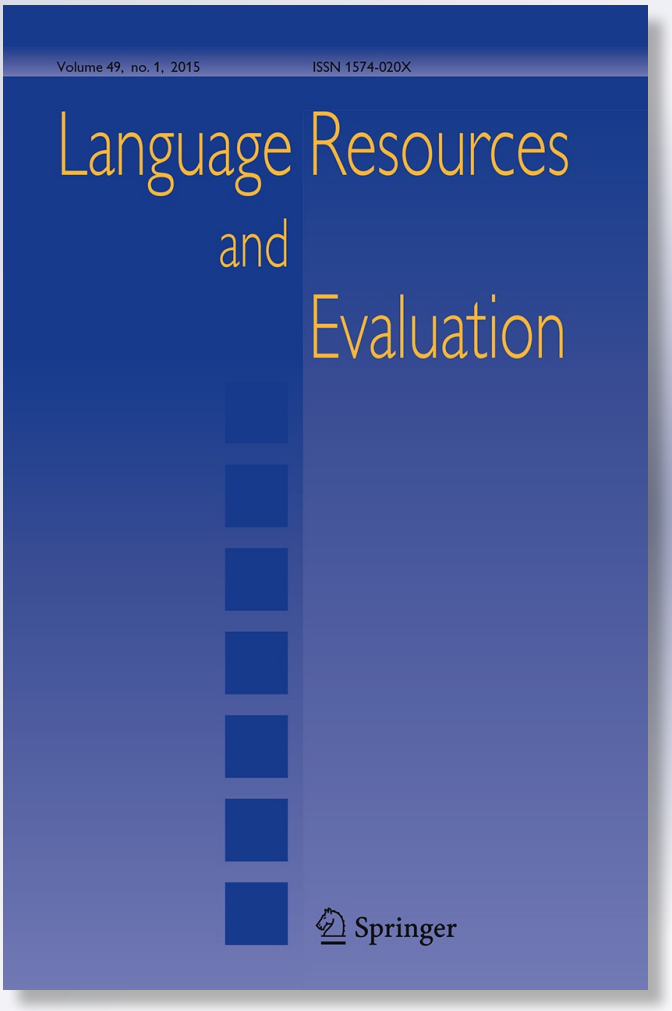
# *Parsing Hebrew CHILDES transcripts*

**Shai Gretz, Alon Itai, Brian MacWhinney, Bracha Nir & Shuly Wintner**

**Language Resources and Evaluation**

ISSN 1574-020X
Volume 49
Number 1

Lang Resources & Evaluation (2015)
49:107-145
DOI 10.1007/s10579-013-9256-x




Springer

Lang Resources & Evaluation (2015) 49:107–145
DOI 10.1007/s10579-013-9256-x

ORIGINAL PAPER

# Parsing Hebrew CHILDES transcripts

**Shai Gretz · Alon Itai · Brian MacWhinney · Bracha Nir · Shuly Wintner**

Published online: 22 November 2013


**Abstract** We present a syntactic parser of (transcripts of) spoken Hebrew: a dependency parser of the Hebrew CHILDES database. CHILDES is a corpus of child–adult linguistic interactions. Its Hebrew section has recently been morphologically analyzed and disambiguated, paving the way for syntactic annotation. This paper describes a novel annotation scheme of dependency relations reflecting constructions of child and child-directed Hebrew utterances. A subset of the corpus was annotated with dependency relations according to this scheme, and was used to train two parsers (MaltParser and MEGRASP) with which the rest of the data were parsed. The adequacy of the annotation scheme to the CHILDES data is established through numerous evaluation scenarios. The paper also discusses different annotation approaches to several linguistic phenomena, as well as the contribution of morphological features to the accuracy of parsing.

**Keywords** Parsing · Dependency grammar · Child language · Syntactic annotation

S. Gretz · A. Itai
Department of Computer Science, Technion, Haifa, Israel
e-mail: shaigretz@gmail.com

A. Itai
e-mail: itai@cs.technion.ac.il

B. MacWhinney
Department of Psychology, Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: macw@cmu.edu

B. Nir
Department of Communication Disorders, University of Haifa, Haifa, Israel
e-mail: bnir123@gmail.com

S. Wintner (✉)
Department of Computer Science, University of Haifa, Haifa, Israel
e-mail: shuly@cs.haifa.ac.il

## 1 Introduction

Child–adult interactions are a basic infrastructure for psycholinguistic investigations of child language acquisition and development. The corpora available through the CHILDES database (MacWhinney 2000), consisting of spoken transcripts in over twenty five languages, have been an indispensable source for this line of research for the past thirty years. This database is unique in that it provides its users not only with raw data from monologic, dyadic, and multi-party interactions (all following a unified and established transcription scheme) but also with tools for the application of theoretically-motivated and well-tested analyses. The most developed feature of the system is the MOR program, an automatic Part-of-Speech tagger with functionality in thirteen varied languages—including Cantonese, Japanese, and Hebrew (in addition to Romance and Germanic languages). More recent work has focused on the automatic syntactic parsing of these data, most notably with the parser that was developed by Sagae et al. (2010) for the English section of CHILDES.

The current paper reports on a similar endeavor, focusing on automatic syntactic parsing of (a subset of) the *Hebrew* section of the CHILDES database. This subset includes two corpora with full, reliable, and disambiguated Part-of-Speech and morphological annotation (Albert et al. 2014): the Berman longitudinal corpus (Berman and Weissenborn 1991) and the Ravid longitudinal corpus. Each of these corpora includes naturalistic data collected on either a weekly or a monthly basis from six Hebrew-speaking children and their care-takers, yielding approximately 110,000 utterances in total (Nir et al. 2010).

Similarly to the syntactic structure that Sagae et al. (2010) induce on the English section of CHILDES, the parser for Hebrew makes use of *dependency relations*, connecting the surface tokens in the utterance through binary, asymmetric relations of *head* and *dependent*. In this we join a growing body of automatic parsers that rely on dependency-based syntactic representations (Kübler et al. 2009). These representations are particularly adequate for the annotation of Hebrew child–adult interactions for a number of reasons. Hebrew is a language with relatively complex morphology and flexible constituent structure; morphologically-rich languages such as Arabic have been successfully analyzed using a dependency-based parser (Hajič and Zemánek 2004). More importantly, a dependency-based parser relies on the explicit specification of dependencies (i.e., the *name* of the functional relation), which provides a representation of the relations assumed to be learned by the child, unlike parsers based on constituent structure, in which such information is implicit and requires further derivation (Ninio 2013). Finally, this choice makes our annotation more consistent with the syntactic structure of the English section of CHILDES, which may be useful for cross-linguistic investigations.

This work makes several contributions. First, we defined the first dependency annotation scheme for spoken Hebrew (Sect. 4). Second, we developed a parser for the Hebrew section of the CHILDES database, annotating utterances with syntactic dependency relations (Sect. 5). The parsed corpus will be instrumental to researchers interested in spoken Hebrew, language acquisition and related fields. We evaluated our parser in different scenarios (Sect. 6); the results demonstrate that

the parser is both accurate and robust. Furthermore, we experimented with alternative annotation methods of several constructions, as well as with different approaches to tokenization of the input and with different sets of morphological features (Sect. 7), showing the impact of such choices on the accuracy of parsing. The parser, the manually annotated corpus, and the automatically parsed corpora are all available for download from the main CHILDES repository.

## 2 Related work

To the best of our knowledge, the only parser of Hebrew was introduced by Goldberg (2011). It focuses on *written* Hebrew and has two versions, one that produces constituent structures and one that generates dependency relations. The scheme presented by Goldberg (2011) was generated from the original format of a constituent structure treebank (Sima'an et al. 2001; Goldberg and Elhadad 2009), a relatively small corpus that includes around 6,200 sentences taken from an Israeli daily newspaper.

Several features set our work apart from Goldberg (2011): first, we focus on *spoken*, colloquial Hebrew, rather than on written, journalistic language. In particular, our corpus includes samples produced by adult, non-expert speakers as well as by children who are still acquiring their grammar. The study of spoken language requires dealing with utterance complexity at varying levels. On the one hand, spoken language implies short utterances [as witnessed by Sagae et al. (2010)] that are thus generally simpler and easier to parse. On the other hand, utterances are partial and lack any standard formal structure, and some may be ungrammatical (in the traditional sense), especially those produced by children. Spoken language is also characterized by repetitions, repairs, and interruptions within the same utterance and even across utterances. Furthermore, a single utterance may contain more than one clause. These characteristics raise issues that to date have not been addressed in the automatic syntactic analysis of Hebrew.

Second, the parser described by Goldberg (2011) relies on data represented via Hebrew orthography. Hebrew orthography is ambiguous, since vowels are represented only partially (if at all), so that many words are homographic (Wintner 2004). Our transcription explicitly encodes vowels as well as other phonological information that is highly relevant for ambiguity resolution, such as lexical stress (Albert et al. 2014). But it is not a phonetic encoding: the transcription distinguishes between similar-sounding phonemes that are written differently in the standard Hebrew script, e.g., *ṭ* "Tet" and *t* "Tav", or *k* "Kaf" and *q* "Quf". This approach significantly reduces morphological ambiguity, and renders moot some of the problems that restrict the capabilities of Goldberg's parser. Moreover, functional elements (the definite article, prepositions, and conjunctions) that are concatenated with the subsequent token in the standard orthography are separated in our transcription; e.g., *we- ha- yeled* "and the child". This allows us to make sure that each lexeme is treated separately and consistently at the lexical, morphological, and syntactic levels. In addition, multi-word expressions are systematically transcribed as a single token, as in the case of *šalāṭ_raxōq* "remote control", *lāyla_ṭov* "good

night", etc. Consequently, the level of morphological ambiguity of each transcribed token is very low. Any remaining ambiguity is subsequently resolved in an automatic process (akin to part-of-speech tagging), so that the input for parsing is completely disambiguated. Albert et al. (2014) show that the automatic morphological disambiguation module achieves an accuracy of 96.6 %. This is in contrast to written Hebrew which suffers from high morphological ambiguity, and for which automatic morphological disambiguation is less accurate (Lembersky et al. 2014).

Furthermore, Goldberg's scheme itself is partial, i.e., not all relations existing in the data are labeled. In contrast, the scheme we developed provides fully annotated data (Sect. 4). As such, our work is similar in spirit to Sagae et al. (2010), who, motivated by the same goals, constructed a parser for the English section of the CHILDES corpus. Sagae et al. (2010) developed a scheme of 37 distinct grammatical relations which was used for annotating the Eve corpus (Brown 1973). The annotation included both manual and automatic analyses of 18,863 utterances, 10,280 adult and 8,563 child. During the annotation procedure, the corpus was used for training a data-driven parser which was then tested on an independent corpus of child–adult interactions. Cross-validation evaluation of the parser's performance showed a low error rate, of between 6 and 8 %. Both the English and the Hebrew data sets follow the CHAT transcription guidelines (MacWhinney 2000), and attempt to reflect the flow of the conversation as accurately as possible. The transcription standard also marks tokens that should be ignored by morphological and syntactic analyses, for example in the case of false starts or repetitions.[1]

Tsarfaty et al. (2012) also address Hebrew; they are concerned with joint evaluation of morphological segmentation and syntactic parsing, specifically in morphologically rich languages such as Hebrew. The motivation is that standard evaluation metrics for syntactic parsing do not take into account the underlying morphological segmentation that may produce errors which do not reflect actual parsing errors. Their proposed method allows for precise quantification of performance gaps between the use of gold morphological segmentation and that of predicted (non-gold) morphological segmentation. This work is representative of an emerging research program whose main goal is to develop techniques for parsing morphologically-rich languages (Tsarfaty et al. 2010, 2013; Seddah et al. 2011).

While this line of works is of course relevant to Hebrew parsing in general, it does not seem to be as relevant to parsing CHILDES data. The transcription we use (Sect. 3.2) allows high accuracy of morphological segmentation and part-of-speech tagging, as many of the ambiguity issues that normally arise in processing Hebrew are already resolved. In addition, for evaluation purposes we manually tag all remaining morphological ambiguity, rendering the morphological analysis disambiguated. Thus, our data do not reflect the rich morphology and high level of

[1] However, the Hebrew transcriptions are not always consistent with respect to these markings. Corrections have been made on these corpora where possible but some problematic instances may remain which may have a negative impact on the quality of the parsing.

ambiguity of standardly written Hebrew, which motivate such a joint evaluation metric.

For parsing (and evaluation) we compare MaltParser (Nivre et al. 2006) and MEGRASP (Sagae and Tsujii 2007). MaltParser is an architecture of transition-based parsers that can support various learning and parsing algorithms, each accompanied with its feature set and parameters, which can be directly optimized. The feature set is derived from the surface forms, the base forms and the morphological information of a subset of the tokens in the data structures (i.e., the queue and the stack) that comprise the state of the parser.

Marton et al. (2013) evaluate the effect of morphological features on parsing Arabic, a sister-language to Hebrew, using MaltParser. This study shows that the most important features for parsing written Arabic utterances are 'case' and 'state', while 'gender', 'number' and 'person' did not improve the accuracy of parsing Arabic when gold morphological information was present. Case is irrelevant for Hebrew; the state of nouns is, but construct state nouns are not common in spoken language, and in our corpus in particular. However, 'gender', 'number' and 'person' may prove to be more helpful when the training set is enhanced with corpora which are automatically annotated for morphological features. We explore this direction in Sect. 6.5.

