

Extracting HPSG Lexicon from Arabic VerbNet

Noureddine Loukil and Kais Haddar

Faculty of Sciences, University of Sfax, Tunisia
MIR@CL Laboratory, University of Sfax, Tunisia
noureddine.loukil@isimsf.rnu.tn
kais.haddar@fss.rnu.tn

Abstract. This paper presents the construction of a HPSG lexicon of Arabic verbal entities, automatically inferred from the Arabic VerbNet, a large coverage verb lexicon where verbs are classified using syntactic alternations. We discuss the main verb specification along with the relation of the syntactic and semantic levels of representation within the HPSG framework. Extensive analysis of the Arabic VerbNet classes has led to the adoption of a finite set of mapping rules between AVN classes and HPSG subcategorization and semantics descriptions covering the majority of the verbal tokens. We employed the adopted mapping rules to extract the syntactic and semantic data from AVN and finally, we describe the resulting TDL descriptions in which the lexicon has been encoded.

Keywords: Arabic VerbNet, HPSG, Lexicon, Arabic, Automatic Extraction

1 Introduction

The SubCategorization Frame (SCF) of a verb is a specification of the number and type of its complements (objects and oblique arguments) but it evolved to include the specification of the subject in some modern theories like the Head driver Phrase Structure Grammar (HPSG) [1]. In this regard, it becomes more close to the concept of valency introduced in the dependency grammar formalism [2] and generally employed to encompass subcategorization, argument structure, selectional preferences on arguments, and mapping rules which relate syntactic with semantic representation, that is, the syntax semantics interface. SCFs are generally embedded in the syntactic lexicon which is considered among the most valuable resources for many natural language processing tasks [3]. It is widely accepted that the verb encapsulates the main part of a sentence meaning, and thus the description of the syntactic behavior of verbs is a crucial goal for linguists. Thus, creating a syntactic lexicon of verbs with the valency information is obviously substantial for many tasks within the Natural Language Processing (NLP) field.

Several linguistic resources embedding valency information have been created manually and required consequent effort and time. Those resources are mainly

theory-independent so they can be used and shared by several communities such as VerbNet (VN) [4] which is considered to be the largest verb lexicon for English. The Arabic VerbNet [5] is a large coverage verb lexicon where verbs are classified using syntactic alternations [6] and adopting the general structure and content elements of the English VN. Levin's alternations are tested on Arabic verbs but other Arabic-specific alternations are collected. Fine-grained classes of semantic descriptions are identified and verbs are affected to a class based on how they lexicalize the meaning components of the class and how they syntactically realize these components.

Although VN and AVN were designed to be domain and theory independent, many attempts have tried to lexically connect them with other resources such as Propbank [7], WordNet [8], Xtag [9] [10], and FrameNet [11] [12]. In this context, the Semlink project [13] [14] aims to create a mapping between PropBank, FrameNet, WordNet and VN, thus allowing these resources to incorporate and to form an enhanced resource. Moreover, the project includes detailed token mapping of verbs to their VN classes. A set of concept-to-concept mappings are employed to combine the different information provided by these different lexical resources. Although the importance of such project for data enhancement and resource interoperability, it doesn't provide a way to use the SCF and semantics within VN for NLP in theory dependent frameworks. For this reason, we propose to transform the VN valency information to a theory based lexicon so we can effectively use it in tasks such text parsing.

In this paper, we detail the automatic building of a verb lexicon of Modern Standard Arabic (MSA) for the HPSG framework, extracted from AVN verbal classes. The building process was driven by a finite set of mapping rules to transform the linguistic information embedded in AVN to HPSG lexical entries. In the following sections, we present the AVN class anatomy and the HPSG lexical specification and we outline the methodology we developed in which we associate, for each concept within AVN, a parallel concept in the HPSG theory. Mapping rules for syntax and semantics projection are presented and discussed. Finally, we discuss the generated lexicon and we present some issues and some improvements that can enhance the mapping process.

