INTRODUCTION

Natural language processing for learner corpus research

Kristopher Kyle
University of Oregon | Yonsei University

The term natural language processing (NLP) refers to the use of computer programs to automatically analyze human language. NLP processes range from the (relatively) simple task of splitting character sequences into words and sentences to much more sophisticated (and challenging) tasks such as converting speech sounds into text and annotating texts for syntactic, semantic, and pragmatic features (among others, see Jurafsky & Manning, 2008 for a survey of common NLP processes; and Meurers & Dickinson, 2017 for specific applications to L2 research). NLP tools of varying complexity have played an important role in the development of corpus linguistics in general and learner corpus research (LCR) in particular. Although relatively simple NLP tools such as concordancers (e.g., AntConc; Anthony, 2019; Wordsmith Tools; Scott, 2020) and related programs (AntWordProfiler; Anthony, 2014; VocabProfile; Cobb, 2018; Range; Heatley & Nation, 1994) have been used extensively in the field of LCR, advances in machine learning[1] have made much more complex analyses possible. Part of speech (POS) taggers, such as TreeTagger (Schmid, 1994), CLAWS (Garside, Leech, & McEnery, 1997), and the Stanford POS Tagger (Toutanova, Klein, Manning, & Singer, 2003), for example, automatically annotate texts with POS tags, allowing for more fine-grained analyses than is possible with unannotated texts (e.g., Bestgen & Granger, 2014; Biber, Gray, & Staples, 2014; Granger & Bestgen, 2017). Syntactic parsers such as MaltParser (Nivre, Hall, & Nilsson, 2006), the Stanford Parser (Chen & Manning, 2014; Klein & Manning, 2003), and spaCy (Explosion AI, 2018) automatically annotate texts for syntactic constituency or dependency relationships. Syntactic parsers allow researchers to automatically investigate even more complex linguistic features such as dependency bigrams (e.g., Kyle & Eguchi, in press; Paquot, 2018, 2019), syntactic complexity (e.g., Alexopoulou, Michel, Murakami,

[1] Machine learning refers to a wide range of techniques used to classify (e.g., annotate) new data based on previously seen data. Examples include multinomial logistic regression, random forests, support vector machines, and neural networks (among others).

https://doi.org/10.1075/ijlcr.00019.int
International Journal of Learner Corpus Research 7:1 (2021), pp. 1–16. issn 2215-1478 | e-issn 2215-1486

& Meurers, 2017; Biber et al., 2014; Kyle & Crossley, 2018; Lu, 2010), and verb argument construction (VAC) use (e.g., Kyle, Crossley, & Verspoor, in press; Kyle & Crossley, 2017) among others. Furthermore, the release of web and desktop-based tools such as Coh-Metrix (Graesser, McNamara, Louwerse, & Cai, 2004; McNamara, Graesser, McCarthy, & Cai, 2014), the L2 Syntactic Complexity Analyzer (L2SCA; Lu, 2010; Lu & Ai, 2015) and the Tool for the Automatic Analysis of Syntactic Sophistication and Complexity (TAASSC; Kyle, 2016), among others, have allowed researchers to leverage powerful NLP tools with little to no computer programming knowledge. Due to recent advances in core NLP processes (such as syntactic annotation), the growing availability of user-friendly tools, and the release of several large learner corpora such as the *EF-Cambridge Open Language Database* (EFCAMDAT; Huang, Murakami, Alexopoulou, & Korhonen, 2018) and many others[2], researchers are increasingly using NLP tools to investigate the development of complex linguistic phenomena in large learner corpora (e.g., Díez-Bedmar & Pérez-Paredes, 2020; Green, 2019; Khushik & Huhta, 2020; Polio & Yoon, 2018).

NLP tools make it possible to automatically analyze a wide range of linguistic phenomenon at scale, which may allow for wider (and more nuanced) generalizations about learner language and linguistic development to be made. Like all analysis tools, however, NLP tools have potential weaknesses that may limit their usefulness for particular analyses, and when used inappropriately may lead to erroneous findings. For example, most automatic annotation tools are trained on well-edited L1 corpora that may be quite different in nature from the types of data used in LCR studies. Although some preliminary research has indicated that commonly investigated linguistic features can be annotated with a reasonably high degree of accuracy (e.g., Lu, 2010), much more research is needed to determine the degree to which factors such as particular language feature, proficiency, target language, particular tools, and particular sets of training data affect the accuracy of analyses. It is important, therefore, for users of NLP tools to be knowledgeable about the relative strengths and weaknesses of the tools they use in order to both maximize affordances and minimize pitfalls. Unfortunately, relatively few resources are available to help learner corpus researchers become literate about these issues. The upshot is that they often either reject the use of NLP tools altogether due to apparent weaknesses even though some analysis tools may be appropriate for their uses or adopt these tools without sufficient knowledge of their potential weaknesses, which may lead to inappropriate use. The goal of this special issue of the *International Journal*

[2] See https://uclouvain.be/en/research-institutes/ilc/cecl/learner-corpora-around-the-world.html for a list of learner corpora representing a wide range of L1s and L2s.

of Learner Corpus Research is to help increase NLP literacy by providing concrete examples of both the affordances of NLP tools and the limitations of those tools.

