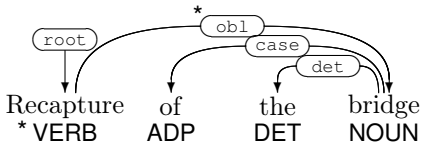
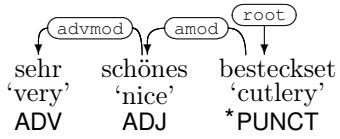


Figure 1: The concept of this paper: an issue of different distribution of text characteristics and its solution by corpus extension.

When we apply a syntax analyzer trained on the UD corpora to such short noun phrases, we often find very wrong analysis results, as exemplified in the output syntax structures of English (4) and German (5). A '*' mark indicates the errors in PoS tags or dependency relations.

(4)


(5)


In (4), the first word "recapture" (which should be NOUN) was incorrectly tagged as VERB as if

Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD 2024)
@LREC-COLING-2024, pages 4–13. May 25, 2024.

the input were an interrogative sentence that starts with a verb, and it causes an incorrect dependency relation between "recapture" and "bridge," which should be `nmod` rather than `obl`. In (5), the noun "Besteckset" ('cutlery') was tagged as PUNCT. The writing is not formal because German nouns should start with a capital letter, but the tagging result PUNCT is apparently incorrect. These are actual results by the Stanza parser that achieved very high scores in the UD parsing shared task (Zeman et al., 2018), and we found other taggers and parsers such as UDPipe and spaCy produced similar errors. These errors have already been recognized in the community and discussed in the GitHub issues of those implementations[1].

If most of the contents in the training corpora contain finite verbs in the sentence rather than only noun phrases, it is not surprising that the taggers and parsers trained on such corpora tend to produce incorrect results for the noun phrase inputs such as (4) and (5). Also, we can assume that very unusual tagging results such as in (5) are caused by the training corpus where most sentences end with a period ('.'). Thus, they are problems in the difference between the training corpus and target input to be analyzed.

Figure 1 illustrates the problem that this paper addresses. Normally, the syntax analyzers are trained and evaluated on the UD corpora, but the real-world input documents have different distributions from those of the UD corpora, and the models trained on the UD corpora cause catastrophic errors in applications. Thus, we manipulate the UD corpora to alter distributions in terms of noun phrases and sentence-end punctuation. Although it is impossible to know the general distribution in the real-world inputs, we can make the parser more robust by manipulating the training corpus to reduce the bias in the current UD data.

The contributions of this paper are: (1) to handle the issue regarding noun phrases in addition to punctuation, (2) to provide an algorithm to manipulate training corpora without any manual annotation work, (3) to propose methods to evaluate this work from multiple viewpoints, including the automatic generation of an evaluation data set of noun phrases, and (4) to show the effects of the corpus manipulation in four languages.

Section 2 reviews the related work regarding UD and existing discussions on punctuation and noun phrases. In Section 3, we define the terms used in this paper. Section 4 shows the statistics in different corpora. In Section 5, we propose the algorithm to manipulate training corpora so that the parser can accept real-world inputs, and the effect is shown in Section 6.

[1] An issue of Stanza https://github.com/stanfordnlp/stanza/issues/488 and of spaCy https://github.com/explosion/spaCy/issues/5596.

2. Related Work

Universal Dependencies (UD) (Nivre et al., 2016) is a worldwide project to provide multilingual syntactic corpora. As of November 2023, 259 treebanks in 148 languages have been released. For all languages, the syntax is represented by dependency trees with 17 PoS tags and 37 dependency labels commonly used for all languages, and each treebank can have language specific extensions. The resources and documentations are available online and incrementally updated.[2] A major shared task of multilingual parsing (Zeman et al., 2018) was held, and a result, UD treebanks is now a de facto standard of multilingual research and many tokenizers and parsers have been trained on them, including a multilingual single parser (Kondratyuk and Straka, 2019).

