Expanding the Universal Dependencies Ancient Hebrew Treebank with Constituency Data

Daniel G. Swanson
Department of Linguistics,
Indiana University,
dangswan@iu.edu

Abstract

This paper presents an effort to expand the annotation pipeline for the Ancient Hebrew Universal Dependencies treebank to make use of additional data, resulting in the addition of over 4000 sentences and roughly 100K words to the released treebank. The resulting treebank contains 5500 sentences and 145K words and the incorporation of converted constituency data has resulted in an annotation process which requires manual intervention in only around 15-20% of sentences, even in previously unseen genres.

1 Introduction

Swanson and Tyers (2022) developed a rule-based parser and used it to produce a UD treebank of portions of the Hebrew Scriptures. In this paper, we extend their processing pipeline to addtionally take input from a partial constituency treebank.

The Hebrew Scriptures are a collection of 39 books primarily written in the first millennium BC in Ancient Hebrew (with a few passages in Aramaic) which were arranged and codified in their current form over the course of the first millennium AD. They are also known as the Tanakh, an acronym of the Hebrew names of the 3 main divisions: תורה /torah/ "law"[1], נביאים /nevi'im/ "prophets" (a category which also includes several books of narrative history), and כתובים /ketuvim/ "writings".

The Universal Dependencies (UD) project (Nivre et al., 2020) is a collaborative effort to create a collection of treebanks in a single cross-linguistically consistent annotation scheme so as to better facilitate studying syntax in multiple languages.

This paper is organized as follows: Section 2 describes the existing corpora used to create the treebank, Section 3 explains the existing pipeline and our modifications and evaluation, Section 4 discusses changes we made to the annotation guidelines for Ancient Hebrew, Section 5 provides statistics on the resulting treebank, and Section 6 concludes.

2 Data Sources

The data for this project comes from two sources: The Biblia Hebraica Stuttgartensia Amstelodamensis (BHSA) and the MACULA project, both of which annotate the same underlying text. Each corpus contributes valuable but incomplete data to the task of producing a UD treebank.

The BHSA (Peursen et al., 2015) provides extensive morphological annotation of the text, and some light semantic annotation. Its syntactic annotations, however, are extremely limited, with the structure of most sentences being restricted to a two-layer constituency tree (phrases and clauses), as shown in Figure 1. Deriving slightly more detailed trees from the BHSA data is possible[2], but Swanson and Tyers (2022) nonetheless ended up building a system that was much closer to a parser than to a treebank converter.

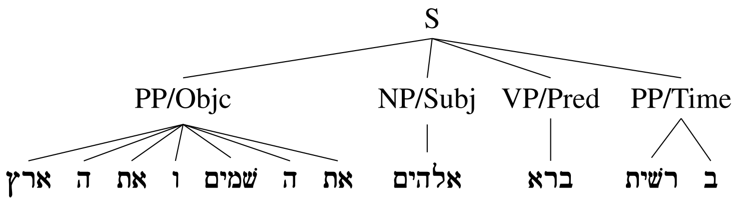
In contrast, MACULA (Clear Bible, 2022), which was released after Swanson and Tyers (2022), has full syntactic information up to the clause level, as shown in Figure 2. However, for more complex sentences, it often leaves the upper layers underspecified, such that each clause is fully annotated, but the relations between clauses are not, as shown in Figure 3.

Fortunately, some of BHSA's more semantic features can help fill this gap. One of the most important is a feature called `txt`, which marks the "level" of the text, in particular distinguishing between narrative, quotations, and quotations embedded within quotations, which is usually sufficient to resolve

[1] Instances of Hebrew script in this paper are followed by a transliteration in slashes according to the ALA-LC scheme (Barry, 1997) and an English translation in quotes.

[2] See, for example `https://github.com/ETCBC/trees`.

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 23–31
August 28-29, 2025 ©2025 Association for Computational Linguistics



ברשית ברא אלהים את השמים ואת הארץ

ארץ	ה	את	ו	שמים	ה	את	אלהים	ברא	רשית	ב
’arets	ha	’et	ve	shamayim	ha	’et	’elohiym	bara’	reshiyt	be
land	the	ACC	and	sky	the	ACC	God	created	beginning	in
subs	art	prep	conj	subs	art	prep	subs	verb	subs	prep

“In the beginning, God created the sky and the land.”

