Dependency parsing of learner English

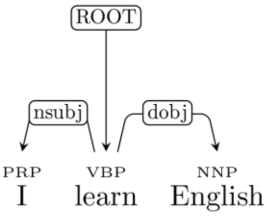
Yan Huang[i], Akira Murakami[ii], Theodora Alexopoulou[i] and Anna Korhonen[i]
[i] University of Cambridge / [ii] University of Tübingen

Current syntactic annotation of large-scale learner corpora mainly resorts to “standard parsers” trained on native language data. Understanding how these parsers perform on learner data is important for downstream research and application related to learner language. This study evaluates the performance of multiple standard probabilistic parsers on learner English. Our contributions are three-fold. Firstly, we demonstrate that the common practice of constructing a gold standard – by manually correcting the pre-annotation of a single parser – can introduce bias to parser evaluation. We propose an alternative annotation method which can control for the annotation bias. Secondly, we quantify the influence of learner errors on parsing errors, and identify the learner errors that impact on parsing most. Finally, we compare the performance of the parsers on learner English and native English. Our results have useful implications on how to select a standard parser for learner English.

Keywords: dependency parsing, learner English, annotation bias, parsing accuracy, learner error

1. Introduction

Researchers are often interested in retrieving syntactic information from learner corpora. In particular, dependency structure is gaining increasing attention and has been annotated for many learner corpora (Berzak et al. 2016b, Dickinson & Lee 2013, Dickinson & Ragheb 2009, Geertzen et al. 2013, Krivanek & Meurers 2011, Ott & Ziai 2010, Ragheb & Dickinson 2011). Dependency structure defines pairwise syntactic relations between words: each relation defines the dependence of a word on the other, i.e. the head, as illustrated in Figure 1.



Id	Word	POS	Head Id	Dependency
1	I	PRP	2	nsubj
2	learn	VBP	0	root
3	English	NNP	2	dobj

Figure 1. The dependency structure of an example sentence

Since manual annotation is costly and even impractical for large corpora, automatic parsers are increasingly used to annotate learner corpora (Geertzen et al. 2013, Granger et al. 2009, Tono & D ėz-Bedmar 2014). Due to the absence of parsers specifically developed for learner data, standard parsers developed for native language data are used. However, the performance of such parsers on learner data has not been investigated systematically. As a result, many corpus linguists refrain from using standard parsers on learner corpora, which partly explains why the number and scope of syntactic studies based on learner corpora is limited (Paquot & Plonsky 2017, Rankin 2015). Furthermore, for those who venture to use a parser, there is no guidance as for which parser, among the many parsers that are available, should be chosen for learner data.

This study provides a systematic evaluation of standard parsers on learner data. In particular, we compare the accuracy scores of multiple dependency parsers on learner English, and evaluate the effect of learner errors on the parsing performance. During the evaluation, we also investigate whether the gold standard constructed by manually correcting the output of a single parser introduces significant bias to the evaluation results. Furthermore, we investigate whether the performance of the standard parsers on native English can predict their performance on learner English.

2. Automatic dependency annotation of learner language

The accuracy of a dependency parser is usually measured by the unlabeled attachment score (UAS) and the labeled attachment score (LAS) (Buchholz & Marsi 2006). UAS refers to the percentage of words that have correct head indices, whereas LAS refers to the percentage of words whose head indices and dependency labels are both correct. Previous research has shown that standard dependency parsers can achieve up to

92.1% UAS and 89.6% LAS on learner English (Geertzen et al. 2013), and 86.4% UAS and 79.3% LAS on learner German (Krivanek & Meurers 2011, Ott & Ziai 2010).

However, the parsing accuracy of learner English is evaluated on a gold standard obtained by manually correcting the pre-annotation of the same parser. The human annotation may have been biased towards the pre-annotation, which can inflate the accuracy scores. Annotation bias has been shown to artificially increase inter-annotator agreement and reduce the annotation quality of part-of-speech (POS) tags and dependency structure on native English and upper-intermediate learner English (Berzak et al. 2016a). It is important to investigate the extent to which human annotation bias is present and its potential impact on parser evaluation on learner data across all proficiency levels.

Meanwhile, no study has compared the performance of different parsers on learner English, and there has been no systematic investigation into the effect of fine-grained learner errors on the performance of standard parsers on learner English. Some investigation have been made on German parsers: Krivanek & Meurers (2011) compare two parsers on learner German, and find that the rule-based *WCDG* parser perform better in identifying core predicate-argument relations, while the probabilistic *MaltParser* is better in establishing adjunct relations; Ott & Ziai (2010) qualitatively observe that for learner German, the omission of verbs is detrimental to parsing performance, whereas learner errors on agreement or word order seldom cause parsing errors. Nevertheless, no attempt has been made to investigate the correlation between the performance of a parser on native language and learner language.

3. Data

We used data from the EF-Cambridge Open Language Database (EFCAMDAT; Geertzen et al. 2013) as our learner data. EFCAMDAT is an open access corpus containing more than 47 million words written by over 109,000 learners. The writings cover 128 different topics, most of which are narratives, such as “Writing about what you do”. The learners span across 16 proficiency levels, covering the whole range of language proficiency defined in Common European Framework of Reference for Languages **(**CEFR) (Council of Europe 2001). Moreover, the learners come from 188 countries and autonomous territories. The wide range of proficiency levels and nationalities makes EFCAMDAT an appropriate data source to study the general accuracy of standard parsers on learner English.

More specifically, we adopted the dataset used by Geertzen et al. (2013), which contained 1,000 sentences (11,067-word tokens) from EFCAMDAT. Geertzen et al. (2013) extracted the dataset by automatically segmenting EFCAMDAT into sentences, and pseudo-randomly sampling the sentences with equal representation from all 16 proficiency levels and five of the best-represented nationalities (i.e. Chinese, Russian, Brazilian, German, and Italian). Nevertheless, some sentences in the original dataset contained segmentation errors. To prevent these segmentation errors from introducing artificial learner errors into the sentences, we manually corrected the segmentation of these sentences. In sum, 68 sentences were changed, which led to an increase of the word tokens to 12,003.

We used the Penn Treebank of Wall Street Journal (PTB-WSJ) (Marcus et al. 1993) as our native English data. This dataset has been widely used in the field of natural language processing (NLP) to train standard parsers for English. In particular, we used Sections 2-21 of PTB-WSJ as the native English training data for some of the parsers we evaluated (see Section 4.1.2 for more details), and the Section 23 of PTB-WSJ for parser evaluation.

4. Method

Our study consists of three parts. Firstly, we evaluate the accuracy of multiple dependency parsers on learner English. During this evaluation, we also investigate the potential of annotation bias and its impact on the evaluation results. In the second part, we investigate the effect of learner errors on dependency parsing. Finally, in the third part, we compare the accuracy scores of the parsers on learner English and native English. This section presents the research design of our studies.

4.1 Parser evaluation on learner English

Our first study seeks to address two questions: (i) what is the accuracy of standard parsers on learner English? (ii) Is there annotation bias in the gold standard created by manually correcting the output of a single parser? If there is, how does the annotation bias influence the accuracy scores?

