LFG parse disambiguation for Wolof

Cheikh M. Bamba Dione
University of Bergen

ABSTRACT

This paper presents several techniques for managing ambiguity in LFG parsing of Wolof, a less-resourced Niger-Congo language. Ambiguity is pervasive in Wolof and This raises a number of theoretical and practical issues for managing ambiguity associated with different objectives. From a theoretical perspective, the main aim is to design a large-scale grammar for Wolof that is able to make linguistically motivated disambiguation decisions, and to find appropriate ways of controlling ambiguity at important interface representations. The practical aim is to develop disambiguation strategies to improve the performance of the grammar in terms of efficiency, robustness and coverage.

To achieve these goals, different avenues are explored to manage ambiguity in the Wolof grammar, including the formal encoding of noun class indeterminacy, lexical specifications, the use of Constraint Grammar models (Karlsson 1990) for morphological disambiguation, the application of the c-structure pruning mechanism (Cahill *et al.* 2007, 2008; Crouch *et al.* 2013), and the use of optimality marks for preferences (Frank *et al.* 1998, 2001). The parsing system is further controlled by packing ambiguities. In addition, discriminant-based techniques for parse disambiguation (Rosén *et al.* 2007) are applied for treebanking purposes.

Keywords: LFG, computational grammar, Constraint Grammar, c-structure pruning, discriminant-based disambiguation, Wolof, underspecification, optimality marks

1 INTRODUCTION

This paper deals with the ambiguity problem in the process of analyzing texts in Wolof, a less-resourced language.[1] Specifically, it reports on several techniques used to manage ambiguity in a broad-coverage computational grammar and parser for Wolof. The grammar is implemented in the linguistic framework of Lexical Functional Grammar (LFG) (Kaplan and Bresnan 1982) using the Xerox Linguistic Environment (XLE) (Crouch *et al.* 2013).[2] In LFG, traditional analyses focus on two levels of syntactic representation (Kaplan and Bresnan 1982): Constituent structure (c-structure) models the surface exponence of syntactic information (e.g., word order, dominance and phrasal groupings), and functional structure (f-structure) represents grammatical functions like subject and object.

Wolof, like most natural languages, has pervasive ambiguity, that is, a word or sentence can be analyzed in more than one way. The language is rich in ambiguities of many kinds, including morphological, lexical, syntactic and semantic ambiguities. The ambiguity phenomenon is perhaps the most serious problem faced by natural language processing (NLP) systems, and this is true for many reasons. First, ambiguity typically pertains to all levels of sentence analysis. As MacDonald *et al.* (1994) noted, theoretically, linguistic information can be ambiguous at any given point in a sentence. Furthermore, many sentences that do not seem ambiguous to humans, due to their extensive world knowledge, may present ambiguities to automatic parsers (and to other NLP systems as well in general). Accordingly, large-scale, linguistically motivated grammars tend to be massively ambiguous. Ambiguities can arise, for example, via alternative definitions of morphological and lexical entries, from syntactic or semantic ambiguities, and the interaction of the different ambiguities.

Second, ambiguity typically increases the range of possible interpretations of natural language, and a parser has to find a way to

[1] Wolof is a member of the Senegambian branch of the Niger-Congo language family mainly spoken in Senegal, Gambia and Mauritania. Some Wolof speakers can also be found in Guinea, Guinea-Bissau, Mali and France (see `http://www.ethnologue.com/language/WOL`).

[2] See Section 4 for a brief description of the implementation of the Wolof LFG grammar.

deal with this. It also increases the search space, therefore leading to a combinatorial explosion, which results from multiplying up each individual ambiguity. For instance, for a ten word sentence in which each word could have three interpretations, there are 59,049 possible interpretations for the whole sentence. The situation is exacerbated by the interaction of independent ambiguities. Due to syntactic, semantic and pragmatic ambiguities, the actual number of possible interpretations will be huge.[3] To attempt to resolve all these interpretations becomes hardly possible in a reasonable time.

In this work, the concern for ambiguity management stems both from theoretical and practical requirements and goals. From a theoretical point of view, an important purpose of this work is to develop a parsing system for Wolof which is able to disambiguate (when necessary) the input text in order to ensure correct analysis of the language. In other words, given a string and a context, the aim is to have a system that is able to distinguish the intended reading from the implausible one, but also to preserve linguistically appropriate ambiguities. For an NLP system, this kind of disambiguation decision is particularly relevant, as has been emphasised by Manning and Schütze (1999, pp. 17-18):

> An NLP system needs to determine something of the structure of text – normally at least enough that it can answer "Who did what to whom?" Conventional parsing systems try to answer this question only in terms of possible structures that could be deemed grammatical for some choice of words of a certain category. [...] Therefore, a practical NLP system must be good at making disambiguation decisions of word sense, word category, syntactic structure, and semantic scope.

A secondary, but no less important objective is to apply methods for ambiguity management with the aim to gain efficiency, while maintaining parsing accuracy. Thus, following previous work done in creating language resources and tools for Wolof (Dione 2012b, 2013a), the present research discusses the avenues explored to improve the efficiency and performance of the parser (i.e., to speed up the grammar

[3] For instance, according to Manning and Schütze (1999), Martin *et al.* (1987) report their system giving 455 parses for the sentence *List the sales of the products produced in 1973 with the products produced in 1972*.

development activity and to increase parsing robustness and coverage) by managing ambiguity. The applied methods aim to avoid or minimise the combinatorial explosion that results from ambiguity, as well as to facilitate maintainability of a large code base.

This work addresses several research questions with respect to dealing with ambiguity in linguistically motivated grammar development projects. The first question is how to choose an approach to ambiguity. Managing ambiguity often takes the form of a binary decision: either eliminate or preserve the ambiguity. While the former is the most obvious approach, it is not always feasible or desirable. For instance, handling ambiguity caused by prepositional phrase (PP) attachment may require context and linguistic intuition. For example, in the sentence *she opens the door with the key*, the key is more likely perceived as an instrument used to open the door, rather than it being a feature of the door. Nevertheless, as Chantree (2004, pp. 2) pointed out, "the decision of whether to disambiguate this sentence or not might depend upon the users' proficiency with English and the context provided by the surrounding text."

Conversely, an obvious approach to ambiguity-preserving parsing is to provide the grammatical descriptions or constraints to generate all possible readings. This approach may be problematic in that the number of potential readings might grow exponentially with the length of the sentence. Thus, even though it is clear that "some ambiguities can safely be left in the text", the question still remains over "which ones can be left and which ones must be removed" (Chantree 2004, pp. 1). In linguistically motivated grammar development projects, there is this split between providing the grammatical descriptions to generate all possible readings on the one hand, and the selection of the appropriate one in a given context on the other hand. This paper is adding the latter view to the Wolof project, building on prior work from the LFG framework and other approaches.

A second important question is when and how to attack ambiguity. Ambiguity management can occur before, during or after parsing. If an ambiguity is dealt with early, there is the possibility of losing the right analysis. If it is dealt with late, there is the computational cost of processing many analyses. As Copperman and Segond (1996, pp. 8) pointed out, "the proper balance point in this tradeoff varies for different types of ambiguities, and there is no universal metric".

Concerning the methodology, there exist in the literature a wide range of advanced techniques that can be applied to tackle ambiguity issues, including statistical and non-statistical ones. However, when dealing with ambiguity in languages like Wolof, there is a restriction on the use of certain disambiguation methods. Due to the lack of resources, there is a very limited possibility to apply statistical approaches that often require a large data set to ensure reliable results.

To address these different research questions and to decide among the alternative ways of managing ambiguity, this work is based on three main premises. First, ambiguities are divided into different categories. This is essential to better distinguish ambiguities that are so liable to misinterpretation or to computational complexity that they should be removed from those which can be allowed to remain. A second important point is to manage ambiguity at various levels of description. As Attia (2008, pp. 4) pointed out, it seems to be a good idea "to deal with ambiguity not as one big problem, but rather as a number of divisible problems spreading over different levels of the sentence analysis: pre-parsing, parsing and post-parsing stages." Finally, the third consideration is to manage ambiguity in a systematic way using approaches that can be applied to languages having a lack-of-data problem.

In this work, disambiguation is divided into three stages: pre-parsing, parsing and post-parsing. Disambiguation at the pre-parsing stage focuses on discarding some morphological analyses that are implausible with respect to a given context. The parsing phase covers the topics of the formal encoding of noun class indeterminacy via underspecification, the application of syntactic constraints, lexical specifications and the use of a probabilistic context-free grammar. The post-parsing stage includes the application of preference marks for ranking analyses, and the use of grammar engineering tools for packing ambiguities. In addition, the post-parsing stage involves manually selecting parse solutions using discriminants (Rosén *et al.* 2007). Discriminant-based disambiguation is employed as a timesaving method for constructing a treebank for the language by automatically parsing a corpus with the Wolof LFG grammar. One of the motivations for building the treebank is to create a gold standard test set for Wolof that can be used to evaluate the parser as well as the effect of the other disambiguation methods, e.g., the use of optimality marks for prefer-

ences (Frank *et al.* 1998, 2001) and statistical disambiguation. Because quality controlled treebanks that can serve as gold standards cannot be constructed without considerable manual effort towards ambiguity resolution (Rosén *et al.* 2007), discriminant-based disambiguation is used as an intelligent way of minimizing these efforts. The aim is to optimize the efficiency of manual disambiguation, as inspecting full analyses proved to be a tedious and time-consuming task.

