The Hebrew Universal Dependency Treebank: Past, Present and Future

Shoval Sadde
Open University of Israel
shovalsa@openu.ac.il

Amit Seker
Open University of Israel
amitse@openu.ac.il

Reut Tsarfaty
Open University of Israel
reutts@openu.ac.il

Abstract

The Hebrew treebank (HTB), consisting of 6221 morpho-syntactically annotated newspaper sentences, has been the only resource for training and validating statistical parsers and taggers for Hebrew, for almost two decades now. During these decades, the HTB has gone through a trajectory of automatic and semi-automatic conversions, until arriving at its UDv2 form. In this work we manually validate the UDv2 version of the HTB, and, according to our findings, we apply scheme changes that bring the UD HTB to the same theoretical grounds as the rest of UD. Our experimental parsing results with UDv2New confirm that improving the coherence and internal consistency of the UD HTB indeed leads to improved parsing performance. At the same time, our analysis demonstrates that there is more to be done at the point of intersection of UD with other linguistic processing layers, in particular, at the points where UD interfaces external morphological and lexical resources.

1 Introduction

The Hebrew Treebank (HTB), initially introduced by Sima'an et al. (2001), is the first, and so far only, gold standard for morphologically and syntactically annotated sentences in Modern Hebrew. It was created with the main goal in mind to enable the development of statistical models for morphological and syntactic parsing for Hebrew, but also to facilitate linguistic investigations into the structure and distribution of linguistic Semitic phenomena. The pilot version of Sima'an et al. (2001) has been minimal — it consisted of 500 sentences, morphologically and syntactically annotated by hand. This modest start, however, defined linguistic conventions and annotation principles that would continue to affect many treebank versions derived from the HTB for many years, including the *universal dependencies* (UD) HTB version.

During these two decades, the HTB has expanded from 500 to 6221 sentences and changed several forms. The different versions of the treebank reflect different theories and formal representation types, that in turn reflect different, and sometimes contradictory, linguistic annotation principles. The reasons for these differences were sometimes practical, e.g., a new version was derived to answer an emerging technological need, and sometimes socio-academic, e.g., because different teams adopted different linguistic theories as their underlying annotation principles.

The HTB thus enabled the development of many statistical morphological and syntactic processing models (Adler, 2007; Bar-haim et al., 2008; Shacham and Wintner, 2007; Tsarfaty, 2006; Goldberg and Tsarfaty, 2008; Goldberg and Elhadad, 2009; Tsarfaty, 2010; Goldberg and Elhadad, 2010, 2011; More and Tsarfaty, 2016; More et al., In Press), but these models were trained on vastly different versions of the treebank, obeying different theories and annotation schemes, which then rendered the reported results mostly non-comparable.

Hebrew dependency parsing presents an acute version of this syndrome. Studies such as Goldberg and Elhadad (2011), Tsarfaty et al. (2012), More et al. (In Press), as well as the SPMRL shared tasks (Seddah et al., 2013, 2014), all present attachment scores on Hebrew dependency parsing. But for reporting these scores they use HTB versions that reflect distinct schemes, sometime reporting different metrics, which makes the numerical comparison between the respective results meaningless (Tsarfaty et al., 2011). This is why the UD initiative comes as a blessing, not only for the cross-linguistic parsing community but also for the Hebrew NLP community — by presenting a unique opportunity to standardize the resources and metrics used for Hebrew parsing.

Proceedings of the Second Workshop on Universal Dependencies (UDW 2018), pages 133–143
Brussels, Belgium, November 1, 2018.

Ideally, the current UDv2 version would make for such a standard Hebrew resource. Unfortunately though, many of the conversion processes since Sima'an et al. (2001) to the present UDv2 have been automatic or semi-automatic, with no point of systematic qualitative validation. This resulted in odd, and sometime plain wrong, dependency structures, with respect to the UD scheme.

In this work we take the opportunity to validate the UDv2 HTB, by manually going through the published trees, identifying systematic errors or annotation inconsistencies, and locating cases where the annotated structures contradict the UD guidelines (or spirit). We identified and corrected three main points of failure in the UD HTB: (i) the classification of argument types, deriving from the classification in the original HTB (ii) a mix-up of morphological and syntactic properties, where morphological features serve as syntactic sub-relations and vice versa, and (iii) a mix up of language-specific versus universal phenomena, where label sub-typing is exploited to indicate a supposedly language-specific phenomenon, which in fact has a designated universal label elsewhere.

Based on these corrections, we present a revised version of the HTB that we call UDv2New. We use UDv2 and UDv2New to train a morphosyntactic parser (More et al., In Press) and provide baseline results on Hebrew UD parsing, in both ideal and realistic scenarios. Comparing our Hebrew parsing results on UDv2 and UDv2New, we verify that the improvement of linguistic coherence and annotation consistency has also led to improved parsing performance. Lessons learned from our empirical analysis concern the systematic organization of natural language grammar in UD, and in particular (i) the need to standardize the interface of UD treebanks to external morphological and lexical resources, and (ii) the need to organize the form-function mapping in a language-specific vs. family-specific vs. strictly-universal relations taxonomy, within and across treebanks.

