A Dependency Grammar for Amharic

Michael Gasser

School of Informatics and Computing
Indiana University, Bloomington, Indiana USA
gasser@cs.indiana.edu

Abstract

There has been little work on computational grammars for Amharic or other Ethio-Semitic languages and their use for parsing and generation. This paper introduces a grammar for a fragment of Amharic within the Extensible Dependency Grammar (XDG) framework of Debusmann. A language such as Amharic presents special challenges for the design of a dependency grammar because of the complex morphology and agreement constraints. The paper describes how a morphological analyzer for the language can be integrated into the grammar, introduces empty nodes as a solution to the problem of null subjects and objects, and extends the agreement principle of XDG in several ways to handle verb agreement with objects as well as subjects and the constraints governing relative clause verbs. It is shown that XDG's multiple dimensions lend themselves to a new approach to relative clauses in the language. The introduced extensions to XDG are also applicable to other Ethio-Semitic languages.

1. Introduction

Within the Semitic family, a number of languages remain relatively under-resourced, including the second most spoken language in the family, Amharic. Among other gaps in the available resources, there is no computational grammar for even a sizable fragment of the language; consequently analysis of Amharic texts rarely goes beyond morphological analysis, stemming, or part-of-speech tagging.

This paper describes a dependency grammar for a fragment of Amharic syntax. The grammar is based on Extensible Dependency Grammar (XDG), developed by Ralph Debusmann and colleagues (Debusmann et al., 2004; Debusmann, 2007). XDG was selected because of its modular structure, its extensibility, and its simple, declarative format. The paper begins with an overview of XDG and a description of some relative aspects of Amharic morphosyntax. Then we look at the extensions to XDG that were implemented to handle Amharic null subjects and objects, agreement of verbs with subjects and objects, and some of the special properties of relative clauses. Most of these extensions will also apply to other Semitic languages.

2. Extensible Dependency Grammar

As in other dependency grammar frameworks, XDG is lexical; the basic units are words and the directed, labeled dependency relations between them. In the simplest case, an analysis ("model" in XDG terms) of a sentence is a graph consisting of a set of dependency arcs connecting the nodes in the sentence such that each node other than the root node has a head and certain constraints on the dependencies are satisfied. As in some, but not all, other dependency frameworks, XDG permits analyses at multiple strata, known as **dimensions**, each corresponding to some level of grammatical abstraction. For example, one dimension could represent syntax, another semantics. Two dimensions may also be related by an explicit interface dimension that has no arcs itself but constrains how arcs in the related dimensions associate with one another. Debusmann includes a total of six simple dimensions and five interface dimensions in the English grammar discussed in his dissertation. In the general case, then, an analysis of a sentence is a multigraph consisting of a separate dependency graph for each dimension over a single sequence of word nodes. Figure 1 shows a possible analysis for the English sentence *John edited the paper* on two dimensions. The analysis follows the XDG convention of treating the end-of-sentence punctuation as the root of the sentence.

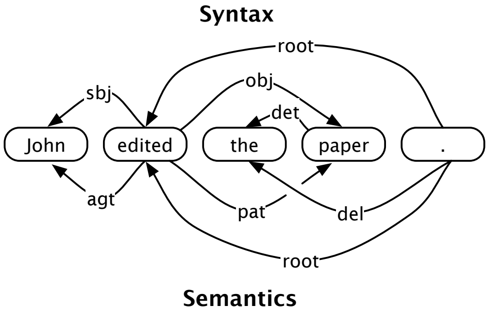

Figure 1: Two-dimensional XDG analysis of an English sentence. Arrows go from head to dependent. Words that do not participate in the semantic dimension are distinguished by delete arcs from the root node.

A grammatical analysis is one that conforms to a set of constraints, each generated by one or another **principle**. Each dimension has its own characteristic set of principles. Some examples:

- Principles concerned with the structure of the graph, for example, it may be constrained to be a tree or a directed acyclic graph.
- The Valency Principle, governing the labels on the arcs into and out of a given node.
- The Agreement Principle, constraining how certain features within some words must match features in other words.

- The Order Principle, concerned with the order of the words in the sentence.

