Measuring the Power of Learning.®

Research Report
ETS RR–16-02

A Protocol for Annotating Parser Differences

James V. Bruno

Aoife Cahill

Binod Gyawali

June 2016

ETS Research Report Series

Since its 1947 founding, ETS has conducted and disseminated scientific research to support its products and services, and to advance the measurement and education fields. In keeping with these goals, ETS is committed to making its research freely available to the professional community and to the general public. Published accounts of ETS research, including papers in the ETS Research Report series, undergo a formal peer-review process by ETS staff to ensure that they meet established scientific and professional standards. All such ETS-conducted peer reviews are in addition to any reviews that outside organizations may provide as part of their own publication processes. Peer review notwithstanding, the positions expressed in the ETS Research Report series and other published accounts of ETS research are those of the authors and not necessarily those of the Officers and Trustees of Educational Testing Service.

The Daniel Eignor Editorship is named in honor of Dr. Daniel R. Eignor, who from 2001 until 2011 served the Research and Development division as Editor for the ETS Research Report series. The Eignor Editorship has been created to recognize the pivotal leadership role that Dr. Eignor played in the research publication process at ETS.

ETS Research Report Series ISSN 2330-8516

RESEARCH REPORT

A Protocol for Annotating Parser Differences

James V. Bruno, Aoife Cahill, & Binod Gyawali

Educational Testing Service, Princeton, NJ

We present an annotation scheme for classifying differences in the outputs of syntactic constituency parsers when a gold standard is unavailable or undesired, as in the case of texts written by nonnative speakers of English. We discuss its automated implementation and the results of a case study that uses the scheme to choose a parser best suited to a downstream task that processes nonnative written text.

Keywords NLP; parsing; parse annotation; syntax

doi:10.1002/ets2.12086

Recently, there has been a notable increase in the availability of syntactic constituency parsers. When choosing a parser to use for a particular application, one of the most obvious metrics to consider is accuracy (generally in terms of F-score) on some manually annotated data set (such as the *Wall Street Journal* [*WSJ*], Section 23). However, many parsers achieve roughly the same level of parsing accuracy on such datasets, making it unclear which one may be best suited for a new application. The speed of a parser can also be a significant factor in choosing a parser (as can licensing constraints). Having said that, when everything appears equal, it is still possible that there are differences in performance that could affect downstream processing applications that these metrics mask.

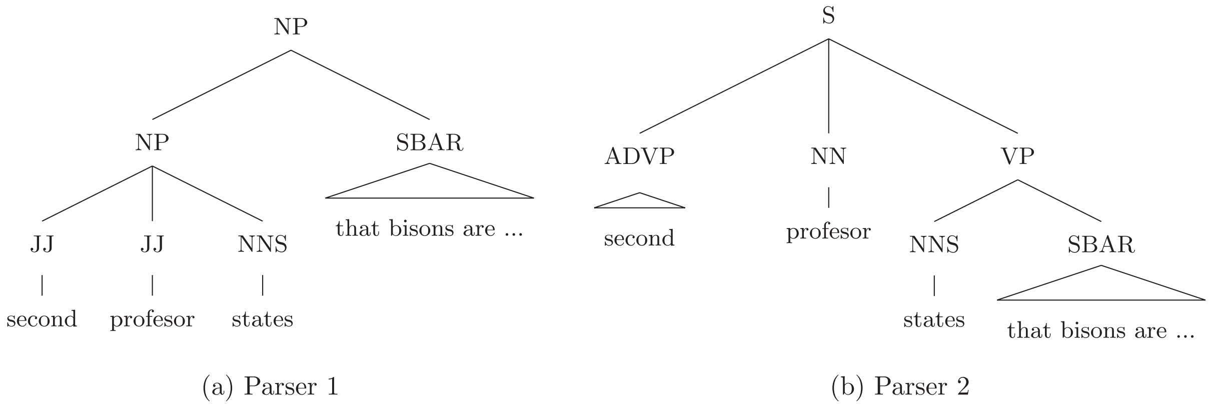
Kummerfeld, Hall, Curran, and Klein (2012) presented a method for systematically comparing the errors that different parsers make according to a gold standard. This is an extremely useful analysis for cases in which a gold standard is available; however, in the case of texts written by nonnative speakers, it is often unclear how a gold standard could even be determined. Because sentences are often ungrammatical, determining a gold standard amounts to the task of assigning a "correct" grammatical analysis to a sentence that is ill formed. The paradox can sometimes be resolved by assuming an intended target grammatical sentence; however, such assumptions are subject to interpretation, and in some cases, the sentences are too garbled to make a sufficiently informed guess as to the intention of the writer.

Despite these challenges, it is still sometimes possible to choose among competing parses which one is best, even in the absence of a gold standard. For example, in the comparison in Figure 1, it is clear that Parser 2 is preferable at least as far as the constituency is concerned, as it has identified the more likely label (VP) for the maximal projection of *states*. In addition, the SBAR is the complement of *states* within the VP, instead of an adjunct attached to an NP. Considerations of parse preferences may have to do with how well the parse reflects the argument structure of the predicates and how well modification and scopal relationships are represented, among other things.

We therefore present a parser *difference* annotation scheme, in a similar spirit to Kummerfeld et al. (2012), but focused on syntactic differences that do not assume the presence of a gold standard. The goal of this annotation scheme is not to classify *errors* but rather to highlight *differences* for manual inspection before choosing the most suitable parser for an application. Our motivation is that if a gold standard is not available (or appropriate), we still need to understand the differences in parser behavior to determine which one best analyzes the constructions that are most important for an application.

This scheme was used in Cahill, Gyawali, and Bruno (2014) to compare two parsers. In that paper, the focus was on the methodology for deriving a new parser from an existing one. Here we focus on the details of the annotation scheme itself (previously only partially described), its automated implementation, and a case study applying this technique to texts written by nonnative speakers. Specifically, we evaluate the performance of three parsers on the constituency they build around misspelled items.

Corresponding author: J. V. Bruno, E-mail: jbruno@ets.org



(a) Parser 1

(b) Parser 2

Figure 1 Parser 2 outperforms Parser 1.