I will attempt to show how the application of the different disambiguation techniques discussed in this paper helps to manage ambiguity and to reduce parse time in the process of analyzing texts in Wolof. However, note that the various disambiguation methods are applied on different parsing levels and parser versions, and thus have interactions that are very difficult to control systematically. Also, note that the purpose is not to give an exhaustive account of all the disambiguation methods used within this research work or to provide an exhaustive overview of their systematic interaction but to illustrate ambiguity management in LFG parsing of Wolof focusing on some example constructions which present particular challenges for grammar development and treebanking work for the language.

This paper is structured as follows. Section 2 provides a general description of ambiguity in natural languages and a common categorization of the different ambiguity types. Section 3 presents evidence that Wolof is massively ambiguous, particularly with respect to morphological, lexical and syntactic ambiguities. Section 4 briefly presents background information on the Wolof grammar relevant for the discussion of ambiguity management in subsequent sections. Section 5 discusses techniques for handling morpho-lexical and syntactic ambiguities, including the formal encoding of noun class indeterminacy, lexical specifications, morphological and lexical disambiguation based on Constraint Grammar (CG) (Karlsson 1990). Section 6 presents some approaches to syntactic ambiguity used for Wolof, including c-structure pruning (Cahill *et al.* 2007, 2008; Crouch *et al.* 2013) and optimality marks (Frank *et al.* 2001). Section 7 presents grammar engineering tools for packing ambiguity in XLE and discusses disambiguation strategies used to increase parsing efficiency by removing spurious ambiguities. Section 8 describes discriminant-based disambiguation techniques to LFG grammars (Rosén *et al.* 2007). A conclusion is given in Section 9.

2 AMBIGUITY CATEGORIZATION

Research on ambiguity typically distinguishes between the scope (global vs. local) (Gazdar and Mellish 1989) and types of ambiguity (Gómez 1996).

2.1 *Scope: global vs. local*

Global ambiguity means that an entire word string has more than one structure associated with it, as in (1).

(1) Flying planes made her duck. (Gómez 1996, pp. 16)

The sentence in (1) has various readings, including the two following ones: (i) the airplanes made her change her position; (ii) the act of piloting made her change her position. In terms of LFG/XLE, global ambiguities give rise to different whole-sentence f-structures. In general, global ambiguities are linguistically appropriate, and therefore may need to be preserved: Their resolution typically requires semantic and/or pragmatic knowledge.

In contrast, the sentence in (2) from Gómez (1996, pp. 16) involves a local ambiguity, because some subparts of the whole string have different readings. Readers who process this sentence and focus on the last three words, might settle on the existence of a sentential subconstituent made up of *SGEL sold Xerox*.

(2) The company that bought SGEL sold Xerox.

In contrast to global ambiguity resolution, local ambiguity can sometimes be resolved by syntactic analysis. From that perspective, local ambiguity includes the following ambiguities discussed in Section (2.2): lexical, morphological, and syntactic ambiguities that are resolved when a larger sentential context is taken into account.

2.2 *Types of ambiguity*

The causes for obtaining different analyses for an input string (a word or a sentence) might be diverse, including lexical, morphological, syntactic and referential, and the interaction of all these levels. This work will concentrate on these aforementioned ambiguity types.

In lexical ambiguity, a given word may be assigned to more than one grammatical category or part of speech (POS) according to the context. For instance, the English word *bank* could be a noun or

verb. Morphological ambiguity typically refers to ambiguity within the same syntactic category, that is, ambiguity of different forms of one lexeme within the same POS. For instance, in the sentence *I saw her run to the bank*, the word *bank* is unambiguously a noun, however, it is still unclear whether it refers to a financial institution or to a river side. This phenomenon is also known as word sense ambiguity. Morpholexical ambiguity is not a uniform phenomenon, but a phenomenon that distinguishes between homonymy and polysemy (Klepousniotou 2002). In theoretical linguistics, the etymological derivation of words and the 'relatedness/unrelatedness' of meaning – a matter of degree that relies on native speaker's feeling – have been proposed for the distinction between homonymy and polysemy. In homonymy, a lexical item accidentally carries two (or more) distinct and unrelated meanings, while in polysemy, a single lexical item has several different but related senses.

Syntactic ambiguity can be divided into structural and functional ambiguity. A sentence is viewed as structurally ambiguous if it can be interpreted or represented by more than one syntactic structure. Attachment of adjuncts (e.g., PP attachment and adjective attachment) represents a canonical case of structural ambiguity. An instance of PP attachment in Wolof is given in (3).[4]

(3) *Góor g-i séen xale b-i ci saxaar g-i.*
man cl-DFP see child cl-DFP in train cl-DFP
"The man saw the child in the train."

The ambiguity here arises from the fact that the grammar provides several sources for the PP. The attachment of the PP *in the train* is syntactically permissible both to the noun phrase (NP) *the child* and the verb *saw*. In general, attachment of adjuncts results semantically in scope ambiguity. The outcome of the attachment depends mainly on two factors: (i) which subcategorization frame the verb prefers and (ii) which attachment is semantically more plausible. In LFG, this ambi-

[4] Abbreviations in the glosses: ADV: adverb; cl: noun class marker; COMP: complementizer; CONJ: conjunction; COP: copula; DET: determiner; DFP: definite proximal; DFD: definite distal; +F: finite; GEN: genitive; INF: infinitive; NDF: indefinite article; NEG: negation; NSFOC: non-subject focus; PREP: preposition; PST: past tense; pl: plural; Rel: relative; S: subject; SFOC: subject focus; sg: singular; VFOC: verb focus; 1, 2, 3: first, second, third person.

guity is reflected both in the c-structure and in the f-structure (adjunct attachment). Adjunct attachment is notoriously difficult: The syntax has no way to determine the attachment, even if humans can.

In contrast, functional ambiguity is semantic without necessarily involving phrase structure distinctions. In LFG, this refers to ambiguity within the f-structure. A typical example is when a constituent can bear both an oblique argument and an adjunct function within the functional structure (see Section 3.1.4).

Referential ambiguity arises, when more than one object is being referred to by a noun phrase or a deictic expression. This is typically the case when readers or listeners are unable to select a unique referent for a linguistic expression out of multiple candidates. For instance, in the sentence *After **they** finished the exam, the students and lecturers left.*, the pronoun *they* is ambiguous: It can refer to *students* only, to *lecturers* only, or to both. One aspect I will point out in the discussion of referential ambiguity is unclear reference of pronominal subjects in some constructions in Wolof (see Section 3.3).

Syntactically legitimate ambiguities contrast with so called spurious ambiguities, which constitute a purely engineering problem. Spurious ambiguities can refer to duplicated solutions – the same full-sentence f-structure associated with different c-structures or processing sequences – or incorrect f-structures, that is, a reading of the sentence that a native speaker would not attest to. This work will focus on the former definition. Spurious ambiguities mainly refer here to multiple parse solutions that are completely identical (Komagata 2004), for example, when many different derivations or trees generate the same structure. As such, this ambiguity type poses serious grammar engineering issues in terms of efficiency, and therefore needs to be removed.

Having discussed the main ambiguity types the present work will deal with, I will now turn to some ambiguity issues in Wolof that present a particular challenge in the context of grammar implementation.

3 AMBIGUITY IS PERVASIVE IN WOLOF

3.1 *Morphological and lexical ambiguity*

As noted above, ambiguities can arise from linguistically justified lexical and morphological ambiguities. Morpholexical ambiguity in the Wolof grammar arises mainly from polysemy and homonymy caused by Wolof noun classes (NC). Conversely, lexical ambiguity stems from different sources, including ideophones acting as verb collocations, words with several parts of speech and verbs with various subcategorization frames. These issues are discussed in the next sections.

3.1.1 Ambiguity due to Wolof noun classes

As is typical for Atlantic languages (Sapir 1971), Wolof is a noun class language with noun class agreement (McLaughlin 2010; Torrence 2005). The language has 13 noun classes identified by their index:[5] 8 singular (*b, g, j, k, l, m, s, w*), 2 plural (*y, ñ*), 2 locative (*f, c*), and 1 manner (*n*). Of the singular noun classes, the *s* class also functions as a diminutive class. As for plural noun classes, *y* is the class of most nouns, while *ñ* is the class of a restricted small set of human nouns. Accordingly, a noun may belong to as many as three classes (McLaughlin 2010): a singular, a plural and a diminutive singular class.

Unlike the noun class system found in Bantu languages, nouns in Wolof lack a class marker on the noun itself. Instead, class membership is marked on noun specifiers such as determiners (definite and indefinite articles, demonstratives) or quantifiers, on relative pronouns, etc. For instance, Wolof possesses two definite and two indefinite articles, all agreeing in class with the noun. Indefinite and definite determiner phrases (DPs) have a different word order, as shown in (4-6). While the definite article obligatorily follows the NP, the indefinite article obligatorily precedes the NP. The vowel suffixes *i* and *a* on the definite articles respectively encode proximity and distance in space, time, or conversation. In contrast, the vowel prefix *a* marks indefiniteness.