English Web Treebank (EWT) (Silveira et al., 2014) is one of the most commonly-used treebanks in UD. Originally, it was designed to cover more informal text, such as email and review documents, which was not included in the treebanks of the Wall Street Journal (WSJ). After the emergence of Universal Dependencies, EWT was converted to a UD-style annotation. Thus, EWT contains noisy sentences with typos and abbreviations, and even sentence splitting is tricky (Udagawa et al., 2023), but their work showed that the parsers trained using EWT had a better capability to parse such informal text than the model trained only with WSJ. Due to this historical reason, the EWT corpus functions as an outlier in the experiments in this paper.

The effects of punctuation in a dependency parser have been discussed by Søgaard et al. (2018). They pointed out that dependency parsers, especially neural implementations, are highly sensitive to punctuation in training corpora, and training parsers without punctuation makes the models better. In this paper, we extend the discussion from punctuation to noun phrases, which are more critical in real-world applications. Nivre and Fang (2017) pointed out that punctuation highly affects the benchmarking scores in a number of corpora even if it is not significant in the actual analysis.

The analysis of noun phrase structures have been discussed (Nakov and Hearst, 2005; Vadas and Curran, 2011) but parsing confusion between noun phrases and finite sentences has been less studied. There was a report that a parser specific to noun phrases improved machine translation quality even if the LAS (labeled attachment score) of dependency parsing was not significantly changed (Green, 2011).

[2] https://universaldependencies.org/

Corpus synthesis is a powerful method to adapt to specific tasks to enhance a production parser (El-Kurdi et al., 2020) and to broaden the supported languages (Tiedemann and Agic, 2016; Dehouck and Gómez-Rodríguez, 2020) and domains (Li et al., 2019; Jia and Liang, 2016). This paper shares a similar motivation with them but we propose a method to extend training corpora with linguistic knowledge to address specific issues without adding new data sources.

3. Terminology

In this section terms used in this paper are defined.

Unit A text string that is regarded as a single "sentence" in corpora[3]. A unit is also given as an input to a PoS tagger, dependency parser, and their downstream applications, which may be a result of sentence splitting. In this paper we do not call it a "sentence" to distinguish it from the *sentence* defined as follows. All of (1), (2), (3), (4) and (5) in Section 1 can be a unit.

Sentence A unit that is governed by a finite verb, including nominal predicate sentences associated with a copula. A sentence corresponds to a non-terminal symbol 'S' in the phrase structure grammar, though this paper does not discuss its definition from a linguistic viewpoint. Example (2) in Section 1 is a sentence.

Noun Phrase (NP) A syntactic tree or subtree whose head word is a noun or a proper noun, namely, its universal PoS (UPOS) tag is either NOUN or PROPN. An NP also does not have a child node of a copula (where dependency relation label is `cop`). Note that in the content-head structure of UD, the head word of a sentence "She is a teacher." is "teacher" rather than "is" (be-verb).

Noun Phrase Unit (NPU) A unit that forms an NP. Examples include (1), (3), (4) and (5) in Section 1.

Ending punctuation (end-punct) A punctuation mark at the end of a sentence or unit. Here, a punctuation mark is a word that is tagged as PUNCT in the UD corpora. In this paper, we only focus on a period ('.'), an exclamation mark ('!') and a question mark ('?'), which are used in European languages, and discard other PUNCT words like parentheses and quotation marks.

Punctuation Omitted Unit (POU) A unit without ending punctuation. Examples include (3), (4) and (5) in Section 1.

4. Observation of Corpora

To determine how many noun phrase units (NPUs) and punctuation omitted units (POUs) existed in the training corpora and expected input documents, we observed two types of corpora in four languages. One is Universal Dependencies (UD), which is used for the training of various syntax analyzers. Here, we observe the development portion in UD Version 2.13. The other is the review data used for the evaluation of sentiment analysis. We randomly selected 100 sentences[4] of each language version of review data from the SemEval shared task data for aspect-based sentiment analysis (Pontiki et al., 2016) for English, French and Spanish, and Amazon reviews used in another shared task (Ruppenhofer et al., 2014) for German.

