Development of maximally reusable grammars: Parallel development of Hebrew and Arabic grammars

Tali Arad Greshler
University of Haifa

Livnat Herzig Sheinfux
University of Haifa

Nurit Melnik
The Open University of Israel

Shuly Wintner
University of Haifa

Proceedings of the 22nd International Conference on
Head-Driven Phrase Structure Grammar

Nanyang Technological University (NTU), Singapore

Stefan Müller (Editor)

2015

Stanford, CA: CSLI Publications

pages 27–40

Arad Greshler, Tali, Livnat Herzig Sheinfux, Nurit Melnik & Shuly Wintner. 2015. Development of maximally reusable grammars: Parallel development of Hebrew and Arabic grammars. In Stefan Müller (ed.), *Proceedings of the 22nd International Conference on Head-Driven Phrase Structure Grammar, Nanyang Technological University (NTU), Singapore*, 27–40. Stanford, CA: CSLI Publications.
DOI: 10.21248/hpsg.2015.2.

Abstract

We show how linguistic grammars of two different yet related languages can be developed and implemented in parallel, with language-independent fragments serving as shared resources, and language-specific ones defined separately for each language. The two grammars in the focus of this paper are of Modern Hebrew and Modern Standard Arabic, and the basic infrastructure, or core, of the grammars is based on "standard" HPSG. We identify four types of relations that exist between the grammars of two languages and demonstrate how the different types of relations can be implemented in parallel grammars with maximally shared resources. The examples pertain to the grammars of Modern Hebrew and Modern Standard Arabic, yet similar issues and considerations are applicable to other pairs of languages that have some degree of similarity.

1 Introduction

Our goal in this paper is to develop deep linguistic grammars of two different yet related languages. We show that such grammars can be developed and implemented in parallel, with language-independent fragments serving as shared resources, and language-specific ones defined separately for each language. The desirability of reusable grammars is twofold. From an engineering perspective, reuse of code is clearly parsimonious. From a theoretical perspective, aiming to maximize the common core of different grammars enables better identification and investigation of language-specific and cross-linguistic phenomena (see Müller, 2015, for further discussion of the motivation for parallel development of grammars).

A number of projects have adopted the notion of parallel development of different HPSG grammars with a common core. In the CoreGram project (Müller, 2015), grammars of ten different languages belonging to diverse language families are being implemented in parallel, using the TRALE system (Meurers et al., 2002).[1] Within the DELPH-IN consortium[2], two projects target languages of the same language family. The ZHONG [|] project (Fan et al., 2015a,b) models grammars of Chinese languages with a common core. It currently includes grammars of Mandarine Chinese and Cantonese. SlaviCore (Avgustinova & Zhang, 2009) is a resource that contains basic analyses known to occur cross-linguistically within the Slavic language family. SlaviClimb (Fokkens & Avgustinova, 2013), an extension of SlaviCore, is a dynamic engineering component, similar to the LinGO Grammar Matrix customization system (Bender et al., 2002), which supports the development of grammars for Slavic languages.

The two grammars in the focus of this paper are of Modern Hebrew (MH) and Modern Standard Arabic (MSA), two related languages, belonging to the Semitic

[†]This research was supported by THE ISRAEL SCIENCE FOUNDATION (grant No. 505/11).

[1]The set of languages includes: German, Danish, Persian, Maltese, Mandarin Chinese, Yiddish, English, Hindi, Spanish, and French.

[2]http://www.delph-in.net/

language family. HeGram, the MH grammar, is based on a starter grammar created with the Grammar Matrix customization system, but involves some major revisions to the "standard" grammar, mostly related to its novel argument-structure representation approach (see section 2.3). AraGram, the MSA grammar, is based on the infrastructure developed for HeGram.

Similarly to the CoreGram Project (Müller, 2015), the development process of the two grammars is "bottom-up". Namely, we examine linguistic phenomena in MH and MSA and identify generalizations which capture both grammars, on the one hand, and on the other, identify distinctions between the grammars. In some cases, to account for phenomena in one language we use a "*bottom-up with cheating*" approach (Müller, 2015); we reuse analyses that have been developed for one language to account for phenomena in the other language, as long as there is no contradicting evidence.

More generally, the parallel development of the two grammars revealed four types of relations that exist between the grammars of two languages:

(i) The two languages share some construction or syntactic phenomenon.