[5] The noun class index functions as a stem to which a determiner/pronoun/etc. affix is added. In this paper, the stem is glossed *cl.*

(4) *a-b xale*
NDF-cl child
"A child"

(5) *xale b-i*
child cl-DFP
"The child (here)"

(6) *xale b-a*
child cl-DFD
"The child (there)"

Although eight singular classes and two plural classes can clearly be distinguished, the morphological paradigms of the noun class system are characterised by noun class syncretism, that is, a single morphological form corresponds to two or more morphosyntactic descriptions (Baerman *et al.* 2005). For example, due to homonymy/polysemy, the word form *ndaw* in (7) corresponds to different noun classes, as marked on the definite articles. The noun surfaces in the same form both in the singular and plural noun class.

(7)

Number	Noun Class	Example
Singular	*g* class	*ndaw* ***g-i***
	s class	*ndaw* ***s-i***
	l class	*ndaw* ***l-i***
Plural	*ñ* class	*ndaw* ***ñ-i***
	y class	*ndaw* ***y-i***

The paradigm in (7) shows that some Wolof nouns like *ndaw* have many readings at the word level, thereby increasing ambiguity in the grammar. The examples in (8) illustrate sentences in which the same form *ndaw* occurs with the noun class *g* in (8a), *s* in (8b), *l* in (8c), *ñ* in (8d) and *y* in (8e).

(8) a. *A-**g** ndaw gàddaay na sama jëmm j-ii.*
NDF-cl youth leave +F.3sg POSS1SG face cl-DEM
"I do not look young anymore."

b. *Ta amaana kon, di-na jël ndaw **s**-i.*
CONJ perhaps then, IPF-+F.3sg take woman cl-DFP
"And he would then possibly marry the woman."

c. *Ndaw **l**-i ñëw na.*
messenger cl-DFP arrive +F.3sg
"The messenger has arrived."

d. *Ma xool ndaw **ñ**-i.*
1sg look.at young cl-DFP
"So, I look at the young people."

e. *Nu doon ndaw **y**-u gëm l-a nu-y wut.*
1pl COP.PST young cl-Rel believe cl-Rel 1pl-IPF look.for
"We were young people who believed in what we were doing."

Likewise, the word form *mag* can occur with at least four noun classes (*j*, *m*, *ñ*, and *y*): for example, *mag* ***j****-i* 'the brother', *mag* ***m****-i* 'the old man' *mag* ***ñ****-i* 'the old people', and *a-****y*** *mag* 'some old people'. Accordingly, in the Wolof grammar, the nominal coordination in (9) has at least 20 readings that result from the ambiguous forms of the two conjuncts.

(9) Mag ak ndaw
old CONJ young
"Old and young people"

3.1.2 Co-verbs using *ne/ni*

In the Wolof grammar, lexical ambiguity arises from ideophonic expressions. *Ideophone* is a common term for expressive vocabulary found in languages in Africa, Eurasia, and Australia. Doke (1935, pp. 118) defines an ideophone as "a vivid representation of an idea in sound" or "a word, often onomatopoeic, which describes a predicate, qualificative or adverb in respect to manner, colour, sound, smell, action, state or intensity".

In morphophonological and syntactic terms, ideophones represent onomatopoeic or synesthetic expressions which tend to have an emotive function and exhibit specific syntactic, morphological, and/or phonological properties that make them a distinct group (Voeltz and Kilian-Hatz 2001). In addition, ideophones are associated with spoken and dramatic registers of speech. Accordingly, a common distributional feature of ideophones is that they tend to occur in collocations with a restricted set of generic verbs such as 'do', 'say', or 'go' (Creissels 2001). Ideophones seem to be well documented, but little work has been done on their implementation in computational grammars.

In Wolof, ideophones "can either accompany a verb as an intensifier and are thus known as coverbal ideophones, or they can be used in quotative constructions with the verb *ne* 'say'" (McLaughlin 2004, pp. 256), as in the examples in (10) and (11).

(10) *Sa mbubb dafa set wecc.*
2sg:POSS gown 3sg:VFOC ADJ:clean IDEO
"Your gown is perfectly clean." (McLaughlin 2004, pp. 256)

(11) *Mu ne tekk.*
3sg say IDEO:of saying
"S/He was quiet."

The use of coverbal ideophones increases the ambiguity of collocational verbs like *ne*, which belongs to the items with the most notorious hotspots of ambiguity. It can additionally be a comparative preposition (12), a complementizer (13), a regular verb without coverbal ideophones (14) and a copular verb (15). Accordingly, a special treatment of ideophones was necessary to limit this ambiguity.

(12) *Mu mel ne xale.*
3sg look like child
"S/He looks like a child."

(13) *Mu xam ne dem na.*
3sg know COMP go +F.3sg
"S/He knows that s/he has left."

(14) *Mu ne leen ñu dem.*
3sg tell 3sg/O 3pl go
"S/He told them to go."

(15) *Mu ne ci kër gi.*
3sg COP prep house cl.DFP
"S/He was in the house."

3.1.3 Lexical ambiguity: POS

As the collocational verb *ne* discussed in the previous section illustrates, in Wolof (like in many languages), most words can have several parts of speech. This includes lexical items that can belong to different word classes such as determiners, bound and free relative pronouns, complementizers, etc. In particular, short tokens like *la* are multiply ambiguous, making it evident that lexical ambiguity is extremely widespread in Wolof. This item can have both a verbal and a non-verbal reading, as shown in (16) from Dione (2014). In this example, *la* can be a non-subject focus morpheme (INFL) (16a), a copular verb (16b), a clitic object (16c), a determiner or a bound pronoun (16d), a free relative pronoun (16e) or a complementizer (16f).

(16) a. *Fas* ***la*** *gis.* *INFL*
horse.w-cl 3sg.FOC see
'It is the horse that he saw.'

b. *Fas* ***la.*** *Non-subject copula*
horse.w-cl COP.3sg
'It is a horse.'

c. *Gis-u-ma* ***la.*** *Clitic object*
see-NEG-1sg 2sgO
'I haven't seen you.'

d. *Ngelaw* ***la*** *agsi* *Determiner/Rel. Pron.*
wind.l-cl l-cl.det/REL arrive
'The wind came around / which came around.'

e. ***la*** *mu gis-oon ...* *Free relative*
free.REL he see-PST
'What he saw ...'

f. ***la*** *mu doon ngelaw lépp ...* *Complementizer*
COMP 3sg ipf.PST be.windy quant
'Despite the fact that it was windy ...'

In the grammar, assigning so many parts of speech to the same word form (e.g., to the lexical entry *la*) poses both ambiguity and efficiency problems.

3.1.4 Lexical ambiguity: subcategorization frames

As with grammatical categories, words have often more than one subcategorization frame. In English, the verb *break* may have a transitive and an intransitive reading (e.g., *I broke it* vs. *It broke*). Likewise, the verb *want* may have bare transitive reading (I want something) or a transitive with infinitive reading (I want it to leave). Similarly, the Wolof verbs can have several subcategorization frames; for example, the verb *dugg* 'enter' in (17) has at least two subcategorization frames: It may have a bare intransitive and an oblique reading.

(17) *Mu dugg ci kër gi.*
3sg enter in house cl.DFP
"S/He entered the house."

In (17), a lexical and functional ambiguity problem arises caused by the semantics associated with the PP *ci kër gi* "in the house". This ambiguity does not involve structural distinctions, since the constituent is clearly a PP that attaches to the verb *dugg*. The question is: Which grammatical function does this PP bear within the verbal phrase (VP)? Is it an argument or an adjunct of the verb?

On the one hand, one might assume that the PP is subcategorized for by the verb. In particular, that amounts to considering the PP as an instance of oblique arguments, that is, "nonsubject arguments which are not of the appropriate morphosyntactic form to be objects and which do not undergo syntactic processes which affect objects" (Butt *et al.* 1999, pp. 50). On the other hand, the PP may be analyzed as an

adjunct, that is, as an optional constituent of the verb that, when removed, will not affect the remainder of the sentence except to discard from it some auxiliary information. As such, the PP is seen as a modifying phrase that depends on the VP, bearing an adverbial function within the latter phrase.[6]

3.2 *Syntactic ambiguity*

In Wolof, ambiguous lexical forms are also a source of syntactic ambiguity; but, even without lexical ambiguity, there are legitimate syntactic ambiguities such as PP attachment and coordination ambiguity. One might want to constrain these to legitimate cases and make sure they are processed efficiently. Some syntactic ambiguity issues in Wolof are discussed in the next sections.