In this introduction, NLP processes that are common and/or show promise for LCR research (tokenization, lemmatization, part of speech tagging, constituency parsing, and dependency parsing) will be introduced with the goal of demystifying the processes and highlighting potential areas of concern. The five articles that comprise this special issue will then be introduced. Each article highlights the analysis of particular linguistic features in learner corpora by describing the feature(s) analyzed, formally evaluating how accurately one or more tools can identify the feature(s), and demonstrating the implications of the automatic analyses on downstream analyses (such as modeling proficiency).

1. Introduction to NLP

The field of NLP is quite broad, and includes a wide range of processes. In this special issue, the focus is on five NLP analyses that are relatively common in LCR research, i.e., tokenization, lemmatization, part of speech annotation, constituency parse annotation, and dependency annotation (each of which is described in some detail in the next section). Although other NLP analyses such as vector space semantics (e.g., latent semantic analysis, word2vec, etc.) also have affordances for LCR research (e.g., Crossley, Kyle, & Dascalu, 2019; Crossley & McNamara, 2012), the focus in this thematic issue will be on linguistic annotation. In this section, a broad overview of how NLP annotation works is provided, followed by a discussion of particular NLP processes and a discussion of issues related to the analysis of accuracy in learner data.

The role of training corpora in NLP

NLP analyses rely on regularities in the linguistic data they are trained on. Linguistic features that are explicitly encoded and are used with little ambiguity (e.g., the English article *the*) in a manually annotated training corpus will be automatically annotated with much greater accuracy than those that are less explicitly encoded and/or are used ambiguously (e.g., prepositional phrase attachment in English). Additionally, the degree to which the use of a particular language feature in the training data is representative of the target language use domain (e.g., the particular learner corpus texts to be processed) will affect the accuracy of automatic annotation. At least two features of the training data will affect representativeness, namely the size of the manually annotated training corpus and the degree of comparability between the register of the training corpus and the target corpus. For syntactic annotation, the largest publicly available manually annotated training corpus in English is the *OntoNotes5* corpus[3] (Weischedel, Palmer, Marcus, Hovy, Pradhan, Ramshaw, Xue, Taylor, Kaufman, & Franchini, 2013) which includes 2.6 million POS and syntactically annotated tokens across a variety of registers. Manually annotated corpus resources for other languages vary, which contributes to differences in automatic annotation accuracy across languages[4]. A general sense of the relative amount of available resources by language can be found on the Universal Dependencies Project website (https://universaldependencies.org/)[5]. Training corpora of over 1 million words are available for languages such as Czech, Japanese, and Russian, while much smaller training corpora data are available for other languages (e.g., around 200,000 words for the largest manually annotated Dutch corpus represented in the link above, and 63,000 words for Greek).

Tokenization

Tokenization involves dividing text into word units. In English and other languages that separate word units using spaces, tokenization is a reasonably straightforward and highly accurate task that involves two main steps. First, most (if not all) punctuation will need to be separated from word units. Because some punctuation marks may be used ambiguously (e.g., periods in English), tokenizers may need to use statistical/machine learning models to accurately separate non-word punctuation from words. Second, words need to be split using white space and any non-word items may need to be removed, depending on the goals of the researcher. For most texts, tokenization can be completed with a high degree of accuracy, though typos (e.g., the omission of spaces as in *pizza is delicious.I love it*) can cause errors.

In languages where word units are not necessarily separated by white space and have ambiguous inflectional morphemes (e.g., Korean, Turkish), word tokenization may be less straightforward and may be a source of error in learner corpus analyses (see Shin & Jung, this volume).

[3] *OntoNotes5* is distributed by the Linguistic Data Consortium (LDC) and is freely available to registered users.

[4] Language-specific features and the classification algorithm(s) used (among others) will also affect automatic annotation accuracy.

[5] The resources included on the Universal Dependencies Project webpage is not exhaustive and may not be representative of available resources. *OntoNotes5*, for example, is distributed by the LDC, and many avenues are available for the distribution of manually annotated data.

