Comparative Analysis of Parsing Techniques: Evaluating Performance and Error Patterns in Two Contrasting Constituency Parsers

Tim Luka Horstmann
Lucy Cavendish College
University of Cambridge
tlh45@cam.ac.uk

Abstract

This study presents a comparative analysis of two contrasting constituency parsers, the Charniak-Johnson parser and the Berkeley Neural Parser, analysing their parsing performance on 11 selected sentences. In addition to an initial quantitative analysis, the main focus of this work lies on a qualitative evaluation of the two parsers, which goes beyond a simple evaluation of performance metrics and identifies the error patterns of the parsers. While the quantitative analysis shows an overall better performance of the Charniak parser, the deeper qualitative analysis reveals that the Berkeley parser generally produces better parses and demonstrates a superior linguistic understanding. This study underscores that a linguistically meaningful parser evaluation requires (qualitative) analysis beyond simple performance metrics.

1 Introduction

Syntactic parsing, which describes the process of analysing sentences to reveal their grammatical structures and relationships, is an active area of research in computational linguistics and often a crucial first step in Natural Language Processing (NLP) tasks (Bai et al., 2023). It encompasses two main types: constituency parsing and dependency parsing (Jaf and Calder, 2019). While the latter examines the (linear) grammatical dependencies amongst words in a sentence, constituency parsing is concerned with constructing a parse tree of sentence constituents like noun and verb phrases, explaining a sentence's hierarchical structure.

Constituency parsing has evolved from traditional parsers based on context-free grammar (CFG), which employs rules to build sentence structures (Collins, 1997; Charniak, 1997), to probabilistic parsers that incorporate complex lexical features like head words for enhanced accuracy (Klein and Manning, 2003). More recently, modern neural parsers have emerged that, despite leaving traditional concepts behind, are shown to implicitly learn and capture the key elements previously provided explicitly (Gaddy et al., 2018).

Given the importance of constituency parsing in NLP and the plethora of available parsers, understanding their specific strengths and weaknesses is crucial for assessing their impact on downstream tasks. Although contemporary research often only evaluates parsers on the basis of simple performance metrics, research has shown that such an analysis is often not sufficient to adequately account for the high number of different parsers and provide linguistically meaningful insights into their performance (Kummerfeld et al., 2012). Building on this notion, this study contributes to going beyond the idea of merely assigning simplistic metrics for parser evaluation and instead also conducts a qualitative assessment of parsers to reveal their behaviour in a linguistically meaningful manner.

To this end, two commonly used constituency parsers are selected as representatives of different parsing paradigms: the Charniak-Johnson parser (Charniak and Johnson, 2005) and the Berkeley Neural Parser (Kitaev and Klein, 2018; Kitaev et al., 2019). I compare both parsers based on 11 sentences *S* of varying length and complexity (see Appendix A), for which gold standard parses were generated by human annotators via crowdsourcing.

An initial quantitative analysis of both parsers reveals the Charniak parser's marginally better overall performance, while the Berkeley parser exhibits slightly higher tagging accuracy. The subsequent qualitative analysis examines the tagging performance of both parsers in detail before their parsing behaviour is classified into five error types and analysed in depth. Following these analyses, as well as a critical examination of the gold standard, the analysis concludes that the Berkeley parser generally creates better parses for the given sentences.

This study emphasises that quantitative parser evaluations should be complemented by nuanced and linguistically informed qualitative assessments.

2 Background

The two constituency parsers presented in this paper follow contrasting methodologies and can offer insights into the evolution of parsing strategies. While the Charniak-Johnson parser (also called *BLLIP reranking parser*) by Charniak and Johnson (2005) is based on a mix of rule-based and statistical techniques, representing a traditional approach, the Berkeley Neural Parser (also called *benepar*) by Kitaev and Klein (2018) and Kitaev et al. (2019) is a modern neural and entirely data-driven method.

2.1 Charniak-Johnson Parser ("Charniak")

The Charniak parser, developed in 2005, displayed a shift in constituency parsing algorithms, moving away from solely employing rule-based algorithms like the Cocke-Kasami-Younger (CKY) algorithm (Kasami, 1965; Younger, 1967) to combining such dynamic programming algorithms with probabilistic models (Charniak and Johnson, 2005). It set a new state-of-the-art parsing performance in the early 2000s (Charniak and Johnson, 2005) and is still used in contemporary research as a benchmark for modern parsers (e.g., Yang et al., 2022).

The Charniak parser consists of two stages: a generative constituent parser that generates 50-best parses, followed by a discriminative maximum entropy (*MaxEnt*) reranker to select the most accurate of these parses (Charniak and Johnson, 2005). The first-stage parser is a probabilistic parser that utilises a so-called *coarse-to-fine* strategy. Here, a dynamic programming bottom-up CFG parser is employed to generate coarse parses (i.e., a *parse forest*) based on a simple grammar. This parse forest is then pruned and subsequently evaluated through a fine-grained probabilistic model (i.e., using a more complex grammar including lexical features) to refine the coarse-grained states found before. The first stage parser returns the 50 best parse trees. The second-stage reranker takes these parse trees as input to another model, a MaxEnt estimator (Riezler et al., 2001), that selects the best parse tree based on features from 13 different feature schema (Charniak and Johnson, 2005).