The remainder of this paper is organized as follows. In Section 2 we describe the trajectory of the HTB from its inception to UDv2. In Section 3 we present our validation process and the scheme changes we applied. In Section 4 we present raw-to-dependencies Hebrew parsing results and in Section 5 we share our future plans and lessons learned. Finally, in Section 6 we conclude.

2 Previous Work and the Trajectory of the Modern Hebrew Treebank

Following the first treebanking efforts, in English (Marcus et al., 1993), Chinese (Xue et al., 2005), and Arabic (Maamouri and Bies, 2004), and with the surge of interest in developing statistical, broad-coverage, parsing models, Sima'an et al. (2001) introduced a pilot treebanking study and a Hebrew treebank (HTB), which included 500 sentences from the Hebrew newspaper *ha'aretz*, morphologically segmented and morpho-syntactically annotated with part-of-speech tags, morphological features, and labeled phrase-structure trees. Following the annotation practices at the time, much of the tagging and labeling scheme was adopted almost as is from the UPenn Treebank (Marcus et al., 1993). However, due to its rich morphology and Semitic phenomena, several annotation decisions in the HTB diverged from these practices.

Firstly, the basic units that appear as leaves of the trees are not space-delimited tokens, but segmented units that we call morphemes.[1] Various prefixes that mark independent function words, including[2] *B* (in), *L* (to), *M* (from), *F* (that), *KF* (when) and *H* (definite article) are segmented away from their host. In addition, pronominal suffixes that appear on top of function words are also segmented away. So, the tokens *FLW* (of him), *LK* (to you), and *AITM* (with them), are segmented into *FL* (of) + *HWA* (he) , *L* (to) + *ATH* (you) , *EM* (with) + *HM* (them) respectively.[3]

The POS tags labeling scheme in the HTB includes quite a few changes from PTB, including the addition of special tags lexicalizing important functional elements in Hebrew: AT (for the accusative marker), H (the definite article), POSS (the possesive marker), and HAM (the yes/no question marker). In addition, the HTB introduces the NNT, JJT, CDT labels, marking the construct-state variants of NN, JJ, CD in the PTB, and a specific tag MOD that tags modifier words which is neither an adjective nor an adverb. On top of that, all open class POS tags as well as auxiliaries have been marked for their inflectional features (gender, number, person, time), yielding in total hundreds of possible fine-grained POS categories.

[1] In the UD terminology these are called syntactic words.

[2] We use the transliteration of Sima'an et al. (2001), and describe the transliteration in our supplementary material.

[3] Note that while combining *prefixes* is fairly straightforward, *suffixes* are fused to hosts in idiosyncratic and non-systematic morpho-phonological processes.

The syntactic labels in the phrase structure trees of the HTB were adopted from the Penn Treebank (PTB) almost as is, with the addition of a PREDP label for marking verbless predicates. The syntactic trees themselves looked superficially like the PTB but they differ in several aspects. Due to word-order freedom at the clause level, S-level categories present a flat structure, where the positions of the arguments do not entail anything about their grammatical function. The HTB provided 3 types of manually verified function tags to indicate such functions: SUBJect, OBJect, and COMplement, the latter marking obligatory arguments of the verb. Finally the HTB defined three types of null elements: *T* marking phonologically empty anaphors, *PRO* for pro-drop subjects, and *NONE* for elliptical elements.

The work of Guthmann et al. (2008) extended the HTB to 6501 sentences, in a manually-validated automatic process.[4] During this process they further added a systematic marking of *mother-daughter* dependencies. That is — due to feature-spreading in Hebrew, morphological features of phrases may be contributed by different daughters, and not necessarily via a single *head*. So they marked each daughter with the role it plays in determining its mothers' features (gender, number, tense, etc.). Using these feature-based dependencies, they performed feature-percolation from daughter to mother, so that phrasal nodes are also marked with their morphological signatures.[5]

Still, the phrase-structure trees yielded by HTB-trained parsers were not useful for downstream applications in Hebrew. This is because Hebrew is a relatively-free word order language, where the position of a constituent does not entail its grammatical function or semantic role. This in particular precludes the use of well known 'head tables' for selecting a single head and deriving labeled and unlabeled dependencies. To overcome this, Tsarfaty (2010) devised a set of rules based on the *daughter-dependencies*, *function tags* and *empty elements*, to automatically derive the *relational-realizational* (RR) version of the HTB. In the RR HTB, each node is marked with its relational network (an unordered set of grammatical functions) mapped to the ordered syntactic constituents. The RR HTB retained the morphological conventions and core non-core distinction of the original HTB.

[4]Excluding repeated sentences, we have 6221 trees.

[5]This marking did not specify a single *head* since a mother node could have multiple *daughter-dependencies*.