3.2.1 Structural ambiguity

Structural ambiguity occurs when the arrangement of words in a grammatical structure permits two or more meanings to emerge, as is the case with PP attachment discussed above. Structural ambiguity can also be caused by an interaction of lexically ambiguous forms and syntactic ambiguity, as illustrated in (18). For example, *bi* can be a determiner, a bound or a free relative pronoun or a complementizer; *moom* can either be a verb, a strong pronoun or a topic adverb; *doon* can be a copula or a past progressive auxiliary, etc. Three possible interpretations of this sentence are shown in the translations in (18).

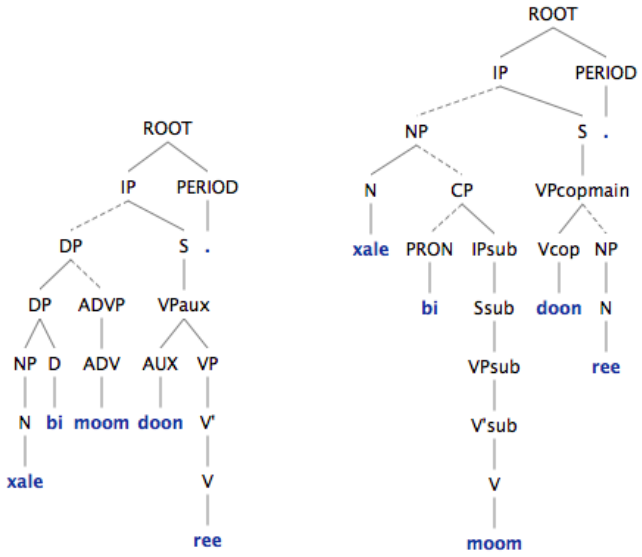
(18) Xale b-i moom doon ree.
child cl-DFP adv.TOP IPF.PST laugh
child cl-Rel own COP laugh
child cl-DFP own IPF.PST laugh

"As for the child, (s)he was laughing."
"The child who owns (something) becomes a laugh."
"The child owns (something) and was laughing."

Before disambiguation, the sentence in (18) has more than 100 c-structure trees that are valid with respect to the grammar. The c-structures for the two first interpretations given in (18) are represented in Figure 1.

[6] For the distinction between arguments and modifiers (in particular between oblique and adjunct functions) and the several tests conducted to illuminate this

Figure 1: Two possible c-structures for *Xale bi moom doon ree*



The third reading arises from coordination without an explicit conjunction. Conjuncts in a coordinate structure can be joined by an overt conjunction (syndetic coordination) or not (asyndetic coordination) (McShane 2005). Like many languages, Wolof permits coordinate structure without an overt conjunction (see Section 6.4).

3.3 *Referential ambiguity: pro-drop and impersonal passive*

In Wolof, an example of referential ambiguity with a global scope arises from pro-drop (Chomsky 1981; Baptista 1995)and Wolof impersonal passive constructions (19).

(19) a. *Góor ñ-i gor na-ñu garab g-i.*
man cl-DFP cut.down +F-3pl tree cl-DFP
"The men cut down the tree."

b. *Gor na-ñu garab g-i.*
cut.down +F-3pl tree cl-DFP
"They cut down the tree."
"The tree was cut down."

distinction, see for example, (Dalrymple 2001).

Example (19) illustrates the pro-drop nature of the language. The sentence in (19a) is similar to the one in (19b), except that in the latter example the overt subject is missing; nevertheless both sentences are grammatical. In (19b), there is no overt subject, because Wolof freely allows the omission of such an argument.

Sentences with a third plural subject like (19b) are ambiguous because they can express both a pro-drop or an impersonal passive reading. Because Wolof lacks a true passive derivation (Voisin-Nouguier 2002), it often uses an active sentence with an impersonal third plural subject to express the passive idea (Torrence 2005). The two different readings of this sentence are reflected in the translations. The ambiguity here is due to the interpretations of the third plural element (also called subject marker) *nañu*. On the one hand, this element can be a referential subject, in which case it is understood to refer to a specific group of individuals who cut down the tree.[7] On the other hand, it can be a third person plural denoting a generalized human subject frequently cited as a source of passives (Givón 1979), meaning that there was cutting down of the tree and that this action has no determinate subject. Impersonal here means simply that the third plural element is not understood to refer to any specific group of individuals.

In short, the prevalence of independent morphological, lexical, syntactic, and referential ambiguities can lead to a combinatorial explosion, making many Wolof sentences massively ambiguous.

4 IMPLEMENTATION OF THE WOLOF GRAMMAR

In the previous section, we have briefly looked at different ambiguity issues in Wolof, paying attention to those relevant to the discussion of developing a grammar for the language. In what follows, I will suggest and discuss some methods for handling these issues, keeping in mind that the application of some of these techniques always has the potential to eliminate a valid analysis. Before suggesting these techniques, I would like to briefly describe the Wolof grammar and the data used for grammar development and evaluation.

[7] Note that the English pronoun *they* in the translation of Example (19b) is to be considered here as a referential and non-arbitrary pronoun.

Developed as part of the Parallel Grammar (ParGram) project (Butt *et al.* 2002), the Wolof grammar provides a formal description of the syntactic analysis of core constructions of the language within LFG, as well as linguistically well motivated analyses of challenging constructions in Wolof, including clitics (Dione 2013a), clefts (Dione 2012a), valency change and complex predicates (Dione 2013b). The grammar parses sentences on the basis of XLE rules and templates, two lexicons, and a cascade of finite-state transducers (FST) (Kaplan *et al.* 2004). In its current state, the grammar has 250 rules (with right-hand sides based on regular expression). The lexicons contain ca. 2000 verb stems and 2836 subcategorization frame–verb stem entries. The preprocessing components of the grammar include a Wolof finite-state morphological analyzer (WoMA) (Dione 2012b), as well as other finite-state modules for tokenization and normalization. The grammar is not part of an application-oriented set-up, meaning that it is not embedded in a larger application pipeline. Consequently, some sources of information that could be applied to eliminate inappropriate readings that may come out of the parser/grammar, such as domain restrictions and selectional restrictions, are mostly not available.

The development of the grammar is based on a corpus of natural Wolof texts. The basic development and test (i.e., unseen) data consist of two disjoint sets of randomly selected sentences from short stories (Cissé 1994; Garros 1997) and a semi-autobiographical novel (Ba 2007). The development and test sets consist respectively of a total of 626 and 2364 sentences used to evaluate the grammar in terms of accuracy and efficiency, but also to assess the effects of design decisions in the grammar and the impact of the disambiguation methods discussed within this work. As the grammar constitutes a starting point for the construction of further NLP resources for Wolof, the test set was run through it to establish a treebank for the language.

5 MORPHOLOGICAL AND LEXICAL DISAMBIGUATION

This section presents systematic approaches used to manage the morphological and lexical ambiguities discussed in Sections 3.1.1-3.1.3. It focuses on the formal encoding of noun class indeterminacy, lexical specifications and CG-based disambiguation.

5.1 *Ambiguity resolution for Wolof noun classes*

In the initial LFG approach to Wolof noun classes, nominal class attributes were represented as atomic feature values. So, the noun and its specifier were elements of the f-structure and had to agree either in the singular (e.g., *NOUN-CLASS-SG*) or plural (e.g., *NOUN-CLASS-PL*) noun classifier. Accordingly, the f-structure for the nominal phrase in (20) was represented as shown in (21). This f-structure representation says that the noun *xale* specifies *b* and *y* as its respective singular and plural noun class, while the specifier *bi* belongs to the *b* class.

(20) *Xale b-i*
child cl-DFP
"The child"

(21)

PRED	'xale'		
NOUN-CLASS-SG	b		
NOUN-CLASS-PL	y		
NUM	sg		
PERS	3		
SPEC	DET	PRED	'bi'
		NOUN-CLASS-SG	b
		DEIXIS	proximal
		DET-TYPE	def

One potential problem with this analysis is that Wolof noun classes typically have forms that can be attributed 'indeterminately' to different values. As Dalrymple *et al.* (2009, pp. 31) noted, "forms that are indeterminately specified for the value of a feature can simultaneously satisfy conflicting requirements on that feature and thus are a challenge to constraint-based formalisms which model the compatibility of information carried by linguistic items by combining or integrating that information."

Similarly, Wolof nouns typically show no noun class distinction. As Example (22) illustrates, a noun like *ndaw* in (8) can satisfy different class requirements. A similar case has been observed for the German noun *Papageien* 'parrots' in (23), which shows no case distinction and can satisfy different CASE requirements (Dalrymple *et al.* 2009).

(22) *Ndaw*
young
G/L/S/Ñ/Y
'young/youth/messenger
(*g*, *l*, *s*, *ñ* or *y* noun class)'

(23) *Papageien*
parrots
NOM/ACC/DAT/GEN
'parrots' (nominative, accusative, dative or genitive)

Because the approach given in (21) relied on specification of simple atomic values for indeterminate features, the integration (typically by unification) of information from head and dependent was problematic. Assuming that a noun like *mag* 'old person' (see Section 3.1.1), for instance, specifies *ñi* for its specifier's nominal class value, and that the determiner *a-y* 'some' and the relative pronoun *ñi-u* 'who/which' specify *ñi*, we obtain a clash of nominal classifiers (e.g., [NOUN-CLASS-PL = y] and [NOUN-CLASS-PL = ñ]) in sentences like (24), leading to the incorrect prediction that the example is unacceptable.