2 Related Work

Arabic is still considered among the less resources languages despite the big interest and the important number of projects that were addressed to it in the recent years. The lack of resources is especially apparent in the field of grammar lexicons. So, manually or automatically developing such resources is of great interest to the Arabic NLP community. [15] reported the development of a manually created SCF resource for Lexical Functional Grammar (LFG) developed for the Arabic LFG parser. It contains 64 syntactic frame, 2,709 lemmas types, and 2,901 lemma frame types, with an average of 1.07 frames per lemma. Additional information about control and specific prepositions with obliques is embedded in the resource. For the HPSG formalism, experimental lexical resource have been

manually developed [16] in TDL within the LKB system. Yet, the coverage of this resource has not been reported.

Automatic acquisition of linguistic resource becomes more than an option by the development of large coverage treebanks for Arabic such as the Penn Arabic TreeBank (PATB)[17]. [18] automatically extract Arabic SCFs by utilizing the automatic Lexical-Functional Grammar (LFG) f-structure annotation algorithm for the PATB developed in [19]. They reported the extraction of 240 frame types for 3,295 lemmas types, with 7,746 lemma frame types (for verbs, nouns and adjectives), averaging 2.35 frames per lemma. There have been also efforts aimed at extracting TAGs from the PATB [20] yet without emphasizing on the lexicon.

3 VerbNet Classes

VerbNet is composed of a list of verb classes. Each VN class contains a set of syntactic descriptions, or syntactic frames (see Fig. 1), containing the possible surface realizations of the argument structure for constructions such as transitive, intransitive, prepositional phrases or resultatives. SElectional Semantic RESTRictions (SELRESTR in AVN) such as “animate” or “human” are used to constrain the types of thematic roles allowed by the arguments, and further restrictions such as syntactic restrictions (SYNRESTR in AVN) may be imposed to indicate the syntactic nature of the constituent likely to be associated with the thematic role. Syntactic frames may also specify the prepositions that are allowed. Each frame is also associated with explicit semantic information, expressed as a conjunction of boolean semantic predicates such as “motion”, “contact”, or “cause”. Each semantic predicate is associated with an event variable E that allows predicates to specify when in the event the predicate is true (start(E) for preparatory stage, during(E) for the culmination stage, and end(E) for the consequent stage).

Table 1 shows a complete entry for a class in Arabic VerbNet 1.1 class \$axara-1 (snort). The 13 verbs in this class use two semantic roles: Agent with selectional restriction “animate” and Theme. The class contains two syntactic frames (intransitive and V NP NP).

Table 1. \$axara-1 class from Arabic VerbNet v1.1

Class: \$axara-1		
Members: 13, Frames: 2		
Members: \$axara, DaraTa, xadafa, EafaTa, xabaja, Eafaqa, EaTasa, fax`a, taja\$`aOa, fasaA, lahava, \$ahaqa, Oaz`a		
Roles:		
Agent [+animate]		
Theme		
Frame	Example	Role assignment
Intransitive	\$axara AlmariyDu. (The patient snorts)	Verb,Agent
V NP NP	\$axara AlmariyDu \$axiyrAF EaAliyAF. (The patient snorts a loud snoring)	Verb,Agent,Theme

```
<FRAME>
<DESCRIPTION primary="V NP fiy deverbal" secondary="deverbal" xtag=""/>
<EXAMPLES>
<EXAMPLE transliteration="$araEa Almuhanodisuwna fiy taxoTiyT AlmaloEab."/>
</EXAMPLES>
<SYNTAX>
  <VERB/>
  <NP value="Agent"> <SYNRESTRS/> </NP>
  <PREP value="fiy"> <SELRESTRS/> </PREP>
  <NP value="Theme">
    <SYNRESTRS>
      <SYNRESTR Value="+" type="deverbal"/>
    </SYNRESTRS>
  </NP>
</SYNTAX>
</FRAME>
```

Fig. 1. AVN Frame from the AVN class badaOa-1, subclass badaOa-1-2

4 The HPSG lexical specification

HPSG [1] is a formal model for the natural language. This model is based on three main components:

- A representation of the linguistic objects as feature structures, making profit of the powerful mathematical apparatus applied on this type of structures and especially the unification operation,
- A specific organization of the space of linguistic objects manipulated by the grammar (the lexicon), designed around the notion of *type*,
- Principles or constraints allowing combination of signs to produce phrases.

