Argument Optionality:
A New Library for the Grammar Matrix Customization System

Safiyyah Saleem

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Arts

University of Washington

2010

Program Authorized to Offer Degree: Department of Linguistics

University of Washington
Graduate School

This is to certify that I have examined this copy of a master's thesis by

Safiyyah Saleem

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____
Emily M. Bender

_____
Fei Xia

Date: _____

In presenting this thesis in partial fulfillment of the requirements for a master's degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Any other reproduction for any purpose or by any means shall not be allowed without my written permission.

Signature_____

Date_____

University of Washington

Abstract

Argument Optionality: A New Library for the Grammar Matrix Customization System

Safiyyah Saleem

Chair of the Supervisory Committee:
Assistant Professor Emily M. Bender
Department of Linguistics

The Grammar Matrix Customization System allows a user-linguist to quickly build a customized starter grammar by describing a language's syntax through a web-based questionnaire. The customization system is a work in progress and new libraries are developed to increase the breadth and depth of linguistic phenomena that the customized starter grammars are able to represent. This thesis describes the motivation behind and subsequent creation of a library that enables these grammars to parse sentences which do not contain overt subjects and/or objects (often referred to as pro-drop, null arguments, object/subject dropping).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
INTRODUCTION

The Grammar Matrix customization system[1] (Bender, Flickinger, and Oepen, 2002; Bender and Flickinger, 2005; Drellishak, 2009) allows users to develop basic HPSG grammars based upon answers to a web-based questionnaire. It is comprised of a core grammar and a collection of libraries which correspond to different syntactic phenomena found in many languages. Some of the phenomena modeled in the Grammar Matrix are universal in that each language must make a choice from a finite list of options, e.g., word order. Others represent constraints that are posited to apply to all languages, e.g., semantic compositionality. Beyond these universal linguistic phenomena, the Grammar Matrix also includes some phenomena that can be in many, but not all, of the world's languages, e.g., case (Drellishak, 2009). Argument optionality (also referred to in the literature as pro-drop, subject dropping, and object dropping) is one phenomenon that is found in many languages but was not represented in the system. This thesis examines the typological literature to determine how argument optionality is realized in different languages and describes how this information was used to inform the creation of an argument optionality library for the Grammar Matrix customization system. It begins by giving a brief overview of HPSG, the grammatical framework that is used in the customization system in Chapter 2. Chapter 2 also provides a short description of the customization system's

[1]For ease of exposition, I will refer to the Grammar Matrix customization system as the Grammar Matrix, the customization system, or simply the system in this work.

architecture. This is followed by a review of the typological literature in Chapter 3, the presentation of my analyses in Chapter 4, and details on how these analyses were implemented in Chapter 5. An explanation of how the newly implemented argument optionality library was evaluated and the results of the evaluation are given in Chapter 6 and the thesis closes with a short conclusion in Chapter 7.

Chapter 2

BACKGROUND

The Grammar Matrix Customization System is developed within the framework of Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994). Since a general understanding of HPSG is necessary in order to understand how the Grammar Matrix works, the first section in this chapter gives a brief explanation of HPSG and how it formulates the licensing of grammatical strings. The second section describes the components of the Grammar Matrix customization system and how they interact with each other to produce a starter-grammar.

2.1 HPSG

The Grammar Matrix is developed within the theoretical framework of Head-driven Phrase Structure Grammar (HPSG). HPSG is highly lexicalist and models natural language using typed feature structures which consist of features and associated values. Lexical items, words, and rules (both lexical and phrase-structure rules) are all represented by these typed feature structures which are conventionally written as attribute value matrices (avm) (Pollard and Sag, 1994). Example (1), adapted from Pollard and Sag (1994), shows the lexical entry for the pronoun *she*.

(1)
$$\begin{bmatrix} \textit{word} \\ \text{STEM} \left\langle \text{she} \right\rangle \\ \text{SYNSEM} \begin{bmatrix} \text{LOCAL} \begin{bmatrix} \text{CATEGORY} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{noun} \\ \text{CASE} \quad \textit{nom} \end{bmatrix} \\ \text{VAL} \begin{bmatrix} \text{COMPS} \quad \langle\rangle \\ \text{SUBJ} \quad \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONTENT} \begin{bmatrix} \text{HOOK} \begin{bmatrix} \text{INDEX} \begin{bmatrix} \text{PER} \quad \textit{3rd} \\ \text{NUM} \quad \textit{sing} \\ \text{GEND} \quad \textit{fem} \end{bmatrix} \\ \text{LTOP} \quad \textit{handle} \\ \text{XARG} \quad \textit{individual} \end{bmatrix} \\ \text{RELS} \quad \textit{diff-list} \\ \text{HCONS} \quad \textit{diff-list} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

A lexical item can undergo lexical rules which can add additional constraints to the feature structure. Rules (both lexical and phrasal) are defined as having the feature ARGS. If a feature structure meets the constraints of a lexical rule's ARGS value, it is allowed to undergo that rule. It is important to note that the rules themselves are typed feature structures and the output of the rule is a typed feature structure as well. Types can inherit from other types. In fact, one type can inherit from multiple types as long as the constraints specified on those types do not conflict with each other.

The Grammar Matrix uses a set of feature names and a geometry associated with those names that are somewhat different from those proposed in Pollard and Sag (1994). For example Pollard and Sag use PHON to represent the form of a word

whereas in the Grammar Matrix, this information is contained in the feature STEM. Henceforth the feature paths and types given as examples are based upon those used in the Grammar Matrix customization system. Let us consider the simplified version of two types that are defined in the system: `transitive-lex-item` and `verb-lex`.[1]

(2)
$$\begin{bmatrix} \textit{verb-lex} \\ \text{SYNSEM} \begin{bmatrix} \text{LOCAL} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \boxed{1} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \text{HOOK} \begin{bmatrix} \text{XARG} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST.FIRST} \ \boxed{1} \begin{bmatrix} \text{LOCAL} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \text{HOOK} \begin{bmatrix} \text{INDEX} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{INFLECTED} \ - \end{bmatrix}$$

(3)
$$\begin{bmatrix} \textit{transitive-lex-item} \\ \text{ARG-ST} \left\langle \begin{bmatrix} \text{LOCAL} \mid \text{CONT} \mid \text{HOOK} & \begin{bmatrix} \text{INDEX} & \boxed{1}\textit{ref-ind} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \text{LOCAL} \mid \text{CONT} \mid \text{HOOK} & \begin{bmatrix} \text{INDEX} & \boxed{2}\textit{ref-ind} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SYNSEM} \mid \text{LKEYS} \mid \text{KEYREL} \begin{bmatrix} \text{ARG1} & \boxed{1} \\ \text{ARG2} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

[1] `transitive-lex-item` is defined in the core Matrix while `verb-lex` is defined in a library. As such the constraints shown on `transitive-lex-item` will be present in any grammar created by the system while the constraints on `verb-lex` may be different depending on the user's choices. See §2.2 for more details on the distinction between the core Matrix and the Matrix libraries.

ARG-ST is a list valued feature which contains a lexical item's core arguments. The type `verb-lex` inherits a constraint from `basic-verb-lex` that the value of HEAD be type `verb` and it places a new constraint that identifies the first item on its ARG-ST list with the sole item on its SUBJ list. It is also [INFLECTED –]. The INFLECTED feature governs whether or not a feature structure is allowed to serve as input to a phrase structure rule.[2] Feature structures which are compatible with [INFLECTED +] are allowed to do so while those which are [INFLECTED –] are not. These feature structures must undergo a lexical rule whose output is [INFLECTED +].[3] The `transitive-lex-item` constrains the item to having exactly two core arguments by restricting the ARG-ST list to contain exactly two elements. It also establishes the item's semantic relationship with the two element's on its ARG-ST list. Since these two types do not have conflicting constraints, using multiple inheritance we can create a new type that combines both of these constraints. `transitive-verb-lex` is a type that constrains the HEAD to `verb` and the COMPS list to have exactly one element, which is identified with the second element on the ARG-ST list.

[2]In other HPSG analyses (e.g. Sag, Wasow, and Bender, 2003), this distinction between feature structures which are able to play in the syntax and those that are not is handled by a type instead of a feature. Specifically, feature structures which are not able to play in the syntax are type `lexeme` and those that are are type `word`.

[3]The distribution of values of this feature is language specific. In English verbs must be inflected before being allowed to serve as daughters to a phrase structure rule, and thus `verb-lex` is constrained to be INFLECTED –. Some languages, however, do not require verbs to undergo any lexical rules and in these languages this value would be underspecified as type `bool`. The other constraints on `verb-lex` are posited to be true for all languages.

(4)
$$\begin{bmatrix} \textit{transitive-verb-lex} \\ \text{STEM}\left\langle \text{BUY} \right\rangle \\ \text{SYNSEM}\left[\text{LOCAL}\,|\,\text{CAT}\,|\,\text{VAL}\,|\,\text{COMPS}\left\langle \boxed{1} \right\rangle\right] \\ \text{ARG-ST}\left\langle \left[\text{LOCAL}\,|\,\text{CAT}\left[\text{HEAD}\quad \textit{noun}\right]\right], \boxed{1}\left[\text{LOCAL}\,|\,\text{CAT}\begin{bmatrix}\text{HEAD} & \textit{noun} \\ \text{VAL} & \left\langle \begin{matrix}\text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle\end{matrix}\right\rangle\end{bmatrix}\right]\right\rangle \\ \text{INFLECTED}\ - \end{bmatrix}$$

As Example (4) shows, in English, it is also constrained to be [INFLECTED –]. In order to meet the conditions to serve as the HEAD-DTR for a phrase structure rule, the verb must first undergo a lexical rule which will change it from [INFLECTED –] to [INFLECTED +]. The `3sing-lex-rule` is one of the lexical rules which does this. Like all lexical rules, it inherits a constraint from `lex-rule` that specifies that its DTR be a feature structure that is type `word-or-lexrule`. The `3sing-lex-rule` further constrains its DTR to be HEAD `verb` and constrains the mother to be [INFLECTED +].

(5)
$$\begin{bmatrix} \textit{3sing-lex-rule} \\ \text{STEM}\left\langle \text{BUY} \right\rangle \\ \text{SYNSEM}\left[\text{LOCAL}\left[\text{CAT}\quad \boxed{1}\left[\text{VAL}\left[\text{SUBJ}\left\langle \left[\text{LOCAL}\left[\text{CONT}\left[\text{INDEX}\begin{bmatrix}\text{PER} & \textit{3rd} \\ \text{NUM} & \textit{sing}\end{bmatrix}\right]\right]\right]\right\rangle\right]\right]\right]\right] \\ \text{DTR}\left[\text{SYNSEM}\left[\text{LOCAL}\left[\text{CAT}\quad \boxed{1}\left[\text{HEAD}\ \textit{verb}\right]\right]\right]\right] \\ \text{INFLECTED}\ + \end{bmatrix}$$

Once the verb has undergone the `3sing-lex-rule`, it is [INFLECTED +] and, thus, able to combine with other words to form phrases via phrase structure rules. Phrases in turn combine with other phrases to form sentences. The grammar also defines a set of feature constraints that a feature structure must meet in order to be classified as a grammatical sentence (the initial symbol).

Following work in Categorial Grammar (Ajdukiewicz, 1935; Bar-Hillel, 1953), in HPSG a head's valence lists contain information about its dependents or required arguments. As heads combine with their arguments via phrase structure rules, the valence lists are shortened. A grammatical sentence is defined as a verbal projection with empty valence lists (i.e. empty SUBJ and COMPS lists). In HPSG, of course, this projection is modeled as feature structure. Thus, a grammatical sentence is a feature structure which is HEAD `verb` and has empty COMPS and SUBJ lists. Let us consider the following strings.

(6) She

(7) She buys

(8) She buys books

Example (6) is not grammatical because as Example (1) shows, *she* is HEAD `noun`. Example (7) is not grammatical either because *buy* is a transitive verb which has a non-empty COMPS list. The `head-subj-phrase` rule specifies that its head daughter must have an empty COMPS list. Thus, Example (7) would never even form a constituent.[4] Example (8) is grammatical because *buys books* can combine via the `head-comp-phrase` rule. The resulting feature structure would be HEAD `verb` and

[4]In a larger grammar that included the SLASH feature and head filler rules, Example (7) would form a constituent, but it would still not be considered a grammatical as a standalone utterance.

COMPS <> (<> denotes an empty list). A feature structure with these values meets the constraints to undergo the `subj-head-phrase` rule which constrains its HEAD-DTR to be HEAD `verb` and COMPS<>. The resulting feature structure would be HEAD `verb` and have empty SUBJ and COMPS lists thus meets the definition of a sentence.

2.2 The Grammar Matrix Customization System

As described in Chapter 1, the Grammar Matrix Customization System creates grammars based on a user-linguist's answers to a web-based questionnaire. The grammars conform to the formalism of Head-driven Phrase Structure Grammar (`HPSG`) which is explained in §2.1. The system can be conceived of as consisting of three major components: The core Matrix, Matrix libraries, and the customization system.

2.2.1 The Core Matrix

Drellishak (2009) describes the core Matrix as consisting of types which are either found in all languages or are closely related groups of types at least one of which is hypothesized to be found in all languages. All languages are posited to contain nouns and verbs and so `basic-noun-lex` and `basic-verb-lex` are types defined in the core Matrix. Drellishak uses types associated with word order as an example of a group of types in which all languages are hypothesized to use at least one of the members. Languages can be distinguished by the order that they require head daughters to take in relation to non-head daughters in the various phrase types. English is an SVO language and so requires that the complement of a verb come after the head verb while the subject must come before the head verb. Thus, English requires a head-comp-phrase rule which is head-initial and a subj-head-phrase rule which is head-final. In an SOV language, both subject and object come before the head verb and thus the

language may not make use of the `head-initial` type;[5] however, since all languages can be expected to use at least one of these types, both are included in the core Matrix.

2.2.2 Libraries

Libraries are described in Drellishak (2009) as consisting of types and associated linguistic analyses that are relevant for many but not all of the worlds languages. Features such as case and gender distinctions in nouns are found in a number of different languages but are absent in others. Therefore, types related to gender and case are housed in the Matrix libraries for these phenomena instead of in the core Matrix. This distinction is important for theoretical and practical reasons. In terms of linguistic theory and hypothesis testing, it is interesting to keep track of which language features are truly universal and which are very widespread but not necessarily universal. From a more practical view, each grammar that is generated using the customization system includes a copy of all of the types in the core Matrix, even those which are not directly used in the resulting grammar. Types defined in the Matrix libraries, however, are only included in those languages for which the type is relevant. Thus a language which does not have gender distinctions would not have the feature GENDER as a part of its feature geometry while a language such as English which does will.

---

[5] If the language has independent determiners and these occur in after the head noun, the grammar generated by the customization system will use `head-initial` for building noun phrases. Also, the user may extend the grammar to cover phrase types which are not supported by the current system, e.g., adpositional phrases which may have a different head-complement ordering

2.2.3 Customization System

The user's experience with the customization system begins by filling out the web-based questionnaire. The answers are stored and must pass validation before the user is allowed to call the script which will create the grammar. Each of these steps is explained in further detail in the subsections which follow.

Questionnaire

The questionnaire is comprised of several pages each of which roughly corresponds to a library. Some of the pages are mandatory so that a user cannot create a grammar unless specific questions on the page have been answered (Word Order, Person, Case, and Lexicon) while others are entirely optional and only have to be entered if the user so desires (Gender, Number, Tense and Aspect, Direct Inverse, Sentential Negation, Matrix Yes/No Questions, and Coordination).