We designed a two-round annotation procedure. Firstly, the dependency structure of a learner English dataset was annotated by manually correcting the output of a single parser. We refer to this parser as the pre-annotation parser, and the manual annotation as the single-parser-based (SPB) annotation throughout the rest of this paper. Secondly, the SPB annotation was compared to the output of several other parsers and, where differences existed, the SPB annotation was reviewed (see details below). The reviewed annotation is hereafter referred to as the multiple-parser-based (MPB) annotation. We then evaluated the parsers on both annotations. We consider the MPB annotation to represent the accurate annotation of the learner data, whilst the comparison of accuracy scores on the MPB and SPB annotations showed whether annotation bias existed and influenced the parser evaluation.

The following sections introduce the dependency scheme, parsers, annotators and annotating procedure.

4.1.1 *Dependency scheme*

We used Stanford typed dependencies (SD) (De Marneffe & Manning 2008), the most widely-used dependency scheme for English in the field of NLP. SD includes dependency relations for loose structures (e.g. “parataxis”, “discourse”) and words that are erroneously separated (“goes-with”). These relations are useful for describing learner errors. For example, when *furthermore* is misspelled as *further more*, *more* can be annotated as being headed on *further* in a dependency relation of “goes-with”.

Our SD scheme varied slightly from that of Geertzen et al. (2013). Firstly, our scheme was newer and included the dependency relations of “discourse” and “goes-with”. Secondly, Geertzen et al. (2013) use the default setting of SD, which treated copulas as the dependents of their complements. This caused inconsistency in representing the dependency relations between verbs and their complements (e.g. *flowers* is regarded as a complement in *they look like flowers*, but the root in *they are flowers*). Contrastingly, we treated copulas as the heads of their complements (i.e. *flowers* is still a complement in *they are flowers*).

4.1.2 *Parsers*

Since rule-based parsers require extensive human effort to define rules and their parsing schemes are difficult to change, our evaluation focused on probabilistic parsers. A probabilistic parser computes the most likely parse of a sentence according to a statistical syntactic model which associates syntactic rules with probabilities. The statistical model is trained on a corpus of syntactic structures. As such, the probabilistic parser can be tailored to the parsing scheme of the training corpus.

Currently there are two ways to obtain SD automatically. The first one, called 'c-parsing' (Kong & Smith 2014), converts the output of a constituency parser to dependency relations by definitive rules (De Marneffe et al. 2006). The other one, called 'd-parsing', extracts the dependency relations directly. We tested three constituency parsers for c-parsing and two dependency parsers for d-parsing. These parsers were chosen because they are well-known and frequently used in NLP (Cer et al. 2010, Kong & Smith 2014). Moreover, we tested two different settings for each of the two constituency parsers. As a result, seven different parsing settings were tested in total. The constituency parsers are as follows:

i. *Stanford* parser, version 3.5.1: We tested two ready-made syntactic models, both of which had been trained on a number of treebanks (See http://nlp.stanford.edu/software/parser-faq.shtml) in addition to PTB-WSJ Sections 2-21. The first syntactic model (hereafter referred to as SU) followed a probabilistic context free grammar (PCFG) (Klein & Manning 2003a), whilst the second model (SL) followed a lexicalized PCFG which integrates head words into the syntactic rules (Klein & Manning 2003b). Since Geertzen et al. (2013) show that the SU parser setting achieved high accuracy on learner data, we selected SU as the pre-annotation parser for the construction of the gold standard (see Section 4.1.4 for the annotation procedure), and provided the POS tags produced by this parser setting to other parsers which require the input of POS tags;

ii. *BLLIP* parser (Charniak & Johnson 2005), the latest version retrieved from the official repository on March 25, 2015: We tested two ready-made syntactic models trained on different datasets – the first one (BS) on OntoNotes-WSJ and the Google Web Treebank; the second one (BW) on PTB-WSJ and about two million sentences from Gigaword.

iii. *Berkeley* parser, version 1.7 (Petrov & Klein 2007) (BK): We used a ready-made syntactic model called 'eng_sm6', which had been trained on PTB-WSJ Sections 2-21.

The constituency structures produced by the aforementioned parsers were converted to collapsed SDs using the Stanford typed dependency converter (version 3.5.1) (De Marneffe & Manning 2008). The converter required the constituency structures in the Penn Treebank (PTB) format. Since the POS tags of auxiliary verbs (AUX) in the constituency output of *BLLIP* parser differed from the PTB format, we replaced these POS tags with their counterparts produced by the pre-annotation parser SU.

For d-parsing, we used *Turbo* parser version 2.1.0 (Martins et al. 2013) (TB) and *Maltparser* version 1.8 (Nivre et al. 2007) (MT). We converted Sections 2-21 of PTB-WSJ to the basic SD format, and trained both dependency parsers with default settings on the dataset. When training the *Maltparser*, we followed the feature template used in the ready-made 'engmalt' model. Since these dependency parsers contained no POS taggers, we provided the POS output of the pre-annotation parser SU to these parsers during the evaluation. The original outputs of these dependency parsers used basic SD format. They were converted to collapsed SD using the converter (De Marneffe & Manning 2008) again.

4.1.3 *Annotators*

Two PhD students in Linguistics participated in the annotation of dependency structure (Section 4.1.4) and learner errors (Section 4.2.2). They independently annotated 30 sentences for training, and 200 sentences for calculating inter-annotator agreement. It turned out that their inter-annotator agreement on both annotation tasks was sufficiently high (see Section 4.1.4), which means that the two annotators were consistent and the annotation was reliable. As a result, only one annotator continued to annotate the rest of the learner dataset, i.e. the remaining 770 learner sentences.

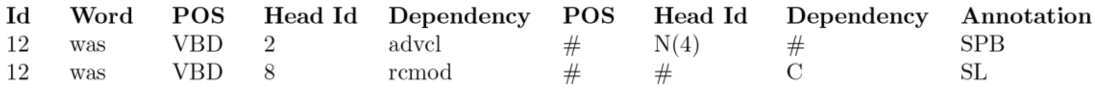
4.1.4 *Annotating procedure*

The training process for dependency annotation is as follows. First, the annotators learned the PTB annotation guideline (Santorini 1990) and the Stanford typed dependencies manual (De Marneffe & Manning 2008). They then independently annotated 30 sentences randomly selected from the learner dataset. During the annotation, the annotators can consult the converted dependency relations from

PTB-WSJ. The two annotators then discussed and resolved their annotation disagreement on the 30 sentences.

Two annotations were produced for each sentence. First, the annotators corrected the output of the pre-annotation parser SU to generate a single-parser-based (SPB) annotation. During this annotation, the annotators had access to the gold standard of Geertzen et al. (2013) for reference. Despite some aforementioned differences in the sentences and the annotation schemes, the gold standard of Geertzen et al. (2013) provided additional human annotation information that may help to improve the annotation accuracy. The annotators could also check the context of the sentence, i.e. the learner essay that contained the sentence.