The two first component belong to the field of lexicon. In fact, lexical entries are encoded as typed feature structures and types are organized in hierarchies integrated to the lexicon. The HPSG lexicon is designed in extension thus all the forms of a certain lexeme have to be individually specified. Many attempts tried to cope with this redundancy. Flickinger in his dissertation [21] proposed the concept of “lexical rules” to extend the original grammar with diverse rules acting at the lexicon level to automatically infer new lexical entries by inflexion or case altering (for example, deduce passives of verbal entities). This generative tool was very efficient in practice although it was in contradiction with the declarative nature of the targeted grammars.

All the benefits of enforcing feature structures to be typed [22] comes essentially in the verification of grammatical coherence. In fact, in this model, it is impossible to introduce features not appropriated for types. Thereby, all feature structures must be typed. Diverse feature structures are constructed by unification incrementally by applying HPSG Immediate Dominance schemata. Principles (and ID schemata) are considered as the core definition of the grammar. They describe the rules of combination that permit the unification of simple signs to form more complex signs.

4.1 Linguistic objects

Pollard et Sag in [1] adopt a more general vision of the notion of sign proposed by De Saussure [23]. In fact, they consider that the objects of a language (words, phrases or all syntactic constructions) are different types of the same conceptual sign. Thus, each sign is a descriptive construction integrating several linguistic components : phonology, morphology, syntax, semantics and context. The union of all this heterogeneous information makes from the sign a very specific object. Fig. 2 shows the description used for signs of type “ word ”. All the lexemes in a lexicon regardless of their nature obey to this typical specification. They include various simple and complex features more or less used in practice.

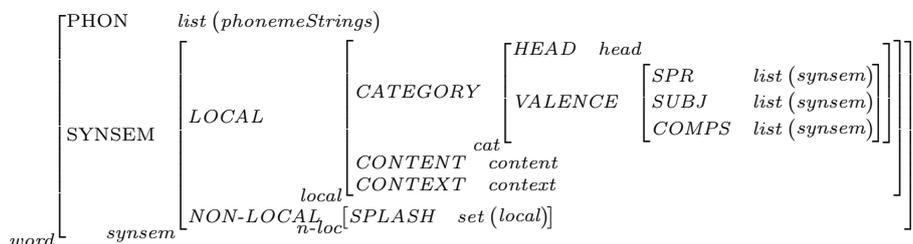


Fig. 2. Typical feature structure of a *word*

- “PHON” : A morphologic description as a value for “ PHONOLOGY ” feature expressed as a list of phonemic tokens.
- “SYNSEM” : Embed the totality of the syntactic and semantic properties of the sign except properties describing unbounded dependencies.
- “LOC” : Describes the sign properties within its actual syntactic construction.
- “NONLOC” : Describes unbounded dependencies that represent relations between the sign and other constituents not belonging to its immediate neighborhood. in this work, We don’t cope with this feature as we express only immediate surface realization of arguments.
- “HEAD” : Describes Part Of Speech properties (type, form, agreement, . . .) along with all other properties without a relation with syntax and semantics.
- “CONTENT” : Describes semantic properties of the sign by specifying the type of the semantic relation and its arguments expressed as a predicative structure.
- “SUBJ” : Describes the subject subcategorized by the sign. Its value is usually composed by just one element.
- “COMPS” : The value of this feature is a list of complements subcategorized by the sign.
- “SPR” : Describes the specifiers subcategorized by the sign.
- “ARG-S” : Describes the argument structure by order of obliquity. It corresponds to the union of feature SUBJ, COMPS and SPR.