Parse Differences Annotation Scheme

Development of the Annotation Scheme

The annotation scheme was developed through a data-driven approach. An initial sample of five essays written by nonnative speakers of English was tokenized into sentences and submitted to two versions of the Charniak and Johnson (2005) (BLLIP) parser,[1] as described in Cahill et al. (2014). The sample contained 69 sentences, and the outputs of the parsers differed on 32 of them. The different parses for these 32 sentences were then manually scrutinized, and a set of 16 linguistically defined differences was developed on the basis of what was found in the initial sample.

We then developed rules to automatically detect these differences by traversing pairs of parse trees[2] and identifying patterns that indicate each of the difference definitions we developed.[3] Finally, we parsed a sample of 10,000 sentences with our two versions of the parser, submitted the parser outputs to our automated rules, and extracted those parses that differed in ways in which our automated rules were unable to categorize. We then took a sample of 50 of these differing parses and subjected them to further manual scrutiny. This effort resulted in the development of one more definition, which brought the total number of definitions to 17. That we were only able to extract one more linguistically meaningful definition from this second round of analysis suggests that our coverage is fairly complete, although we cannot claim with certainty that it is exhaustive.[4] More details about our coverage appear in the section Coverage Comparison, in which we compare our coverage with the Kummerfeld et al. (2012) system.

Definitions of Differences

Our annotation scheme is presented here in its entirety. Later, we demonstrate the use of this scheme toward the specific goal of selecting the best parser for downstream applications in which misspellings may play a role; however, we make no claim as to the impacts of these differences for any particular task. We present a taxonomy of the ways in which parsers can differ. The impact of these differences is an empirical question that remains to be investigated in future work. The definitions that comprise our annotation scheme are as follows:

Attachment site. A constituent that exists in both parses attached within a different XP in each parse, where XPs are identified by their heads.[5] We further categorize this difference according to the label of the constituent (e.g., PP attachment vs. NP attachment).[6]

Constituent label. A pair of corresponding spans that have a different maximal label in each parse. Note that this difference does not take the internal structure into account.

POS of misspelled terminal. A pair of corresponding terminals with different POS tags, where the terminals do not exist in our dictionary of known English[7] words.

POS. A pair of corresponding terminals with different POS tags, where the terminals do exist in our dictionary of known English words.

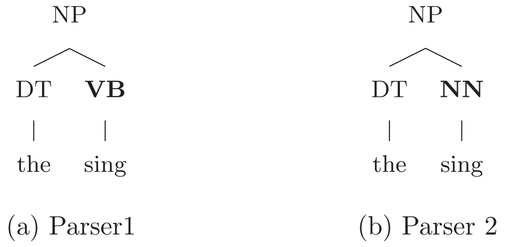


Figure 2 Headedness: "*… the sing …*"

Constituency of misspelled item. A difference in constituency[8] involving a terminal that does not exist in our dictionary of English words. This difference is later exemplified in Figures 12, 13, and 14 and discussed in the section titled Case Study.

Coordinated structures. A difference in the hierarchical arrangement of XPs with respect to coordinating conjunctions.

Headedness. An XP is headed by a terminal of an inappropriate syntactic category in one parse, but not the other, as in Figure 2, in which Parser 1 has as output an NP headed by a verb, whereas Parser 2 has not.

Identification of subordinating conjunction. Corresponding INs that are immediately dominated by PP in one parse and by SBAR in the other, as shown in Figure 3.

Infinitival complementation. Corresponding TOs that are immediately dominated by VP in one parse and by PP in the other. This difference is exemplified in Figure 11.

Internal constituent versus adjunct. A difference in which a constituent is parsed under a single XP node in one parse, whereas it is parsed as an adjunct (with an additional copy of the XP) in the other parse, as exemplified by the difference in positioning of the SBAR node in Figure 4.

Label projection. A pair of corresponding terminals in which the POS tag has projected its label to an XP in one parse but not in the other, as exemplified by the ADJP projection of *poor* present only in the output of Parser 1 in Figure 5.

Main predicate selection. In each parse, a different token is parsed as the head of the root node.

Premodified preposition. In one parse, there is an object to left of the preposition within the PP, whereas in the other parse, the preposition is the leftmost object within the PP, as in the different treatments of *due* in Figure 6.

Relative clause versus clausal complement. A pair of corresponding spans labeled SBAR, in which SBAR is headed by [IN that] in one parse and by [WDT that] in the other, as in Figure 7.

Presence of SINV, FRAG, and PRN nodes. There is an SINV, FRAG, or PRN node in one parse but not in the other.

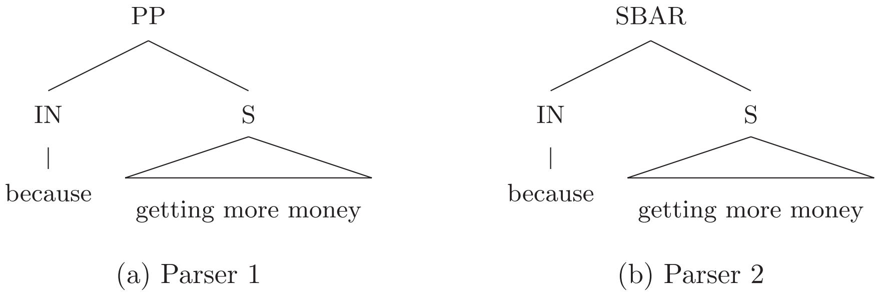
Sentential components. A span exhaustively dominated by S in one parse but not in the other, as shown in Figure 8. Parser 1 recognized that [NP it] and [VP will consists that …] are the components of a sentential constituent, whereas Parser 2 did not. Similarly, [NP People in big cities] and [VP would then buy a lot more cars] constitute the sole components of an S-node in the Parser 1 output, whereas the leftmost S in the Parser 2 output contains too much material.

Sentential constituency. This difference refers to different hierarchical arrangements of S nodes with respect to other S nodes, as exemplified in Figure 9, in which each S node has been indexed with its head. In the output of Parser 1, the S headed by *understand* contains the S headed by *started*, which contains the S headed by *have*. In the output of Parser 2, the S headed by *have* contains the S headed by *understand*.