After completing the SPB annotation, the annotators generated a multiple-parser-based (MPB) annotation by reviewing the SPB annotation according to alternative annotations provided by the other parsers. Specifically, we extracted the words where the outputs of at least one of the other six parser settings disagreed with the pre-annotation parser SU (hereafter referred to as annotation mismatches), and displayed all the disagreements as well as the SPB annotation to the annotators. The annotators then re-annotated these cases. When an annotation (i.e. POS tag, head index or dependency label) of a parser setting was correct and that of the SPB annotation was incorrect, the correct annotation was marked with C (correction). When an annotation of a parser setting was different from that of the SPB annotation but both annotations were acceptable, the annotation provided by the parser setting was marked with M (multiple options). Furthermore, if all annotations were incorrect, the annotation of SPB was corrected and marked with N (non-replacement correction). We then generated the MPB annotation by substituting the annotations marked with C and N for their counterparts in the SPB annotation, and including the alternative annotations marked with M. Figure 2 illustrates the annotation procedure: SL annotated the head index as 8 and the dependency label as “advcl” (adverbial clause), while the SPB annotation annotates the head index as 2 and the dependency label as “rcmod” (relative clause). The annotator decided that both annotations of the head index were incorrect and that the correct head index was 4. He therefore marked the head index of SPB with N and provided the correct head index in parentheses (4). By contrast, the annotation of SL on the dependency label was correct while that of SPB was not, so the annotator marked the dependency label of SL with C.



Id	Word	POS	Head Id	Dependency	POS	Head Id	Dependency	Annotation
12	was	VBD	2	advcl	#	N(4)	#	SPB
12	was	VBD	8	rcmod	#	#	C	SL

Figure 2. The format of the re-annotation based on annotation mismatches

As mentioned earlier, the two annotators annotated another 200 sentences after the training. We measured the inter-annotator agreement on the MPB annotations. According to the kappa metric, the inter-annotator agreement on the annotation of POS tags, head indices and dependency labels was 0.961, which was similar to the inter-annotator agreement achieved by Geertzen et al. (2013) (0.971). Alternatively, according to the conventional parsing evaluation metrics, our inter-annotator agreements were 97.03% on POS accuracy, 94.46% on UAS, and 91.69% on LAS, which were close to those achieved by Ragheb & Dickinson (2013) (around 99% on POS accuracy, 97% on UAS and 95% on LAS; note that their scores are not directly comparable to ours due to differences in the annotation schemes). These results show that the inter-annotator agreement between our annotators was sufficiently high.

4.2 Investigating the impact of learner errors on parsing

To investigate the impact of learner errors on parsing, we first annotated the learner errors on our learner data and then investigated the correlation between the learner errors and the parsing errors of the pre-annotation parser SU. We then analyzed the parsing errors that were influenced most by learner errors and the learner errors that caused most parsing errors.

4.2.1 *Learner error scheme*

We used the learner error scheme of the Cambridge Learner Corpus (CLC-FCE) (Nicholls 2003). The scheme includes over 80 learner error types. The majority of the learner errors were defined along two dimensions: the deviation of the learner error from the target hypothesis and the syntactic category of the target hypothesis word. For example, the learner error “MV” represents a missing (M) verb (V). These two dimensions are most descriptive for learner errors (James 2013); combining them helps to achieve a fine-grained annotation scheme that allows for consistent annotation of learner errors.

In addition to the original taxonomy, we added two learner error types: “C” (Capitalization error) for capitalization errors, and “SP” (Space error) for wrongly split or concatenated words. In CLC-FCE, these two types of learner errors were somewhat inappropriately annotated as “RP” (punctuation needs replacement).

The annotation of learner errors follows the format of XML markup illustrated as follows:

I <ns type="TV"><i>graduate</i><c>graduated</c></ns> in 1983 .

where the erroneous sentence segment *graduate* was marked by <i>, while the target hypothesis *graduated* was marked by <c>; the learner error type was indicated by <ns type="TV">, which means wrong verb tense.

4.2.2 *Annotating learner errors*

The two annotators followed the procedure in Section 4.1.3 to annotate learner errors. Table 1 shows the kappa inter-annotator agreement (Rosen et al. 2014) of the learner errors that appeared at least five times on the 200 sentences.

Table 1. The kappa inter-annotator agreement of learner errors

Learner error	Kappa	Avg. # tags
S	0.897	44
C	0.877	21
MD	0.841	19
MT	0.787	17
MP	0.665	15
RP	0.623	15
RT	0.614	12
RD	0.699	10
AGV	1	10
UD	0.699	10
FN	0.823	9
MV	0.624	8
SP	0.705	7
FV	0.615	7

AS	-0.003	7
M	0.152	6
RA	0.909	6
MC	0.909	6
AGN	1	5
W	0.213	5
UT	0.889	5
RV	0.666	5
RJ	0.666	5

Table 1 indicates that most learner errors were annotated consistently, especially the spelling error (“S”), capitalization error (“C”), missing a determiner (“MD”), wrong form of a noun (“FN”), a pronoun needs replacing (“RA”), missing a conjunction (“MC”) and an unnecessary preposition (“UT”) ($\kappa > 0.8$) .

However, learner errors of incorrect argument structure (“AS”), something missing (“M”) and incorrect word order (“W”) are not consistent between the two annotators ($\kappa < 0.4$). Further analysis shows that these errors are subject to more varied target forms, and are therefore not easy to annotate in the same way among different annotators. The finding is similar to that of Rosen et al. (2013) on the annotation of learner Czech: they find that learner errors like incorrect morphology, whose target forms are easy to establish, can be annotated consistently, whereas learner errors like incorrect complex verb forms or wrong lexis cannot be annotated consistently due to varied target forms.

In general, the inter-annotator agreement was high, which showed that the two annotators were reliable in identifying learner errors. As a result, one annotator finished the rest of learner error annotation. This annotator also annotated the effect of learner errors on the parsing errors of all learner sentences (see the following section).

4.2.3 *Annotating the relation between learner errors and parsing errors*

We operationally defined that a learner error caused a parsing error if the removal of the learner error led to the disappearance of the parsing error. Since learner errors may jointly affect dependency parsing, it was important to annotate the effect of both individual and combined learner errors. However, the number of learner error combinations increases exponentially with the number of learner errors in a sentence.

For example, a sentence that contains 5 learner errors has $2^5 - 1$ (i.e. 31) combinations of learner errors. Observing whether the correction of these combinations leads to the disappearance of a parsing error is time-consuming. To limit the scale of our problem, we evaluated the effect of learner errors only on the pre-annotation parser SU.

The annotation procedure was as follows. We first extracted 344 sentences that contained both learner errors and parsing errors. Secondly, we corrected various combinations of the learner errors to produce partly or totally corrected sentences. Thirdly, we parsed the corrected sentences with SU. The annotator then annotated the effect of the learner errors in the following way: if the correction of a learner error combination resulted in the disappearance of a specific parsing error, the parsing error was annotated as related to all the learner errors in that combination. Only the minimum combination of learner errors was annotated; any other learner error combinations which included these learner errors and caused the disappearance of the same parsing error were not annotated. For the sentences that contained less than 6 learner errors (332 sentences), the annotator examined all their corrected sentences. It turned out that most learner error combinations which affected parsing errors involve fewer than four learner errors. As a result, the annotator examined only the correction of fewer than four learner errors for the remaining 12 sentences.

4.3 Comparison of dependency parsing on learner English and native English

We evaluated the parsers on the native English dataset and compared the results to the evaluation on the MPB annotation of the EFCAMDAT learner dataset. The gold standard of native dependency structures was achieved by converting Section 23 of the PTB-WSJ (Marcus et al. 1993) to the collapsed SD format.