In a parallel effort, and with the surge of interest in dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007),[6] Goldberg (2011) automatically converted the HTB into its first, unlabeled, dependency version. The automatic conversion procedure assumed that heads are functional rather than lexical. As a result, the coordination marker would head coordination structures, the accusative marker would head direct object phrases, and so on. On top of that, in order to remain compatible with the wide-coverage lexicon of Itai and Wintner (2008), this version of the HTB adopted the POS tags scheme of Adler (2007), rather than the POS tags of Sima'an et al. (2001)

Based on this version, Goldberg and Elhadad (2009) presented the first Hebrew dependency parsing results, only *unlabeled attachment scores* (UAS) at this point. Here too, as with the phrase-structure trees, it was impossible to devise an external procedure that would infer dependency labels for the unlabeled arcs — and there were no labeled dependencies to train such a labeler on.

At that point, where the need for Hebrew labeled dependencies had become pressing, Tsarfaty (2013) presented the Unified-Stanford Dependencies (Unified-SD) version of the HTB, extending the *Stanford dependencies* (SD) scheme to cover both morphological and syntactic phenomena. Similar to SD, U-SD assumed a labeling hierarchy, with several changes: the hierarchy now included branches for *head-types (hd)*, *dependency types (dep)*, and *functional types (func)*. In particular, dependencies in the *func* branch mark syntactic functions that are in fact interchangeable with morphology, when considering these functions from a typological perspective.

Tsarfaty used the U-SD labels to edit three versions of the HTB: (i) to mark the original phrase-structure trees in the HTB with the labels as dash-features, (ii) to relabel the relational networks in RR trees with U-SD labels, and (iii) to derive a *labeled* dependencies version of the HTB. As with the unlabeled dependencies of Goldberg, the U-SD HTB assumed functional heads across the board, and the POS tags layer was again changed to comply with the wide-coverage lexicon (HEBLEX) of Itai and Wintner (2008). The labeled dependencies treebank of U-SD then provided the Hebrew section of the SPMRL shared tasks (Seddah et al., 2013, 2014).

[6]Notably, Hebrew did not take part in these shared tasks.

3 The Hebrew UD Treebank

3.1 Overview

The RR version of the Unified-SD HTB provided the basis for automatically converting the Hebrew trees into UDv1 trees. The UD HTB assumes the same segmentation principles as the first edition of the HTB, segmenting off prefixes and suffixes, with the addition of splitting off genitive pronominal clitics from nouns.

Goldberg and Tsarfaty (2014) devised an automatic process that chooses a *lexical head* in each relational network of each constituent in the RR treebank. They also mapped the fine-grained POS categories to the coarse-grained UPOS categories in UD, and remaining POS distinctions in HebLex (HebBinyan, construct-states, etc.) are stored in FEATS. The label set of U-SD was automatically mapped to UD, and relations from U-SD outside of UD were kept as relation:subtype.

The conversion of UDv1 to UDv2 was also done automatically, by augmenting the script of Goldberg and Tsarfaty (2014). Points of failure of the UDv1 version of the HTB to comply with UDv2 were identified by aiming to locate skewed distributions of tags or labels, and they were corrected in the conversion script on a case by case basis. This process has stopped when the treebank complied with the UDv2 validation script. The converted HTB is documented on the UD webpage.[7]

3.2 Validation

The present version of UDv2 thus results from a sequence of automatic and semi-automatic conversions on the trees of Guthmann et al. (2008). In order to validate the current UDv2 trees, we reviewed the list of UD POS tags, relation labels and features, and for each of these items we identified the dependency structures in the HTB *dev set* that contain them. At this point, for each item, a linguist characterized the role such item actually fulfills in the Hebrew grammatical structures, (as opposed to the role it was designed to fulfill in the UD scheme).

During this process the linguist documented errors and inconsistencies that were found, either between the realistic use of a function in the UDv2 HTB and the UDv2 guidelines, or simply attesting insufficient or incorrect coverage of the linguistic structure that this particular label, tag or feature is supposed to describe.

This validation process[8] was conducted on the entire HTB UDv2 *dev set*[9] and it was followed by a sequence of discussions in which our research team, consisting of two linguists, two NLP specialists, and a senior NLP researcher, discussed possible solutions for each error. The discussions were focused on explicitly assessing the merits of each solution alternative according to the six criteria of the Mannings Law. That is: linguistically adequate, typologically adequate, suitable for rapid, consistent annotation, suitable for parsing with high accuracy, easily comprehended by non-linguists, and provides good support for downstream NLP tasks.[10] After narrowing down the list of adequate solutions, the final decision about which revisions to make leaned on their importance and feasibility. For example, a very important, yet easily executable revision was to simply replace all instances of prepositional *iobj* with *obl*. Just as important, but far more complex, was to switch between a head and a dependent in the case of structures containing *auxiliaries* (e.g., modals, as we illustrate shortly).