Lemmatization

Lemmatization involves grouping inflected forms of words (e.g., *ran*) by their uninflected form (e.g., *run*) so they can be analyzed as a single word form. While lemmatization is not required or necessarily preferable in all situations (or for all languages!), lemmatization is commonly employed in many learner corpus studies (particularly with English as an L2). A common method for text lemmatization is to use a surface-form based lemma list, such as the one publicly available on Laurence Anthony's AntConc page[6] (this type of lemmatization is also referred to as flemmatization; Pinchbeck, 2017). While surface-form lists are commonly used, the existence of homographs (e.g., the verb *run* [*run, runs, running, ran*] and the noun *run* [*run, run*] may result in imprecision in the calculation of lexical diversity and/or frequency scores (see Jarvis & Hashimoto, this volume). An alternative method is to use a POS tagger to annotate each word for POS (see description of part of speech taggers below). If highly accurate part of speech annotation can be obtained, then homographs with different parts of speech (as in our example above) can be disambiguated (though different senses of a word with the same part of speech will still be conflated). An alternative to lemmatization is familization (Bauer & Nation, 1993) which substitutes an inflected or derivational form of a word for its root. Familization is based on surface-form lists (much like flemmatization) such as the ones available on *Victoria University of Wellington*'s webpage[7]. However, because both inflected and derived forms (including zero-derivation) of a root are included, homography is likely a minor issue.

For unedited texts (including those produced by learners), typos and misspelled words will lower the accuracy of lemmatization which in turn may affect downstream linguistic analyses (e.g., calculation of lexical diversity or frequency scores). See Jarvis and Hashimoto (this issue) for a systematic analysis of the effects of different types of lemmatization on the calculation of lexical diversity scores.

Part of speech annotation

Part of speech (POS) annotation has many affordances for learner corpus researchers. As noted in the previous section, POS annotation can be used to disambiguate homographs, but POS annotation can also be used to enable lexicogrammatical analyses of language use (e.g., Bestgen & Granger, 2014; Biber et al., 2014; Granger & Bestgen, 2017). Additionally, POS annotations provide the

[6] https://www.laurenceanthony.net/software/antconc/.

[7] https://www.wgtn.ac.nz/lals/resources/paul-nations-resources/vocabulary-lists.

foundation for complex syntactic annotation such as constituency and dependency parse annotation (see below).

There are a number of specific approaches to automatic part of speech tagging that vary with regard to the feature sets used and statistical/machine learning algorithms used. However, most POS taggers take the same basic approach. First, all words with unambiguous part of speech tags in the training data are assigned their respective tag. Then, a number of contextual features such as the POS tag of the preceding word or words, the endings of the target word and preceding word or words, the target word itself, etc. are used as predictors in a statistical or machine learning algorithm to predict the part of speech of words that have ambiguous tags in the training data or are not attested in the training data. POS taggers can achieve high levels of annotation accuracy for both well-edited, L1 texts that match the language use domain(s) of the training corpus and for many types of L2 texts. State of the art accuracy for L1 English texts is around 97% (averaged across all tags; e.g., Schmid, 1995; Toutanova et al., 2003). High levels of annotation accuracy have also been reported for some L2 English texts. Berzak, Kenney, Spadine, Wang, Lam, Mori, Garza, and Katz (2016), for example, report annotation accuracy of 94.28% (averaged across all tags)[8] for a mixed-proficiency sample of L2 English sentences ($n=500$) from the *Cambridge Learner Corpus* (Yannakoudakis, Briscoe, & Medlock, 2011). Jarvis and Hashimoto (this volume) also report high annotation accuracy in mixed proficiency (CEFR A1–B2) written L2 narrative retellings for most tags (e.g., 96.8% for nouns and 97.8% for verbs).

As with tokenization and lemmatization, typos and spelling errors, language-specific issues may cause POS annotation errors. Additionally, because POS tagging relies on the regularities in word sequencing, word order and collocational errors in a learner text may affect the accuracy of part of speech annotations for words whose surface form could be assigned multiple tags. Berzak et al. (2016), for example, found that POS annotation accuracy for tokens used ungrammatically was 88.61%, compared with 95.37% for tokens used grammatically. Accordingly, proficiency is highly likely to affect POS annotation for some tags. Language use domain (e.g., mode or register) may also cause annotation errors in learner texts as linguistic regularities may differ based on domain. It can be assumed, for example, that a POS tagger trained on well-edited L1 written texts will achieve lower annotation accuracy for texts produced in spoken modes, by lower proficiency language users, and/or in less formal genres. However, there has been relatively little work done in this area, and more research is needed. Additionally, for some

[8] Berzak et al. (2016) use a relatively small training corpus (204,586 tokens), which likely resulted in lower accuracy than would have been obtained if a larger training corpus had been used.

languages and some linguistic features, automatic annotation may not be feasible (see, e.g., Shin & Jung, this volume for a discussion of the annotation of passive constructions in Korean).

Constituency parse annotation

Constituency parsers generate syntactic constituency trees for sentences in a text. One popular use for automatic constituency parse annotation has been the calculation of syntactic complexity measures such as mean length of T-unit (MLTU). Lu (2010) introduced the L2 syntactic complexity analysis tool (L2SCA), which calculates 14 measures of syntactic complexity in English texts using constituency parses generated by the Stanford Parser (Klein & Manning, 2003). Syntactic constituency annotation is also used to calculate a number of other indices of syntactic use (see, e.g., McNamara et al., 2014; Weiss & Meurers, this volume).