Certain aspects of the questionnaire are tightly constrained. For the most part, the user is only able to enter information about syntactic phenomena that are covered in the libraries. Since, presently, there is no library which addresses adjectives, adverbs, and other adjuncts, the user is unable to enter information about these types of words.

In addition to constraints on the range of phenomena for which information is elicited, there are also constraints on the way in which answers can be given. Some of the questions force the user to choose among a pre-defined, typologically motivated, set of analyses while others allow more free-form responses. The Word Order page is an example of the former, the Gender page is an example of the latter, and the Tense and Aspect page includes both approaches. On the Word Order page, the user is directed to choose among a predefined list of possible values for the position that subjects, objects and verbs take in matrix clauses. In contrast, the Gender page asks the user to input types that will form the hierarchy for the GENDER feature. These

are entirely user-generated. The Tense and Aspect page on the other hand offers the user the option of choosing from a predefined list of tenses or creating an entirely new TENSE hierarchy.

The user is also given freedom to augment the feature geometry used in the starter grammar through the Other Features page. This page allows the user to create new syntactic and semantic features. In general, the user is able to create an arbitrary and in theory infinite number of features and feature values as well as verb and noun types, stems, and semantic predication values. Arbitrarily many morphological rules can also be created.

Since the questionnaire is dynamic, the choices that a user makes on each page affect what is seen after that. The number of pages and the questions asked on the questionnaire are static but answers in one part affect the options that are available in other parts. For example, if a user answers questions about tense and aspect on the Tense and Aspect page, then these choices will appear as possible features to choose from on the Lexicon page where lexical types and stems are defined and morphological rules are created. This dynamism plays an important role in the implementation of argument optionality; see §5 for more details on how the argument optionality implementation takes advantage of this aspect of the questionnaire.

Choices File

As the user-linguist answers questions on the questionnaire, his or her answers are stored in a file aptly named 'choices'. Users are able to download their choices file and they are also able to upload a choices file into the questionnaire. Since the choices files can be saved and then uploaded to the questionnaire, users are not forced to complete the questionnaire in a single session. The choices file is also how the customization system receives information from the questionnaire. Once a user clicks on the 'Create

Grammar' button, the customization script is called and the choices file serves as input for this script.

Validation and Customize

The user is not allowed to click on 'Create Grammar' until the choices file has passed validation. Each time that the user submits a page of the questionnaire, the validation script is called to determine whether the current state of the choices will lead to viable grammar. If this is not the case, on the main page red asterisks will appear next to the name of the subpage(s) which contain(s) the problem(s). The issue may simply be that the user has not completed a required subpage or the user may have given conflicting information. Once the choices file is in a state in which a grammar can be created, the 'Create Grammar' button is ungreyed. Clicking on this button calls the customization script, which takes the choices file and creates a folder which contains the files that are common to all grammars (matrix.tdl and various files which support use in the LKB and PET parsers) as well the language-specific files whose contents are determined by writing the types and constraints that each library says is necessary based on the answers contained in the choices file. Once the script has run, the user is directed to a webpage where he or she is able to download the newly created grammar.

Since the validation code forces the user to create at least one noun type, one intransitive verb type and one transitive verb type, the starter-grammar can be loaded into the LKB or PET to immediately start parsing simple strings.

Chapter 3
TYPOLOGICAL LITERATURE REVIEW

Argument realization varies across the world's languages. Some languages always require free overt subjects and objects while others allow certain arguments to be dropped. Languages which allow arguments to be dropped may or may not use markings on the verb to denote person, number, and gender. Some languages require marked word order when arguments are dropped. Others only allow argument dropping in certain person/number/gender or tense/aspect/mood environments. Each of these differences will be explored in §3.1. In addition to variation in syntactic constraints, there are also differences in the semantic interpretation of dropped arguments. Dropped arguments can either be referential or non-referential, definite or indefinite. §3.2 describes ways in which the semantics associated with dropped arguments have been analyzed.

3.1 Syntactic Variation

3.1.1 Subject Dropping

A typological survey of pronominal subjects in WALS (Haspelmath et al., 2008) conducted by Dryer (2008) categorizes languages according to how pronominal subjects are realized. Dryer classifies 674 languages as falling into one of six categories.

1. Pronominal subjects are expressed by pronouns in subject position that are normally if not obligatorily present

2. Pronominal subjects are expressed by affixes on verbs

3. Pronominal subjects are expressed by clitics with variable hosts

4. Pronominal subjects are expressed by subject pronouns that occur in a different syntactic position from full noun phrase subjects

5. Pronominal subjects are expressed only by pronouns in subject position, but these pronouns are often left out

6. Pronominal subjects are expressed by more than one of the above types with none dominant

Dryer found that having pronominal affixes expressed on the verb was the dominant strategy by far (409 out of 674). Requiring that subjects be expressed by separate pronouns in the subject position (as in English) was a distant second at 77 languages. Interestingly, the prevalence of subject dropping suggests that referring to this phenomenon as 'dropping' is actually a misnomer if one takes the most frequently occurring strategy as the default. The customization system was already able to handle category 1. This project sought to add functionality for categories 2 and 5. Categories 3 and 4 are not represented in the Grammar Matrix and fall outside of the scope of this project as implementing them would require a number of improvements to the word order library.

In order to implement argument optionality in the customization system, more information was needed about how argument optionality interacts with other properties. Further research into the literature uncovered that there was variation in the interaction between person, tense/aspect/mood and subject dropping (Ackema et al., 2006).

In Finnish for example, subject dropping is allowed in all tense/aspect/moods, but only for certain persons. As shown in the following examples adapted from (Vainikka

and Levy, 1999, pages 614, 657), subject dropping of the third person singular is typically disallowed. It is licensed for expletives, certain embedded clauses and generic subjects.

The licensing of third person subject dropping for generic subjects imposes new word order constraints. Finnish is canonically SVO and when a first or second person subject is dropped, it is permissible for the verb to take sentence initial position as it does in Example (9b); however, as Example (10) shows, V-initial sentences are prohibited when third person subjects are dropped.

(9) a. *Nousi junaan
step-PAST/3SG train-into

(He/She) boarded the train. [fin]

b. Nousin junaan
step-PAST-1SG train-into

I boarded the train. [fin]

(10) a. *Voi anoa lainaa pankista.
can-3SG apply-INF loan-PAR bank-ELA

One can apply for a loan at the bank. [fin]

b. Pankista voi anoa lainaa.
bank-ELA voi can-3SG apply-INF loan-PAR

One can apply for a loan at the bank. [fin]

c. Lainaa voi anoa pankista.
loan-PAR can-3SG apply-INF bank-ELA

One can apply for a loan at the bank. [fin]

Hebrew has a similar but slightly different set of restrictions on subject dropping. In addition to restricting third person subject dropping to expletives, certain types of non-matrix clauses, and generic reference, it only allows first and second person subjects to be dropped in the past and future tense. Expletives and generic impersonal references allow subject dropping in the present tense as shown in Example (11), which is taken from Vainikka and Levy (1999, pages 655, 658).

(11) a. Nire she-Nurit tenaceax.
seem-PRES/SGM that-Nurit win-FUT/3SG

It seems that Nurit will win. [heb]

b. Oxlim harbe ba-xoref
eat-PRES/PL much in-winter

One eats a lot in the winter. [heb]

In addition to syntactic contexts, there was the suggestion that in at least one language, Tamil [tam] (Dravidian), licensing of subject dropping is lexically-based. In general Tamil allows subjects and objects to be freely dropped, but there is a class of weather-related verbs for which overt subjects are required (Asher, 1985). Example (12a) is adapted from Asher (1985, page 53) and Example (12b) is assumed given his statement that subjects can normally be deleted except for in sentences like Example (12a).

(12) a. maZe pey-r-atu
rain.NOM fall-PRES-3SN

It's raining. [tam]

b. *pey-r-atu
fall-PRES-3SN

It's raining. [tam]

3.1.2 Object Dropping

To my knowledge, there has not been a large-scale survey done for object realization of transitive verbs. However, I was able to identify three areas in which there is variation in how object dropping is realized: the existence of verbal affixes and their co-occurrence with overt objects; verbal affix co-occurrence with types of noun phrases; and lexical licensing.

Affixes and Overt Objects

Looking at a few languages which allow object dropping, there appears to be considerable variation in how the verbal affixes interact with full noun phrase objects. In Tamil there is no marker left on the verb and all transitive verbs allow object dropping (Asher, 1985). In Arabic, all verbs allow object dropping and a marker is required for most verbs if the full noun phrase is not present. If the full noun phrase is present, then the marker is optional (see below for a discussion of verbs which can drop objects without a marker). When the full noun phrase and object marker co-occur, the full noun phrase is interpreted as an appositive. In Nkore-Kiga [nyn] (Bantoid) which is spoken primarily in Uganda, the object marker is required when an object precedes the verb or when it is pronominal (Taylor, 1985). In Hausa [hau] (Chadic), person, number, and gender of the object is not marked on the verb; however, the verb form differs according to whether the direct object is present immediately after the verb and whether it is a full noun phrase or a pronominal. Different verbs forms are used in each of these cases (Newman, 2000). Looking at these languages, a verbal affix may be required, optional, or not permitted when an overt argument is also present. It also appears that verbal affixes can be required or not permitted when an object is dropped. Assuming that affix optionality may also be possible when an object is dropped, we have the nine logical possibilities for affix marking in the presence of an

	Affix w/Dropped Obj	Affix w/Overt Obj	Language
1	required	required	Abkhaz
2	optional	optional	Larike
3	not permitted	not permitted	Tamil
4	required	optional	Arabic
5	optional	not permitted	Unattested
6	not permitted	required	Unattested
7	required	not permitted	French?, Ngalakan
8	optional	required	Unattested
9	not permitted	optional	Unattested

Table 3.1: Logical possibilities of affix and overt object co-occurrence restrictions

overt or dropped argument shown in Table 3.1.

Affixes and NP Types

As indicated by the examples of Arabic, Hausa and Nkore-Kiga in the previous section, the overt argument and affix co-occurrence restrictions are not as simple as Table 3.1 suggests. In some languages, verbal affixes marking the object are in complementary distribution with certain types of noun phrases but can co-occur with others. Corbett (2003) gives examples of other typologically diverse languages with these types of restrictions in subject as well as object affixes. Morimoto (2002) provides details about a number of Bantu languages. According to these two sources, the NP features upon which affix and overt NP co-occurrence restrictions are based can summarized as follows.

- Common vs. Proper Nouns

- Pronominal vs. Full NP
- Proper Name vs Non-Proper Name
- Animate vs. Inanimate

Lexically-based

While I found only one example of a language which only licenses subject dropping for a certain class of verbs, lexically-based object dropping seems to occur more frequently. In fact, it could be argued that English exhibits this phenomenon (Fillmore, 1986).

English is typically thought of as a language which does not license argument dropping in matrix clauses. Indeed it has such a strong preference for overt arguments that 'dummy' pronouns are used to take the syntactic subject position even when there is no real world referent. In Example (13) *it* has no referent yet the sentence is ungrammatical without it.

(13) a. It is clear that pronouns are important in English.

b. *Is clear that pronouns are important in English.

Within FrameNet, a lexicography project that collects data on English verb valence patterns, Fillmore, Johnson, and Petruck (2003) identify three types of null objects: constructional null instantiations (CNIs), definite null instantiations (DNIs), and indefinite null instantiations (INIs). CNIs refer to syntactically-based object dropping while DNIs and INIs are lexically-based. Liu (2008) draws upon data from the British National Corpus to argue that there is a class of verbs in English (e.g. promise, understand, know, tell, etc.) that allows objects to be optionally deleted.

(14) A. "Constable Perkins is here," Mrs. Clancy said.

B. "Yes, I know."

Example (14) makes it clear that *know* is in fact referring back to "Constable Perkins is here". This would be an example of definite null instantiation since the object is known to both speaker and the hearer. In contrast, Example (15) shows indefinite instantiation. Exactly what the speaker was eating is unknown.

(15) I was eating when you called.

While some verbs allow definite and/or indefinite null instantiation for certain contexts, others do not allow their objects to be dropped under any circumstances. Although, 'devour' and 'eat' are closely related semantically, Example (15) is grammatical while Example (16) is not.

(16) *I was devouring when you called.

This suggests that English could be analyzed as having lexically based object dropping. Although the idea of definite and indefinite null instantiation as defined by FrameNet was developed based on English, the theoretical principles used have also been applied to Spanish, German, and Japanese (Fillmore, Johnson, and Petruck, 2003). Thus, there is reason to think that this phenomenon is cross-linguistic. There is also some evidence that it exists in Arabic. Normally an object marker is required when there is no overt object; however with certain verbs it is possible to drop the object without the marker. Example (17) is an example of the former while Example (18) demonstrates the latter.

(17) a. ishtaraa kitaab-an
3ms.buy.past book-acc

He bought a book. [arb]

b. ishtaraa-hu
3ms.subj.buy.past-3ms.obj

He bought it. [arb]

c. *ishtaraa
3ms.buy.past

He bought. [arb]

(18) a. darasa al-dars-a
3ms.study.past def-lesson-acc

He studied the lesson. [arb]

b. darasa-hu
3ms.subj.study.past-3ms.obj

He studied it. [arb]

c. darasa
3ms.study.past

He studied. [arb]

3.2 Semantic Distinctions

As detailed in §3.1.2, null objects can be interpreted as definite or indefinite. The way that this occurs may differ from language to language. Languages also differ in the meaning assigned to dropped arguments and/or affixes which co-occur with overt arguments. Sulkala and Merja (1992) state that in Finnish third person singular subjects can only be dropped with a generic interpretation in traditional matrix clauses and objects can be omitted with a generic interpretation. In Nkore-Kiga, subject markers are always required and both subject and object free pronouns are always emphatic (Taylor, 1985); although fused subject/object affixes are required in West

Greenlandic [kal], free pronouns can occur in non-emphatic contexts. in West Greenlandic (Fortescue, 1984). In Kiyaka [yaf] (Bantoid), the presence of an object marker precludes an indefinite interpretation while objects of verbs which do not contain the object marker can be interpreted as either definite or indefinite (Morimoto, 2002).

This thesis focuses on modeling the syntactic variation in how argument optionality is realized, and thus, I do not present a detailed typological review of these semantics distinction nor do the analyses in Chapter 4 or implementation in Chapter 5 address the semantic representation(s) of argument optionality. I will note, however, that this is an important area for future work in expanding the Argument Optionality library.[1]

3.3 Summary

The typological literature shows that there is a great deal of diversity in the ways in which argument optionality is realized in the world's languages. These difference range from the syntactic to the semantic. The semantic differences were described briefly as they fall outside of the scope of this project. The syntactic differences include permissibility of any argument dropping, context dependent and lexically-based argument dropping, and verbal affix/overt NP co-occurrence restrictions. For subjects, the most common pattern by far is to allow dropping. At least one language, Hausa, even requires that pronominal subjects be dropped. English appears to be among a minority of languages which typically or always require overt subjects in matrix clauses. Context-dependent dropping appears to be comparatively rare. I was only able to find two languages which demonstrate this pattern—Hebrew and Finnish. There was no evidence of widespread lexically-based subject dropping; Tamil was

[1]Bender and Goss-Grubbs (2008) propose a way to model the semantic representation of referential and deictic distinctions as well as overt pronouns and zero anaphora which may serve as a good starting point.

the only language which demonstrated this pattern. On the other hand, there was considerable evidence that lexically-based object dropping may be prevalent in many languages, even ones like English which strictly require overt subjects. Finally, the literature review showed that affix/overt NP co-occurrence restrictions are complex and may be based on the specific properties of the NP and may also create new word order constraints. The next chapter provides HPSG-based analyses of the syntactic variation uncovered in this chapter.