(24) *Ma gis a-y ndaw ñi-u am xam-xam.*
1sg see NDEF-cl young.people cl-Rel have knowledge
"I saw some wise young people."

This problem implies shifting away from the initial approach described in (21). This shift in approach has two different, but interrelated, objectives: to avoid coverage problems for cases like (24), which show that indeterminate forms can stand in for two values simultaneously (like syncretic forms in many languages); and to reduce the number of readings for normal cases by assuming a suitable underspecified representation rather than a disjunctive listing of all options.

Accordingly, the analysis in (21) above is replaced by an approach similar to the representation of CASE proposed in Dalrymple *et al.* (2009).[8] Following this representation, nouns such as *ndaw* and *mag* in (9) will have the feature structure for the noun class attribute, as respectively shown in (25) and (26).

(25) Noun class feature for *ndaw*

$$\begin{bmatrix} \text{NOUN-CLASS} & \begin{bmatrix} \text{G} & + \\ \text{L} & + \\ \text{S} & + \\ \tilde{\text{N}} & + \\ \text{Y} & + \end{bmatrix} \end{bmatrix}$$

(26) Noun class feature for *mag*

$$\begin{bmatrix} \text{NOUN-CLASS} & \begin{bmatrix} \text{J} & + \\ \text{M} & + \\ \tilde{\text{N}} & + \\ \text{Y} & + \end{bmatrix} \end{bmatrix}$$

The value of this attribute allows specification of each noun class by means of a separate Boolean-valued attribute: G, L, S, Ñ, Y, etc. A

[8] Alternatively, the set-based approach to feature resolution (Dalrymple and Kaplan 1997) could be used to handle feature indeterminacy. It allows for an account of complex agreement phenomena like those found in German free relatives, case in Polish coordination and noun class in Xhosa coordination.

negative value indicates the inability of a form to satisfy the corresponding noun class requirement. Nouns and their modifiers specify negative values or do not specify any value for the noun classes they do not express, and specify or are compatible with positive values for the classes they do express.

The noun class specification for the form *ndaw*, which is class-indeterminate, is given in (27); this can be read as requiring that, within the NOUN-CLASS structure, the value for G, L, S, Ñ or Y must be +. Thus, *ndaw* must express some noun class or other, but there are no restrictions on which noun class it expresses. This permits the form to occur in contexts compatible with a positive specification of one or more of the noun classes, and does not impose any negative class specification that would rule out class possibilities for the form.

(27) *Ndaw*; NOUN-CLASS{G|L|S|Ñ|Y} = +

The output for the word form *ndaw* produced by the Wolof morphological analyzer (Dione 2012b) is shown in (28). The FST translates the form into a string that represents its morphological makeup: a noun that agrees with its modifier in the classes *g*, *l*, *s*, *ñ* or *y*. All class indexes compatible with this form should be contained in the output.

(28) ndaw+Noun+Common+g+l+s+ñ+y

We may note in passing that, in the Wolof lexicon, polysemous and homonymous nouns are treated in a similar way. This means that words like *ndaw* that have different related and unrelated meanings are associated with only one lexical entry. This follows the goal to reduce ambiguity for lexical items that have many readings which, however, do not affect the syntax. A similar approach has been taken by the ParGram LFG English grammar (Riezler *et al.* 2002). The different readings of a polysemous item like *bank* ("river bank" or "financial bank") are not distinguished in the grammar, but rather in a semantic post-processor, that is, the English transfer rules.

In the Wolof grammar, this underspecification approach led to substantial reductions in morpholexical ambiguity and parse time. To assess the impact of underspecification, the ambiguity rate and the time the grammar needs to parse the test data have been measured before and after the application of this approach. The results show

that the ambiguity rate decreased by approximately 8%, leading to a reduction of parse time by 4%. In the grammar, this change affected ca. 10 rules, 20 morphological tags, and 39 templates, which are called at many places in the different rules and lexical entries.

In the context of grammar implementation, the advantage of underspecification over the disjunctive approach in terms of processing efficiency has also been attested in previous work (Flickinger 2000; Crysmann 2005). According to Flickinger (2000), the compactness of linguistic description achieved by the elimination of disjunctive features provides a great benefit in terms of processing efficiency. The performance comparison of the disjunctive and the underspecification approach shows that the latter outperforms the former by a factor of 3–4, with an otherwise unchanged grammar[9] running on the same processing platform (PAGE, Uszkoreit *et al.*, 1994).

5.2 *Disambiguating co-verbs using ne/ni*

Like Wolof noun class ambiguity, ambiguity caused by ideophones are dealt with using a systematic approach. Wolof ideophones behave like particles that are selected by the verb. In this respect, they show some similarity to Norwegian particles like *ut* in the sentence in (29).

(29) *Han vil slippe ut hund-en.*
3sg will release out dog-DEF
"He will let the dog out."

Given this similarity, the coverbal ideophones are treated as particles. As in the Norwegian grammar (Dyvik 2000), these particles are introduced by a special c-structure category PART of adverbial type (i.e., PART[adv]). The verbs like *ne* which subcategorize for ideophones are constrained to specify the lexical form (30) of the particle.

(30) V-SUBJ-PRT (P PART) = @(CONCAT P '* PART %FN)
@(VOICE (↑ PRED) = '%FN < (↑ SUBJ) > ')
(↑ PRT-FORM) =$_c$ PART.

The rule in (30) makes use of the XLE built-in template *CONCAT* (Crouch *et al.* 2013) to concatenate all arguments except the last argument and to produce the final argument. If applied to the structure *ne*

[9] The LinGO English Resource Grammar (Copestake and Flickinger 2000).

tekk in (11), %FN will be set to *ne*tekk* once the *tekk* particle is found and (↑ PART) is set to *tekk*. The rule shows a subcategorization frame for an intransitive verb like *ne* functioning as the verb in the structure. The frame uses a constraining equation (↑ PRT-FORM) = c PART to require a special particle selected by the verb, constraining the value of the 'PRT-FORM', introduced by *PART*. This rule also specifies that such a structure – the verb and co-verbs taken together – requires a special treatment. The *CONCAT* device allows for concatenation of two independent lexical entries that coreference each other in the lexicon. The lexical entries of base verbs introduce the semantic form of the particle verb with its argument structure. The lemma of the base verb and the form of the particle are concatenated via the device so that the combination of the two, rather than just the lemma of the base verb, is the PRED of the f-structure.

One of the reasons for treating the verb and the particle in this way is that syntactic constituents can intervene between the verb and the particle, as illustrated in (31). Another reason is that, for instance, the verb *ne* can appear with an OBJ, but not if there is a PRT-FORM *tekk*, which is provided by the particle. In such a case, this verb can only be intransitive.

(31) *Mu ne ma jàkk.*
3sg say 1sg IDEO:of staring at someone
"S/He was staring at me."

Figure 2 shows an example analysis of Wolof coverbal ideophones.

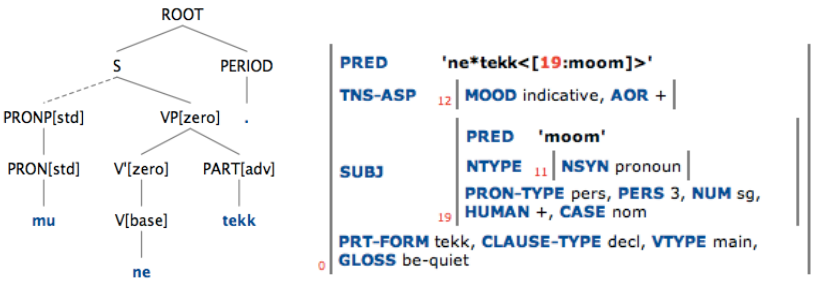


Figure 2: Analysis of Sentence (11) as an illustration of the treatment of coverbal ideophones in Wolof

In addition, some further steps were required. First, the ideophonic particle forms had to be explicitly listed in the lexical entry of the collocational verb. Second, several subcategorization frames were

defined to allow for the different verb argument structures. For a verb like *ne*, the frames given in Table 1 were defined.[10] Finally, optimality marking (see Section 6.2) was used to state a preference reading for ideophones as such, when they occur with a collocational verb. For instance, some rare ideophones like *tekk* in (11) may belong to another grammatical category. In fact, *tekk* can also be a noun. However, in this configuration, the noun reading is very unlikely, not to say impossible. Hence, for such rare cases, the use of the preference mark for the ideophone reading helps to discard the implausible readings arising from nouns from the output.[11]

Table 1: Subcategorization frames for *ne* as a collocational verb

Subcategorization frame	Examples
V-SUBJ-PRT	*ne cell* 'to be silent'
V-SUBJ-OBJ-PRT	*ne jàkk* 'to stare at someone / something'
V-SUBJ-OBL-TH-PRT	*ne mërr ak* 'to disappear with'
V-SUBJ-OBJ-OBJ-TH-PRT	*ne keww kenn dara* 'to stare at somebody with something'
V-SUBJ-XCOMP-PRT	*ne mes ànd ak njaxlaf* 'to disappear quickly and in a dynamic way'
V-SUBJ-OBL-COMPAR-PRT	*ne ràyy ni melax* 'to flash like a lightning'

Applied on the test set, this approach substantially reduces the ambiguity rate related to the coverbal ideophones by ca. 4%. Also, the parse time for coverbal ideophones could be reduced by 16%, while maintaining the parsing accuracy. This change affected 7 templates and ca. 139 verb subcategorization frames.