2.2 Berkeley Neural Parser ("Berkeley")

In contrast to the Charniak parser, the Berkeley (Neural) parser, as introduced in 2018 (Kitaev and Klein, 2018) and refined in 2019 (Kitaev et al., 2019), is based on neural networks. Achieving state-of-the-art results, this approach represents another shift in the evolution of parsing strategies towards modern neural architectures, reflecting their significant recent successes (Kitaev and Klein, 2018; Gaddy et al., 2018; Fried et al., 2019).

The parser follows an encoder-decoder design (see Figure 1). Kitaev and Klein explored different input encodings and achieved the best results with ELMo word representations (Peters et al., 2018) that are fed into the encoder, which creates wordwise vector representations with context for each word (Kitaev and Klein, 2018). The encoder follows the design proposed by Vaswani et al. (2017) but implements a *factored self-attention*, in which attention probabilities for content and position information are considered separately (Kitaev and Klein, 2018). The encoder's output vectors are used to calculate span scores $s(i,j,l)$ according to the chart parsing algorithm by Stern et al. (2017), with i and j denoting a constituent's fencepost positions in a sentence and l being the constituent's label (Kitaev and Klein, 2018). The span scores are transferred to the decoder, which is also adapted from the chart parser and incorporates additional modifications proposed by Gaddy et al. (2018). Here, out of all possible parse trees T, the optimal tree $\hat{T}$ is determined through a CKY-based algorithm solving $\hat{T} = \arg\max_T \sum_{(i,j,k)\in T} s(i,j,l)$ (Kitaev and Klein, 2018).

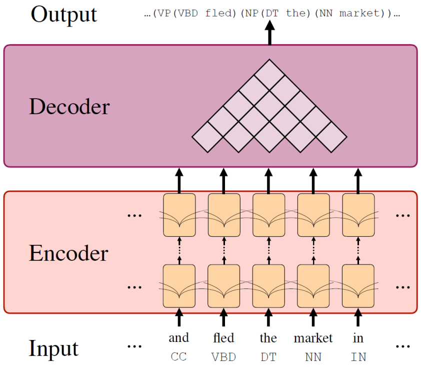


Figure 1: Architecture of the Berkeley parser, image by Kitaev and Klein (2018)

In addition to optimisations for multilingual parsing and other improvements, the refinement of the Berkeley parser presented by Kitaev et al. (2019) is characterised above all by the introduction of pre-training for determining word representations. Specifically, the input part of the model (and thus also ELMo) is replaced by a BERT model (Devlin et al., 2019), which leads to significant improvements in parsing performance (Kitaev et al., 2019).

2.3 Methodological Details for Reproducibility

Both the Charniak[1] and the Berkeley[2] parser implementations are available via GitHub. All eleven input sentences were parsed using the following release versions of the respective Python packages: *bllipparser 2021.11.7*[3] and *benepar 0.2.0*[4]. For the Charniak parser, the *RerankingParser* based on the *WSJ+Gigaword-v2* parsing model was used; for the Berkeley parser, the *benepar_en3* model. All other parameters were left unchanged.

3 Evaluation

In this section, I compare the performance and error patterns of the Charniak and Berkeley parsers against the 11 given gold standard parses. The evaluation is divided into a quantitative and qualitative analysis, with particular emphasis on the qualitative part. While the quantitative evaluation of the parsers in Section 3.1 is primarily carried out using the PARSEVAL metrics (Black et al., 1991), the qualitative evaluation in Section 3.2 aims to go beyond the limited expressiveness of these single metrics by following a three-step process: first, assessing the part-of-speech (POS) tagging accuracy of both parsers; second, mechanically categorising parsing errors to identify prevalent types; and third, analysing specific instances and error types to gain deeper insights into the parsing intricacies.

All analyses in this section are based on the assumption that the provided gold standard parses are accurate (an in-depth discussion of the Gold standard is presented in Section 4). It should also be noted that the tool-based evaluation of the Charniak parser made manual adjustments necessary so that its parse trees could be compared with the Berkeley and Gold parse trees. Hyphenated words in input S8 and S10A were separated by whitespace. Post-parsing, "-" in S7 was manually tagged with *:* and the apostrophe before the first "there" in S9, incorrectly tagged as *POS*, was tagged with *''*.

3.1 Quantitative Analysis: EVALB Metrics

The PARSEVAL metrics for both parsers were determined using the EVALB[5] tool for parser evaluation, a canonical realisation of the PARSEVAL metrics, which abstracts from grammar-specific details. In addition to overall performance metrics such as labelled recall (R), labelled precision (P), and $F1$, which are calculated as described in the equations below (Black et al., 1991), the bracket behaviour and POS tag accuracy were determined.

$$R = \frac{\text{\# of correct constituents in parser's parse}}{\text{\# of total constituents in gold parse}}$$

$$P = \frac{\text{\# of correct constituents in parser's parse}}{\text{\# of total constituents in parser's parse}}$$

$$F1 = \frac{2PR}{P+R}$$

To prepare the parser outputs and gold standard trees for evaluation using EVALB, the gold parses were adjusted to match the format generated by both parsers. This includes the removal of index numbers, the formation of correct past and plural verb forms (e.g., "break+ed" → "broke"), capitalisation, and replacing left and right brackets with "-LRB-" and "-RRB-" respectively. The *S1* root node was removed from all Charniak parses, while the Berkeley parse trees remained unchanged.

All EVALB metrics were calculated for both parsers across all 11 sentences in comparison with the gold standard. The full per parser results of this analysis are shown in Appendix C.

