CCGweb: a New Annotation Tool and a First Quadrilingual CCG Treebank

Kilian Evang
University of Düsseldorf
Germany
evang@hhu.de

Lasha Abzianidze
University of Groningen
The Netherlands
l.abzianidze@rug.nl

Johan Bos
University of Groningen
The Netherlands
johan.bos@rug.nl

Abstract

We present the first open-source graphical annotation tool for combinatory categorial grammar (CCG), and the first set of detailed guidelines for syntactic annotation with CCG, for four languages: English, German, Italian, and Dutch. We also release a parallel pilot CCG treebank based on these guidelines, with 4x100 adjudicated sentences, 10K single-annotator fully corrected sentences, and 82K single-annotator partially corrected sentences.

1 Introduction

Combinatory Categorial Grammar (CCG; Steedman, 2000) is a grammar formalism distinguished by its transparent syntax-semantics interface and its elegant handling of coordination. It is a popular tool in semantic parsing, and treebank creation efforts have been made for Turkish (Çakıcı, 2005), German (Hockenmaier, 2006), English (Hockenmaier and Steedman, 2007), Italian (Bos et al., 2009), Chinese (Tse and Curran, 2010), Arabic (Boxwell and Brew, 2010), Japanese (Uematsu et al., 2013), and Hindi (Ambati et al., 2018). However, all of these treebanks were not directly annotated according to the CCG formalism, but automatically converted from phrase structure or dependency treebanks, which is an error-prone process. Direct annotation in CCG has so far mostly been limited to small datasets for seeding or testing semantic parsers (e.g., Artzi et al., 2015), and no graphical annotation interface is available to support such efforts, making the annotation process difficult to scale. The only exceptions we are aware of are the Groningen Meaning Bank (Bos et al., 2017) and the Parallel Meaning Bank (Abzianidze et al., 2017), two annotation efforts which use a graphical user interface for annotating sentences with CCG derivations and other annotation layers, and which have produced CCG treebanks for English, German, Italian, and Dutch. However, these efforts are focused on semantics and have not released explicit guidelines for syntactic annotation. Their annotation tool is limited in that annotators only have control over lexical categories, not larger constituents. Even though CCG is a lexicalized formalism, where most decisions can be made on the lexical level, there is no full control over attachment phenomena in the lexicon. Moreover, these annotation tools are not open-source and cannot easily be deployed to support other annotation efforts.

In this paper, we present an open-source, lightweight, easy-to-use graphical annotation tool that employs a statistical parser to create initial CCG derivations for sentences, and allows annotators to correct these annotations via *lexical category constraints* and *span constraints*. Together, these constraints make it possible to effect (almost) all annotation decisions consistent with the principles of CCG. We also present a pilot study for multilingual CCG annotation, in which a parallel corpus of 4x100 sentences (in English, German, Italian, and Dutch) was annotated by two annotators per sentence, a detailed annotation manual was created, and adjudication was performed to create a final version. We publicly release the manual, the annotation tool, and the adjudicated data. Our release also includes an additional $> 10\,\mathrm{K}$ derivations, each manually corrected by a single annotator, and an additional $> 82\,\mathrm{K}$ sentences, each partially corrected by a single annotator.

2 An Annotation Tool for CCG

Our annotation tool CCGweb[1] is Web-based, implemented in Python, PHP, and JavaScript, and should be easy to deploy on any recent Linux dis-

[1] https://github.com/texttheater/ccgweb

Proceedings of the 13th Linguistic Annotation Workshop, pages 37–42
Florence, Italy, August 1, 2019.