Table 1 shows the ratio of the NPUs and POUs in the UD corpora and review documents. Particularly in the UD corpora of French and Spanish, the ratios of NPUs and POUs are very low, that is, the UD corpora tend to consist of formal sentences with finite verbs with ending punctuation marks as their units. The UD English corpus has a relatively higher ratio of NPUs and POUs because there are many informally written documents in EWT corpus as mentioned in Section 2.

The review corpora tend to have many NPUs and POUs, except for the English SemEval data set. There are fewer POUs in SemEval data set (particularly the English one) as expected, that is, most of the units end with a period. The SemEval corpora are supposed to be controlled to have periods for the purpose of extraction of positive or negative expressions with aspects.

As we previously observed, the distribution of syntactic characteristics is very diverse, and those trends highly depend on the formality or cleanliness of the contents of the data set and languages. This shows that it is quite difficult to expect a fixed corpus such as those of UD to represent the distribution of real-world documents that are given to the applications of syntactic analyzers.

5. Corpus Extension

In this section, we propose a method to extend the training corpora for syntactic analyzers, to address the problem of differences in characteristics of corpora discussed in Section 4. Our goal is to build

[3] In the CoNLL-U format used in UD, a unit is represented by a metadata tag '`# text =` '.

[4] Those review data sets do not have syntactic annotation, thus we made manual observation in the limited sentences.

language	corpora UD	corpora review	NPU ratio (%) UD	NPU ratio (%) review	POU ratio (%) UD	POU ratio (%) review
English	EWT	SemEval	23.0	3.0	19.5	1.0
French	GSD	SemEval	3.2	36.0	0.8	3.0
German	GSD	Gestalt	6.1	28.0	1.3	12.0
Spanish	AnCora	SemEval	4.5	25.0	0.8	7.0

Table 1: Ratios of noun phrase units (NPUs) and punctuation omitted units (POUs) in UD and review corpora of four languages.

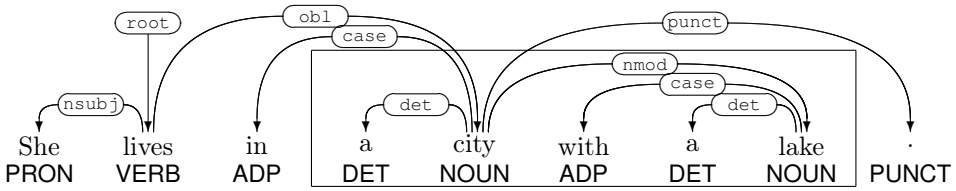


Figure 2: Extraction example of an NP (indicated as a box) from a sentence.

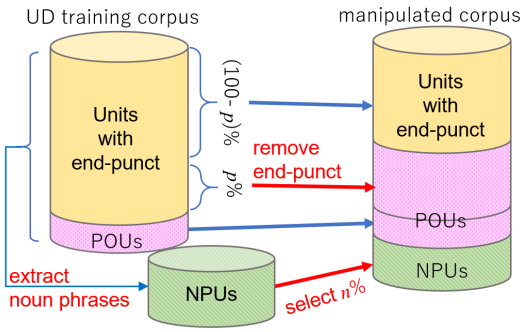


Figure 3: Training corpus extension.

models useful in real applications by reducing the bias in the training model to the UD corpora as illustrated in Figure 3.

5.1. Removal of punctuation

We assume that the incorrectly assigned PUNCT tag to a noun in (5) is caused by the PoS tagging model trained on the corpus where most of last word is tagged as PUNCT. A desirable model is robust to the existence of punctuation, that is, the result should be consistent with or without an end-punct.

A straightforward solution to the problem of the bias to the training corpora is to reduce end-punct at a certain ratio p, in other words, to add POUs, and then to retrain models. Most end-puncts do not have any child nodes in the dependency tree, and thus, it is quite easy to remove an end-punct from a unit, maintaining the validity of the tree[5].