Figure 1: The syntax tree and gloss of Genesis 1:1, as given by the BHSA (the source used for prior work). Phrase nodes are marked with both their category and their function label (here, “object”, “subject”, “predicate”, and “time”).

the attachment of clauses which are complements of speaking verbs or are simply coordinated. (Other types of subordination present further challenges, which are discussed below.)

3 Methodology

Swanson and Tyers (2022) used Constraint Grammar (CG) (Bick and Didriksen, 2015) as the basis for the following pipeline:

1. Convert BHSA to CG format
2. Parse with CG
3. Convert to CoNLL-U format
4. Apply UDapy to attach punctuation (Popel et al., 2017)
5. Manually review trees

For our work, we extend the pipeline to the process depicted in Figure 4 by adding the following steps:

Rule extraction This script converts the constituency structure of MACULA into a dependency structure. Each node which has multiple children (apart from the top-level sentence node) has attributes which specify which child is the head and what parsing rule generated the node. For each such node, the head is set as the parent of all the other children, and the children receive the rule name as a tag. This process is depicted in Figure 6. This step is substantially simpler than what often appears in other conversion projects, since it only requires a tree traversal to collect all the relations without needing a set of heuristics to identify the heads at each level (Arnardóttir et al., 2020; Chun et al., 2018; Kuzgun et al., 2021).

Alignment and arc projection This step determines the correspondences between word IDs in BHSA and MACULA. Since the two corpora represent the same underlying text, this is trivial at the sentence level, but presents some challenges at the word level due to differences in tokenization, some of which are shown in (1).

(1) לָבוֹא אֶל יִצְחָק אַבִיו אַרְצָה כְּנָעַן

/labo’ ’el yitsḥaḳ ’aviṿ ’artsah ken‘an/

BHSA	כְּנָעַן		אַרְצָה
MACULA	כְּנָעַן	ה-	אַרְצָ
UD	כְּנָעַן		אַרְצָה
Gloss	Canaan	LOC	the land of

	אָבִיו	יִצְחָק	אֶל	בוֹא	לָ-
ו-	אָבִי	יִצְחָק	אֶל	בוֹא	לָ-
ו-	אָבִי	יִצְחָק	אֶל	בוֹא	לָ-
his	father	Isaac	toward	come	to

“in order to come to Isaac his father in the land of Canaan” (from Genesis 31:18)

Here we see the two most frequent divergences: MACULA treats the locative suffix as a separate unit, unlike BHSA and UD, and