Chapter 4
ANALYSIS

4.1 Overview

As described in §2.1, verbs begin with a nonempty SUBJ list and transitive verbs begin with a nonempty COMPS list. A sentence is modeled as a verbal projection in which both of these lists are empty. In the grammars constructed through the Grammar Matrix customization system, this is accomplished by having a transitive verb and its direct object undergo a subtype of the `basic-head-comp-phrase-rule` and combine with a noun phrase subject via a subtype of the `basic-subj-head-phrase-rule`. The subtypes that are used and the order in which they are applied depend upon the word order requirements of the language, but a transitive verb must undergo a subtype of both rules in order to have a resulting feature structure that has empty COMPS and SUBJ lists. Both of these rules are binary and require both a head daughter (the verb) and a non-head daughter (the object or subject depending on the rule). The question, then, is how can strings which do not contain an overt object or subject satisfy these requirements? Logically, there are four ways in which a valence list can be shortened.

1. Instead of mapping each element of the ARG-ST to a feature on the valence list, one or more of these elements could be suppressed.

2. The lexical rule which attaches an affix to the verb could also shorten the appropriate valence list

3. A unary phrase structure rule could shorten the valence list(s) of the mother.

4. A phonologically empty element could fill the relevant argument slot.

This chapter begins by reviewing how argument optionality has been treated in HPSG and other frameworks. I then present a set of HPSG analyses that use various phrase structure and lexical rules based on the feature geometry and types used in the Grammar Matrix customization system. Areas were this geometry differs in a significant way from the feature geometry common to HPSG community will be pointed out.

4.2 Related Work

As mentioned previously, argument optionality or 'pro-drop' is a phenomenon that has been studied within a number of different grammatical frameworks. The following paragraphs describe the approaches taken within Chomskyan-based transformational frameworks, Lexical-Functional Grammar (LFG) and HPSG. A simple overview of the general approach to analyzing argument dropping is given along with a few examples of work that has been done on applying this general approach to particular languages. I also point out how this relates to the four ways of shortening the valence lists given above and highlight where the analyses that I will present differ from other approaches.

Within transformational frameworks, Chomsky (1982) takes the fourth approach listed in §4.1. He posits that a phonologically empty pronoun 'pro' fills the subject position. In Chomsky's initial proposal, whether or not a language licenses 'pro-drop' is determined by the verbal inflectional system. Working from this basic analysis, Rizzi (1986) provides an account of the licensing constraints on pro-drop that was motivated by contrasting Italian licensing of object dropping with English. Ackema et al. (2006) edited a book devoted to presenting analyses that sought to model corner cases such as the partial pro-drop found in Finnish and Hebrew. The phonologically empty element approach is not pursued here because in general, syntacticians working within HPSG tend to avoid having their analyses depend on phonologically empty

elements. In addition to these theoretical issues, there are practical implications for implemented systems as phonologically empty elements make parsing much more difficult.

Within lexicalist approaches, Bresnan (2001) gives an account of 'pro-drop' and pronoun incorporation for Lexical-functional Grammar (LFG) and uses Chichewa [nya] as an example. Here, if a verbal affix and overt argument cannot co-occur, the phenomenon is treated as pronoun incorporation. The affix obligatorily fills the argument position by supplying a 'pro' predication for the slot. If an affix and overt argument can co-occur, then the affix optionally fills the argument position. If an overt argument is present, it does not provide the 'pro' predication and if there is no overt argument, the 'pro' predication is given. Although LFG uses different nomenclature, this approach is essentially the same as the second approach. For those languages such as Japanese, Chinese, and Korean which do not use verbal argument marking affixes, a similar approach is taken, with the exception that there is no actual phonological element that is being used to trigger the rule which fills the syntactic slot. This analysis has been implemented in computational grammars for Japanese (Masuichi et al., 2003) and then ported for use in a Korean grammar (Kim et al., 2003)

To avoid the phonologically empty 'pro' used in transformational frameworks, many HPSG analyses are similar to the first approach given in §4.1. Ginzburg and Sag (2000), who work within HPSG, also follow the first approach for English and Melnik (2007) does so for Hebrew. In the context of our multilingual resources, this would require a different version of the Argument Realization Principle for each argument optionality pattern.

In an implemented grammar for Maltese [mlt], Müller (2009) takes the second approach for object dropping and the third approach for subject dropping. The second approach treats argument marking affixes as incorporated pronouns that actually

fulfill a valence slot. This approach may be more aesthetically pleasing for languages such as Nkore-Kiga where the the object marker affix does appear to function as an incorporated pronoun.

For the grammars created by the customization system, rather than concentrating on the structure of the parse tree itself, the primary focus of the analysis is to generate a well formed semantic representation and specific steps taken to build the tree are only required in so far as they lead to that eventual goal. This semantic representation is contained in the MRS (Copestake et al., 2005) which is built compositionally as a tree is formed and encodes the semantic relationship between different parts of the sentence. Since the final MRS would be the same whether the valence list is shortened by a lexical rule or a phrase structure rule, for ease of implementation and cross-linguistic generalizability, I have chosen to develop analyses that take the third approach whether an affix functions as an agreement marker or an incorporated pronoun. The third approach–using phrase structure rules–has also been followed by others. For example, Branco and Costa (2008) made use of it in their implemented grammar for Portuguese. My analyses are presented in the following sections and begin by looking at how affix co-occurrence restrictions can be modeled.

4.3 Affix Co-occurrence Restrictions

The analyses of affix and overt argument co-occurrence restrictions for subject and object dropping parallel each other. For ease of exposition, I will discuss the analyses in terms of object dropping; however, they are easily adapted to subject dropping as well.

If a language allows object dropping, then in addition to the `head-comp-phrase` rule, it should also contain a `head-opt-comps-phrase` rule. Instead of a binary rule which requires a head daughter and a non-head daughter such as the `head-comp-phrase`

rule, this rule is unary and empties the COMPS list without a complement being present.

If a language allows object dropping for all verbs and does not have any co-occurrence restrictions on the markers (e.g. markers are required if the argument is dropped and optional otherwise), then either of these two rules would empty the COMPS list and allow the resulting feature structure to serve as the head daughter of the `head-subj-phrase-rule` to combine with its subject.[1] If a language has lexically-based or object marker co-occurrence restrictions, then further care must be taken to ensure that only licensed strings are parsed as grammatical. This is accomplished by constraining the head daughters of `head-comp-phrase` and `head-opt-comp-phrase` rules appropriately.

The constraints fall onto a new feature called OPT.[2] Verbs can constrain the value of OPT for the items on their SUBJ and COMPS lists. OPT can be underspecified as type `bool`, or it can be constrained to + or –. When this feature is set to –, the subject or object constrained is not optional and is not allowed to be dropped. When it is set to underspecified or +, dropping is allowed.

Section §3.1.2 discusses the nine logical possibilities in affix co-occurrence restrictions. The simple cases are when the argument markers have the same distribution, i.e., object markers are required whether or not dropping occurs, optional whether or not dropping occurs, or not permitted whether or dropping occurs. They are handled by simply instantiating the `opt-comp-phrase-rule` and any applicable lexical rules.

Let us consider the case of a language such as Arabic [arb], which for strictly transitive verbs, requires an object marker when an object is dropped, but does not

[1]Or, in the case of OSV or VSO word orders, allowing the SV or VS constituent to serve as a standalone utterance.

[2]By 'new' I am referring to the fact that it had not been manipulated by the grammars output by customization system and is not a commonly used feature within the HPSG community. OPT was, however, included in the feature geometry of the core Matrix.

require it when the object is overt.

(19) muna ishtaraa-t al-kitaab-a
muna.3FS buy.past-3FS.OB DEF-book-ACC

Muna bought the book. [arb]

(20) muna ishtaraa-t-hu
muna.3fs buy.past-3fs.subj-3ms.obj

Muna bought it. [arb]

(21) *muna ishtaraa-t
muna.3FS buy.past-3FS.SUBJ

Muna bought. [arb]

Recall that there are two ways that a transitive verb can empty its COMPS list and meet the constraints on the `head-subj-phrase` rule or the initial symbol: undergo the `head-comp-phrase` rule or undergo the `head-opt-comp-phrase` rule. To ensure that an ungrammatical string which does not contain either an overt object or an object marker is not accepted, verbs which do not have an object marker must be prevented from going through the `head-comp-phrase` rule. This is accomplished by constraining the first item on the COMPS list for the head daughter of the `head-comp-phrase` rule to be OPT –. Example (22) shows an abbreviated avm for this phrase structure rule.

(22) $\begin{bmatrix} \textit{head-comp-phrase} \\ \text{HEAD-DTR.SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.SYNSEM.LOCAL.CAT.HEAD.OPT} \ - \end{bmatrix}$

Conversely, for the `head-opt-comp-phrase` rule, the first item on the head daughter's COMPS list is OPT +. The default is for all verbs to leave the OPT value underspecified. Unless this is changed, any verb would be able to undergo either of these rules and ungrammatical strings such as Example (21) would be accepted. Thus,

the OPT value for different verbs must be set according to whether or not it has an object marker attached. In general, affixation is done through having a lexical item go through a series of lexical rules. In order to ensure that each verb's complement has the correct OPT value, verbs are required to go through an `obj-marker-lex` rule. This rule has subtypes which specify the person, number, and gender for the affix.

For the example sentences, there would be two sub rules. One rule, (`3ms-obj-marker-lex-rule`), specifies a particular affix (*hu*) which constrains the complement to be third person masculine singular. The other rule (`obj-marker-no-drop-lex-rule`) does not attach an affix and constrains the first item on its COMPS list to be OPT –. Since the affix is possible both with an overt object and with a dropped object, we want the result of the `3ms-obj-marker-lex-rule` to be able to undergo either the `head-comp-phrase` rule or the `head-comp-opt-phrase` rule. Thus the OPT value of the complement is left underspecified. Since the object marker is required in order for object dropping to occur, we do not want words that do not have the affix to go through the `head-opt-comp-phrase` rule. The OPT – specified for the `obj-marker-no-drop-lex-rule` conflicts with the OPT + constraint on the `head-opt-comp-phrase` rule and thus words without the affix are not allowed to drop their complements.

Until now, I have shown the lexical items, lexical rules, and phrase structure rules as attribute value matrices; however this is not the representation that is used within the Grammar Matrix customization system. Instead of representing feature structures as avms, The Grammar Matrix customization system writes them in type description language (TDL) (Krieger and Schäfer, 1994). The TDL output for the rules described in the previous section is given in Example (23) below. As is generally true in HPSG typed feature structures, these rules take advantage of the fact that types can inherit from multiple other types as long as the constraints do not conflict with each other. Note that these rules inherit from types that were a part of the Grammar Matrix

before the addition of the library described in this thesis.

(23)
```
obj-marker-lex-rule := lex-to-word-rule &
   [DTR transitive-verb-lex].

3fs-lex-rule := infl-ltow-rule & obj-marker-lex-rule &
   [ SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.LOCAL.CONT.HOOK.INDEX.PNG
                                              [ NUM sing,
                                                PER 3rd,
                                                GEND fem ] ].

obj-marker-no-drop-lex-rule := const-ltow-rule & obj-marker-lex-rule &
   [ SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.OPT - ].

basic-head-comp-phrase :+ [ SYNSEM.LOCAL.CAT.MC #mc,
   HEAD-DTR.SYNSEM.LOCAL.CAT [ MC #mc,
                               VAL.COMPS.FIRST.OPT - ] ].
```

The OPT feature is manipulated in a similar way to reflect the co-occurrence restrictions for the nine logical possibilities. The same strategy is employed for co-occurrence restrictions on subject affixes as well. Table 4.1 summarizes the constraints on lexical items, lexical rules, and phrase structure rules that are specified for each of these logically possible patterns.

Dropped Arg Affix/ Overt Arg Affix	Head-Comp-Rule	No-Marker-Rule	Marker-Rule	Transitive Verb Lex
required/required	underspecified	none	underspecified	INFLECTED–
optional/optional	underspecified	none	underspecified	underspecified
not permitted/ not permitted	underspecified	none	none	underspecified
required/optional	OPT–	OPT–	underspecified	INFLECTED–
optional/ not permitted	OPT–	OPT–	OPT+	underspecified
not permitted/ required	OPT–	OPT+	OPT–	INFLECTED–
required/ not permitted	OPT–	OPT–	OPT+	INFLECTED–
optional/required	OPT–	OPT+	underspecified	INFLECTED–
not permitted/ optional	OPT–	OPT+	OPT–	underspecified

Table 4.1: Constraints associated with logically possible affix co-occurrence

4.4 Context Dependent

As described in §3.1.1, the typological research showed that in some languages subject dropping is restricted to particular syntactic contexts. For these languages, I created a phrase structure rule for each context that inherits from the `decl-head-opt-subj-phrase`[3] rule that was already declared in the core Matrix, but was not output to customized grammars. The new rule further constrains the head daughter so that it can only apply in the correct context. For example, if a language were to allow subject dropping only in the past tense, this rule would not license subject dropping in other tenses.

(24)
```
context1-decl-head-opt-subj-phrase := decl-head-opt-subj-phrase &
    [ HEAD-DTR.SYNSEM.LOCAL.CONT.HOOK.INDEX.E.TENSE past ].
```

4.5 Lexically Based

For a language such as Tamil which does not allow subject dropping for a certain class of verbs, the OPT feature is changed to – for those verb types. Example (25) shows the TDL for verb types which require an overt subject. Types for verbs that allow subject dropping are left underspecified so that a subject can be dropped or not depending on affix co-occurrence restrictions which are handled by the combination of lexical and phrase structure rules described earlier.

(25)
```
no-subj-drop-verb-lex := verb-lex &
    [SYNSEM.LOCAL.CAT.VAL.SUBJ.FIRST.OPT -].
```

[3] According to Grammar Matrix naming conventions, 'decl' is used to denote that this type inherits from the '`declarative-clause`' type. This type is actually underspecified as to whether the clause is a proposition or a question and is instead used to contrast with relative and imperative clauses.

Similarly, for a language that has lexically-based object dropping (see §3.1.2 for examples of how English could be described in this way), verbs which require that their complements be overtly realized would inherit from a type that has the constraints shown in Example (26).

(26)
```
no-comps-drop-verb-lex := transitive-verb-lex &
        [SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.OPT -].
```

4.6 Summary

In this chapter, I have shown that a grammar can be developed to parse strings which represent non-context dependent argument optionality through a combination of phrase structure and lexical rules which manipulate the OPT and INFLECTED features. Furthermore, context dependent argument optionality requires that separate head-opt-subj-phrase rules be created which constrain the appropriate person/number/gender and tense/aspect/mood features for each syntactic context in which argument dropping is allowed. In addition, constraints can be placed on lexical types so that languages which exhibit lexically-based argument dropping can constrain which verbs allow subjects and or object dropping. While each of the argument optionality patterns discussed in this chapter have been analyzed in terms of one or more of the frameworks presented in §4.2, to my knowledge, this is the first attempt at providing a cohesive set of analyses that, taken together, account for all of the patterns discussed and thus provides a computationally tractable way of modeling argument optionality in a multi-lingual environment. The next chapter describes how these analyses were implemented in the Grammar Matrix customization system.