(ii) Some phenomenon is present in one language but is absent from the other.

(iii) The two languages share some construction, but impose different constraints on its realization.

(iv) Some phenomenon seems similar in the two languages, but is in fact a realization of different constructions.

While the challenge is to maximize the common parts of the grammars, it is important to be cautious with seemingly similar phenomena across the two languages. In some cases, as we will show, the solution is to define a shared construction with different language-specific constraints. Conversely, other cases are best accounted for by the definition of distinct constructions.

This paper demonstrates how the different types of relations can be implemented in parallel grammars with maximally shared resources. The examples pertain to the MH and MSA grammars, yet similar issues and considerations are applicable to other pairs of languages that have some degree of similarity.

2 Reusable grammars of Modern Hebrew and Modern Standard Arabic

2.1 Modern Hebrew and Modern Standard Arabic

Modern Hebrew is one of the official languages of Israel (along with Modern Standard Arabic). MH is a continuation of Biblical Hebrew (attested from 10th century BCE) and Mishnaic Hebrew (1st century CE). It was revived in Europe and Palestine toward the end of the 19th century and into the 20th century, influenced by

Yiddish, as well as Polish, Russian, German, English, Ladino and Arabic. It has had native speakers for about four generations.

Modern Standard Arabic is the literary standard of the Arab world. It is based on Classical Arabic (attested from the 6th century), which originated from Proto-Arabic or Old Arabic (attested from 7th century BCE). The modern period of Arabic dates approximately from the end of the eighteenth century with the spread of literacy, the concept of universal education, and journalism. MSA is the language of written Arabic media, e.g., newspapers, books, journals etc., and it is also the language of public speaking and news broadcasts on radio and television. However, MSA does not have native speakers, as Arabs are fluent in at least one dialect of spoken Arabic, which is their mother tongue, and only become literate in MSA in school (Ryding, 2005).

As MH and MSA are related, they exhibit a number of shared phenomena which can be attributed to their Semitic roots (see Figure 1). Nevertheless, since the languages diverged several millennia ago, the end grammars are quite different and do require language-specific accounts.

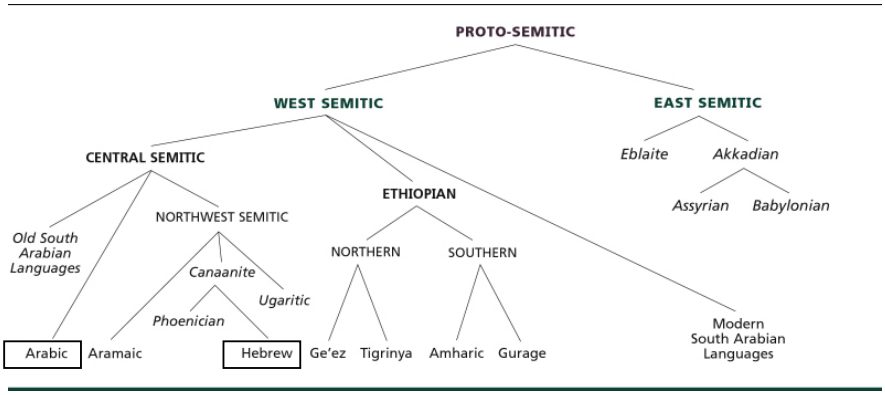

Figure 1: Semitic languages

2.2 Parallel Grammar Development

Our starting point is HeGram, a deep linguistic processing grammar of Modern Hebrew (Herzig Sheinfux et al., 2015). HeGram is grounded in the theoretical framework of HPSG and is implemented in the LKB (Copestake, 2002) and ACE systems. AraGram, the MSA grammar, utilizes the types defined in HeGram, as long as they are relevant for Arabic. In cases where the two languages diverge with respect to particular phenomena, language-specific types are defined in separate language-specific modules. More technically, the two grammars make extensive use of the ":+" operator provided by the LKB in order to define a type in a shared file, and to add language-specific constraints to its definition in distinct files (see (9)-(10) and (13)-(14) below).

The parallel development of the two grammars with their shared resources requires careful examination of the common and distinct properties of the two languages. Types, features, values and constraints can only be added or modified in a way that does not negatively affect the grammar of the other language. In order to guarantee that the changes introduced by the grammar of one language do not damage the grammar of the other we developed test suites of grammatical and ungrammatical sentences for both Arabic (160 sentences, 41 ungrammatical) and Hebrew (432 sentences, 106 ungrammatical) and test the grammar rigorously with [incr tsdb()] (Oepen, 2001). The test suites are continuously extended as analyses of more phenomena are introduced.