Relations Among the Differences

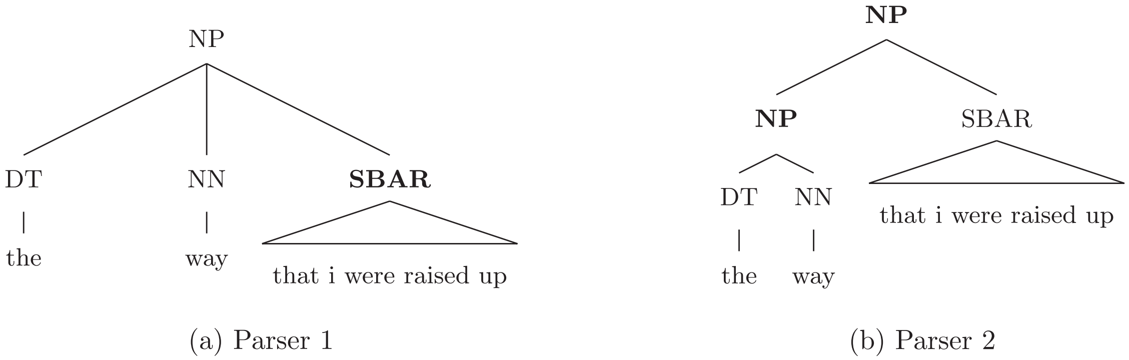
Owing to the interrelatedness of parsing phenomena, there is a high degree of overlap among many of these differences. For example, the presence of the *relative clause versus clausal complement* difference entails the presence of the *POS* difference, because part of the definition of a relative clause is the WDT tag. Although this is the only case of a true proper subset relation in our taxonomy, other differences are highly likely to co-occur.

For example, *identification of subordinating conjunction* has almost 100% overlap with *sentential components*. If an IN projects a PP, then its complement is likely to be an NP, as in the output of Parser 1 in Figure 10. However, if it projects



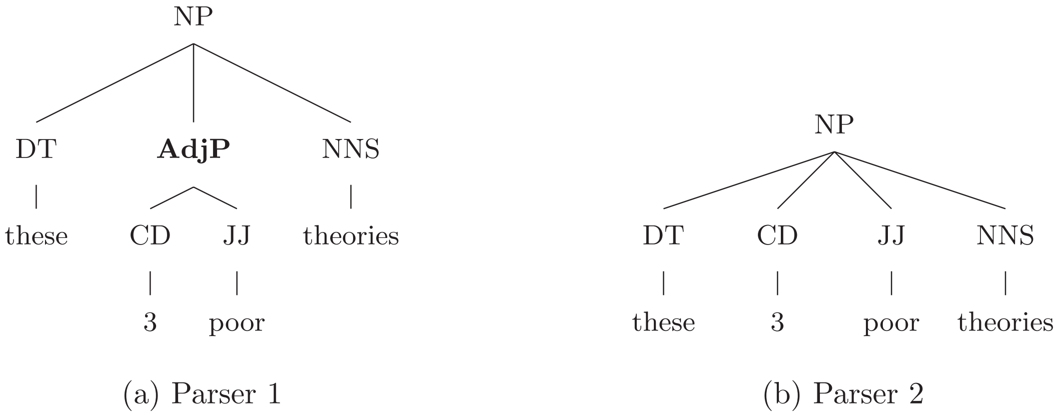
(a) Parser 1 (b) Parser 2

Figure 3 Identification of subordinating conjunction: " … *because getting more money*."



(a) Parser 1 (b) Parser 2

Figure 4 Internal constituent versus adjunct: " … *the way that i were raised up* … "



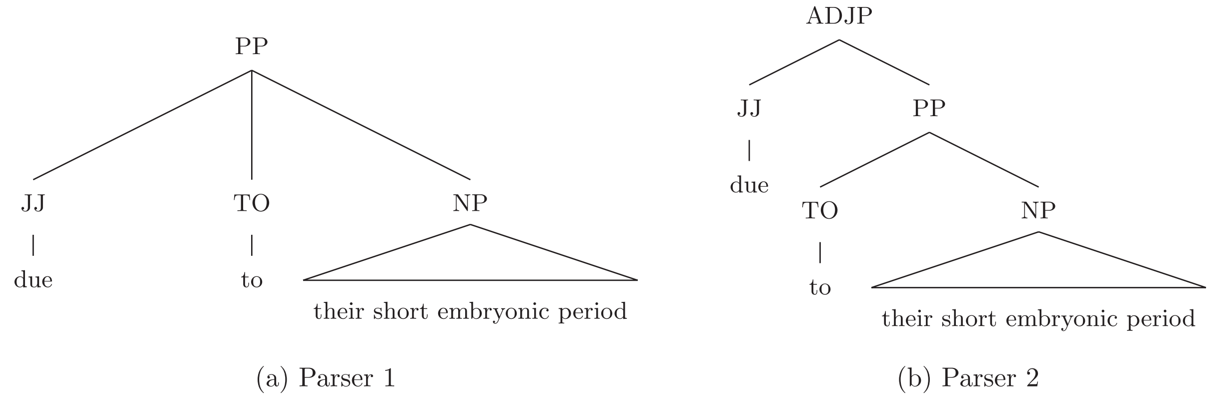
(a) Parser 1 (b) Parser 2

Figure 5 Label projection: " … *these 3 poor theories*."

an SBAR, then its complement is likely to be an S, as in the output of Parser 2. Therefore, when the complement of an IN has the same constituency in each parse, and there is an *identification of subordinating conjunction* difference, a *sentential components* difference is virtually entailed.