An evaluation of the results reveals a similar overall performance of both parsers in terms of R, P, and F1. With an average F1 score of 54.16, the Charniak parser performs slightly better overall than the Berkeley parser (53.94) on the given sentences. It is generally noticeable that the performance of both parsers decreases with increasing sentence length and more complex sentences such as 9, 10A and 10B, which contain intricate elements like nested structures, lists, and technical terminology, as indicated by worse performance metrics for these sentences. The number of crossing brackets also increases for such sentences. This number counts how often the parsers' structures cross over the gold standard ones and reflects structural mismatches. The average crossing numbers (Charniak: 2.18, Berkeley: 2.55) show that the Berkeley parser has more difficulties in maintaining the parse tree structure (i.e., syntactic boundaries or relationships) of the gold standard, highlighting its different approach to syntactic analysis. Although both parsers demonstrate a high tagging accuracy on the total 205 words across all sentences (Charniak: 90.73%, Berkeley: 91.71%), the Berkeley

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

[2] https://github.com/nikitakit/self-attentive-parser

[3] https://pypi.org/project/bllipparser/

[4] https://pypi.org/project/benepar/

[5] https://nlp.cs.nyu.edu/evalb/

parser is slightly more accurate in assigning POS tags to words. This enhanced performance might be attributed to its advanced neural architecture, incorporating a pre-trained BERT language model, which is known to perform well in language understanding tasks (Devlin et al., 2019). At the same time, the Charniak parser's abilities fundamentally rely on hand-crafted rules that are limited in encompassing the full spectrum of linguistic diversity.

In conclusion, the quantitative analysis reveals that while the Charniak parser's overall parsing performance is marginally better in overall F1 score and average crossing than the Berkeley parser, the latter demonstrates a slightly higher POS tagging accuracy. Nonetheless, these metrics only offer a limited view that primarily focuses on describing the broad accuracy of the parsers without exposing their deeper linguistic strengths and weaknesses.

3.2 Qualitative Analysis: In-Depth Parsing Evaluation

Building on the initial findings of the quantitative evaluation, this section presents an in-depth qualitative analysis of the two parsers. This critical analysis will enable a comprehensive understanding of the underlying factors contributing to the observed inscrutable scores and is crucial for a well-founded assessment of both parsers.

3.2.1 POS Tagging Accuracy

As the foundational step of parsers (and many other NLP tasks), POS tagging plays an important role. It is essential for identifying the grammatical structures of sentences and can have a significant impact on downstream NLP tasks (Chiche and Yitagesu, 2022). Although both parsers demonstrated an overall high POS tagging accuracy in the quantitative analysis, it is necessary to understand which POS tags they failed to correctly identify, as the severity and implications of this can vary widely. Confusing content words (e.g., misidentifying a noun as a verb), for example, can drastically change the meaning of a sentence. Figure 2 and Figure 3 show which POS tags, as given by the Penn Treebank[6], the parsers confused at least twice (i.e. predicted a tag other than the gold standard).

Both parsers predicted *UH* (interjection) or *IN* (preposition or subordinating conjunction) instead of *RB* (adverb), indicating a limited understanding of the usage context of these POS types. It is noteworthy, however, that both parsers agreed on tagging the second "as" in "as well as" in S5 as *IN* and "no" in S9 as *UH*. Since "as well as" acts as a multi-word preposition according to the Cambridge Dictionary (Cambridge University Press, 2024) and "no" as an interjection according to the Penn Treebank tagging guidelines (Santorini, 1990), both parsers seem to be correct for these words, indicating potential for refinement of the gold standard and thus motivating its critique in Section 4.

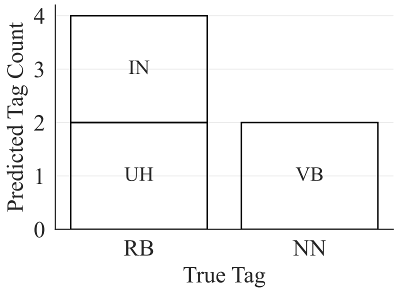


Figure 2: Recurring POS tag confusions (Charniak)

The Charniak parser twice mistook a noun for a verb (S1: "drum" and S10B: "tag"), as the words in question have dual usage in English. This confusion suggests possible limitations in the disambiguation capabilities of the parser. The Berkeley parser, on the other hand, twice failed to recognise the base form of the verbs "do" and "have" in S10B — behaviour that, in practice, might affect the interpretation of actions and events. It also wrongly tagged "there" in S9 as a foreign word (*FW*), which might stem from its isolation by quotation marks, typical for foreign words in English and potentially prevalent in the data the parser was trained on.