In the following sections we focus on a number of phenomena which illustrate different types of relations between the two languages and their implementation. We begin with a discussion of the way subcategorization is handled by the two grammars. We show that while semantic selection is found to be language-independent, the syntactic realization of arguments may be subject to language-specific constraints. Next, we describe the way the nominals of the two languages are represented in the lexical type hierarchy. In this case, the MH hierarchy is found to be a sub-hierarchy of the MSA one. Finally, we move on to clause structure. We discuss one case where two seemingly similar constructions are found to be licensed by distinct mechanisms, and another where the two languages share the same basic construction, yet impose different constraints on its realization.

2.3 Maximally shared resources: subcategorization

The architecture of HeGram embodies significant changes to the way argument structure is standardly viewed in HPSG. The main one is that it distinguishes between semantic selection and syntactic selection, and provides a way of stating constraints regarding each level separately. Moreover, one lexical entry can account for multiple subcategorization frames, including argument optionality and the realization of arguments with different syntactic phrase types (e.g., *want food* vs. *want to eat*). This involves the distribution of valence features across ten categories.[3] Each valence category is characterized in terms of its semantic role, as well as the types of syntactic phrases which can realize it (referred to as *syntactic realization classes*). Consequently, the semantic relations denoted by predicates consist of coherent argument roles, which are consistent across all predicates in the language.

Table 1 presents the ten valence categories used in HeGram, along with the corresponding semantic roles and syntactic realization phrases.[4] For example, Arg2 corresponds to the *Theme* semantic role, and can be realized in MH as an NP, an infinitive VP, a CP or a PP. The association between semantic roles and syntactic phrases is based on corpus investigation of MH which included at least 100 ran-

[3] Our restructuring of the VALENCE complex is inspired by Haugereid's packed argument frames (Haugereid, 2012).

[4] This architecture is similar in spirit to work done on Polish by Przepiórkowski et al. (2014).

domly selected examples of sentences containing each of the 50 most frequent verb lemmas in the 60-million token WaCky corpus of Modern Hebrew (Baroni et al., 2009).

Label	**Semantic Selection**	**Syntactic Realization**
Arg1	Actor, Perceiver, Causer	NP, PP
Arg2	Theme	NP, VP_{inf}, CP, PP
Arg3	Affectee, Benefactive, Malfactive , Recipient	NP, PP
Arg4	Attribute	AdjP, AdvP, PP, NP, $VP_{beinoni}$
Arg5	Source	PP
Arg6	Goal	PP
Arg7	Location	PP, AdvP
Arg8	Topic of Communication	PP
Arg9	Instrument	PP
Arg10	Comitative	PP

Table 1: Semantic roles and realization classes in HeGram

Each predicative lexical type in our grammars inherits from types which specify the possible semantic roles of its dependents and their possible syntactic realizations. As an example, consider the lexical type which licenses the MH verb *higiʕa* (‘*came*’).

(1) MH *higiʕa* (‘*came*’)

```
arg1-15-16-156_p_p := arg1_n & arg5_p & arg6_p &
[ SYNSEM.LOCAL.CAT.VAL.R-FRAME arg1-15-16-156 ].
```

The verb semantically selects three arguments: an *Actor* (arg1), a *Source* (arg5), and a *Goal* (arg6). Moreover, it requires that its *Actor* role be syntactically realized, yet allows for the omission of the latter two roles. This is captured by the value of its lexical type’s R(EALIZATION)-FRAME feature, *arg1-15-16-156*, which lists the different realization frames in which the verb can appear, separated by dashes. For example, *arg1* is an intransitive syntactic frame and *arg156* represents the realization of all three semantic arguments.

The syntactic realization of the semantic arguments is defined via inheritance. The lexical type in (1) inherits from three subtypes, each pertaining to one of its semantic arguments, and each determining the syntactic category of the phrases which realize that semantic role (noun, preposition, and preposition, respectively). The name of this type (i.e., *arg1-15-16-156_p_p*) reflects the different realization frames, as well as the syntactic category of its dependents (since Arg1 is always realized as an NP, its syntactic realization is omitted from the name of the type).