All revisions were made with the python Pandas package, and they were applied to all, *dev*, *train* and *test*, sets. Revisions were made with respect to linguistic patterns that refer to existing labels, tags or features, with no consideration of any particular (lexicalized) Hebrew words. Furthermore, we refrained from manual changes of specific errors, considering that their source might be a vaster problem, to be dealt with in the future. As an example for simple edits, consider adding a label compound:affix. For this, all rows containing the feature 'Prefix=Yes' had to be retrieved, and the label was changed to *compound:affix*. As a more complex case, consider the case involving modality mentioned above. Here, all rows with the *xcomp* label were retrieved. For each row, if the head had a morphological feature 'VerbType=Mod', the head's label was relabeled 'aux',

[7] http://universaldependencies.org/treebanks/he_htb/index.html.

[8] It is important to note that our analysis proceeded label-by-label, and tag-by-tag, which is a faster process than going through the treebank trees one-by-one. But it also bears the risk of missing out rare peculiarities and singleton errors.

[9] In this work we primarily aimed to correct the main issues that appeared across the board, rather than tackling idiosyncratic or incidental errors. So, observing the *dev set* was enough, as it well reflects the main linguistic phenomena in the language.

[10] en.wikipedia.org/wiki/Manning%27s_Law

ID	FORM	UPOSTAG	FEATS	HEAD	DEPREL
6	IS	AUX	VerbType=Mod	0	root
7	LPNWT	VERB	VerbForm=Inf	6	xcomp
6	IS	AUX	VerbType=Mod	7	aux
7	LPNWT	VERB	VerbForm=Inf	0	root

Table 1: Turning Auxiliary Heads into Modal Dependents. The top pair represents the UDv2 structure, the lower pair represents the UDv2New revision.

the row itself was relabeled with the original label of the head, and the numbers were changed respectively in the 'HEAD' column (see Table 1).

3.3 Revision

Adhering to UDv2 guidelines provided an opportunity to make a consistent decision about topics under debate, and to generally revise inconsistencies in the system. Our revisions typically fall under one of the following three categories: predicate/argument types distinctions (3.3.1), morphological vs. syntactic distinctions (3.3.2), and Hebrew-specific vs. universal distinctions (3.3.3).

3.3.1 Predicate Argument Types Distinctions

Open Clausal Complements. In the UDv2 HTB, predicative complements were labeled advmod when adjectival. Following the UDv2 guidelines, we label them xcomp, as they are subordinated predicates, after all, even if not verbal.

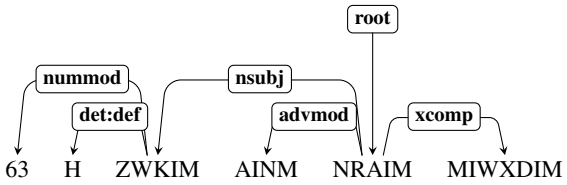

Figure 1: UDv2New treatment of predicative complements as xcomp rather than advmod. The adjective *MIWXDIM* 'special' is a complement of *NRAIM* 'look'

"63 H-ZWKIM AINM NRAIM
63 DET-winner.PL.M be.NEG.PL.M look.PL.M
MIWXDIM."
special.PL.M

'The 63 winners do not look special'

Argument iobj vs. obl. Some UD definitions stand in clear contrast with the canonical syntactic analysis of Hebrew. Perhaps the most salient case is of core arguments. The canonical view of Hebrew core arguments (Coffin and Bolozky (2005) p. 290) is of a direct object, marked by an accusative case when definite, and an indirect object, marked by an oblique case marker when a pronoun, and preceded by a preposition when common or proper noun. UDv2 dedicates an iobj (indirect object) relation to secondary core arguments which are not preceded by prepositions, and arguments which do follow a preposition are labeled obl, whether core or non-core. We revised the labels accordingly.

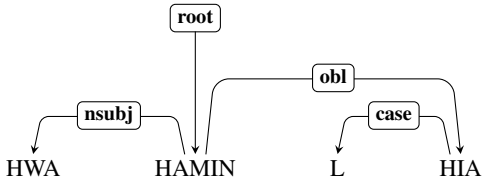

Figure 2: The noun *HIA*, following the preposition *L*, although being a core argument of the verb *HAMIN*, is labeled obl in UDv2New, as opposed to iobj in previous versions.

"HWA HAMIN L-HIA."
he.3SG.M believe.Tsg.PST DAT-she.3SG.F

'He believed her'

Predicate types: the case of auxiliaries As part of the shift towards a lexically-driven analysis, structural changes were made to sentences containing auxiliary elements and copulas. There are three main sets of these: (i) Auxiliary elements marking modality, (ii) Auxiliary verbs which mostly mark habituality, but occasionally participate in negation or tense inflection when the predicate has no past/future form, and (iii) Positive or negative copulars.

Modals do not constitute any uniform syntactic class in Hebrew, and there is an ongoing debate as to the POS of each modal expression (cf. Netzer et al. (2007)). In line with Netzer et al's conclusion, these are tagged as AUX in the UD HTB. In UDv2, the modal served as the head of the clause, while the following predicate was labeled xcomp, as it is consistently realized in Hebrew in infinitive form. As of UDv2New, those modals which are tagged as AUX are also labeled aux, and the subsequent predicate receives the label which was attributed to the modal. See Table 1.