5. Results

This section reports the evaluation results on the annotation bias, the impact of learner errors on the parsing performance of the baseline parser, and the comparison of parsing performance on learner English and native English.

5.1 Annotation bias on learner English

First, we evaluated the parsers against the SPB annotation. Table 2 shows the accuracy scores of the parsers. The accuracy was measured by the proportions of the words that received correct POS tags (POS), unlabeled attachments (UAS), labeled attachments (LAS), and the combination of POS tags and labeled attachments (All), as well as the proportions of the sentences that were free of the errors in each of the aforementioned aspects. The d-parsing parsers have no POS accuracy scores because they do not perform POS tagging. It turned out that the pre-annotation parser performed the best on all criteria. The maximum performance gaps between the parsers were smaller on POS tags than on dependency relations.

Table 2. The accuracy of the parsers on the SPB annotation

Parsing approach	**Parser**	**Accuracy by word (%)**				**Accuracy by sentence (%)**			
		POS	**UAS**	**LAS**	**All**	**POS**	**UAS**	**LAS**	**All**
c-parsing	SU	96.31	91.49	88.03	87.12	72.0	59.7	49.8	46.7
	SL	95.25	89.25	85.06	83.66	62.3	50.3	40.5	36.5
	BS	94.95	90.53	86.88	84.88	59.4	55.0	43.5	35.9
	BW	95.00	90.64	86.96	85.10	59.7	56.3	45.2	37.8
	BK	94.81	90.26	86.36	84.40	61.2	54.9	43.3	36.9
d-parsing	TB	--	89.88	86.32	--	--	54.1	43.0	--
	MT	--	88.38	84.67	--	--	48.4	38.5	--
Max. Diff.		1.50	3.11	3.36	3.46	12.6	11.3	11.3	10.8

The coincidence that the pre-annotation parser performed the best on the SPB annotation seems to suggest the presence of an annotation bias in the SPB annotation towards the pre-annotation parser. We then evaluated the parsers against the MPB annotation. The results (Table 3) confirmed the hypothesis about annotation bias. In this evaluation, the *BLLIP* parser turned out to be the best in all aspects except the sentence-based POS accuracy, on which the *Berkeley* parser performed the best. Specifically, BW, the parsing setting where the *BLLIP* parser was trained on Gigaword and PTB-WSJ, achieved the best results. On the other hand, the rank of the pre-annotation parser SU dropped to the third on the accuracy of word-based POS, the fifth on word-based UAS and LAS, and even the sixth on sentence-based UAS and LAS. The changes in the accuracy scores of the pre-annotation parser and the word-based accuracy scores of the best-performing parser between the two evaluations were significant according to chi-squared tests. These differences demonstrated that the SPB annotation was indeed biased towards the pre-annotation parser. The bias changed the ranking of the parsers, affecting the accuracy scores of the pre-annotation parser and the best-performing parser most.

Furthermore, the maximum performance gaps between the parsers diminished, especially on POS (from 1.50% to 0.40% on the word level, and from 12.6% to 2.1% on the sentence level). This means that the annotation bias in the SPB annotation also artificially increased the performance gaps between the parsers. In fact, the performance of various parsers on POS tagging was similar.

Table 3. The accuracy of the parsers on the MPB annotation*

Parsing approach	**Parser**	**Accuracy by word (%)**				**Accuracy by sentence (%)**			
		POS	**UAS**	**LAS**	**All**	**POS**	**UAS**	**LAS**	**All**
c-parsing	SU	95.41***	89.77***	86.05***	84.67***	64.7***	52.6**	42.5**	37.9***
	SL	95.38	89.70	85.46	84.06	62.6	53.7	42.9	36.9
	BS	95.63*	91.43*	87.77*	86.09**	63.7*	59.6*	47.7*	39.5
	BW	95.64*	91.53*	87.84*	86.28**	63.6*	60.5*	48.4	40.8
	BK	95.24	90.65	86.76	85.03	64.7	56.3	44.6	37.8
d-parsing	TB	--	90.53*	86.77	--	--	57.2	44.3	--
	MT	--	88.85	85.06	--	--	51.7	41.1	--
Max. Diff.		0.40	2.68	2.78	2.22	2.1	8.8	7.3	3.9

* The marks of significance (chi-squared tests): *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$.

To better understand the annotation bias, we quantitatively and qualitatively investigated the re-annotations that produced the MPB annotation. First, we identified cases where the annotation of a single parser disagreed with the pre-annotation parser SU and cases where the annotation of at least one parser disagreed with SU. We then further classified these cases into two groups: one where the SPB annotation agreed with the pre-annotation parser SU, and the other where the SPB annotation disagreed with SU. Table 4 shows the number of annotation mismatches with regard to each non-SU parser setting and the proportion of the cases that were marked with correction ("C") or multiple options ("M") due to the correct reference provided by that parser setting.

Table 4. The analysis of the annotation mismatches where the annotation of a parser disagreed with the pre-annotation parser SU

Cases	Parsing approach	Parser	POS			Head			Dependency label		
			#	C (%)	M (%)	#	C (%)	M (%)	#	C (%)	M (%)
SPB annotation agreed with SU	c-parsing	SL	252	19.8	5.2	668	15.1	5.4	556	6.8	4.7
		BS	362	22.7	6.4	584	20.9	8.4	424	13.7	7.8
		BW	352	22.7	6.8	544	22.4	8.1	409	13.4	8.3
		BK	332	18.4	6.6	581	17.0	5.0	450	11.6	4.2
	d-parsing	TB	--	--	--	530	20.8	7.7	383	12.0	5.5
		MT	--	--	--	709	14.1	5.2	525	9.0	2.1
	All parsers		647	16.4	6.3	1,644	11.7	5.5	1,351	7.3	4.3
SPB annotation disagreed with SU	c-parsing	SL	200	0.5	0.5	579	1.0	0.7	594	0.2	0.2
		BS	274	0.0	0.0	642	0.8	0.9	682	0.3	0.9
		BW	274	0.0	0.0	610	0.8	1.3	664	0.5	1.1
		BK	229	0.4	0.9	610	1.1	1.1	650	0.0	0.6
	d-parsing	TB	--	--	--	494	0.8	0.6	530	0.6	0.2
		MT	--	--	--	515	0.4	0.2	544	0.6	0.6
	All parsers		338	0.6	0.6	893	1.2	1.2	969	0.6	0.9

Table 4 shows that the correction rates on the cases where the SPB annotation agreed with SU were much higher than where they disagreed. In the former situation, around 20% of the annotation mismatches with respect to individual parsers on the POS tag and head index required corrections. The correction rate on the dependency label varied across different parser settings but also went beyond 10% in most cases. By contrast, the correction rates on the cases where the SPB annotation disagreed with SU dropped to less than 0.5% on POS tags, 1.1% on head indices and 0.6% on dependency labels for each parser setting. This contrast means that during the SPB annotation, the precision of correcting parsing errors was high (i.e. when a parsing error was corrected, the correction was accurate), but the recall of parsing errors was relatively low (i.e. the annotator accepted some wrong parsing choices during the SPB annotation).