[5]As an exception, the UD_English-EWT training corpus contained a unit in which a conjunction "and" attached to a period at the end of the sentence. In such a case, we did not apply the modification of punctuation removal.

5.2. Addition of noun phrases

In addition to sentences headed by a finite verb, a training corpus should contain noun phrases as units, to handle similar inputs in real applications. To make such a corpus, we add NPUs by extracting noun phrase subtrees from the original corpus in the following manner.

- Identify nouns (a word tagged as NOUN or PROPN) in the original dependency trees.
- Find noun phrases, selecting nouns whose subtree headed by the noun consists of more than three sequential words[6].
- Exclude a preposition and punctuation that should not be a part of a noun phrase. This treatment is needed because the syntactic structure in UD is designed in a content-head manner, and thus, a number of function words are included in a subtree of noun phrases. Functional words that attach to the head of the noun phrase with a dependency label `case` or `punct` are removed from the noun phrase. In the case shown in Figure 2, the preposition "in" is excluded from the noun phrase headed by "city" because it attaches to "city" with `case` relation.
- Pool the noun phrases extracted in this way, and randomly select a number of them in a given ratio (n) to add them to the training corpus, keeping all of the original units in the corpus.

[6]If the children or descendant nodes have a gap due to non-projectivity, such noun phrases are ignored.

In Section 6, we will show the effects of corpora conversion by changing the ratio of punctuation removal and noun phrase addition.

6. Experiments

6.1. Data for evaluation

We will evaluate the syntax analyzer trained on the extended corpora in three ways using three different data sets in four languages: English, French, German and Spanish.

6.1.1. Noun phrase

We evaluate the robustness of the syntax analyzers to the input strings of noun phrases, as a unit test of our approach. For this purpose, we automatically generated the test set of noun phrases in the following procedure.

- Obtain section titles[7] of Wikipedia articles of four languages
- Extract section titles that consist of three or more words
- Exclude those that contain special characters such as numbers, symbols, quotation and punctuation marks
- Exclude those containing non-canonical upper/lower cases (e.g. "RNAb", "AIESEC")
- Exclude those that were judged as different languages from that of Wikipedia
- For English, French and Spanish, change the initial character of each word into lower case
- Remove duplication
- Diversify the first word so that there are no more than three entries that share the first word. This is to reduce frequent patterns such as "List of XX"
- Randomly select 1,500 entities for each language

This process almost perfectly extracts noun phrases in each language, and by definition, the last word is not punctuation. Table 2 shows examples in four languages.

In the experiments in this section, we will apply PoS taggers and dependency parsers to these data to calculate the following two scores:

[7] Note that they are different from the titles of articles because the majority of article titles are proper nouns, and they are not appropriate to test our method because names are not confused with sentences, and movie titles are hard to determine the desirable annotation (e.g. "Gone with the Wind").

Wrong punctuation The number of cases where the last word is tagged as PUNCT or its dependency label is `punct`. A lower number is better.

NP detection The ratio of the dependency trees of which the root node is tagged as NOUN. A higher ratio is better.

6.1.2. Universal Dependencies

We use the UD corpora for the intrinsic evaluation of dependency parsers. The F1 score of LAS is used as a representative evaluation metric. In our experiments, we extend the *train* and *dev* portions of the UD corpora with the methods presented in Section 5, and the *test* portion for evaluation is not changed. This means the distributions of units are different between the test and training corpora. As a result, the LAS score on the UD test corpus will be theoretically decreased, and thus, minimizing the downgrade of the LAS score indicates the success of our approach.

6.1.3. Sentiment detection

We also conduct an extrinsic evaluation using multilingual sentiment detection (Kanayama and Iwamoto, 2020; Iwamoto et al., 2021) as an application of dependency parsing. For the evaluation, we used sentiment analysis data sets that were observed in Section 4. Those data sets for four languages were derived from shared tasks (Pontiki et al., 2016; Ruppenhofer et al., 2014) and all of them are customer's review data in a domain per language (restaurant for English, French and Spanish, cutlery for German). Each of them contains 500 units, and the annotations were simplified so that each unit has a unit-level polarity flag (either positive or negative) as shown in Table 3.