Syntactic constituency parsers use phrase structure rules generated from training corpora to create sentence level syntactic constituency trees. Texts are first tagged for part of speech, then the tags are used, in conjunction with the phrase structure rules, to generate a number of competing sentence-level parse trees. Finally, statistical/machine learning algorithms are used to select the most probable parse tree for the sentence (see, e.g., Jurafsky & Manning, 2008). State of the art system accuracy for constituency parsers is above 90% (averaged across all nodes) for English (e.g., Kitaev & Klein, 2018). To my knowledge, there are no published accounts of annotation accuracy for constituency annotation for L2 texts per se. However, there is evidence that downstream annotation (such as the identification of T-units and clauses) can be achieved in L2 texts with a reasonable degree of accuracy. Lu (2010), for example, found that L2SCA annotated written L2 university-level English texts with an accuracy of over 90% for most features (e.g., 96.1% for clauses, 97.6% for T-units, but 83% for complex nominals). He also found large correlations between syntactic complexity scores based on manually and automatically annotations. Correlations were above $r=.900$ for some features (e.g., $r=.932$ for mean length of clause, $r=.987$ for mean length of T-unit) and between $r=.800$ and .899 for others (e.g., $r=.840$ for dependent clauses per clause, $r=.867$ for complex nominals per clause). Polio and Yoon (2018) found similar (if slightly lower) correlations between manually and automatically annotated syntactic complexity scores in L2 English argumentative and narrative texts written by university-level ESL students. In this volume, Weiss and Meurers report annotation accuracy for a number of syntactic complexity features in L2 German that are based on the Stanford Parser (Chen & Manning, 2014; Klein & Manning, 2003).

Constituency annotation accuracy will depend on the accuracy of dependent processes such as tokenization and part of speech tagging. Accordingly, previously discussed factors such as language proficiency, mode, and register will affect results, as will the degree to which sentences structures in an L2 text are represented in the training corpus (e.g., are well-formed).

Dependency relation annotation

Dependency parsers annotate texts for syntactic dependency relationships. Each word in a sentence is assigned a single dependency head (e.g., the head of a subject is a main verb) but may have multiple dependents (e.g., a main verb may have a subject, direct object, auxiliaries, and or adverbials as dependents, among others). Dependency parsers are becoming increasingly popular (e.g., Stanford Dependency Parser; Chen & Manning, 2014; spaCy; Explosion AI, 2018; MaltParser; Nivre et al., 2006) and have been recently used in a number of LCR studies (e.g., Kyle et al., in press; Kyle & Crossley, 2017; Paquot, 2018, 2019), likely owing to the ease at which relevant syntactic relationships can be extracted. Kyle and colleagues, for example, used the Stanford Neural Network Dependency Parser to extract verb argument constructions from a reference corpus and from cross-sectional (Kyle & Crossley, 2017) and longitudinal learner corpora (Kyle et al., in press). Paquot (2018, 2018) also used the Stanford Dependency Parser to extract collocations that are constrained to particular dependency relationships (e.g., verb-direct object). Kyle and Eguchi (in press) took a similar approach using spaCy (Explosion AI, 2018). Also see Picoral, Staples, and Reppen (this volume) and Rubin (this volume) for analyses that use dependency parsers for English and Dutch respectively.

Although early dependency annotation was derived from constituency parse annotation, most current dependency annotation is derived directly based on corpora annotated for dependency relationships. Dependency parsers use part of speech tags, word forms, lemmas, direction of dependencies and distance between words (among others, see e.g., Jurafsky & Martin, 2019) as feature sets to predict dependency relations. A variety of specific approaches and statistical/machine learning are used by various dependency parsers, but the distribution of feature set items in the training corpus are used to predict the dependency head of each word.

State of the art accuracy (averaged across dependency tags) for dependency parsers is above 90% (e.g., Choi, Tetreault, & Stent, 2015) for well-edited L1 English. Dependency parsing models are available for a number of languages, and accuracy varies by language (due to the structure of the language itself and the amount of annotated training data available). Berzak et al. (2016) reported an average annotation accuracy of 88.07% for labeled dependency tags in 500 written L2 English sentences from the *Cambridge Learner Corpus* (Yannakoudakis et al., 2011). Geertzen, Alexopoulou, and Korhonen (2013) found similar results for L2 texts from the

EFCAMDAT corpus. With regard to specific dependency relationships, Kyle and Eguchi (in press) reported that spaCy achieved annotation accuracies above 95% for noun-adjective (96.9%), verb-adverb (98.6%), verb-direct object (96.0%), and verb-subject (95.4%) in a subset of sentences from argumentative L2 essays representing various proficiencies. Lower accuracies have been reported when using other parsers (e.g., Paquot, Naets, & Gries, in press) and for the annotation of more complex linguistic features. Kyle et al. (in press), for example, reported 80% annotation accuracy for verb argument constructions (which were defined as a main verb and all of its direct, non-auxiliary dependents) in lower proficiency L2 English essays using spaCy. In the current volume, Picoral et al. and Rubin use dependency parsers to annotate L2 English and L2 Dutch texts respectively for a range of linguistic features with varying levels of success.