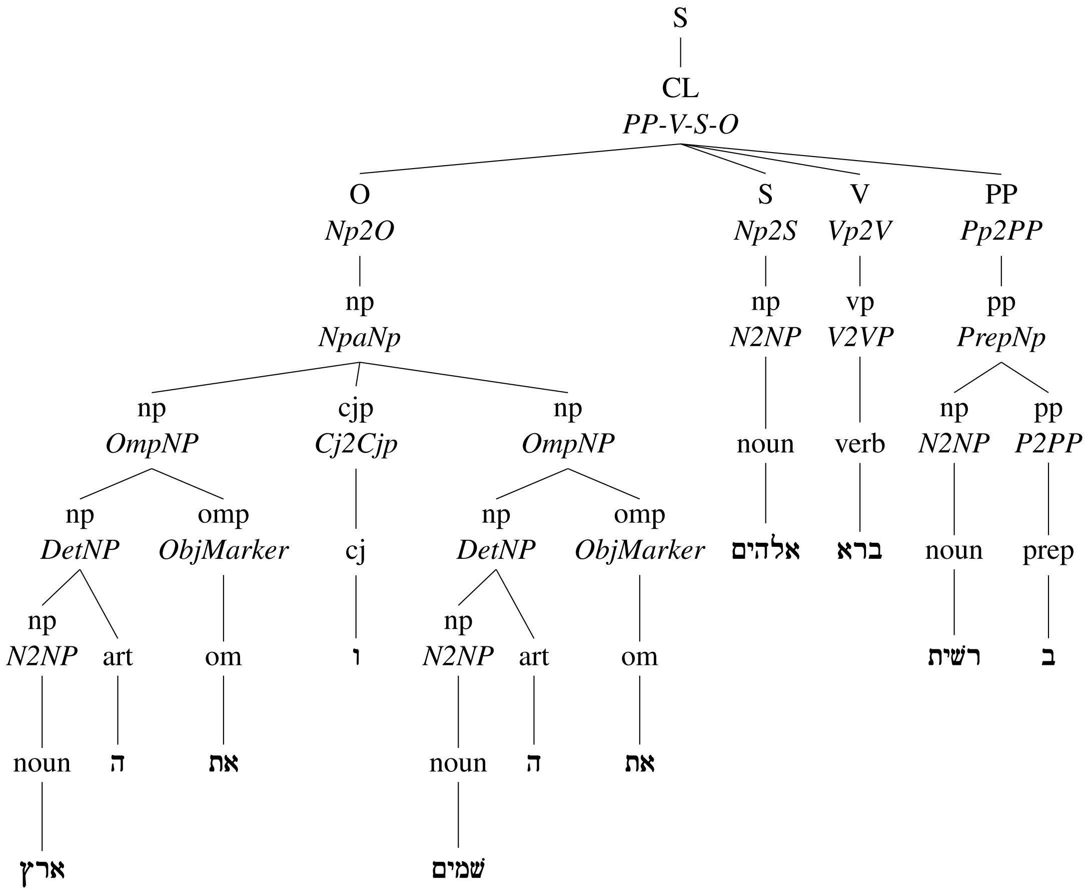


Figure 2: The syntax tree of Genesis 1:1, as given by MACULA (the source added in this work). Labels in *italics* are the names of the parsing rules that generated the nodes. The corresponding gloss can be found in Figure 1.

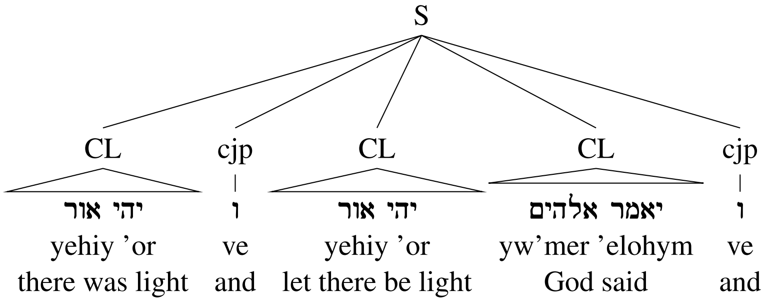


Figure 3: The top layer of the syntax tree of Genesis 1:3 in MACULA. While each CL (clause) node shown has full internal structure, none of the annotations give any indication of how these constituents are related to one another, hence the continued need for the data from BHSA.

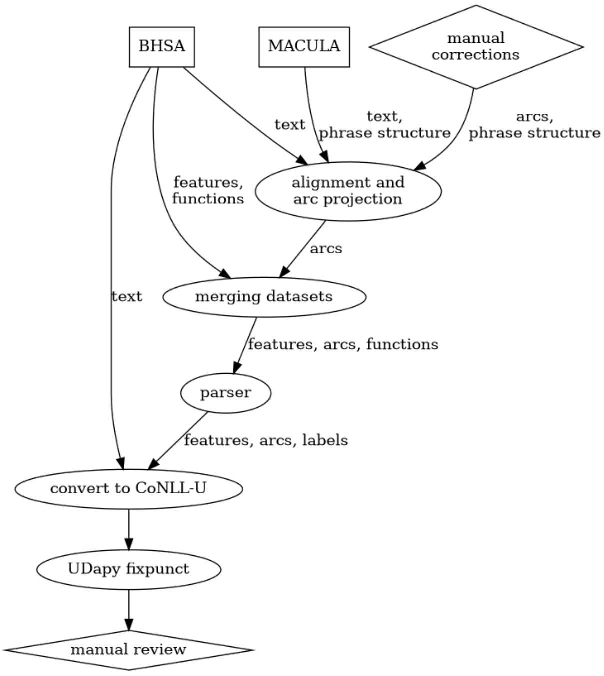


Figure 4: The relationship between the different soruces of data and the scripts that combine them. Rectangular nodes represent external corpora, oval nodes represent scripts and tools, and diamond nodes represent data reviewed or created by the authors. Edge labels indicate the information that is passed between the nodes.