As the framework is completely lexical, it is at the level of words or word classes that the principles apply. For example, the constraint that a finite present-tense verb in English must agree with its subject on the syntactic dimension could appear in the lexicon in this form:[1]

```
- gram: V_FIN_PRES
  syn:
    agree: [sbj]
```

The lexicon is organized in an inheritance hierarchy, with lexical entries inheriting attributes from their ancestor classes. For example, the verb *eats* would inherit the subject-verb agreement constraint from the V_FIN_PRES class.

Parsing and generation within the XDG framework take the form of constraint satisfaction. Given an input sentence to be parsed, lexicalization of the words invokes the principles that are referenced in the lexical entries for the words (or inherited from their ancestors in the lexical hierarchy). Each of these principle invocations results in the instantiation of one or more constraints, each applying to a set of variables. For example, a variable is associated with the label on the arc between two given nodes, and the domain for that variable is the set of possible arc labels that can appear on the arc. Among the constraints that apply to such a variable are those that are created by the Valency Principle. For example, for English transitive verbs, there is a valency constraint that requires that exactly one of the arcs leaving the verb must have an `obj` label. Constraint satisfaction returns all possible combinations of variable bindings, each corresponding to a single analysis of the input sentence.

The XDG framework has been applied to a number of languages, including a small fragment of Arabic (Odeh, 2004), but no one has yet addressed the complexities of morphosyntax that arise with Semitic languages. This paper represents a first effort.

3. Relevant Amharic Morphosyntax

3.1. Verb morphology

As in other Semitic languages, Amharic verbs are very complex (see Leslau (1995) for an overview), consisting of a stem and up to four prefixes and four suffixes. The stem in turn is composed of a root, representing the purely lexical component of the verb, and a template, consisting of slots for the root segments and for the vowels (and sometimes consonants) that are inserted around and between these segments. The template represents tense, aspect, mood, and one of a small set of derivational categories: passive-reflexive, transitive, causative, iterative, reciprocal, and causative reciprocal. For the purposes of this paper, we will consider the combination of root and derivational category to constitute the verb lexeme.

Each lexeme can appear in four different tense-aspect-mood (TAM) categories, conventionally referred to as perfect(ive), imperfect(ive), jussive/imperative, and gerund(ive). We represent verb lexemes in the lexicon in terms of the conventional citation form, the third person singular masculine perfective. For example, the verb *aywededm*[2] 'he is not liked' has the lemma *tewedede* 'he was liked', which is derived from the verb root *w.d.d*.

Every Amharic verb must agree with its subject. As in other Semitic languages, subject agreement is expressed by suffixes alone in some TAM categories (perfective and gerundive) and by a combination of prefixes and suffixes in other TAM categories (imperfective and jussive/imperative). Amharic is a null subject language; that is, a sentence does not require an explicit subject, and personal pronouns appear as subjects only when they are being emphasized for one reason or another.

An Amharic verb may also have a suffix representing the person, number, and gender of a direct object or an indirect object that is definite.[3] The corresponding suffixes in other Semitic languages are often considered to be clitics or even pronouns, but there are good reasons not to do so for Amharic. First, one or two other suffixes may follow the object suffix. Second, as with subjects, object personal pronouns may also appear but only when they are being emphasized. Thus we will consider Amharic to have optional object agreement as well as obligatory subject agreement and to be a null object as well as a null subject language.

3.2. Noun phrases

Amharic nouns without modifiers take suffixes indicating definiteness and accusative case for direct objects and prefixes representing prepositions:

hakim
doctor
'a doctor' (1)

hakimu
doctor-DEF
'the doctor' (2)

hakimun
doctor-DEF-ACC
'the doctor (as object of a verb)' (3)

lehakimu
to-doctor-DEF
'to the doctor' (4)

However, when a noun is modified by one or more adjectives or relative clauses, it is the first modifier that takes

[1] We use YAML syntax (`http://www.yaml.org/`) for lexical entries.