- “INDEX” Describes agreement features. Generally used as a reference to the sign.
- “NUCLEUS” describe predicative structure for verbal signs

The VALENCE feature is a specification of the subcategorization of a word. It gives the list of the *synsems* of signs that the word can combine with to make a phrase. the order of those *synsems* is not the surface order but an order of obliqueness defended by many grammatical theories: Subjects first, followed by object complements, prepositional, verbal et adjunct complements.

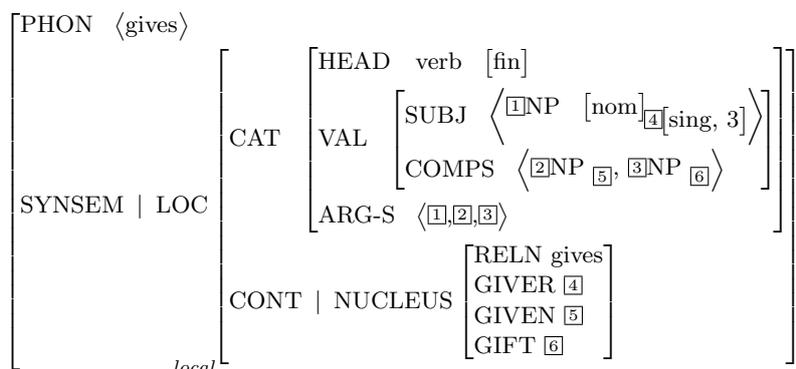


Fig. 3. Lexical Entry of verb “ gives”

The CONTENT Feature in a specification of the semantics of the sign. The representation of semantic relations are borrowed from situational semantics [24]. Semantic roles are assigned within the CONTENT of a lexical entry. A role is assigned by means of structure sharing between the INDEX feature of an element in the SUBJ/COMPS list and the value of some attribute of the sign’s CONTENT value.

4.2 Type Description Language

TDL [25] is a type description language for HPSG. Its syntax provides type definitions organized in hierarchies propagating inheritance relations from root types to instances. Moreover it expresses lexical entry definitions (instances of types) and lexical rules definitions. TDL is a practical language used to implement many HSPG grammars within the LKB [26] system and especially those using the LinGo Grammar Matrix framework [27]. We used a small part of the TDL syntax to define types, subtypes, feature structures, lists and coreferences. A simplified syntax for those definitions is given in Table 2. For reasons of efficiency and interoperability, we declare all the new defined types on top of a subset of the HPSG grammar Matrix types (just the *sign* type as defined in Fig. 4), So the produced resources can be used for processing in the LKB system.

Table 2. TDL definitions

Definition	TDL syntax
Type	$type \rightarrow body\{, option\}^*$
Subtype	$Subtype \rightarrow Supertype \& constraints$
Attribute value Matrix	$avmtype \rightarrow := body\{, option\}^*$
Feature	$feature \rightarrow [attrval\{, attrval\}^*]$
Attribute value	$attribute \rightarrow [attribute\{ : restriction\}]$
List	$list \rightarrow \langle \rangle \langle nonemptylist \rangle$
Coreference	$coreference \rightarrow \#identifier$

$$\begin{aligned}
synsem &:= avm \& \left[\begin{array}{l} \text{OPT} \quad \text{bool, DEFOPT} \quad \text{bool,} \\ \text{LOCAL} \quad \text{mod-local, NON-LOCAL} \quad \text{non-local-min} \end{array} \right] \\
sign &:= basic-sign \& \left[\begin{array}{l} \text{SYNSEM} \quad \text{synsem, ARGS} \quad \text{list,} \\ \text{INFLECTED} \quad \text{bool, ROOT} \quad \text{bool} \end{array} \right]
\end{aligned}$$