MEGRASP was chosen because it was used for dependency parsing of the English section of CHILDES (Sagae et al. 2010). The algorithm is a dependency version of the data-driven constituent parsing algorithm for probabilistic GLR-like parsing described by Sagae and Lavie (2006). The parser is compatible with the format of the CHILDES database since CHILDES syntactic annotations are represented as labeled dependencies.

## 3 Characteristics of the Hebrew CHILDES corpus

### 3.1 Representation

The corpora in the CHILDES database include three levels of data, each referred to as a *tier* (MacWhinney 2000). The *main* tier contains (a transcription of) actual utterances; the *mor* tier contains disambiguated lexical and morphological analyses of the utterances; and the *gra* tier lists the syntactic analyses of the utterances. The data are organized in a one-to-one format, where every token in the main tier has exactly one counterpart in the mor tier and the gra tier.[2]

In the CHILDES database and throughout this work, dependency structures are represented linearly as triplets *i*|*j*|*REL*, where *i* is the index of a token in the utterance (starting from 1), *j* is the index of the head of the current token (the special `Root` marker is given the index 0), and *REL* is the label of the relation between them.

[2] This one-to-one alignment is automatically verified by the Chatter program: http://talkbank.org/software/chatter.html.

Example 1 shows an utterance with all three tiers. The elements in each of the three tiers are vertically aligned. The utterance consists of two tokens, *ʔat* "you" and *mešaqēret* "lie". Below each token, the mor tier specifies the morphological analysis. The first token is analyzed as a personal pronoun (`pro:pers`) of feminine gender and singular number; the second token is a participle (`part`), again in feminine singular. The third line specifies the gra tier: the first token is a dependent of the second, and the dependency relation is `Aagr` (agreeing argument, Sect. 4); and the second token is the root, depending on the special index 0. For convenience, we also add a fourth line, with a full English translation of the utterance.

(1)

| *ʔat "you"* | *mešaqēret "lie"* |
|---|---|
| `pro:pers\|gen:fm&num:sg` | `part\|gen:fm&num:sg` |
| `1\|2\|Aagr` | `2\|0\|Root` |

"You are lying!"

### 3.2 Tokenization

Tokenization is a crucial aspect of data preparation required for successful syntactic analysis, particularly for morphologically complex languages. Hebrew, which tends to be synthetic, is rich in bound morphology, and tokens that consist of more than a single morpheme are quite common. One such example is the class of simple prepositions *be-* "in", *le-* "to", and *ke-* "as" that are frequently fused with the definite article, as in the case of *ba-* "in the". Two other common examples are complex prepositions with suffixed personal pronouns, for example, *bišvilēḳ* "for you", and nouns that systematically take pronominal pronouns as bound inflectional suffixes marking possession, for example, *ʔaxotī* "my sister". These cases make it difficult to determine how many morphemes are to be counted in the analysis of one token (Dromi and Berman 1982).

Previous studies on dependency-based parsers for other morphologically complex languages have considered similar issues, as in the case of the Prague Arabic Dependency Treebank (Hajič and Zemánek 2004). Arabic, even more than Hebrew, is rich in bound, inflected morphemes, not only in the nominal system (as in the case of nouns with possessive suffixes) but also in the verbal system, where verbs can take accusative suffixes. The morphological analysis of the Prague Arabic Treebank is based on the idea of splitting words into *morphs* or *segments*, each receiving its own morphological analysis (Smrž and Pajas 2004). Thus, prepositions and suffixed pronouns are considered separate tokens (e.g., *biduni* "without me" is split into *bi- duni*), even if they are orthographically attached in Modern Standard Arabic script. Nouns with possessive suffixes are also split into two tokens (the noun and the pronominal suffix).

A similar approach is presented by the dependency-based parser for Turkish (Eryiğit et al. 2008). In the case of this agglutinative language, dependency relations are specified between morphemes (referred to as *inflectional groups* or IGs) rather than between words. This morpheme-driven analysis resulted in more accurate parsing. However, since the mechanisms for bound morphology are much less

productive in Hebrew compared to Arabic (let alone the agglutinative Turkish), we resorted to the following solution: the data were pre-processed and particular tokens were split to allow for a partial morpheme-bound representation. Thus, definite prepositions such as *ba-* "in the" are split to two tokens, as in *be-* "in" *ha-* "the"; prepositions fused with personal pronouns are treated similarly, for example, *bišvilēḵ* "for you" is split into *bišvīl* "for" *ʔat* "you", as are nouns inflected for possession; we split such cases to three morphemes: for example, *ʔaxotī* "my sister" is split into *ʔaxōt* "sister" *šel* "of" *ʔanī* "I".

Our motivation is largely computational rather than linguistic; presumably, such a split representation reduces data sparseness. Certain fused forms occur only rarely, possibly making it more difficult for the parser to identify and analyze them correctly. However, these changes are only aimed at improving syntactic analysis. In order to avoid any theoretically controversial decision as well as more practical implications, for example to the analysis of Mean Length of Utterance (Dromi and Berman 1982), we merge the split tokens back together once parsing is complete, omitting any intra-word relations. Unsurprisingly, with only a single exception, none of the split morphemes is involved in non-local dependency relations, so this merge operation is well-defined.

## 4 Annotation scheme for Hebrew CHILDES

As noted above, our scheme is inspired by the grammatical relations defined for the annotation of the English section of CHILDES (Sagae et al. 2010). This is done mostly for issues that are not language-specific but rather represent general characteristics of spoken language, such as repetitions and repairs, on the one hand, and the inclusion of vocatives and communicative elements, on the other. In addition, some of the relations that were defined similarly in the two schemes relate to specific features of child–adult interactions, including onomatopoeias, enumerations, serialization of verbs, and topicalizations.

Moreover, our scheme remained consistent with the issue of how to treat coordinate constructions, a relation that poses challenges for dependency annotation in general. In adopting the English scheme, the coordinating conjunction was defined as the head of a coordination construction and the coordinated elements were defined as the dependents. This relation is labeled `Coord`. Also, we took into consideration the work of Goldberg (2011) on dependency parsing of written Hebrew, specifically in Sect. 7 where we evaluate alternative approaches for specific relations.

In contrast to the English scheme, we distinguish between three types of dependents for Hebrew: arguments [A], modifiers [M] and others. *Arguments* are dependents that are typically semantically required by the head, their properties are determined by the head, and they can occur at most once (often, exactly once). *Modifiers*, on the other hand, are non-obligatory dependents: typically, they select the head that they depend on and, consequently, they may occur zero or more times. The *Others* group includes relations in which the dependents are neither arguments nor modifiers of the heads, or relations in which the dependents do not relate

specifically to any other token in the utterance. For example, the `Com` label marks the relation in which a communicator is the dependent. A communicator is generally related to the entire utterance, and so we mark the root of the utterance as the head of the communicator. The *Others* group also contains relations for two special cases where we present two optional analyses for a construction: the copula construction (Sect. 7.1) and constructions containing the accusative marker (Sect. 7.2). The second approach for both of these linguistic issues is marked with a relation whose name starts with `X`.

Typically in dependency-based parsers, the root of an utterance is an inflected verb or a copula in verbless copula utterances, carrying the tense marking in the clause. In utterances with no verb and no copula, where there is no element carrying a tense, the head is the predicating element. Copulas and existential markers, as well as other forms of the verb *hayā* "be", are discussed elaborately in Sect. 7.1 When an utterance is lacking any of the above, the root is the element on which the other elements depend (such as the noun with respect to its modifiers). In single word utterances, the single token is by default the root.

The annotation scheme is comprised of 24 basic dependency relations and a few more complex dependency relations (combinations of two basic dependency relations; see Sect. 4.4). The complete list of the basic dependency relations is given in Appendix 1. We discuss below some of the main constructions covered by our scheme.

### 4.1 Verb arguments, agreeing and non-agreeing

Two main relations are defined between verbs and their arguments. One relation, `Aagr`, requires the verb and its argument to agree; at most one argument can stand in this relation with any given verb. The other relation, `Anonagr`, imposes no agreement constraints, and the number of such arguments can be zero or more.[3]

In Example 2, the verb *rocē* "want" is the head of its *agreeing* argument *ʔanī* "I" (both are singular); it is also the head of its *non-agreeing* argument *ṭipōt* "drops" (a plural noun).

(2)

| *ʔanī "I"* | *loʔ "no"* | *rocē "want"* | *ṭipōt "drops"* |
|---|---|---|---|
| `pro:person\|num:sg` | `neg` | `part\|num:sg` | `n\|num:pl` |
| `1\|3\|Aagr` | `2\|3\|Mneg` | `3\|0\|Root` | `4\|3\|Anonagr` |

"I don't want drops."

Other non-agreeing relations include the relation between the verb and its *indirect* (or *oblique*) objects, where the nominal element is preceded by a preposition. The `Anonagr` dependency is marked on the prepositional element and the nominal element is marked as the argument of that preposition, `Aprep`; see Example 3. An alternative representation of prepositional phrases is discussed in Sect. 7.3

[3] The standard terminology, of course, is *subject* for `Aagr` and *object* for `Anonagr`. We use formal, rather than functional labels, for consistency and to avoid theory-specific controversies.

(3)

| *ʔoy "oh_no"* | *, ","* | *pagāʕti "I-hurt"* | *be- "in"* | *Siwān "Sivan"* |
|---|---|---|---|---|
| co | , | v | prep | pro:person |
| 1\|3\|Com | 2\|3\|Punct | 3\|0\|Root | 4\|3\|Anonagr | 5\|4\|Aprep |

"Oh no, did I hurt Sivan?"

Non-agreeing arguments can also occur as finite clausal dependents of the verb. In such cases, the subordinating conjunction is marked as Anonagr. In addition, it is treated as the head of the subordinate clause, and the finite verb of the subordinate clause is dependent on it in a `SubCl` relation, as in Example 4.

(4)

| *ʔatā "you"* | | *rocē "want"* | *še- "that"* | *ʔanī "I"* |
|---|---|---|---|---|
| pro:person\|gen:ms&num:sg | | part\|gen:ms&num:sg | conj:subor | pro:person\|num:sg |
| 1\|2\|Aagr | | 2\|0\|Root | 3\|2\|Anonagr | 4\|5\|Aagr |
| *ʔesarēq "comb"* | *ʔet "ACC"* | *ʔatā "you"* | | |
| v\|num:sg | acc | pro:person | | |
| 5\|3\|SubCl | 6\|5\|Anonagr | 7\|6\|Aprep | | |

"Do you want me to comb your hair?"

When the subordinate clause is introduced by something other than a subordinating conjunction, the finite verb of the clause is directly dependent on the finite verb of the main clause, again in a `Anonagr` relation, such as in Example 5, where the verb *qarā* "happen" (the head of the subordinate clause) is directly dependent on the matrix verb *tirʔī* "look".

(5)

| *ʔoy "oh_no"* | *tirʔī "look"* | *ma "what"* | *qarā "happen"* | *le- "to"* | *hiʔ "she"* |
|---|---|---|---|---|---|
| co | v | que | v | prep | pro:person |
| 1\|2\|Com | 2\|0\|Root | 3\|4\|Aagr | 4\|2\|Anonagr | 5\|4\|Anonagr | 6\|5\|Aprep |

"Oh no, look what happened to her."

When the non-agreeing argument is an infinitival verb phrase, the relation between the head of the verb phrase and its (verbal or nominal) head is `Ainf`; see Example 6.

(6)

| *ʔaz "so"* | *titēn "you-let"* | *le- "to"* | *ʔanī "I"* | *laʕavōr "pass"* |
|---|---|---|---|---|
| adv | v | prep | pro:person | v |
| 1\|2\|Com | 2\|0\|Root | 3\|2\|Anonagr | 4\|3\|Aprep | 5\|2\|Ainf |

"So let me pass."

## 4.2 Modifiers

Modification in Hebrew may occur for both nouns and verbs. Several relations specify nominal modifiers; these include `Mdet` for the relation between a determiner and a noun, `Mquant` for quantifiers, and `Madj` for adjectival modification. Another type of nominal modification is represented in noun-noun compounds, which in Hebrew are constructed by combining a morphologically-marked noun (said to be in the *construct* state) with another noun (recall that when such compunds are idiomatic they are represented as a single token). We mark the relation between the two nouns as `Mnoun`, as in Example 7.

(7)

| *we- "and"* | *bifnīm "inside"* | *yeš "there_is"* | *xelqēy "part"* | *matēḳet "metal"* |
|---|---|---|---|---|
| conj | adv | exs | n | n |
| 1\|0\|Root | 2\|3\|Madv | 3\|1\|Coord | 4\|3\|Aexs | 5\|4\|Mnoun |

"And inside there are parts of metal."

Verbal modifiers include `Madv` for adverbs (Example 8) and `Mneg` for negation (Example 2), as well as `Mpre` for prepositional phrase modifiers (Example 9).

(8)

| *ma "what"* | *ʕoṣīm "do"* | *ʕaḳšāyw "now"* |
|---|---|---|
| que | part | adv |
| 1\|2\|Anonagr | 2\|0\|Root | 3\|2\|Madv |

"What do we do now?"