As with the other NLP processes described above, a variety of factors will affect the accuracy of dependency relation annotation. These include tokenization and POS annotation accuracy, the alignment between the language use domain(s) of the training corpora and the learner corpora to be annotated, the features and availability of training data in a particular language and the linguistic errors extant in the learner corpora. With regard to the effects of errors, Berzak et al. (2016) found that dependency annotation accuracy for tokens used ungrammatically was 82.66%, compared with 89.11% for tokens used grammatically in a corpus of L2 English texts. This indicates that language proficiency will likely affect annotation accuracy.

2. Some specific challenges for calculating accuracy in LCR research

NLP annotation is the cumulative result of multiple processes, which is commonly referred to as a pipeline. A dependency annotation pipeline, for example, could involve tokenization, sentence segmentation, POS annotation, lemmatization, and dependency annotation. Errors in any step of the pipeline are likely to affect the accuracy of any downstream processes. In most cases, however, the accuracy figures reported for an NLP process presume that all previous processes were completed perfectly (this is the norm in computer science publications, but not in the few related LCR studies that have been published). Accuracy is usually calculated by training an annotation algorithm on a large part (90–95%) of a manually annotated training corpus, and then evaluating its performance with the rest of the corpus (5%–10%). Practically speaking, the tokenization of the annotation algorithm has to be aligned with the manually tagged corpus in order to compare the output of other processes such as POS tagging or dependency parsing, which means that the potential for tokenization accuracy to affect downstream processes is ignored.

Second, in order to compare the performance of (and improve) different annotation systems (e.g., POS annotators that use different approaches), only one piece of the pipeline is evaluated at a time. The upshot is that most reported accuracy figures for complex NLP annotation processes such as constituency or dependency syntactic annotation is somewhat optimistic for in-domain and well edited texts and may be very misleading for out of domain and/or unedited learner texts. Further, summaries of system performance that are reported on resource pages for NLP tools often report an average accuracy figure, which may lead to misperceptions regarding the accuracy of the annotation of the particular features a researcher is interested in investigating. Some linguistic features may be annotated with near perfect accuracy, while others may not be annotated accurately at all.

At least two other potential complications exist for LCR researchers attempting to evaluate the performance of a particular NLP tool. The first is that NLP annotators are usually designed for downstream processes other than those interested in language acquisition (Meurers & Dickinson, 2017). The result is that part of speech and syntactic annotation systems will not always identify the language features that researchers are interested in without some degree of further processing. To identify T-units, for example, Lu's (2010) L2SCA first uses the Stanford Parser (Klein & Manning, 2003) to generate a constituency parse of the text, then uses Tregex (Levy & Andrew, 2006) to search for a set of parse-tree patterns that align with Lu's operational definition of a T-unit. To calculate mean length of T-unit, the number of words must also be calculated (which involves determining with tokenized units should count as words). In order to determine how reliable automatically generated MLTU scores are in a particular study, multiple pieces of evidence are particularly helpful. Lu (2010), for example, trained human annotators to identify eight linguistic features (including T-units) in a sample of 30 essays from the much larger learner corpus used in his study. He then calculated precision, recall, and F1[9] scores for the automatic annotation. Finally, correlations between computed syntactic complexity scores (e.g., MLTU) from the manual and automatic annotations were computed. The inclusion of specific accuracy figures (e.g., for each annotation type instead of an average) and multiple types of accuracy information can be particularly helpful when deciding whether a tool may be appropriate for a particular application or not.

A second complication is a lack of previously existing manually annotated training and evaluation corpora (see Berzak et al., 2016; Meurers & Dickinson, 2017). Currently (to my knowledge), there is only a single publicly available L2 corpus manually annotated for POS tags and syntactic information of L2 English writing (Berzak et al., 2016) and a second of transcribed L2 English speech that will be

[9] F1 scores are accuracy scores that consider both precision and recall.

released in 2021 (Kyle & Eguchi, in progress). This means that in most cases, LCR researchers must have the resources (including the availability of knowledgeable human annotators and ways to compensate them) to annotate (usually smaller subsets of) their corpora before they have an idea of how accurate a particular tool is in their context.

3. The present issue

The goal of this special issue is to introduce some of the potential affordances provided by NLP tools while also indicating potential weaknesses of these tools. A range of L2s, proficiency levels, and registers are represented in the contributions to this issue. In each study the default version of the automatic annotator(s) is used, which is likely representative of the version that would be used by most LCR researchers.