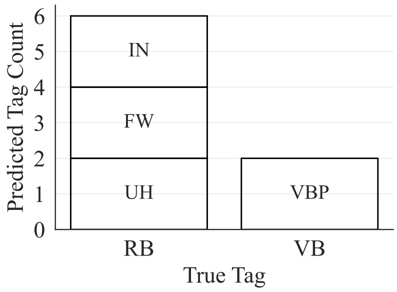


Figure 3: Recurring POS tag confusions (Berkeley)

3.2.2 Categorisation of Parsing Errors

To understand the parsing behaviour of the Charniak and Berkely parsers in detail and uncover the basis for the results of the quantitative analysis, this

[6] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Sent.	NP Int. Struct.		1-Word+Unary		Mod. Attach.		PP Attach.		Diff. Label	
	Charniak	Berkeley	Charniak	Berkeley	Charniak	Berkeley	Charniak	Berkeley	Charniak	Berkeley
1	2	2	0	0	0	0	0	0	0	0
2	1	1	2	2	1	1	0	0	0	0
3	0	0	2	3	1	0	0	0	2	0
4	1	1	4	4	1	1	0	0	0	0
5	0	0	6	6	0	0	0	0	4	4
6	3	3	1	1	1	1	1	1	2	2
7	1	0	7	11	0	0	0	0	4	2
8	0	0	6	7	3	2	1	1	2	2
9	0	1	14	11	6	6	7	6	4	10
10 A	4	3	4	4	1	2	2	2	0	2
10 B	5	7	8	7	8	3	10	11	2	0
$\sum$	17	18	54	56	22	16	21	21	20	22

Table 1: Comparison of qualitative performance of Charniak and Berkeley parsers

and the following section present a thorough evaluation of the error behaviour of both parsers against the gold standard. First, errors are categorised into five key types:

- *NP Internal Structure*: incorrect parsing of the internal structure of a noun phrase (NP), e.g. *(NP (DT The) (JJ old) (NN car))*, where *(JJ old)* should first form a *NP* with *(NN car)*.
- *1-Word+Unary*: incorrect parsing of single word phrases (SWP) or unary productions, e.g. when *(PRT(RP(up)))* should only be *(RP(up))*
- *Modifier Attachment*: incorrect attachment of modifiers (e.g., particles) to words, e.g. *(VP (VBD broke) (PRT (RP up)) (PP ...))*, where *broke* and *up* should first form a (phrasal) verb.
- *PP Attachment*: incorrect attachment of prepositional phrases (PP) to the wrong part of a sentence, e.g. misplacing *(PP (IN in) (NP (NNP February)))*
- *Different Label*: incorrect assignment of a label to a part of the sentence, e.g. labelling a sentence (*S*) as a subordinate clause (*SBAR*)

This categorisation, derived from the work by Kummerfeld et al. (2012), provides a framework for the detailed analysis in Section 3.2.3. Using the analysis tool provided by Kummerfeld et al.[7], the parse trees of both parsers were compared against the gold standard. The results for the five presented error types are recorded in Table 1.

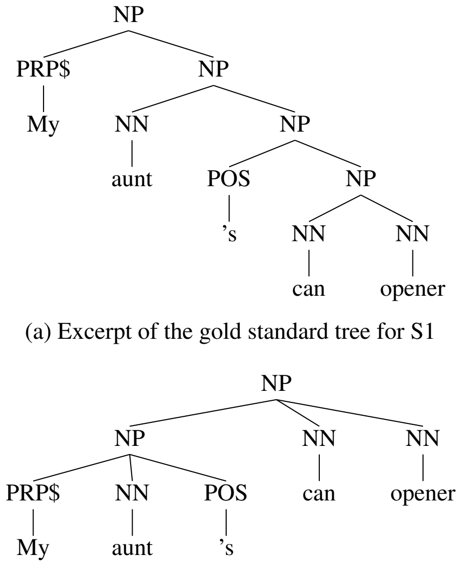
The analysis reveals that both parsers made the highest number of errors in the *1-Word + Unary* category, which indicates challenges in resolving ambiguities in single-word interpretations that human interpreters (who created the gold standard) do not have. Although the Charniak parser appears to have yielded superior results at first glance — it makes fewer errors than the Berkeley parser in three categories — the latter performs equally well or better in the two significant error categories *Modifier Attachment* and *PP Attachment*, where errors can strongly influence the semantic interpretation of a sentence. For instance, a sentence such as "I discussed the problem with my friend" can be about discussing a problem involving a friend or discussing a problem with them. This example highlights the fundamental challenge of *(syntactic) ambiguity* and how it can lead to parsing errors affecting sentence meaning (Church and Patil, 1982; Xin et al., 2021). Hence, a parser's ability to resolve such ambiguities is crucial (Mitchell and Gaizauskas, 2004).

During the evaluation, I found that a single span can induce errors across multiple categories. The Berkeley parse *(NP (DT The) (JJ old) (NN car))* in S2, for example, was classified as an *NP Int. Struct.* as well as a *1-Word+Unary* error, thereby complicating detailed assessments. While this analysis thus offers initial insights into the parsers' qualitative strengths and weaknesses, it remains unclear what specific linguistic errors they make and how these can be tied back to the results in Table 1.

3.2.3 Detailed Error Analysis

This section builds on the qualitative analysis motivated by Kummerfeld et al. (2012) and provides a detailed linguistic assessment of the error behaviour of both parsers based on the gold standard.

[7] https://github.com/jkkummerfeld/berkeley-parser-analyser



(a) Excerpt of the gold standard tree for S1

(b) Excerpt of the Charniak & Berkeley parse trees for S1

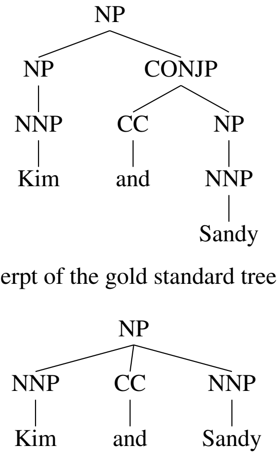
Figure 4: Excerpt of parse trees for S1. Although inaccurate, both parsers generate a clearer, easier-to-interpret tree structure than the gold standard, where the possessive ending "'s" clearly belongs to its corresponding noun "aunt". Unlike the gold parse, they assign equal syntactic significance to "My aunt's" and "can opener".