(9)

| *be- "in"* | *masrēq "comb"* | *ʔaxēr "different"* | *tistarqī "you-comb_oneself"* |
|---|---|---|---|
| prep | n\|gen:ms&num:sg | adj\|gen:ms&num:sg | v |
| 1\|4\|Mpre | 2\|1\|Aprep | 3\|2\|Madj | 4\|0\|Root |

"Comb your hair with a different comb."

When a subordinate clause is a *modifier* (rather than an argument) of a verb or a noun, the relation between the verb or noun and the subordinating conjunction is labeled `Msub`. If the clause is a relative clause, the relation between the relativizer and the head of the relative clause is labeled `RelCl`, as in Example 10.

(10)

| *balōn "balloon"* | *še- "that"* | *hitpocēc "burst"* |
|---|---|---|
| n\|gen:ms&num:sg | conj:subor | v\|gen:ms&num:sg |
| 1\|0\|Root | 2\|1\|Msub | 3\|2\|RelCl |

"A balloon that burst."

### 4.3 Other relations

*Vocatives* are named entities that refer to another speaker in the conversation, most commonly followed by a question or request in the second person. Vocatives depend on the root of the utterance in a `Voc` relation (Example 11).

(11)

| *ʔasaf "Asaf"* | , "," | *tedabēr "speak"* |
|---|---|---|
| n:prop | , | v |
| 1\|3\|Voc | 2\|3\|Punct | 3\|0\|Root |

"Asaf, speak up."

*Communicators* include discourse markers such as *ʔavāl* "but", *ʔaz* "so", *ken* "yes", etc., as well as verbs such as *tirʔē* "look" and *bōʔi* "come_here". Like `Voc`, the root of the utterance is the head of the relation and the communicator is the dependent. The main difference between the two relations is that `Com` does not include named entities. See Examples 5, 6.

The relation `Coord` specifies coordination, relating between conjuncts and conjunctions, most commonly *we-* "and". As noted above, we follow Sagae et al. (2010) in dealing with these constructions: the head is the coordinating conjunction and the dependents are the conjuncts. If there are two or more conjuncts with multiple coordinators, the coordinators are linked from left to right (the rightmost coordinator is the head of the others) by a `Coord` relation. In the absence of a coordinator the rightmost conjunct is the head of the relation. See Example 12.

(12)

| *huʔ* "he" | *rac* "run" | *we-* "and" | *hitxabēʔ* "hide" |
|---|---|---|---|
| pro:person\|gen:ms&num:sg | v\|gen:ms&num:sg | conj | v\|gen:ms&num:sg |
| 1\|3\|Aagr | 2\|3\|Coord | 3\|0\|Root | 4\|3\|Coord |

| *meʔaxorēy* "behind" | *ha-* "the" | *ʕec* "tree" |
|---|---|---|
| prep | det | n |
| 5\|4\|Mpre | 6\|7\|Mdet | 7\|5\|Aprep |

"He ran and hid behind the tree."

## 4.4 Elision relations

Spoken language often includes missing elements, whether as a result of true ellipsis or of interruptions and incomplete utterances. In the English section of CHILDES, Sagae et al. (2010) decided to mark missing elements as elided and to relate to them in the analysis using *elision relations*. Such relations combine two basic relations: one between the elided element and its presumed head, and one between the elided element and its dependent. Following the scheme for English, we also mark missing elements with elision relations.

In Example 13, *ha-* "the" is marked with the `Mdet-Aprep` relation. `Mdet` stands for the relation between *ha-* "the" and a missing element, presumably a noun; `Aprep` stands for the relation that would have held between the missing noun and the preposition *leyād* "near".

(13)

| *leyād* "near" | *ha-* "the" |
|---|---|
| prep | det |
| 1\|0\|Root | 2\|1\|Mdet-Aprep |

"Near the."

## 4.5 Child invented language

As the CHILDES corpus is comprised of child and adult interactions, child-specific forms and constructions are rather frequent. These include neologisms, babbling, and incoherent speech. Such forms can be detached from the utterance, labeled with the `Unk` relation which marks unknown relations (Example 14); or, when the syntactic function of such forms is known to the annotator, they can take the place of a known relation (e.g., the neologism *bdibiyabi* in Example 15).

(14)

| *curu* "curu" | *gam* "also" | *lecayēr* "paint" |
|---|---|---|
| chi | adv | v |
| 1\|3\|Unk | 2\|3\|Madv | 3\|0\|Root |

"??? also to paint."

(15)

| *ʕaḵšāyw* "now" | *ʔanī* "I" | *holēḵet* "walk" | *le-* "to" | *bdibiyabi* "bdibiyabi" |
|---|---|---|---|---|
| adv | n\|num:sg | part\|num:sg | prep | chi |
| 1\|3\|Madv | 2\|3\|Aagr | 3\|0\|Root | 4\|3\|Anonagr | 5\|4\|Aprep |

"Now I am going to bdibiyabi."

**Table 1** Statistics of corpora used for evaluation

| Corpus | Files | Utterances | | Tokens | | MLU | | MLUw | |
|---|---|---|---|---|---|---|---|---|---|
| | | Total | CS | Total | CS | Total | CS | Total | CS |
| Ravid | 8 | 4,107 | 1,541 | 13,863 | 3,975 | 3.9 | 2.9 | 3.4 | 2.6 |
| Berman | 4 | 2,224 | 1,126 | 9,392 | 4,241 | 4.9 | 4.3 | 4.2 | 3.8 |

MLU is the average number of morphemes per utterance. MLUw is the average number of tokens per utterance. CS is Child Speech

## 5 Methodology

### 5.1 Parsing

We manually annotated a subset of the Hebrew CHILDES corpus described in Sect. 3 according to the schema of Sect. 4. The data were annotated by two lexicographers; all disagreements were resolved by a third annotator, a linguist who specializes in syntactic analysis.

This manually annotated corpus consists of 12 files: 8 files from the Ravid corpus and 4 from the Berman corpus. The 8 files of the Ravid corpus contain transcriptions of the same child at different ages (ranging from 1;11 to 2;05). The 4 files of the Berman corpus reflect 4 different children (all different from the child in the Ravid corpus) at different ages (2;04, 3;00, 3;03 and 3;06). Statistical data of the corpora are given in Table 1. The data presented here refer to the corpora after splitting fused morphemes (Sect. 3.2) and exclude punctuation.

We then trained two dependency parsers on the manually-annotated texts, MEGRASP (Sagae and Tsujii 2007) and MaltParser (Nivre et al. 2006). MEGRASP works directly on the CHILDES format in which the corpora are stored. MaltParser supports a number of formats, including the CoNLL-X shared task format (Nivre et al. 2007). An advantage of using MaltParser is that it also supports costume-made formats, allowing variation in the lexical and morphological information available for the learning algorithm. We used a format similar to CoNLL, but added columns to represent independent morphological attributes (instead of the concatenated FEATS column). Using MaltParser, we examined the effect of adding morphological features (e.g., number and person) to the default feature set (Sect. 6.5).

To achieve the best possible results using MaltParser we used the recently developed *MaltOptimizer* (Ballesteros and Nivre 2012). MaltOptimizer analyzes the training data in a three-phase process and outputs the recommended configuration under which to run MaltParser (e.g., a certain parsing algorithm or a feature set that yield the best results). Since MaltOptimizer is restricted to the CoNLL format and does not support custom formats, we used it as follows. We concatenated the morphological features into the FEATS column, to adapt the input to the CoNLL format. We ran this version of the parser with MaltOptimizer, and converted the files back to our custom format as suggested by MaltOptimizer. For example, MaltParser supports a *Split* function that splits the values of a certain column according to a delimiter. If MaltOptimizer suggested to split the FEATS column, we

did so by placing the morphological information in the separate morphological columns. In the following sections, there is practically no difference between using our format and the CoNLL format. The main difference is when we evaluated the contribution of the morphological data to parsing (Sect. 6.5); there we examined various kinds of subsets of features, not all of which are supported by the regular CoNLL format.

## 5.2 Evaluation

We conducted both *In-domain evaluation*, where training is on parts of the Ravid corpus and testing is on other parts of the same corpus (held out during training), and *Out-of-domain evaluation*, where training is done on the files of the Ravid corpus and testing is done on the files of the Berman corpus. We did not explore any domain adaptation techniques (Nivre et al. 2007; Plank 2011), we merely evaluated the robustness of the parser when tested on a different domain. We ran both MEGRASP and MaltParser on these evaluation scenarios. We also ran a fivefold cross-validation on the Ravid corpus and on both corpora combined.

The evaluation metrics that we used are *unlabeled attachment score* (UAS) and *labeled attachment score* (LAS). In UAS a token is considered correctly annotated if its head is the same head that is marked in the gold-standard—regardless of the grammatical relation. In LAS a token is considered correctly annotated if both the head and the grammatical relation are the same as in the gold-standard. In addition we report *Exact Match* (EXM), the percentage of utterances that are parsed without any errors. These are standard metrics in the evaluation of dependency parsing (Kübler et al. 2009).

To examine the quality of the parsers and the annotation scheme on individual relations, we used further metrics that are relation specific—$URecall_r$ (unlabeled recall), $LRecall_r$ (labeled recall), $UPrecision_r$ (unlabeled precision) and $LPrecision_r$ (labeled precision) for some relation $r$ (Kübler et al. 2009). Let $l_g(x)$ be the gold label of token $x$ and $l(x)$ the label assigned by the parser. Similarly, let $h_g(x)$ be the head that token $x$ is attached to in the gold file and $h(x)$ the head that token $x$ is attached to by the parser. Then:

$$URecall_r = \frac{|\{x \mid l_g(x) = r \wedge h_g(x) = h(x)\}|}{|\{x \mid l_g(x) = r\}|}$$

$$LRecall_r = \frac{|\{x \mid l_g(x) = l(x) = r \wedge h_g(x) = h(x)\}|}{|\{x \mid l_g(x) = r\}|}$$

$$UPrecision_r = \frac{|\{x \mid l(x) = r \wedge h_g(x) = h(x)\}|}{|\{x \mid l(x) = r\}|}$$

$$LPrecision_r = \frac{|\{x \mid l_g(x) = l(x) = r \wedge h_g(x) = h(x)\}|}{|\{x \mid l(x) = r\}|}$$

The first two metrics are refinements of the recall metric for each relation as the analysis is with respect to the appearances of the relation in the gold standard files. The $Recall_r$ measures compute the percentage of tokens labeled $r$ in the gold data

that were correctly parsed. The other two metrics are refinements of the precision metric for each relation as the analysis is with respect to the appearances of the relation in the parsed model files. The $Precision_r$ measures compute the percentage of tokens labeled $r$ in the parsed data that were correctly parsed. For each of the UAS and LAS precision and recall pairs we report also the (balanced) $F$-score, the harmonic mean of precision and recall.

In addition to testing the corpus as a whole we show results that relate separately to two types of data: child-directed speech (CDS) and child speech (CS). First, we trained and tested on both types of data (All–All); to investigate whether children learn primarily from the language spoken to them or from their peers, we trained on CDS and tested on CS (CDS–CS), and then trained and tested on CS only (CS–CS); for completion, we also trained and tested on CDS (CDS–CDS).

In the in-domain evaluation scenario we built the training set and test set for each of these configurations separately, using 8 files of the Ravid corpus. The files of the Ravid corpus are chronologically ordered by the age of the target child and thus in the in-domain evaluation scenario the held-out set always contains utterances of the same child at an older age. In this configuration, the training set is comprised of 80 % of the utterances of the relevant data type in the corpus, holding out 20 % for the test set. The training set of the All–All configuration contains 3,286 utterances (11,155 tokens), the CS training set contains 1,237 utterances (3,246 tokens) and the CDS training set contains 2,066 utterances (7,946 tokens). The 80 % of CS and CDS were derived from the set of utterances (of their respective data types) in the corpus, and not from the training set of both data types. Consequently, the sum of the sizes of the CS and CDS training sets does not necessarily equal the size of the training set of the All–All configuration. In the CDS–CS configuration the training set and test set are comprised of utterances of different data types so the entire set of utterances of each data type in the corpus was used, and not just 80 % of it.

In the out-of-domain evaluation scenario the training sets and test sets of the different configurations were taken from different sets of files, so the entire set of utterances of the respective data type was used. For all evaluation scenarios, we excluded punctuation and single-token utterances, to avoid artificial inflation of scores.

## 6 Results

### 6.1 In-domain evaluation

We first present the results for the in-domain scenario. Recall that we ran MaltOptimizer in order to achieve the best parser configuration with respect to the training set. In the All–All configuration, according to MaltOptimizer, the training set contained approximately 3.9 % utterances with non-projective trees.[4]

[4] A dependency tree is *projective* if it has no crossing edges. For some languages, especially when word order is rather free, non-projective trees are preferred for better explaining sentence structure (Kübler et al. 2009, p. 16).