Fig. 4. TDL lexical types

5 Mapping Methodology

The method we are using to transform the verb classes provided by AVN to a HPSG lexicon is based on a set of mapping rules in a similar way to the approach followed in the SemLink project [13]. Each rule performs a projection of a particular concept within AVN to a parallel concept within the HPSG lexicon as shown in Table 3. Each AVN verbal class represents several verbs and will be inflated to several HPSG lexical entries. Moreover, the simple HPSG lexical entry represents just one specific syntactic behavior whereas the AVN class may include many syntactic frames. This constrains us to generate multiple different HPSG entries for the same AVN class and even for the same AVN verb. Thereby, we produce $(x \times y)$ HPSG entries starting from an AVN class having x members and y syntactic frames. A mapping rule consists of two parts *IN* and *OUT*. The *IN* specifies a pattern in the AVN class XML. The *OUT* part specifies the corresponding generated TDL type. It can also be a modification in an already generated TDL description. In the conversion phase from AVN to HPSG, the rules are executed. and each time the IN part of a certain rule is identified in an AVN class, a TDL description driven by the OUT part of the rule is generated.

5.1 Class Mapping

The class is the main component within AVN. It encompasses syntax and semantics shared among a set of verbs (the members of the class). The syntax portion of a frame holds argument structure and thematic roles of those arguments which is considered as a semantic information in HPSG. On the other hand, a lexical entry within the HPSG theory is an individual entity that may share properties with other entries through the type inheritance mechanism. The Feature “SYNSEM” of type *synsem* is describing the syntax and semantics of

the lexical entry. Thus, the type *synsem* is the equivalent concept to an AVN class and we set the main mapping rule defined by Rule 1. By the same way, an AVN subclass (used for verbs classes such najaHa-1) is processed in a similar manner as an AVN class. The second mapping rule is defined by Rule 2.

Rule 1 IN <AVNCLASS ID="avn_id">
 <FRAME><DESCRIPTION primary="descr">...
 ⇒ **OUT** avn_id_descr := synsem

The frame having descr as a value of the description property in an AVN class identified by avn_id is mapped to a subtype of synsem named avn_id_descr:

Rule 2 IN <AVNSUBCLASS ID="avn_sub_id">
 <FRAME><DESCRIPTION primary="descr">...
 ⇒ **OUT** avn_sub_id_descr := synsem

Table 3. AVN HPSG mappings

AVN Concept	HPSG Concept
Syntactic Frame	synsem Subtype
Verb Member	several HPSG entries
THEME Roles	Thematic roles (CONTENT Feature)
FRAME SYNTAX	Argument structure (ARG-S Feature)
FRAME SEMANTICS	Not used

5.2 Syntax Mapping

An AVN frame syntax as depicted in Fig. 1 describes one of the many possible surface realizations of the argument structure for transitive, intransitive and prepositional phrases among other existing constructions. The feature describing the surface realization of the argument structure within the HPSG theory is the ARG-S. Thus, we map the list of nodes within the SYNTAX, except the <VERB/> argument which indicate the verb position, to the ARG-S Feature. The order on the ARG-S list follow exactly the same order of the AVN SYNTAX which also reflect the thematic role order in the AVN THEMEROLES.

The argument list in ARG-S is shared with VALENCE features (SUBJ and COMPS) as shown in the HPSG entry of Fig. 3. We have a degree of freedom in how to share this information because VALENCE does not necessarily show the surface realization. On the other hand, we assume that the first argument, except the verb, represents the subject of the verb and thus is mapped to the SUBJ feature. Indexing is used to relate between the ARG-S argument list and the VALENCE arguments as shown in rule 3.

Rule 3 IN $\langle SYNTAX \rangle \langle ARG1 \rangle \langle /ARG1 \rangle \dots \langle ARGn \rangle \langle /ARGn \rangle$
 $\langle /SYNTAX \rangle \Rightarrow$
OUT $avn_id := synsem \ \& \ [SYNSEM \ | \ LOC \ | \ CAT \ | \ ARG-S \langle \#ind1, \dots, \#indn \rangle]$
 $\ \& \ [SYNSEM \ | \ LOC \ | \ CAT \ | \ VAL \ | \ SUBJ \langle \#ind1 \ ARG1 \ val-arg1 \rangle]$
 $\ \& \ [SYNSEM \ | \ LOC \ | \ CAT \ | \ VAL \ | \ COMPS \langle \#ind2 \ ARG2 \ val-arg2, \dots, \#indn$
 $\ ARGn \ val-argn \rangle]$

Arguments of an AVN class or subclass avn_id are mapped to a subtype of $synsem$ named avn_id whose $SUBJ$ feature takes the first argument and $COMPS$ feature takes the remaining arguments. All arguments are coindexed with $ARG-S$ list.