Complex and Deeply Nested Structures

Both parsers fail to replicate deeply nested structures common in the gold standard, resulting in the *NP Int. Struct.* errors captured in Table 1. Figure 4 illustrates, using S1 as an example, how both parsers simplify and compress parse structures by increasing the width (i.e., adding more child nodes) and reducing the depth of a tree. Especially the Berkeley parser simplifies trees, while the Charniak parser creates parses of medium complexity. Contrasting the depth of the different parse trees with averages of 11 (Gold), 9 (Charniak), and 8 (Berkeley) levels per parse tree (see Table 3, Appendix B), as well as the already presented Berkeley parser's higher average number of crossing brackets, corroborates this insight. Despite often diverging from the gold standard, these parses simplify complexity while preserving meaning, better reflecting human syntactic intuition. Research has shown that this kind of syntactic simplification yields desirable properties in many NLP applications (Chandrasekar et al., 1996; Siddharthan, 2006)

Handling Conjunctions and Coordination

Similarly, both parsers deviate from the gold standard when dealing with conjunctions and coordination. Here, as shown in Figure 5, both parsers overlook nuances in coordinated structures, which also contributes to the *NP Int. Struct.* error count in Table 1. Nevertheless, the representations of the parsers are, again, linguistically more intuitive, as they do not seem to favour any subject in terms of syntactical importance. The verb phrase (VP) "broke in and stole my TV" in S3, where, in contrast to the gold standard, both parsers keep the VPs "broke in" and "stole my TV" on one level, is another example of this behaviour.



(a) Excerpt of the gold standard tree for S4

(b) Excerpt of the Charniak & Berkeley parse trees for S4

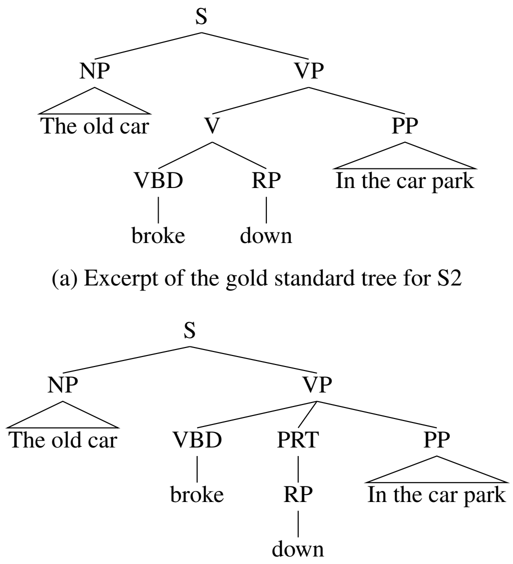
Figure 5: Excerpt of parse trees for S4. While the gold standard tree is more detailed, the Charniak & Berkeley parsers simplify the representation and place "Kim" and "Sandy" on the same hierarchical level.

Verb Phrases and Particle Recognition

In contrast to the gold standard, the Charniak and Berkeley parsers do not form a dedicated *V* structure when a verb phrase is composed of a verb (e.g., *VBD*) and a particle (*RP*). Instead, they assign both components directly to the parent VP. The example in Figure 6 further demonstrates that the parsers frequently employ bracket labels such as the phrase level tag *PRT* (Particle) in different instances than the gold standard, leading to mismatches and hence the significant number of *1-Word+Unary* errors shown in Table 1. The different usage of bracket labels and function tags, which are used to more accurately describe situations where words/phrases are used for other functions than their syntactic label alone can define (Bies et al., 1995), also leads to *Mod. Attach.* and *PP Attach.* errors.

Different Labels and Tagging

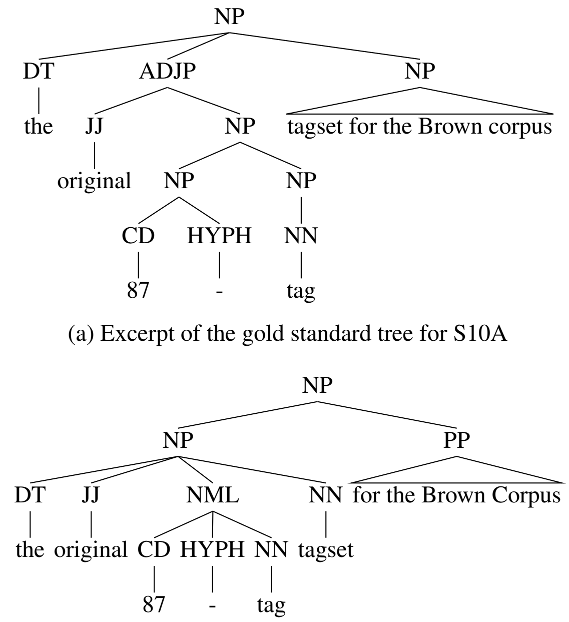
The analysis so far indicates comparable parsing performance between the two parsers. Yet, certain



(a) Excerpt of the gold standard tree for S2

(b) Excerpt of the Charniak & Berkeley parse trees for S2

Figure 6: Excerpt of parse trees for S2. The parsers omit a dedicated *V* structure, placing both verb and particle directly under the parent VP. This example also symbolises the frequent mismatch of unary productions and SWPs due to bracket labels or function tags.