BHSA does not treat pronominal suffixes as separate units, unlike MACULA and UD. The alignment script applies a set of rules describing these divergences to the two texts until it achieves an exact match. If such a match is not found, it reports the sentence to the developer for review.

Having generated these alignments, the script converts the constituency structure of the MACULA nodes into dependency arcs between BHSA words. To do this, it takes the head of each phrase node (which MACULA specifies) and adds an arc from it to the head of each of its siblings, using the rule name as the arc label. Each word is then identified with all the phrases it is the head of, producing a dependency tree between words. This process is depicted in Figure 6.

Manual corrections This consists of two sets of files, both of which override specific parts of the Alignment and arc projection step. One set identifies nodes in the MACULA data, and changes which child is marked as the head. One case where this occurs is in copular clauses where the heuristics in the rule extraction script are not always able to select the correct predicate, such as when the correct predicate is a locative adverb. The second set identifies a BHSA word, and changes which word is its head and/or adds tags (including the dependency label). This is most frequently used in cases where the parser fails to correctly attach subordinate clauses. Examples of both types are given in Figure 5. These two sets of overrides replace the previous system, in which the full CoNLL-U of any tree that required manual correction would be copied to a separate file.

```
#60
w1245 w1251
w1247 w1251 @obl
#77
w1607 _ @conj

#393
0101701400120180 2
```

Figure 5: Examples of entries in the manual override system for Genesis. The first two entries specify that particular words in sentence 60 should have different heads, and the second additionally specifies that the relation label should be set to obl. The third, meanwhile, specifies that a particular word in sentence 77 should have the relation conj, but should keep whatever head was extracted from MACULA. Finally, the last entry specifies that in sentence 393, when extracting rules from a particular MACULA node, daughter node 2 should be treated as the head.

With these changes, we were able to replace significant portions of the original parser with a set of rules that largely amount to a decision tree converting MACULA's rule names into UD relations, using BHSA morphological and semantic labels to disambiguate them (such rules now make up roughly one third of all rules in the parser). An example is given in (2).

(2)

```
WITH NOMAPPED (NpAdjp) {
  MAP @det (prde) OR (ppde) ;
  MAP @acl:relcl (verb) ;
  MAP @nummod (adjv ordn) ;
  MAP @amod (*) ;
} ;
```

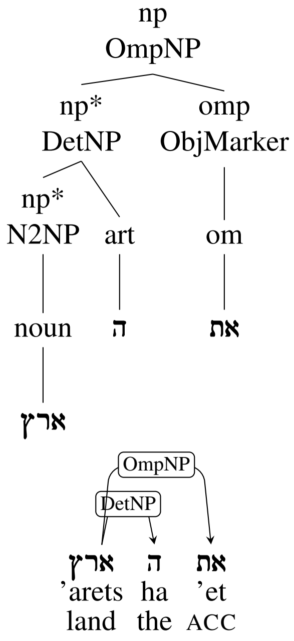
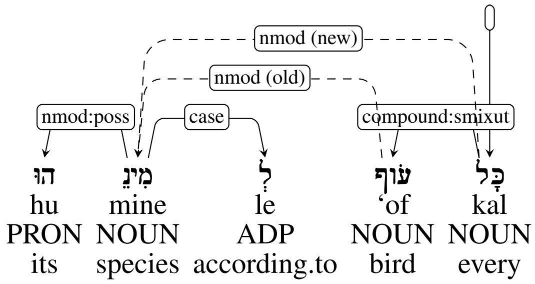


Figure 6: The process of converting from MACULA trees into initial dependencies. Nodes marked with * are the heads of their parent rules. Note that the rule of the N2NP node does not appear in the resulting tree, because it has only one child. These labels will be changed case and det by the Constraint Grammar rules.