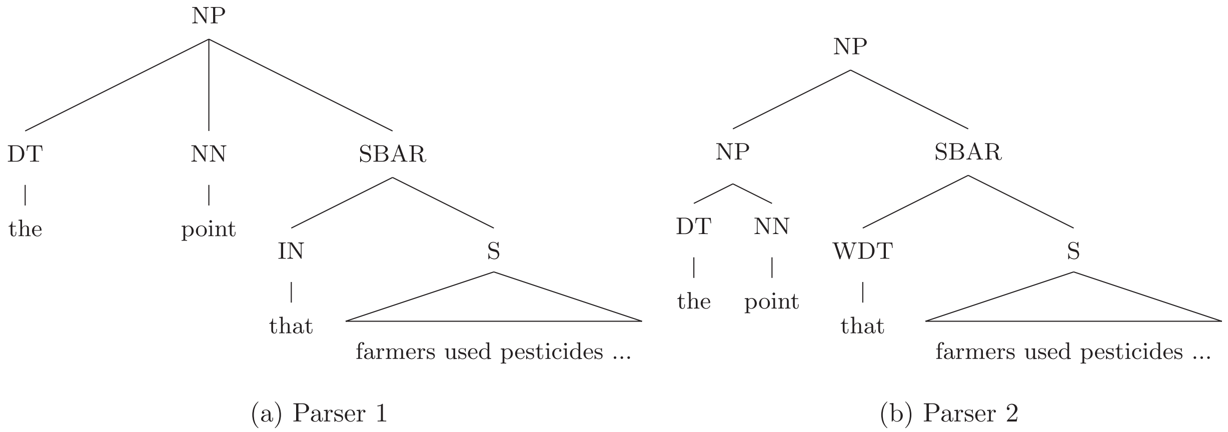
Another difference with a high degree of overlap with *sentential components* is *infinitival complementation*, as shown Figure 11. The *infinitival complementation* difference is sensitive to whether a TO is dominated by a PP, as in the output of Parser 1, or by a VP, as in the output of Parser 2. That the VP is often dominated by an S means that this difference is often accompanied by a *sentential components* difference.

Moreover, the VP/PP distinction is related to the *POS* difference: *hunt* was tagged as an NN by Parser 1 and as a VB by Parser 2. This further relates to a *constituent label* difference, because *to hunt shrimps* has a maximal label of PP in the output of Parser 1, whereas it has a maximal label of S in the output of Parser 2. Thus Figure 11 demonstrates a four-way interaction among differences. Such interactions are common in parsing phenomena, and accordingly, we expect many of our differences to co-occur.



(a) Parser 1 (b) Parser 2

Figure 6 Premodified preposition: "… due to their short embryonic period."



(a) Parser 1 (b) Parser 2

Figure 7 Relative clause versus clausal complement: "… *the point that farmers used pesticides on organic foods …*"

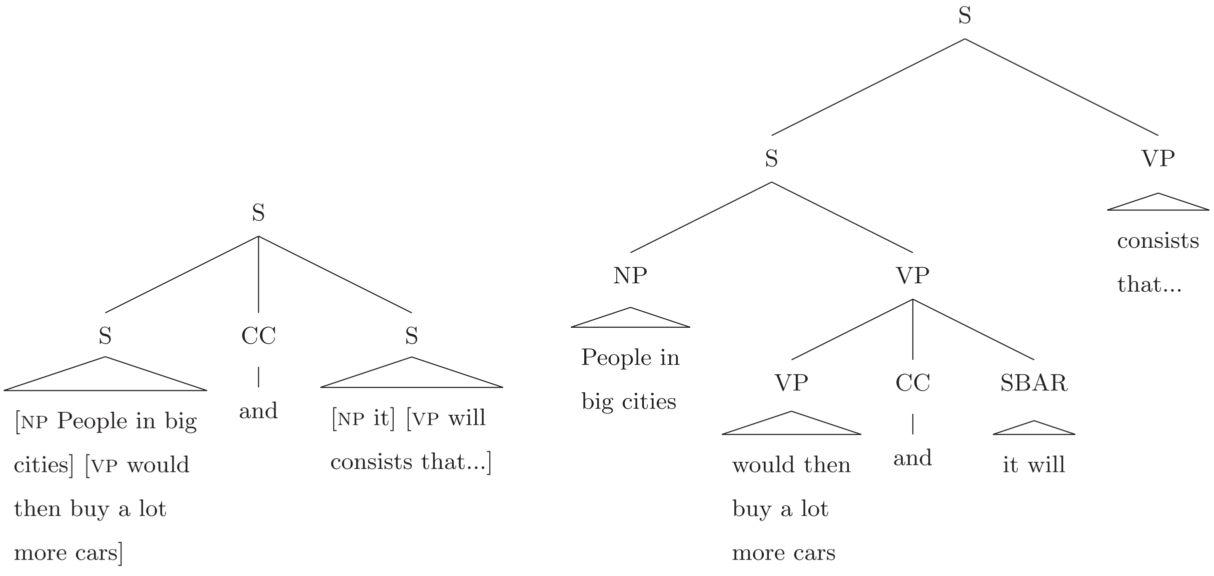
System Output

In this section, we present the output of our scheme and compare it to the output of the Kummerfeld et al. (2012) system. We compare on a corpus of 10,000 sentences of nonnative English.[9] We compare three parsers: the Stanford shift-reduce parser (SSR; Manning et al., 2014), the Berkeley parser (UCB; Petrov, Barrett, Thibaux, & Klein, 2006), and Zpar (ZP; Zhang & Clark, 2011). Our goal is to select from these three parsers the one that most meets our needs in terms of analyzing nonnative text for downstream applications. These parsers were chosen arbitrarily; our goal here is to demonstrate our proposed methodology with a concrete example rather than make a claim about any particular parser. The reported F-score for these three parsers on *WSJ* Section 23 is as follows: SSR = 90.4, UCB 90.1, ZP = 89.8.

Ten Thousand Sentences of Nonnative Text

We randomly extract approximately 10,000 sentences (almost 624,000 words) from a collection of essays written as part of a test of English proficiency usually administered to college-level students who are nonnative speakers of English. This corpus contains 690 essays written by speakers of varying proficiency levels ranging from poor to excellent.