[2] Amharic is written using the Ge'ez script. While there is no single agreed-on standard for romanizing the language, the SERA transcription system, which represents Ge'ez graphemes using ASCII characters (Firdyiwek and Yaqob, 1997), is common in computational work on Amharic and is used in this paper. This transcription system represents the orthography directly, failing to indicate phonological features that the orthography does not encode, in particular, consonant gemination and the presence of the epenthetic vowel that breaks up consonant clusters.

[3] In the interest of simplification, indirect objects will be mostly ignored in this paper. Most of what will be said about direct objects also applies to indirect objects.

these affixes (Kramer, 2009). If a noun takes a determiner, the noun phrase needs no other indication of definiteness, but it is the determiner that takes the accusative suffix or prepositional prefix.

senefu	*hakim*
lazy-DEF	doctor

‘the lazy doctor’ (5)

lesenefu	*hakim*
to-lazy-DEF	doctor

‘to the lazy doctor’ (6)

yann	*senef*	*hakim*
that-ACC	lazy	doctor

‘that lazy doctor (as object of a verb)’ (7)

3.3. Relative clauses

Relative clauses in Amharic consist of a relative verb and zero or more arguments and modifiers of the verb, as in any clause. A relative verb is a verb in either the imperfective or perfective TAM with a prefix indicating relativization. As with a main clause verb, a relative verb must agree with its subject and may agree with its direct object if it has one. Both subjects and objects can be relativized.

yemiwedat	*sEt*
REL-he-likes-her	woman

‘the woman that he likes’ (8)

yemiwedat	*wend*
REL-he-likes-her	man

‘the man who likes her’ (9)

As noted above, when a noun is modified by a relative clause and has no preceding determiner, it is the relative clause that takes suffixes indicating definiteness or accusative case or prepositional prefixes.

yetemereqew	*lj*	*wendmE*	*new*
REL-he-graduated-DEF	boy	my-brother	is

‘The boy who graduated is my brother.’ (10)

yetemereqewn	*lj*	*alawqm*
REL-he-graduated-DEF-ACC	boy	I-don’t-know

‘I don’t know the boy who graduated.’ (11)

When a sequence of modifiers precedes a noun, it is the first one that takes the suffixes or prefixes.[4]

yetemereqew	*gWebez*	*lj*
REL-he-graduated-DEF	clever	boy

‘the clever boy who graduated’ (12)

Because the first modifier of a noun determines the syntactic role of the noun phrase in the clause as well as its definiteness, we will treat this modifier, rather than the noun, as the syntactic head of the noun phrase. There are at least two other reasons for doing this.

- The head noun of a noun phrase with an adjective or relative clause modifier is optional.

tlqun	*’merTalehu*
big-DEF-ACC	I-choose

  ‘I choose the big one.’ (13)

yemiwedat	*alderesem*
REL-he-likes-her	he-didn’t-arrive

  ‘(He) who likes her didn’t arrive. (14)

  Headless relative clauses are found in many languages, for example, in the English translation of sentence (14). What makes Amharic somewhat unusual is that headless relative clauses and adjectives functioning as noun phrases can be formed by simply dropping the noun.

- Relative verbs agree with the main clause verbs that contain them. For example, in example (14) above, the third person singular masculine subject in the main clause verb agrees with the third person singular masculine subject of the relative clause verb.

Therefore we interpret relative clause modifiers as syntactic heads of Amharic nouns. Because XDG offers the possibility of one or more dimensions for semantics as well as syntax, it is straightforward to make the noun the semantic head, much as auxiliary verbs function as syntactic heads while the main verbs they accompany function as semantic heads in Debusmann’s XDG grammar of English. This is discussed further below.

4. XDG for Amharic

In its current incomplete version, our Amharic grammar has a single layer for syntax and a single layer for semantics. The Syntax dimension handles word order, agreement, and syntactic valency.[5] The Semantics dimensions handles semantic valency.

Because the grammar still does not cover some relatively common structures such as cleft sentences and complement clauses, the parser has not yet been evaluated on corpus data.

4.1. Incorporating morphology

For a language like Amharic, it is impractical to list all wordforms in the lexicon; a verb lexeme can appear in more than 100,000 wordforms. Instead we treat the lexeme/lemma as the basic unit; for nouns this is their stem.[6]

[4] With two adjectives, both may optionally take the affixes (Kramer, 2009). We consider this to fall within the realm of coordination, which is not handled in the current version of the grammar described in this paper.