**Table 2** Results: accuracy of parsing, in-domain

| Evaluation scenario | | | | MEGRASP | | | MaltParser | | |
|---|---|---|---|---|---|---|---|---|---|
| Train | Size | Test | Size | UAS | LAS | EXM | UAS | LAS | EXM |
| All | 3,286 | All | 590 | 87.4 | 82.3 | 62.7 | 91.2 | 86.6 | 71.1 |
| CS | 1,237 | CS | 183 | 91.9 | 87.3 | 75.4 | 93.9 | 89.1 | 78.1 |
| CDS | 2,066 | CDS | 400 | 85.4 | 80.8 | 56.7 | 89.2 | 83.9 | 63.2 |
| CDS | 2,566 | CS | 969 | 84.2 | 78.5 | 62.3 | 88.2 | 82.5 | 68.2 |

MaltOptimizer recommended using the Stack parsing algorithm in its non-projective eager version (Nivre 2009; Nivre et al. 2009). See Appendix 2.1 for a full description of the parameters chosen by the optimizer.

Table 2 shows the accuracy of parsing obtained by both parsers, in all four evaluation scenarios. Considering the relatively small training set, both parsers achieve reasonable results. Evidently, MaltParser proves to be better than MEGRASP on this domain. The difference in the All–All configuration is statistically significant for all three metrics ($p < 0.05$).

To show the contribution of MaltOptimizer, we also ran MaltParser with its default parameters, which allows only projective dependency trees. The settings of the default parsing algorithm are discussed in Appendix 2.2. In the All–All configuration, the UAS was 84.5 and the LAS was 80.5—lower than the results obtained by both the optimized MaltParser and MEGRASP. Thus, the adaptation of the parsing algorithm and feature set to our corpora using MaltOptimizer was clearly instrumental for improving parsing accuracy.

In general, the Exact Match accuracy is high, mostly due to the relatively short length of the utterances in our corpora. It is interesting to compare these results to Exact Match results of other tasks. In the CoNLL-X shared task (Nivre et al. 2007), different parsers were evaluated for token-based measures, such as LAS and UAS, by parsing 13 test-sets of various languages. Ballesteros et al. (2012) expanded this evaluation of parsers by calculating not only token-based but also sentence-based measures, such as Exact Match. They also drew a correlation between average sentence length and Exact Match accuracy. The test-set of Arabic had the highest average sentence length (37.2 tokens) and the lowest Exact Match score (9.6 with MaltParser, 6.2 averaged across parsers). On the other hand, the test-sets of Chinese and Japanese had the shortest average sentence length (5.9 and 8.9 tokens, respectively) and the highest Exact Match scores (68.1 with MaltParser and 49.5 averaged across parsers for Chinese, 75.3 with MaltParser and 59.6 averaged across parsers for Japanese). These results are in accordance with our results, as both the Ravid and Berman corpora exhibit a short average utterance length and high Exact Match scores, arising from the fact that they reflect adult-child interactions at early stages of language development.

Note also the low EXM when testing on CDS as opposed to the high EXM when testing on CS. Recall that the utterances in CDS are longer on average (Table 1) and

**Table 3** Results: accuracy of parsing of some individual relations, in-domain

| Relation | UAS | | | LAS | | |
|---|---|---|---|---|---|---|
| | Recall | Precision | F-score | Recall | Precision | F-score |
| Root | 93.2 | 92.7 | 92.9 | 93.2 | 92.4 | 92.8 |
| Aagr | 92.1 | 95.2 | 93.6 | 89.5 | 91.9 | 90.7 |
| Aprep | 99.4 | 98.2 | 98.8 | 99.4 | 97.6 | 98.5 |
| Anonagr | 94.6 | 92.6 | 93.6 | 89.2 | 84.3 | 86.7 |
| Mdet | 99.1 | 98.6 | 98.8 | 98.6 | 98.6 | 98.6 |
| Madv | 87.1 | 80.4 | 83.6 | 78.5 | 71.6 | 74.9 |
| Com | 74.6 | 71.2 | 72.7 | 65.7 | 66.7 | 66.2 |
| Mpre | 79.7 | 75.8 | 77.7 | 61.0 | 58.1 | 59.5 |
| Aexs | 97.6 | 95.2 | 96.4 | 97.6 | 95.2 | 96.4 |
| Mquant | 78.6 | 87.5 | 82.8 | 50.0 | 87.5 | 63.6 |

so there is a higher chance that one of the tokens in an utterance is tagged erroneously.

To see which relations are more difficult for the parsers to predict, we evaluated the accuracy of the parsers on specific relations. Table 3 shows the relation-specific metrics for interesting individual relations, in the All–All configuration. Relations that occur with a small set of tokens as dependents (such as `Mdet`, where the dependent is mainly the token *ha-* "the"), or after a specific type of token (such as `Aprep`, occurring after a preposition) achieved a score of 97 % or above in all the four metrics. The frequent relations `Aagr` and `Root` reached high scores of over 92 % unlabeled recall and precision, 89 % labeled. Also accurate were the relations `Mneg` and `Aexs`. The more problematic relations were `Com` and `Voc` and modifiers such as `Madv`, `Mquant` and `Mpre`, which can sometimes be ambiguous even for human annotators. Amongst the modifiers the labeled scores of `Mpre` were especially low, due to the confusion between it and `Anonagr` when deciding whether a preposition is an argument or a modifier of a verb, in certain cases a decision that could be hard for a human annotator.

Figure 1 shows the learning curves of MEGRASP and MaltParser on this task. We trained the parsers on an increasingly larger training set, from 400 utterances up to 3,200 utterances with increments of 400, and tested on a fixed test set of 590 utterances (2,474 tokens) in the All–All configuration. We plotted UAS and LAS scores as a function of the number of utterances in the training set. The curves suggest that more training data could further improve the accuracy of the parser.

### 6.2 Out-of-domain evaluation

We also evaluated the parsers on a different domain than the one they were trained on. For the All–All configuration, according to MaltOptimizer, the training set contained approximately 3.8 % utterances with non-projective trees. Similarly to the in-domain scenario, MaltOptimizer suggested the Stack algorithm (Nivre 2009;

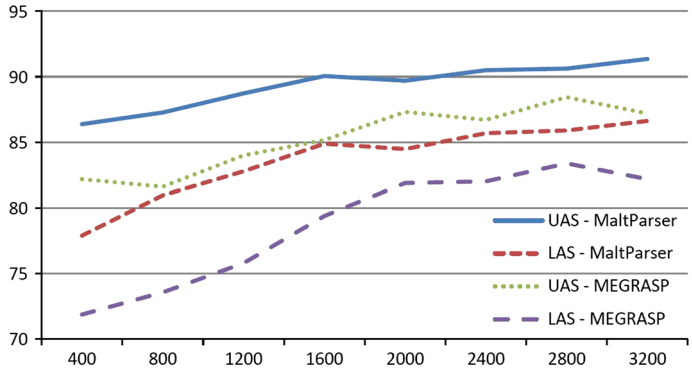



**Fig. 1** MEGRASP and MaltParser in-domain learning curves

**Table 4** Results: accuracy of parsing, out-of-domain

| Evaluation scenario | | | | MEGRASP | | | MaltParser | | |
|---|---|---|---|---|---|---|---|---|---|
| Train | Size | Test | Size | UAS | LAS | EXM | UAS | LAS | EXM |
| All | 4,107 | All | 1,614 | 78.4 | 73.1 | 51.3 | 82.0 | 77.1 | 55.6 |
| CS | 1,541 | CS | 761 | 69.2 | 61.4 | 42.0 | 74.7 | 68.0 | 50.7 |
| CDS | 2,566 | CDS | 853 | 81.3 | 76.3 | 48.8 | 85.0 | 79.7 | 53.6 |
| CDS | 2,566 | CS | 761 | 73.6 | 66.6 | 47.7 | 77.8 | 72.1 | 55.5 |

Nivre et al. 2009), but in contrast to the in-domain scenario, it recommended the Stack non-projective version. This algorithm postpones the SWAP transition of the Stack algorithm as much as possible. The parameters selected for this configuration are discussed in Appendix 2.3.

We trained the parsers on the 8 files of the Ravid corpus and tested on the 4 files of the Berman corpus. Table 4 lists the results.

Unsurprisingly, the accuracy of the parser in the out-of-domain evaluation scenario is considerably lower than in the in-domain evaluation scenario. The decrease in accuracy when parsing the CS data type can be explained by the fact that the test set of the Berman corpus contains utterances by four different children, all different from the child who is recorded in the training set. They are also children of different ages, and three of the four children in the test set are recorded at an older age than the child in the training set.

Another point to notice is that also in this scenario, MaltParser performed better than MEGRASP, but the differences between the parsers are slightly smaller in some metrics than in the in-domain evaluation scenario. One possible explanation is that MaltParser was run with optimized parameters as suggested by MaltOptimizer (e.g., parsing algorithm and feature set) that are configured according to the training set. In the out-of-domain evaluation scenario the differences in the types of utterances between the training set and the test set are more substantial than in the in-domain

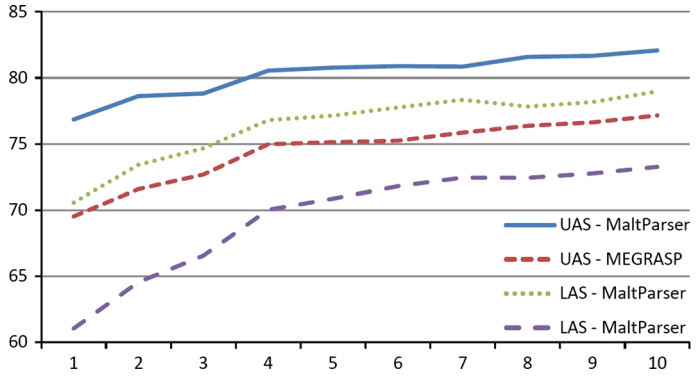



**Fig. 2** MEGRASP and MaltParser out-of-domain learning curves

evaluation scenario. As a result the optimized parameters are less effective and hence the accuracy is poorer. Still, the advantage of MaltParser over MEGRASP in the All–All configuration is significant for all three metrics ($p < 0.05$).

As in the in-domain evaluation scenario, we present a learning curve of the parsers when parsing the same out-of-domain dataset on training sets varying in size (Fig. 2). The size of the test set is 1,614 utterances (8,750 tokens). Here, too, the learning curves of both parsers suggest that there is room for improvement with more training data.

### 6.3 Learning from child-directed speech versus child speech

Is it better to train the parser on child speech or on child directed speech? The in-domain and out-of-domain tests yield conflicting evidence. The in-domain data suggest that for parsing child speech it is better to learn from child speech than from child-directed speech. This is despite the fact that in the CDS–CS configuration the training set is larger.

To examine the possibility that the specific CS test set used in both configurations contributes to this difference, we evaluated the CDS–CS configuration with a training set similar in size to the CS–CS training size (i.e., 1,237 utterances) and with an identical test set to the one used in the CS–CS configuration. Table 5 shows the results of the modified CDS–CS evaluation (line 2) compared to the CS–CS evaluation (line 1) and the original CDS–CS evaluation (line 3).

**Table 5** Results: accuracy of parsing, in-domain, CDS versus CS

| Evaluation scenario | | | | MEGRASP | | | MaltParser | | |
|---|---|---|---|---|---|---|---|---|---|
| Train | Size | Test | Size | UAS | LAS | EXM | UAS | LAS | EXM |
| CS | 1,237 | CS | 183 | 91.9 | 87.3 | 75.4 | 93.9 | 89.1 | 78.1 |
| CDS | 1,237 | CS | 183 | 91.1 | 85.5 | 69.9 | 92.3 | 88.0 | 77.6 |
| CDS | 2,566 | CS | 969 | 84.2 | 78.5 | 62.3 | 88.2 | 82.5 | 68.2 |

When running the modified CDS–CS configuration, accuracy was considerably higher than the original CDS–CS configuration, possibly due to this CS test set being easier to parse than the 969 utterances of the test set of the CDS–CS configuration presented in line 3. This could be contributed also to the fact that the test set was taken from the recordings of the child at an older age, thus it is perhaps more similar to CDS data than the CS test set of the original CDS–CS configuration which consists of the entire CS data. The scores of the modified CDS–CS configuration were slightly lower than the CS–CS scores, but the differences are not statistically significant.

The fact that training on CS has some advantage over training on CDS when parsing CS can be partially explained by the fact that the age range of the files of the Ravid corpus is rather small, the difference between the first file and the eighth file being only 7 months. Note that in the CDS–CDS configuration the scores are also relatively low. It is apparent that training on CDS confuses the parser to some degree. This can be explained by the richer structure of CDS compared to CS and by the different constructions and relations uttered by the same adults when the child matures.

However the out-of-domain data (Table 4) suggest that when parsing child speech it is better to learn from child-directed speech than from child speech. To further examine this result we trained the parsers on a CDS dataset similar in size to the CS dataset (i.e., the training set consists of 1,541 CDS utterances). Table 6 shows the results of the modified CDS–CS evaluation (line 2) compared to the CS–CS evaluation (line 1) and the original CDS–CS evaluation (line 3). The results suggest that there is some advantage to training on child-directed speech when parsing child speech, in contrast to the trend that emerged from the in-domain task.

The best scores for out-of-domain training, and the closest scores to the in-domain case, are obtained in the CDS–CDS configuration. This should most probably be attributed to the smaller variance that is expected in CDS between different adults, in contrast to the relatively substantial differences in CS.