In the opposite direction, auxiliary verbs, such as the ones in sets ii and iii were tagged as VERB. As the UDv2 scheme dedicates an AUX tag to function words in auxiliary functions even when they are verbs, we changed them to AUX as well

in UDv2New. Finally, consistency across sets ii and iii was achieved by unifying the labeling of copular verbs as cop regardless of their inflection, whereas previous versions labeled past and future inflections of copular verbs as aux.

3.3.2 Morphology vs. Syntax

Eliminating acl:inf to acl. The automatic conversion to UD has kept fine-grained labels as sub-relations, resulting with the language-specific label acl:inf. Since the UD guidelines permit infinitive structures in acl, it is unnecessary to mark infinity as a sub-relation. Moreover, all cases of acl:inf bear the feature 'VerbForm=Inf'. So eliminating the morphological feature inf from the sub-relation acl:inf does not lead to any information loss.

"NSIWN-W H-AXRWN FL MILR
attempt.SG.M-POSS DET-last POSS Miller
LHFIG KSPIM"
get.INF money.PL

'Miller's last attempt to get money'

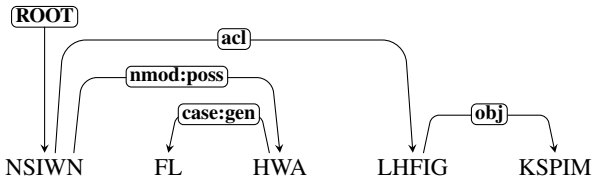

Figure 3: The label acl:inf was reduced into simply acl for the infinitive verb *LHFIG* (to get)

Adding compound:affix This new relation is dedicated to non-standalone words, which function semantically like affixes, but syntactically surface as separate words, at times separated by a hyphen and in others by white-space. A subset of these words are loan words (mainly from English, like 'non', 'multi' etc.) where originally they surface syntactically as affixes. In UDv2 these items were marked by the feature Prefix=Yes. However, since they mark a certain type of Hebrew compounds, we used sub-typing to indicate it.[11] In "KLL-EWLMIT" for example, *KLL* 'uni-' is semantically a prefix to *EWLMIT* 'worldly', but in Hebrew the two are separate words.

3.3.3 HTB-to-UD: language-specific representation with relation:subtype

As UD aspires to present a set of tags which are relevant to as many languages as possible, naturally many language-specific phenomena are left unanswered. To allow representation of these, the UD scheme allows for sub-relations in the form of relation:subtype, as exemplified above. However, although originally aiming toward coverage of language-specific phenomena, this structure can be frequently seen as a subtype of relation which is present in many languages (e.g. nsubj:pass, which is in use for subjects of passive sentences - not unique to any one language or even a family of languages). In our revision to adhere to UDv2 guidelines, we tried as much as possible to narrow the use of relation:subtype to Hebrew-specific phenomena, eliminating any hierarchical structure of dependency relations. As a result, the following subtypes were reduced to their parent relation: (i) det:quant, originally marking an arbitrary subset of existential quantifiers, was reduced to simply det , and (ii) advmod:phrase, originally marking multi-word adverbials, were re-structured as advmod+fixed, in line with the UD guidelines for multi-word-expressions.

From conj:discourse to parataxis An interesting case is with labels not used at all in the older versions of the UD HTB, while language-specific labels stand to mark their function. The UD label parataxis, for instance, describes a relation between two (or more) sentences which are syntactically independent (i.e. do not stand in subordination or conjunction relation to one another), but are thematically connected, and consequently punctuated as the same sentence. Previously, this relation was labeled in the HTB as conj:discourse, simply classifying conjunctions that are not explicitly marked as of type discourse. In our revised version, we comply with UD guidelines and label this relation 'parataxis'.

From PART to ADP The accusative and possessive case markers in Hebrew, AT and FL respectively, are realised as separate tokens, as opposed to some other case markers, which prefix the following nouns. Furthermore, a possessive case marker may also morphologically suffix the noun, whether instead of or in addition to the above-mentioned particle. In older versions of HTB, while preposition (whether standalone or not) were tagged IN, the accusative case marker was tagged AT and the possessive case marker was tagged POSS. As a result, automatic conversions led to converting IN to ADP across the board,

[11] All analyses are visualized in the supp. materials.

while AT and FL were converted into PART. As there is no real difference between AT and FL and prepositions according to the UDv2 scheme, and as they are in no way particles, we converted them into ADP.

3.4 Unsolved Issues

Some inconsistencies in the treebank were spotted but not yet fixed as their automatic full retrieval and change is more complicated [12]. For example, *it*-extraposition construction is represented in UDv2 by a combination of nsubj and ccomp or advcl, but should be a combination of expl+csubj, as defined in the guidelines (see example 9 in the supplements).