The results also indicate that displaying the different output of various parsers helped the annotator to detect annotation errors. The contrast provided more information for reference during annotation, and helped to promote awareness of annotation errors. Nevertheless, since there was overlap in the correct references provided by different parsers (e.g. two or more parsers provided the same correct pre-annotation which led to the correction of a SPB annotation), the correction rates with respect to all parsers (i.e. the proportion of the annotation mismatches where at least one parser was correct) were generally lower than the correction rates with respect to individual parsers. This indicated that as the number of parsers adopted in the contrast-based annotation increased, the marginal benefit of adding a parser diminished.

We split the re-annotations with regard to the types of annotation, i.e. POS tags, head indices, and dependency labels, and summarized the linguistic structures that were prone to annotation bias.

5.1.1 *Annotation errors on POS tags*

Table 5 lists the types of POS annotation errors that occurred more than four times. Throughout this paper, we use the format of "wrong tag - correct tag" to refer to an annotation error or parsing error. Apart from the annotation error of "VBD-VBN" (past tense verb - past participle verb), all other annotation errors involved POS tag pairs which were listed as "easily confused" in the PTB annotation guideline (Santorini 1990). In other words, most of the annotation bias with respect to POS tags was related to choices between inherently confusing POS tag pairs. Further qualitative analysis revealed some prominent causes of the confusions as follows.

Table 5. Annotation errors on POS tags (named by "wrong tag-correct tag")

Error type	Freq.	Error type	Freq.
VB-VBP	21	RB-IN	5
RP-IN	11	IN-WDT	5
RP-RB	7	VBD-VBN	4
NNP-NN	7	VBG-NN	4
VBN-JJ	6	JJ-NN	4
RB-NN	6	IN-RB	4

i. Overlapping domain between inflectionally defined and functionally defined POS tags: for example, this factor contributed to the annotation errors of "VBN-JJ" (past participle verb - adjective), "VBG-NN" (gerund or present participle verb - singular or mass noun) and "VBG-JJ". VBN and VBG were defined by verbal inflection, whereas JJ and NN were defined by the function of words in context. The two sets of POS tags were not mutually exclusive. For instance, in *joy of learning*, *learning* can be either a noun or a verb.

ii. Overlapping domain between the POS tags in a containment relation: for example, annotation errors involving RP (particle) and RB (adverb) or IN (preposition) were related to this factor. RP was a subclass of RB and bore some functional characteristics of IN as well. Basically, RB or IN seemed plausible for many cases where RP was annotated. The PTB annotation guideline (Santorini 1990) defined rules and diagnostic tests for distinguishing between RP, RB and IN. However, these rules and tests may not apply to all cases. For instance, *two websites* […] *will be compared with** includes the redundant word *with*. According to the PTB annotation guideline, *with* should be tagged as IN. Nevertheless, the learner error here seems to indicate that the learner used *with* as an "RP".

iii. Same word forms with different POS tags: an example of the annotation errors related to this factor was "JJ-NN". If the word forms of a mass noun and an adjective were the same, confusion can occur when the words were used as prenominals or predicatives. For instance, in *plastic bottles* and *they are fun*, *plastic* and *fun* can be either NN or JJ.

POS annotation errors can also arise with ambiguous structures. For example, VB (base-form verb) was not usually confused with NN. However, in *go to work*, *work* can be regarded as a VB, with *to* as an auxiliary; on the other hand, *work* can also be regarded as a NN, with *to* as a preposition.

5.1.2 *Annotation errors on head indices*

The annotation errors regarding head indices mostly occurred in the following linguistic structures:

i. Prepositional phrases: in Example (1), the prepositional phrase *with the head* *teacher* can be regarded as dependent on *build*; this was syntactically acceptable and semantically plausible (i.e. the head teacher was involved in building up the cooperation). However, from the context we can see that the intended meaning of the construction was "to cooperate with the head teacher". Therefore, the prepositional phrase should be annotated as dependent on *cooperation* rather than *build*.

(1) I will build up a better cooperation with the head teacher which will ensure a better relation between those who make decisions and us students who are mainly affected by them.

ii. Modifiers: for a sequence of nouns where one heads the others, identifying which noun as the head can be challenging for an annotator. For example, for a locational phrase in the form of "city, country" like *in Manus, Brazil*, it is plausible to analyze the city as a modifier that specifies an area of the country. On the other hand, the comma between the two nouns can indicate post modification, in which the country is the modifier of the city.

iii. Coordinating conjuncts: the SD scheme determined that in a coordination the first conjunct should be the head of all the other conjuncts. However, if a parser failed to identify the first conjunct, attaching a conjunct to e.g. the second conjunct should not be regarded as a parsing error, because the conjunct relation was established. For instance, in Example (2), normally *raincoat*, *flash light*, *clothes*, and *sleeping bags* should be attached to *umbrella* with the dependency label of "conj_and". However, if a parser attached *flash light* to *raincoat* by "conj_and", the conjunct relation was also established and should not be regarded as a parsing error.

(2) I need to take my umbrella, my raincoat, my flash light, my clothes and my sleeping bag.

5.1.3 *Annotation errors on dependency labels*

The annotation errors on dependency labels were usually related to the following linguistic structures:

i. Linguistic structures subject to annotation errors on POS tags or head indices: since dependency parsing depends on POS tags, the errors in the latter may affect the former. For example, the dependency error of "amod-nn" (adjectival modifier - noun compound modifier) was based on the POS error of "JJ-NN". Similarly, "prt-prep" (phrasal verb particle - prepositional modifier) was based on "RP-IN", and "prt-advmod" (phrasal verb particle - adverb modifier) on RP-RB.

ii. Prepositional phrases and infinitive clauses (adjunct vs. complement): sometimes adjuncts and complements are difficult to disambiguate (Korhonen 2002), which led to the annotation error of "vmod-xcomp" (reduced non-finite verbal modifier - open clausal complement). For instance, in Example (3) the purpose clause *to raise fund* can also be taken as a complement clause in the absence of subcategorization information for individual verbs.

(3) I'd lead the student council to raise fund

iii. Conjuncts (multiple dependencies): the SD dependency scheme dictated that each word can have only one head. However, sometimes a word may be dependent on multiple words. This happened most frequently in conjunct structures. When a conjunct involved elliptical material, the dependents of the elided element may have multiple heads. For instance, in Example (4), *year* may be seen as either an object of the verb *have*, or a modifier of the word *warranty* at the end of the sentence. It was hard to choose between these two heads: choosing one dependency leads to the loss of information for the other dependency. Of course, it is worth noting that the ellipsis in Example (4) is ungrammatical, which adds to the annotation challenge.

(4) Our notebooks have a 1 year*, our pens two weeks warranty

Summarized above are the linguistic structures that were prone to annotation bias. Another major source of annotation bias is learner errors. For example, when we annotated the dependency relation between *help* and its head *hope* in *I hope this help* you*, two options seemed acceptable: if we assumed that the learner had confused the subcategorization frame of *hope* as that of *let* in *let somebody do something*, "xcomp" (open clausal complement) should be chosen; however, if we assumed that the learner used the right frame but made a mistake in the tense or number, "ccomp" (clausal complement) should be chosen. Furthermore, Example (5) shows a sentence that was unintelligible due to learner errors. There were many ways to interpret the sentence, each of which led to a different dependency structure. For example, if *demand* was intended to be *demanded*, *professional* should be annotated as “nsubjpass” (passive nominal subject), or if *is demand* was intended to be *demands*, *professional* should be annotated as “nsubj” (nominal subject).