Similarly to the previous work on multilingual sentiment detection (Kanayama and Iwamoto, 2020), we calculated precision and recall as metrics. Precision depends on the quality of the sentiment lexicon and handling of syntax phenomena such as negation. Recall is related to the coverage of the sentiment lexicon and accuracy in detection of the root node in dependency analysis. The experiments in this paper have few factors that change the precision of sentiment detection, and thus, we focus on recall as it is affected by syntactic structures related to noun phrases.

6.2. Parser retraining

We applied the two kinds of conversion described in Section 5 to the training portions of the UD corpora in four languages (German-GSD, French-GSD, Spanish-AnCora and English-EWT), and retrained models of the Stanza version 1.1.1 (Qi et al.,

English	all passenger trains
	cobordism of manifolds with additional structure
French	ponts sur d'autres cours d'eau ('bridges over other waterways')
	instance vérité et dignité ('Truth and Dignity Commission')
German	Meine Daten und ich ('My data and I')
	Mangelnde wissenschaftliche Grundlage ('Lack of scientific basis')
Spanish	recopilatorios y discos especiales ('compilations and special discs')
	contenido de agua en el suelo ('water content in the soil')

Table 2: Examples of noun phrases in the Wikipedia section title data set.

English	This has got to be one of the most overrated restaurants in Brooklyn.	Negative
	Best Pastrami I ever had and great portion without being ridiculous.	Positive
French	Aucune commande de dessert n'a été prise après une demie heure d'attente à la fin de le plat. ('No dessert order was taken after half an hour wait at the end of the dish.')	Negative
	Petit restaurant à le décor soigné, à les tables bien mises. ('Small restaurant with neat decoration, well-set tables')	Positive
German	Die Griffe sind schön geformt, die Messer liegen angenehm in der Hand und sind scharf. ('The handles are beautifully shaped, the knives are comfortable to hold and sharp.')	Positive
	Rostflecken nach Spülmaschine ('Rust spots on dishwasher')	Negative
Spanish	El servicio es muy bueno y la calidad de la comida al mismo nivel. ('The service is very good and the quality of the food at the same level.')	Positive
	Un restaurante al que no pienso volver. ('A restaurant which I don't want to come back to')	Negative

Table 3: Examples of sentiment polarity data. The second example of each language is a noun phrase.

2020) with the extended training corpora. For all languages, we retrained PoS tagging models (`pos`) and dependency parsing models (`depparse`) with maximum iteration of 5,000 times[8], and other models for tokenization (`tokenize`), multi-word tokens (`mwt`) and lemmatization (`lemma`) were not changed from the default ones.

We tested various ratios for the removal of punctuation (p) and addition of noun phrases (n). $p = 0$, $n = 0$ means the original UD corpus as it is, and thus, it is the baseline for each language. We evaluated two scores using the noun phrase data sets described in Section 6.1: number of incorrect punctuation and ratio of NP detection. We also evaluated the LAS score using the UD test corpus, and the recall of sentiment detection using the review corpus.

Stanza's retraining process is randomized and the resultant models are not deterministic, and thus, we conducted 10-times retraining on the baseline settings ($p = 0$ and $n = 0$) to report the average and standard deviation of each score.

[8]Setting `max_steps=5000`, one tenth of the default setting. This is to reduce training time with small sacrifice of accuracy.

6.3. Results

Tables 4, 5, 6 and 7 show the results of all metrics for German, French, Spanish, and English, respectively. The top row ($p = 0, n = 0$) shows the baseline scores with the model trained on the original corpus. The next section (remove punct) shows the effects of reducing end-punct by p, and the last section (add NP) reports the scores by adding NPs to the training corpus varying n, including combination of both modification with p and n.