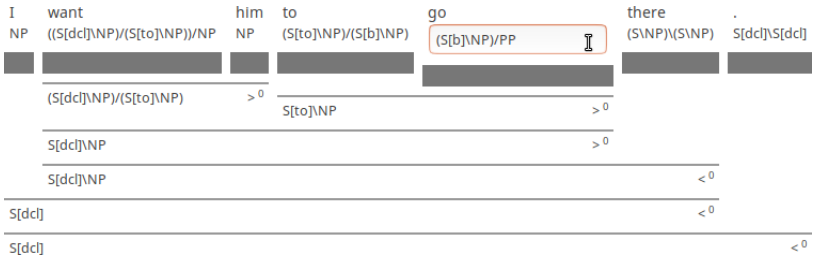

Figure 1: Correcting a lexical category.

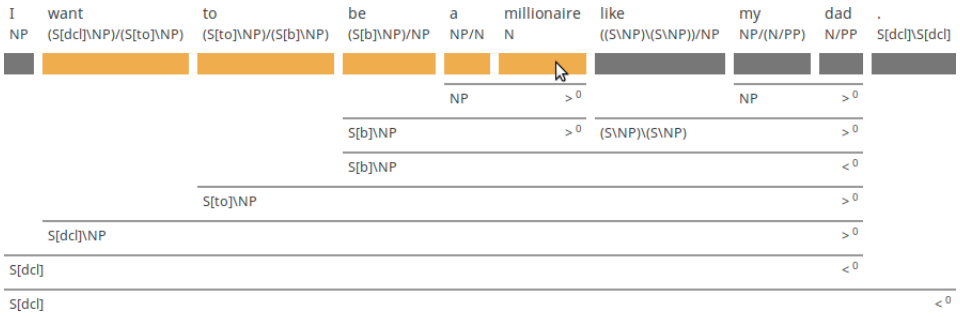

Figure 2: Correcting attachments by selecting a span that need to form a constituent.

tribution. It has two main views: the home page shows the list of sentences an annotator is assigned to annotate. Those already done are marked as "marked correct". Clicking on a sentence takes the annotator to the sentence view. Annotators can also enter arbitrary sentences to annotate, e.g., for experimenting or for producing illustrations.

Dynamic Annotation Annotation follows an approach called *dynamic annotation* (Oepen et al., 2002) or *human-aided machine annotation* (Bos et al., 2017), in which sentences are automatically analyzed, annotators impose *constraints* to rule out undesired analyses, sentences are then reanalyzed subject to the constraints, and the process is repeated until only the desired analysis remains. The current system is backed by the EasyCCG parser (Lewis and Steedman, 2014), slightly modified to allow for incorporating constraints, and other CCG parsers could be plugged in with similar modifications.

What You See Is What You Get Derivations are rendered in the same graphical format that is used in the literature, representing nodes as horizontal lines placed underneath their children. Annotators directly interact with this graphical representation when annotating, following the WYSIWYG (what you see is what you get) principle.

Lexical Category Constraints As an example of editing, consider Figure 1. Suppose that the parser has analyzed *there* as an adjunct with category $(S \setminus NP) \setminus (S \setminus NP)$, but we wish to analyze it as an argument to the verb *go* with category PP. As a result, the category of the verb also has to change, *viz.* from $S[b] \setminus NP$ to $(S[b] \setminus NP) / PP$. To do this, the annotator clicks on the category and changes it, as shown in the figure. When they hit enter or click somewhere else, the sentence is automatically parsed again in the background, this time with the *lexical category constraint* that *go* has category $(S[b] \setminus NP) / PP$. In many cases, the parser will directly find the desired parse, with *there* being a PP, and the annotator only has to check it, not make another edit.

Span Constraints Although constraining lexical categories is often enough to determine the entire CCG derivation (cf. Bangalore and Joshi, 1999; Lewis and Steedman, 2014), this is not *always* the case. For example, consider the sentence *I want to be a millionaire like my dad.* Assuming that *like my dad* is a verb phrase modifier (category $(S \setminus NP) \setminus (S \setminus NP)$), it could attach to either *to be* or *want*, giving very different meanings (cf. Zimmer, 2013). We therefore implemented one other type of edit operation/constraint: *span constraints*. By simply clicking and dragging across a span of tokens as shown in Figure 2, annotators can constrain this span to be a constituent in the resulting parse.