The MSA counterpart of *higiʕa* (‘*came*’) is *ʒaːʔa* (‘*came*’). The lexical type with which it is associated is illustrated in (2).

(2) MSA *ʒaːʔa* ('*came*')

```
arg1-15-16-156_p_np := arg1_n & arg5_p & arg6_np &
[ SYNSEM.LOCAL.CAT.VAL.R-FRAME arg1-15-16-156 ].
```

The only difference between the two types is in the realization of arg6. In MSA, *Goal* arguments can be realized by either NPs or PPs. This is captured in the type definition by the supertype *arg6_np*, which unlike its MH counterpart, *arg6_p*, also includes nouns as possible syntactic realizers. Consequently, the name of the type reflects this disjunctive value in its suffix (*np* instead of *p*). (3) and (4) demonstrate the realization of the *Goal* argument as a PP in MH and as an NP or a PP in MSA, respectively.

(3) *ha-qcinim higiʕu el ha-ʃagrirut ha-micrit*
*the-officers came.*3PM *to the-embassy the-Egyptian*
'The officers came to the Egyptian Embassy.'

(4) *ʒaːʔuː dˤ-dˤubaːtˤ-u s-sifaːrat-a l-misˤriyyat-a / ʔilaː*
*came.*3PM *the-officers*-NOM *the-embassy*-ACC *the-Egyptian*-ACC */ to*
s-sifaːrat-i l-misˤriyyat-i
the-embassy-GEN *the-Egyptian*-GEN
'The officers came to the Egyptian Embassy.'

The difference between the two languages with respect to the realization of *Goal* arguments required a slight modification of the MH schema shown in Table 1 to account for the MSA data. An additional modification involved the realization class of Arg2, since MSA uses the subjunctive in environments in which MH uses infinitives. Other than these slight language-specific details regarding syntactic realization, corpus investigations of the corresponding 50 MSA verbs using the 115-million token *arTenTen* corpus of Arabic (Arts et al., 2014) showed that they share the semantic frames identified for their MH counterparts, and consequently no changes were required in the overall argument representation scheme.

The non-standard argument structure representation of HeGram was found to be instrumental for distinguishing between general and language-specific properties of the grammar. In sum, the realization classes associated with different semantic roles are found to vary to some extent between languages while the semantic roles themselves appear to be more general.

2.4 Similarities between the languages: nominals in the lexical type hierarchy

MH and MSA are languages with rich, productive morphologies. Nouns in the two languages have natural or grammatical gender, and are marked for number. Adjectives decline according to a number-gender inflectional paradigm. Both categories are also morphologically marked for definiteness. Consequently, the grammars of

the two languages require an elaborate nominal type hierarchy, where types are cross-classified according to the three dimensions: NUMBER, GENDER and DEFINITENESS.[5]

The nominal type hierarchy described above is sufficient for MH, while MSA requires an extension of the hierarchy in order to account for two additional properties: *dual number* and *Case*. A sketch of the basic shared hierarchy, along with the MSA extensions (in the boxes) is given in Figure 2 . All MH nominals (i.e., nouns and adjectives) are instances of types which realize all the cross-classification combinations of the three MH-relevant dimensions (e.g., *sm-def-nom-lex*).

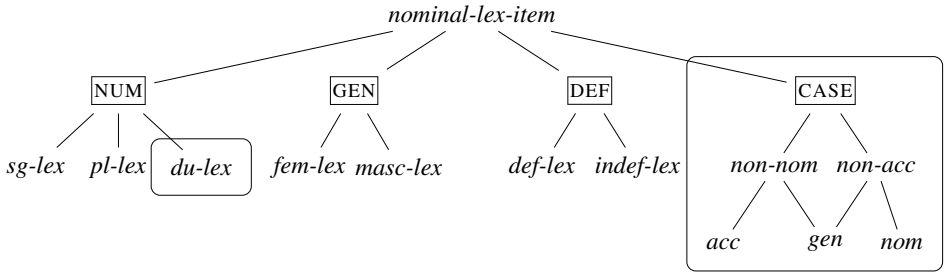

Figure 2: The nominal type hierarchy

Case in MSA is morphologically marked on all nominals by word-final vowels. Thus, in principle, all lexemes are cross-classified according to four dimensions: NUMBER, GENDER, DEFINITENESS, and CASE. The MH lexical entry for 'boy' (5) is an instance of a lexical type cross-classified according to three dimensions, whereas its MSA counterpart in (6) is an instance of a lexical type which is additionally classified as accusative (marked in a box).[6]