[5] Amharic word order is considerably simpler than that of a language such as English or German, and there are none of the problems of long-distance dependences in questions and relative clauses that we find in those languages. The only non-projective structures are those in cleft sentences and sentences with right dislocation, neither of which is handled in the current version of our grammar. In a later version, we will separate a projective linear precedence layer from a non-projective immediate dominance layer, as Debusmann does for English and German (2007).

[6] Unlike in most other Semitic languages, most Amharic nouns do not lend themselves to an analysis as template+root.

For verbs, as noted above, this is the root plus any derivational morphemes.

In parsing a sentence, we first run a morphological parser over each of the input words. We use the HornMorpho Amharic parser available at `http://www.cs.indiana.edu/~gasser/Research/software.html` and described in Gasser (2009). Given an Amharic word, this parser returns the root (for verbs only), the lemma, and a grammatical analysis in the form of a feature structure description (Carpenter, 1992; Copestake, 2002) for each possible analysis. For example, for the verb *ywedatal* 'he likes her', it returns the following (excluding features that are not relevant for this discussion):

```
'wedede', {'tam': 'impf',
           'rel': False,
           'sb': [-p1,-p2,-plr,-fem],
           'ob': [-p1,-p2,-plr,+fem]}
```

That is, it indicates that this a non-relative verb whose lemma is 'wedede' in imperfective TAM with a third person singular masculine subject and a third person singular feminine object.

It is this sequence of lemma-structure tuples rather than raw wordforms that is the input to the usual XDG lexicalization process that initiates parsing. We have not yet implemented generation, but the reverse process will occur there; that is, the output of constraint satisfaction will be a sequence of lemma-structure tuples which will then be passed to a morphological generator (also available in HornMorpho).

4.2. Null subjects and objects

XDG is grounded in the words occurring in a sentence, but it has to come to grips with the mismatch between nodes in different dimensions. For example, we probably do not want a strictly grammatical word such as *the* to correspond to anything at all on the semantic dimension. Debusmann handles the *deletion* of surface nodes using `del` arcs from the sentence root; this can be seen in the semantic dimension in Figure 1.

Now consider the reverse problem, that of nodes in some dimension that correspond to nothing on the surface. Null subjects and objects in a language such as Amharic present such a problem. They correspond to arguments that need to be explicit at the semantic level but are not present in the input to parsing. We are also working on a synchronous version of XDG with dimensions representing syntactic analyses in different languages. For a language pair such as Amharic-English, with Amharic as the input language, the nodes corresponding to English subject and object pronouns will have to come from somewhere. The problem of "vertex expansion" has come up in the XDG literature (Pelizzoni and Nunes, 2005), but we are unaware of any implementations.

We solve the problem by introducing "empty nodes" in the syntactic dimension. Each verb creates an empty node for its subject, and each transitive verb creates an additional one for its object. The nodes are used only when no explicit argument fills their role. We introduce a new XDG principle to handle these cases, the Empty Node Principle. When a word invoking this principle is found during lexicalization, a constraint is created that sanctions an arc from the verb with the relevant label (`sbj` or `obj`) to either an explicit word or the associated empty node, but not both. Figure 4.3. shows the analysis returned by our parser for the following sentence.[7]

yoHans *ywedatal*
Yohannis he-likes-her
'Yohannis likes her.' (15)

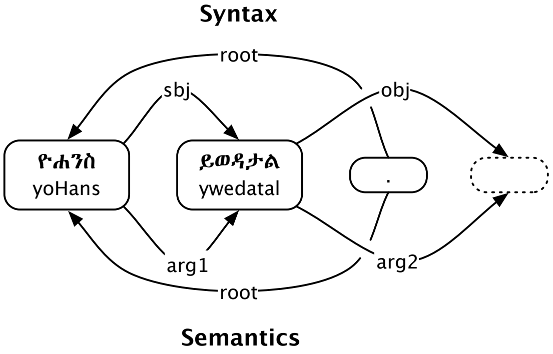

Figure 2: Empty nodes in Amharic. The transitive verb *ywedatal* 'he likes her' has no explicit object, so it is linked to an empty node by an `obj` arc in the Syntax dimension.