Chapter 5

IMPLEMENTATION

5.1 Overview

The goal in adding an argument optionality library to the customization system is to accurately model the way that non-overt subjects and objects are treated in the language that the user-linguist is describing. As stated earlier in §2.2.3, the user interacts with the Grammar Matrix customization system through a dynamic web-based questionnaire which is divided into several pages. The review of the typological literature (see Section §3) showed that there is considerable variation in how argument optionality is realized in different languages. This dictated the development of various analyses that account for the different patterns. Determining how argument optionality is realized in a language and consequently which of the analyses presented in Chapter 4 is appropriate cannot be done with a simple yes or no question. Thus, in order to elicit the information necessary to develop a grammar which allows for argument optionality, I added an Argument Optionality page to the questionnaire.[1]

This chapter details the creation of the webpage and how the choices that the user makes are collected and used to generate and instantiate new phrase-structure and lexical rules as well as place new constraints on existing rules and lexical items. The modifications made to the customization script are illustrated by walking through three sample choices files.

---

[1]The page can be viewed at http://uakari.ling.washington.edu/matrix/ssaleem/matrix.cgi?subpage=arg?opt

5.2 Questionnaire

The Argument Optionality page is divided into three major sections. The first section gives a brief explanation of argument optionality including examples from a few languages. The user-linguist is prompted to continue reading the page and answer the questions on it if argument optionality exists in the language being described. If argument optionality does not exist in the language, the user-linguist is directed to leave the page blank.

The second section asks a series of questions about subject dropping that are designed to cover the variation found in typological literature.

- Is subject dropping allowed for all verbs or is it lexically based?
- Are there verbal affixes which contain information about the subject?
- What is the co-occurrence of subject marking verbal affixes (assuming they exist) with independent noun phrases?
- Are there restrictions on subject dropping that depend on person/number/gender and tense/aspect/mood contexts?

The third section asks a similar set of questions about object dropping. The primary difference between the subject and object dropping sections is the question about contexts. I did not find descriptions of any languages where object dropping was disallowed for certain person/number/gender combinations or tense/aspect/mood environments. There was evidence of object dropping being allowed in certain coordination structures (e.g. Icelandic in Rögnvaldsson, 1990); however that is outside the scope of this project and would probably require a different analysis as the dropped object of the second verb would need to be identified in the semantic representation

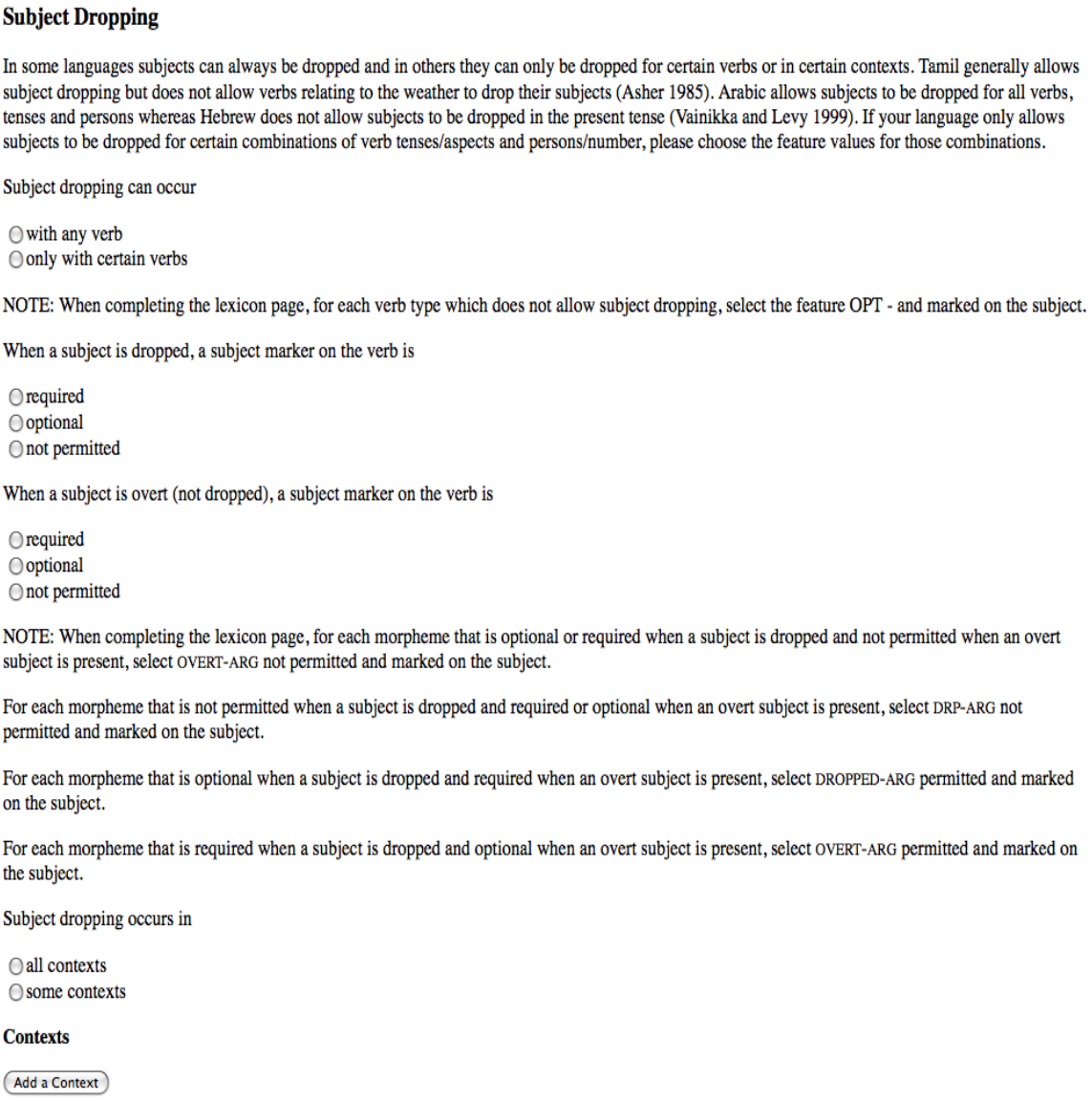
with the overt object of the first verb. Thus, context dependent object dropping is not supported by the current implementation; however since it would parallel context dependent subject dropping, it would be very easy to add this functionality if evidence of its usefulness is ever found.

The implementation of the analyses takes advantage of the dynamic nature of the questionnaire by making certain features available on the Lexicon page depending on the selections made on the Argument Optionality page. I added three new features which are only necessary for certain argument optionality patterns. So that the user is not bothered with unnecessary (and possible error-generating) choices, these features are only enabled if the user describes his or her language as having one these patterns.

The new features are OPT, OVERT-ARG, and DRP-ARG. OPT appears when the user describes the language as having lexically based argument optionality. The user is directed to specify [OPT +] for verb types which allow dropped arguments and [OPT –] for those which do not. OVERT-ARG appears if the user describes the language as having affixes which are required for dropped arguments and optional or not permitted for overt arguments. DRP-ARG appears if the user describes the language as having affixes which are required for overt arguments but are optional or not permitted for dropped arguments. It is important to note that while OPT corresponds to an actual feature in the Grammar Matrix, OVERT-ARG and DRP-ARG do not. They are simply flags which prompt the creation of certain lexical and/or phrase structure rules depending on their presence or absence.

5.2.1 Lexically Based

Figure 5.2 shows a partial choices file for a pseudo-language with choices indicating that it has lexically-based subject dropping. Figure 5.3 shows the lexicon section of the same choices file. Note that `verb-type2` gives the OPT feature the value 'plus'



Subject Dropping

In some languages subjects can always be dropped and in others they can only be dropped for certain verbs or in certain contexts. Tamil generally allows subject dropping but does not allow verbs relating to the weather to drop their subjects (Asher 1985). Arabic allows subjects to be dropped for all verbs, tenses and persons whereas Hebrew does not allow subjects to be dropped in the present tense (Vainikka and Levy 1999). If your language only allows subjects to be dropped for certain combinations of verb tenses/aspects and persons/number, please choose the feature values for those combinations.

Subject dropping can occur

- with any verb
- only with certain verbs

NOTE: When completing the lexicon page, for each verb type which does not allow subject dropping, select the feature OPT - and marked on the subject.

When a subject is dropped, a subject marker on the verb is

- required
- optional
- not permitted

When a subject is overt (not dropped), a subject marker on the verb is

- required
- optional
- not permitted

NOTE: When completing the lexicon page, for each morpheme that is optional or required when a subject is dropped and not permitted when an overt subject is present, select OVERT-ARG not permitted and marked on the subject.

For each morpheme that is not permitted when a subject is dropped and required or optional when an overt subject is present, select DRP-ARG not permitted and marked on the subject.

For each morpheme that is optional when a subject is dropped and required when an overt subject is present, select DROPPED-ARG permitted and marked on the subject.

For each morpheme that is required when a subject is dropped and optional when an overt subject is present, select OVERT-ARG permitted and marked on the subject.

Subject dropping occurs in

- all contexts
- some contexts

Contexts

Add a Context

Figure 5.1: Subject dropping portion of Argument Optionality webpage

while `verb-type4` gives it the value 'minus'. As described in §2.2.3, when a user clicks on the 'Create Grammar' button, the customization script is called. In order to generate a grammar that licenses lexically-based subject dropping, the implemented

```
section=arg-opt
subj-drop=subj-drop-lex
subj-mark-drop=subj-mark-drop-not
subj-mark-no-drop=subj-mark-no-drop-not
subj-con=subj-con-always
```

Figure 5.2: Argument Optionality portion of a choices file

Argument Optionality library does two things:

(i). The absence of the 'subj-drop-con-some' flag triggers the output of '`decl-head-opt-subj := decl-head-opt-subj-phrase.`' to the rules file.

(ii). The presence of

```
verb3_feat1_name=OPT
verb3_feat1_value=minus
verb3_feat1_head=subj
```

triggers constraining the OPT value to – for verb-type3.

The system outputs

```
verb-type3-verb-lex := intransitive-verb-lex &
[SYNSEM.LOCAL.CAT.VAL.SUBJ.FIRST.OPT -].
```

Item (i) was accomplished by creating a new arg_opt() function and placing within it a command to add this TDL to the rules file given the previously mentioned triggers. Item (ii) was accomplished by modifying customize_feature_values(), the existing function which constrains feature values, to check for the OPT feature as well and constrain it to '-' if the input value is 'minus'. This new constraint is merged with other information given about this type and this results in the TDL shown in (ii). The

```
section=lexicon
noun1_det=imp
noun1_stem1_orth=n1
noun1_stem1_pred=_n1_n_rel
noun2_det=imp
noun2_stem1_orth=n2
noun2_stem1_pred=_n2_n_rel
verb1_name=verb-type1
verb1_feat1_head=subj
verb1_valence=intrans
verb1_stem1_orth=iverb1
verb1_stem1_pred=_iv_v_rel
verb2_name=verb-type2
verb2_feat1_name=OPT
verb2_feat1_value=plus
verb2_feat1_head=subj
verb2_valence=trans
verb2_stem1_orth=tverb1
verb2_stem1_pred=_tv_v_rel
verb3_name=verb-type3
verb3_feat1_head=subj
verb3_valence=intrans
verb3_stem1_orth=iverb2
verb3_stem1_pred=iv2_v_rel
verb4_name=verb-type4
verb4_feat1_name=OPT
verb4_feat1_value=minus
verb4_feat1_head=subj
verb4_valence=trans
verb4_stem1_orth=tverb2
verb4_stem1_pred=tv2_v_rel
```

Figure 5.3: Lexicon portion of a choices file

system functions analogously for lexically-based object dropping.

5.2.2 Context Dependent

If subject dropping is dependent upon the syntactic context, then the user is prompted to specify the syntactic contexts in which it is allowed. Once the user clicks on the 'Add a context' button he or she is presented with a set of drop down boxes that contain features and values previously specified in other parts of the questionnaire. The user is able to select those features and their associated values which are compatible with subject dropping. For example, if the user is describing a hypothetical language which only allows subject dropping in the past tense when the subject is in the third person, then the portion of the choices file associated with these answers would look like Figure 5.4. To create the appropriate TDL, the new

```
subj-con=subj-con-some
context1_feat1_name=tense
context1_feat1_value=past
context1_feat1_head=verb
context1_feat2_name=person
context1_feat2_value=3rd
context1_feat2_head=subj
```

Figure 5.4: Partial choices file for language with context dependent subject dropping.

arg_opt() function collects the features and values for each context and creates a version of the `decl-head-opt-subj-phrase rule` that incorporates these constraints on the head daughter. Adding the constraints was done by further modification to the customize_feature_values() function described in §5.2.1. When dealing with a context,

```
if afffix-overt-subj-not-permitted and
(affix-drop-subj-optional or affix-drop-subj-required) or
if affix-overt-subj-required and
(affix-drop-subj-optional or affix-drop-subj-not-permitted)
language.add('basic-head-subj-phrase
             :+ [HEAD-DTR.SYNSEM.LOCAL.CAT.VAL.SUBJ.FIRST.OPT -].')
```

Figure 5.5: Code that checks affix co-occurrence patterns

the modified function sets the feature geometry to begin with 'HEAD-DTR.SYNSEM' to ensure that the constraints that are given in the choices file apply to the head daughter of the rule instead of the mother.

5.2.3 Affix co-occurrence restrictions

For certain affix co-occurrence restrictions it is necessary to constrain the `head-subj-phrase` and/or `head-obj-phase` rules to have the first item on the head daughter's SUBJ or COMPS list be OPT – so that verbs which either have (or do not have depending on the restrictions) the affix are not allowed to combine with an overt subject (or object). This is accomplished by checking for each of these co-occurrence patterns and adding an addendum to the type definition of the `head-subj-phrase` rule. A slightly modified version of the code that used to implement this check is shown in Figure 5.5.

In addition to the new constraints on the phrase structure rules, for these affix co-

occurrence patterns, a new lexical rule subtype needs to be created to add additional constraints on the verbs which do not contain the affix so that they are prevented from going through the wrong phrase structure rule. Since this occurs at the lexical-rule level, on the Argument Optionality page, users are given further instructions on how to complete the Lexicon page. If they have described the language as having one of these affix co-occurrence patterns, users are directed to declare whether an overt argument or dropped argument is permitted for these affixes.

To implement this, I modified the existing morphotactics code that is used to create lexical rules. O'Hara (2008) gives details about the structure of this code. My modifications consisted of inserting code that searched the choices file for the presence of the flags associated with these affix co-occurrence patterns and adding the TDL for the appropriate non-inflectional rule to the language.tdl file.

```
if affix-overt-subj-optional and affix-dropped-subj-required:
ltype = name + '-no-drop-lex-rule'
language.add(lytpe + ':= '+ supertype)
language.add(lytpe '':= [SYNSEM.LOCAL.CAT.VAL.SUBJ.FIRST.OPT -].')

if affix-overt-subj-required and affix-dropped-subj-optional:
ltype = name + '-no-drop-lex-rule'
language.add(lytpe + ':= '+ supertype)
language.add(lytpe '':= [SYNSEM.LOCAL.CAT.VAL.SUBJ.FIRST.OPT +].')
```

5.3 Summary

To implement the analyses described in Chapter 4, I created a new page on the questionnaire that elicits information about the way in which argument optionality is realized in the language being modeled. The page directs the user to choose whether argument optionality is lexically-based, has any syntactic contextual constraints, or has co-occurrence constraints on affixes appearing the overt and dropped subjects. I also added a new function to the customize script and modified two existing functions in order to instantiate phrase structure rules, create new lexical rules, and add constraints to lexical types and rules where appropriate.