In the first contribution to this issue, Picoral, Staples, and Reppen investigate the degree to which NLP tools can accurately annotate four phrasal and clausal language features in L1 and L2 English student academic writing. Specifically, the automatic annotation of attributive adjectives, noun-noun sequences, finite relative clauses, and complement clauses was examined using the Biber Tagger (Biber, 1988) and two commonly used and open-source dependency parsers: MaltParser (Nivre et al., 2006) and the Stanford Neural Network Dependency Parser (Chen & Manning, 2014). The manually annotated evaluation corpus comprises academic student writing from three L2 English groups (L1 speakers of Arabic, Chinese, and Korean), and L1 speakers of English. Picoral et al. provide transparent descriptions of their manual annotation guidelines and report a detailed analysis of the accuracy of each tool both across the entire dataset and for each L1 group represented, including a discussion of the causes of annotation error.

In the second contribution, Shin and Jung investigate the (semi)automatic annotation of two passive constructions in texts produced by L2 writers of Korean. The article highlights potential issues that may occur for researchers of languages for which tokenization is a more difficult task and/or that have fewer open-source NLP resources. Additionally, issues surrounding the annotation of linguistic features not explicitly encoded by extant annotation guidelines are discussed and an example of how to deal with related issues is given. Shin and Jung provide clear examples of how to use the NLP tools and techniques to one's advantage, even when fully automated approaches are not possible.

The third contribution by Weiss and Meurers investigates the accuracy of various automated linguistic annotation tools when used in the context of short L2 German texts. After outlining the context of the larger task (predicting proficiency level based on responses to reading comprehension questions), a detailed analysis of the accuracy of the underlying NLP annotators is conducted. Results are reported for various levels of granularity, ranging from the accuracy of the annotation for specific morphemes to the percentage of texts that were annotated perfectly, which provides a rich impression of the strengths and weaknesses of the annotators used in this context. Finally, the effects of the annotation errors on the calculation of indices of linguistic complexity are reported, which provide a contextualized understanding of the accuracy figures.

In the fourth contribution, Rubin investigates the degree to which language proficiency affects the extraction of dependency relationships in L2 Dutch. Rubin focuses on three dependency relationships (verb-direct object, adjective – noun, adverb – verb), three proficiency levels (CEFR B1, B2, C1), and two dependency parsers: Alpino (van Noord, 2006) and Frog (van den Bosch, Busser, Canisius, & Daelemans, 2007). Accuracy figures are reported for each dependency relationship, proficiency level, and parser. Importantly, the effects of annotator error on the calculation of mutual information (MI) scores are explored with regard to mean scores and correlations. The varied results provide an excellent starting point for researchers of L2 Dutch who are interested in exploring phraseological development using dependency relationships.

In the fifth and final contribution Jarvis and Hashimoto investigate the relationship between lemmatization choices on the calculation of lexical diversity indices in written L2 English texts. Specifically, the effects of using different lemmatization schemes (including a comparison of automated and manually corrected POS-specific lemmas) on the strength of the relationships between three measures of lexical diversity and human judgements of lexical diversity were analyzed. Jarvis and Hashimoto examine these relationships from multiple perspectives and also provide a detailed account of the characteristics of outlier texts. The study provides an evidence-based starting point for L2 English researchers who are deciding precisely how lexical diversity should be operationalized in their own work.

Together, the contributions to this issue help shed some light on the potential affordances of some frequently used NLP tools to investigate linguistic features that are common in NLP research. Overall, the results are optimistic, and demonstrate that automated annotation can be highly accurate for many linguistic features in many L2 contexts. The results also highlight the fact that the use of automated annotation needs to be informed by a variety of factors such as the target language and the specific features to be annotated. It is important for LCR researchers to determine the degree to which their intended analyses can be accurately conducted with NLP tools. As more studies and resources are published, this will become easier as researchers can cite previous studies (such as those in this issue) and/or use manually annotated L2 datasets (e.g., Berzak et al., 2016). In the meantime, however, most researchers will need to conduct accuracy analyses on smaller subsets of their data following previous studies (e.g., Kyle & Eguchi, in press; Lu, 2010; Paquot et al., in press; Polio & Yoon, 2018). It is hoped that future researchers will continue to explore the affordances of various NLP tools and their weaknesses in a variety of contexts. It is also hoped that LCR researchers will work to develop (publicly available) manually annotated L2 corpora that represent a variety of contexts, which will assist in the creation of more accurate annotation models (Berzak et al., 2016; Meurers & Dickinson, 2017), highlight particular areas for improvement, and make reporting accuracy less resource dependent.

References

Alexopoulou, T., Michel, M., Murakami, A., & Meurers, D. (2017). Task Effects on Linguistic Complexity and Accuracy: A Large-Scale Learner Corpus Analysis Employing Natural Language Processing Techniques. *Language Learning*, 67(S1), 180–208. https://doi.org/10.1111/lang.12232

Anthony, L. (2014). *AntWordProfiler (Version 1.4. 1)[Computer Software]*. Tokyo, Japan: Waseda University.

Anthony, L. (2019). *AntConc (3.5.8) [Computer software]*. Tokyo, Japan: Waseda University.

Bauer, L., & Nation, I. S. P. (1993). Word families. *International Journal of Lexicography*, 6(4), 253–279. https://doi.org/10.1093/ijl/6.4.253

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 *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (pp. 737–746). Stroudsburg: Association for Computational Linguistics.