In another case, lack of congruence was found between our treatment of participles and Adler et al. (2008). The feature of VerbForm=Part marks both deverbal nouns and present tense clauses, as in the following sentence.

> "EFRWT ANFIM MGIEIM
> ten.PL.F person.PL.M arrive.PTCP
> M-TAILND L-ISRAL"
> from-Thailand to-Israel
>
> 'Tens of people come from Thailand to Israel.'

Hebrew makes various uses of the dative case, some of them fulfill purely discursive functionality (Borer and Grodzinsky, 1986). The current representation of the dative case marker in UDv2New does not give way to all possible meanings, including experiencer dative (Berman, 1982) as opposed to ethical dative, the regular dative where the dative argument is subcategorized by the verb. The current UDv2 guidelines do not distinguish between the different types of dative, so an educated decision must be made locally as for how to tell them apart.

- IS **LW** HMIWMNWT HNXWCH BFBIL LHIWT MWFL
  'He has what it takes to be a governer.'
- HRAF HIHWDI MMCIA **LNW** PTNTIM.
  'The Jewish mind invents **(us)** patents'
- HW QRA **LH** FQRNIT
  'He called **her** a liar.'

[12] For reasons of brevity we do not discuss all of them in this work.

4 HTB Experiments and Parsing Results

Goal: We wish to examine the empirical impact of our effort to correct the treebank and retain linguistic (as well as cross-treebank) coherence in its annotation scheme. Indeed, ease of parsing should not be the indication for selecting one scheme over another, but the hypothesis is that, within *one and the same set of guidelines*, a version that presents better coherence and consistency will also be more suitable for statistical training and will yield better results.

Settings: To gauge the effect of our revision we conducted two sets of experiments: one with the HTB UDv2 version used in the recent shared task, and another our revised UDv2New. We use the syntactic evaluation script provided by the CoNLL shared task 2018. We train on the portion defined as train set and report results on the dev set. For training and parsing we used *yap*,[13] a transition-based morphosyntactic parser written in go, which includes a morphological analyzer, a morphological disambiguator, and syntactic parser. In previous work *yap* was shown to obtain state of the art results on Hebrew parsing using the SPMRL version of the treebank (More et al., In Press). Here we report its performance on the UD HTB.

Scenarios: Because of its rich morphology and orthographic convention to attach or fuse adpositions and pronominals onto open-class categories, there is severe ambiguity in the morphological analysis of the Hebrew input tokens. This is further magnified by the lack of diacritics in Hebrew written texts. Hence, it is unknown upfront how many morphemes (in the HTB terminology) or syntactic words (in the UD terminology) are in the space-delimited tokens. We examine two kinds of scenarios:

- *ideal:* assuming gold morphological analysis and disambiguation given by an oracle.
- *realistic:* assuming automatically predicted morphological analysis and disambiguation.

We use *yap* for predicting morphological analysis (MA) and morphological disambiguation (More, 2016), and we contrast the use of a data-driven lexicon *baselinelex* with an external broad-coverage lexicon *HebLex*. To gauge the effect of the lexical

[13] https://github.com/habeanf/yap

coverage of the morphological resource, we contrast each variant with an *infused* scenario, where the correct analysis is injected into the lattice. Note that the input in the infused cases is still high as there are many MA alternatives. However, the correct morphological disambiguation is guaranteed to be one of the morphological MA provided to the system as input.

Results: Table 2 shows the parsing results in an ideal scenario, assuming gold morphology. Here we see that there is a consistent improvement for all metrics. This supports our conjecture that a more consistent and coherent annotation of the treebank will benefit parsing, and it corroborates a wider conjecture, that, when it comes to supervised learning, the quality of the annotated data is as important as the learning algorithm (and maybe more important).

Table 3 shows the parsing results in realistic scenarios, where we assume automatically predicted morphological analysis and disambiguation. As expected, the results substantially drop relative to the ideal scenario. Also expected is the result that assuming an external broad-coverage lexicon substantially improves the results relative to a data-driven lexicon learned from the treebank. The result that seems less expected here is that, as opposed to the ideal scenario, we see no improvement in the results of UDv2New relative to UDv2. For some of the metrics the results slightly drop.

This drop could be either due to parser errors, or due to the lack of lexical coverage of the lexicon with respect to our revised UDv2New scheme. To test this, we execute an *infused* scenario where the morphological analysis lattices are guaranteed to also include the correct analysis. Here we see a substantial improvement for both types of lexica, on all the different metrics, for the UDv2New version. This result suggests that the drop has indeed been due to the insufficient lexical coverage of the resources, or due to mismatches between the lexicon and the new scheme. As far as the statistical components for morphological and syntactic analysis and disambiguation go, the revised version helps the parser obtain better disambiguation, in line of our results in the gold experiments.