Chapter 6

EVALUATION

6.1 Overview

The project underwent a four-tiered evaluation process. First, I ensured that the changes and additions to the Grammar Matrix customization system did not adversely affect the system by following the established regression testing procedures. Next, the logical possibilities for affix and full noun phrase co-occurrence and person/number/gender, tense/aspect/mood and lexical constraints were tested. Then, I verified that the system was able to accurately model the argument optionality patterns evidenced in the languages which informed the design of the library (Arabic, Hausa, Tamil, and Finnish). These test suites were designed to focus on the argument optionality library and so only include test items that directly relate to it. Finally, to determine whether the library can be used to model languages that were not considered during the design and to see how it interacts with other linguistic phenomena, I also tested the way that it can be used to model six additional languages using more elaborate test suites that were designed to test other libraries as well (e.g., case, tense, agreement, etc.).[1]

For each tier, I developed unit tests which consist of a test suite and associated choices file. The choices files were developed by answering the questionnaire. I then customized a grammar using the customization system, loaded this grammar into the

[1]I used the test suites and choices files created for an as yet unpublished paper by Bender et al. (2010) as a base and added items related to argument optionality to the test suites and filled out the Argument Optionality page on the questionnaire.

LKB (Copestake, 2002), a grammar development environment, and evaluated the test suite coverage using [incr tsdb()] (Oepen, 2001), a system which supports systematically evaluating and benchmarking grammars. A more detailed description of how the unit tests for each tier were created follows. For each tier, aside from regression testing, a sample choices file and test suite are included in Appendix A.

6.2 Regression Testing

To ensure that new additions work properly and do not interfere with the existing system, before adding changes to the live version, Matrix developers are responsible for creating unit-tests which test the functionality that they are adding as well as successfully passing old unit tests. I verified that my modified system parsed all existing unit tests from other libraries and then added the test suites created to test the logical possibilities as my unit-tests.

6.3 Logical Possibilities

I created unit tests to test: lexically vs. non-lexically-based subject and object dropping, all logical possibilities for affix co-occurrence restrictions for subject and object dropping, and context-restricted subject dropping. Each unit test consists of a test suite of grammatical and ungrammatical strings in a pseudo-language that has the properties being tested and an associated choices file. Some example strings from the test suite for a pseudo-language that does not allow object dropping but has lexically-based subject dropping with a subject affix required for both dropped and overt subjects are given in Figure 6.1.

After creating the test suite, I answered the questionnaire. On the Argument

Grammatical	Ungrammatical
iverb1-subj-3fs	*iverb2-subj-3fs
n1 iverb1-subj-3fs	*iverb2-subj-3fs n2
n1 iverb2-subj-3fs	*iverb1-subj-3fs n1
n2 iverb1-subj-3fs	*iverb2-subj-3fs n1
n2 iverb2-subj-3fs	*tverb2-subj-3fs n2
n1 tverb1-subj-3fs n2	*tverb1-subj-3fs
n1 tverb2-subj-3fs n2	*tverb2-subj-3fs
tverb1-subj-3fs n2	*n1 tverb1-subj-3fs
	*n2 tverb1-subj-3fs
	*n1 tverb2-subj-3fs
	*n2 tverb2-subj-3fs
	*iverb1

Figure 6.1: Example strings from a test suite for a pseudo-language

Optionality page, for this same pseudo-language, I chose:

- Subject dropping can occur only with certain verbs
- When a subject is dropped a subject marker is required
- When a subject is overt a subject marker is required
- object dropping is not allowed

One the lexicon page, I defined:

- Two noun types n1 and n2
- Four verb types
  - one intransitive verb type with the OPT + feature on the subject
  - one intransitive verb type with the OPT - feature on the subject
  - one transitive verb type with the OPT + feature on the subject
  - one transitive verb type wit the OPT - feature on the subject
- One verb slot which has one morpheme which specifies third person, feminine, singular on the subject.

Tables 6.1 and 6.2 show the patterns tested in unit tests for object and subject dropping respectively. I uncovered a few bugs through running these unit tests and made the changes necessary to get 100% coverage of all grammatical strings and 0% overgeneration of ungrammatical strings. The next step was to see how the system fared with modeling the ways that argument optionality is actually realized in natural languages.

	Dimensions		
	Lexically Based	Affix w/Drop	Affix w/o Drop
Test 1, 10	no, yes	not permitted	optional
Test 2, 11	no, yes	not permitted	required
Test 3, 12	no, yes	optional	optional
Test 4, 13	no, yes	optional	required
Test 5, 14	no, yes	required	not permitted
Test 6, 15	no, yes	required	optional
Test 7, 16	no, yes	required	required
Test 8, 17	no, yes	not permitted	not permitted
Test 9, 18	no, yes	optional	not permitted

Table 6.1: Unit tests for object dropping

6.4 Verification

For the third tier of the evaluation process, I selected four natural languages which exhibited interesting argument optionality patterns and developed a test suite and choices file for each. The patterns found in these languages were considered during the development of the library and indeed provided the motivation for some of the features. For Finnish, Hausa, and Tamil, the test suites and answers to the customization system questionnaire were based on information obtained from reference and descriptive grammars. For Arabic, I primarily relied on my own understanding of the language. A brief description of the argument optionality patterns found in each of these languages follows along with a discussion of the customized grammars' coverage over the test suites.

	Dimensions			
	Lexically Based	Affix w/Drop	Affix w/o Drop	Context Dependent
Test 19, 30	no, yes	not permitted	optional	no
Test 20, 31	no, yes	not permitted	required	no
Test 21, 32	no, yes	optional	optional	no
Test 22, 33	no, yes	optional	required	no
Test 23, 34	no, yes	required	not permitted	no
Test 24, 35	no, yes	required	optional	no
Test 25, 36	no, yes	required	required	no
Test 26, 37	no, yes	not permitted	not permitted	no
Test 27, 38	no, yes	optional	not permitted	no
Test 28	yes	required	required	one feature
Test 29	yes	required	required	two features

Table 6.2: Unit tests for subject dropping

Arabic [arb] (Semitic): Person, number, and gender marking are always required for subjects. Free pronouns are usually dropped for both subjects and objects; however, they can occur for emphasis or stylistic purposes. For strictly transitive verbs, when no overt object is present, an affix marking person, number and gender of the implied direct object is required. There appears to be a class of verbs (similar to English as explained in §3.1.2, which allows objects to be dropped without leaving an object marker (Suleiman, 1990).

Finnish [fin] (Uralic): According to Sulkala and Merja (1992), first and second person pronouns are often dropped when they occur in the subject position. When used as a subject, first or second person free pronouns have an emphatic or contrastive meaning. Third person subject pronouns are usually required for a referential interpretation; however there are contexts in which dropping is allowed. If a speaker is referring to himself/herself with the third person form, then dropping is allowed. This type of anaphora resolution is not a part of the Grammar Matrix and is outside the scope of this project. Third person singular pronouns are dropped when a generic impersonal meaning is intended. In Finnish, the second person and first person agreement morphemes are non-zero. The zero agreement morpheme can be interpreted as a specific reference to an entity/person that is not the speaker or hearer or as a generic impersonal construction that is non-referential. Since the fourth person is often used to denote syntactic distinctions between generic and specific referents, I treated the presence of an overt subject as an instance of the third person and the absence of an overt subject as the fourth person. Thus, the grammar developed for verification analyzes Finnish as allowing subject dropping in the first, second and fourth persons and not permitting it for the third person.

There are also word order constraints. Finnish is canonically SVO; however other word orders are possible. According to Vainikka and Levy (1999), when a first or

second person subject is dropped, VO order is maintained. This is not the case in sentences that use what I am referring to as the fourth person–the zero subject marker occurring without an overt subject. In this context, an element in the VP (direct object, indirect object, or adverb) must be fronted. V-initial word order is prohibited in this circumstance. While subjects are marked on the verb, objects are not; however they can be dropped from some verbs with a generic interpretation as well. The interpretation appears to resemble Fillmore's (1986) description of indefinite null instantiation discussed in §3.1.2.

Hausa [hau] (Chadic): Newman (2000) states that Hausa requires that free pronoun subjects be dropped. Sentences containing free pronouns in the subject position are considered ungrammatical. The Hausa verb phrase consists of person aspect complex (PAC) followed by a lexical verb and optional objects and adjuncts. For some PACs, tense, aspect, and mood are morphologically segmentable from the person, number, gender markers. In these cases, the person number gender markers are optional when an overt full noun phrase is the subject and required when none is present. Although unbound pronouns cannot occur in the subject position, independent pronouns may appear as direct objects. Person, number, and gender are not marked for objects; however, the form that the verb takes is dependent on whether a direct object follows it and if so whether it is a pronoun or not. This form does not contain information about person number and gender. Subject dropping is required for all verbs and there is no evidence that object dropping is lexically-based.

Tamil [tam] (Dravidian): According to Asher (1985), Tamil allows both subjects and objects to be dropped. Information about direct objects is not marked on the verb; however, subject person, number, and gender markers are always required whether or not the subject is overt. There is a special class of verbs associated

with the weather that do not allow subject dropping. For the purposes of this thesis Tamil is analyzed as having lexically based subject dropping and non-lexically based object dropping.

Language.	Test Items	Grammatical	Ungrammatical	Coverage/Overgeneration
Arabic	13	10	3	90%/0%
Finnish	11	9	3	100%/0%
Hausa	20	8	12	100%/0%
Tamil	7	5	2	100%/0%

Table 6.3: Verification Results

6.5 Held-out languages

Six languages that had not been considered during the library's development were chosen for this tier of the evaluation process. To minimize the influence of areal and genetic influences, the languages each come from a different language family and are primarily spoken in very different areas of the world (North America, Greenland, Africa, Europe, Asia, and Australia). For this tier of the evaluation process, I began with test suites and choices files that were used to test the previous Matrix libraries and extended the test suites to include items demonstrating argument optionality and the choices file to include answers to questions related to argument optionality. Table 6.4 shows the results of test items related to argument optionality. Adding the capability to handle the various argument optionality patterns did not cause any loss of coverage or additional overgeneration in unrelated test items. More detailed information about the way that argument optionality is realized in each language and how the argument optionality library was able to handle the patterns follows.

Language	Items	Grammatical	Ungrammatical	Coverage	Overgeneration
Abkhaz	10	6	4	100	10
Chemehuevi	8	6	2	83.3	0
Jingulu	9	6	3	100	0
Malayalam	4	4	0	100	0
Nkore-Kiga	10	4	6	100	83.3
W. Greenlandic	5	3	2	100	0

Table 6.4: Evaluation results of attested realizations in six held-out languages

Abkhaz [abk] (North Caucasian) (Hewitt, 1979): Abkhaz allows unexpressed noun phrases for subject, direct object, indirect object, and possessor in the genitive construction. Free pronouns can occur in non-emphatic contexts. Subject agreement is marked on intransitive and transitive verbs and direct object agreement is marked on transitive verbs. There are word order constraints on the appearance of the direct object marker. If a non-human third person singular or third person plural direct object immediately precedes the verb, then the object marker is mandatorily deleted. This is not the case for other persons/genders.

100% coverage was achieved for this library; however one item overgenerated. This was due to the fact that the system currently does not support morphological changes based on word order; therefore, the grammar is unable to rule out strings in which the direct object is non-human or third person plural and immediately precedes the verb.

Chemehuevi [ute] (Uto-Aztecan) Press (1979): Chemehuevi allows subject dropping. If the subject is not overt, then an enclitic must be attached to the first word in the sentence. This corresponds to the third pattern of pronominal subject realization

discussed in §3.1.1. There was no explicit statement that overt NP objects were required for transitive verbs; however, examples were given of ditransitive verbs which allowed both objects to be dropped. I interpreted Press's statement that verbs are marked for transitivity and some allow for elliptical arguments such as *tIka* "eat" to mean that object dropping is lexically based. There was no mention that argument dropping depended on person/number/gender or tense/aspect/mood.

The system does support second position clitics and, thus, as mentioned in §3.1.1 supporting the pattern of subject dropping where pronominal subjects are realized as clitics on variable hosts falls outside the scope of this project. Therefore, subject dropping in Chemehuevi was not well supported; however, the argument optionality library was able to capture the lexically-based object dropping. Overall coverage was 83.3% and no items were overgenerated.

Jingulu [jig] (Australian) (Pensalfini, 2003): Jingulu allows subject and object dropping. Free pronouns are usually used in emphatic or contrastive contexts. Person and number agreement are marked on the verb for both objects and subjects. This marking is required whether or not subject and/or object dropping has occurred. There was no evidence of lexical constraints on either type of dropping.

This was a simple pattern of having subject/object markers required at all times. The system is able to correctly model this pattern. There was 100% coverage of the argument optionality and other libraries and no overgeneration.

Malayalam [mal] (Dravidian) (Asher and Kumari, 1997): Malayalam allows subject and object dropping. This often occurs in contexts where the referent is already known. First and second person pronouns are usually dropped. Since it appears that it is possible for free pronouns to occur in non-emphatic contexts, Malayalam is clas-

sified as having optional (instead of preferred) subject and object dropping. Unlike the other languages in this survey, Malayalam does not mark person, number, and gender agreement on the verb for either subjects or objects.

There was 100% support for this pattern in the argument optionality library with no overgeneration.

Nkore-Kiga [nyn] (Bantoid) (Taylor, 1985): In Nkore-Kiga, free pronouns only appear in emphatic contexts; they are dropped otherwise. Nkore-Kiga is a Bantu language spoken in Uganda. Like other languages within this family there is an elaborate noun class system. According to Taylor (1985), Nokore-Kiga has 17 noun classes. The noun class that the subject belongs to is always marked on the verb. This information is also marked on the verb for objects in certain syntactic contexts. The canonical word order is SVO. The object is not allowed to appear before the subject; however, it can appear before or after the verb (SOV, VSO). If the object appears before the verb then the noun class is marked on the verb. The noun class is also marked on the verb if the object is a free pronoun. If the object is not a free pronoun and appears after the verb, the noun class is not marked.

As with Abkhaz, the library was not able to accurately model the interaction between word order constraints and argument optionality. Requiring object markers to appear when the object was fronted before the verb was not supported. Overall coverage was 100% and overgeneration was 83.3%. The overgeneration was high because free word order was chosen even though this is not actually an accurate way of modeling Nkore-Kiga. If only one of the six permutations of SVO were chosen, overall coverage would decrease and overgeneration would be eliminated.

West Greenlandic [kal] (Inuit) (Fortescue, 1984): In West Greenlandic, free pro-

nouns only appear in emphatic contexts. Person and number are always marked on the verb for both objects and subjects. There was no evidence of lexical or person/number/gender constraints on either type of dropping.

The new library was able to accurately model this pattern. There was full coverage of the argument optionality related test items. Overall coverage was 100% and no items were overgenerated.

Overall the argument optionality library performed well on these languages. It is difficult to generalize from such a small sample, but the fact that all of the languages were at least partially supported and three of languages were fully supported (Jingulu, Malayalam, West Greenlandic) suggests that the Argument Optionality library is able to (partially) model a wide range of typologically diverse languages. Furthermore, it succeeded in modeling all of the argument optionality patterns deemed to be within the scope of the project. The languages that it was unable to fully model had argument optionality patterns that depended upon word order constraints (e.g. second position clitics in Chemehuevi and overt pronominal and nominal arguments appearing in different positions in Nkore-Kiga). As stated in §3.1.1, these types of patterns fall outside the scope of this project. The ability to more accurately model languages which exhibit these patterns will require improvements in the word order library.