Note that our empty nodes are similar to the hidden nodes used in annotation for the Quranic Dependency Treebank project (Dukes et al., 2010).

4.3. Subject and object agreement

In the XDG grammars described by Debusmann and other researchers within the framework, agreement applies to two separate verb attributes. The `agrs` attribute is a list of possible features for the verb form, while the `agree` attribute is a list of arc labels for daughters that must agree with the verb. For example, the following could be part of the entry for the English very `eats`, representing the fact that this word has a single possibility for its agreement feature (third person singular) and the constraint that its subject must also be third person singular.

```
- word: eats
  syn:
    agrs: [3ps]
    agree: [sbj]
```

This limited approach to agreement fails to address the complexity of a language such as Amharic. First, the `agrs` attribute must distinguish subject, direct object, and indirect object features. Second, the `agree` attribute must specify which agreement feature of the mother verb agrees with the daughter on the specified arc. Third, the `agree` attribute must also allow for agreement with different features of the daughter when the daughter is a verb itself, that is, when it is the verb of a relative clause. Consider the entry for transitive verbs (actually a combination of several entries):

[7] In the Amharic dependency graphs in the figures we show the original Ge'ez forms that are the actual input to the parser as well as the transcribed forms.

```
- gram: V_T
  syn:
    agree: {sbj: [sbj, [^,sbj,obj,iobj]],
            obj: [obj, [^,sbj,obj,iobj]]}
```

This specifies that a transitive Amharic verb agrees with the words on both its outbound `sbj` and `obj` arcs, that the subject agrees with the `sbj` feature of the verb and the object agrees with the `obj` feature of the verb, and that the agreement feature of the daughter (subject or object) is either the whole word (denoted by ^) or, in the case of a relative verb, its `sbj`, `obj` or `iobj` feature.

The following sentence is an example of a transitive verb whose subject and object features agree with nouns. The output of the parser on the Syntax dimension for this sentence is shown in Figure 3.

astEr	*yoHansn*	*twedewalec*
Aster	Yohannis-ACC	she-likes-him

‘Aster likes Yohannis.’ (16)

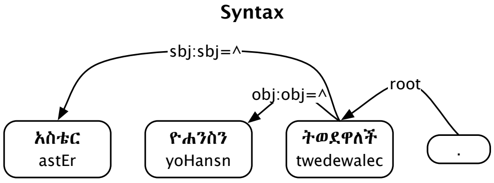

Figure 3: Simple subject-verb and object-verb agreement in Amharic. In addition to their arc labels, two arcs show mother and daughter features that agree. In these cases, the arc label precedes the colon, and the mother and daughter features are separated by “=”.

Note that the verb agreement feature and the arc label need not be the same. For example, for an important subclass of Amharic verbs, the object suffix of the verb agrees with a syntactic argument that we will call the “topic”, which does not take the accusative marker and is not the syntactic subject. In the following example, the verb’s object suffix is third person singular feminine, agreeing with the nominative topic *astEr*.

astEr	*dekmWatal*
Aster	it-has-tired-her

‘Aster is tired.’ (17)

The verb in this sentence, *dekeme* ‘tire’, has the following in its entry:

```
- lexeme: dekeme
  syn:
    agree: {top: [obj, [^,sbj,obj,iobj]]}
```

Figure 4 shows the parser’s analysis of sentence (17).

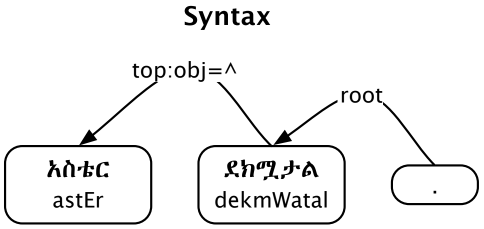

Figure 4: Agreement of a topic with a verb’s object suffix.