In the baseline models of German, French and Spanish, there were 3.2 to 4.2% of catastrophic punctuation errors. Removing end-puncts effectively reduced such errors, even with a small ratio of p. By setting $p = 20\%$, such errors were completely avoided in the four languages.

However, just removing punctuation did not improve the scores of other metrics, although there are a number of settings that improved NP detection in French and Spanish. Also, the changes of LAS and sentiment recall were marginal. The large decrease of LAS scores for $p = 100\%$ (3 points decrease in German and French) is as expected because $p = 100\%$ means all end-puncts were removed from the training corpora, and the punctuation marks that remain in the test corpora are difficult to handle with the model trained by the training corpora without any end-puncts.

	(%)		Section title				UD		Sentiment	
	p	n	Wrong punct (↓)		NP detection (↑)		LAS (↑)		Recall (↑)	
baseline	0	0	3.2	±1.75	97.4	±0.16	79.68	±0.25	52.1	±1.0
remove punct	10	0	**0**	+	97.3		79.85		**53.2**	+
	20	0	**0**	+	97.1		78.98		51.0	−
	50	0	**0**	+	97.3		79.73		50.4	−
	100	0	**0**	+	97.4		76.78	−	49.6	−
add NP	0	10	**0**	+	**98.1**	+	79.59		52.9	
	10	10	**0**	+	**97.8**	+	79.87		**54.6**	+
	20	10	**0**	+	97.5		**80.20**	+	52.9	
	0	20	**0**	+	**97.7**	+	79.64		**53.8**	+
	0	50	**0**	+	**98.1**	+	79.23	−	51.5	
	50	50	**0**	+	**97.9**	+	79.70		52.4	
	0	100	**0**	+	**98.4**	+	79.60		51.5	

Table 4: Results of syntax analysis and sentiment detection in German using the models trained on the extended UD corpora with p punctuation removal and n noun phrase addition. In percent except for the number of incorrect punctuation marks. The top row ($p = 0$, $n = 0$) shows the baseline scores with the original corpus, with the average score of 10 trials and standard deviation. In other rows, a bold number with a + mark indicates that the score is significantly better than the baseline with a difference higher than the standard deviation. A − mark indicates the score is worse against the baseline.

			Section title				UD		Sentiment	
	p	n	Wrong punct (↓)		NP detection (↑)		LAS (↑)		Recall (↑)	
baseline	0	0	4.2	±0.55	91.4	±0.55	87.14	±0.18	43.0	±0.5
remove punct	10	0	**1**	+	**92.5**	+	87.01		42.6	
	20	0	**0**	+	90.6		87.31	−	43.2	
	50	0	**0**	+	91.1		**87.57**	+	**43.6**	+
	100	0	**0**	+	**92.3**	+	84.57	−	42.0	−
add NP	0	10	**3**	+	**93.2**	+	**87.33**	+	42.0	−
	10	10	**0**	+	**93.2**	+	87.09		43.0	
	20	10	**0**	+	**92.9**	+	87.19		42.4	−
	0	20	**0**	+	**93.2**	+	87.25		**44.0**	+
	0	50	**0**	+	**94.4**	+	86.76	−	42.6	
	50	50	**0**	+	**93.6**	+	87.02		42.8	
	0	100	**0**	+	**95.5**	+	86.37	−	**43.6**	+

Table 5: French results. See the caption of Table 4 for details.

			Section title				UD		Sentiment	
	p	n	Wrong punct (↓)		NP detection (↑)		LAS (↑)		Recall (↑)	
baseline	0	0	4.1	±2.90	91.5	±0.68	87.58	±0.16	37.5	±0.6
remove punct	10	0	**0**	+	91.3		87.63		37.8	
	20	0	**0**	+	**93.5**	+	87.28	−	36.4	−
	50	0	**0**	+	91.0		87.52		38.0	
	100	0	**0**	+	91.9		86.83		37.8	
add NP	0	10	**1**	+	**93.1**	+	**88.21**	+	37.2	
	10	10	**0**	+	**92.7**	+	87.67		36.8	
	20	10	**0**	+	**93.1**	+	87.28	−	37.6	
	0	20	**1**	+	**92.8**	+	**88.02**	+	37.8	
	0	50	**1**	+	**94.2**	+	87.37	−	**38.4**	+
	50	50	**0**	+	**94.4**	+	87.52		38.0	
	0	100	**0**	+	**94.7**	+	87.59		**38.2**	+

Table 6: Spanish results. See the caption of Table 4 for details.