6.6 Summary

With the addition of the argument optionality library, the Grammar Matrix customization system was able to at least partially model disparate argument realization strategies in all ten natural languages that it was tested against. 100% coverage was obtained for eight of the ten languages while only two of the natural languages had

	Object Dropping	Subject Dropping	Word Order Constraints	Lexical Constraints
Abkhaz	opt	opt	yes	none
Chemehuevi	opt	opt	none	yes
Jingulu	opt	opt	none	none
Malayalam	opt	opt	no	none
Nkore-Kiga	pref	pref	yes	none
West Greenlandic	pref	pref	none	none

Table 6.5: Existence of and constraints on argument optionality in six languages

one or more test items which overgenerated. The library was also able to work as expected on the logically possible realizations that are given by different combinations of lexical, context, and affix co-occurrence restrictions. Improvements to the library still need to be made particularly where it comes to the semantic distinctions in the various argument optionality patterns that languages employ. Further work also needs to be done to extend coverage to word order constraints. This will likely be done in combination with overall extensions to the word order library.

Chapter 7
CONCLUSION

The typological literature discussed in this thesis showed that argument dropping is a widespread phenomenon that is realized in a variety of ways depending on the language. Languages differ in whether licensing is dependent upon the verb, syntactic context, and whether there are co-occurrence restrictions on the appearance of certain affixes and overt/dropped arguments. I presented a set of HPSG analyses that are able to model these varied patterns and then described how these analyses were implemented in the Grammar Matrix Customization system. A four-tiered evaluation which included testing the system on pseudo-languages designed to test specific aspects of the implementation as well as natural languages showed that it was able to adequately model the argument optionality patterns demonstrated in all of the pseudo-languages and most of the natural languages while maintaining the performance of existing libraries. Further work is required in order to accurately capture licensing constraints that interact with word order. In addition, no attempt was made to model the semantic differences between the presence and absence of an overt argument. These are two major areas in which the Argument Optionality library could be improved; however, to my knowledge, this is the first attempt to provide a comprehensive set of analyses that model the different dimensions in licensing argument optionality that are implemented in a deep, precision grammar and evaluated for accuracy and coverage.

References

Ackema, Peter, Patrick Brandt, Maaike Schoorlemmer, and Fred Weerman, editors. 2006. *Arguments and Agreement*. Oxford University Press, Oxford.

Ajdukiewicz, Kazimierz. 1935. Die syntaktische konnexität. *Studia Philosophica*, 1:1–27.

Asher, R.E. 1985. *Tamil*. Croom Helm, London.

Asher, R.E. and T.C. Kumari. 1997. *Malayalam*. Routledge, New York.

Bar-Hillel, Yehoshua. 1953. A quasi-arithmetical notation for syntactic description. *Language*, 29:47–58.

Bender, Emily M., Scott Drellishak, Antske Fokkens, Lauri Poulson, and Safiyyah Saleem. 2010. Grammar customization. ms University of Washington and Universität des Saarlandes.

Bender, Emily M. and Dan Flickinger. 2005. Rapid prototyping of scalable grammars: Towards modularity in extensions to a language-independent core. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing IJCNLP-05 (Posters/Demos)*, Jeju Island, Korea.

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan.

Bender, Emily M. and David Goss-Grubbs. 2008. Semantic representations of syntactically marked discourse status in crosslinguistic perspective. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 17–29.

Branco, António and Francisco Costa. 2008. A computational grammar for deep linguistic processing of portuguese: LXGram, version A.4.1. Technical report. University of Lisbon, Department of Informatics.

Bresnan, Joan. 2001. *Lexical Functional Syntax*. Blackwell, Boston.

Chomsky, Noam. 1982. *Lecture on government and binding: the Pisa lectures*. Foris publications, Holland.

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, CA.

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language & Computation*, 3(4):281–332.

Corbett, Greville G. 2003. Agreement: The range of the phenomenon and the principles of the surrey database of agreeemnt. *Transactions of the Philological Society*, 101.

Drellishak, Scott. 2009. *Widespread But Not Universal: Improving the Typological Coverage of the Grammar Matrix*. Ph.D. thesis, University of Washington.

Dryer, Matthew. 2008. Expression of proniminal subjects. In Martin Haspelmath, Matthew Dryer, David Gil, and Bernard Comrie, editors, *The World Atlas of Language Structures Online*. Max Planck Digital Library, chapter 101.

Fillmore, Charles. 1986. Pragmatically controlled zero anaphora. In *Proceedings of the 12th annual meeting of the Berkeley Linguistics Society*, pages 95–107.

Fillmore, Charles J., Christopher R. Johnson, and Miriam R.L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16:235–250.

Fortescue, Michael. 1984. *West Greenlandic*. Croom Helm, London.

Ginzburg, Johnathan and Ivan Sag. 2000. *Interrogative Investigations*. CSLI, Stanford.

Haspelmath, Martin, Matthew S. Dryer, David Gil, and Bernard Comrie, editors. 2008. *The World Atlas of Language Structures Online*. Max Planck Digital Library, Munich. http://wals.info.

Hewitt, B. G. 1979. *Abkhaz*. Croom Helm, Amsterdam.

Kim, Roger, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, Hiroshi Masuichi, and Tomoko Ohkuma. 2003. Multilingual grammar development via grammar porting. In *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development*, pages 49–56.

Krieger, Hans-Ulrich and Ulrich Schäfer. 1994. TDL - A Type Description Language for constraint-based grammarsanguage for constraint-based grammars. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 893–899, Kyoto, Japan.

Liu, Dilin. 2008. Intransitive or object deleting? Classifying English verbs used without an object. *Journal of English Linguistics*, 36(4):289–313.

Masuichi, Hiroshi, Tomoko Ohkuma, Hiroki Yoshimura, and Yasunari Harada. 2003. Japanese parser on the basis of the lexical-functional grammar formalism and its evaluation. In Kim Teng Lua Dong Hong Ji, editor, *Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17)*, pages 298–309.

Melnik, Nurit. 2007. Extending partial pro-drop in modern hebrew: A comprehensive analysis. In Stefan Müller, editor, *The Proceedings of the 14th International Conference on Head-Driven Phrase Structure Grammar*, pages 173–193, Stanford. CSLI Publications.

Morimoto, Yukiko. 2002. Prominence mismatches and differential object marking in bantu. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG02 Conference*, pages 292–314.

Müller, Stefan. 2009. Towards an HPSG analysis of Maltese. In Bernard Comrie, Ray Fabri, Beth Hume, Manwel Mifsud, Thomas Stolz, and Martine Vanhove, editors, *Introducing Maltese Linguistics. Papers from the 1st International Conference on Maltese Linguistics*. John Benjamins, Amsterdam, pages 83–112.

Newman, Paul. 2000. *The Hausa Language: An encyclopedic reference grammar*. Yale University Press, New Haven.

Oepen, Stephan. 2001. [incr tsdb()] — Competence and performance laboratory. User manual. Technical report, COLI, Saarbrücken, Germany.

O'Hara, Kelly. 2008. A morphosyntactic infrastructure for a grammar customization system. Master's thesis, University of Washington.

Pensalfini, Rob. 2003. *A Grammar of Jingulu: An Aboriginal language of the Northern Territory*. Pacific Linguistics, Canberra.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.

Press, Margaret. 1979. *Chemehuevi: A grammar and lexicon*. University of California Press, Berkeley.

Rizzi, Luigi. 1986. Null objects in Italian and the theory of pro. *Linguistic Inquiry*, 17(3):501–557.

Rögnvaldsson, Eiríkur. 1990. Null objects in Icelandic. In Joan Maling and Annie E Zaenen, editors, *Modern Icelandic Syntax*. Academic Press, San Diego, pages 367–379.

Sag, Ivan A., Thomas Wasow, and Emily M. Bender. 2003. *Synactic Theory: A Formal Introduction*. CSLI, Stanford, CA, second edition.

Suleiman, Saleh M. 1990. The semantic functions of object deletion in classical Arabic. *Language Sciences*, 12(2-3):255–266.

Sulkala, Helena and Karjalaninen Merja. 1992. *Finnish*. Routledge, New York.

Taylor, Charles. 1985. *Nkore-Kiga*. Croom Helm, London.

Vainikka, Anne and Yonata Levy. 1999. Empty subjects in Finnish and Hebrew. *Natural Language and Linguistic Theory*, 17:613–671.

Appendix A

EVALUATION TEST SUITES AND CHOICES FILES

Choices file for a pseudo-language with lexically-based subject dropping and a subject marker that is always required

```
version=17

section=general
language=arg-opt-lex-subj-drop-marker-req-wth-drop-req-wthout
archive=no

section=word-order
word-order=svo
has-dets=no
has-aux=no

section=number
number1_name=sing
number2_name=plural

section=person
person=1-2-3
first-person=none
```

section=gender
gender1_name=fem
gender2_name=masc

section=case
case-marking=none

section=direct-inverse

section=tense-aspect

section=other-features

section=sentential-negation

section=coordination

section=matrix-yes-no

section=arg-opt
subj-drop=subj-drop-lex
subj-mark-drop=subj-mark-drop-req
subj-mark-no-drop=subj-mark-no-drop-req
subj-con=subj-con-always

section=lexicon

noun1_det=imp

noun1_stem1_orth=n1

noun1_stem1_pred=_n1_n_rel

noun2_det=imp

noun2_stem1_orth=n2

noun2_stem1_pred=_n2_n_rel

verb1_name=verb-type1

verb1_feat1_name=OPT

verb1_feat1_value=plus

verb1_feat1_head=subj

verb1_valence=intrans

verb1_stem1_orth=iverb1

verb1_stem1_pred=_iv_v_rel

verb2_name=verb-type2

verb2_feat1_name=OPT

verb2_feat1_value=plus

verb2_feat1_head=subj

verb2_valence=trans

verb2_stem1_orth=tverb1

verb2_stem1_pred=_tv_v_rel

verb3_name=verb-type3

verb3_feat1_name=OPT

verb3_feat1_value=minus

verb3_feat1_head=subj

verb3_valence=intrans

verb3_stem1_orth=iverb2

verb3_stem1_pred=iv2_v_rel

verb4_name=verb-type4

verb4_feat1_name=OPT

verb4_feat1_value=minus

verb4_feat1_head=subj

verb4_valence=trans

verb4_stem1_orth=tverb2

verb4_stem1_pred=tv2_v_rel

verb-slot1_name=subj-marker

verb-slot1_order=after

verb-slot1_input1_type=verb

verb-slot1_morph1_name=3fs

verb-slot1_morph1_orth=-subj-3fs

verb-slot1_morph1_feat1_name=number

verb-slot1_morph1_feat1_value=sing

verb-slot1_morph1_feat1_head=subj

verb-slot1_morph1_feat2_name=person

verb-slot1_morph1_feat2_value=3rd

verb-slot1_morph1_feat2_head=subj

verb-slot1_morph1_feat3_name=gender

verb-slot1_morph1_feat3_value=fem

verb-slot1_morph1_feat3_head=subj

section=test-sentences

Test Suite for pseudo-language with lexically-based subject dropping and subject marker that is always required.

iverb1-subj-3fs

n1 iverb1-subj-3fs

n1 iverb2-subj-3fs

n2 iverb1-subj-3fs

n2 iverb2-subj-3fs

n1 tverb1-subj-3fs n2

n1 tverb2-subj-3fs n2

tverb1-subj-3fs n2

∗iverb2-subj-3fs

∗iverb2-subj-3fs n2

∗iverb1-subj-3fs n1

∗iverb2-subj-3fs n1

∗tverb2-subj-3fs n2

∗tverb1-subj-3fs

∗tverb2-subj-3fs

∗n1 tverb1-subj-3fs

∗n2 tverb1-subj-3fs

∗n1 tverb2-subj-3fs

∗n2 tverb2-subj-3fs

∗iverb1

∗n1 iverb1

∗n1 iverb2

∗n2 iverb1

∗n2 iverb2

∗n1 tverb1 n2

∗n1 tverb2 n2

∗tverb1 n2

∗iverb2

∗iverb2 n2

∗iverb1 n1

∗iverb2 n1

∗tverb2 n2

∗tverb1

∗tverb2

∗n1 tverb1

∗n2 tverb1

∗n1 tverb2

∗n2 tverb2

Arabic Test Suite

#Ex 1

Source: author

Vetted: f

Judgment: g

Phenomena: pro-d

naamat fatimatu

naama-t fatima-u

sleep-3sf.past fatima-nominative

Fatima slept

#Ex 2

Source: author

Vetted: f

Judgment: u

Phenomena: pro-d

naama fatimatu

naama fatima-u

sleep.3ms.past fatima-nominative

Fatima slept

#Ex 3

Source: author

Vetted: f

Judgment: g

Phenomena: pro-d

naamat

naama-t

sleep-3fs.past

She slept

#Ex 4

Source: author

Vetted: f

Judgment: g

Phenomena: pro-d

ishtarat fatimatu kitaabaan

ishtaraa-t fatima-u kitaab-an

buy-3fs.past fatima-nom book-acc.indef

Fatima bought a book

#Ex5

Source: author

Vetted: f

Judgment: g

Phenomena: pro-d

ishtarathu

ishtaraa-t-hu

buy-3fs.subj.past-3ms.obj

She bought it

#Ex 6

Source: author

Vetted: f

Judgment: u

Phenomena: pro-d

ishtaraa fatimatu

ishtaraa fatima-u

buy-3ms.past fatima-nom

Fatima bought

Ex 7

Source: author

Vetted: f

Judgment: u

Phenomena: pro-d

ishtarat

ishtaraa-t

buy-3fs.past.subj

She bought

#Ex 8

Source: author

Vetted: f

Judgment: g

Phenomena: pro-d

ishtarathu fatimatu

ishtaraa-t-hu fatima-u

buy-3fs.subj-3ms.obj fatima-nom

Fatima bought it

#Ex 9

Source: author

Vetted: f

Judgment: g

Phenomena: pro-d

ishtarat kitaabaan

ishtaraa-t kitaab-an

buy-3fs.past book-acc.indef

She bought a book

#Ex 10

Source: author

Vetted: f

Judgment: g

Phenomena: pro-d

ishtaraahu

ishtaraa-hu

buy.3ms.subj.past-3ms.obj

He bought it

#Ex 11

Source: author

Vetted: f

Judgment: g

Phenomena: pro-d

ishtaraa al-kitaaba

ishtaraa al-kitaab-a

buy-3ms.past definite-book-acc

He bought the book

#Ex 12

Source: author

Vetted: f

Judgment: g

Phenomena: pro-d

darasat kitaabaan

darasa-t kitaab-an

study-3fs.past book-indef.ac

She studied a book

#Ex 13

Source: author

Vetted: f

Judgment: g

Phenomena: pro-d

darasat

darasa-t

study-3fs.past

She studied

Arabic Choices File

version=20

section=general

language=Arabic

archive=no

section=word-order

word-order=free

has-dets=no

has-aux=no

section=number

number1_name=singular

number2_name=dual

number3_name=plural

section=person

person=1-2-3

first-person=none

section=gender

gender1_name=feminine

gender2_name=masculine

section=case
case-marking=nom-acc
nom-acc-nom-case-name=nominative
nom-acc-acc-case-name=accusative

section=direct-inverse

section=tense-aspect
tense-definition=choose
past=on
present=on
future=on

section=other-features

section=sentential-negation

section=coordination

section=matrix-yes-no

section=arg-opt
subj-drop=subj-drop-all
subj-mark-drop=subj-mark-drop-req
subj-mark-no-drop=subj-mark-no-drop-req
subj-con=subj-con-always