Bestgen, Y., & Granger, S. (2014). Quantifying the development of phraseological competence in L2 English writing: An automated approach. *Journal of Second Language Writing*, 26, 28–41. https://doi.org/10.1016/j.jslw.2014.09.004

Biber, D. (1988). *Variation across speech and writing*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511621024

Biber, D., Gray, B., & Staples, S. (2014). Predicting Patterns of Grammatical Complexity Across Language Exam Task Types and Proficiency Levels. *Applied Linguistics*, 37(5), 639–668. https://doi.org/10.1093/applin/amu059

Chen, D., & Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 740–750). Stroudsburg: Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1082

Choi, J. D., Tetreault, J., & Stent, A. (2015). It depends: Dependency parser comparison using a web-based evaluation tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 387–396). Stroudsburg: Association for Computational Linguistics.

Cobb, T. (2018). Web VocabProfile (WebVP). [Computer Software].

Crossley, S.A., Kyle, K., & Dascalu, M. (2019). The Tool for the Automatic Analysis of Cohesion 2.0: Integrating semantic similarity and text overlap. *Behavior Research Methods*, 51(1), 14–27. https://doi.org/10.3758/s13428-018-1142-4

Crossley, S.A., & McNamara, D.S. (2012). Predicting second language writing proficiency: The roles of cohesion and linguistic sophistication. *Journal of Research in Reading*, 35(2), 115–135. https://doi.org/10.1111/j.1467-9817.2010.01449.x

Díez-Bedmar, M.B., & Pérez-Paredes, P. (2020). Noun phrase complexity in young Spanish EFL learners' writing: Complementing syntactic complexity indices with corpus-driven analyses. *International Journal of Corpus Linguistics*, 25(1), 4–35. https://doi.org/10.1075/ijcl.17058.die

Explosion AI. (2018). *spaCy language models*. Retrieved from https://spacy.io/models/en#en_core_web_sm

Garside, R., Leech, G.N., & McEnery, T. (1997). *Corpus annotation: Linguistic information from computer text corpora*. Harlow: Longman. https://doi.org/10.4324/9781315841366

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. Marcos Miguel, A.M. Tseng, A. Tuninetti, & D. Walter (Eds.), *Selected Proceedings of the 2012 Second Language Research Forum* (pp. 240–254). Somerville, MA: Cascadilla Proceedings Project.

Graesser, A.C., McNamara, D.S., Louwerse, M.M., & Cai, Z. (2004). Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*, 36(2), 193–202. https://doi.org/10.3758/BF03195564

Granger, S., & Bestgen, Y. (2017). Using collgrams to assess L2 phraseological development: A replication study. In P. Haan, R. de Vries, & S. van Vuuren (Eds.), *Language, Learners and Levels: Progression and Variation* (pp. 385–408). Louvain-la-Neuve: Presses universitaires de Louvain.

Green, C. (2019). Enriching the academic wordlist and Secondary Vocabulary Lists with lexicogrammar: Toward a pattern grammar of academic vocabulary. *System*, 87, 102158. https://doi.org/10.1016/j.system.2019.102158

Heatley, A., & Nation, I.S.P. (1994). *Range. [Computer Software]*. Victoria University of Wellington, NZ. Retrieved from http://Www.Vuw.Ac.Nz/Lals/

Huang, Y., Murakami, A., Alexopoulou, T., & Korhonen, A. (2018). Dependency parsing of learner English. *International Journal of Corpus Linguistics*, 23(1), 28–54. https://doi.org/10.1075/ijcl.16080.hua

Jurafsky, D., & Manning, C.D. (2008). *Speech and language processing: An introduction to natural language processing, speech recognition, and computational linguistics* (2nd ed.). Upper Saddle River: Prentice-Hall.

Jurafsky, D., & Martin, J.H. (2019). *Speech and Language Processing* (Unpublished Manuscript). October 2019. Retrieved from https://web.stanford.edu/~jurafsky/slp3/

Khushik, G.A., & Huhta, A. (2020). Investigating Syntactic Complexity in EFL Learners' Writing across Common European Framework of Reference Levels A1, A2, and B1. *Applied Linguistics*, 41(4), 506–532. https://doi.org/10.1093/applin/amy064

Kitaev, N., & Klein, D. (2018). Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2676–2686). Stroudsburg: Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1249

Klein, D., & Manning, C. D. (2003). *Accurate unlexicalized parsing*. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (pp. 423–430). Stroudsburg: Association for Computational Linguistics. https://doi.org/10.3115/1075096.1075150

Kyle, K. (2016). Measuring Syntactic Development in L2 Writing: Fine Grained Indices of Syntactic Complexity and Usage-Based Indices of Syntactic Sophistication (Unpublished doctorial dissertation). Georgia State University, Atlanta. http://scholarworks.gsu.edu/alesl_diss/35/

Kyle, K., & Crossley, S. A. (2017). Assessing syntactic sophistication in L2 writing: A usage-based approach. *Language Testing*, 34(4), 513–535. https://doi.org/10.1177/0265532217712554

Kyle, K., & Crossley, S. A. (2018). Measuring Syntactic Complexity in L2 Writing Using Fine-Grained Clausal and Phrasal Indices. *The Modern Language Journal*, 102(2), 333–349. https://doi.org/10.1111/modl.12468

Kyle, K., Crossley, S. A., & Verspoor, M. (in press). Measuring longitudinal writing development using indices of syntactic complexity and VAC sophistication. *Studies in Second Language Acquisition*.