5 Discussion and Lessons Learned

The original HTB (Sima'an et al., 2001; Guthmann et al., 2008) has seen many revisions all of which executed automatically, or semi-

UDv2 Shared-Task Version	LAS	MLAS	BLEX
he_htb-ud-dev-*yap*-gold	79.51	72.76	47.76
UDv2New Revised Version	LAS	MLAS	BLEX
he_htb-ud-dev-*yap*-gold	81.24	75.58	50.16

Table 2: Parsing Results of the HTB *dev* set for UDv2 vs UDv2New, in an *ideal* parsing scenario assuming GOLD morphology.

UDv2 Shared-Task Version	LAS	MLAS	BLEX
he_htb-dev-*yap*_baselinelex	51.99	37.62	29.50
he_htb-dev-*yap*_heblex	60.71	39.53	33.82
he_htb-dev-*yap*_baselinelex-infused	58.45	43.70	32.94
he_htb-dev-*yap*_heblex-infused	71.19	61.08	41.71
UDv2New Revised Version	LAS	MLAS	BLEX
he_htb-dev-*yap*-baselinelex	52.42	38.08	30.32
he_htb-dev-*yap*_heblex	60.34	37.95	34.71
he_htb-dev-*yap*-baselinelex-infused	58.54	44.06	33.30
he_htb-dev-*yap*_heblex-infused	73.66	64.73	44.32

Table 3: Parsing Results of the HTB *dev* set for UDv2 vs UDv2New, in a *realistic* parsing scenario assuming PREDICTED morphology. We compare a data-driven *baseline* lexicon with an external lexicon, *heblex*, and we contrast *uninfused* or *infused* setting for both

automatically. Our endeavor here has been to manually verify the current version of the UD HTB resulting analyses, and to correct lingering errors. Apart from being linguistically justified, this process has proven to be also empirically valuable, as indeed this revision has led to a improvement in parsing results.

Much work is still needed in order to bring the level of performance to be adequate for downstream applications, in particular in realistic scenarios. We conjecture that in order to obtain decent performance, the work on the treebank should be complemented by adapting language-specific lexica to the set of guidelines for word segmentation and for representing morphology, as defined by UD. Even when external lexica assumes the same labeling scheme as UD, gaps between the theories underlying the development of these resources could lead to lack of coverage that substantially harms parsing performance.

Additional lessons learned from our manual verification process have to do with the organization of morphological features and syntactic subtypes within the HTB and in the UD treebanks collection in general. In the HTB UDv2, there appeared to be a mix between the linguistic notions expressed using these two mechanisms. For example, subtypes were sometimes used to indicate morphological features (see the case for acl:inf)

while the features column is exploited to express syntactic properties. We argue that clearer guidelines are needed in the general UD scheme, instructing directly what kind of linguistic information should go where, by which formal means.

Furthermore, it seems to us that the language-specific mechanisms are exploited for expressing phenomena that could potentially be cross-linguistic, or at least shared by a language family. An example to this is the feature *HebBinyan* in the UD HTB, which stores the value of the morphological template of the verb. The phenomenon of *Binyan* (a root-template construction) is clearly not Hebrew specific — in fact all Semitic languages have *Binyanim* (morphological constructions) in their grammar, so we see no good reason for not unifying this feature across the Semitic sub-family. Same goes with marking construct state nouns, a phenomenon that extends beyond Semitic languages, and is currently marked differently in each language (Hebrew, Arabic, Persian, etc.).

We propose that the next major revision of the UD treebank scheme could ideally focus on *the universal organization of the grammar*, and will center around these themes:

- *subtypes:* A universal inventory and management of the sub-label system which will define what linguistic phenomena can count as subtype of a label, and will maintain cross-linguistic consistency in its use for shared phenomena.
- *features:* A universal inventory and management of features which will define what can count as a feature, and will foster cross-linguistic reuse.
- *lexical resources:* For languages that have external lexica, especially in the case of morphologically rich and resource scarce languages, an effort is needed to verify that the labeling scheme theoretical guidelines underlying lexica are harmonized with the UD guidelines. Such lexica can be made available via the CoNLL-UL format (More et al., 2018) to benefit the entire UD community.
- *semantic applications:* in addition to aligning lexical resources, it is important to advance the usability of UD in down-stream application scenarios, by making available the additional layer of *enhanced* dependencies.

6 Conclusion

In this paper we describe the long and multi-phased process of coming-into-existence of the Hebrew version of the HTB. Most of the process has consisted of automatic conversions between different schemes. In this work we manually verified the recent UD HTB version and corrected lingering errors. The revised version is more linguistically and cross-linguistically consistent and obtains better parsing results in scenarios that are not dependent on the coverage of external lexica. Our future plans include a comprehensive revision of the lexical and morphological resources associated with the UD scheme, to improve the empirical parsing results in realistic scenarios, and the addition of enhanced dependencies, which would be more adequate for downstream semantic tasks.

Acknowledgments

We thank the ONLP team at the Open University of Israel for fruitful discussions throughout the process. We further thank two anonymous reviewers for their detailed and insightful comments. This research is supported by the European Research Council, ERC-StG-2015 scheme, Grant number 677352, and by the Israel Science Foundation (ISF), Grant number 1739/26, for which we are grateful.