4.4. Relative clauses

As argued above, relative verbs are best treated as the heads of their noun phrases. When a relative verb has a head noun, the verb’s subject, object, or indirect object feature must agree with that noun, depending on the role it plays in the verb’s argument structure. In our grammar, we join the relative verb to its head noun in the Syntax dimension by an arc with a label specifying this role, that is, `sbj`, `obj`, or `iobj`. Since verbs are already constrained to agree with their arguments, the agreement between the relative verb and the noun it modifies does not need to be stated separately in the grammar. For illustration, however, we show what this constraint would look like in the entry for object relative verbs.

```
- gram: V_REL_OBJ
  syn:
    agree: {obj: [obj, ^]}
```

Sentence (18) is an example of a sentence with an object relative clause. The analysis of the sentence by our system on the Syntax dimension is shown in Figure 5. The object feature of the relative verb *yemtTelaw* ‘that she hates him’ agrees with the modified noun *wendlj* ‘boy’; both are third person singular masculine. Two other agreement constraints are also satisfied in this sentence. The subject feature of the main verb *tameme* ‘he-got-sick’ agrees with the object feature of the relative verb; both are third person singular masculine. The subject feature of the relative verb agrees with its subject *astEr*; both are third person singular feminine.

astEr	*yemtTelaw*	*wendlj*	*tameme*
Aster	REL-she-hates-him	boy	he-got-sick

‘The boy that Aster hates got sick.’ (18)

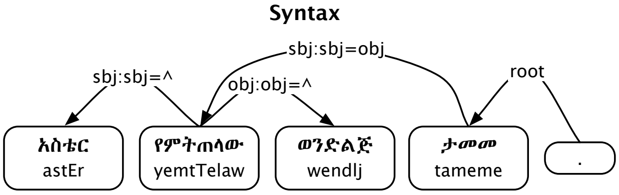

Figure 5: Syntactic analysis of a sentence with a relative clause.

We model the semantics of a sentence with a relative clause as a directed acyclic graph in which the shared noun has multiple verb heads. The relative clause predicate is distinguished from the main clause predicate by a `rel` rather than a `root` arc into it from the sentence root. Figure 6 shows the analysis of sentence (18) on the Semantics dimension.

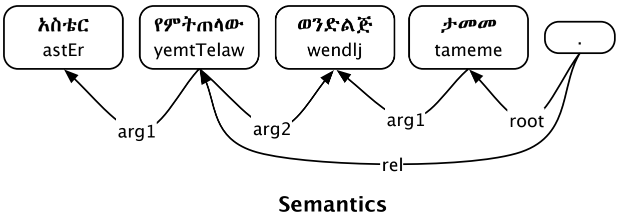

Figure 6: Semantic analysis of a sentence with a relative clause.

Relative clauses without nouns have no overt form corresponding to the shared semantic argument, so we introduce this argument as an empty node. Sentence (19) is sentence (18) with the noun *wendlj* 'boy' dropped. The analyis of this sentence is shown in Figure 7.

astEr	*yemtTelaw*	*tameme*
Aster	REL-she-hates-him	he-got-sick

'The one that Aster hates got sick.' (19)

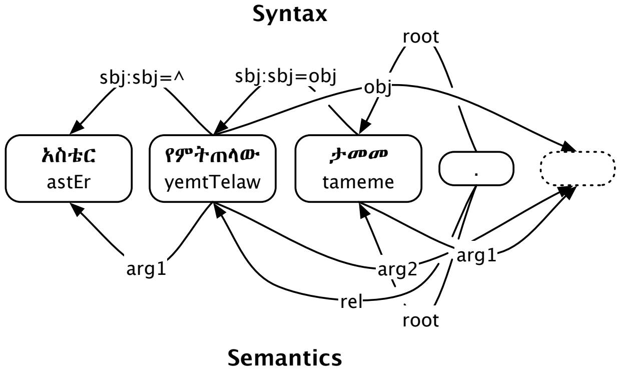

Figure 7: Analysis of a relative clause with no modified noun.

Without further constraints, however, the grammar assigns multiple analyses to some sentences and parses some ungrammatical sentences with relative clauses. Consider the following ungrammatical sentence.