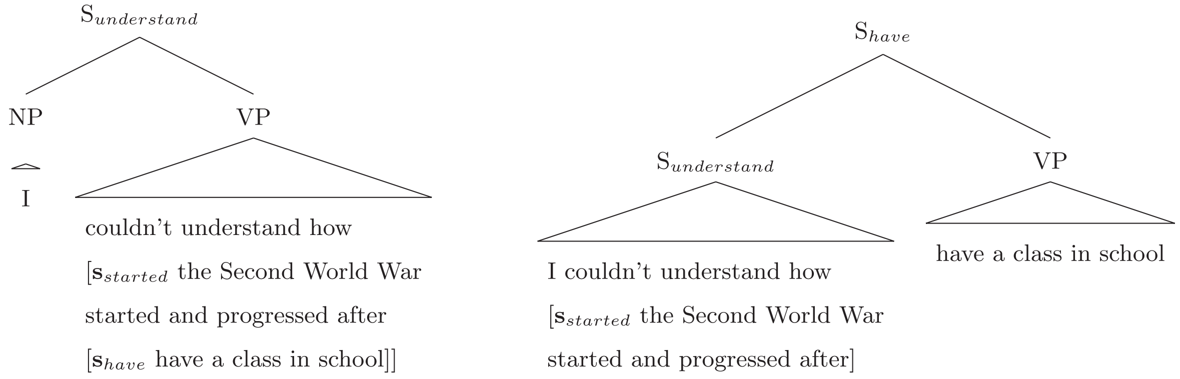
We parse the 10,000 sentences with each of the three parsers and compare the parsers in a pair-wise fashion. For each sentence, we compare the parse generated by two of the parsers and record any differences detected by the system. This results in three comparisons for each sentence. Table 1 gives a high-level overview of the parser comparisons on this dataset. The first row in the table, titled No. sentences with identical parses, shows how often the parsers agreed exactly with each other: The parsers do not agree very often, that is, in only between 5% and 7% of cases. The next row, titled Total differences, shows for each pair of parsers how many differences in total are found (where multiple differences may be found for the same sentence). The figures show that UCB and ZP are closest to each other, whereas the SSR differs most


S
S
CC
S
[NP People in big cities] [VP would then buy a lot more cars]
and
[NP it] [VP will consists that...]
S
S
VP
NP
VP
consists that...
People in big cities
VP
CC
SBAR
would then buy a lot more cars
and
it will

(a) Parser 1

(b) Parser 2

Figure 8 Sentential components: "*People in big cities would then buy a lot more cars and it will consists that it would be much more pollution in our environment*."


S_understand
NP
VP
I
couldn't understand how [S_started the Second World War started and progressed after [S_have have a class in school]]
S_have
S_understand
VP
I couldn't understand how [S_started the Second World War started and progressed after]
have a class in school

(a) Parser 1

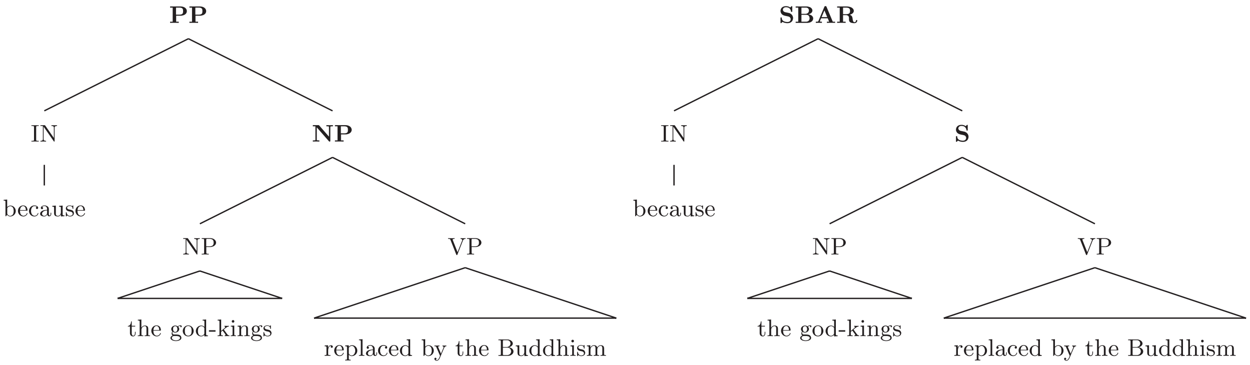
(b) Parser 2

Figure 9 Sentential constituency: "*I couldn't understand how the Second World War started and progressed after have a class in school*."

from UCB. Table 1 also gives some examples of the kinds of differences found when comparing each pair of parsers. The patterns for the individual differences tend to be the same as the patterns for the totals; however, this is not always the case, as can be seen for PP attachment, in which SSR and ZP are closest to each other.

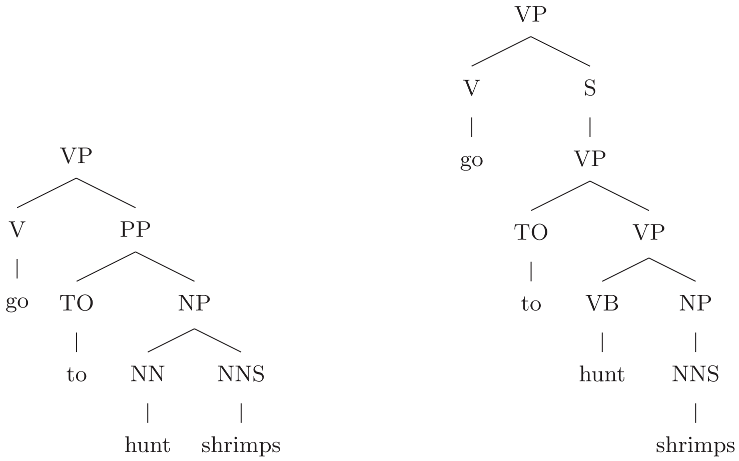
Comparison With Kummerfeld and Colleagues

We also run the Kummerfeld et al. (2012) system on this dataset. When running that system, it is necessary to choose a gold standard, and for this dataset, we do not have a gold standard. As previously mentioned, the creation of such a gold standard would be difficult and costly. Therefore, to examine the outputs of the Kummerfeld et al. system on these data, we run six comparisons in total (where each parser functions as the gold standard once per pair). The summary results are given in Table 2. The first thing to note is that the Kummerfeld et al. system is not symmetrical (by design), and so we see different numbers of so-called errors, depending on which parser we chose as the gold standard. This confirms that the Kummerfeld et al. system is only applicable when a clear gold standard is available. Otherwise, it becomes necessary to consider a pair of values for each pair of parser comparisons (one for each parser as the gold standard).