5.3 *Coping with POS ambiguity*

One of the major causes of non-determinism in a computational grammar is POS ambiguity. When a word can belong to two different grammatical categories, a non-deterministic parser may have to explore both possibilities.

As noted in Section 3.1.3, *la* in Wolof is very ambiguous between different grammatical categories, and because of this, the sentence in (16a), repeated in (32), has ca. 42 readings. The multi-tagged text of this sentence before disambiguation is displayed in (33). The analysis

[10] *PRT* is the abbreviation for *particles*.

[11] See Section 6.2 for a more detailed discussion of using optimality marks in the Wolof grammar.

line <"la"> has received seven different readings in the morphology analysis.

(32) *Fas* ***la*** *gis.*
horse.w-cl 3sg.FOC see
'It is the horse that he saw.'

(33) <"fas"> fas + V + Base + Main + Active
fas + N + Common + w + y + Count
fas + N + Common + g + y + Count

<"la"> la + Comp + Free
la + Det + Def + l + Sg + Dist
la + Pron + Rel + l + Sg + Dist
la + Pron + Free + l + Sg + Dist
la + INFL + NonSubjCopula + 3SgSubj
la + INFL + CompFoc + 3SgSubj + Indic
la + Clt + Obj + Pers + 2 + Sg + Weak + Acc

<"gis">
gis + V + Base + Main + Active
gis + N + Common + b + y + Count

<".">
. + PERIOD

A possible method to tackle the non-trivial issue of POS ambiguity is to use a methodological paradigm that is based on local morphological disambiguation performed by context-sensitive disambiguation constraints. Local disambiguation refers to "constraints or strategies that make it possible to discard some readings just by local inspection of the current cohort" (i.e., the set of readings from a word form) "without invoking any contextual information" (Karlsson 1990, pp. 2).

Constraint Grammar (CG) (Karlsson 1990) is an example of such a mechanism that allows this kind of local disambiguation. CG is a language-independent formalism for surface-oriented, morphology-based parsing of running text (Karlsson 1990). In this formalism, context dependent rules are compiled into a grammar that assigns readings to words or other tokens in a given text. Tags can be of different types, including lexeme, base form, syntactic or semantic tags, valency, etc. Constraints are used to discard as many alternatives as possible. Constraint rules typically consist of two parts: (i) an operation on a pattern and (ii) a context. Each rule either adds, removes,

selects or replaces a tag or a set of grammatical tags in a given sentence context. A context can be defined as any combination of words or tags in a given sentence. Context conditions can be linked to any tag or tag set of any word anywhere in the sentence, either locally (defined distances) or globally (undefined distances). Context conditions in the same rule may be linked (i.e., conditioned upon each other) negated, or blocked by interfering words or tags.

The idea that lexical ambiguity can be reduced for a given sentence by using the CG model is particularly attractive for at least two reasons. First, the CG-based model does not require a large data set for training. Second, the model allows a grammar writer to select meanings or remove them from words or other tokens, depending on local information. The context sensitive constraints of this model provide a disambiguation possibility that is generally unavailable in context-free grammar approaches. This constitutes one of the main motivations of using CG in this work.

Note that the development of the Wolof grammar follows in many respects Maxwell and Kaplan's (1993) model, according to which parsing time can be speeded up if conditions on certain finite-valued syntactic features are translated from f-structure constraints to variant c-structure categories. This means that the constraints can be enforced by the polynomial context-free c-structure system and not by the possibly exponential f-structure satisfiability algorithm. This works particularly well for features that can be evaluated fairly locally in the tree. For instance, the Wolof grammar uses parameterized c-structure categories (also known as complex categories) (Crouch *et al.* 2013; Butt *et al.* 1999) provided by XLE as a way of systematically propagating and enforcing features that provide subclasses of context-free categories. However, while this approach is beneficial as it provides the means to prune inconsistent analyses early (i.e., in the chart building phase instead of the unification phase), it does not provide the same gain in efficiency as the separate CG component does. One of the reasons is that the use of parameterised c-structure categories also increases the number of categories which are built in the XLE chart. A further, and perhaps more important, reason is that, with the CG-based approach, XLE will not even try to build c-structure for the undesired analyses, as these readings are removed at earlier stages.

Accordingly, morphological disambiguation based on CG has been incorporated into the Wolof grammar. The implementation of the CG model used for Wolof is developed by Didriksen (2003) within the VISL NLP framework,[12] and is based on the third-generation compiler *vislcg3* (Bick 2000). As the Wolof CG disambiguator is discussed in details in Dione (2014), I will here only briefly outline the use of the CG model to handle lexical ambiguity.

To illustrate how CG-based disambiguation works for Wolof, let us consider Example (33). In order to remove undesired readings for this input sentence, a number of detailed constraints have been developed. Some of these are exemplified in the rules in (34)–(36), which are written in accordance with the CG-3 compiler documentation.[13]

In (33), a large number of ambiguities can be resolved by looking at the Wolof noun class agreement. For instance, specifiers such as determiners or demonstratives and modifiers such as relative pronouns agree with the head noun. Accordingly, those analysis lines in (33) which contain a noun class tag that does not occur in the analysis line of the adjacent noun can be removed. This means, for example, that the determiner reading of *la* can be safely removed: It refers to the *l* class which differs from the possible classes for the noun *fas*, which can take either the *g* or the *w* index. This is accomplished by the constraint rule in (34).

(34)
```
REMOVE (Det Def) + $$NC
IF (NEGATE -1 NOM + $$NC)
    (NEGATE *-1 Pron + $$NC BARRIER CLB);
```

- `REMOVE (Det Def) + $$NC`: remove a definite determiner with a noun class index, `IF`
  - `(NEGATE -1 NOM + $$NC)`: there is no nominal (`NOM`), with the same class, occurring immediately to the left (`-1`).
  - `(NEGATE *-1 Pron + $$NC BARRIER CLB)`: there is no pronoun with the same noun class anywhere (`*`) to the left of the first neighboring position, and there is no clause boundary (`CLB`) in between (`BARRIER`).

[12] See `http://beta.visl.sdu.dk`.

[13] See Bick (2009) and `http://beta.visl.sdu.dk/cg3.html`.

Likewise, the relative pronoun reading can be removed using a rule similar to (34). In addition, relative pronouns can be directly removed in a more general context, for example, if the right adjacent constituent is a prepositional phrase, a conjunction or a punctuation symbol ('.', parenthesis, etc.).

The rule in (35) removes the non-subject copular reading, depending on the part of speech of the left adjacent and right adjacent word.

```
(35)  REMOVE (Icop)       IF  (-1 Verb LINK 2 Verb);
```

- `REMOVE (Icop)`: remove a copular reading, `IF`
  - `(-1 Verb LINK 2 Verb)`: left adjacent and right adjacent words are verbs.

The rule in (36) removes *la* as a complementizer if an unambiguous (C) transitive verb occurs anywhere to the right from the first neighbouring position, and if there is no clause boundary in between.

```
(36)  REMOVE ("la" Comp)
      IF (*1C (Verb Trans) BARRIER CLB);
```

Having applied the rules in (34-36), only three analysis lines of *la* in (33) will be retained. While many local ambiguities can be resolved using the given rules, in some cases it is difficult to fully disambiguate. For example, the disambiguation of free relative pronouns and object clitics requires a careful rule design. With respect to the example discussed so far, in the current Wolof CG disambiguator, the surviving analysis lines may remain undisambiguated.

The Wolof CG disambiguator consists of a modest size of rules (ca. 250 rules), but is relatively effective. Applied on the Wolof test data (Cissé 1994; Garros 1997; Ba 2007), it helped to reduce the average numbers of readings per token from 2.69 to 1.55. The Wolof CG disambiguator is evaluated along with the c-structure pruning mechanism which has been used to tackle some issues of syntactic ambiguity, as discussed in Section 6.1.

6 SYNTACTIC DISAMBIGUATION

The simplest method of reducing syntactic ambiguity would be to write more restrictive rules. In some cases, it could be possible to find a restriction that rules out exactly the undesired analyses, for example,

by disallowing attachment of some PPs to the sentence level in ambiguity involving PP attachment. This strategy is obviously not always possible, as it may lead to incorrect analyses (e.g., of some PPs) or eliminate analyses containing particular ambiguities (e.g., global ambiguities) that need to be preserved. Accordingly, structural or scoping ambiguities have often been dealt with by ranking the different analyses, using either statistical models or linguistic intuition.

To deal with syntactic ambiguity in Wolof, I have explored various disambiguation models, including probabilistic as well as non-statistical ones. The former build upon the c-structure pruning mechanism of XLE (Cahill *et al.* 2007, 2008; Crouch *et al.* 2013), while the latter are based on optimality marks (Frank *et al.* 2001). In addition, I adopt ambiguity preserving approaches for constructions involving global ambiguity. These different approaches are discussed in the following sections.