**astEr*	*yemtTelaw*	*wendlj*	*tamemec*
Aster	REL-she-hates-him	boy	she-got-sick

'The boy that Aster hates (she) got sick.' (20)

This satisfies the constraint that the subject of the main verb *tamemec* agree with some feature of the relative verb (its subject) and the constraint that the some feature of the relative verb (its object) agree with the modified noun *wendlj*. To exclude sentences like this, we need a further XDG principle, which we call the Cross-Agreement Principle. This specifies a fundamental fact about relative clauses in all languages, that the same noun functions as an argument of two different verbs, the main clause verb and the relative verb. The Cross-Agreement Principle forces the same feature of the relative verb to agree with the main clause verb and the modified noun. By this principle our parser finds no analysis for sentence (20) because the feature of the relative verb *yemtTelaw* that agrees with the modified noun (its object) differs from the feature that agrees with the main verb (its subject). This is illustrated in Figure 8. The grammar fails to parse this sentence because the features marked with red boxes do not agree.

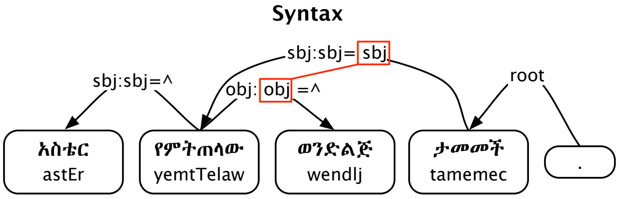

Figure 8: Violation of the Cross-Agreement Principle. The features in red boxes should match.

5. Conclusions

This paper has described an implementation of Extensible Dependency Grammar for the Semitic language Amharic. Amharic is interesting because it suffers from a serious lack of computational resources and because its extreme morphological complexity and elaborate interactions of morphology with syntax present challenges for computational grammatical theories. Besides the strongly lexical character that it shares with other dependency grammar frameworks, XDG is attractive because of the modularity offered by separate dimensions. We have seen how this modularity permits us to handle the agreement constraints on a relative verb by treating such verbs as the heads of noun phrases on the Syntax, but not the Semantics dimension. We have also seen that XDG requires some augmentation to deal with null subjects and objects and the intricacies of verb agreement. These complexities of Amharic are not unique. Much of what has been said in this paper also applies to other Ethio-Semitic languages such as Tigrinya. In addition to expanding the coverage of Amharic, further work on this project will be directed at developing synchronous XDG grammars to support translation between the different Semitic languages spoken in Ethiopia and Eritrea.

6. References

Bob Carpenter. 1992. *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge.

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

Ralph Debusmann, Denys Duchier, and Geert-Jan M. Kruijff. 2004. Extensible dependency grammar: A new methodology. In *Proceedings of the COLING 2004 Workshop on Recent Advances in Dependency Grammar*, Geneva/SUI.

Ralph Debusmann. 2007. *Extensible Dependency Grammar: A Modular Grammar Formalism Based On Multigraph Description*. Ph.D. thesis, Universität des Saarlandes.

Kais Dukes, Eric Atwell, and Abdul-Baquee M. Sharaf. 2010. Syntactic annotation guidelines for the Quranic Arabic treebank. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, Valletta, Malta.

Yitna Firdyiwek and Daniel Yaqob. 1997. The system for Ethiopic representation in ASCII. URL: citeseer.ist.psu.edu/56365.html.

Michael Gasser. 2009. Semitic morphological analysis and generation using finite state transducers with feature structures. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 309–317, Athens, Greece.

Ruth Kramer. 2009. *Definite Markers, Phi Features, and Agreement: a Morphosyntactic Investigation of the Amharic DP*. Ph.D. thesis, University of California, Santa Cruz.

Wolf Leslau. 1995. *Reference Grammar of Amharic*. Harrassowitz, Wiesbaden, Germany.

Marwan Odeh. 2004. Topologische dependenzgrammatik fürs Arabische. Technical report, Saarland University. Forschungspraktikum.

Jorge Marques Pelizzoni and Maria das Graças Volpe Nunes. 2005. N:m mapping in XDG — the case for upgrading groups. In *Proceedings of the Workshop on Constraint Solving and Language Processing*, Roskilde, Denmark.