(a) Parser 1 (b) Parser 2

Figure 10 *Sentential components* and *identification of subordinating conjunction* are likely to co-occur.



(a) Parser 1 (b) Parser 2

Figure 11 Four-way interaction among *sentential components*, *infinitival complementation*, *POS*, and *constituent label*.

Another immediately obvious difference is that the overall total numbers of differences are quite divergent from our system, even for comparable definitions such as NP attachment. This is due to differences in our definitions and methodology. The Kummerfeld et al. (2012) system transforms a test parse into a gold standard by either moving a constituent or adding or deleting a node. Errors (i.e., differences) are defined by the kinds of transformations required to transform one tree into another. Our system defines differences according to rules that directly compare the parse trees with one another. For example, we identify a PP attachment site difference if a PP is attached to constituents *headed by* a different terminal in each tree. The Kummerfeld et al. system identifies a PP attachment site error if the transformation moves a PP or if incorrect bracketing dominates a PP.

Coverage Comparison

The final comparison we make with the Kummerfeld et al. (2012) system is in coverage. We determine the number of sentences with parses that are nonidentical in each of the three parser comparisons. In Table 3, we report the number of these sentences for which no difference was captured by each system. For the Kummerfeld et al. system, because we have two numbers depending on which parser we chose as the gold standard, we only report the minimum number of uncategorized errors. We find that the Kummerfeld et al. system is unable to categorize differences in many more sentences than our system, as noted in the All column. However, upon further investigation, we find that most of the differences in these sentences are differences in POS tags, as noted in the POS column. The Kummerfeld et al. system does not identify

Table 1 Summary and Breakdown by Difference Type on 10,000 Sentences of Nonnative Text

	SSR-UCB		SSR-ZP		UCB-ZP	
	n	% of ttl.	*n*	% of ttl.	*n*	% of ttl.
No. sentences with identical parses	1,639		1,936		2,066	
Total differences	66,742		60,578		56,208	
NP attachment	3,693	5.5	3,617	6.0	2,646	4.7
PP attachment	2,795	4.2	2,230	3.7	2,520	4.5
VP attachment	1,380	2.1	1,249	2.1	779	1.4
SBAR attachment	1,323	2.0	1,179	1.9	1,168	2.1
ADVP attachment	965	1.4	880	1.5	758	1.3
S attachment	925	1.4	863	1.4	991	1.8
ADJP attachment	163	0.2	165	0.3	106	0.2
Other attachment	185	0.3	159	0.3	130	0.2
Mismatched label attachment	2,487	3.7	2,182	3.6	1,918	3.4
Sentential components	18,025	27.0	16,067	26.5	14,562	25.9
Label projection	9,097	13.6	8,303	13.7	7,740	13.8
POS	7,300	10.9	6,757	11.2	7,675	13.6
Coordinated structures	3,081	4.6	2,780	4.6	2,139	3.8
Headedness	2,916	4.4	2,647	4.4	2,800	5.0
Sentential constituency	2,771	4.2	2,567	4.2	2,295	4.1
Constituent label	2,249	3.4	2,153	3.6	2,166	3.9
Main predicate selection	1,764	2.6	1,625	2.7	1,145	2.0
POS of misspelled terminal	1,436	2.2	1,273	2.1	1,352	2.4
Constituency of misspelled item	1,311	2.0	1,118	1.8	1,135	2.0
Presence of FRAG node	806	1.2	853	1.4	417	0.7
Internal constituent vs. adjunct	759	1.1	683	1.1	636	1.1
Identification of subordinating conjunction	474	0.7	413	0.7	379	0.7
Presence of PRN node	186	0.3	153	0.3	190	0.3
Relative clause vs. clausal complement	181	0.3	242	0.4	146	0.3
Presence of SINV node	169	0.3	123	0.2	128	0.2
Premodified preposition	156	0.2	145	0.2	160	0.3
Infinitival complementation	145	0.2	152	0.3	127	0.2

Table 2 Kummerfeld et al. (2012) Output for the 10,000 Dataset

	SSR-UCB	UCB-SSR	SSR-ZP	ZP-SSR	UCB-ZP	ZP-UCB
Clause attachment	3,925	4,321	3,192	3,377	3,388	3,115
Coordination	3,079	2,673	2,447	2,338	1,826	2,091
Different label	3,049	2,841	2,973	2,951	2,202	2,366
Modifier attachment	2,319	2,374	1,893	1,886	2,042	2,002
NP attachment	1,701	1,449	1,301	1,213	1,074	1,248
NP internal structure	1,717	1,599	1,514	1,359	1,432	1,434
PP attachment	3,567	3,500	2,661	2,571	3,212	3,128
Single-word phrase	5,599	5,635	4,733	4,807	4,510	4,519
Unary	5,233	5,877	4,996	5,337	4,130	3,901
VP attachment	1,206	1,018	1,131	1,034	686	780

Note. The parser on the left of each pair is considered to be the gold standard.

this difference because it does not involve a change of bracketing. If we ignore the POS tag differences, as we do in the All-POS column, we find that our systems are roughly equivalent in terms of coverage, with our system able to categorize slightly more of the sentences.

Comparison With Related Work

Other research has endeavored to categorize parser differences in linguistically informative ways, but most of it has focused on dependency parsers, and all of the work we are aware of assumes the presence of a gold standard (e.g., Hara, Miyao,

Table 3 Nonidentical Parses in 10,000 Sentences of Nonnative Text Not Captured by Our System Versus the Kummerfeld et al. (2012) System

		Kummerfeld et al.		
	Our system (all)	All	POS	All-POS
SSR-UCB	131	694	591	103
SSR-ZP	126	932	794	138
UCB-ZP	86	1,001	885	116

Note. Sentences in the POS column are a subset of those in the All column.

& Tsujii, 2009; Rimell, Clark, & Steedman, 2009). Additionally, most other work performs some sort of manipulation or tagging of the parse trees and then categorizes differences based on the manipulation. Our system identifies differences in constituency parsers by directly comparing the parses themselves.