Syntactic arguments included in the syntactic frame but not playing a thematic role (such as the argument *fiy* in Fig. 1) are not supposed to be considered as complements in the valence HPSG feature (Particularly $COMPS$ and $ARG-S$ feature). Nevertheless, those arguments carry a valuable information about the surface realization of the actual complements. In the mapping process, we choose to map them as specifiers of the arguments following them. Moreover, those arguments are not included on the $CONTENT$ feature. in [1], they are treated as markers. A marker is a functional but not substantial word with *mark* as part of speech. They bear an additional feature called $MARKING$ of sort *marking*. Constituents with marker daughter inherits the $MARKING$ value from that daughter. In our case, as we don't try to specify the constituent itself but a verb, we don't consider the argument preceded by a marker as a *word* but as a phrasal *sign* having a marker daughter. The mapping of a non nominal arguments (such as prepositions) is given in rule 4.

Rule 4 IN $\langle SYNTAX \rangle \dots \langle PREP \ value=valprep \rangle \langle ARGX \rangle \langle /ARGX \rangle \dots$
 $\langle /SYNTAX \rangle \Rightarrow$
OUT $avn_id := synsem \ \& \ [SYNSEM|LOC|CAT \ | \ ARG-S \langle \#ind1, \dots, \#indX \rangle]$
 $\ \& \ [SYNSEM \ | \ LOC \ | \ CAT \ | \ VAL \ | \ COMPS \langle \dots, \#indX \ ARGX \ [MARKING$
 $\ valprep], \dots \rangle]$
The value of prepositional arguments is mapped as a $MARKING$ feature of the following nominal argument.

5.3 Semantics Mapping

The $CONTENT$ feature express the semantics of a lexical entry. For a verb, the $CONTENT$ specifies a semantic relation and the semantic or thematic roles for this relation. In the Fig. 3, the semantic roles $GIVEN$, $GIVER$ and $GIFT$ are fine grained and can be organized in a hierarchy of roles. AVN $THEMEROLEs$ does not offer such a fine-grained thematic roles but offer a much generalized thematic roles that can be used within the HPSG lexicon. $THEMEROLEs$ are mapped to the $CONTENT$ part of the *sign*. Each $THEMEROLE$ within an AVN class avn_id is mapped to a semantic role having the same name in the $CONTENT$ feature of the HPSG lexical entry as shown in rule 5.

Rule 5 IN `<THEMEROLE> type="th_role" </THEMEROLE> ⇒`
OUT `avn_id := synsem & [SYNSEM | LOC | CONT | NUCLEUS[th_role #reference]]`

Each theme role belonging to theme roles of an AVN class or subclass `avn_id` is mapped to a subtype of `synsem` named `avn_id` whose NUCLEUS|th_role feature share its value via a reference with the CONT|INDEX feature of the corresponding surface argument in the COMPS list. The semantic relation feature CONT|RELN takes the name of the concerned verb.

5.4 Restrictions Mapping

Syntactic restrictions are syntactic constraints expressed specially for some argument. They are descriptions of the surface argument filling in the role. As shown in Fig. 5, the restriction is about marking a nominal argument by a certain marker (Oano_comp in this example). We simply map the type of the marker as a MARKING feature of the argument. This mapping is given by rule 6.