(a) Excerpt of the gold standard tree for S10A

(b) Excerpt of the Berkeley parse tree for S10A

Figure 7: Excerpt of parse trees for S10A. Although inaccurate, the Berkeley parser exhibits a superior grasp of linguistic nuances, such as the assessment of syntactic relevance and association of phrases to syntactic groups.

linguistic nuances reveal that the Berkeley parser exhibits a higher level of linguistic understanding than its counterpart. This becomes apparent, for example, through evaluating the parser's tagging consistency beyond the word level (i.e., assigning labels to phrases). Although the parser often deviates from the gold parse in this category as well, causing the *Diff. Label* errors in Table 1, its labelling behaviour leads to more precise parses from a linguistic standpoint. Figure 7 exemplary uses S10A to show that the Berkeley parser is closest to breaking down the presented part of the sentence according to the actual syntactic meaning of its parts. It accurately employs the *NML* tag to identify "87-tag" in its role as a nominal modifier (Warner et al., 2004). Differing from the gold parse, it consolidates "the original 87-tag tagset" within a single NP level, reflecting the syntactic meaning of these elements. Labelling "for the Brown Corpus" as *PP* and excluding "tagset" from this phrase is also accurate. In contrast, the Charniak parser, while nearly identical in structure for S10A, less precisely treats "87-tag" as a singular noun (*NN*).

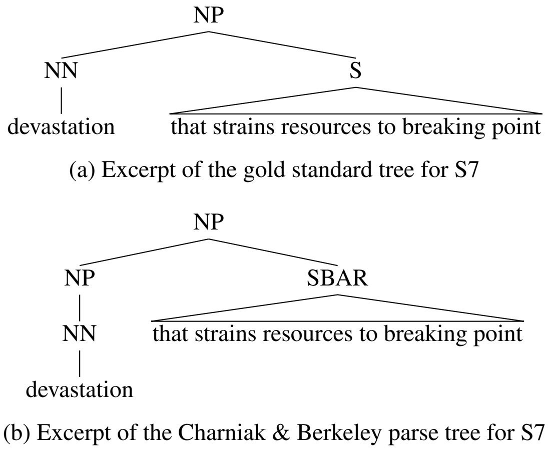
The widely different construction of phrase structures shown here (which can also be found in other sentences (e.g., S6, S9)) can lead to significant differences in interpretation and impact on potential downstream tasks. The example also highlights the importance of challenging the gold standard and base parser selection on the specific task at hand.

4 Critique of the Gold Standard

As indicated in previous sections, the gold standard used for evaluating the Charniak and Berkeley parsers has shown limitations in linguistic accuracy. This can be attributed to its crowdsourced nature, leading to subjective interpretations by different human annotators and inherent ambiguities and errors. Based on the analyses carried out, the gold standard can and should be challenged on two points in particular: **inaccuracy in tagging/labelling** and its **highly complex phrase structures**. Both of these points of criticism are exemplified in the following.

4.1 Inaccuracy in Tagging/Labelling

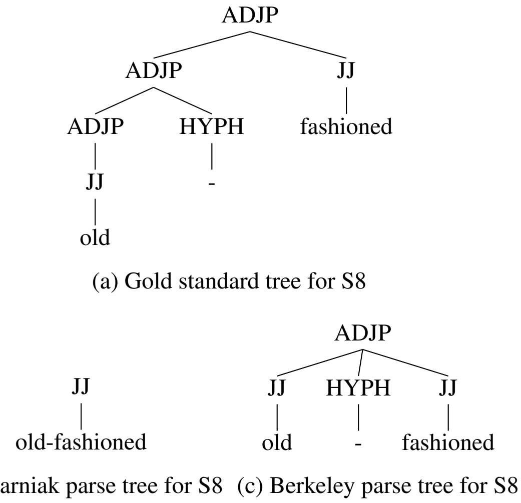
Besides the previously mentioned labelling differences, it is noteworthy that both parsers correctly make use of the clause type *SBAR* (e.g., S5, S6, S7, S8) for relative and subordinate clauses (Bies et al., 1995). Figure 8 illustrates how *SBAR* underscores that a clause like "that strains resources to breaking point" does not constitute a complete sentence but modifies a main clause. Given this context, the gold standard for these phrases could be modified to define them more accurately in linguistic terms.


NP
NN
S
devastation
that strains resources to breaking point
NP
NP
SBAR
NN
that strains resources to breaking point
devastation

(a) Excerpt of the gold standard tree for S7

(b) Excerpt of the Charniak & Berkeley parse tree for S7

Figure 8: Excerpt of parse trees for S7. Both parsers employ the *SBAR* clause type to accentuate cases where clauses are subordinates. The gold standard lacks this added level of linguistic precision and marks such phrases in the same way as complete sentences (*S*).



(a) Gold standard tree for S8

(b) Charniak parse tree for S8 (c) Berkeley parse tree for S8

Figure 9: Excerpt of parse trees for S8. The gold standard exhibits a complex nested *ADJP* structure, while the Charniak parser (without manual adjustment) treats the phrase as a single *JJ*. The Berkeley parser maintains the separation of "old" and "fashioned" without nesting.

Incorrect labelling can also be identified in S3, S5 and S7, for example. In S3, both parsers correctly define "At least two" as a quantifier phrase (*QP*), while the gold standard labels it as an adverb phrase (*ADVP*). The gold standard's "as well as", labelled *ADVP*, in S5, should be a *CONJP*, since both phrases it connects are of equal syntactic importance (Bies et al., 1995). In S10B, the Berkeley parser is the only one that correctly identifies "e.g." as a foreign word (*FW*), which is consistent with the official tagging guidelines by Santorini (1990).

4.2 Highly Complex Phrase Structures

As outlined in Section 3.2.3, the gold standard stands out particularly for its highly complex and nested phrase structures. In S9, for example, the gold standard tree is nearly twice as deep as the trees generated by both parsers (see Table 3, Appendix B). It is generally noticeable that most gold standard trees, with a few exceptions, follow a binary structure close to the Chomsky Normal Form (Chomsky, 1959), leading to the narrow, deep structure of the trees. Although this does not necessarily imply that the gold standard trees are inferior or inaccurate, several examples affirm that the trees generated by the Charniak and Berkeley parsers are often linguistically more precise.

Similar to the phrase structure shown in Figure 4, for example, "my aunt's car" in the gold standard S6 is also deeply nested and incorrectly assigns the *POS* element "'s" to "car", instead of "aunt", whose possessive ending it is (Santorini, 1990).

Another example is the representation of the phrase "old-fashioned" in S8. As shown in Figure 9, the gold standard counter-intuitively splits this compound adjective across three levels. The Berkeley parser strikes a good balance, as it splits the adjective and thus preserves linguistic details while keeping it on one level and forming an adjective phrase (*ADJP*), in line with Bies et al. (1995).

5 Conclusion

Single quantitative metrics, as often used in the evaluation of natural language parsers, provide only limited meaningful linguistic information about a parser's strengths and weaknesses. Following a systematic approach, this study has shown that quantitative analysis can only be a precursor to a more detailed qualitative analysis to determine significant error patterns of parsers as well as their potential impact on downstream tasks. Furthermore, the assumptions underlying an assessment, the gold standard, should always be questioned.

Based on this insight, the performance of two constituency parsers, the Charniak-Johnson parser and the Berkeley Neural parser, were evaluated against gold parses on 11 selected sentences of varying complexity. Despite minimal errors, the Berkeley parser overall emerged as more proficient in this analysis, displaying enhanced linguistic comprehension and precise syntactic representation. Despite Berkeley's superiority in this study, the optimal parser for practical NLP applications depends on the specific requirements of the task.

Limitations

This study evaluated the Charniak-Johnson parser and Berkeley Neural parser based on a very limited set of only 11 selected sentences. Although these sentences exhibit different levels of complexity and intricacies challenging for parsers, a more in-depth analysis should consider larger data sets that cover more nuances of natural language.

Furthermore, both parsers were tested in their default configurations. Different settings and hyperparameters could be explored to see how they might affect the performance of the parsers.

References

Xuefeng Bai, Jialong Wu, Yulong Chen, Zhongqing Wang, and Yue Zhang. 2023. Constituency Parsing using LLMs. ArXiv:2310.19462 [cs].

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing Guidelines for Treebank II Style Penn Treebank Project 1.

Ezra Black, Steven Abney, Dan Flickenger, Claudia Gdaniec, Ralph Grishman, Philip Harrison, Donald Hindle, Robert Ingria, Frederick Jelinek, Judith Klavans, Mark Liberman, Mitchell Marcus, Salim Roukos, Beatrice Santorini, and Tomek Strzalkowski. 1991. A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.

Cambridge University Press. 2024. As well as. Publisher: Cambridge Dictionary.

Raman Chandrasekar, Christine Doran, and Srinivas Bangalore. 1996. Motivations and Methods for Text Simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, AAAI'97/IAAI'97, pages 598–603, Providence, Rhode Island. AAAI Press.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan. Association for Computational Linguistics.

Alebachew Chiche and Betselot Yitagesu. 2022. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9(1):10.

Noam Chomsky. 1959. On certain formal properties of grammars. *Information and Control*, 2(2):137–167.

Kenneth Church and Ramesh Patil. 1982. Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table. Technical report, Massachusetts Institute of Technology, Laboratory for Computer Science.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98/EACL '98, pages 16–23, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805 [cs].

Daniel Fried, Nikita Kitaev, and Dan Klein. 2019. Cross-Domain Generalization of Neural Constituency Parsers. ArXiv:1907.04347 [cs].

David Gaddy, Mitchell Stern, and Dan Klein. 2018. What's Going On in Neural Constituency Parsers? An Analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 999–1010, New Orleans, Louisiana. Association for Computational Linguistics.

Sardar Jaf and Calum Calder. 2019. Deep Learning for Natural Language Parsing. *IEEE Access*, 7:131363–131373. Conference Name: IEEE Access.

Tadao Kasami. 1965. An efficient recognition and syntax analysis algorithm for context-free languages. Technical Report AFCRL-65-758, Air Force Cambridge Research Laboratory.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual Constituency Parsing with Self-Attention and Pre-Training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 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)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan. Association for Computational Linguistics.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 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*, pages 1048–1059, Jeju Island, Korea. Association for Computational Linguistics.

Brian Mitchell and Robert Gaizauskas. 2004. A Labelled Corpus for Prepositional Phrase Attachment. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. ArXiv:1802.05365 [cs].

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, and Mark Johnson. 2001. Parsing the wall street journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 271, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Beatrice Santorini. 1990. Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision). Technical Report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania, Philadelphia.

Advaith Siddharthan. 2006. Syntactic Simplification and Text Cohesion. *Research on Language and Computation*, 4(1):77–109.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A Minimal Span-Based Neural Constituency Parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. ArXiv:1706.03762 [cs] version: 1.

Colin Warner, Ann Bies, Christine Brisson, and Justin Mott. 2004. Addendum to the Penn Treebank II Style Bracketing Guidelines: BioMedical Treebank Annotation. Technical report, University of Pennsylvania Linguistic Data Consortium, 3600 Market Street, Suite 810 Philadelphia, PA 19104, USA.

Yida Xin, Henry Lieberman, and Peter Chin. 2021. PATCHCOMM: Using Commonsense Knowledge to Guide Syntactic Parsers. *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, 18(1):712–716. Conference Name: Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning.

Sen Yang, Leyang Cui, Ruoxi Ning, Di Wu, and Yue Zhang. 2022. Challenges to Open-Domain Constituency Parsing. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 112–127, Dublin, Ireland. Association for Computational Linguistics.

Daniel H. Younger. 1967. Recognition and parsing of context-free languages in time n^(3*). *Information and Control*, 10:189–208.

A Data Basis: 11 Selected Sentences

Sent.	Text
S1	My aunt's can opener can open a drum.
S2	The old car broke down in the car park.
S3	At least two men broke in and stole my TV.
S4	Kim and Sandy both broke up with their partners.
S5	The horse as well as the rabbits which we wanted to eat has escaped.
S6	It was my aunt's car which we sold at auction last year in February.
S7	Natural disasters – storms, flooding, hurricanes – occur infrequently but cause devastation that strains resources to breaking point.
S8	Letters delivered on time by old-fashioned means are increasingly rare, so it is as well that that is not the only option available.
S9	English also has many words of more or less unique function, including interjections (oh, ah), negatives (no, not), politeness markers (please, thank you), and the existential 'there' (there are horses but not unicorns) among others.
10A	The Penn Treebank tagset was culled from the original 87-tag tagset for the Brown Corpus.
10B	For example, the original Brown and C5 tagsets include a separate tag for each of the different forms of the verbs do (e.g., C5 tag VDD for did and VDG tag for doing), be and have.

Table 2: Overview of the 11 selected sentences utilised for parser evaluation.

B Parse Tree Depth

Sent.	Gold	Charniak	Berkeley
1	6	5	5
2	6	5	5
3	6	5	5
4	6	5	5
5	9	10	10
6	11	9	9
7	5	4	4
8	13	9	9
9	22	12	12
10A	10	8	8
10B	23	22	15
Average	10.6	8.5	7.9

Table 3: Comparison of maximum depths of parse trees (levels with non-terminals)

Sent.	Sent. Len.	Recall	Prec.	Matched Brackets	Gold Brackets	Test Brackets	Crossing Brackets	Correct Words	Correct Tags	Tag Acc.
1	10	62.50	83.33	5	8	6	1	9	8	88.89
2	10	55.56	83.33	5	9	6	0	9	9	100.00
3	11	60.00	75.00	6	10	8	0	10	8	80.00
4	10	45.45	83.33	5	11	6	0	9	8	88.89
5	15	66.67	62.50	10	15	16	0	14	13	92.86
6	16	63.16	75.00	12	19	16	1	15	14	93.33
7	21	57.69	68.18	15	26	22	1	18	15	83.33
8	27	58.82	90.91	20	34	22	0	24	22	91.67
9	53	30.36	48.57	17	56	35	8	43	42	97.67
10A	18	38.89	77.78	7	18	9	2	16	16	100.00
10B	40	33.33	36.36	12	36	33	11	38	31	81.58
Sum/Average:		47.11	63.69	114	242	179	24	205	186	90.73

Table 4: Charniak parser EVALB evaluation results

Sent.	Sent. Len.	Recall	Prec.	Matched Brackets	Gold Brackets	Test Brackets	Crossing Brackets	Correct Words	Correct Tags	Tag Acc.
1	10	62.50	83.33	5	8	6	1	9	9	100.00
2	10	55.56	83.33	5	9	6	0	9	9	100.00
3	11	60.00	66.67	6	10	9	1	10	9	90.00
4	10	45.45	83.33	5	11	6	0	9	8	88.89
5	15	66.67	62.50	10	15	16	0	14	13	92.86
6	16	63.16	75.00	12	19	16	1	15	14	93.33
7	21	50.00	76.47	13	26	17	0	18	17	94.44
8	27	61.76	87.50	21	34	24	0	24	23	95.83
9	53	35.71	60.61	20	56	33	6	43	38	88.37
10A	18	38.89	63.64	7	18	11	3	16	16	100.00
10B	40	25.00	27.27	9	36	33	16	38	32	84.21
Sum/Average:		46.69	63.84	113	242	177	28	205	188	91.71

Table 5: Berkeley parser EVALB evaluation results

C EVALB Results

Table 4 and Table 5 show the results of the EVALB-based quantitative evaluation of the Charniak-Johnson and Berkeley Neural parser, respectively. Except for Precision and POS tagging accuracy, the Charniak parser is, on average, marginally superior to the Berkeley parser.