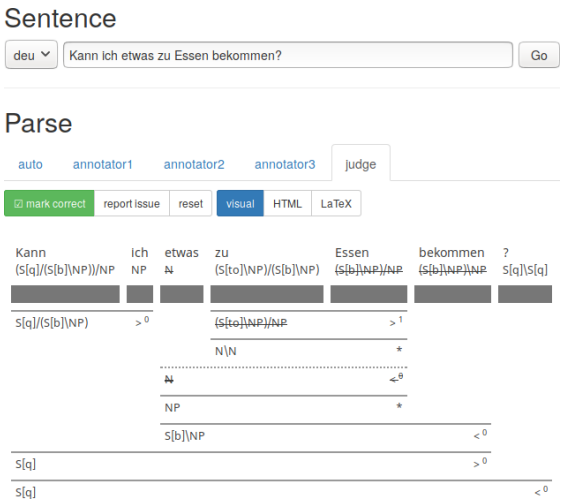

Figure 3: The judge user sees all annotators' versions and a diff view where categories with disagreements are struck through and spans with disagreements are dotted.

Additional Features Our tool offers annotators some additional convenient features. When unsure about some annotation decision, they can click the "report issue" button to open a discussion thread in an external forum, such as a GitHub issue tracker. To erase all constraints and restart annotation from the parser's original analysis, an annotator can click the "reset" button. And the buttons "HTML" and "LaTeX" provide code that can be copied and pasted to use the current derivation as an illustration on a web page or in a paper.

Adjudication Support Once two or more annotators have annotated a sentence, disagreements need to be discovered, and a final, authoritative version has to be created. Our tool supports this adjudication process through the special user account judge. This user can see the derivations of other annotators in a tabbed interface as shown in Figure 3. In order to enable the judge to easily spot disagreements, categories that annotators disagree on are struck through, and constituents that annotators disagree on are dashed.

3 A Quadrilingual Pilot CCG Treebank

To test the viability of creating multilingual CCG treebanks by direct annotation, we conducted an annotation experiment on 110 short sentences from the Tatoeba corpus (Tatoeba, 2019), each in four translations (English, German, Italian, and Dutch). The main annotation guideline was to copy the annotation style of CCGrebank (Honnibal et al., 2010), a CCG treebank adapted from CCGbank (Hockenmaier and Steedman, 2007), which is in turn based on the Penn Treebank (Marcus et al., 1993). Since CCGrebank only covers English and lacks some constructions observed in our corpus, an annotation manual with more specific instructions was needed. We initially annotated ten sentences in four languages and discussed disagreements. The results were recorded in an initial annotation manual, and the initial annotations were discarded. Each of the remaining 4x100 sentences was then annotated independently by at least two of the authors.

Table 1 (upper part) shows the number of non-overlapping category and span constraints that each annotator created on average per sentence before marking the sentence as correct. Annotated sentences were manually classified by the first author into four classes: (0) sentences without any disagreements, (1) sentences with only trivial violations of the annotation guidelines (e.g., concerning attachment of punctuation or underspecifying modifier features), (2) sentences with only apparent oversights, such as giving a determiner a pronoun category, (3) sentences with more intricate disagreements which required additional guidelines to resolve. Table 1 (upper part) shows the distribution of disagreement classes, and Table 2 shows examples of class (3). The first author adjudicated all disagreements and updated the annotation manual accordingly. We release the manual and the full adjudicated dataset.[2]

To make the resource more useful (e.g., for training parsers), we also include in the release the syntactic CCG derivations created so far in the Parallel Meaning Bank (Abzianidze et al., 2017). These do not follow the annotation guidelines in detail due to their focus on semantics, nor have they been adjudicated, but instead corrected by a single annotator. However, they are much greater in number. For an even greater number, we also release *partially corrected* derivations, meaning that the annotator made at least one change to the automatically created derivation. Table 1 (lower part) shows statistics of this additional data.