Rule 6 IN `<SYNTAX>... <ARGX><SYNRESTR Value="+" type=restrvalue /></ARGX>... </SYNTAX> ⇒`

OUT `avn_id := synsem & [SYNSEM | LOC | CAT | VAL | COMPS<,...,#indX ARGX [MARKING restrvalue],...>]`

The value of nominal arguments having a syntactic restriction of type x is mapped as the value of the MARKING feature of the same argument.

```
<NP value="Theme">
<SYNRESTRS>
<SYNRESTR Value="+" type="Oano_comp"/>
</SYNRESTRS>
</NP>
```

Fig. 5. Syntactic restriction on marker

Syntactic restrictions of type “plural” or “dual” (See Fig. 6) are mapped to the PERSON feature included in the INDEX feature of the argument specified in the verb’s COMPS valency list. Within a frame, syntactic restrictions may inform about the particular surface realization such as “sentential”, “genetive” and “deverbal” which are fine grained sorts of arguments. We simply ignore those types of restrictions for the argument at this level.

6 Implementation and Discussion

We used the mapping rules described in the previous section in order to develop a prototype program called AVN2HPSG for the conversion process. AVN2HPSG is

```

<NP value="Theme">
  <SYNRESTRS logic="or">
    <SYNRESTR Value="+" type="plural"/>
    <SYNRESTR Value="+" type="dual"/>
  </SYNRESTRS>
</NP>

```

Fig. 6. Syntactic restriction on number

written in C# to benefit from the *LINQ* (Language INtegrated Query)chnology through the use of Linq2XML, a powerful query framework for XML documents that simplify the task of information extraction from AVN based on specified criteria. The mapping rules are applied at specific conditions while the program navigates the XML of each AVN file.

Each XML File in AVN is a definition of an AVN class. AVN2HPSG parse the AVN class and creates two files: a TDL types file consisting of types equivalent to AVN class frames and a second TDL file consisting on verb instance definition and equivalent to AVN verb mermbers. Starting from 783 syntactic frames and 5835 verb members of the Arabic VerbNet 1.1, the program execution produced TDL files containing 555 TDL types and 14323 TDL verb specifications. There was a loss of information due to frames not conforming to the rules input specifications or simply because we ignored some cases

In order to keep the resulted resource as coherent as possible, we made some conditions to eliminate some information that may generate confusing TDL descriptions. For example, AVN contains frames in which verbs are not the first argument. Whereas, they are representing practical SCF, those frames describing nominal sentences in Arabic languages shouldn't be mapped like they are presented but firstly converted to equivalent verbal SCF.

We note a certain redundancy in the produced TDL specifications because of the absence of a type hierarchy. In fact AVN is not designed to capture syntactic generalization but semantic ones. So classes represent distinctive semantics but tend to always repeat the same argument structure descriptions. We plan to cope with this redundancy by enhancing the proposed method to automatically create a special type hierarchy to eliminate this issue.

The proposed methodology and the developed prototype program was applied on AVN but there is no assumptions that was made prohibiting the evaluation for the English VN. We plan to map the English VN to HPSG so wa can have more scope on evaluating the output lexicon compared to well elaborated HPSG lexica for English.

Though the generated lexicon is in TDL and can be directly used in LKB system, we plan to adopt the TDL type hierarchy employed in matrix grammars which can yield to more organized lexicon and easy operation by grammar writers.

7 Conclusion

We detailed the process of building a HPSG lexicon from Arabic VerbNet. We have successfully extracted a verbal lexical resource for the HPSG formalism expressed in TDL to offer an easy way to use it in within the LKB system. The entire process is driven by specific mapping rules that can be enhanced to extract more information into the target lexicon. We think this resource is of great value for natural language processing and for evaluating purposes against other automatically and manually constructed resources for Arabic. We plan to include more organization in the generated lexicon by using an adapted TDL type hierarchy. Moreover, since AVN integrate deverbal and participle lemma for each verb, we plan to extract those lexemes and associate them syntactically and semantically with their corresponding verbs to build a large coverage lexicon.