This fragment uses the recently added compound rules (Swanson et al., 2023) to create a nested conditional, where the first line restricts the subsequent rules to apply only to words whose rule label is NpAdjp (adjective phrase modifying a noun phrase). The rules check first if the word is tagged as demonstrative, then if it is a verb, then if it is an ordinal number, and finally apply amod (adjectival modifier) if none of the other conditions apply.

As we adjusted the parser to use MACULA input, we regularly checked its output against the previously validated trees (both those that appeared in the released treebank and another roughly 300 trees which had not been released since they did not constitute a complete document). In the process, we discovered and fixed a number of inconsistencies, largely in modifier attachment, such as the one shown in (3), where the dashed lines indicate the old analysis and the new analysis.

(3)



Quantifiers in Hebrew are morphologically nouns, which combine with other nouns via a highly productive compounding construction. In the existing trees, it was common for modifiers on such phrases, such as the phrase meaning "according to its species" in (3), to attach to the lexical noun (here "bird") which would in such cases arguably be the semantic head, rather than the quantifier which is the morphosyntactic head. We have updated these instances to be more consistent in their treatment of such constructions, so that modifiers are attached to the entire compound phrase unless there is a good reason to do otherwise (adjectives are still sometimes attached to the dependent noun if their agreement features do not match those of the head noun).

Once this updating was complete, we performed an analysis of the accuracy of the parser on unvalidated data. In virtually all orderings of the books of the Hebrew Scriptures, the book of Exodus is the second book after Genesis, and thus seemed a natural continuation of the project. According to the method of splitting sentences which had been implemented in the treebank, Exodus contains 1151 trees[3], 118 of which had already been validated. We then manually inspected the remaining 1033 trees, and judged 810 (78.4%) of them to be fully correct without further modification.

For the remaining 223 trees, most only needed corrections for a handful of words. In a number of cases, this was due to there being rule names which had not come up in the Genesis data which needed to be added to the section of the parser that converts rule names to dependency labels. The largest source of such instances was for sequences of coordinated phrases, because MACULA has a separate rule name for each possible sequence, such as NPNPaNPaNPaNP, indicating a sequence of five noun phrases with a conjunction between each pair

[3] This is slightly lower than the traditional number of verses because of a handful of instances where a long list of objects crosses verse boundaries, resulting in a verse which is not a sentence.

except the first, and NPNPaNPNPaNP which is similar but also lacks a conjunction between the third phrase and the fourth. (We later adjusted the rules which handle coordination to recognize the pattern of such rules rather than requiring a fixed list, which should reduce the number of unknown rules in future expansions.)

Interestingly, the parser had roughly 80% sentence-level accuracy almost regardless of the genre of the text, as shown in Table 1. There is a very slight drop in performance on narrative in comparison to other genres, though this is likely a result of the longer sentences. Genesis and Ruth, the texts the parser was developed for, are both almost entirely narrative. Exodus, meanwhile, also contains some songs, a legal code, and building instructions. The songs, despite being poetry and in a noticeably more archaic style than the surrounding narrative, were actually the sections where the parser performed best. The only trees that were marked as incorrect were two which contained a word that had not been included in the list of subordinating conjunctions and thus did not receive a part of speech tag. This high performance is probably due to the fact that the poetic sections contain relatively few subordinate clauses and feature slightly shorter sentences. Their divergence from other genres is partly lexical, which affects hardly any rules, and partly word order, which also has little effect in this case, because most of the relations that differ are for nominal arguments (subject, object, etc.) which are usually accepted from MACULA as-is.

After completing this analysis for Exodus, we applied a similar methodology to the subsequent books. For each one, we made a first pass through the book, marking the trees that were already correct and leaving the rest for further processing. Then, in the course of correcting the remaining trees, we made various improvements to the rules before starting on the next book. The results of this process are given in Table 2, and they show a general upward trend reaching 84% by the end of the project.