(5) My professional is demand to one teach in the idiom English*.

In summary, this section has shown the accuracy of standard parsers on the EFCAMDAT learner data (Table 3). We also confirm the existence of annotation bias in the SPB annotation setting, and identify the linguistic structures that were prone to annotation bias. The next section moves on to evaluate the effect of learner errors on parsing.

5.2 Impact of learner errors on parsing errors

The reasonably high accuracy of the parsers on the learner data may create the impression that, after all, the learner errors do not have a significant impact on parser performance. It is crucial to understand if the parsers are indeed robust to learner errors.

We analyzed the overall effect of learner errors on parsing from two aspects: the proportion of the parsing errors that were caused by learner errors (hereafter referred to as “LE-caused PEs”), and the proportion of the learner errors that caused parsing errors. We then analyzed the effect of individual learner errors on parsing, summarizing the most frequent parsing errors and learner errors that are involved.

Table 6 shows the proportion of the parsing errors (PEs) that were caused by learner errors (LEs). Among the words that contained PEs, 39.2% had at least one LE-caused PE. Furthermore, when categorizing the PEs by the annotation types, we can see that the percentage of LE-caused PEs increased across POS tags (37.5%), head indices (40.4%) and dependency labels (43.2%). This means that dependency labels were most vulnerable to learner errors. A similar trend can be observed on the sentence level.

Table 6. Distribution of parsing errors caused by learner errors

Level	PE	# containing PEs	LE-caused PEs (%)
Sentence	General*	626	41.4
	POS	359	38.7
	Head index	478	40.2
	Dependency label	473	46.3
Word	General	1866	39.2
	POS	568	37.5
	Head index	1243	40.4
	Dependency label	1232	43.2

* In "General", a word was counted as containing a parsing error if any annotation type of the word, i.e. the POS tag, the head index or the dependency label, was incorrect.

Table 7 shows that 53.5% of the sentences contained at least one LE, and 63% of the LEs caused at least one PE. The high percentages of LE-caused PEs among PEs and LEs showed that learner errors had a great impact on the dependency parsing of learner English.

Table 7. Distribution of learner errors which caused parsing errors

Level	# (containing) LEs	(containing) LE-caused PEs (%)
Sentence	535	48.4
LE	1131	63.0

We now investigate the most frequent LE-caused PEs. Table 8 shows the types of LE-caused POS errors that occurred more than five times in our dataset. These POS errors made up 39.8% of all LE-caused POS errors.

One of the most frequent LE-caused POS errors was "JJ-NN" (adjective - noun). The causes of this parsing error included wrong derivation of nouns ("DN", e.g. in *some different*, the correct form of *different* should be *difference*s), missing a determiner ("MD", e.g. missing *a* in *I'm a pensioner*) or spelling errors ("S").

Another frequent LE-caused POS error was "NNP-NN" (proper noun - noun). The main causes were missing a determiner ("MD") at the beginning of a sentence or inaccurate capitalization of a common noun ("C").

Verbs were sometimes misrecognized as nouns ("NN-VB") because of erroneous argument structure ("AS", e.g. *are over love their own babies* should be corrected as *love their own babies very much*), missing a preposition ("MT", e.g.

missing *to* in *like play badminton*) or using a wrong verb form ("FV", e.g. *think about change my career*), etc.

The errors of misrecognizing proper nouns as common nouns ("NNP-NN") and pronouns as foreign words ("FW-PRP") were exclusively caused by capitalization errors. Specifically, "FW-PRP" was caused by using the lower-case *i* for the first-person singular pronoun *I*. Except for these two types of POS errors, most LE-caused POS errors involved varied learner errors.

Table 8. Most frequent types of LE-caused POS errors (named by "wrong tag-correct tag")

POS error	Most frequent relevant LEs	Freq.
JJ-NN	DN(4)*, MD(3), S(3), FA, UY, CN, AS	11
NNP-NN	MD(6), C(4), S, DN, W	11
NN-VB	AS(3), MT(2), FV(2), MD, MC, DN, DA	10
FW-PRP	C(8)	8
NN-NNP	C(7)	7
VBG-NN	S(2), FN(2), RP, UN, MD, MC, AS	7
NN-RB	S(3), RP(2), UT	6
NN-VBP	RP(3), DA(2), TV, FV	6
VB-VBP	S(3), RP(2), M	6
NNP-JJ	MD(3), W, C	5
RB-IN	S(2), UC, M, FV	5

* The bracketed numbers denote the frequencies of the PEs that were caused by a particular LE more than once. Note that a PE may be caused by more than one LE.

Table 9 shows the LE-caused dependency label errors that occurred more than five times in our dataset. These errors made up 28.1% of the LE-caused dependency label errors: compared to LE-caused POS errors, LE-caused dependency label errors were more varied.

The two most frequent types of dependency label errors concerned the core structure of the sentences: "ccomp-root" refers to misjudging a root as a clausal complement, and "root-parataxis" refers to misrecognizing a parataxis clause (i.e. a coordinate or subordinate clause without an explicit link verb) as a root. The major cause of these errors was comma splice. This learner error is marked by the learner error code "RP" (punctuation needs replacing), as the commas should be replaced by semi-colons or full-stops. Meanwhile, other dependency label errors had no dominantly related learner errors.

Table 9. Most frequent types of LE-caused dependency label errors (named by “wrong label-correct label”)

Dependency label error	Most frequent relevant LEs	Freq.
ccomp-root	RP(21), MP(3), S(2), UA, MC, FV, DA, CN	29
root-parataxis	RP(24), MP(3), MC, DV	29
nsubj-dobj	AS(4), MP(3), RP(2), MC(2), C	13
amod-nn	S(3), FN(2), C(2), UN, RP, MD, MC, AS	12
dep-parataxis	MP(4), AS(4), RP(3), DA(2), C(2), RT, CE	10
appos-conj_and	MC(3), RC(2), AS	6
vmod-root	RP(4), SP, MV	6
nn-amod	MD(2), W, S, RJ, C	6
advmod-erased	S(3), W, UN, M, FV, AS	6
root-erased	RP(2), MD, M, AS	5
aux-root	W, S, RP, MD, C, AGV	5
dep-dobj	UV, RP, M, DV, DA, C, AS	5
nn-conj_and	AS (2), SP, S, RC, MT	5
rcmod-parataxis	MP(3), RP(2)	5
acomp-xcomp	MD(3), MA, RC	5
root-aux	UV, UT, UA, S, FV	5

We ranked the learner errors according to the frequency of the parsing errors they caused. Table 10 shows the top learner error types. It turned out that erroneous punctuations caused most parsing errors. This can also be observed from Table 8 and Table 9. Apart from the comma splice, another major punctuation error was substituting a backtick (`) for an apostrophe in the contracted forms of verbs (e.g. *I’m*, *I’ve*), negations (e.g. *don’t*), and the possessive form of nouns (e.g. *Asia’s*), which caused problems including misjudging present tense verbs as common nouns (“NN-VBP”) and misjudging the possessive morpheme *’s* as a root (“root-erased”).