obj-drop=obj-drop-all

obj-mark-drop=obj-mark-drop-req

obj-mark-no-drop=obj-mark-no-drop-opt

section=lexicon

noun1_name=proper-fem-noun-lex

noun1_feat1_name=gender

noun1_feat1_value=feminine

noun1_det=imp

noun1_stem1_orth=fatima

noun1_stem1_pred=_name_n_rel

noun2_name=masc-noun-lex

noun2_feat1_name=gender

noun2_feat1_value=masculine

noun2_det=imp

noun2_stem1_orth=kitaab

noun2_stem1_pred=_book_n_rel

noun3_name=3rd_masc

noun3_feat1_name=person

noun3_feat1_value=3rd

noun3_feat2_name=gender

noun3_feat2_value=masculine

noun3_feat3_name=number

noun3_feat3_value=singular

noun3_feat4_name=case

noun3_feat4_value=nom

noun3_det=imp

noun3_stem1_orth=huwa

noun3_stem1_pred=_pronoun_n_rel

noun-slot1_name=indefinite

noun-slot1_order=after

noun-slot1_input1_type=noun2

noun-slot1_morph1_name=acc-indef-marker

noun-slot1_morph1_orth=-an

noun-slot1_morph1_feat1_name=case

noun-slot1_morph1_feat1_value=acc

noun-slot1_morph2_name=nom-indef-marker

noun-slot1_morph2_orth=-un

noun-slot1_morph2_feat1_name=case

noun-slot1_morph2_feat1_value=nom

noun-slot2_name=definite

noun-slot2_order=before

noun-slot2_input1_type=noun2

noun-slot2_morph1_name=def-marker

noun-slot2_morph1_orth=al-

noun-slot3_name=def-case

noun-slot3_order=after

noun-slot3_input1_type=noun-slot2

noun-slot3_input2_type=noun1

noun-slot3_morph1_name=nom-def-case-marker

noun-slot3_morph1_orth=-u

noun-slot3_morph1_feat1_name=case

noun-slot3_morph1_feat1_value=nom

noun-slot3_morph2_name=acc-def-case-marker

noun-slot3_morph2_orth=-a

verb1_name=verb-type1

verb1_valence=nom-acc

verb1_stem1_orth=ishtaraa

verb1_stem1_pred=_buy_v_rel

verb2_name=verb-type2

verb2_valence=nom-acc

verb2_stem1_orth=darasa

verb2_stem1_pred=_study_v_rel

verb3_name=intran

verb3_valence=nom

verb3_stem1_orth=naama

verb3_stem1_pred=_sleep_v_rel

verb-slot1_name=subj-marker

verb-slot1_order=after

verb-slot1_input1_type=tverb

verb-slot1_morph1_name=3fs-subj-marker

verb-slot1_morph1_orth=-t

verb-slot1_morph1_feat1_name=number

verb-slot1_morph1_feat1_value=singular

verb-slot1_morph1_feat1_head=subj

verb-slot1_morph1_feat2_name=person

verb-slot1_morph1_feat2_value=3rd

verb-slot1_morph1_feat2_head=subj

verb-slot1_morph1_feat3_name=gender

verb-slot1_morph1_feat3_value=feminine

verb-slot1_morph1_feat3_head=subj

verb-slot1_morph1_feat4_name=tense

verb-slot1_morph1_feat4_value=past

verb-slot1_morph1_feat4_head=verb

verb-slot1_morph2_name=3ms-subj-marker

verb-slot1_morph2_feat1_name=person

verb-slot1_morph2_feat1_value=3rd

verb-slot1_morph2_feat1_head=subj

verb-slot1_morph2_feat2_name=gender

verb-slot1_morph2_feat2_value=masculine

verb-slot1_morph2_feat2_head=subj

verb-slot1_morph2_feat3_name=number

verb-slot1_morph2_feat3_value=singular

verb-slot1_morph2_feat3_head=subj

verb-slot1_morph2_feat4_name=tense

verb-slot1_morph2_feat4_value=past

verb-slot1_morph2_feat4_head=verb

verb-slot2_name=obj-marker

verb-slot2_order=after

verb-slot2_input1_type=verb-slot1

verb-slot2_morph1_name=3ms-obj-marker

verb-slot2_morph1_orth=-hu

verb-slot2_morph1_feat1_name=number

verb-slot2_morph1_feat1_value=singular

verb-slot2_morph1_feat1_head=obj

verb-slot2_morph1_feat2_name=person

verb-slot2_morph1_feat2_value=3rd

verb-slot2_morph1_feat2_head=obj

verb-slot2_morph1_feat3_name=gender

verb-slot2_morph1_feat3_value=masculine

verb-slot2_morph1_feat3_head=obj

verb-slot2_morph1_feat4_name=overt-arg

verb-slot2_morph1_feat4_value=permitted

verb-slot2_morph1_feat4_head=obj

verb-slot2_morph2_name=3fs-obj-marker

verb-slot2_morph2_orth=-haa

verb-slot2_morph2_feat1_name=number

verb-slot2_morph2_feat1_value=singular

verb-slot2_morph2_feat1_head=obj

verb-slot2_morph2_feat2_name=person

verb-slot2_morph2_feat2_value=3rd

verb-slot2_morph2_feat2_head=obj

verb-slot2_morph2_feat3_name=gender

verb-slot2_morph2_feat3_value=feminine

verb-slot2_morph2_feat3_head=obj

verb-slot2_morph2_feat4_name=overt-arg

verb-slot2_morph2_feat4_value=permitted

verb-slot2_morph2_feat4_head=obj

verb-slot3_name=intran-subj-marker

verb-slot3_order=after

verb-slot3_input1_type=iverb

verb-slot3_morph1_name=3fs-intran-subj-marker

verb-slot3_morph1_orth=-t

verb-slot3_morph1_feat1_name=number

verb-slot3_morph1_feat1_value=singular

verb-slot3_morph1_feat1_head=subj

verb-slot3_morph1_feat2_name=person

verb-slot3_morph1_feat2_value=3rd

verb-slot3_morph1_feat2_head=subj

verb-slot3_morph1_feat3_name=gender

verb-slot3_morph1_feat3_value=feminine

verb-slot3_morph1_feat3_head=subj

verb-slot3_morph1_feat4_name=tense

verb-slot3_morph1_feat4_value=past

verb-slot3_morph1_feat4_head=verb

section=test-sentences

Nkore-Kiga Test Suite

#Ex 1 demonstrates SVO word order. The basic word order.
Source: a:91
Vetted: f
Judgment: g
Phenomena: word order)
omuntu akwata enkoni
o-mu-ntu a-kwata e-n-koni
c1.vwl-c1.prfx-person c1.3sg.subj-hold c9.vwl-c9.prfx-stick
Someone is holding a stick.

#Ex 2 demonstrates 0SV word order. If the object is fronted, it must be marked on the verb.
Source: a:91
Vetted: f
Judgment: g
Phenomena : word order, agr
enkoni omuntu agikwata
e-n-koni o-mu-ntu a-gi-kwata
c9.vwl-c9.prfx-stick c1.vwl-c1.prfx-person c1.3sg.subj-c9.obj-hold
Someone is holding a stick.

#Ex 3 demonstrates SOV word order. If the object precedes the verb, it must be marked on the verb.
Source: author

Vetted: f
Judgment: g
Phenomena: word order
omuntu enkoni agikwata
o-mu-ntu e-n-koni a-gi-kwata
c1.vwl-c1.prfx-person c9.vwl-c9.prfx-stick c1.3sg.subj-c9.obj-hold
Someone is holding a stick.

#Ex 4 demonstrates ungrammtical SOV word order. If the object precedes the verb, it must be marked on the verb. Here it does not.
Source: author
Vetted: f
Judgment: u
Phenomena: word order
omuntu enkoni akwata
o-mu-ntu e-n-koni a-kwata
c1.vwl-c1.prfx-person c9.vwl-c9.prfx-stick c1.3sg.subj-hold
Someone is holding a stick.

#Ex 5 demonstrates ungrammtical 0SV word order. If the object is fronted, it must be marked on the verb.
Source: author
Vetted: f
Judgment: u
Phenomena : word order
enkoni omuntu akwata

e-n-koni o-mu-ntu a-kwata

c9.vwl-c9.prfx-stick c1.vwl-c1.prfx-person c1.3sg.subj-hold

Someone is holding a stick.

#Ex 6 demonstrates ungrammatical VSO word order.

Source: a:91

Vetted: f

Judgment: u

Phenomena: word order

akwata omuntu enkoni

a-kwata o-mu-ntu e-n-koni

c1.3sg.subj-hold c1.vwl-c1.prfx-person c9.vwl-c9.prfx-stick

Someone is holding a stick.

#Ex 7 demonstrates ungrammatical VOS word order.

Source: author

Vetted: f

Judgment: u

Phenomena: word order

akwata enkoni omuntu

a-kwata e-n-koni o-mu-ntu

c1.3sg.subj-hold c9.vwl-c9.prfx-stick c1.vwl-c1.prfx-person

Someone is holding a stick.

#Ex 8 demonstrates OVS word order.

Source: author

Vetted: f
Judgment: u
Phenomena: word order
enkoni agikwata omuntu
e-n-koni a-gi-kwata o-mu-ntu
c9.vwl-c9.prfx-stick c1.3sg.subj-c9.obj-hold c1.vwl-c1.prfx-person
Someone is holding a stick.

#Ex 9 demonstrates ungrammatical subj-verb agreement. A subject prefix that agrees with the class of the subject must attach to the verb stem. Here it is missing
Source: author
Vetted: f
Judgment: u
Phenomena: agr, pro-d
omuntu kwata enkoni
o-mu-ntu kwata e-n-koni
c1.vwl-c1.prfx-person hold c9.vwl-c9.prfx-stick
Someone is holding a stick.

#Ex 10 demonstrates grammatical first person singular subject-verb agreement. Free pronouns are only used for emphasis. Here only the bound subject prefix occurs.
Source: author
Vetted: f
Judgment: g
Phenomena: agr, pro-d
nyeshongora

n-eshongora

c1.1sg.subj-sing

I sing.

Nkore-Kiga Choices File

version=18

section=general
language=NkoreKiga
iso-code=nyn
archive=no

section=word-order
word-order=free
has-dets=no
has-aux=yes
aux-comp-order=before
aux-comp=v
v-cluster=yes

section=number
number1_name=singular
number2_name=plural

section=person
person=1-2-3
first-person=none

section=gender

gender1_name=class-1

gender2_name=class-5

gender3_name=class-9

gender4_name=class-2

section=case

case-marking=none

section=direct-inverse

section=tense-aspect

tense-definition=build

tense1_name=present

tense1_supertype1_name=tense

tense2_name=past

tense2_supertype1_name=tense

tense3_name=future

tense3_supertype1_name=tense

tense4_name=remote-past

tense4_supertype1_name=past

tense5_name=yesterday-past

tense5_supertype1_name=past

tense6_name=today-past

tense6_supertype1_name=past

aspect1_name=continuous

aspect1_supertype1_name=aspect

aspect2_name=habitual

aspect2_supertype1_name=aspect

aspect3_name=perfective

aspect3_supertype1_name=aspect

aspect4_name=perfective-continuous

aspect4_supertype1_name=perfective

aspect5_name=perfective-habitual

aspect5_supertype1_name=perfective

nf-subform1_name=non-modified

nf-subform2_name=modified

fin-subform1_name=negated

fin-subform2_name=non-negated

section=other-features

feature1_name=AUX2

feature1_type=head

feature1_value1_name=plus2

feature1_value1_supertype1_name=AUX2

feature1_value2_name=minus2

feature1_value2_supertype1_name=AUX2

feature2_name=PN

feature2_type=head

feature2_value1_name=plus2

feature2_value1_supertype1_name=PN

feature2_value2_name=minus2

feature2_value2_supertype1_name=PN

section=sentential-negation

infl-neg=on

section=coordination

cs1_n=on

cs1_pat=mono

cs1_mark=affix

cs1_orth=na-

cs1_order=before

cs2_vp=on

cs2_s=on

cs2_pat=mono

cs2_mark=word

cs2_orth=kandi

cs2_order=before

section=matrix-yes-no

q-part=on

q-part-order=after

q-part-orth=voice

section=arg-opt

subj-drop=subj-drop-all

subj-mark-drop=subj-mark-drop-req

subj-mark-no-drop=subj-mark-no-drop-req

subj-con=subj-con-always
obj-drop=obj-drop-all
obj-mark-drop=obj-mark-drop-req
obj-mark-no-drop=obj-mark-no-drop-not

section=lexicon
noun1_name=class1-2
noun1_feat1_name=gender
noun1_feat1_value=class-1, class-2
noun1_det=imp
noun1_stem1_orth=ntu
noun1_stem1_pred=_person_n_rel
noun1_stem2_orth=shaija
noun1_stem2_pred=_man_n_rel
noun1_stem3_orth=kazi
noun1_stem3_pred=_woman_n_rel
noun2_name=class5
noun2_feat1_name=gender
noun2_feat1_value=class-5
noun2_det=imp
noun2_stem1_orth=zooba
noun2_stem1_pred=_sun_n_rel
noun3_name=class9
noun3_feat1_name=gender
noun3_feat1_value=class-9
noun3_det=imp