4 Annotation Decisions

In the process of revising the treebank, there were hundreds of local changes, such as the modifier attachment discussed above, and a handful of larger systematic ones. In this section, we present three of the latter kind: the introduction of fixed expressions, a change in the tokenization guidelines relating to quotations, and the use of the expletive pronoun relation.

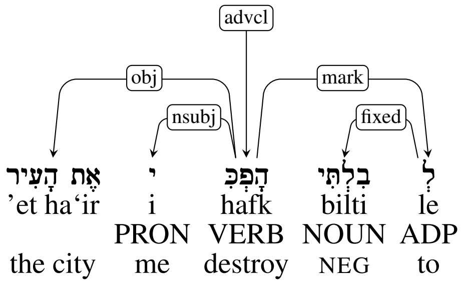
Further details about the annotation decisions of the treebank in general can be found in the UD documentation for Ancient Hebrew at https://universaldependencies.org/hbo/ and particularly the documentation of syntactic relations at https://universaldependencies.org/hbo/dep/.

4.1 Fixed Expressions

In the original version of the treebank, the fixed relation was only used for two constructions: עַד כִּי /‘ad ki/ and עַד אִם /‘ad ’im/, both of which function as subordinating conjunctions with the general meaning "until". In both cases the phrase is composed of the preposition עַד /‘ad/ "until" followed by a subordinating conjunction. The preposition was previously tagged SCONJ in order to comply with the requirements of the validator. However, since UD version 2.16, the validator accepts the feature ExtPos as an alternative to having a particular part of speech tag for various checks, so עַד is now marked as ADP and ExtPos=SCONJ in this construction.

We have also identified a few more fixed expressions. One of these is a combination of כִּי /ki/ "because" and אִם /’im/ "if" to form כִּי אִם /ki ’im/ "unless". This combination is both relatively frequent and also non-compositional, leading to our determination that fixed is an appropriate relation. There are a few cases where this sequence has compositional meaning, but in those cases the two conjunctions attach to different clauses, with כִּי /ki/ introducing a subordinate clause and אִם introducing a conditional which is further subordinate to that.

In addition, there is the לְבִלְתִּי /levilti/ "in order not to", which is morphologically the preposition לְ /le/ "to" followed by the noun בִּלְתִּי /bilti/, which does not appear independently. A typical construction is that in (4).



(4)

Genre	Chapters	Sentences	Approved	Accuracy	Avg. Length
instruction	14	409	326	79.7%	25.9
narrative	20	479	371	77.5%	31.5
narrative and geneaology	2	32	20	62.5%	34.5
narrative and instruction	3	89	71	79.8%	28.6
poetry	1	24	22	91.7%	23.3

Table 1: Distribution of chapters and sentences in Exodus by genre. “Sentences” is the number of sentences examined and “Approved” is the number which did not require corrections after the initial MACULA conversion. “Accuracy” is the percentage of the total sentences that were approved in that initial pass and “Avg. Length” is the mean number of syntactic words in each sentence.

Book	Total	Prior	Remaining	First Pass	Accuracy
Exodus	1151	118	1033	810	78.4%
Leviticus	820	53	767	635	82.8%
Numbers	1179	116	1063	877	82.5%
Deuteronomy	879	21	858	722	84.1%

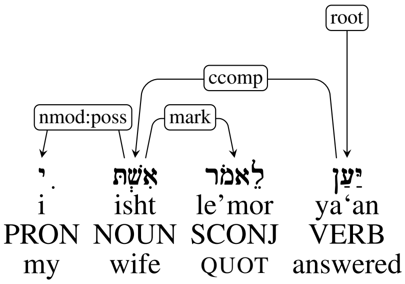
Table 2: The improvement of the parser over the course of the project. “Total” is the number of sentences in the book, “Prior” is the number of sentences validated in the course of Swanson and Tyers (2022), “Remaining” is the number of sentences that needed to be examined in the present work, “First Pass” is the number of sentences validated without adjustmnet, and “Accuracy” is the sentence-level accuracy of the parser on that book before making further updates to the rules.

“in order for me not to destroy the city”

Here the phrase לְבִלְתִּי /levilti/ precedes an infinitive verb, producing a negative purpose clause. We used `fixed` in this case since the phrase introduces a particular kind of clause and the noun which would be the head if this were a normal prepositional phrase is not used in any other context.