### 6.4 Cross-validation

In addition to evaluating our annotation scheme on the same domain and on a different domain, we also tested it on the corpora as a whole without any distinction to participants or ages. The cross-validation process allows for a more robust evaluation of the entire data. To this end we evaluated the entire set of 12 files (concatenated into one large file) using fivefold cross-validation. Similarly to

**Table 6** Results: accuracy of parsing, out-of-domain, CDS versus CS

| Evaluation scenario | | | | MEGRASP | | | MaltParser | | |
|---|---|---|---|---|---|---|---|---|---|
| Train | Size | Test | Size | UAS | LAS | EXM | UAS | LAS | EXM |
| CS | 1,541 | CS | 761 | 69.2 | 61.4 | 42.0 | 74.7 | 68.0 | 50.7 |
| CDS | 1,541 | CS | 761 | 72.8 | 64.9 | 45.3 | 76.4 | 69.8 | 49.7 |
| CDS | 2,566 | CS | 761 | 73.6 | 66.6 | 47.7 | 77.8 | 72.1 | 55.5 |

**Table 7** Results: fivefold cross-validation

| Evaluation scenario | | MEGRASP | | | MaltParser | | |
|---|---|---|---|---|---|---|---|
| Train | Test | UAS | LAS | EXM | UAS | LAS | EXM |
| All | All | 84.0 | 79.3 | 60.6 | 89.5 | 85.8 | 70.2 |

**Table 8** Results: fivefold cross-validation, Ravid corpus

| Evaluation scenario | | MEGRASP | | | MaltParser | | |
|---|---|---|---|---|---|---|---|
| Train | Test | UAS | LAS | EXM | UAS | LAS | EXM |
| All | All | 87.0 | 82.2 | 63.5 | 90.8 | 86.7 | 71.0 |

previous evaluations, each fold of the cross-validation analysis had its parsing algorithm and feature set selected using MaltOptimizer. The results, presented in Table 7, clearly show the advantage of MaltParser over MEGRASP (the differences are statistically significant). They also underline the robustness of both parsers across domains and speakers.

In addition, we performed a similar fivefold cross-validation on the 8 files of the Ravid corpus, thereby restricting the cross-evaluation to a single domain (Table 8). Here we retained the domain, but ignored the age factor of the participant in the interaction, since, unlike in the regular in-domain scenario, the test set is not made of conversations in which the participant is necessarily older than in the training set. This scenario should be compared to the results of the evaluation of CHILDES in English (Sagae et al. 2010). Cross-validation on the Eve corpus of the English section of CHILDES (using MEGRASP) yielded an average result of 93.8 UAS and 92.0 LAS. However, the English training set was considerably larger (around 60,000 tokens compared to around 15,000 in the training set of each fold of our in-domain cross-validation evaluation).

The advantage of MaltParser over MEGRASP is statistically significant ($p < 0.05$) for both the Berman and the Ravid corpora, across all three measures.

### 6.5 Adding morphological features to improve parsing

Several morphological features are relevant for parsing. The gender, number and person of tokens are crucial for determining agreement. The argument of a verb can be either an agreeing argument (specified by the `Aagr` relation) or a non-agreeing argument (specified by the `Anonagr` relation). The 'form' feature of a token can indicate whether a verb is in the imperative mood or the infinitive mood. More specifically, the 'form' feature can help determine the `Ainf` relation, which only holds for infinitival verbs. Awareness of such features was proven useful for parsing of Arabic (Marton et al. 2013). In this section we investigate the impact on parsing accuracy of using such features. To this end, we modify the feature set of MaltParser (such functionality is currently limited in MEGRASP).

**Table 9** Results: feature improvements

| Feature set | UAS | LAS | EXM |
|---|---|---|---|
| NoMorph | 90.5 | 85.9 | 71.7 |
| NoMorph + VERBFORM (Lookahead [0]) | 90.9 | 86.2 | 71.9 |
| NoMorph + PERS, NUM and GEN (Stack [0], Stack[1], and Stack [2]) | 91.6 | 86.8 | 71.5 |

*NoMorph* is the MaltOptimizer suggested feature set without the morphological extended information

In some cases the optimized parameters for MaltParser, suggested by MaltOptimizer, already include morphological features. In this section we start with a feature set that does not include any of the four morphological features mentioned above (we refer to this set of features as *NoMorph*). We then add different subsets of features to the feature set and evaluate the accuracy of MaltParser using these features. The subsets that we test include adding up to three tokens from the top of the data structures used by the selected parsing algorithm, with references to the following data custom columns:

- VERBFORM, indicating the 'form' feature described above
- NUM, indicating the 'number' feature of the token
- PERS, indicating the 'person' feature of the token
- GEN, indicating the 'gender' feature of the token

As described in Sect. 6.1, the configuration suggested by MaltOptimizer for the in-domain All–All scenario included using the Stack parsing algorithm. The morphological features may appear in the various data structures of the algorithm, and the parser may use their values to aid its decisions.

Table 9 shows the accuracy of parsing (the in-domain task in the All–All configuration) with the features whose addition to NoMorph yields the highest improvement. The test set consisted of 590 utterances (2,474 tokens). Although VERBFORM provided some improvement in itself (second row), it did not provide further improvement when added to the combination of PERS, NUM and GEN (line 3). None of the improvements is statistically significant ($p > 0.1$).

Table 10 depicts the changes in the scores of some specific relations when PERS, NUM, and GEN of the elements in the three top positions of the stack were added to the NoMorph features.[5] This set of features improved the scores of these relations (except `Anonagr`) in almost every metric. Specifically, the big improvement in `Ainf` is clearly attributed to the verb form information that was made available to the parser. Note also that for some relations the increase in the labeled scores is higher than in the unlabeled scores, indicating the contribution of the features to identifying the grammatical relation correctly.

[5] The number of occurrences of the relation `Root` is not identical to the number of utterances since in some cases an elision relation (e.g., `Aagr-Root`) was used instead.

**Table 10** Results: feature improvements, individual relations

| Relation | Occurrences | UAS | | LAS | |
|---|---|---|---|---|---|
| | | Recall | Precision | Recall | Precision |
| Root | 585 | +0.8 | +1.1 | +0.8 | +0.9 |
| Anonagr | 295 | +0.7 | −0.8 | −0.4 | −1.7 |
| Aagr | 343 | +1.5 | +1.0 | +2.9 | +0.8 |
| Ainf | 16 | +6.2 | +10.0 | +12.5 | +14.9 |

Occurrences refers to the actual number of times this relation appears in the test set

## 7 Linguistic issues

Different frameworks of dependency-based parsers produce different analyses for existing linguistic controversies (Nivre 2005). In addition to testing for feature improvement, our work aims to investigate whether contrasting approaches to actual syntactic annotation yield different accuracy rates. Several syntactic constructions that frequently occur in our data can be annotated in two distinctly-motivated ways. In this section, we check empirically these different approaches to syntactic analysis. All evaluations used MaltParser and were conducted on the in-domain task, in the All–All configuration; the size of the training set was thus 3,286 utterances (11,155 tokens) and the size of the test set was 590 utterances (2,474 tokens).

In the following sections, we use two terms to refer to alternative analyses. The term *Approach A* refers to the annotation scheme described in Sect. 4, while the term *Approach B* refers to an alternative approach that we present for the first time in this section.

### 7.1 Copula constructions and other forms of *hayā* "be"

First, we examine utterances with some form of the functional verb *hayā* "be", which we term *hayā constructions*. In Hebrew, *hayā* constructions function in a variety of contexts, mainly copula and existential constructions (Rosen 1966; Berman 1978). For both types of constructions, which are quite common in Hebrew, the verbal form appears in either past or future tense. In present tense, the two constructions diverge. For copula constructions, the realization of the tense-carrying element is in some cases optional, and it usually takes the form of a pronoun when it is explicitly expressed. Thus, the same clause can occur without a copula, as in *ʔeitan gavōah* "Eitan tall", or with a copula in the form of a pronoun, as in *ʔeitan huʔ gavōah* "Eitan he tall". For existential constructions, the verbal form alternates with the suppletive (non-verbal) forms *yeš* "there_is" or *ʔeyn* "there_is_not".

Previous dependency-based parsers have suggested different ways to deal with copula constructions.[6] The scheme used for annotating English data within

[6] Examples in this section do not necessarily label the dependencies, either because the original work did not label them or because the label names are not relevant in our context.

CHILDES (Sagae et al. 2010) views the verbs 'be', 'become', 'get', etc., like other verbs, as the heads, and their nominal predicates as the dependents, as in Example 16:

(16)
| *Mary* | *is* | *a* | *student* |
|---|---|---|---|
| 1\|2 | 2\|0 | 3\|4 | 4\|2 |

In Hebrew, however, there is no consistent paradigm of copula verbs. Moreover, the optionality of the copula in present-tense constructions requires consideration (Haugereid et al. 2013). The Stanford Parser English scheme (de Marneffe et al. 2006), for example, is motivated by the need for adaptablity to languages in which the copula is not necessarily explicitly represented. In this scheme, the nominal predicate is viewed as the head and the copula as its dependent. The subject is also dependent on the nominal predicate (Example 17). According to de Marneffe and Manning (2008), an additional motivation for this decision was to help applications extract the semantic information of the clause through the direct relation between the subject and the predicate.

(17)
| *Bill* | *is* | *big* |
|---|---|---|
| 1\|3 | 2\|3 | 3\|0 |

Alternatively, while the Prague Arabic Dependency Treebank (Hajič and Zemánek 2004) treats a group of verbs which may act as copulas (referred to as "*kana* "be" and her sisters") as a subset of the entire verbal group and thus as heads, in clauses with zero copula the nominal element is analyzed as the head, as in Example 18.

(18)
| *al-'amru "The-matter"* | *wāḍiḥun "clear"* |
|---|---|
| 1\|2 | 2\|0 |

"The matter is clear."

The scheme for annotating written Hebrew presented by Goldberg (2011) is similar in this respect to the Prague Arabic dependency scheme. In a non-verbal clause with zero copula, the predicate is the head and the subject is its dependent (Example 19); when the copula is present, the predicate is also the head and the copula is its dependent (Example 20).

(19)
| *ha- "the"* | *yēled "child"* | *xaḳām "smart"* |
|---|---|---|
| 1\|2 | 2\|3 | 3\|0 |

"The child is smart."

(20)
| *ha- "the"* | *yēled "child"* | *huʔ "he"* | *xaḳām "smart"* |
|---|---|---|---|
| 1\|2 | 2\|4 | 3\|4 | 4\|0 |

"The child is smart."

The past and future forms of *hayā*, in contrast, are viewed as ordinary verbs, and form the root of the sentence they appear in (Example 21).

(21)
| *ha- "the"* | *menorā "lamp"* | *haytā "be"* | *semel "symbol"* | *xašūv "important"* |
|---|---|---|---|---|
| 1\|2 | 2\|3 | 3\|0 | 4\|3 | 5\|4 |

"The lamp was an important symbol."

Our scheme for spoken Hebrew uses the label `Acop` to mark the relation between the copula and its argument, the copula being the head (Approach A). Alternatively, we use the label `Xcop` to mark the relation in which the copula is the dependent (Approach B). Similarly, we use the labels `Aexs` and `Xexs` for the two approaches of the existential marker. We automatically converted the annotation of Approach A to Approach B. Example 22 depicts an utterance containing an existential marker annotated according to Approach A, where the head is the existential element. The result of its conversion to Approach B is shown in Example 23.

(22)

| *yeš "there_is"* | *le "to"* | *hi? "she"* | *dimyōn "imagination"* |
|---|---|---|---|
| exs | prep | pro:person | n |
| 1\|0\|Root | 2\|1\|Anonagr | 3\|2\|Aprep | 4\|1\|Aexs |

"She is imaginative."

(23)

| *yeš "there_is"* | *le "to"* | *hi? "she"* | *dimyōn "imagination"* |
|---|---|---|---|
| exs | prep | pro:person | n |
| 1\|4\|Xexs | 2\|1\|Anonagr | 3\|2\|Aprep | 4\|0\|Root |

"She is imaginative."

We trained MaltParser on data annotated with both approaches and evaluated the accuracy of parsing. The test set included 590 utterances (2,474 tokens) out of which 45 utterances (271 tokens) included at least one occurrence of either `Acop` (`Xcop` in Approach B) or `Aexs` (`Xexs` in Approach B) according to the gold standard annotation. Table 11 shows the accuracy of parsing with the two alternatives of *hayā* constructions. Table 12 shows the accuracy when evaluating the alternative approaches only on the 45 utterances (271 tokens) that contain the `Acop` (`Xcop`) or `Aexs` (`Xexs`) relations. Evidently, Approach A yields slightly better results, but the differences between the two approaches are not statistically significant ($p > 0.1$).

Copula-less constructions are rather common in Hebrew, and are far more common than utterances with a pronominal copula. Still, the training set of the in-domain evaluation scenario includes only 45 of them, just above 1 % of all utterances. Since nominal predicates are more often dependent on verbs, it is inconsistent to mark them as the root of utterances when they contain a *hayā* form.