noun3_stem1_orth=koni

noun3_stem1_pred=_stick_n_rel

noun4_name=class-9_pronoun

noun4_det=imp

noun4_stem1_orth=yo

noun4_stem1_pred=_pron_n_rel

noun-slot1_name=initial-noun-prefix

noun-slot1_order=before

noun-slot1_input1_type=noun1

noun-slot1_input2_type=noun2

noun-slot1_input3_type=noun3

noun-slot1_morph1_name=class1-intitial-noun

noun-slot1_morph1_orth=mu-

noun-slot1_morph1_feat1_name=gender

noun-slot1_morph1_feat1_value=class-1

noun-slot1_morph1_feat2_name=number

noun-slot1_morph1_feat2_value=singular

noun-slot1_morph2_name=class2-initial-noun

noun-slot1_morph2_orth=ba-

noun-slot1_morph2_feat1_name=gender

noun-slot1_morph2_feat1_value=class-2

noun-slot1_morph2_feat2_name=number

noun-slot1_morph2_feat2_value=plural

noun-slot1_morph3_name=class5-initial-noun

noun-slot1_morph3_orth=ri-

noun-slot1_morph3_feat1_name=gender

noun-slot1_morph3_feat1_value=class-5

noun-slot1_morph4_name=class9-initial-noun

noun-slot1_morph4_orth=n-

noun-slot1_morph4_feat1_name=gender

noun-slot1_morph4_feat1_value=class-9

noun-slot2_name=initial-vowel-prefix

noun-slot2_order=before

noun-slot2_input1_type=noun-slot1

noun-slot2_morph1_name=class1-initial-vowel

noun-slot2_morph1_orth=o-

noun-slot2_morph1_feat1_name=gender

noun-slot2_morph1_feat1_value=class-1

noun-slot2_morph2_name=class2-initial-vowel

noun-slot2_morph2_orth=a-

noun-slot2_morph2_feat1_name=gender

noun-slot2_morph2_feat1_value=class-2

noun-slot2_morph3_name=class5-initial-vowel

noun-slot2_morph3_orth=e-

noun-slot2_morph3_feat1_name=gender

noun-slot2_morph3_feat1_value=class-5

noun-slot2_morph4_name=class9-initial-vowel

noun-slot2_morph4_orth=e-

noun-slot2_morph4_feat1_name=gender

noun-slot2_morph4_feat1_value=class-9

noun-slot3_name=hack-to-allow-coord-suffix

noun-slot3_order=before

noun-slot3_input1_type=noun-slot2

noun-slot3_morph1_name=hack-for-coord

verb1_name=trans

verb1_feat1_name=AUX2

verb1_feat1_value=minus2

verb1_feat1_head=verb

verb1_valence=trans

verb1_stem1_orth=kwata

verb1_stem1_pred=_hold_v_rel

verb2_name=intrans

verb2_feat1_name=AUX2

verb2_feat1_value=minus2

verb2_feat1_head=verb

verb2_valence=intrans

verb2_stem1_orth=eshongora

verb2_stem1_pred=_sing_v_rel

verb2_stem2_orth=renga

verb2_stem2_pred=_set_v_rel

verb2_stem3_orth=zaana

verb2_stem3_pred=_play_v_rel

aux1_name=perfect

aux1_sem=no-pred

aux1_feat1_name=aspect

aux1_feat1_value=perfective

aux1_subj=np

aux1_compfeature1_name=form

aux1_compfeature1_value=modified

aux1_stem1_orth=ba

verb-slot1_name=present-tense-marker

verb-slot1_order=before

verb-slot1_input1_type=verb-slot6

verb-slot1_morph1_name=present-universal

verb-slot1_morph1_feat1_name=tense

verb-slot1_morph1_feat1_value=present

verb-slot1_morph1_feat1_head=verb

verb-slot1_morph1_feat2_name=aspect

verb-slot1_morph1_feat2_value=habitual

verb-slot1_morph1_feat2_head=verb

verb-slot1_morph1_feat3_name=AUX2

verb-slot1_morph1_feat3_value=minus2

verb-slot1_morph1_feat3_head=verb

verb-slot1_morph1_feat4_name=form

verb-slot1_morph1_feat4_value=non-negated

verb-slot1_morph1_feat4_head=verb

verb-slot1_morph2_name=present-continuous

verb-slot1_morph2_orth=ni-

verb-slot1_morph2_feat1_name=tense

verb-slot1_morph2_feat1_value=present

verb-slot1_morph2_feat1_head=verb

verb-slot1_morph2_feat2_name=aspect

verb-slot1_morph2_feat2_value=continuous

verb-slot1_morph2_feat2_head=verb

verb-slot1_morph2_feat3_name=AUX2

verb-slot1_morph2_feat3_value=minus2

verb-slot1_morph2_feat3_head=verb

verb-slot1_morph2_feat4_name=form

verb-slot1_morph2_feat4_value=non-negated

verb-slot1_morph2_feat4_head=verb

verb-slot2_name=non-present-tense-dummy

verb-slot2_order=before

verb-slot2_input1_type=verb-slot6

verb-slot2_morph1_name=remote-past-tense-dummy

verb-slot2_constraint1_type=req

verb-slot2_constraint1_other-slot=verb-slot3

verb-slot3_name=remote-marker

verb-slot3_order=before

verb-slot3_input1_type=verb

verb-slot3_input2_type=verb-slot5

verb-slot3_morph1_name=remote-past-marker

verb-slot3_morph1_orth=ka-

verb-slot3_morph1_feat1_name=tense

verb-slot3_morph1_feat1_value=remote-past

verb-slot3_morph1_feat1_head=verb

verb-slot4_name=yesterday-past-modified-verb-form

verb-slot4_order=after

verb-slot4_input1_type=verb-slot6

verb-slot4_morph1_name=yesterday-past-marker

verb-slot4_morph1_orth=-ire

verb-slot4_morph1_feat1_name=tense

verb-slot4_morph1_feat1_value=yesterday-past

verb-slot4_morph1_feat1_head=verb

verb-slot4_morph2_name=modified-verb-form-marker

verb-slot4_morph2_orth=-ire

verb-slot4_morph2_feat1_name=form

verb-slot4_morph2_feat1_value=modified

verb-slot4_morph2_feat1_head=verb

verb-slot5_name=object-marker

verb-slot5_order=before

verb-slot5_input1_type=tverb

verb-slot5_morph1_name=class9-obj-marker

verb-slot5_morph1_orth=gi-

verb-slot5_morph1_feat1_name=gender

verb-slot5_morph1_feat1_value=class-9

verb-slot5_morph1_feat1_head=obj

verb-slot5_morph2_name=class1-1s-obj-marker

verb-slot5_morph2_orth=n-

verb-slot5_morph2_feat1_name=person

verb-slot5_morph2_feat1_value=1st

verb-slot5_morph2_feat1_head=obj

verb-slot5_morph2_feat2_name=number

verb-slot5_morph2_feat2_value=singular

verb-slot5_morph2_feat2_head=obj

verb-slot5_morph2_feat3_name=gender

verb-slot5_morph2_feat3_value=class-1

verb-slot5_morph2_feat3_head=obj

verb-slot5_morph3_name=class1-2s-obj-marker

verb-slot5_morph3_orth=ku-

verb-slot5_morph3_feat1_name=number

verb-slot5_morph3_feat1_value=singular

verb-slot5_morph3_feat1_head=obj

verb-slot5_morph3_feat2_name=person

verb-slot5_morph3_feat2_value=2nd

verb-slot5_morph3_feat2_head=obj

verb-slot5_morph3_feat3_name=gender

verb-slot5_morph3_feat3_value=class-1

verb-slot5_morph3_feat3_head=obj

verb-slot5_morph4_name=class1-3s-obj-marker

verb-slot5_morph4_orth=mu-

verb-slot5_morph4_feat1_name=person

verb-slot5_morph4_feat1_value=3rd

verb-slot5_morph4_feat1_head=obj

verb-slot5_morph4_feat2_name=number

verb-slot5_morph4_feat2_value=singular

verb-slot5_morph4_feat2_head=obj

verb-slot5_morph4_feat3_name=gender

verb-slot5_morph4_feat3_value=class-1

verb-slot5_morph4_feat3_head=obj

verb-slot5_morph5_name=class2-1pl-obj-marker

verb-slot5_morph5_orth=tu-

verb-slot5_morph5_feat1_name=number

verb-slot5_morph5_feat1_value=plural

verb-slot5_morph5_feat1_head=obj

verb-slot5_morph5_feat2_name=person

verb-slot5_morph5_feat2_value=1st

verb-slot5_morph5_feat2_head=obj

verb-slot5_morph5_feat3_name=gender

verb-slot5_morph5_feat3_value=class-2

verb-slot5_morph5_feat3_head=obj

verb-slot5_morph6_name=class2-2-3pl-obj-marker

verb-slot5_morph6_orth=ba-

verb-slot5_morph6_feat1_name=number

verb-slot5_morph6_feat1_value=plural

verb-slot5_morph6_feat1_head=obj

verb-slot5_morph6_feat2_name=person

verb-slot5_morph6_feat2_value=2nd, 3rd

verb-slot5_morph6_feat2_head=obj

verb-slot5_morph6_feat3_name=gender

verb-slot5_morph6_feat3_value=class-2

verb-slot5_morph6_feat3_head=obj

verb-slot5_morph7_name=class5-obj-marker

verb-slot5_morph7_orth=ri-

verb-slot5_morph7_feat1_name=number

verb-slot5_morph7_feat1_value=singular

verb-slot5_morph7_feat1_head=obj

verb-slot5_morph7_feat2_name=person

verb-slot5_morph7_feat2_value=3rd

verb-slot5_morph7_feat2_head=obj

verb-slot5_morph7_feat3_name=gender

verb-slot5_morph7_feat3_value=class-5

verb-slot5_morph7_feat3_head=obj

verb-slot6_name=subject-marker

verb-slot6_order=before

verb-slot6_input1_type=verb-slot3

verb-slot6_input2_type=verb-slot9

verb-slot6_input3_type=verb-slot8

verb-slot6_input4_type=verb

verb-slot6_input5_type=verb-slot5

verb-slot6_morph1_name=class1-3s-subj-marker

verb-slot6_morph1_orth=a-

verb-slot6_morph1_feat1_name=gender

verb-slot6_morph1_feat1_value=class-1

verb-slot6_morph1_feat1_head=subj

verb-slot6_morph1_feat2_name=person

verb-slot6_morph1_feat2_value=3rd

verb-slot6_morph1_feat2_head=subj

verb-slot6_morph1_feat3_name=number

verb-slot6_morph1_feat3_value=singular

verb-slot6_morph1_feat3_head=subj

verb-slot6_morph2_name=class1-1s-subj-marker

verb-slot6_morph2_orth=n-

verb-slot6_morph2_feat1_name=number

verb-slot6_morph2_feat1_value=singular

verb-slot6_morph2_feat1_head=subj

verb-slot6_morph2_feat2_name=person

verb-slot6_morph2_feat2_value=1st

verb-slot6_morph2_feat2_head=subj

verb-slot6_morph2_feat3_name=gender

verb-slot6_morph2_feat3_value=class-1

verb-slot6_morph2_feat3_head=subj

verb-slot6_morph3_name=class1-2s-subj-marker

verb-slot6_morph3_orth=o-

verb-slot6_morph3_feat1_name=number

verb-slot6_morph3_feat1_value=singular

verb-slot6_morph3_feat1_head=subj

verb-slot6_morph3_feat2_name=person

verb-slot6_morph3_feat2_value=2nd

verb-slot6_morph3_feat2_head=subj

verb-slot6_morph3_feat3_name=gender

verb-slot6_morph3_feat3_value=class-1

verb-slot6_morph3_feat3_head=subj

verb-slot6_morph4_name=class2-1pl-subj-marker

verb-slot6_morph4_orth=tu-

verb-slot6_morph4_feat1_name=number

verb-slot6_morph4_feat1_value=plural

verb-slot6_morph4_feat1_head=subj

verb-slot6_morph4_feat2_name=person

verb-slot6_morph4_feat2_value=1st

verb-slot6_morph4_feat2_head=subj

verb-slot6_morph4_feat3_name=gender

verb-slot6_morph4_feat3_value=class-2

verb-slot6_morph4_feat3_head=subj

verb-slot6_morph5_name=class2-2pl-subj-marker

verb-slot6_morph5_orth=mu-

verb-slot6_morph5_feat1_name=number

verb-slot6_morph5_feat1_value=plural

verb-slot6_morph5_feat1_head=subj

verb-slot6_morph5_feat2_name=person

verb-slot6_morph5_feat2_value=2nd

verb-slot6_morph5_feat2_head=subj

verb-slot6_morph5_feat3_name=gender

verb-slot6_morph5_feat3_value=class-2

verb-slot6_morph5_feat3_head=subj

verb-slot6_morph6_name=class2-3pl-subj-marker

verb-slot6_morph6_orth=ba-

verb-slot6_morph6_feat1_name=number

verb-slot6_morph6_feat1_value=plural

verb-slot6_morph6_feat1_head=subj

verb-slot6_morph6_feat2_name=person

verb-slot6_morph6_feat2_value=3rd

verb-slot6_morph6_feat2_head=subj

verb-slot6_morph6_feat3_name=gender

verb-slot6_morph6_feat3_value=class-2

verb-slot6_morph6_feat3_head=subj

verb-slot6_morph7_name=class5-subj-marker

verb-slot6_morph7_orth=ri-

verb-slot6_morph7_feat1_name=gender

verb-slot6_morph7_feat1_value=class-5

verb-slot6_morph7_feat1_head=subj

verb-slot6_morph7_feat2_name=number

verb-slot6_morph7_feat2_value=singular

verb-slot6_morph7_feat2_head=subj

verb-slot6_morph7_feat3_name=person

verb-slot6_morph7_feat3_value=3rd

verb-slot6_morph7_feat3_head=subj

verb-slot6_morph8_name=class9-subj-marker

verb-slot6_morph8_orth=e-

verb-slot6_morph8_feat1_name=number

verb-slot6_morph8_feat1_value=singular

verb-slot6_morph8_feat1_head=subj

verb-slot6_morph8_feat2_name=person

verb-slot6_morph8_feat2_value=3rd

verb-slot6_morph8_feat2_head=subj

verb-slot6_morph8_feat3_name=gender

verb-slot6_morph8_feat3_value=class-9

verb-slot6_morph8_feat3_head=subj

verb-slot7_name=neg

verb-slot7_order=before

verb-slot7_input1_type=verb-slot6

verb-slot7_morph1_name=negation

verb-slot7_morph1_orth=ti-

verb-slot7_morph1_feat1_name=negation

verb-slot7_morph1_feat1_value=plus22

verb-slot7_morph1_feat1_head=verb

verb-slot7_constraint1_type=req

verb-slot7_constraint1_other-slot=verb-slot8

verb-slot7_constraint2_type=req

verb-slot7_constraint2_other-slot=verb-slot9

verb-slot8_name=neg-pres-tense

verb-slot8_order=before

verb-slot8_input1_type=verb

verb-slot8_morph1_name=neg-pres

verb-slot8_morph1_feat1_name=tense

verb-slot8_morph1_feat1_value=present

verb-slot8_morph1_feat1_head=verb

verb-slot8_morph1_feat2_name=aspect

verb-slot8_morph1_feat2_value=habitual

verb-slot8_morph1_feat2_head=verb

verb-slot8_morph1_feat3_name=form

verb-slot8_morph1_feat3_value=negated

verb-slot8_morph1_feat3_head=verb

verb-slot8_morph2_name=neg-pres-cont

verb-slot8_morph2_orth=riku-

verb-slot8_morph2_feat1_name=tense

verb-slot8_morph2_feat1_value=present

verb-slot8_morph2_feat1_head=verb

verb-slot8_morph2_feat2_name=aspect

verb-slot8_morph2_feat2_value=continuous

verb-slot8_morph2_feat2_head=verb

verb-slot8_morph2_feat3_name=form

verb-slot8_morph2_feat3_value=negated

verb-slot8_morph2_feat3_head=verb

verb-slot9_name=neg-past-tense

verb-slot9_order=before

verb-slot9_input1_type=verb

verb-slot9_morph1_name=neg-past

verb-slot9_morph1_orth=ra-

verb-slot9_morph1_feat1_name=tense

verb-slot9_morph1_feat1_value=remote-past

verb-slot9_morph1_feat1_head=verb

verb-slot10_name=perfect-present-tense

verb-slot10_order=before

verb-slot10_input1_type=verb-slot6

verb-slot10_morph1_name=perfect-present-continuous

verb-slot10_morph1_orth=ni-

verb-slot10_morph1_feat1_name=tense

verb-slot10_morph1_feat1_value=present

verb-slot10_morph1_feat1_head=verb

verb-slot10_morph1_feat2_name=aspect

verb-slot10_morph1_feat2_value=perfective-continuous

verb-slot10_morph1_feat2_head=verb

verb-slot10_morph1_feat3_name=AUX2

verb-slot10_morph1_feat3_value=plus2

verb-slot10_morph1_feat3_head=verb

verb-slot10_morph2_name=perfect-universal-present

verb-slot10_morph2_feat1_name=tense

verb-slot10_morph2_feat1_value=present

verb-slot10_morph2_feat1_head=verb

verb-slot10_morph2_feat2_name=aspect

verb-slot10_morph2_feat2_value=perfective-habitual

verb-slot10_morph2_feat2_head=verb

verb-slot10_morph2_feat3_name=AUX2

verb-slot10_morph2_feat3_value=plus2

verb-slot10_morph2_feat3_head=verb

verb-slot11_name=neg-remote-past-modifier

verb-slot11_opt=on

verb-slot11_order=after

verb-slot11_input1_type=verb-slot7

verb-slot11_morph1_name=neg-remote-past-modifier-suffix

verb-slot11_morph1_orth=-ire

section=test-sentences