4 Conclusions and Future Work

We have presented the first open-source graphical annotation tool for combinatory categorial grammar. Its features include dynamic annotation via

[2] https://ccgweb.phil.hhu.de/

		English	German	Italian	Dutch
adjudicated sentences		100	100	100	100
∅ length		6.8	8.1	6.6	7.5
∅ category constraints per annotator		1.8	2.7	2.6	2.5
∅ span constraints per annotator		1.1	1.1	1.2	1.1
by disagreement	(0) none	10	32	27	34
	(1) trivial	45	17	16	12
	(2) oversight	1	7	4	8
	(3) intricate	44	44	53	46
single annotator, fully corrected		7 182	1 703	941	868
∅ length		6.4	5.7	5.4	5.9
single annotator, partially corrected		74 769	4 331	2 652	1 130
∅ length		8.6	7.4	6.9	7.4

Table 1: Corpus statistics and disagreements

Language	Disagreement
English	Argument or adjunct? Take$_{((S[b]\NP)/PP)/NP}$ a taxi$_{PP/NP}$ to the hotel . Take$_{(S[b]\NP)/NP}$ a taxi$_{(S\NP)\(S\NP)}$ to the hotel .
	Clausal argument or adjunct? Can I have something$_{NP/(S[to]\NP)}$ to$_{(S[to]\NP)/(S[b]\NP)}$ eat$_{S[b]\NP}$? Can I have something$_{N}$ to$_{(S[to]\NP)/(S[b]\NP)}$ eat$_{(S[b]\NP)/NP}$?
	Modification of copula or adjective? My mother is always$_{(S[adj]\NP)/(S[adj]\NP)}$ busy . My mother is always$_{(S\NP)\(S\NP)}$ busy .
German	Treatment of quoted speech Sag$_{(S[b]\NP)/NP}$ nur ja$_{N}$ oder$_{(N\N)/N}$ nein$_{N}$. Sag$_{(S[b]\NP)/S[intj]}$ nur ja$_{S[intj]}$ oder$_{(S[intj]\S[intj])/S[intj]}$ nein$_{S[intj]}$.
	Analysis of *wh*-questions Wer$_{S[wq]/(S[dcl]\NP)}$ hat$_{(S[dcl]\NP)/(S[pt]\NP)}$ diesen Brief geschrieben ? Wer$_{S[wq]/(S[q]\NP)}$ hat$_{(S[q]\NP)/(S[pt]\NP)}$ diesen Brief geschrieben ?
	Scope of negation Rufen Sie mich nicht$_{(S/S)/(S/S)}$ mehr an ! Rufen Sie mich nicht$_{S\S}$ mehr an !
Italian	Analysis of *wh*-questions Ci potete$_{S[q]/(S[b]\NP)}$ aiutare ?$_{S[q]\S[q]}$ Ci potete$_{S[dcl]/(S[b]\NP)}$ aiutare ?$_{S[q]\S[dcl]}$
	Category ambiguity in parts of multiword expressions Sono tre anni che Tom è andato$_{((S[pt]\NP)/PP)/NP}$ via$_{N}$ da Boston . Sono tre anni che Tom è andato$_{((S[pt]\NP)/PP)/PR}$ via$_{PR}$ da Boston .
	di: preposition or complementizer? Gli ho chiesto$_{((S[pt]\NP)\NP)/PP}$ di$_{PP/(S[b]\NP)}$ farlo . Gli ho chiesto$_{((S[pt]\NP)\NP)/(S[to]\NP)}$ di$_{(S[to]\NP)/(S[b]\NP)}$ farlo .
Dutch	Argument or adjunct? Een eekhoorntje verstopte$_{((S[dcl]\NP)/PP)/NP}$ zich tussen$_{PP/NP}$ de takken . Een eekhoorntje verstopte$_{(S[dcl]\NP)/NP}$ zich tussen$_{((S\NP)\(S\NP))/NP}$ de takken .
	Participles in attributive use Windows is het meest$_{(N/N)/(N/N)}$ gebruikte$_{N/N}$ besturingssysteem in de wereld . Windows is het meest$_{(N/N)/(S[pss]\NP)}$ gebruikte$_{S[pss]\NP}$ besturingssysteem in de wereld .
	met: nominal or verbal argument? Hij is gestopt met$_{PP/NP}$ roken$_{N}$. Hij is gestopt met$_{PP/S[b]\NP}$ roken$_{S[b]\NP}$.