(5) MH *yeled* ('*boy*')

```
ild := indef-cmn-3sm-noun-lex &
       [ STEM < "ild" >,
       SYNSEM.LKEYS.KEYREL.PRED _boy_n_rel ].
```

(6) MSA *walad-an* ('*boy*')

```
wlda := indef-cmn-[acc]-3sm-lex &
        [ STEM < "wlda" >,
        SYNSEM.LKEYS.KEYREL.PRED _boy_n_rel ].
```

[5]Since the PERSON dimension is only relevant to nouns, not to adjectives, it is not presented here as part of the nominal type hierarchy.

[6]In our grammars we use 1:1 transliteration schemes for both MH and MSA. These schemes lack vowel representations as vowels are not represented in MH and MSA scripts. In glossed examples, however, we use phonemic transcription that includes vowels.

Note that the hierarchy below Case is structured to represent two different disjunctive groupings: non-nominative and non-accusative. As some MSA nominals are orthographically underspecified for Case, this intermediate level of the hierarchy was added as an engineering choice, in order to avoid repetition in the lexicon.

2.5 Deep and superficial similarities: clause structure

MH and MSA have different unmarked clause structures. In MH, SVO is the canonical word order, while in MSA it is VSO. Nevertheless, the unmarked clause order of MH is a marked structure in MSA, and vice versa. In addition, a notable property of MSA clauses is that subject-verb agreement depends on the subject position; verbs in SVO clauses exhibit full person-number-gender agreement with the subject, while in VSO clauses number agreement is suppressed and the verb is invariably singular. This is not the case in MH, where the verb fully agrees with the subject regardless of its position.[7]

2.5.1 Superficial similarities, different constructions: SVO

The SVO clauses of the two languages are remarkably similar; the finite verb exhibits full person-number-gender agreement with the subject which precedes it. As examples, consider the following SVO clauses in MH (7) and MSA (8).

(7) *ha-yeladim axlu et ha-leħem*
*the-boys ate.*3PM ACC *the-bread*
'The boys ate the bread.'

(8) *ʔl-ʔawlaːd-u ʔakaluː l-xubz-a*
the-boys-NOM *ate.*3PM *the-bread*-ACC
'The boys ate the bread.'

While superficially almost identical, the SVO clauses of the two languages are given distinct analyses in our grammars. The unmarked MH SVO clause is licensed by a *subject-head-phrase* phrase type. The syntactic tree pertaining to example (7) is shown in Figure 3.

The syntactic structure of VSO and SVO Arabic clauses has been thoroughly discussed in the literature (Fassi Fehri, 1993; Mohammad, 2000; Aoun et al., 2010; Alotaibi & Borsley, 2013, among others). The main challenge is the agreement asymmetries between SVO and VSO clauses. The analysis put forth by Aoun et al. (2010) and elaborated and cast in HPSG by Alotaibi & Borsley (2013) proposes that clause structure in MH is invariantly VSO, where number agreement is supressed. Full agreement on the verb is found only in SVO structures and in cases of *pro*-drop. In both constructions, they claim, the manifestation of full agreement is triggered by the existence of a post-verbal *pro* subject. In SVO structures this

[7] Exceptions to this generalization are colloquial verb-initial constructions (e.g., Melnik, 2006).

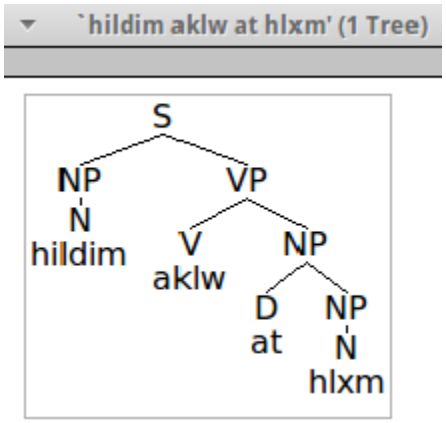

Figure 3: SVO in Modern Hebrew

pro subject is a resumptive pronoun which is associated with what looks like a preverbal subject, but is in fact a topic. The fact that subject arguments in SVO clauses are required to be definite supports this analysis.