References

Meni Adler. 2007. *Hebrew Morphological Disambiguation: An Unsupervised Stochastic Word-based Approach*. Ph.D. thesis, Ben-Gurion University of the Negev, Beer-Sheva, Israel.

Meni Adler, Yael Dahan Netzer, Yoav Goldberg, David Gabay, and Michael Elhadad. 2008. Tagging a hebrew corpus: the case of participles. In *LREC*. Citeseer.

Roy Bar-haim, Khalil Sima'an, and Yoad Winter. 2008. Part-of-speech tagging of Modern Hebrew text. *Natural Language Engineering*, 14(2):223–251.

Ruth A Berman. 1982. Dative marking of the affectee role: Data from modern hebrew.

Hagit Borer and Yosef Grodzinsky. 1986. Syntactic cliticization and lexical cliticization: The case of hebrew dative clitics in the syntax of pronominal clitics. *Syntax and semantics*, 19:175–217.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL-X*, pages 149–164.

Edna Amir Coffin and Shmuel Bolozky. 2005. *A reference grammar of Modern Hebrew*. Cambridge University Press.

Yoav Goldberg. 2011. *Automatic syntactic processing of Modern Hebrew*. Ben Gurion University of the Negev.

Yoav Goldberg and Michael Elhadad. 2009. Hebrew dependency parsing: Initial results. In *Proceedings of the 11th International Conference on Parsing Technologies*, IWPT '09, pages 129–133, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yoav Goldberg and Michael Elhadad. 2010. Easy first dependency parsing of modern hebrew. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, SPMRL '10, pages 103–107, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yoav Goldberg and Michael Elhadad. 2011. Joint Hebrew segmentation and parsing using a PCFGLA lattice parser. In *Proceedings of ACL*.

Yoav Goldberg and Reut Tsarfaty. 2008. A single framework for joint morphological segmentation and syntactic parsing. In *Proceedings of ACL*.

Yoav Goldberg and Reut Tsarfaty. 2014. Htb to ud conversion.

Noemie Guthmann, Yuval Krymolowski, Adi Milea, and Yoad Winter. 2008. Automatic annotation of morpho-syntactic dependencies in a modern hebrew treebank. *LOT Occasional Series*, 12:77–90.

Alon Itai and Shuly Wintner. 2008. Language resources for hebrew. *Language Resources and Evaluation*, 42(1):75–98.

Mohamed Maamouri and Ann Bies. 2004. Developing an arabic treebank: Methods, guidelines, procedures, and tools. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Semitic '04, pages 2–9, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

Amir More. 2016. *Joint Morpho-Syntactic Processing of Morphologically Rich Languages in a Transition-Based Framework*. Ph.D. thesis, The Interdisciplinary Center, Herzliya.

Amir More, Özlem Çetinoğlu, Çağri Çöltekin, Nizar Habash, Benoît Sagot, Djamé Seddah, Dima Taji, and Reut Tsarfaty. 2018. Conll-ul: Universal morphological lattices for universal dependency parsing. In *11th Language Resources and Evaluation Conference*.

Amir More, Amit Seker, Victoria Basmova, and Reut Tsarfaty. In Press. Joint transition-based models for morpho-syntactic parsing: Parsing strategies for mrls and a case study from modern hebrew. In *Transactions of ACL*.

Amir More and Reut Tsarfaty. 2016. Data-driven morphological analysis and disambiguation for morphologically rich languages and universal dependencies. In *Proceedings of COLING*, pages 337–348. The COLING 2016 Organizing Committee.

Yael Dahan Netzer, Meni Adler, David Gabay, and Michael Elhadad. 2007. Can you tag the modal? you should. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, SEMITIC@ACL 2007, Prague, Czech Republic, June 28, 2007*, pages 57–64.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. 2014. Introducing the spmrl 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 103–109, Dublin, Ireland. Dublin City University.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, Washington, USA. Association for Computational Linguistics.

Danny Shacham and Shuly Wintner. 2007. Morphological disambiguation of Hebrew: A case study in classifier combination. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 439–447, Prague, Czech Republic. Association for Computational Linguistics.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and N. Nativ. 2001. Building a tree-bank of Modern Hebrew text. *Traitment Automatique des Langues*, 42(2).

Reut Tsarfaty. 2006. Integrated morphological and syntactic disambiguation for modern Hebrew. In *Proceedings ACL-CoLing Student Research Workshop*, pages 49–54, Stroudsburg, PA, USA. ACL.

Reut Tsarfaty. 2010. *Relational-realizational parsing*. Ph.D. thesis.

Reut Tsarfaty. 2013. A unified morphosyntactic scheme for stanford dependencies. In *Proceedings of ACL*.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012. Joint evaluation of morphological segmentation and syntactic parsing. In *Proceedings of ACL*, ACL '12, pages 6–10, Stroudsburg, PA, USA.

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

Naiwen Xue, Fei Xia, Fu-dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Nat. Lang. Eng.*, 11(2):207–238.