Table 10. Learner errors that caused parsing errors most frequently

LE	Description	# LE-caused PEs
RP	Punctuation error	100
S	Spelling	76
C	Capitalization	55
AS	Wrong argument structure	54
MP	Missing a punctuation	47
MD	Missing a determiner	44

To conclude, this section has confirmed that learner errors do have impact on dependency parsing. The question then is why the parsers still achieve high performance if learner errors do have a significant impact. We turn to this issue in the next section where we compare the performance of the parsers on learner and native data.

5.3 Parser performance on learner English and native English

Table 11 presents the evaluation results of the standard parsers on learner English (MPB annotation) and native English (PTB-WSJ Section 23) on the word level. The accuracy scores of each standard parser were significantly lower ($p < 0.001$ according to chi-squared tests) on learner English than on native English. On average, the parsers achieved 95.46% vs. 96.69% on POS accuracy, 90.35% vs. 92.48% on UAS, 86.53% vs. 90.09% LAS, and 85.23% vs. 88.66% on the accuracy of all tags. The average performance gap between learner English and native English increased across the POS tag (1.23%), unlabeled attachment (2.13%), and labeled attachment (3.43%). This indicates that compared to POS tagging, dependency parsing is subject to more influence from the difference between learner English and native English.

Even though the accuracy gaps between learner English and native English may seem small to human eyes, it does not mean that the parsers are robust to learner errors, as demonstrated in the previous section. Geertzen et al. (2013) argued that the seemingly high accuracy scores of the parsers on learner English might result from prevalence of short and simple sentences in learner English. To testify whether parsers perform better on shorter sentences than on longer ones, we grouped the native English sentences by sentence length, calculating the average parsing accuracy scores of each group that had more than five sentences, and computing the Pearson correlation between the accuracy scores and sentence length. It turned out that the UAS and LAS were significantly and negatively correlated with the sentence length (UAS: $r = -0.776$, $p < 0.01$; LAS: $r = -0.603$, $p < 0.01$). This means that the performance of dependency parsing was indeed better on shorter sentences. Since the average sentence length of our learner English dataset was 13.5 whereas that of the native one was 23.5, the UAS and LAS gaps between learner English and native English have been partly offset by the differences in sentence length. Nevertheless, POS tagging showed a positive correlation with sentence length ($r = 0.415$, $p < 0.01$); careful examination shows that this was because POS tagging was already quite accurate; when few POS errors occurred, shorter sentences had fewer words in total, which dragged down their POS accuracy scores.

Table 11. The accuracy of the parsers on the learner data and the native data

Parsing approach	**Parser**	**MPB annotation**				**PTB-WSJ section 23**			
		POS	**UAS**	**LAS**	**All**	**POS**	**UAS**	**LAS**	**All**
c-parsing	SU	95.41	89.77	86.05	84.67	96.37	90.70	88.11	86.49
	SL	95.38	89.70	85.46	84.06	96.65	90.98	88.16	86.61
	BS	95.63	91.43	87.77	86.09	96.71	94.09	91.89	90.12
	BW	95.64	91.53	87.84	86.28	96.76	94.22	92.08	90.33
	BK	95.24	90.65	86.76	85.03	96.98	93.44	91.32	89.76
d-parsing	TB	--	90.53	86.77	--	--	92.67	90.20	--
	MT	--	88.85	85.06	--	--	91.26	88.85	--
Average		95.46	90.35	86.53	85.23	96.69	92.48	90.09	88.66
Max. Diff.		0.40	2.68	2.78	2.22	0.61	3.52	3.97	3.84

On the other hand, the performance of the parsers on learner English seemed to correlate with their performance on native English. The best parser setting for learner English, the *BLLIP* parser trained on PTB-WSJ and Gigaword, also performed the best on native English except on POS tagging where it came second following the *Berkeley* parser. To verify the correlation, we ranked the parsers according to their performance on each dataset and computed the Spearman’s rho correlation between the two rankings. It turned out that the correlation was significant on UAS ($r = 0.857$, $p < 0.05$), LAS ($r = 0.821$, $p < 0.05$) and the combination of all tags ($r = 0.900$, $p < 0.05$). Nevertheless, there was no significant correlation between the rankings on POS tags alone; this was possibly because the performance of the parsers on POS tagging was similarly high which made the ranking on the POS tag less meaningful.

The aforementioned correlation between the performance of dependency parsing on learner English and native English seems to contradict the result of Krivanek & Meurers (2011), who show that the *MaltParser* performed better on native German but worse on learner German than the *WCDG* parser. However, their study compares only two parsers, which makes it impossible to identify a reliable correlation between the performance on learner data and native data. Furthermore, the study compared a rule-based parser to a probabilistic parser, whereas our study compared a number of probabilistic parsers. Last but not least, they investigated learner German. German has a different word order and morphological cues on nouns and verbs compared to English; as a result, the impact of learner errors on the dependency parsing of German may well be different. Nevertheless, based on our study, we can safely conclude that the performance of a probabilistic parser on native English can predict its performance on learner English.

6. Conclusion

Our study showed that annotation bias exists when a human annotator generates an annotation by correcting the output of a parser. This annotation bias arises from the inherent ambiguity of some linguistic structures, the annotation schemes and learner errors. The annotation bias reduces the recall of parsing errors during annotation; using a gold standard that contains the annotation bias can significantly influence the result of parser evaluation in favor of the pre-annotation parser.

The annotation bias may be controlled in several ways. Firstly, we can adopt a contrast-based annotation method, presenting annotators with mismatches between several parsers. Secondly, we can improve the annotation scheme for parsing. In particular, we need principles that can help to distinguish the ambiguity arising from learner errors. Multi-layered annotation (Dickinson & Ragheb 2009) which uses different layers of features to describe the contradictory aspects of learner errors may be a way forward. Nevertheless, our results indicated that learner errors may lead to ambiguity where many interpretations of the structure are possible. This ambiguity poses a challenge to the design of appropriate layers for annotation.

We also showed that learner errors do have an impact on parsing output. More than one third of the parsing errors were caused by learner errors, and over 60% of the learner errors caused at least one parsing error. These results indicate that the parsers are not very robust to learner errors. Learner errors on punctuation, spelling, capitalization, argument structures, determiners and prepositions caused most parsing errors. Correcting these learner errors will be an effective pre-processing technique to reduce parsing errors for downstream NLP application on learner English.

Given the impact of learner errors on parsing, it is surprising that the accuracy scores of the parsers on learner English are lower than those on native English by only small margins. We showed that this is because the average sentence length of learner English is shorter than that of native English. In other words, the impact of learner errors is offset by the simplicity of learner language.

Finally, we demonstrated that the performance of probabilistic parsers on learner English can be predicted by their performance on native English. This implies that when we want to choose a probabilistic parser for learner English, the most accurate parser on native English can be a good candidate.

Reference

Berzak, Y., Huang, Y., Barbu, A., Korhonen, A., & Katz, B. (2016). Anchoring and agreement in syntactic annotations. In J. Su (Ed.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 2215–2224). Austin, TX: ACL.

Berzak, Y., Kenney, J., Spadine, C., Wang, J. X., Lam, L., Mori, K. S., Garza, S., & Katz, B. (2016). Universal dependencies for learner English. In K. Erk & N. A. Smith (Eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 737–746). Berlin: ACL.

Buchholz, S., & Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. In L. Marquez & D. Klein (Eds.), *Proceedings of the Tenth Conference on Computational Natural Language Learning* (pp. 149–164). New York, NY: ACL.