The addition of noun phrases had larger impacts in all metrics. When the noun phrases were added ($p = 0$, $n > 0$), NP detection ratio was improved in all four languages, and it was consistently increased with n. Considering that the noun phrases extracted from the UD corpora and those in the Wikipedia section data are independent, we can say that the addition of noun phrases to the training corpora has a positive impact on the analysis of noun phrase inputs generally. There were cases

			Section title				UD		Sentiment	
	p	n	Wrong punct (↓)		NP detection (↑)		LAS (↑)		Recall (↑)	
baseline	0	0	0.7	±0.67	91.6	±0.63	83.84	±0.14	48.9	±0.9
remove punct	10	0	**0**	+	91.7		83.81		47.6	−
	20	0	**0**	+	91.1		**84.06**	+	49.2	
	50	0	2		91.4		**84.03**	+	49.4	
	100	0	**0**	+	90.1	−	83.46	−	49.2	
add NP	0	10	1		**93.9**	+	**84.09**	+	49.0	
	10	10	**0**	+	**94.2**	+	83.71		49.0	
	20	10	1		**93.7**	+	83.96		49.6	
	0	20	**0**	+	**94.6**	+	83.91		49.6	
	0	50	**0**	+	**95.3**	+	83.88		48.6	
	50	50	**0**	+	**95.4**	+	83.75		47.6	−
	0	100	**0**	+	**95.3**	+	**84.00**	+	48.8	

Table 7: English results. See the caption of Table 4 for details.

that were not detected as nouns even for $n = 100\%$, but a number of remaining errors were due to automatic noun phrase extraction from Wikipedia section titles.

The addition of NPs reduced the punctuation errors as well, even without explicit removal of punctuation (e.g. $p = 0$ cases). This is because the noun phrases added to the corpus did not have end-puncts, and thus, it helped models avoid bias to corpora consisting of POUs.

Although these treatments for noun phrase inputs obviously made positive impacts to the Wikipedia section title data, there is a potential risk of damage to the existing benchmarking. In the results of the LAS score in the UD test corpora, the decrease in general dependency parsing performance was observed in a number of cases with high ratios of p and n, but in most of cases, LAS scores were equal to or better than the baseline settings.

Because our motivation in this work is to build a robust parser for real-world applications, an extrinsic evaluation should be a main focus. In French, German and Spanish, recall scores in sentiment detection were increased with a moderate ratio of end-punct removal or NP addition, even though the optimal ratio of p and n varies by languages.

In English, the sentiment detection was not improved from the baseline. These results can be supported by the observation in Section 4: UD_English-EWT data contains NPUs and POUs with higher ratios compared to other corpora, and the English version of SemEval data was highly controlled with formal sentences without NPUs and POUs, and thus, our approach to corpus expansion did not work for this settings, but it is notable that negative impacts were limited as well.

7. Conclusion

This paper presented methods to make robust PoS taggers and dependency parsers to inputs for real-world applications by reducing the discrepancy of the ratios of noun phrases and punctuation omitted units between the training corpora and expected input documents. In addition to the removal of punctuation, which has been attempted to build more consistent models, we added noun phrases to the training corpus by automatically extracting noun phrases from existing annotations using syntactic operations. The experimental results showed that retraining on the extended training corpora made positive impacts on all three experiments simultaneously; a unit test for noun phrases, intrinsic evaluation of the dependency parser, and extrinsic evaluation of it on sentiment detection. The selection of the optimal values in the corpus expansion (ratios of punctuation removal and noun phrase addition) is our future work.