### 7.2 The accusative marker

Another form that presents a challenge for dependency-based parsing is the Hebrew accusative marker, *ʔet*. This morpheme behaves much like a preposition: it can either introduce a lexical noun phrase or inflect with a pronominal suffix, and it expresses Verb-Patient relations, similarly to other prepositions in Hebrew. Although the analysis of *ʔet* as a preposition is conventional (Danon 2001), its distributional properties distinguish it from other prepositions: it is restricted and is expressed on the surface if and only if the following noun is definite. The syntactic status of *ʔet* is thus unclear, and two types of analysis are possible: one option is to treat the dependency between *ʔet* and the noun following it similarly to the relation specified for all other prepositions in our scheme, with the noun functioning as the argument of *ʔet*; the alternative is to treat the accusative marker as a dependent of the noun. In the first type of analysis we label the relation between the verb and the

**Table 11** Linguistic issues: *hayā* constructions

| Approach A | | Approach B | |
|---|---|---|---|
| UAS | LAS | UAS | LAS |
| 91.2 | 86.6 | 90.9 | 86.3 |

**Table 12** Linguistic issues: only utterances containing *hayā* constructions

| Approach A | | Approach B | |
|---|---|---|---|
| UAS | LAS | UAS | LAS |
| 90.0 | 88.6 | 87.4 | 85.2 |

accusative marker `Anonagr` and between *ʔet* and the nominal element `Aprep`, as in Example 24.

(24)

| *loʔ* "no" | *rocē* "want" | *ʔet* "ACC" | *ha-* "the" | *ṭipōt* "drops" |
|---|---|---|---|---|
| neg | part\|num:sg | acc | det | n\|num:pl |
| 1\|2\|Mneg | 2\|0\|Root | 3\|2\|Anonagr | 4\|5\|Mdet | 5\|3\|Aprep |

"(I) don't want the drops!"

In the second analysis, the nominal element is viewed as directly dependent on the verb (in a relation labeled `Anonagr`), with a relation labeled `Xacc` assigned to *ʔet*, as shown in Example 25.

(25)

| *lo ʔ* "no" | *rocē* "want" | *ʔet* "ACC" | *ha-* "the" | *ṭipōt* "drops" |
|---|---|---|---|---|
| neg | part\|num:sg | acc | det | n\|num:pl |
| 1\|2\|Mneg | 2\|0\|Root | 3\|5\|Xacc | 4\|5\|Mdet | 5\|2\|Anonagr |

The implication of the first analysis is that all constructions containing a verb followed by a preposition are treated systematically. This representation, however, results in inconsistency between definite and indefinite direct object constructions: in the latter case, since *ʔet* is not realized, the noun is directly dependent on the verb as `Anonagr`, and so these two parallel constructions are structurally distinct (Example 26). While the second analysis reflects consistency between definite versus indefinite constructions, it renders cases of explicit *ʔet* inconsistent with other prepositional phrases (e.g., Example 9 above).[7]

(26)

| *loʔ* "no" | *rocē* "want" | *ṭipōt* "drops" |
|---|---|---|
| neg | part\|num:sg | n\|num:pl |
| 1\|2\|Mneg | 2\|0\|Root | 3\|2\|Anonagr |

"(I) don't want drops!"

We automatically converted our original annotation scheme of *ʔet* as the head of the following nominal element (Approach A, Example 24) to the alternative

[7] In the annotation scheme for written Hebrew (Goldberg 2011), the marker *ʔet* is the head of the nominal element. According to Goldberg (2011), the reason for this decision is to adapt to cases where *ʔet* may appear whereas the subsequent nominal element is elided. These types of sentences are rather formal and we do not expect to encounter them in spoken language. No such constructions occur in our corpus.

**Table 13** Linguistic issues: accusative marker

| Approach A | | Approach B | |
|---|---|---|---|
| UAS | LAS | UAS | LAS |
| 91.2 | 86.6 | 90.6 | 86.1 |

scheme, where *ʔet* is a dependent of the nominal head (Approach B, Example 25). We trained MaltParser on data annotated in accordance with both approaches and evaluated the accuracy of parsing, again for the in-domain evaluation task in the All–All configuration. Table 13 shows the accuracy of parsing for the two alternatives. The test set contained 590 utterances (2,474 tokens) out of which 41 utterances (215 tokens) contained at least one occurrence of *ʔet*. We also show (Table 14) the accuracy when parsing only the 41 utterances that contain *ʔet*. While there is a small advantage to Approach A, the differences between the two approaches are not statistically significant ($p > 0.1$).

The small difference in accuracy between the two approaches is supported by the distribution in the training data of prepositional arguments of verbs (consistent with Approach A) and of indefinite nominal arguments of verbs (consistent with Approach B). Both are relatively common in the training data, perhaps explaining why neither approach has a significant advantage over the other.

### 7.3 Prepositions as dependents

In the annotation scheme we presented, prepositional phrases are headed by the preposition, labeled with the `Aprep` relation. An alternative analysis views prepositional phrases as headed by the nominal element, with the preposition depending on this head. In order to examine this alternative, we reversed the direction of all occurrences of the `Aprep` relation. In Approach A, this relation is headed by the preposition (including the accusative marker *ʔet* and the possessive marker *šel*). In Approach B, the nominal element is the head and the preposition depends on it in an `Xprep` relation. As a result, in Approach B the nominal element is directly dependent on the verb or noun that the preposition was dependent on in Approach A. Since the accusative marker is also affected by this transformation, this is an extension of the approach discussed in Sect. 7.2

Example 27 presents an utterance containing a prepositional phrase annotated according to Approach A, where the head is the preposition. The result of the conversion is shown in Example 28.

(27)

| *Siwān "Sivan"* | *tagīd "say"* | *le "to"* | *ʔīmaʔ "mom"* |
|---|---|---|---|
| n:prop | v | prep | n |
| 1\|2\|Aagr | 2\|0\|Root | 3\|2\|Anonagr | 4\|3\|Aprep |

"Sivan will tell mom."

(28)

| *Siwān "Sivan"* | *tagīd "say"* | *le "to"* | *ʔīmaʔ "mom"* |
|---|---|---|---|
| n:prop | v | prep | n |
| 1\|2\|Aagr | 2\|0\|Root | 3\|4\|Xprep | 4\|2\|Anonagr |

"Sivan will tell mom."

**Table 14** Linguistic issues: only utterances containing an accusative marker

| Approach A | | Approach B | |
|---|---|---|---|
| UAS | LAS | UAS | LAS |
| 95.3 | 92.6 | 94.9 | 90.7 |

**Table 15** Linguistic issues: prepositions

| Approach A | | Approach B | |
|---|---|---|---|
| UAS | LAS | UAS | LAS |
| 91.2 | 86.6 | 88.8 | 82.6 |

We trained the parser (again, on the in-domain task in the All–All scenario) with the alternative approaches and evaluated the accuracy of parsing. Table 15 compares the results of the original scheme (Approach A) with the alternative representation (Approach B).

There is a significant advantage to Approach A in both UAS and LAS ($p < 0.05$). This advantage can be explained by the decrease in accuracy of the `Anonagr` and `Mpre` relations, especially the latter. Recall that there is a direct relation between a verb and a direct object labeled `Anonagr`, regardless of the annotation approach. With Approach B, nominal elements in prepositional phrases are directly attached to verbs as well, with no intervening preposition that could indicate whether this nominal element is a modifier or an argument. Thus, nominal elements are possibly mistaken for arguments when they are in fact modifiers.

## 7.4 Prepositional arguments of verbs

A prepositional phrase following a verb can be either the verb's argument or its modifier. The decision is hard, even for human annotators, and the phenomenon is ubiquitous, with over 1,100 cases in our corpora. For example, the preposition *le-* "to" is considered a modifier of the verb (with the `Mpre` relation) in Example 29, but an argument (with the `Anonagr` relation) in Example 30.

(29)

| *ʔanī* "I" | *ʔaʕaṣē* "will-do" | *le-* "to" | *ʔatā* "you" | *baʕayōt* "problems" |
|---|---|---|---|---|
| pro:person\|pers:1&num:sg | v\|pers:1&num:sg | prep | pro:person | n |
| 1\|2\|Aagr | 2\|0\|Root | 3\|2\|Mpre | 4\|3\|Aprep | 5\|2\|Anonagr |

"I will cause you problems."

(30)

| *ʕaḳšāyw* "now" | *tagīdi* "say" | *le-* "to" | *ʔanī* "I" | *Siwān* "Sivan" |
|---|---|---|---|---|
| adv | v | prep | pro:person | n:prop |
| 1\|2\|Madv | 2\|0\|Root | 3\|2\|Anonagr | 4\|3\|Aprep | 5\|2\|Voc |

"Now tell me, Sivan."

**Table 16** Linguistic issues: introduction of the Averb relation

| Approach A | | Approach B | |
|---|---|---|---|
| UAS | LAS | UAS | LAS |
| 91.2 | 86.6 | 90.6 | 86.6 |

**Table 17** Linguistic issues: token representation

| Approach A | | Approach B | |
|---|---|---|---|
| UAS | LAS | UAS | LAS |
| 90.7 | 85.9 | 90.0 | 85.1 |

These subtle differences between prepositional arguments and modifiers of verbs lead to poor (labeled) recall and precision of the Mpre relation, as is evident in Table 3. In order to improve the overall accuracy of the parser, we altered the annotation scheme and created a new relation, Averb, that uniformly labels the attachment between a verb and a preposition, independently of whether the prepositional phrase is an argument or a modifier. The Mpre relation remains when a preposition is dependent on a noun, and the Anonagr relation now represents arguments of verbs which are not prepositions. We then trained the parser (again, on the in-domain task in the All–All scenario) and evaluated the accuracy of parsing. Table 16 compares the results of the original scheme (Approach A) with the alternative representation (Approach B).

In the All–All configuration, there seems to be a slight overall decrease in unlabeled accuracy, but the difference is not statistically significant. Closer inspection of the confusion matrices shows that in Approach B, the accuracy of the Averb relation is quite high (over 90 % in all individual metrics), and the accuracy of Anonagr actually improves slightly (compared with Approach A), but the accuracy of Mpre drops dramatically. Indeed, Mpre is confused with both Averb and Anonagr. We believe that a larger training corpus may be able to shed more light on this result.

### 7.5 Token representation

The last issue we examined with respect to a potential effect on parsing accuracy is token representation. A *morph-based* approach calls for the split of words into morphemes, the atomic units that are combined to create words, whereas a *word-based* approach refers to words as the minimal units of the language (Blevins 2006; Tsarfaty and Goldberg 2008). Recall that in order to reduce sparseness of data, we pre-processed the transcripts, splitting pronominal suffixes and inflected prepositions to separate tokens; this renders the representation of the corpora partially morph-based. Using our annotated data and a simple conversion script we can investigate the differences between the word-based approach and the morph-based

approach. More specifically, we examine the accuracy of parsing on data in which pronominal suffixes and inflected prepositions were *not* split.

We trained MaltParser on a version of the data that reflects such a word-based approach to token representation (Approach B). We compared the accuracy of parsing in this case to the accuracy obtained by MaltParser on the split data (Approach A), in the in-domain evaluation task and the All–All scenario. Table 17 shows the results. The morph-based representation is better, but the differences are not significant ($p > 0.1$). Evaluation restricted to utterances that include a split token reveals similar results.

### 7.6 Summary

In this section we examined various alternatives for relations in our annotation scheme, where both annotation approaches are linguistically plausible. Most of our evaluations showed no significant difference between the alternatives, except one (preposition as dependents) which showed a significant advantage to Approach A. The reason for these results could be that the data set is too small for any significant advantage for either approach. It is possible that the characteristics of these specific corpora—especially the relatively short utterances and the lack of morphological ambiguity—had an effect on the outcome, preventing any significant advantage. Most likely, which annotation approach of the two is selected is less important, as long as it is plausible (both linguistically and computationally) and consistent. Thus, we conclude that the annotation scheme proposed originally (as described in Sect. 4) is as suitable for our corpora as the alternative annotation approaches. This is the scheme we use for the released version of our corpora.

## 8 Conclusions

We presented a new annotation scheme for Hebrew spoken language, as part of the Hebrew section of the CHILDES database. The scheme handles some of the unique linguistic characteristics of Hebrew spoken language in general and child and child-directed language in particular. We showed that a parser trained on data annotated using this scheme achieves good results when parsing the same domain and is also adaptable to other domains, in spite of the relatively small data set available. It is worth noting that the transcriptions were sometimes erroneous or ill-formed; this had an effect on the quality of the syntactic analysis, and future acquired data should help in this respect.

We showed that both MaltParser and MEGRASP produced relatively good accuracy. In both evaluation scenarios, MaltParser proved to be the better of the two, thanks to parameter tuning done by MaltOptimizer. In future work, it would be interesting to experiment with more parsers, especially transition-based ones that may be more adequate for a corpus of short utterances such as ours, but also graph-based ones. Several such parsers exist that could be trained on our corpus (McDonald et al. 2005; Bohnet 2010; Zhang and Clark 2011).

We examined the differences between learning from CDS and CS. Within the same domain there was no significant difference and both configurations yielded relatively high accuracy. However, when parsing out-of-domain (and, crucially, on different children) there was a clear advantage to training on CDS. We attribute this to the simplicity of the CS in the in-domain scenario as well as to differences in CS between the training set and the test set (and within the test set) in the out-of-domain scenario. We conclude that, as expected, there is some difficulty to adapt CS from one domain to another (also recalling the age gap between the domains) whereas CDS is more stable and less varied across domains.