For example, Kulick et al. (2014) classified *errors* according to a gold standard. They converted constituency trees from UCB to dependency trees and automatically tagged the nonterminal nodes with 49 possible regular expressions that enhance the node label with information such as headedness, recursivity, and directionality of modification. They then compared the output of several parsers against the gold standard and provided measures of attachment site and span accuracy, as well as measures of matching heads. This allowed them to draw conclusions, such as "the parser does well on determining the right-edge of verbal structures [in *WSJ* data, but not on out-of-domain data]" (p. 672).

Using a hand-annotated gold standard of 1,000 sentences, Bender, Flickinger, Oepen, and Zhang (2011) evaluated the precision and recall of seven dependency parsers on 10 linguistic phenomena in which dependency plays an explicit role, including control structures, tough constructions, and absolutive constructions. Their automation employed 364 phenomenon-specific regular expressions that operated on dependency labels. In at least some cases, the regular expressions were parser-specific as well. It was concluded, for example, that most of the parsers evaluated do well on identifying the head[10] in control structures, some parsers do far better than others on tough constructions, and all parsers perform poorly on absolutive predicates.

Case Study

In this section, we present the results of a manual analysis carried out to select the best parser of the three for a task involving automated processing of nonnative text. We focus on the *constituency of misspelled item* difference because our downstream application is the automated analysis of nonnative text, in which there are often many misspelled items.

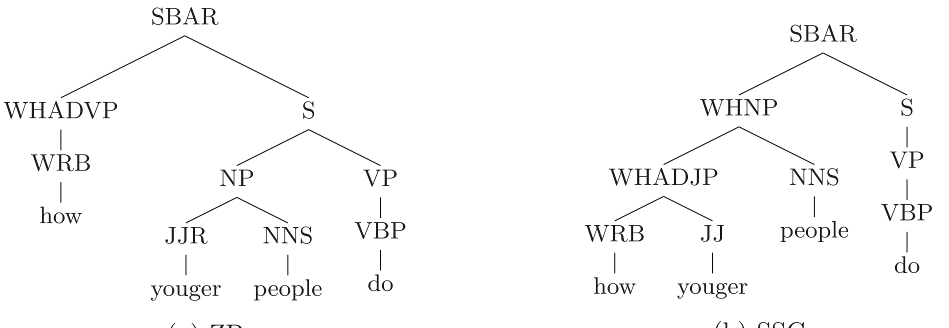
Method

We sampled 50 pairs of parses with this difference from each of the three parser comparisons for a total of 150 pairs. Two annotators with graduate degrees in linguistics manually examined each pair and made a qualitative judgment on which parser built the most correct constituency around misspelled items.

For example, the ZP parse in Figure 12a was judged to be better than the SSR parse in Figure 12b because *youger* rightly forms a constituent with *people* to the exclusion of *how*, because in functional terms, *youger* is the single modifier of *people* and *how* scopes over the entire phrase (assuming the target to have been *how younger people do*[11]). The SSR parse creates a constituent of *how* and *youger*, which misses both the modification and scopal relationships.

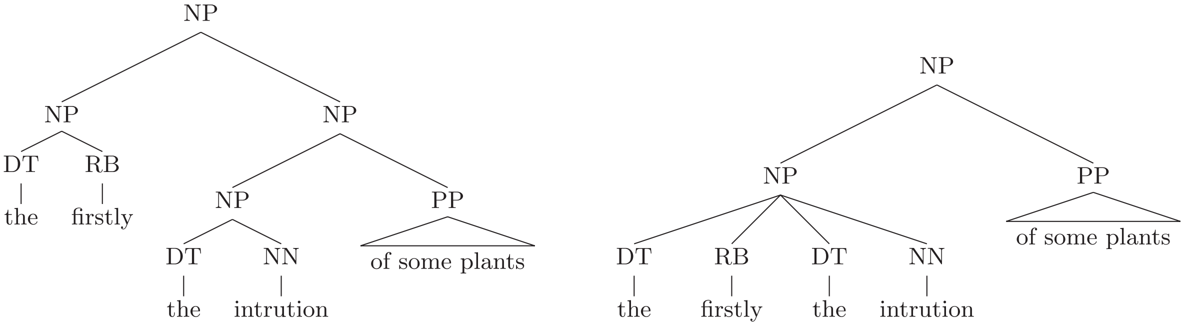
Annotators prioritized considerations of constituency over node labels so that a parse with correct constituency but incorrect labels would be judged superior to a parse with correct labels but incorrect constituency. Additionally, they only considered the constituency built around the misspelled item and ignored differences elsewhere in the tree.

It was often the case that annotators could not determine which parse was better. Sometimes this was because both parses were obviously incorrect or both parses were equally acceptable (e.g., owing to an ambiguity in coordination). There were also cases in which the indeterminacy was because the text was too garbled, as in Figure 13, or the difference was thought to be too subtle, such as whether *ok* should form a constituent with the SBAR node in Figure 14. Whenever annotators found themselves in internal debate over the syntactic analysis, they gave a judgment of "unable to determine." Annotators reported that when they were able to make a firm judgment, it was relatively immediate and obvious.



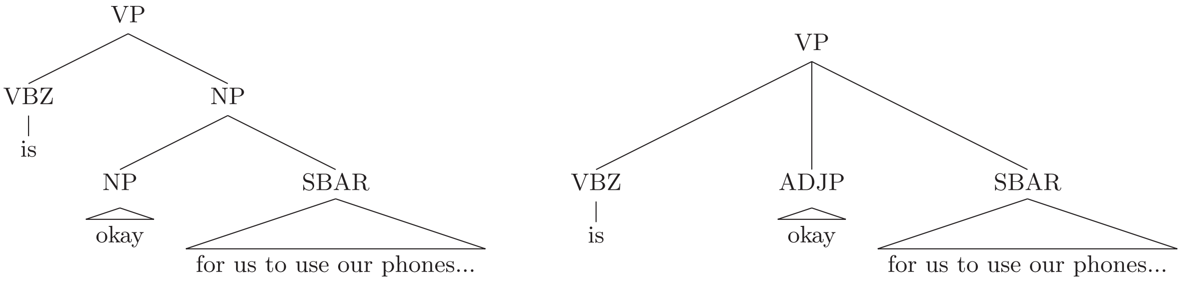
(a) ZP (b) SSC

Figure 12 ZP parse judged superior to SSC parse … *how youger people do* …



(a) Parser 1 (b) Parser 2

Figure 13 Sentence judged too garbled to determine the better parse.