We adopt the topic analysis of SVO clauses for MSA, and model such clauses as instances of a *filler-head-phrase* type. The syntactic tree of example (8) is given in Figure 4. Consequently, the *subject-head-phrase* type is defined only in the MH grammar.

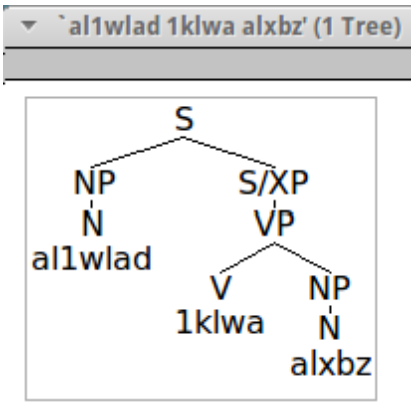

Figure 4: SVO in Modern Standard Arabic

The types dedicated to long-distance dependency constructions are shared by the two languages. Nevertheless, the MH grammar is more restrictive with regard to topicalization; it confines the phenomenon only to non-subjects in order to avoid vacuous structural ambiguity with SVO clauses, and restricts subject extraction only to *wh*-questions. MSA, on the other hand, allows all dependents to

be topicalized, but restricts subject extraction to definite subjects. This disparity is implemented by using *extracted-subject-phrase* as a shared resource, and adding language-specific constraints in each grammar. This is easily done in the LKB by using the ":+" operator.

(9) MH: Subject extraction only occurs with questions

```
extracted-subj-phrase :+
[ SYNSEM.LOCAL.CONT.HOOK.INDEX.SF ques ].
```

(10) MSA: Extracted subjects must be definite

```
extracted-subj-phrase :+
[ SYNSEM.NON-LOCAL.SLASH.LIST.FIRST.CAT.HEAD.DEF + ].
```

The use of a shared type reflects the generalization that both languages have subject extraction and allows maximal reusability of the type hierarchy below the shared *extracted-subject-phrase* type.

2.5.2 Different constraints on the same construction: VSO

VSO constructions in both MH (11) and MSA (12) have a *head-subj-comp-phrase* phrase type, and thus its type definition is shared.[8]

(11) *et ha-leħem axlu ha-yeladim*
ACC *the-bread ate.*3PM *the-boys*
'The bread, the boys ate it.'

(12) *ʔakala l-ʔawlaːd-u l-xubz-a*
*ate.*3SM *the-boys*-NOM *the-bread*-ACC
'The boys ate the bread.'

There are, however, additional language-specific constraints which further restrict this clause type. In Hebrew, VSO constructions are only licensed in a V2 configuration, where some clause-initial material precedes the verb, e.g., *et ha-leħem* ('*ACC the-bread*') in (11). An additional Hebrew-specific constraint restricts this phrase type only to cases where the verb has undergone extraction (13). The MSA grammar, on the other hand, imposes its own language-specific constraint: the verb is invariably singular (14).

(13) MH Head Subject Complement constraint

```
VS-basic-head-subj-phrase :+
[ HEAD-DTR.SYNSEM.NON-LOCAL.SLASH 1-dlist ].
```

[8] Since only unary and binary branches are employed in the grammar, the *head-subj-comp-phrase* phrase type is implemented with two types: *head-subject* and *head-comp* (with a realized subject).

(14) MSA Head Subject Complement constraint

```
VS-basic-head-subj-phrase :+
[ HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.CNCRD png-s ].
```

This mechanism, where two languages share a construction and each language adds a different constraint to it without damaging the rest of the hierarchy, is an excellent utilization of HPSG type hierarchies, allowing maximal reusability in developing and implementing two grammars with a common core.

3 Current status and future prospects

We have adapted HeGram (Herzig Sheinfux et al., 2015) to Arabic along the lines discussed above. AraGram currently covers a plethora of syntactic phenomena, including Case marking, subject-verb and noun-adjective agreement, SVO and VSO word order, relatively free complement order, multiple subcategorization frames, selectional restrictions of verbs on their PP complements, topicalization, passive and unaccusative verbs. Many of these phenomena required only minor adaptations to the Hebrew grammar. Therefore, the development of AraGram took only several weeks (excluding corpus investigation and literature review). For comparison, the development of HeGram to its stage when we started developing AraGram took about a year. AraGram currently shares 95.5% of its types with HeGram, while HeGram currently shares 99.2% of its types with AraGram.