Table 2: Examples of intricate disagreements

lexical label constraints and span constraints, adjudication support, and various conveniences.

We have used this tool to create the first published CCG resource that comes with an explicit annotation manual for syntax and has been created by direct annotation, rather than conversion from a non-CCG treebank. It is multilingual, currently including English, German, Italian, and Dutch, and aims for cross-lingually consistent annotation guidelines.

For future work, we envision more extensive direct annotation of multilingual data with CCG derivations, and putting them to use for evaluating unsupervised and distantly supervised CCG parsers. We would also like to investigate the use of our tool as an interactive aid in teaching CCG.

Acknowledgments

The work of the first author was funded by the Consolidator Grant "TreeGraSP" of the European Research Council (ERC). The work of the second and third author was funded by the NWO-VICI grant "Lost in Translation – Found in Meaning" (288-89-003)". We would like to thank the three anonymous reviewers for their valuable comments.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247. Association for Computational Linguistics.

Bharat Ram Ambati, Tejaswini Deoskar, and Mark Steedman. 2018. Hindi CCGbank: A CCG treebank from the Hindi dependency treebank. *Language Resources and Evaluation*, 52(1):67–100.

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710. Association for Computational Linguistics.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics, Volume 25, Number 2, June 1999.*

Johan Bos, Valerio Basile, Kilian Evang, Noortje J. Venhuizen, and Johannes Bjerva. 2017. The Groningen Meaning Bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 463–496. Springer Netherlands, Dordrecht.

Johan Bos, Cristina Bosco, and Alessandro Mazzei. 2009. Converting a dependency treebank to a categorial grammar treebank for Italian. In *Eight international workshop on treebanks and linguistic theories (TLT8)*, pages 27–38. Educatt.

Stephen A. Boxwell and Chris Brew. 2010. A pilot Arabic CCGbank. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. European Languages Resources Association (ELRA).

Julia Hockenmaier. 2006. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 505–512. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics, Volume 33, Number 3, September 2007.*

Matthew Honnibal, James R. Curran, and Johan Bos. 2010. Rebanking CCGbank for improved NP interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 207–215. Association for Computational Linguistics.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics, Volume 19, Number 2, June 1993, Special Issue on Using Large Corpora: II.*

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods Treebank: Motivation and preliminary applications. In *COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes.*

Mark Steedman. 2000. *The syntactic process.* MIT press Cambridge, MA.

Tatoeba. 2019. Tatoeba: Collection of sentences and translations. `https://tatoeba.org/`. Accessed: 2019-04-08.

Daniel Tse and James R. Curran. 2010. Chinese CCGbank: extracting CCG derivations from the Penn Chinese Treebank. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1083–1091. Coling 2010 Organizing Committee.

Sumire Uematsu, Takuya Matsuzaki, Hiroki Hanaoka, Yusuke Miyao, and Hideki Mima. 2013. Integrating multiple dependency corpora for inducing wide-coverage Japanese CCG resources. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1042–1051. Association for Computational Linguistics.

Ben Zimmer. 2013. Attachment ambiguity in "Frazz". http://languagelog.ldc.upenn.edu/nll/?p=4566.

Ruken Çakıcı. 2005. Automatic induction of a CCG grammar for Turkish. In *Proceedings of the ACL Student Research Workshop*, pages 73–78, Ann Arbor, Michigan. Association for Computational Linguistics.