Working with MaltParser allowed us to evaluate the impact of features derived from the morphological tier of the corpora. Although the accuracy of parsing using the feature set without extended morphological data is quite high, due to the fact that the basic feature set was optimized by running MaltOptimizer and to the presence of a gold standard morphological tier, when we used detailed morphological information we were able to improve the accuracy of parsing even more. The best accuracy was exhibited using the morphological attributes 'gender', 'person' and 'number'. Future work in this area can embark on a more systematic approach that has the sole purpose of examining the contribution of morphological features. This includes extracting more morphological attributes other than those that were used in this work, as well as a more elaborate search for subsets of features that are derived from MaltParser data structures.

We examined different annotation approaches for a few linguistic constructions, such as *hayā* constructions and accusative marker constructions. In most cases, significant advantages to either approach were not revealed. This can be attributed to the characteristics of this corpus, and in particular its small size. Another possible explanation may very well be that as long as the annotation is consistent it can produce reasonable results, regardless of the specific annotation approach. It would be interesting to see if this is a cross-linguistic phenomenon, e.g., for copula constructions that are challenging in several languages.

We utilized the fact that the input to the syntactic process is a fully disambiguated gold standard morphological tier. An interesting extension is to evaluate the parser on data with a morphological tier that was created automatically. Apart from an obvious decrease in accuracy we expect that this may also introduce some different effects when examining feature sets or linguistic issues. Another extension to this work is parsing of Hebrew spoken language from other domains. We leave these research directions for future research.

**Acknowledgments** This research was supported by Grant No. 2007241 from the United States-Israel Binational Science Foundation (BSF). We are grateful to Alon Lavie for useful feedback, and to Shai Cohen for helping with the manual annotation. We benefitted greatly from the comments of two anonymous reviewers.

## Appendix 1: Dependency relations

Table 18 summarizes the basic dependency relations we define in this work. We list below all the relations, providing a brief explanation and a few examples.

**Table 18** Taxonomy of dependency relations

| Arguments | Modifiers | Others |
|---|---|---|
| Aagr | Mdet | Voc |
| Anonagr | Madj | Com |
| Aprep | Mpre | Coord |
| Ainf | Mposs | Srl |
| Acop | Mnoun | Enum |
| Aexs | Madv | RelCl |
| | Mneg | SubCl |
| | Mquant | Unk |
| | Msub | Punct |

AgreementArgument (Aagr) Specifies the relation between an argument and a predicate that mandates agreement.

(31) **ʔat** *"you"* **mešaqēret** *"lie"*
pro:person|gen:fm&num:sg part|gen:fm&num:sg
1|2|Aagr 2|0|Root
"You are lying!"

Non-agreementArgument (Anonagr) Specifies any argument of a verb which need not agree with the verb, including indirect arguments.

(32) *ʔanī "I"* *loʔ "no"* **rocē** *"want"* **ṭipōt** *"drops"*
pro:person|num:sg neg part|num:sg n|num:pl
1|3|Aagr 2|3|Mneg 3|0|Root 4|3|Anonagr
"I don't want drops."

Subordinate Clause (SubCl) Specifies the relation between a complementizer and the main verb of a subordinate clause.

(33) *ʔatā "you"* **rocē** *"want"* **še-** *"that"* *ʔanī "I"* **ʔesarēq** *"comb"*
pro:person|gen:ms&num:sg part|gen:ms&num:sg conj:subor pro:person|num:sg v|num:sg
1|2|Aagr 2|0|Root 3|2|Anonagr 4|5|Aagr 5|3|SubCl
*ʔet "ACC"* *ʔatā "you"*
acc pro:person
6|5|Anonagr 7|6|Aprep
"Do you want me to comb your hair?"

ArgumentOfPreposition(Aprep) Specifies the relation between a preposition and its argument.

NonFiniteArgument (Ainf) This relation is specified between a verb or a noun in the main clause and its non-finite verbal argument.

(34) *ʔaz "so"* **titēn** *"you-let"* *le- "to"* *ʔanī "I"* **laʕavōr** *"pass"*
adv v prep pro:person v
1|2|Com 2|0|Root 3|2|Anonagr 4|3|Aprep 5|2|Ainf
"So let me pass."

ArgumentOfCopula (Acop) Specifies the relation between a copula and its predicate (either nominal or adjectival). See Sect. 7.1 for further discussion regarding this relation.

| *Dani* "Dani" | **hayā** "be" | **šam** "there" |
|---|---|---|
| n:prop | cop | adv |
| 1\|2\|Aagr | 2\|0\|Root | 3\|2\|Acop |

"Dani was there."

ArgumentOfExistential (Aexs) Specifies a relation between an existential element and a nominal or adjectival predicate. See Sect. 7.1 for further discussion regarding this relation.

(36)

| *ʔavāl* "but" | *kaʔn* "here" | **ʔeyn** "is_not" | **yarōq** "green" |
|---|---|---|---|
| conj | adv | exs | adj |
| 1\|3\|Com | 2\|3\|Madv | 3\|0\|Root | 4\|3\|Aexs |

"But there is no green here."

Mdet Specifies a relation between a determiner and a noun.

(37)

| **ha-** "the" | **ʔōzen** "ear" |
|---|---|
| det | n |
| 1\|2\|Mdet | 2\|0\|Root |

"The ear."

Madj Specifies a relation between an adjective and a noun.

Mpre Specifies a relation between a dependent preposition and a head noun or a verb.

(38)

| **be-** "in" | **masrēq** "comb" | **ʔaxēr** "different" | **tistarqī** "you-comb_oneself" |
|---|---|---|---|
| prep | n\|gen:ms&num:sg | adj\|gen:ms&num:sg | v |
| 1\|4\|Mpre | 2\|1\|Aprep | 3\|2\|Madj | 4\|0\|Root |

"Comb your hair with a different comb."

Mposs Specifies a relation between a noun and a subsequent possessive marker, noted by the token '*šel*', headed by the noun.

(39)

| *ʔavāl* "but" | *ze* "this" | *ha-* "the" | **cad** "side" | **šel** "of" | **ʔanī** "I" |
|---|---|---|---|---|---|
| conj:coord | pro:den\|gen:ms&num:sg | det | n\|gen:ms&num:sg | prep | pro:person |
| 1\|4\|Com | 2\|4\|Aagr | 3\|4\|Mdet | 4\|0\|Root | 5\|4\|Mposs | 6\|5\|Aprep |

"But this is my side."

Mnoun Specifies a noun–noun relation, where the first noun, the head, is in the construct state.

(40)

| *holkīm* "walk" | *lanūax* "rest" | *be-* "in/at" | **cel** "shadow" | *ha-* "the" | **ʕec** "tree" |
|---|---|---|---|---|---|
| v | v | prep | n | det | n |
| 1\|0\|Root | 2\|1\|Ainf | 3\|2\|Mpre | 4\|3\|Aprep | 5\|6\|Mdet | 6\|4\|Mnoun |

"Going to rest in the tree's shadow."

Madv Specifies a relation between a dependent adverbial modifier and the verb it modifies.

(41)

| *ma* "what" | **ʕoṣīm** "do" | **ʕaḳšāyw** "now" |
|---|---|---|
| que | part | adv |
| 1\|2\|Anonagr | 2\|0\|Root | 3\|2\|Madv |

"What do we do now?"

Mneg Specifies a negation of a verb or a noun.

(42) *ʔanī̄ "I"* **loʔ** *"no"* **makīr** *"recognize"* *sipūr "story"* *ʕal "on"* *yanšūf "owl"*
pro:person|num:sg neg part|num:sg n prep n
1|3|Aagr 2|3|Mneg 3|0|Root 4|3|Anonagr 5|4|Mpre 6|5|Aprep
"I don't know a story about an owl."

Mquant Specifies a relation between a noun and a nominal quantifier, headed by the noun.

(43) *yeš "there_is"* *le- "to"* *ʔatā "you"* **raq** *"only/just"* **bēṭen** *"stomach"*
adv prep pro:person qn n
1|0|Root 2|1|Anonagr 3|2|Aprep 4|5|Mquant 5|1|Aagr
"You only have a stomach."

Msub Specifies a relation between a nominal element and a relativizer of a relative clause, headed by the nominal element. The main predicate of the subordinate clause is marked as the dependent of the relativizer with a `RelCl` relation.

(44) **balōn** *"balloon"* **še-** *"that"* **hitpocēc** *"explode"*
n|gen:ms&num:sg conj:subor v|gen:ms&num:sg
1|0|Root 2|1|Msub 3|2|RelCl
"A balloon that exploded."

Voc Specifies a vocative.

(45) **Asaf** *"Asaf"* , *","* **tedabēr** *"speak"*
n:prop , v
1|3|Voc 2|3|Punct 3|0|Root
"Asaf, speak up."

Com Specifies a communicator.

(46) **ʔaz** *"so"* *huʔ "he"* **nafāl** *"fall"* *mi- "from"* *po "adv"*
adv pro:persom|gen:ms&num:sg v|gen:ms&num:sg prep adv
1|3|Com 2|3|Aagr 3|0|Root 4|3|Mpre 5|4|Aprep
"So he fell from here."

Coordination (Coord) Specifies a coordination relation between coordinated items and conjunctions, most commonly *we-* "and", headed by the conjunction.

(47) *tagīdi "say"* *ze "this"* **hayā** *"be"* *kše- "when"* *hayīt "be"*
v pro:dem|gen:ms&num:sg v|gen:ms&num:sg conj:subor v|gen:fm&num:sg
1|7|Com 2|3|Aagr 3|7|Coord 4|3|Msub 5|4|SubCl
*qṭanā "small"* **we-** *"and"* *ʕaḵšāyw "now"* *ʔat "you"* **gdolā** *"big"*
adj|gen:fm&num:sg conj adv pro:person|gen:fm&num:sg adj|gen:fm&num:sg
6|5|Acop 7|0|Root 8|10|Madv 9|10|Aagr 10|7|Coord
"Tell me, this was when you were little and now you are grown up?"

Serialization (Srl) Specifies a serial verb.

(48) **bōʔi** *"come"* **nevaqēr** *"visit"* *maxār "tomorrow"* *ʔet "ACC"* *ʔīma ʔ "mother"*
v|form:imp&gen:fm&num:sg v adv acc n
1|2|Srl 2|0|Root 3|2|Madv 4|2|Anonagr 5|4|Aprep
*šel "of"* *hiʔ "she"*
prep pro:person
6|5|Mposs 7|6|Aprep
"Let's visit her mother tomorrow."

Enumeration (Enum) Specifies an enumeration relation.

(49) **ʔaxāt** *"one"* , *","* **štāyim** *"two"* , *","* **šalōš** *"three"* , *","* **ʔārbaʕ** *"four"*
num , num , num num
1|11|Enum 2|11|Punct 3|11|Enum 4|11|Punct 5|11|Enum 6|11|Punct 7|11|Enum
, *","* **xamēš** *"five"* , *","* **šeš** *"six"*
, num , num
8|11|Punct 9|11|Enum 10|11|Punct 11|0|Root
"One, two, three, four, five, six."

Unknown (Unk) Specifies an unclear or unknown word—most commonly a child invented word—which appears disconnected from the rest of the utterance and often functions as a filler syllable.

(50) *boʔ "come"* **naʕaṣe** *"do"* *parcūf "face"* *šel "of"* **e** *"e"* *ṣaqīt "bag"*
v v n prep chi n
1|2|Srl 2|0|Root 3|2|Anonagr 4|3|Mposs 5|2|Unk 6|4|Aprep
"Let's make a face of a bag."

Punctuation (Punct) Specifies a punctuation mark, always attached to the root.

(51) *ʔat "you"* *loʔ "no"* **criḳā** *"necessary"* *lefaxēd "be_scared"* , *","*
pro:person|gen:fm&num:sg neg adj|gen:fm&num:sg v ,
1|3|Aagr 2|3|Mneg 3|0|Root 4|3|Ainf 5|3|Punct
*xamudā "cute"* . *"."*
n|gen:fm&num:sg .
6|3|Voc 7|3|Punct
"You shouldn't be scared, sweety."

## Appendix 2: The effect of MaltOptimizer

### 2.1 The features chosen by MaltOptimizer

The Stack non-projective eager algorithm uses three data structures: a stack *Stack* of partially processed tokens; a queue *Input* which holds nodes that have been on Stack; and a queue *Lookahead* which contains nodes that have not been on Stack. This algorithm facilitates the generation of non-projective trees using a *SWAP* transition which reverses the order of the top two tokens on Stack by moving the top token on Stack to Input. The recommended feature set for the All–All configuration is depicted in Table 19. The features reflect positions within these data structures, where '0' indicates the first position. For example, the feature 'POSTAG (Stack[0])' specifies the part-of-speech tag of the token in the first position (i.e., the top) of the Stack data structure. The *NUM*, *GEN*, *PERS* and *VERBFORM* features are short for the number, gender, person and verb form morphological features, respectively. *Merge* and *Merge3* are feature map functions which merge two feature values and three feature values into one, respectively. *ldep* returns the leftmost dependent of the given node; *rdep* return the rightmost dependent; *head* returns the head of the node. For definitions of the rest of the features, refer to Nivre et al. (2007).