6.1 *Coping with structural ambiguity by using c-structure pruning*

In Section 3.2.1, structural ambiguities in Wolof have been discussed. In Example (18), the word form *bi* can have different grammatical categories. For instance, it can be a determiner or a relative pronoun, leading to different c-structures, for example, for the constituent *xale bi moom*, which can be analyzed as a DP or as an NP with an embedded relative clause. The probability that this constituent occurs as a relative NP in some given texts is lower than the probability that the same constituent occurs as a DP. Similar facts can be noted about the constituent *doon ree*, which, in principle, is much more likely to be an auxiliary VP (VPaux) than a copular VP (VPcopmain). A probabilistic grammar takes these probabilities into account in a way that a non-probabilistic grammar does not. Accordingly, it is possible to assign a probability to a sentence, and base a given analysis of the sentence (from the set of possible analyses) on the probability associated with it.

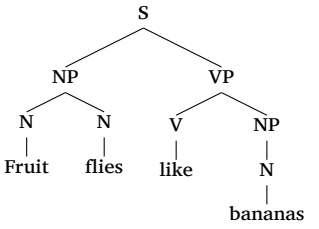
Thus, to deal with structural ambiguities such as those discussed in Section 3.2.1, I have conducted various experiments based on the c-structure pruning mechanism of XLE (Cahill *et al.* 2007, 2008; Crouch *et al.* 2013), in combination with CG during the development of the Wolof grammar. The experiments are extensively discussed in Dione (2014). In the following, I will outline the main aspects and results achieved by using this approach.

The c-structure pruning mechanism of XLE provides a possible method to control structural ambiguities and to make parsing faster by discarding low-probability c-structures before functional annotations (f-annotations)[14] are solved. Typically, XLE parses a sentence in a series of passes (Crouch *et al.* 2013). First, the morphology analyzes the sentence, looks up each morpheme in the lexicon and initializes a chart with the morphemes and their constraints. Then, the chart builds all possible constituents out of the morphemes using the c-structure rules given in the grammar. Constraints are processed after all of the constituents have been built. Next, the unifier processes the constraints bottom up, only visits those constituents that are part of a tree with the correct root category that covers the sentence, and builds a constraint graph for each subtree. Subsequent passes are concerned with finding locally incomplete analyses and solving the Boolean satisfaction problem for edges. One main reason for using c-structure pruning is that unification is typically the most computation-intensive part of LFG parsing. This is particularly true for Wolof. The typical proportions of overall runtime of some XLE components with the Wolof grammar are: Morphology (0.1%), Chart (3.1%) and Unifier (85.5%).

The basic procedure of testing the c-structure pruning mechanism consists in training a probabilistic context-free grammar (PCFG) on a corpus annotated with syntactic bracketing, and, subsequently, discarding all c-structures that are *n* times less probable than the most probable c-structure. Context-free rewrite rules typically consist of one non-terminal symbol on the left-hand side and a combination of terminal and/or non-terminal symbols on the right-hand side. XLE grammar rules are context-free rules augmented with f-annotations. Examples of PCFGs are given in Figures 3–4, which represent two different analyses of the sentence “Fruit flies like bananas”.

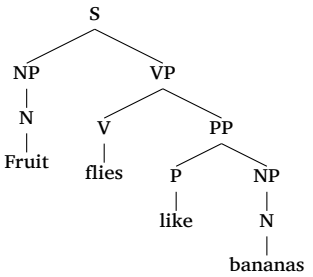
As can be seen in the Figures 3–4, each c-structure has hypothetical probabilities attached to it: 8.4375E-14 and 4.21875E-12 for Analysis 1 and Analysis 2, respectively. Accordingly, Analysis 1 is 50 times less probable than Analysis 2. Thus, depending on how the c-

[14] Functional annotations refer to the set of f-structure constraints associated with the analysis of a sentence. For example, the constraint (f TENSE) = PAST specifies that the feature TENSE in the f-structure f has the value PAST.



S	→	NP VP	0.5000
NP	→	N N	0.1500
N	→	Fruit	0.0010
N	→	flies	0.0015
VP	→	V NP	0.2000
V	→	like	0.0050
NP	→	N	0.5000
N	→	bananas	0.0015
			8.4375E-14

Figure 3:
Analysis (1) for the string *Fruit flies like bananas* with hypothetical probabilities



S	→	NP VP	0.5000
NP	→	N	0.5000
N	→	Fruit	0.0010
VP	→	V PP	0.1000
V	→	flies	0.0025
P	→	like	0.0500
PP	→	P NP	0.9000
NP	→	bananas	0.0015
			4.21875E-12

Figure 4:
Analysis (2) for the string *Fruit flies like bananas* with hypothetical probabilities

structure pruning mechanism is set, Analysis 1 may be discarded even before corresponding f-annotations are solved.

The probabilities for the rule can be estimated as relative frequencies found in a parsed (and disambiguated) corpus.[15] With these estimations, XLE makes use of a chart-based mechanism to prune subtrees at the level of individual constituents in the chart. A subtree is

[15] See Crouch *et al.* (2013) on how XLE computes the rule probabilities.

pruned if its probability is lower than the best probability by a given factor. For that purpose, the grammar writer can specify a so-called **cutoff** value (typically between 4 and 10), which corresponds to the natural logarithm of that factor. For instance, a value of 5 means that a subtree will be pruned if its probability is about a factor of 150 less than the best probability.

To test the c-structure pruning mechanism for Wolof,[16] a PCFG was built, trained and tested in two different ways: (i) only using the regular Wolof grammar without CG-based disambiguation, and (ii) using the CG parser (see Section 3.1.3) for morphological disambiguation. This had the purpose of evaluating the parsing system in terms of parsing time, accuracy and ambiguity reduction. The LFG metric used to measure the parsing quality is based on the comparison of full f-structures, represented as *relation(predicate,argument)* triples. Accordingly, the triples of the system are compared to a triple-based gold standard manually built for this purpose. For each comparison, the best match, that is, the reading that comes closest to the intended analysis (out of all source analyses) is chosen. The metric, referred to as the **oracle f-score** is defined as the geometrical mean of precision and recall (i.e., $F = (2 * P * R)/(P + R)$) which is calculated from the set of the triples in best match solution.

The results of applying the c-structure pruning mechanism on the development set, as reported by Dione (2014), show that a cutoff of 10 seems to provide the best trade-off between time and accuracy, if the LFG parsing is not combined with CG.[17] Otherwise, if CG-based disambiguation is used in addition to c-structure pruning, a cutoff of 9 seems to perform best on the development data. Having established the best cutoff values for the two training forms, the c-structure pruning mechanism is applied to the Wolof test set.

The results on the test set are given in Table 2. These show that c-structure pruning and CG-based disambiguation, independently, yield a great reduction in parsing time. Using only the c-structure pruning (with a cutoff of 10) leads to a speed-up over 36%. If the test set is disambiguated using CG, a cutoff value of 9 allows for a speed-up of

[16] See Dione (2014) for details about the experiments, the training data, and on how the gold standard data have been built.

[17] For a discussion of how the pruning algorithm is trained on the Wolof data and the process used to establish the best cutoff values, see Dione (2014).

30%. Using only CG-based disambiguation, parsing efficiency can be improved by ca. 40%. In total, combining c-structure pruning with CG-based disambiguation leads to a speed-up of 58%.

	Without CG		With CG	
Pruning Level	None	10	None	9
Total CPU Time (sec)	7374	4779	4473	3164
Oracle f-score	93.02	92.05	90.52	89.40
# Full Parses	1712	1613	1551	1434
# Fragment Parses	627	737	775	917
# Time Outs	10	5	8	6
# Skimmed Sentences	348	240	191	125

Table 2:
Results of the c-structure pruning experiments on Wolof test data

However, as can be seen in Table 2, this increase in speed leads to a relatively significant drop in f-score. The c-structure pruning and CG-based disambiguation, independently, have a negative impact on the quality of the f-structure: The number of fragment parses increases.[18] Without CG-based disambiguation, a cutoff of 10 leads to a drop in f-score of 0.97 points. CG pre-filtering without c-structure pruning causes a drop in f-score of 2.5 points. Using CG-based disambiguation and a cutoff of 9, the f-score decreases by 1.12 points. In total, combining c-structure pruning (with a threshold of 9) with CG-based disambiguation results in a drop in f-score of 3.62 points.

		Pruning Cutoff	Ambiguity Rate	Ambiguity Reduction
1	w/o CG	None	209.77	72.92%
		10	56.81	
2	w/o CG	None	174.38	77.64%
	with CG	None	56.45	
3	w/o CG	None	154.66	80.66%
	with CG	9	29.92	

Table 3:
Ambiguity reduction when using c-structure pruning and CG-based disambiguation

Table 3 shows the ambiguity reduction achieved by using the c-structure pruning algorithm and CG. Because the ambiguity rate was

[18] Fragments are produced when the grammar is unable to provide a full parse for the input sentence. This partial parsing technique allows the sentence to be analyzed as a sequence of well-formed chunks with both c-structure and f-structure associated with them. Similarly, skimmed parses are produced, when the amount of time or memory spent on a sentence exceeds a threshold. This technique is used to avoid time-out and memory problems.

measured relative to the common full parse solutions produced by the specific test run, the values for ambiguity rate are not absolute, but rather relative values. Combining c-structure pruning with CG-based disambiguation (Row 3) provides the best results with over 80% ambiguity reduction.