Cer, D. M., De Marneffe, M.-C., Jurafsky, D., & Manning, C. D. (2010). Parsing to Stanford dependencies: Trade-offs between speed and accuracy. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, & D. Tapias (Eds.), *Proceedings of the Seventh International Conference on Language Resources and Evaluation* (pp. 1628–1632). Valletta: ELRA.

Charniak, E., & Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In K. Knight (Ed.), *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 173–180). Stroudsburg: ACL.

Council of Europe. (2001). *Common European Framework of Reference for Languages:*

Learning, Teaching, Assessment. Cambridge: Cambridge University Press.

De Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In N. Calzolari, K. Choukri, A. Gangemi, B. Maegaard, J. Mariani, J. Odijk, & D. Tapias (Eds.), *Proceedings of the Fifth International Conference on Language Resources and Evaluation* (pp. 449–454). Genoa: ELRA.

De Marneffe, M.-C., & Manning, C. D. (2008). *Stanford typed dependencies manual* (Technical Report). Retrieved from https://nlp.stanford.edu/software/dependencies_manual.pdf (last accessed February 2018).

Dickinson, M., & Lee, C. M. (2013). Modifying corpus annotation to support the analysis of learner language. *CALICO Journal*, *26*(3), 545–561.

Dickinson, M., & Ragheb, M. (2009). Dependency annotation for learner corpora. In M. Passarotti, A. Przepiorkowski, S. Raynaud, & F. Van Eynde (Eds.), *Proceedings of the Eighth Workshop on Treebanks and Linguistic Theories* (pp. 59–70). Milan: EDUCatt.

Geertzen, J., Alexopoulou, T., & Korhonen, A. (2013). Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge Open Language Database (EFCAMDAT). In R. T. Miller, K. I. Martin, C. M. Eddington, A. Henery, N. M. Miguel, A. Tseng, A. Tuninetti & D. Walter (Eds.), *Proceedings of the 31st Second Language Research Forum: Building Bridges Between Disciplines*. Somerville: Cascadilla Proceedings Project.

Granger, S., Dagneaux, E., Meunier, F., & Paquot, M. (2009). *The International Corpus of Learner English. Version 2. Handbook and CD-ROM*. Louvain-la-Neuve: Presses Universitaires de Louvain.

James, C. (2013). *Errors in Language Learning and Use: Exploring Error Analysis*. New York, NY: Addison Wesley Longman.

Klein, D., & Manning, C. D. (2003a). Accurate unlexicalized parsing. In E. W. Hinrichs & D. Roth (Eds.), *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1* (pp. 423–430). Sapporo: ACL.

Klein, D., & Manning, C. D. (2003b). Fast exact inference with a factored model for natural language parsing. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in Neural Information Processing Systems 15* (pp. 3–10). Cambridge, MA: MIT Press.

Kong, L. & Smith, N. A. (2014). An empirical comparison of parsing methods for stanford dependencies (arXiv preprint). Retrieved from https://arxiv.org/abs/1404.4314 (last accessed February 2018.

Korhonen, A. (2002). Semantically motivated subcategorization acquisition. In J. Pentheroudakis, N. Calzolari, & A. Wu (Eds.), *Proceedings of the ACL-02 workshop on*

Unsupervised lexical acquisition-Volume 9 (pp. 51–58). Philadelphia, PA: ACL.

Krivanek, J., & Meurers, D. (2011). Comparing rule-based and data-driven dependency parsing of learner language. In K. Gerdes, E. Hajičová, & L. Wanner (Eds.), *Proceedings of the First International Conference on Dependency Linguistics (Depling 2011)* (pp. 310–317). Barcelona: IOS Press.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330.

Martins, A. F. T., Almeida, M., & Smith, N. A. (2013). Turning on the Turbo: Fast third-order non-projective Turbo parsers. In H. Schuetze (Ed.), *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 617–622). Sofia: ACL.

Nicholls, D. (2003). The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In A. Dawn, P. Rayson, A. Wilson, & T. McEnery (Eds.), *Proceedings of the Corpus Linguistics 2003 conference* (pp. 572–581). Lancaster: UCREL.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., & Marsi, E. (2007). MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, *13*(2), 95–135.

Ott, N., & Ziai, R. (2010). Evaluating dependency parsing performance on German learner language. In M. Dickinson, K. Müürisep & M. Passarotti (Eds.), *Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories* (pp. 175–186). Tartu: NEALT.

Paquot, M., & Plonsky, L. (2017). Quantitative research methods and study quality in learner corpus research. *International Journal of Learner Corpus Research*, *3*(1), 61–94.

Petrov, S., & Klein, D. (2007). Improved inference for unlexicalized parsing. In B. Carpenter, A. Stent & J. D. Williams (Eds.), *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)* (pp. 404–411). Rochester: ACL.

Ragheb, M., & Dickinson, M. (2011). Avoiding the comparative fallacy in the annotation of learner corpora. In G. Granena, J. Koeth, S. Lee-Ellis, A. Lukyanchenko, G. P. Botana, & E. Rhoades (Eds.), *Selected Proceedings of the 2010 Second Language Research Forum: Reconsidering SLA Research, Dimensions, and Directions* (pp. 114–124). Somerville, MA: Cascadilla Proceedings Project.

Ragheb, M., & Dickinson, M. (2013). Inter-annotator agreement for dependency annotation of learner language. In J. Tetreault, J. Burstein, & C. Leacock (Eds.), *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 169–179). Atlanta, GA: ACL.

Rankin, T. (2015). Review of *Clausal Complements in Native and Learner English: A Corpus-Based Study with LINDSEI and VICOLSE*. *International Journal of Learner Corpus Research*, *1*(2), 279–283.

Rosen, A., Hana, J., Štindlová, B., & Feldman, A. (2014). Evaluating and automating the annotation of a learner corpus. *Language Resources and Evaluation, 48*(1), 65–92.

Santorini, B. (1990). *Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision, 2nd printing)* (Technical report). Retrieved from https://catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf (last accessed February 2018).

Tono, Y., & Díez-Bedmar, M. B. (2014). Focus on learner writing at the beginning and intermediate stages: The ICCI corpus. *International Journal of Corpus Linguistics*, *19*(2), 163–177.

Authors' addresses

Yan Huang
Language Technology Lab
Theoretical and Applied Linguistics, Faculty of Modern and Medieval Languages
University of Cambridge
Faculty of English Building, 9 West Road
Cambridge, CB3 9DB
United Kingdom

yh358@cam.ac.uk

Akira Murakami
LEAD Graduate School & Research Network
University of Tübingen
Gartenstraße 29
Tübingen, 72074
Germany

akira.murakami@philosophie.uni-tuebingen.de

Theodora Alexopoulou
Language Technology Lab
Theoretical and Applied Linguistics, Faculty of Modern and Medieval Languages
University of Cambridge
Faculty of English Building, 9 West Road
Cambridge, CB3 9DB
United Kingdom

ta259@cam.ac.uk

Anna Korhonen
Language Technology Lab
Theoretical and Applied Linguistics, Faculty of Modern and Medieval Languages
University of Cambridge
Faculty of English Building, 9 West Road
Cambridge, CB3 9DB
United Kingdom

alk23@cam.ac.uk