References

1. Pollard, C., Sag, I.A.: Head-Driven Phrase Structure Grammar. University of Chicago Press, Chicago (1994)
2. Tesnière, L.: *Eléments de syntaxe structurale*. Klincksieck, Paris (1959)
3. Briscoe, T., Carroll, J.A.: Automatic extraction of subcategorization from corpora. In: ANLP. (1997) 356–363
4. Schuler, K.K.: VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. PhD thesis, University of Pennsylvania (2005)
5. Jaouad, M.: Classifying arabic verbs using sibling classes. In: In the Proceedings of the International Workshop on Computational Semantics, Oxford, UK (2011)
6. Levin, B.: *English Verb Classes and Alternations: a preliminary investigation*. University of Chicago Press, Chicago and London (1993)
7. Palmer, M., Kingsbury, P., Gildea, D.: The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics* **31** (2005)
8. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Wordnet: An on-line lexical database. *International Journal of Lexicography* **3** (1990) 235–244
9. Doran, C., Egedi, D., Hockey, B.A., Srinivas, B., Zaidel, M.: Xtag system – a wide coverage grammar for english (1994)
10. Ryant, N., Kipper, K.: Assigning xtag trees to verbnet. In: Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms, Vancouver, Canada (2004)
11. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley framenet project. In: In proceedings of the COLING-ACL. (1998) 86–90
12. Pazienza, M.T., Pennacchiotti, M., Zanzotto, F.M.: Mixing wordnet, verbnet and propbank for studying verb relations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC-2006). (2006)
13. Palmer, M., Bonial, C., Mccarthy, D.: Semlink+: Framenet, verbnet and event ontologies. In: ACL2014, Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014). (2014)
14. Loper, E., ting Yi, S., Palmer, M.: Combining lexical resources: Mapping between propbank and verbnet. In: In Proceedings of the 7th International Workshop on Computational Linguistics. (2007)

15. Attia, M.: Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation. PhD thesis, The University of Manchester, Manchester, UK (2008)
16. Boukedi, S., Haddar, K.: HPSG grammar for arabic coordination experimented with LKB system. In: Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference, FLAIRS 2014, Pensacola Beach, Florida, May 21-23, 2014. (2014)
17. Maamouri, M., Bies, A.: Developing an arabic treebank: Methods, guidelines, procedures, and tools. In: Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages. Semitic '04 (2004) 2–9
18. Attia, M., Shaalan, K.F., Tounsi, L., van Genabith, J.: Automatic extraction and evaluation of arabic LFG resources. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23-25, 2012. (2012) 1947–1954
19. Tounsi, L., Attia, M., Genabith, J.V.: Automatic treebank-based acquisition of arabic lfg dependency structures. In: Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages. Semitic '09, Stroudsburg, PA, USA, Association for Computational Linguistics (2009) 45–52
20. Habash, N., Rambow, O.: Extracting a tree adjoining grammar from the penn arabic treebank. In: In Traitement Automatique du Langage Naturel (TALN-04. (2004)
21. Flickinger, D.: Lexical Rules in the Hierarchical Lexicon. PhD thesis, Linguistics Department, Stanford University, Stanford, California (1987)
22. Blache, P.: Constraints, linguistic theories and natural language processing. In: Natural Language Processing - NLP 2000, Second International Conference, Patras, Greece, June 2-4, 2000, Proceedings. (2000) 221–232
23. Saussure, F.D.: Cours de linguistique générale. v.C. Bally and A. Sechehaye (eds.), Paris/Lausanne (1916) English translation: Course in General Linguistics. London: Peter Owen, 1960.
24. Barwise, J., Perry, J.: Situations and Attitudes. The MIT Press: Cambridge, MA, USA (1983)
25. Krieger, H.U.: TDL - A Type Description Language for HPSG. Part 2: User Guide. DFKI, Saarbruecken. (1994) Research Report D-94-14.
26. Copestake, A.: Implementing Typed Feature Structure Grammars. CSLI Publications, Stanford, CA (2002)
27. Bender, E.M., Drellishak, S., Fokkens, A., Goodman, M.W., Mills, D.P., Poulson, L., Saleem, S.: Grammar prototyping and testing with the lingo grammar matrix customization system. In: ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, System Demonstrations. (2010) 1–6