Kyle, K., & Eguchi, M. (in press). Automatically assessing lexical sophistication using word, bigram, and dependency indices. In S. Granger (Ed.), *Perspectives on the Second Language Phrasicon: The View from Learner Corpora*. Bristol: Multilingual Matters.

Kyle, K., & Eguchi, M. (in progress). *A gold standard part of speech tagged and dependency parsed corpus of L2 speech*.

Levy, R., & Andrew, G. (2006). *Tregex and Tsurgeon: Tools for querying and manipulating tree data structures*. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)* (pp. 2231–2234). European Language Resources Association (ELRA).

Lu, X. (2010). Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15(4), 474–496. https://doi.org/10.1075/ijcl.15.4.02lu

Lu, X., & Ai, H. (2015). Syntactic complexity in college-level English writing: Differences among writers with diverse L1 backgrounds. *Journal of Second Language Writing*, 29, 16–27. https://doi.org/10.1016/j.jslw.2015.06.003

McNamara, D. S., Graesser, A. C., McCarthy, P. M., & Cai, Z. (2014). *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511894664

Meurers, D., & Dickinson, M. (2017). Evidence and interpretation in language learning research: Opportunities for collaboration with computational linguistics. *Language Learning*, 67(S1), 66–95. https://doi.org/10.1111/lang.12233

Nivre, J., Hall, J., & Nilsson, J. (2006). MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings of the fifth international conference on language resources and evaluation (LREC'06)* (pp. 2216–2219). European Language Resources Association (ELRA).

Paquot, M. (2018). Phraseological Competence: A Missing Component in University Entrance Language Tests? Insights From a Study of EFL Learners' Use of Statistical Collocations. *Language Assessment Quarterly*, 15(1), 29–43. https://doi.org/10.1080/15434303.2017.1405421

Paquot, M. (2019). The phraseological dimension in interlanguage complexity research. *Second Language Research*, 35(1), 121–145. https://doi.org/10.1177/0267658317694221

Paquot, M., Naets, H., & Gries, S. T. (in press). Using syntactic co-occurrences to trace phraseological complexity development in learner writing: Verb + object structures in LONGDALE. In B. LeBruyn & M. Paquot (Eds.), *Learner Corpus Research Meets Second Language Acquisition*. Cambridge: Cambridge University Press.

Pinchbeck, G. G. (2017). Vocabulary Use in Academic-Track High-School English Literature Diploma Exam Essay Writing and its Relationship to Academic Achievement (Unpublished doctoral dissertation). University of Calgary, Calgary.

Polio, C., & Yoon, H. (2018). The reliability and validity of automated tools for examining variation in syntactic complexity across genres. *International Journal of Applied Linguistics*, 28(1), 165–188. https://doi.org/10.1111/ijal.12200

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing* (pp. 44–49). Manchester, UK.

Schmid, H. (1995). Treetagger: A language independent part-of-speech tagger [Computer software] Institut Für Maschinelle Sprachverarbeitung, Universität Stuttgart, Stuttgart.

Scott, M. (2020). WordSmith Tools (8.0) [Computer software]. Liverpool: Lexical Analysis Software.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). *Feature-rich part-of-speech tagging with a cyclic dependency network*. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology – Volume 1* (pp. 173–180). Stroudsburg: Association for Computational Linguistics.

van den Bosch, A., Busser, B., Canisius, S., & Daelemans, W. (2007). An efficient memory-based morphosyntactic tagger and parser for Dutch. In P. Dirix, I. Schuurman, V. Vandeghinste, & F. Van Eynde (Eds.), *Proceedings of the 17th meeting of Computational Linguistics in the Netherlands* (pp. 191–206).

van Noord, G. (2006). At last parsing is now operational. In *TALN 2006* (pp. 20–42).

Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., & Franchini, M. (2013). *Ontonotes release 5.0*. Philadelphia: Linguistic Data Consortium. Retrieved from https://catalog.ldc.upenn.edu/LDC2013T19

Yannakoudakis, H., Briscoe, T., & Medlock, B. (2011). A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 180–189). Stroudsburg: Association for Computational Linguistics.

Address for correspondence

Kristopher Kyle
University of Oregon
Department of Linguistics
161 Straub Hall
Eugene Oregon 97403-1290
USA

kristopherkyle1@gmail.com