The development of AraGram is ongoing. In the near future, we will focus on additional constructions, including wh-questions, control, raising, the copular construction, and multi-word expressions. We also intend to work on automatic translation between the languages using semantic MRS transfer and generation.

References

Alotaibi, Mansour & Robert D. Borsley. 2013. Gaps and resumptive pronouns in Modern Standard Arabic. In Stefan Müller (ed.), *Proceedings of the 20th international conference on hpsg*, 6–26. Stanford: CSLI Publications.

Aoun, Joseph E., Elabbas Benmamoun & Lina Choueiri. 2010. *The syntax of Arabic*. Cambridge University Press.

Arts, Tressy, Yonatan Belinkov, Nizar Habash, Adam Kilgarriff & Vit Suchomel. 2014. arTenTen: Arabic corpus and word sketches. *Journal of King Saud University-Computer and Information Sciences* 26(4). 357–371.

Avgustinova, Tania & Yi Zhang. 2009. Parallel grammar engineering for Slavic languages. Workshop on Grammar Engineering Across Frameworks at the 2009 ACL/IJCNLP conference, Singapore.

Baroni, Marco, Silvia Bernardini, Adriano Ferraresi & Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources And Evaluation* 43(3). 209–226.

Bender, Emily M., Dan Flickinger & Stephan Oepen. 2002. The grammar matrix: an open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Coling-02 workshop on grammar engineering and evaluation*, 1–7. Morristown, NJ, USA: Association for Computational Linguistics. doi:http://dx.doi.org/10.3115/1118783.1118785.

Copestake, Ann. 2002. *Implementing typed feature structure grammars*. Stanford: CSLI Publications.

Fan, Zhenzhen, Sanghoun Song & Francis Bond. 2015a. Building Zhong [|], a Chinese HPSG shared-grammar. In Stefan Müller (ed.), *Proceedings of the 22nd international conference on Head-Driven Phrase Structure Grammar, Singapore*, 97–110. Stanford, CA: CSLI Publications.

Fan, Zhenzhen, Sanghoun Song & Francis Bond. 2015b. An HPSG-based shared-grammar for the Chinese languages: Zhong [|]. In *Proceedings of the grammar engineering across frameworks (GEAF) 2015 workshop*, 17–24.

Fassi Fehri, Abdelkader. 1993. *Issues in the structure of Arabic clauses and words*. Dordrecht: Kluwer.

Fokkens, Antske & Tania Avgustinova. 2013. SlaviCLIMB: Combining expertise for Slavic grammar development using a metagrammar. In *Workshop on high-level methodologies for grammar engineering ESSLLI 2013*, 87.

Haugereid, Petter. 2012. A grammar design accommodating packed argument frame information on verbs. *International Journal of Asian Language Processing* 22(3). 87–106.

Herzig Sheinfux, Livnat, Nurit Melnik & Shuly Wintner. 2015. Representing argument structure in computational grammars. Submitted.

Melnik, Nurit. 2006. A constructional approach to verb-initial constructions in Modern Hebrew. *Cognitive Linguistics* 17(2). 153–198.

Meurers, W. Detmar, Gerald Penn & Frank Richter. 2002. A web-based instructional platform for constraint-based grammar formalisms and parsing. In *Proceedings of the acl workshop on effective tools and methodologies for teaching NLP and CL*, 18–25.

Mohammad, Mohammad A. 2000. *Word order, agreement, and pronominalization in Standard and Palestinian Arabic*. Amsterdam: John Benjamins.

Müller, Stefan. 2015. The CoreGram project: Theoretical linguistics, theory development and verification. *Journal of Language Modelling* 3(1). 21–86. http://hpsg.fu-berlin.de/~stefan/Pub/coregram.html.

Oepen, Stephan. 2001. [incr tsdb()] — competence and performance laboratory. User manual. Technical report Computational Linguistics, Saarland University Saarbrücken, Germany.

Przepiórkowski, Adam, Elżbieta Hajnicz, Agnieszka Patejuk, Marcin Woliński, Filip Skwarski & Marek Świdziński. 2014. Walenty: Towards a comprehensive valence dictionary of Polish. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the ninth international Conference on Language Resources and Evaluation, LREC 2014*, 2785–2792. Reykjavik, Iceland: ELRA. http://www.lrec-conf.org/proceedings/lrec2014/index.html.

Ryding, Karin C. 2005. *A reference grammar of Modern Standard Arabic*. Cambridge university press.