**Table 19** In-domain, All–All configuration, MaltOptimizer recommended feature set

| |
|---|
| POSTAG (Stack[0]) |
| POSTAG (Stack[1]) |
| POSTAG (Stack[2]) |
| POSTAG (Input[0]) |
| POSTAG (Lookahead[0]) |
| POSTAG (Lookahead[1]) |
| Merge(POSTAG (Stack[1]), POSTAG (Stack[0])) |
| Merge3(POSTAG (Stack[2]), POSTAG (Stack[1]), POSTAG (Stack[0])) |
| Merge3(POSTAG (Stack[1]), POSTAG (Stack[0]), POSTAG (Lookahead[0])) |
| Merge3(POSTAG (Stack[0]), POSTAG (Lookahead[0]), POSTAG (Lookahead[1])) |
| DEPREL, ldep(Stack[0]) |
| DEPREL, rdep(Stack[0]) |
| DEPREL, ldep(Stack[1]) |
| POSTAG, ldep(Stack[1]) |
| DEPREL, rdep(Stack[1]) |
| Merge3(POSTAG (Stack[1]), DEPREL, ldep(Stack[1]), DEPREL, rdep(Stack[1])) |
| FORM (Stack[0]) |
| FORM (Stack[1]) |
| FORM (Lookahead[0]) |
| LEMMA (Stack[1]) |
| LEMMA (Stack[2]) |
| NUM (Stack[0]) |
| GEN (Stack[0]) |
| PERS (Stack[0]) |
| VERBFORM (Stack[0]) |

## 2.2 MaltParser's default features

MaltParser's default parsing algorithm is Nivre arc-eager (Nivre 2003), which uses two data structures: a stack *Stack* of partially processed tokens and a queue *Input* of remaining input tokens. The feature set used by Nivre-arc is depicted in Table 20.

## 2.3 The features chosen by MaltOptimizer for the out-of-domain configuration

The feature set for the out-of-domain configuration suggested by MaltOptimizer is depicted in Table 21. The similarities between the suggested MaltOptimizer configurations of the in-domain and out-of-domain scenarios are not surprising, as the training set of the in-domain scenario is a subset of the training set of the out-of-domain scenario.

**Table 20** In-domain, All–All configuration, MaltParser default feature set

| |
|---|
| POSTAG (Stack[0]) |
| POSTAG (Stack[1]) |
| POSTAG (Input[0]) |
| POSTAG (Input[1]) |
| POSTAG (Input[2]) |
| POSTAG (Input[3]) |
| DEPREL (Stack[0]) |
| DEPREL, ldep(Stack[0]) |
| DEPREL, rdep(Stack[0]) |
| DEPREL, ldep(Input[0]) |
| FORM (Stack[0]) |
| FORM (Input[0]) |
| FORM (Input[1]) |
| FORM, head(Stack[0]) |

**Table 21** Out-of-domain, All–All configuration, MaltOptimizer recommended feature set

| |
|---|
| POSTAG (Stack[0]) |
| POSTAG (Stack[1]) |
| POSTAG (Stack[2]) |
| POSTAG (Lookahead[0]) |
| POSTAG (Lookahead[1]) |
| POSTAG (Lookahead[2]) |
| Merge(POSTAG (Stack[1]), POSTAG (Stack[0])) |
| Merge3(POSTAG (Stack[2]), POSTAG (Stack[1]), POSTAG (Stack[0])) |
| Merge3(POSTAG (Stack[1]), POSTAG (Stack[0]), POSTAG (Lookahead[0])) |
| Merge3(POSTAG (Stack[0]), POSTAG (Lookahead[ 0]), POSTAG (Lookahead[1])) |
| Merge3(POSTAG (Lookahead[ 0]), POSTAG (Lookahead[1]), POSTAG (Lookahead[2])) |
| DEPREL, ldep(Stack[0]) |
| DEPREL, rdep(Stack[0]) |
| DEPREL, ldep(Stack[1]) |
| POSTAG, ldep(Stack[1]) |
| DEPREL, rdep(Stack[1]) |
| Merge3(POSTAG (Stack[0]), DEPREL, ldep(Stack[0]), DEPREL, rdep(Stack[0])) |
| Merge3(POSTAG (Stack[1]), DEPREL, ldep(Stack[1]), DEPREL, rdep(Stack[1])) |
| Merge(POSTAG (Stack[1]), FORM (Lookahead[0])) |
| FORM (Stack[0]) |
| FORM (Stack[1]) |
| FORM (Lookahead[0]) |
| LEMMA (Stack[0]) |

**Table 21** continued

| |
|---|
| LEMMA (Lookahead[0]) |
| LEMMA (Stack[1]) |
| NUM (Stack[0]) |
| GEN (Stack[0]) |
| PERS (Stack[0]) |
| VERBFORM (Stack[0]) |
| NUM (Stack[1]) |
| GEN (Stack[1]) |
| PERS (Stack[1]) |
| VERBFORM (Stack[1]) |

## References


Albert, A., MacWhinney, B., Nir, B., & Wintner, S. (2014). The Hebrew CHILDES corpus: Transcription and morphological analysis. *Language Resources and Evaluation*.

Ballesteros, M., Herrera, J., Francisco, V., & Gervás, P. (2012). Analyzing the CoNLL-X shared task from a sentence accuracy perspective. *SEPLN: Sociedad Española Procesamiento del Lenguaje Natural*, *48*, 29–34.

Ballesteros, M., & Nivre, J. (2012). MaltOptimizer: A system for MaltParser optimization. In *Proceedings of the eighth international conference on language resources and evaluation (LREC'12)*, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Berman, R. A. (1978). *Modern Hebrew structure*. Tel Aviv: University Publishing Projects.

Berman, R. A., & Weissenborn, J. (1991). *Acquisition of word order: A crosslinguistic study*. Jerusalem, Israel: German-Israel Foundation for Research and Development (GIF); In Hebrew.

Blevins, J. P. (2006). Word-based morphology. *Journal of Linguistics*, *42*(4), 531–573. doi:10.1017/S0022226706004191.

Bohnet, B. (2010). Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 89–97). Stroudsburg, PA: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=1873781.1873792.

Brown, R. (1973). *A first language: The early stages*. Cambridge, MA: Harvard University Press.

Danon, G. (2001). Syntactic definiteness in the grammar of Modern Hebrew. *Linguistics*, *39*(6), 1071–1116. doi:10.1515/ling.2001.042.

de Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure trees. In *Proceedings of LREC-2006*. http://nlp.stanford.edu/pubs/LREC06_dependencies.pdf.

de Marneffe, M.-C., & Manning, C. D. (2008). The Stanford typed dependencies representation. In *COLING workshop on cross-framework and cross-domain parser evaluation*. pubs/dependencies-coling08.pdf.

Dromi, E., & Berman, R. A. (1982). A morphemic measure of early language development: Data from Modern Hebrew. *Journal of Child Language*, *9*, 403–424. ISSN 1469-7602. http://journals.cambridge.org/article_S0305000900004785.

Eryiğit, G., & Nivre, J., & Oflazer, K. (2008). Dependency parsing of Turkish. *Computational Linguistics*, *34*(3), 357–389. ISSN 0891-2017. doi:10.1162/coli.2008.07-017-R1-06-83.

Goldberg, Y. (2011). *Automatic syntactic processing of Modern Hebrew*. PhD thesis, Ben Gurion University of the Negev, Israel.

Goldberg, Y., & Elhadad, M. (2009). Hebrew dependency parsing: Initial results. In *Proceedings of the 11th international workshop on parsing technologies (IWPT-2009), 7–9 October 2009* (pp. 129–133). Paris, France: The Association for Computational Linguistics.

Hajič, J., & Zemánek, P. (2004). Prague Arabic dependency treebank: Development in data and tools. In *Proceedings of the NEMLAR international conference on Arabic language resources and tools* (pp. 110–117).

Haugereid, P., Melnik, N., & Wintner, S. (2013). Nonverbal predicates in Modern Hebrew. In S. Müller (Ed.), *The proceedings of the 20th international conference on head-driven phrase structure grammar*. CSLI Publications.

Kübler, S., McDonald, R. T., & Nivre, J. (2009). *Dependency parsing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Lembersky, G., Shacham, D., & Wintner, S. (2014). Morphological disambiguation of Hebrew: A case study in classifier combination. *Natural Language Engineering*. ISSN 1469-8110. doi: 10.1017/S1351324912000216.

MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk (3 ed)*. Mahwah, NJ: Lawrence Erlbaum.

Marton, Y., Habash, N., & Rambow, O. (2013). Dependency parsing of Modern Standard Arabic with lexical and inflectional features. *Computational Linguistics*, *39*(1), 161–194.

McDonald, R., Crammer, K., & Pereira, F. (2005). Online large-margin training of dependency parsers. In *Proceedings of the 43rd annual meeting on association for computational linguistics* (pp. 91–98). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1219840.1219852.

Ninio, A. (2013). Dependency grammar and Hebrew. In G. Khan (Ed.), *Encyclopedia of Hebrew language and linguistics*. Leiden: Brill.

Nir, B., MacWhinney, B., & Wintner, S. (2010). A morphologically-analyzed CHILDES corpus of Hebrew. In *Proceedings of the seventh conference on international language resources and evaluation (LREC'10)* (pp. 1487–1490). European Language Resources Association (ELRA), ISBN 2-9517408-6-7.

Nivre, J. (2003). An efficient algorithm for projective dependency parsing. In *Proceedings of the eighth international worskshop on parsing technologies (IWPT-2003)* (pp. 149–160).

Nivre, J. (2005). *Dependency grammar and dependency parsing*. Technical report, Växjö University.

Nivre, J. (2009). Non-projective dependency parsing in expected linear time. In *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing* (pp. 351–359). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=1687878.1687929.

Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., et al. (2007). The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL* (pp. 915–932), Prague.

Nivre, J., Hall, J., & Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC-2006* (pp. 2216–2219).

Nivre, J., Kuhlmann, M., & Hall, J. (2009). An improved oracle for dependency parsing with online reordering. In *Proceedings of the 11th international conference on parsing technologies (IWPT-09)* (pp. 73–76).

Plank, B. (2011). *Domain adaptation for parsing*. Ph.D. Thesis, University of Groningen.

Rosen, H. B. (1966). *Ivrit Tova (Good Hebrew)*. Kiryat Sepher, Jerusalem, in Hebrew.

Sagae, K., Davis, E., Lavie A., MacWhinney, B., & Wintner, S. (2010). *Morphosyntactic annotation of CHILDES transcripts. Journal of Child Language, 37*(3), 705–729. doi:10.1017/S0305000090 9990407.

Sagae, K., & Lavie, A. (2006). A best-first probabilistic shift-reduce parser. In *Proceedings of the COLING/ACL poster session* (pp. 691–698). Association for Computational Linguistics.

Sagae, K., & Tsujii, J. (2007). Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the CoNLL shared task session of EMNLP-CoNLL 2007* (pp. 1044–1050). http://www.aclweb.org/anthology/D/D07/D07-1111.

Seddah, D., Tsarfaty, R., & Foster, J., eds. (October 2011). *Proceedings of the second workshop on statistical parsing of morphologically rich languages*. Dublin, Ireland: Association for Computational Linguistics. http://www.aclweb.org/anthology/W11-38.

Sima'an, K., Itai, A., Winter, Y., Altman, A., & Nativ, N. (2001). Building a tree-bank of Modern Hebrew text. *Traitement Automatique des Langues*, *42*(2), 247–380.

Smrž, O., & Pajas, P. (2004). *MorphoTrees of Arabic and their annotation in the TrEd environment* (pp. 38–41). ELDA.

Tsarfaty, R., & Goldberg, Y. (2008). Word-based or morpheme-based? Annotation strategies for Modern Hebrew clitics. In *Proceedings of the sixth international conference on language resources and evaluation (LREC'08)*. European Language Resources Association (ELRA). ISBN 2-9517408-4-0. http://www.lrec-conf.org/proceedings/lrec2008/.

Tsarfaty, R., Nivre, J., & Andersson, E. (2012). Joint evaluation of morphological segmentation and syntactic parsing. In *Proceedings of the 50th annual meeting of the association for computational linguistics* (vol. 2, pp. 6–10).

Tsarfaty, R., Seddah, D., Goldberg, Y., Kübler, S., Candito, M., Foster, J., et al. (2010). Statistical parsing of morphologically rich languages (spmrl): What, how and whither. In *Proceedings of the NAACL HLT 2010 first workshop on statistical parsing of morphologically-rich languages* (pp. 1–12). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=1868771.1868772.

Tsarfaty, R., Seddah, D., Kübler, S., & Nivre, J. (2013). Parsing morphologically rich languages: Introduction to the special issue. *Computational Linguistics*, *39*(1), 15–22.

Wintner, S. (2004). Hebrew computational linguistics: Past and future. *Artificial Intelligence Review*, *21*(2), 113–138. ISSN 0269-2821. doi:10.1023/B:AIRE.0000020865.73561.bc.

Zhang, Y., & Clark, S. (2011). Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151. doi:10.1162/coli_a_00037.