(a) SSR (b) UCB

Figure 14 Difference judged too subtle to determine the better parse.

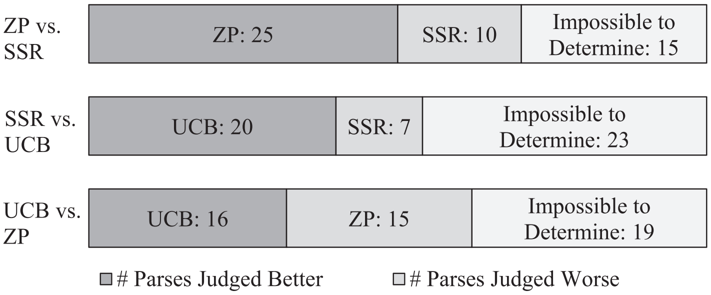


Figure 15 Judgments for *constituency of misspelled item* difference.

Results

The annotators independently evaluated nonoverlapping sets, and no formal agreement was calculated; however, annotators reported the same general trends. The results of their evaluation are presented in Figure 15.

The constituents involving misspelled items that were judged best were those built by ZP and UCB, with both parsers outperforming SSR by a ratio of greater than 2:1, as can be seen in the first two rows of the chart. In the comparison between UCB and ZP, neither parser clearly outperformed the other, because the difference between them is only one judgment. We observe a consistent robust pattern, free of paradox. For example, if ZP outperformed UCB, and UCB outperformed SSR, we would regard it as suspicious indeed if SSR were to outperform ZP. On the basis of these results, we are able to make an informed choice to remove the SSR parser from consideration. The choice between UCB and ZP would then be made on other considerations, such as speed.

Conclusion

We have presented our annotation scheme for annotating *differences* between the output of two parsers when a gold standard is either not available or not desirable, as in the case of nonnative text. We have outlined and exemplified a methodology for choosing a parser according to its suitability for a downstream task, based on an analysis that makes use of rules to automatically detect differences as defined by our annotation scheme. In our example, we evaluate three parsers according to how well they build constituency around misspelled items, a difference we choose according to the nature of our downstream application. We take a small sample of parses for manual comparison and evaluation. On the basis of the results of our evaluation, we are able to make a linguistically informed decision about which parsers are best suited to our application in the absence of a gold standard. Future work will include comparing parsers in a task-based evaluation and investigating whether the choices made based on manual inspection of differences lead to meaningful differences in overall performance.

Acknowledgments

We would like to thank Patrick Houghton for his help with annotation. We would also like to thank Keelan Evanini, Klaus Zechner, Paul Deane, and three anonymous reviewers for their helpful comments and suggestions on earlier drafts.

Notes

[1] https://github.com/BLLIP/bllip-parser.

[2] This work is designed for constituency trees and would require some work to extend to dependency graphs.

[3] Our implementation is in Python and uses the NLTK library for tree traversing.

[4] Most of the differences we found in the second round of analysis had to do with the parsing of punctuation.

[5] We use Stanford Core NLP for identifying heads.

[6] In some cases, the label of the constituent changed in addition to the attachment site. In these cases, we categorized the difference as *Mismatched Label Attachment*.

[7] We use the Python package enchant with U.S. spelling dictionaries to carry out spelling error detection.

[8] In our automatic rules, the span of the first non-unary-branching node dominating the misspelled item is examined and a difference is noted when the spans are not identical.

[9] In preliminary work, we also compared on the WSJ Section 23 corpus and observed similar results.

[10] In the dependency sense, not the syntactic one.

[11] In cases of ill-formed sentences, annotators assumed a target of the closest well-formed sentence, whenever it was possible to make such a determination.

References

Bender, E. M., Flickinger, D., Oepen, S., & Zhang, Y. (2011). Parser evaluation over local and nonlocal deep dependencies in a large corpus. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 397–408). Stroudsburg, PA: Association for Computational Linguistics.

Cahill, A., Gyawali, B., & Bruno, J. (2014). Self-training for parsing learner text. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages* (pp. 66–73). Dublin, Ireland: Dublin City University.

Charniak, E., & Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL05)* (pp. 173–180). Stroudsburg, PA: Association for Computational Linguistics.

Hara, T., Miyao, Y., & Tsujii, J. (2009). Descriptive and empirical approaches to capturing underlying dependencies among parsing errors. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (Vol. 3, pp. 1162–1171). Stroudsburg, PA: Association for Computational Linguistics.

Kulick, S., Bies, A., Mott, J., Kroch, A., Santorini, B., & Liberman, M. (2014). Parser evaluation using derivation trees: A complement to evalb. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Vol. 2, pp. 668–673). Stroudsburg, PA: Association for Computational Linguistics.

Kummerfeld, J. K., Hall, D., Curran, J. R., & Klein, D. (2012). Parser showdown at the Wall Street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 1048–1059). Stroudsburg, PA: Association for Computational Linguistics.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). *The Stanford Core NLP natural language processing toolkit*. Paper presented at the 52nd annual meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD.

Petrov, S., Barrett, L., Thibaux, R., & Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics* (pp. 433–440). Stroudsburg, PA: Association for Computational Linguistics.

Rimell, L., Clark, S., & Steedman, M. (2009). Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (Vol. 2, pp. 813–821). Stroudsburg, PA: Association for Computational Linguistics.

Zhang, Y., & Clark, S. (2011). Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics, 37*(1), 105–151.

Suggested Citation:

Bruno, J. V., Cahill, A., & Gyawali, B. (2016). *A protocol for annotating parser differences* (ETS Research Report No. RR-16-02). Princeton, NJ: Educational Testing Service. http://dx.doi.org/10.1002/ets2.12086

Action Editor: Keelan Evanini

Reviewers: Paul Deane and Klaus Zechner