4.2 Quotations

Direct quotations in the text are often preceded by לֵאמֹר /le'mor/, which was originally tokenized as a single word and tagged as a subordinating conjunction, such as in (5).

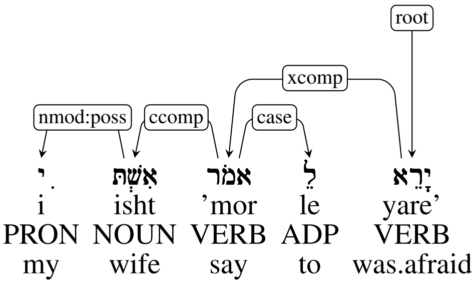


(5)

“He answered, saying ‘[she is] my wife’.”

However, in terms of morphology, this token is the preposition לְ /le/ “to” followed by the infinitive verb אֱמֹר /'emor/ “say” (infinitive verbs usually have prepositional prefixes in Hebrew, and ל is the most common one).

In fact, there are a few cases where the same surface string is a full verb, such as in (6).



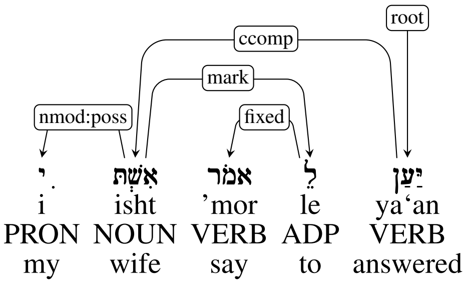
(6)

“He was afraid to say ‘[she is] my wife’.”

Here לֵאמֹר is a normal infinitive acting as a complement to the control verb יָרֵא /yare'/ “he was afraid”.

The result is that the tokenization guidelines called for prepositional prefixes such as ל to be split from their host words in all cases except the quotation marker. In light of the changes to the `fixed` and `ExtPos` guidelines discussed in the previous sections, we decided to remove the inconsistency here by tokenizing לֵאמֹר the same way everywhere and then marking the two pieces as a fixed expression. The analysis of (5) thus changes to that of (7).

(7)



“He answered, saying ‘[she is] my wife’.”

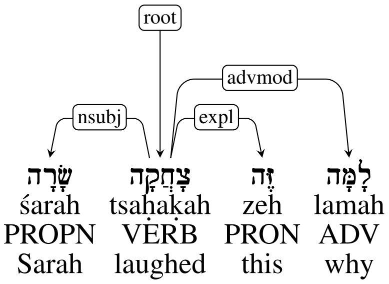
And לֵ /le/ “to” has the feature `ExtPos=SCONJ` to satisfy the validator constraint that children of `mark` relations must be subordinating conjunctions.

4.3 Expletive Pronouns

The `expl` relation is used for nominals which fill a slot in the syntactic argument structure of a clause without filling any slot in the semantic argument structure. In the original version of the treebank, this relation was unused, but we found a few cases in which we concluded it was appropriate.

In questions, an interrogative pronoun or adverb will sometimes be followed by a demonstrative pronoun for emphasis, such as in (8).

(8)



“Why did Sarah laugh?” (from Genesis 18:13)

Here the sentence would be semantically identical if זֶה /zeh/ “this” were not present. Strictly speaking, it thus does not fill one of the main argument slots of the verb, and thus `expl` is not a perfect fit, but we determined that this was likely the best option for a nominal with no semantic role.

5 Treebank Statistics

This project has roughly quadrupled the total size of the Ancient Hebrew treebank. The exact numbers are given in Table 3.

Half of the added data (Exodus and Leviticus) was included in the 2.16 release of Universal Dependencies. The other half (Numbers and Deuteronomy) will be included in 2.17.

Book	Sentences	Tokens	Words
Genesis	1,494	25,282	36,822
Exodus	1,151	20,612	29,882
Leviticus	820	14,844	21,769
Numbers	1,179	20,221	28,925
Deuteronomy	879	17,421	26,171
Ruth	85	1,564	2,297
Total	5,608	99,944	145,866

Table 3: The sizes of the texts included in the treebank. Genesis and Ruth were previously released and the rest are new.