In this paper we handled multiple European languages where the definition of noun phrases and punctuation is relatively easy. In other languages, the structure of noun phrases is more diverse and complicated, and thus, more linguistic discussion and empirical studies will be needed. We applied the proposed technique to the UD corpora, but this can be integrated with the corpus augmented method using raw corpora (El-Kurdi et al., 2020), so that more applicable syntax analyzers can be developed.

The results of our experiments suggest that the current UD corpora are not perfect to train models for practical syntactic analyzers, and that it is important to know the characteristics of corpora and input documents to analyze, and to adjust the corpora to generate better models not just for the benchmarking on UD, but also for the practical use cases.

8. Bibliographical References

Mathieu Dehouck and Carlos Gómez-Rodríguez. 2020. Data augmentation via subtree swapping for dependency parsing of low-resource languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3818–3830, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yousef El-Kurdi, Hiroshi Kanayama, Efsun Sarioglu Kayi, Vittorio Castelli, Todd Ward, and Radu Florian. 2020. Scalable cross-lingual treebank synthesis for improved production dependency parsers. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 172–178, Online. International Committee on Computational Linguistics.

Nathan Green. 2011. Effects of noun phrase bracketing in dependency parsing and machine translation. In *Proceedings of the ACL 2011 Student Session*, pages 69–74, Portland, OR, USA. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Ran Iwamoto, Hiroshi Kanayama, Alexandre Rademaker, and Takuya Ohko. 2021. A Universal Dependencies corpora maintenance methodology using downstream application. In *Proceedings of the Third Workshop on Computational Typology and Multilingual NLP*, pages 23–31, Online. Association for Computational Linguistics.

Ran Iwamoto, Issei Yoshida, Hiroshi Kanayama, Takuya Ohko, and Masayasu Muraoka. 2023. Incorporating syntactic knowledge into pre-trained language model using optimization for overcoming catastrophic forgetting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10981–10993, Singapore. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

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 (LREC 2020)*, 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 (EMNLP-IJCNLP)*, pages 2779–2795.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Zuchao Li, Junru Zhou, Hai Zhao, and Rui Wang. 2019. Cross-domain transfer learning for dependency parsing. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 835–844. Springer.

Preslav Nakov and Marti Hearst. 2005. Search engine statistics beyond the n-gram: Application to noun compound bracketing. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 17–24, Ann Arbor, Michigan. Association for Computational Linguistics.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

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)*, Portorož, Slovenia.

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 (LREC 2020)*, pages 4034–4043, Marseille,

France. European Language Resources Association.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal Dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 86–95, Gothenburg, Sweden. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.

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.

Josef Ruppenhofer, Roman Klinger, Julia Maria Struß, Jonathan Sonntag, and Michael Wiegand. 2014. IGGSA shared tasks on German sentiment analysis (GESTALT). In *Workshop Proceedings of the 12th Edition of the KONVENS Conference*, pages 164–173, Hildesheim, Germany. Universität Heidelberg.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 2897–2904.

Anders Søgaard, Miryam de Lhoneux, and Isabelle Augenstein. 2018. Nightmare at test time: How punctuation prevents parsers from generalizing. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 25–29.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Jörg Tiedemann and Zeljko Agic. 2016. Synthetic treebanking for cross-lingual dependency parsing. *Journal of Artificial Intelligence Research*, 55:209–248.

Takuma Udagawa, Hiroshi Kanayama, and Issei Yoshida. 2023. Sentence identification with BOS and EOS label combinations. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 343–358, Dubrovnik, Croatia. Association for Computational Linguistics.

David Vadas and James R. Curran. 2011. Parsing noun phrases in the Penn Treebank. *Computational Linguistics*, 37(4):753–809.

Daniel Zeman, Filip Ginter, Jan Hajič, Joakim Nivre, Martin Popel, and Milan Straka. 2018. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium.