Unified Syntactic Annotation of English in the CGEL Framework

Brett Reynolds
Humber College
brett.reynolds@humber.ca

Aryaman Arora
Georgetown University
aa2190@georgetown.edu

Nathan Schneider
Georgetown University
nathan.schneider@georgetown.edu

Abstract

We investigate whether the *Cambridge Grammar of the English Language* (2002) and its extensive descriptions work well as a corpus annotation scheme. We develop annotation guidelines and in the process outline some interesting linguistic uncertainties that we had to resolve. To test the applicability of CGEL to real-world corpora, we conduct an interannotator study on sentences from the English Web Treebank, showing that consistent annotation of even complex syntactic phenomena like gapping using the CGEL formalism is feasible. Why introduce yet another formalism for English syntax? We argue that CGEL is attractive due to its exhaustive analysis of English syntactic phenomena, its labeling of both constituents and functions, and its accessibility. We look towards expanding CGELBank and augmenting it with automatic conversions from existing treebanks in the future.

1 Introduction

Ask a linguist about a detail of English grammar, and chances are they will reach for the *Cambridge Grammar of the English Language* (CGEL; Huddleston and Pullum, 2002). The product of the labors of two editors and 13 other chapter authors over more than a decade, CGEL is the most recent comprehensive reference grammar of English, describing nearly every syntactic facet of present-day Standard English in its 1700+ pages (Culicover, 2004). As but one example, a section[1] is devoted to the form and function of sentences like the first sentence of this paragraph, where the part before *and* is grammatically imperative but interpreted as a condition, and the part after *and* is interpreted as a consequence. CGEL is a gold mine for such idiosyncrasies that a sharp-eyed English student (or linguist, or treebanker) might want to look up, in bottom-up fashion. It is also a systematic top-down survey of the building blocks of the language—in this respect, aided by a lucid companion textbook (SIEG2; Huddleston et al., 2021).

In a review for *Computational Linguistics*, Brew (2003) argued that CGEL is a descriptive reference that echos precise formal thinking about grammatical structures; and as such, it holds considerable relevance for computational linguistics, supplementing formal grammars and treebanks like the venerable Penn Treebank (PTB; Marcus et al., 1994). Brew exhorts: “it should become a routine part of the training of future grammar writers and treebank annotators that they absorb as much as is feasible of this grammar”. It has certainly had an impact, for example, on the Universal Dependencies project (UD; Nivre et al., 2016, 2020; de Marneffe et al., 2021), whose annotation guidelines cite CGEL many times in discussing particular phenomena[2] (though the UD trees themselves, for reasons of lexicalism and panlingualism, diverge significantly from the representations given in CGEL).

We ask: **What would it take to develop an annotation scheme based on CGEL?** If CGEL’s attention to terminological precision and rigor is as strong as Brew suggests, it should not be nearly as difficult as mounting an effort of a completely new annotation framework. Most substantive questions of grammatical analysis should be addressed by CGEL, leaving only minor points to flesh out for treebanking. On the other hand, because CGEL was not designed for annotation, and therefore not tested systematically on corpora, perhaps it has substantial holes, regularly missing constructions that occur in real data.

To answer this question, we bootstrap an annotation manual and small set of sentences based on the

[1] “Imperatives interpreted as conditionals” (pp. 937–939)

[2] References to CGEL can be found, for example, at https://universaldependencies.org/u/overview/complex-syntax.html (regarding content clauses and secondary predicates) and https://universaldependencies.org/u/overview/specific-syntax.html (regarding comparative constructions).

descriptions from CGEL. We examine what blanks in the CGEL specifications need to be filled in to realize full-sentence trees in our data—both qualitatively (through working sentence-by-sentence) and quantitatively (by conducting an interannotator agreement study).

What practical benefits hinge on the answer to this question? We are cognizant that substantial English treebanks already exist—constituency treebanks following the Penn Treebank standard, dependency treebanks, and others (§2). Thus, we do not anticipate a significant amount of from-scratch annotation in the CGEL framework. Yet we see practical benefits of the CGEL style of description, perhaps induced automatically as a new "view" of gold PTB trees. First, **exhaustiveness**: CGEL trees systematize *both constituent categories and functions in a unified framework*, whereas mainstream approaches for English prioritize either constituent structure (like PTB) or dependency structure (like UD). And second, **accessibility**: the trees would be consistent with human-readable descriptions and linguistic argumentation in the CGEL and SIEG2 texts, allowing users of a treebank (or parser) to look up the constructions in question.[3]

Through developing guidelines and annotating data, we find that CGEL offers a powerful foundation for treebanking, though there are points where further specification is needed. Our small but growing treebank—which we call **CGELBank**—and accompanying code for validation and measuring interannotator agreement are available at `https://github.com/nert-nlp/cgel/`. We also publish our annotation manual, which stands at about 75 pages (mostly of example trees): Reynolds et al. (2023).

2 Related Work

Even considering just English, there have been many formalisms deployed for syntactic annotation. A sample is given in Table 1. Each formalism makes different theoretical claims (e.g., is deep structure distinct from surface structure?) which bring computational tradeoffs (e.g. complexity vs. parsing efficiency). Many, beginning with PTB, have been used to annotate the Wall Street Journal corpus (WSJ; Marcus et al., 1993). CGEL shares ideas with many treebanks, such as constituency structure (PTB, TAG, etc.), labelled dependency relations (SD, UD, etc.), gapping (PTB), among other features.

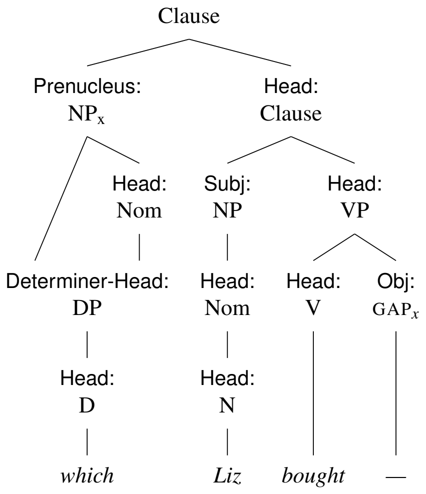

Figure 1: CGEL-style tree for the interrogative clause in *I wonder which Liz bought*.

A corpus that likewise integrates constituent categories and functions in a single tree is the TIGER treebank for German (Brants et al., 2004).

3 The CGEL Framework

An example parse in the CGEL framework appears in Figure 1.[4] Its building blocks are constituents, each of which receives a *category* indicating the type of unit it is and a *function* (notated with a colon) indicating its grammatical relation within the higher constituent. The constituent structure is a hierarchical bracketing of the sentence, which is projective with respect to the order of words in the sentence. Terminals consist of lexical items (omitting punctuation) as well as *gaps* used to handle constructions with noncanonical word order.

Categories. CGEL posits a distributionally-defined set of **lexical** categories, on which basis we developed a part-of-speech tagset with 11 tags: N (noun), N$_{\text{pro}}$ (pronoun), V (verb), V$_{\text{aux}}$ (auxiliary verb), P (preposition), D (determinative),[5] Adj (adjective), Adv (adverb), Sdr (subordinator), Coordinator, and Int (interjection). (See

[3] PTB has an extensive annotation manual (Bies et al., 1995), but that serves a different purpose from a reference grammar: an annotation manual is a set of policies for an expert reader, not a complete presentation of syntactic phenomena or a defense of design decisions. Moreover, the terminology in the PTB manual draws heavily from particular syntactic theories like Government and Binding, whereas CGEL employs more general descriptive terminology.

[4] See Appendix C for more examples and a comparison with PTB.

[5] In CGEL, *determinative* is a lexical category whereas *determiner* is a function within an NP. A determinative heads

Framework		Representative Citations
Constituency		
PTB	Penn Treebank	(Marcus et al., 1994; Bies et al., 2012; Pradhan et al., 2013)
TAG	Tree-Adjoining Grammar	(Chen and Vijay-Shanker, 2000)
MG	Minimalist Grammars	(Torr, 2018)
RRG	Role and Reference Grammar	(Bladier et al., 2018)
Dependency		
SD	Stanford Dependencies	(de Marneffe et al., 2006)
UD	Universal Dependencies	(Nivre et al., 2016)
SUD	Surface Universal Dependencies	(Gerdes et al., 2018)
FGD	Functional Generative Description	(Čmejrek et al., 2005)
Constraint-Based		
LFG	Lexical-Functional Grammar	(Sulger et al., 2013)
HPSG	Head-Driven Phrase Structure Grammar	(Oepen et al., 2002; Miyao et al., 2004; Flickinger et al., 2012)
Categorial		
CCG	Combinatory Categorial Grammar	(Hockenmaier and Steedman, 2007)

Table 1: A sample of grammatical frameworks that have been applied to English corpora.

(Reynolds et al., 2022) for further details and comparison to PTB/UD tagsets, especially regarding P and D.) Pronouns and proper nouns are a subset of nouns, though we have created a distinct tag for pronouns; auxiliary verbs are a subset of verbs. All of these categories except subordinator and coordinator project higher-level **phrasal** constituents, e.g. N ← Nom (nominal) ← NP (noun phrase). The basic phrasal categories are: Nom, NP, VP, Clause (the various subtypes of which are unmarked here except $\text{Clause}_{\text{rel}}$ for relative clauses), PP, DP, AdjP, AdvP, and IntP. Phrases are typically binary- or unary-branching, but *n*-ary branches are also possible. There is also a non-phrasal constituent category: Coordination, which may have ternary branching or higher.

Functions. Each constituent has a function indicating its syntactic role in the higher constituent. A *phrasal* constituent is headed, i.e. it has exactly one child in Head function along with zero or more dependents. Coordination constituents are the main exception: there is no head, and each element (conjunct) in the coordination receives a function of Coordinate. Figure 2 illustrates the main CGEL functions, organized into a hierarchy. Note that CGEL contrasts adjuncts (Mod, Supplement) with complements (Comp and subtypes, including Subj, Obj, PredComp, and others). Other dependent functions include Determiner (Det) function in an NP, and Marker for grammatical words that mark but do not head a phrase, notably coordinators and subordinators.

Gaps. CGEL employs gap constituents and coindexation, as in Figure 1, to handle unbounded dependency constructions (UDCs) and other constructions that deviate from the canonical declarative order, showing where there is a clear structural gap. Nevertheless "the account is quite informal" (R. Huddleston, personal communication). To make it more formal, we have restricted the use of gaps to UDCs, subject-auxiliary inversion (SAI), and pre- and post-posing of complements. We also use gaps for adjunct fronting when it triggers SAI (e.g., *Only once had I – seen it –*). Subject–dependent inversion (SDI) is a double-gapped construction with a subject gap and a complement gap in the VP (e.g., *Here – is – Jim*). All subject-relatives have a gap, as do delayed right constituent coordination and end-attachment coordination. Coindexation is used with and only with a gap. Every gap must be coindexed with exactly one overt constituent (and possibly other gaps). There are no gaps for ellipsis.

Fusion. Certain constructions are analyzed with *fusion of functions*, in which a constituent participates in two different higher constituents (Payne et al., 2007; Pullum and Rogers, 2008). This is shown in Figure 1 for the NP *which*, short for something like "which items": the DP is taken to fulfill both the Determiner function in the NP and the Head in its Nominal. Other constructions where CGEL employs fusion of functions include compound determinatives (e.g. *someone*), other determinatives or adjectives as NP heads (*the* ***rich***, *the* ***tallest***, *those* ***three***[6]), and fused (a.k.a. free or head-

a determinative phrase (DP), not to be confused with the notion of a *determiner phrase* in generative grammar (Abney, 1987).

[6] Elazar and Goldberg (2019) offer an NLP approach to reasoning about numeric fused heads.

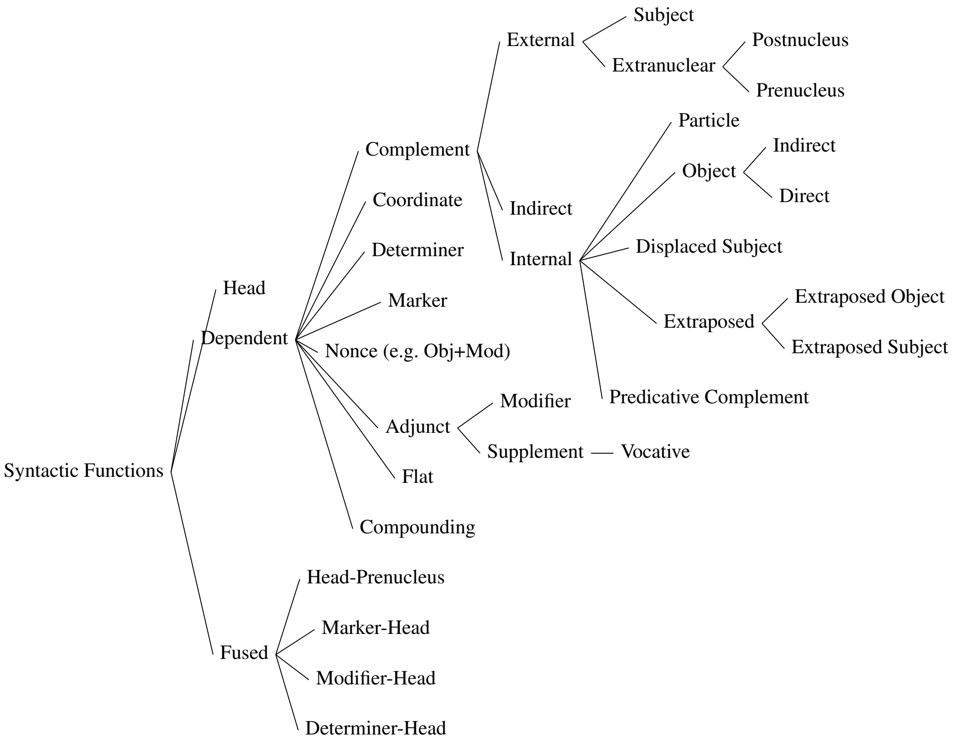

Figure 2: Hierarchy of functions. The ones annotated directly in the data are the leaves plus Complement (Comp), Object (Obj), and Supplement. The distinction between direct and indirect objects is made only in double object constructions.

less) relative constructions (***whatever** you want*). The hyphenated notation such as Determiner-Head indicates its dual function. Thus, technically the parse is a graph rather than a tree. However, the longer of the two incoming edges can be inferred deterministically based on the Determiner-Head label and the rest of the structure. For computational purposes, then, we can omit the longer edge wherever there is fusion of functions, maintaining the tree property, and automatically add it in postprocessing for visualization. We therefore refer to CGEL-style parses as trees.

4 Towards CGELBank

Despite its detail and richness, in 1700+ pages, CGEL includes just 40 trees, and on some points is inexplicit. Annotating naturally occurring sentences (§5) brought many of these ambiguities to the fore. Here we identify questions we faced and the decisions we made.

4.1 Categorizing individual lexemes

Creating part-of-speech (POS) tagsets and defining tag boundaries have been contentious in treebanking (Atwell, 2008). CGEL's guidance in this area is extensive but dispersed, and lists of closed-category items are inexhaustive. For CGELBank, we compiled mentions of lexemes and their categories from CGEL and applied CGEL principles to classify numerous unmentioned lexemes.[7] Examples include the determinative *said* (e.g., *as in said contract*), the coordinator *slash* (e.g., *Dear God slash Allah slash Buddha slash Zeus*), and the preposition *o'clock* (Pullum and Reynolds, 2013).

4.2 Simplifying and un-simplifying

CGEL uses various subtypes of head within clause structure (Nucleus, Predicate, Predicator); we collapse these to Head. CGEL sometimes removes intermediate unary nodes, such as eliminating Head:Nom between Head:N and its projected NP. We consistently include these nodes.

4.3 Gaps

CGEL posits gaps in tree structures for prenucleus position constituents, but is inconsistent in indicating them. We explicitly indicate a gap in most cases and outline our decisions for unclear cases.

Subject gaps. For open interrogatives such as (1a) and (1b), CGEL's position is unclear. Given

[7] Conducted since 2006 in consultation with Huddleston and Pullum, recorded in Simple English Wiktionary.

ambiguity, we follow the standard position that a gap exists (e.g. Maling, 2000; Bies et al., 1995) in questioned or relativized subject clauses.

(1) a. What did she tell you?
b. Who told you that?

Adjunct gaps. Adjuncts may appear in various locations, with some not appearing clause finally. We decided against including a gap, except in relative and open interrogative clauses where CGEL marks a gap.

Phrasal genitives. In NPs ending in a gap, we attach *'s* to the gap, as in *a guy I know __'s house*.

Coordination and comparatives. CGEL's Gapped Coordination refers to ellipsis, not gaps. CGELBank does not include gaps in tree structure for coordination and comparatives.

4.4 Branching & tree structure

In CGEL, some rare phenomena are not explicitly depicted in tree form due to the limited number of actual syntax trees in the text. Also, unary nodes (e.g. N → Nom → NP) are inconsistently indicated due to space considerations. In general, we sought to ensure that tree structure was consistent and thus had to make some decisions on how to treat phenomena such as coordination, complementation, etc.

Lexical Projection Principle. Outside of morphologically derived expressions, and excepting coordinators and subordinators, **a lexical node almost always projects a phrase of the corresponding category**. Thus, every N must serve as head within a Nom; every V must head a VP; every Adj must head an AdjP; and so forth. The one exception is that subject-auxiliary inversion targets auxiliaries specifically (rather than the VP they would project in normal position), so if the constituent in Prenucleus function consists of a single unmodified V_{aux}, it will not project a VP there.

Coordinates & markers. A coordi**nation** is a non-headed construction with coordi**nates** as children (CGEL p. 1278). Therefore, coordi**nates** in coordinations are neither heads nor dependents. Consider, though, the following coordination *the guests and indeed his family too* (p. 1278), reproduced here as Figure 3.

Unlike coordinations, NPs like *and indeed his family too* are headed constructions in CGEL. The NP has two modifiers: *indeed* and *too*, which, like all modifiers, are dependents requiring a head sibling. But if the *his family* is not a head but a coordinate as labeled, then this NP is headless, an internal contradiction in CGEL.

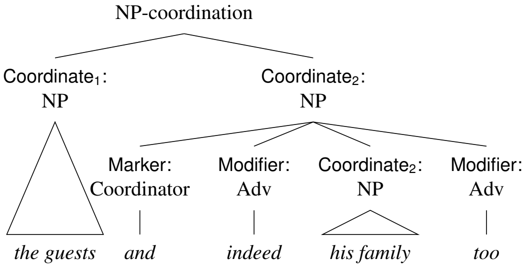

Figure 3: CGEL flat coordination—rejected in CGELBank, where *indeed his family too* is an NP serving as the Head of the second coordinate.

Markers[8] are siblings of heads when they are subordinators (see (9) on p. 954 and (51) on p. 1187), so a marker is a dependent. This, however, is incompatible with the analysis in Figure 3 (p. 1277).

To resolve these inconsistencies, the NP *his family* in Figure 3 must be a head and not a coordinate. We generalize from this to the principle that, contra Figure 3, a coordinate is never the child of a non-coordination, and a marker is always a dependent with a sibling head.

Indirect complements. Indirect complements, such as in *enough time to complete the work*, are licensed by a dependent in the phrase. We construct a superordinate phrase of the head type and branch the indirect complement from that. When the complement is further delayed, we do the same for nearest possible parent phrase.

Verbless clauses. CGEL's treatment of verbless clauses (VlCs) is incompatible with its general treatment of clauses. VlCs have no verb and no VP, so they must not be clauses in the syntactic sense that CGEL implies. We treat certain PPs as having two complements, analogous to complex transitive verbs, and PPs like *while happy* as taking predicative complements. Supplement VlCs are analyzed as headless nonce constructions.

Names. CGEL claims that the syntactic structure of proper names mostly conforms to the rules for ordinary NPs, but it also notes that there is no convincing evidence for treating one element as head in personal names. We treat proper names, along

[8] Though CGEL uses "marker" both non-technically (e.g., marker of distinctively informal style), and technically as a function term, we discuss only the latter.

Split	Trees	Tokens	Nodes	Ann.
EWT	100	1,864	5,110	2
Twitter	65	824	2,316	2
EWT-trial	27	500	1,365	1
Twitter-trial	10	257	727	1
Pilot	5	61	174	1 + 2
IAA	50	642	1,747	1 + 2
Total	257	4,148	11,439	

Table 2: Overall statistics about the treebank and its splits. *Nodes* is the sum of the count of all constituents and gaps in each tree, including tokens. *Ann.* indicates the annotators involved.

with chemical compounds, as single lexical items, analyzing multiple tokens using the Flat relation.

5 Annotation Process

What began as a pet project to make CGEL-style trees for interesting sentences found in the wild eventually became a corpus-building effort, with two linguists interested in the CGEL framework (the first and third authors) serving as annotators. To date, this has resulted in over 200 trees of naturally occurring sentences—some handpicked, others sampled at random from a corpus. Statistics for **CGELBank** are presented in Tables 2 and 3.

CGELBank trees were drawn from multiple sources, and were annotated in four phases.

1. **Twitter**: Exploratory annotation of real-world sentences taken from Twitter by Annotator 2 resulted in this set of 65 trees. At this point, there were no agreed-upon guidelines for CGEL annotation and the project was largely informal.
2. **EWT**: A set of 100 sentences sampled from the English Web Treebank (Bies et al., 2012) was annotated by Annotator 2 and simultaneously the guidelines were composed in discussion with Annotator 1. To maintain consistency with the guidelines, both the EWT and Twitter treebanks were validated and iteratively corrected.
3. **EWT-trial, Twitter-trial**: Once the guidelines and validation script were mostly complete, and as the browser-based annotation workflow was under development, the two annotators used it to make 37 more trees (27 from additional EWT sentences, 10 from Twitter and other sources). These trees were singly annotated and validated but not adjudicated.
4. **IAA** and **Pilot**: For an interannotator study, both annotators independently annotated and then adjudicated a pilot set of 5 trees and then a larger set of 50 trees. These were also drawn from EWT.

```
# sent_id = which-liz-bought
# text = which Liz bought.
# sent = which Liz bought --
(Clause
   :Prenucleus (x / NP
      :Head (Nom
         :Det-Head (DP
            :Head (D :t "which"))))
   :Head (Clause
      :Subj (NP
         :Head (Nom
            :Head (N :t "Liz")))
      :Head (VP
         :Head (V :t "brought" :l "bring" :p ".")
         :Obj (x / GAP))))
```

Figure 4: Illustration of the `.cgel` data format for the clause from Figure 1. Note that the bracketed notation forms a proper tree: the reentrancy of the fused determiner-head is automatically added post hoc. The verb lemma is included as it differs from its inflected form. Features on nodes are extensible: for example, CGELBank uses `:p` for punctuation, `:note` to offer commentary on a construction (with CGEL page references), and `:correct` to indicate corrections to typos. Finer-grained morphosyntactic information (inflectional features, clause types, etc.) may be added in the future.

The initial 165 Twitter and EWT sentences were annotated in LaTeX using the `forest` package and converted into the `.cgel` format using an ad hoc Python script. Later annotation was done with a customized version[9] of Active DOP, a browser-based graphical treebanking tool (van Cranenburgh, 2018). Active DOP incorporates disco-dop (van Cranenburgh et al., 2016), an active learning parser, which considerably sped up annotation. We trained disco-dop on the 202 trees created prior to the start of the IAA pilot. As input to the Active DOP tool, EWT sentences were preannotated with POS tags and gaps heuristically derived from gold UD and PTB trees; the tagging was then manually edited in a text editor.[10] For the 50 IAA sentences, after trees were exported to the `.cgel` format, adjudication was performed cooperatively between the two annotators using a text editor with a file comparison mode.

Each split is stored in a separate file in the `.cgel` data format illustrated in Figure 4. This combines the sentence metadata style from the CONLL-U format[11] with trees in a bracketed format adapted

[9] https://github.com/nschneid/activedop

[10] At present, Active DOP does not support editing of tokenization or gap positions in the browser interface. This should be added in the future to make the tool more usable.

[11] Described in the UD docs.

POS		Nonlexical Category		Function	
1091	N	1701	Nom	6817	Head
537	P	1400	NP	935	Mod
535	V	1196	VP	630	Comp
470	D	927	Clause	627	Obj
404	N_{pro}	558	PP	457	Det
338	V_{aux}	470	DP	453	Subj
267	Adj	300	AdjP	320	Coordinate
199	Adv	201	AdvP	299	Marker
156	Coordinator	156	Coordination	142	PredComp
143	Sdr	141	$\text{Clause}_{\text{rel}}$	133	Supplement
8	Int	9	NP+PP	111	Flat
		8	IntP	79	Det-Head
		5	NP+Clause	72	Prenucleus
		3	NP+AdvP	19	Postnucleus
		3	AdjP+PP	12	Particle
155	*GAP*	1	NP+AdjP	11	Comp_{ind}

Table 3: Counts in CGELBank of lexical categories (POS tags), nonlexical categories, and grammatical functions. Special phrasal categories for coordination and some functions are not listed due to low frequency.

from PENMAN notation (Kasper, 1989). CGELBank includes a Python API for working with this format, including a script to export it to LaTeX for visualization (with any reentrancies due to fusion of functions).

During the initial phases of development, it became clear that certain structural properties (like the number of Nom layers in a complex NP) were sources of annotator inconsistency. We therefore developed a **validator**, a script to check structural properties for obvious errors (e.g., misspelled labels; phrases with no Head) as well as less obvious errors (a category occurring in an unusual position in the tree; an unnecessary level of nesting of a phrase; a Modifier forming a ternary-branching structure; invalid coindexation of a gap; improper structure of Coordinates in coordination). Some of the validation rules are conservative and need to be broadened as new data is encountered; but flagging them is an opportunity for the annotator to check for an error or inconsistency. In our experience, the rules (implemented in 500 lines of Python) often find small problems that might otherwise have gone undetected. We quantify the impact of the validator in the next section.

6 Interannotator Study

To test the consistency of our CGEL annotation guidelines, we conducted an interannotator study. As a pilot, five sentences sampled from the English Web Treebank (Bies et al., 2012) were annotated independently by the first and third authors. After adjudicating annotation disagreements and adapting to the annotation tool, we sampled 50 new sentences from EWT for the interannotator study. The annotators independently annotated the new set and then jointly adjudicated disagreements.

6.1 Evaluation Metric

A variety of measures for interannotator agreement on constituency syntax annotation exist in the literature. The standard metric is Parseval (Black et al., 1991), which computes precision and recall of the token spans that each constituent corresponds to. One problem with the usual implementation of Parseval is that it ignores hierarchy when comparing unary nodes (i.e. multiple constituents share the same token span).[12] Furthermore, there is no obviously correct way to compare trees with non-identical leaves using Parseval—which can be caused by disagreement on tokenization (e.g. on hyphenated terms) or the existence/placement of gaps, both of which we encountered in our study.

To be able to compare trees with unary nodes and potentially nonidentical tokenized strings, we turn to **Tree Edit Distance** (TED), which has been pointed out as an alternative to Parseval's reliance on token spans (Emms, 2008). TED defines a correspondence between trees via insertion, deletion (which promotes children), and substitution of nodes—it can be thought of as an extension of Levenshtein distance from strings to trees.[13] Like Levenshtein distance, TED is solved with dynamic programming; we adapt Zhang and Shasha's (1989) algorithm, with details in Appendix A. We compute microaveraged precision and recall scores based on the three types of edit costs, editing the gold tree to produce the predicted tree: deletions contribute to recall error, insertions to precision error, and substitution cost is split equally between the two. We then compute F_1 from precision and recall, which is equivalent to the *TreeDice* metric of Emms (2008) (as explained in Appendix A).

Score types. We report several scores using TED, based on different criteria for scoring candidate node alignments (matches/substitutions). In increasing order of strictness:

[12] For example, consider one tree with unary nodes $\{A,B,C\}$ and another with $\{A,C,B\}$, all corresponding to the same token span. Parseval will report both precision and recall to be 100%, which is too lenient for our purposes since the order of unary nodes matters in CGEL.

[13] If the tree is viewed as a bracketed string, structural operations insert or delete a pair of brackets and the associated node label.

Metric	1~2	1~adj	2~adj
unlab	94.8	98.1	96.0
flex	93.9	97.6	95.5
strict	91.6	96.0	94.2
gap	87.2	100.0	87.2
full-tree	18.0	54.0	32.0

Table 4: Results of the 50-sentence interannotator agreement study after the validation script. Scores are all microaveraged F1, except for `full-tree` which is the percentage of trees that are identical. See Table 2, "IAA" row for statistics of the adjudicated data.

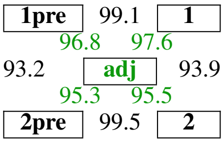

Table 5: Agreement F1 scores on 50 IAA sentences via the `flex` metric before and after validation and adjudication. **1pre** denotes the trees from annotator 1 prior to running the validation script. **1** indicates annotator 1's final trees after revisions to address warnings from the validation script. **adj** denotes the final adjudicated trees. (Exact tree match scores appear in Appendix B.)

Operation	Cost	Unit Cost
insertion	98.00	1.00
deletion	82.00	1.00
substitution	31.75	
category	11.00	0.25
function	18.75	0.25
lexeme	2.00	0.25
gap ant.	0.00	0.25

Table 6: Costs by error type for the **1~2** interannotator comparison with the `flex` metric (sum across 50 trees). E.g., 75 nodes were identified as substitutions with a different function; each of these incurs a cost of 0.25, hence 18.75 function cost. A single substitution can involve a mixture of multiple subtypes whose costs would be added together. The gap antecedent error subtype did not occur in this comparison (gaps either were inserted/deleted or had matching antecedents).

- `unlab`: Unlabelled constituents. This metric examines the tree structure alone.
- **`flex`: Labelled with function, category, and (for lexical nodes) token string, with partial credit for a node that differs in some of these respects.** For each of these components, a mismatch incurs a cost of 0.25; together these comprise the node substitution cost. An exact match has cost 0. We consider `flex` to be the main metric as it is most nuanced and should therefore induce the most accurate alignment between the two trees.
- `strict`: Labelled with all components, and no partial credit: the substitution cost is 1 for any two nodes that are not fully identical.

For gaps, the category is `GAP` and the token string is empty. Gaps are coindexed to an antecedent; this is factored into the scores by checking, after running the TED algorithm, whether two otherwise matched gaps have "the same" (aligned) antecedents. If not, the gap is not considered a full match (the `flex` penalty is 0.25).

Other metrics are:

- `gap`: F1 score of gaps per the alignment induced by the `flex` metric.
- `full-tree`: Proportion of trees that match exactly.

6.2 Results

Agreement scores between the two annotators as well as between the unadjudicated and the final adjudicated trees are reported in Table 4.[14] For all metrics, agreement F1 exceeds 90%. In particular, the `flex` metric shows an interannotator agreement F1 of 93.9%. Therefore, we are confident that, with reference to our guidelines, the CGEL formalism can be applied to the annotation of real-world text in a consistent manner.

As expected, the `strict` score of 91.6% is lower than the `flex` score, while the `unlab` score (which considers structure only) is higher, at 94.8%.

A breakdown of `flex` costs by edit type appears in Table 6. Among nodes aligned by TED, function disagreements were more numerous than category disagreements (75 vs. 44 occurrences, costing 0.25 each). But many nodes were inserted/deleted, e.g. due to attachment differences.

Zooming in to just gaps, of which there were 21 in the 50 adjudicated trees, we find good (but lower) agreement F1 of 87.2%. A major source of disagreement was a phrase in sentence 5 involving a shared object between 4 coordinated verbs—annotator 1 indicated this with 4 gaps while annotator 2 used none. Still, overall this demonstrates that even complex phenomena described in CGEL can be analysed consistently by trained annotators.

Finally, only 18.0% of trees (`full-tree`) are identical between the two annotators. However, many more of the trees between the annotators and the adjudicated set are identical—54.0% (annotator

[14] Full output of the scorer on the 50 IAA sentences is provided at: https://github.com/nert-nlp/cgel/blob/b95309f6c2ada885728b80a21b6d576bd85a20c9/datasets/iaa/iaa.out

1) and 32.0% (annotator 2).

Impact of validator. Output from the validation script was shown to each annotator after their initial pass through the 50 trees.[15] Table 5 shows the impact of the validator by reporting `flex` agreement scores before and after validation. (See also Appendix B for validator effects on exact tree accuracy.) Self-agreement before vs. after validation was 99.1% (A1) and 99.5% (A2). Agreement between the two annotators improved after validation, 93.2% → 93.9%, as the tool helped to identify spurious errors like missing or extra Nom levels in an NP, and categories in implausible functions. Agreement with the final adjudicated data increased measurably as well (A1: 96.8% → 97.6%; A2: 95.3% → 95.5%).

Note that all of the trees in the IAA experiment were created by editing trees proposed by the active learning parser, which at least featured locally well-formed structures—reducing the rate of spurious errors compared to annotation from scratch.

Qualitative findings. Many of the uncertainties and disagreements in the IAA experiment concerned structured names and measurements, including street addresses, age expressions, and temperature expressions. The phrase *over $300* exposed the problem of treating currency symbols in orthographic order, as CGEL assigns the structure [*over 300*] *dollars*, with a complex DP. Consequently, we added a guideline requiring currency expressions to be treebanked in pronunciation order, regardless of orthographic order.

Another recurring difficulty came from compounds that might have been hyphenated, like *flight test* functioning as a verb: should these be treated as one lexeme or two?

The choice of function for certain types of phrases (especially PPs) seems to lie on a continuum between Complement, Modifier, and Supplement. On substitutions, the scoring script reports 18 Comp vs. Mod disagreements and 11 Mod vs. Supplement disagreements. While it may be possible to further clarify the boundaries, it seems that some subjectivity along this continuum is inevitable.

Finally, one IAA sentence contained a fronted partitive PP (of the form *Out of* X *and* Y, *which is the best?*). We could not find an explicit account of partitive fronting in CGEL, and plan to revisit this in future work.

7 Conclusion

Using the analysis developed in CGEL (Huddleston and Pullum, 2002), we introduced a new expressive and linguistically-informed syntactic formalism to corpus annotation of English, which unifies constituent and dependency information in an accessible format. Creating annotation guidelines confirmed that CGEL was a strong foundation for syntactic analysis, but also revealed some minor points of underspecification for which new policies were necessary. Using our guidelines, we have created trees from naturally occurring sentences in multiple genres, and we conducted an interannotator study. We find high annotator agreement overall and even on the complex phenomenon of gapping. Overall, we are confident that the formalism of CGEL is suitable for consistent annotation of real-world text. In the future, we intend to take advantage of existing resources in other frameworks to obtain CGEL-style trees and parsers on a larger scale and in a wider range of genres.

Acknowledgements

Author contributions: All authors contributed to the scientific direction, writing, and analysis of this work. ▷ *BR:* Initiator of making ad hoc CGEL-style trees (shared via `https://twitter.com/DailySyntaxTree`) that blossomed into this project; annotation and guidelines lead. ▷ *AA:* Design of evaluation metric, scripting including data format reader/tree API, quantitative analysis. ▷ *NS:* Secondary annotator, editing of guidelines; design of data format, design of evaluation metric, scripting including validator, scoring, annotation tool customizations, quantitative analysis.

We thank John Payne, Geoffrey Pullum, and Pairoj Kunanupatham for useful discussion on many linguistic issues that arose in annotation as well as the CGEL formalism in general. We also thank various Twitter users (including Rui P. Chaves and Russell Lee-Goldman) who commented on syntactic analyses posted by Brett Reynolds, as well as anonymous reviewers. This research was supported in part by NSF award IIS-2144881.

References

Steven P. Abney. 1987. *The English noun phrase in its sentential aspect*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

[15] A handful of warnings were false positives, prompting changes to the script.

ES Atwell. 2008. Development of tag sets for part-of-speech tagging. In *Corpus Linguistics: An International Handbook*, volume 1, pages 501–526. Walter de Gruyter.

Ann Bies, Mark Ferguson, Karen Katz, Robert MacIntyre, Victoria Tredinnick, Grace Kim, Mary Ann Marcinkiewicz, and Britta Schasberger. 1995. Bracketing guidelines for Treebank II style Penn Treebank project.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English Web Treebank. Technical Report LDC2012T13, Linguistic Data Consortium, Philadelphia, PA.

E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. 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*.

Tatiana Bladier, Andreas van Cranenburgh, Kilian Evang, Laura Kallmeyer, Robin Möllemann, and Rainer Osswald. 2018. RRGbank: a Role and Reference Grammar corpus of syntactic structures extracted from the Penn Treebank. In *Proc. of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*, pages 5–16, Oslo, Norway.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation*, 2(4):597–620.

Chris Brew. 2003. The Cambridge Grammar of the English Language, Rodney Huddleston and Geoffrey K. Pullum. *Computational Linguistics*, 29(1):144–147.

John Chen and K. Vijay-Shanker. 2000. Automated extraction of TAGs from the Penn Treebank. In *Proceedings of the Sixth International Workshop on Parsing Technologies*, pages 65–76, Trento, Italy. Association for Computational Linguistics.

Martin Čmejrek, Jan Cuřín, Jan Hajič, and Jiří Havelka. 2005. Prague Czech-English Dependency Treebank: resource for structure-based MT. In *Proc. of EAMT*, pages 73–78, Budapest, Hungary.

Peter W. Culicover. 2004. The Cambridge Grammar of the English Language (review). *Language*, 80(1):127–141.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. of LREC*, pages 449–454, Genoa, Italy.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Yanai Elazar and Yoav Goldberg. 2019. Where's my head? Definition, data set, and models for numeric fused-head identification and resolution. *Transactions of the Association for Computational Linguistics*, 7:519–535.

Martin Emms. 2008. Tree distance and some other variants of evalb. In *Proc. of LREC*, Marrakech, Morocco.

Dan Flickinger, Yi Zhang, and Valia Kordoni. 2012. DeepBank. A dynamically annotated treebank of the Wall Street Journal. In *Proc. of the 11th International Workshop on Treebanks and Linguistic Theories*, pages 85–96.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or surface-syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 66–74, Brussels, Belgium. Association for Computational Linguistics.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Proc. of NAACL-HLT*, pages 1011–1019, Los Angeles, California.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Rodney Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge.

Rodney Huddleston, Geoffrey K. Pullum, and Brett Reynolds. 2021. *A Student's Introduction to English Grammar*, 2nd edition. Cambridge University Press.

Robert T. Kasper. 1989. A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Wolfgang Maier. 2010. Direct parsing of discontinuous constituents in German. In *Proc. of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 58–66, Los Angeles, CA, USA.

Joan Maling. 2000. A simple argument for subject gaps. In Yosef Grodzinsky, Lewis P. Shapiro, and David Swinney, editors, *Language and the Brain*. Academic Press, San Diego.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. 2004. Corpus-oriented grammar development for acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In *Proc. of IJCNLP*, Lecture Notes in Computer Science, pages 684–693, Hainan Island, China.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proc. of LREC*, pages 4027–4036, Marseille, France.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods Treebank: Motivation and Preliminary Applications. In *Proc. of COLING*.

Mateusz Pawlik and Nikolaus Augsten. 2011. RTED: A robust algorithm for the tree edit distance. arXiv:1201.0230 [cs.DB].

Mateusz Pawlik and Nikolaus Augsten. 2016. Tree edit distance: Robust and memory-efficient. *Information Systems*, 56:157–173.

John Payne, Rodney Huddleston, and Geoffrey K. Pullum. 2007. Fusion of functions: The syntax of *once, twice* and *thrice*. *Journal of Linguistics*, 43(3):565–603.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proc. of CoNLL*, pages 143–152, Sofia, Bulgaria.

Geoffrey K. Pullum and Brett Reynolds. 2013. New members of 'closed classes' in English. Manuscript.

Geoffrey K. Pullum and James Rogers. 2008. Expressive power of the syntactic theory implicit in the Cambridge Grammar of the English Language. In *Annual Meeting of the Linguistics Association of Great Britain*. University of Essex.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2004. Mapping dependencies trees: an application to question answering. In *Proc. of the International Symposium on Artificial Intelligence and Mathematics (AIM)*.

Brett Reynolds, Aryaman Arora, and Nathan Schneider. 2022. CGELBank: CGEL as a framework for English syntax annotation. arXiv:2210.00394 [cs.CL].

Brett Reynolds, Nathan Schneider, and Aryaman Arora. 2023. CGELBank annotation manual v1.0. arXiv:2305.17347 [cs.CL].

Milos Simic. 2022. Tree Edit Distance. Baeldung on Computer Science. Accessed 23 April 2023.

Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Agnieszka Patejuk, Özlem Çetinoğlu, I Wayan Arka, and Meladel Mistica. 2013. ParGramBank: The ParGram parallel treebank. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 550–560, Sofia, Bulgaria. Association for Computational Linguistics.

John Torr. 2018. Constraining MGbank: Agreement, L-selection and supertagging in Minimalist Grammars. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 590–600, Melbourne, Australia. Association for Computational Linguistics.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2011. Evaluating dependency parsing: Robust and heuristics-free cross-annotation evaluation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 385–396, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Andreas van Cranenburgh. 2018. Active DOP: A constituency treebank annotation tool with online learning. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 38–42, Santa Fe, New Mexico. Association for Computational Linguistics.

Andreas van Cranenburgh, Remko Scha, and Rens Bod. 2016. Data-oriented parsing with discontinuous constituents and function tags. *Journal of Language Modelling*, 4(1):57–111.

Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2006. Using dependency-based features to take the 'para-farce' out of paraphrase. In *Proc. of the 2006 Australasian Language Technology Workshop*, pages 131–138, Sydney, Australia.

Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262.

A Tree Edit Distance Details

For our evaluation metrics, we adapted Zhang and Shasha's (1989) TED algorithm as described in the pseudocode of Simic (2022). This is a simple recursive algorithm that compares spans of subforests in both trees, and runs in $O(n^4)$ time with memoization where n is the greater number of nodes of the two trees.[16] More efficient implementations have been proposed since, such as RTED (Pawlik and Augsten, 2011) and AP-TED (Pawlik and Augsten, 2016), but memoized TED was sufficient for our purposes—50 trees could be compared in <10 seconds with a straightforward Python implementation.

An unexpected source of inefficiency we ran into at first was the direction of recursion. If subtrees are recursed into from the rightmost child, the algorithm is an order of magnitude slower than if recursion starts from the leftmost child. Inspection of the memo-table size revealed that leftmost recursion requires much fewer function calls. We think this is because English tends to be a right-branching language, and so recursing beginning from the right increases the possible number of spans to compare between trees.

TED has been used to evaluate parsers in the past, including parsers with discontinuous constituents (Maier, 2010) and dependency parsers (Tsarfaty et al., 2011). It has also been applied or extended for other uses of comparing parse trees, such as measures of paraphrase, entailment, and answers to questions (e.g., Punyakanok et al., 2004; Wan et al., 2006; Heilman and Smith, 2010).

Relation of TED F_1 to TreeDice. Emms (2008) presents *TreeDice*, a TED-based metric for comparing constituency trees. Briefly: TED can be used to obtain the edits required to transform a gold tree into a predicted tree. With G as the size (number of nodes in) the gold tree, T as the size of the predicted tree, D as the number of deletions, I as the number of insertions, and S as the number of substitutions (where a node's label changes), *TreeDice* is given by

$$TreeDice = 1 - \frac{D+I+S}{G+T} \tag{1}$$

Using the invariant that $T = G - D + I$, one can substitute $G + I - T$ for D and show that this equals

$$\frac{2T - 2I - S}{G+T} = \frac{2(T - I - \frac{1}{2}S)}{G+T} \tag{2}$$

While Emms (2008) does not explicitly present precision and recall metrics based on TED (only ones based on `evalb` a.k.a. Parseval), we observe that the substitution cost can be split between precision and recall. Defining

$$Prec = \frac{T - I - \frac{1}{2}S}{T} \tag{3}$$

$$Rec = \frac{G - D - \frac{1}{2}S}{G} = \frac{T - I - \frac{1}{2}S}{G} \tag{4}$$

it is easily shown that the F_1 of these is equal to the *TreeDice* score (echoing the correspondence between F_1-score and the Dice coefficient over sets).

$$\begin{aligned}
F_1 &= \frac{2}{Rec^{-1} + Prec^{-1}} && (5)\\
&= 2\left(Rec^{-1} + Prec^{-1}\right)^{-1} && (6)\\
&= 2\left(\frac{G}{T - I - \frac{1}{2}S} + \frac{T}{T - I - \frac{1}{2}S}\right)^{-1} && (7)\\
&= 2\left(\frac{G+T}{T - I - \frac{1}{2}S}\right)^{-1} && (8)\\
&= \frac{2(T - I - \frac{1}{2}S)}{G+T} && (9)\\
F_1 &= TreeDice && (10)
\end{aligned}$$

B Exact tree accuracy

For comparison with the `flex` metric IAA results in Table 5, we also report exact tree accuracy below.

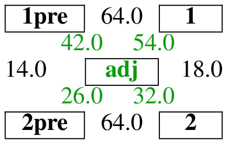

Table 7: Exact tree accuracy scores (i.e. whether trees are identical) on 50 IAA sentences before and after validation and adjudication. **1pre** denotes the trees from annotator 1 prior to running the validation script. **1** indicates annotator 1's final trees after revisions to address warnings from the validation script. **adj** denotes the final adjudicated trees.

[16] Or, more precisely, $O(m^2n^2)$, where m and n are the sizes of the respective trees, as the recurrence is parameterized by a contiguous span of nodes in each tree under postorder traversal.

C EWT Examples in PTB and CGELBank

Trees in the two styles appear in Figures 5 and 6 for comparison. Further cross-framework comparisons appear in Reynolds et al. (2022, §4).

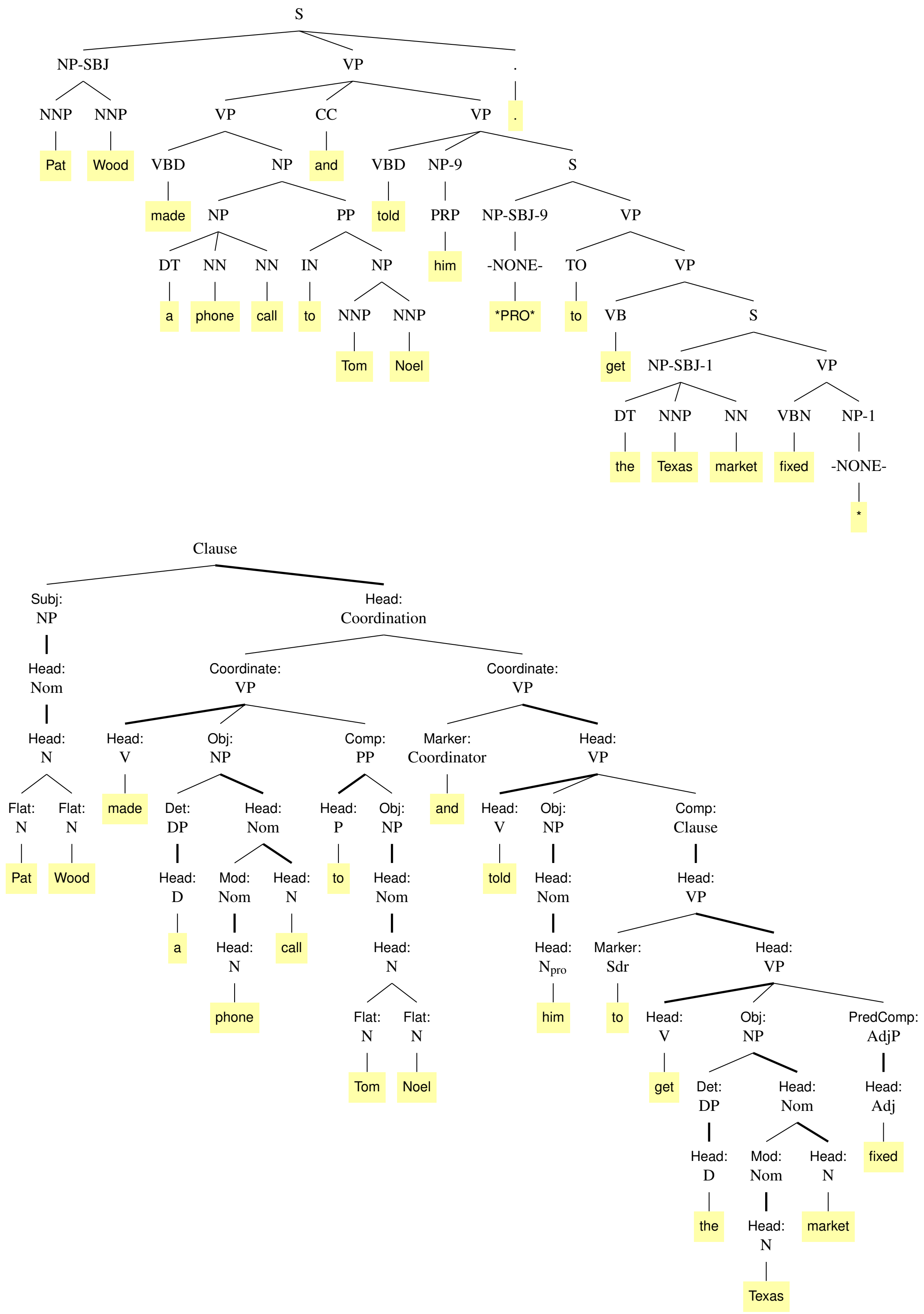

Figure 5: PTB-style and CGELBank trees for an EWT sentence with VP coordination. (Note that NPs are flatter in PTB style, and that control is indicated in PTB style with ∗PRO∗, but not in CGELBank.)

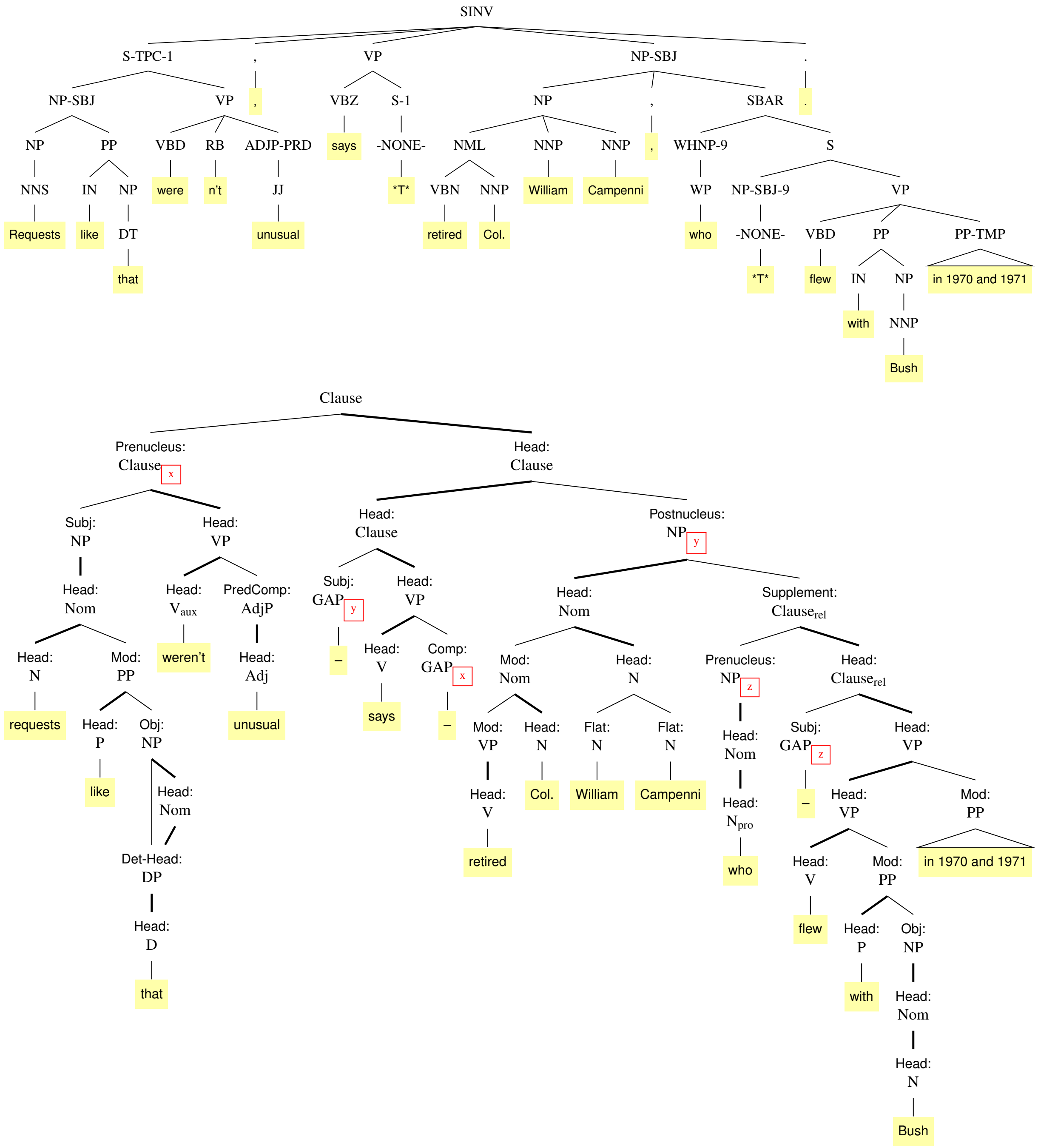

Figure 6: PTB-style and CGELBank trees for an EWT sentence with inversion and a relative clause (part of the tree is collapsed for space). Traces indicated with ∗T∗ in PTB generally map to gaps in CGELBank.