Book	Phrases		Arcs	
Genesis	11	(11)	164	(93)
Exodus	21	(19)	191	(108)
Leviticus	2	(2)	128	(67)
Numbers	2	(2)	128	(67)
Deuteronomy	7	(7)	161	(90)
Ruth	2	(1)	12	(8)
Total	45	(42)	784	(433)

Table 4: The number of manual overrides in each book. “Phrases” is the number of overrides to the headedness of MACULA nodes and “Arcs” is the number of overrides to heads or labels in the initial dependency structure. Numbers in parentheses indicate the number of distinct sentences.

Table 4 gives the frequency of manual overrides to the parser, which occur in around 8% of all sentences. This suggests 92% as a rough upper bound on the accuracy of the parser when applied to new data.

We also evaluated how much the new process changed the data that had already been released. To do this, we took the most recent released version of Genesis and Ruth (UDv2.15) and calculated the labeled and unlabeled attachment scores (UAS and LAS) between that version and our version. In order to make them properly comparable, we used UDapi to undo the tokenization change discussed in Section 4.2. The result was a UAS of 96.51 and an LAS of 95.39, which is consistent with our experience of a limited revision that nonetheless affected a substantial portion of the sentences in the corpus.

6 Conclusion

In this paper we have presented an effort to expand the Universal Dependencies Ancient Hebrew treebank by converting an existing partial constituency treebank. This process revealed various inconsistencies and areas for improvement in the existing annotations, which have now been fixed. In addition, it has greatly reduced the amount of manual effort required to produce new trees, since the accuracy of the parser is now high enough that a typical tree can be simply validated rather than leading to further debugging of the parser.

The treebank now includes approximately a quarter of the source text, and we intend to apply this process to annotate the remainder.

Acknowledgments

Thank you to Bryce Bussert and Amir Zeldes for fruitful discussions of the annotation decisions and to Sandra Kübler and Francis Tyers for reading drafts of this paper.

References

Þórunn Arnardóttir, Hinrik Hafsteinsson, Einar Freyr Sigurðsson, Kristín Bjarnadóttir, Anton Karl Ingason, Hildur Jónsdóttir, and Steinþór Steingrímsson. 2020. A Universal Dependencies conversion pipeline for a Penn-format constituency treebank. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 16–25, Barcelona, Spain (Online). Association for Computational Linguistics.

Randall K Barry. 1997. ALA-LC romanization tables-transliteration schemes for non-roman scripts. In *Library of Congress, 1997*.

Eckhard Bick and Tino Didriksen. 2015. CG-3 — beyond classical constraint grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 31–39, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Jayeol Chun, Na-Rae Han, Jena D. Hwang, and Jinho D. Choi. 2018. Building Universal Dependency treebanks in Korean. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Clear Bible. 2022. MACULA Hebrew linguistic datasets.

Aslı Kuzgun, Oğuz Kerem Yıldız, Neslihan Cesur, Büşra Marşan, Arife Betül Yenice, Ezgi Sanıyar, Oguzhan Kuyrukçu, Bilge Nas Arıcan, and Olcay Taner Yıldız. 2021. From constituency to UD-style dependency: Building the first conversion tool of Turkish. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 761–769, Held Online. INCOMA Ltd.

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 *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

W.T. van Peursen, C. Sikkel, and D. Roorda. 2015. Hebrew text database ETCBC4b.

Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 96–101, Gothenburg, Sweden. Association for Computational Linguistics.

Daniel Swanson, Tino Didriksen, and Francis M. Tyers. 2023. WITH context: Adding rule-grouping to VISL CG-3. In *Proceedings of the NoDaLiDa 2023 Workshop on Constraint Grammar - Methods, Tools and Applications*, pages 10–14, Tórshavn, Faroe Islands. Association of Computational Linguistics.

Daniel Swanson and Francis Tyers. 2022. A Universal Dependencies treebank of Ancient Hebrew. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2353–2361, Marseille, France. European Language Resources Association.