While statistical disambiguation is convenient if a corpus annotated with syntactic bracketing exists, it is also a source of errors, which are often caused by a lack of data. Also, the application of this disambiguation technique may be inappropriate in some cases. On the one hand, c-structure pruning will not often be able to disambiguate between two constructions if they are both very frequent in the corpus data. For example, in Wolof, constructions involving asyndetic coordination might be undesired in many cases. They do, however, have a relatively high frequency in the data, so that a statistical disambiguator will not readily prune them, and even if it did, this would often result in incorrect analyses. On the other hand, the c-structure pruning mechanism cannot be used to manage some syntactic ambiguities like those discussed in Section 3.1.4, which involve the f-structure rather than the c-structure. Thus, managing such ambiguity might require the use of non-statistical mechanisms such as optimality marking (Frank *et al.* 1998, 2001).

6.2 *Using optimality marks*

When dealing with syntactic ambiguities, humans can make use of extra-linguistic knowledge and context. Parsers, however, have only the grammar as a knowledge base and they deliver all possible solutions, including potentially many implausible ones. This might adversely affect parsing efficiency, often making a manual correction of the output necessary. In this respect, a possible method to constrain these ambiguities to legitimate cases and to indicate a preference for one syntactic analysis over another is the use of the formal mechanism based on optimality marks (Frank *et al.* 1998, 2001).

Optimality Theory (OT) was first developed by Prince and Smolensky (1993) for phonology, and later extended to other areas such as syntax and semantics. Theoretical OT models grammars as systems that provide mappings from inputs to outputs; the inputs are viewed as underlying representations and the competing output candidates (or analyses) as their surface realizations. Accordingly, grammars are

seen as having a set of violable constraints. The constraints are universal and ranked by each language, giving rise to cross-linguistic variation. Constraint ranking determines the winning candidate, that is, the candidate that incurs fewer violations than all other candidates.

For example, given constraints C1, C2, and C3, where C1 dominates C2, which dominates C3, A is optimal if it outperforms B on the highest ranking constraint which assigns them a different number of violations. If A and B tie on C1, but A does better than B on C2, A is optimal, even if A has 100 more violations of C3 than B. Table 4 shows an example of the standard table notation for OT analyses.

Table 4: Example of the standard table notation in OT

	/**Input**/	CONSTRAINT 1	CONSTRAINT 2	CONSTRAINT 3
☞	Candidate A	*	*	***
	Candidate B	*	**!	

The optimal candidate is highlighted with a pointing finger in the tableau, and each cell displays an asterisk for each violation for a given candidate and constraint. Once a candidate does worse than another candidate on the highest ranking constraint distinguishing them, it incurs a fatal violation, resulting in the elimination of the candidate (marked in the tableau by an exclamation mark '!'). Once a candidate incurs a crucial violation, there is no way for it to be optimal, even if it outperforms the other candidates on the rest of the universal constraint set.

The OT model has been adopted and extended within the LFG framework for ranking preferences and constraints. Two fundamental aspects in the extension of the OT model in LFG can be described as follows:

- The model used in LFG is a violable constraint system used as a preference filter on analyses. The possible analyses are ranked using this preference filter, which does not necessarily rule out sub-optimal structures entirely;
- The violation of a constraint is not always negative. There are positive constraints, whose fulfillment is desired in some context.

In LFG, OT is an additional projection (*o*-projection or *o*-structure) formally defined as a multiset of constants (constraints or 'optimal-

ity marks'). The constraints are projected from c-structure (Frank *et al.* 1998) and are introduced by *o*-descriptions within the grammar. The *o*-structure serves as a record of constraints used for each candidate analysis.[19] The XLE grammar development environment provides an implementation of OT in LFG, incorporating the idea of ranking and (dis)preference. This utility allows for filtering syntactic and lexical ambiguities in a way that aims to reconcile robustness and accuracy.

Unlike theoretical OT which only includes dispreference marks, XLE OT defines both preference and dispreference marks. Preference marks come to use when one specific reading out of a set of analyses is preferred. In general, they allow one to mark more frequent structures, which are preferred to the less frequent ones. Example (37) (from Frank *et al.* 1998, pp. 5) illustrates the use of such marks to state a preference for multiword terms in technical documentation.

(37) a. I want [print quality] images.
b. *I want [print] [quality] images.

With a respective preference mark, the analysis with the multiword expression (37a) will be preferred over all other readings. However, if there is no valid analysis for the multiword expression (as in 37b), an analysis using the individual lexicon entries is still possible.

Dispreference marks are used for rare grammatical constructions which need to be covered, but interact in unexpected ways with frequent constructions, making them 'dispreferred'. For example, these marks may be used to exclude NPs being headed by adjectives from the candidate set. Dispreferred constructions are selected only when no other, more plausible, analysis is possible. Yet, it can be difficult, in general, to decide whether to use a preference or dispreference mark. The difficulty stems mainly from two main issues: (i) whether there is any interaction between the marks, and (ii) which analysis is easier to mark. For instance, it is easier to mark a multiword expression with a preference than to mark all of its components with a dispreference.

In addition, XLE provides other special marks such as STOPPOINTs. STOPPOINT marks slowly increase the search space of the

[19] Note that the *o*-structure is just a set of marks, not of f-structure features. The f-structure of the grammar remains unaltered. Optimality marks are in their own projection, the extra representation level referred to as the *o*-structure.

grammar if no good solution can be found. They constitute a way of increasing the robustness of the grammar without sacrificing performance. For instance, in the OT field of the configuration, STOPPOINT marks can be inserted into the hierarchy of preference and dispreference marks (e.g., to the right or left of STOPPOINT in optimality order). As such, dispreferred constructions like rare or computationally expensive constructions will only be considered if the core grammar fails to find a valid analysis for frequent ones.

In the context of Wolof, I have conducted experiments with the OT mechanism following two main objectives: (i) to select preferred analyses for ideophones (see Section 5.2) and ambiguous verb subcategorization frames (see Section 6.3), and (ii) to increase robustness by managing ambiguity caused by computationally expensive constructions like coordination without an overt conjunction (see Section 6.4).

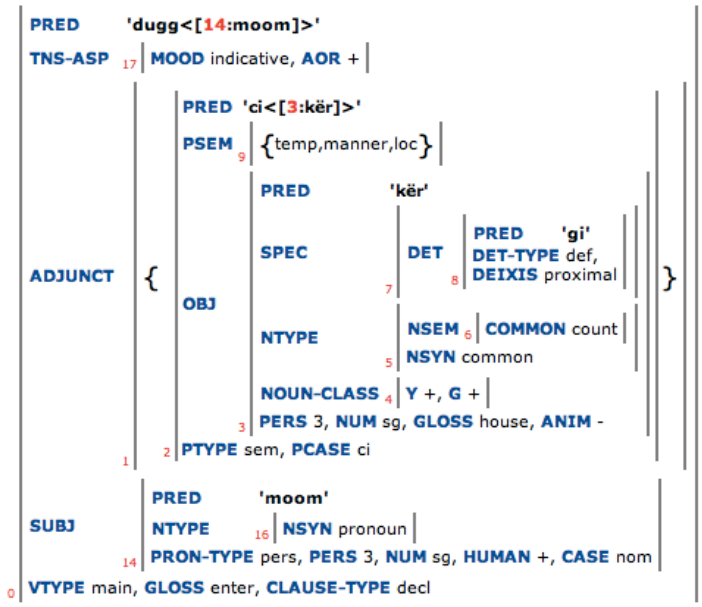
6.3 *Managing ambiguity caused by subcategorization frames*

Section 3.1.4 discussed verbs in Wolof like *dugg* "enter" in (17), repeated in (38), which have several subcategorization frames. These include a bare intransitive and an oblique reading, as illustrated by Figures 5 and 6, respectively.

(38) *Mu dugg ci kër gi.*
3sg enter in house cl.DFP
"S/He entered the house."

For Wolof, I have attempted to suppress ambiguity caused by subcategorization frames through the use of optimality marks. Accordingly, I have introduced preference marks in the grammar to help indicate the preferred reading in (38), for example, to select the oblique reading over the adjunct one. As Example (39) shows, the OT constraints are used within a disjunction at the level of functional annotation. This example specifies that a Wolof verbal phrase (VP) may expand into a verb V followed by an optional determiner phrase (DP) and several prepositional phrases (PP*). A PP may be realized either as an oblique argument or an adjunct, and each choice is marked with an *o*-projection mark for preference ranking. The disjunction under PP in rule (39) is a typical source of optimality marks for sentence (38).

Figure 5: Analysis of *dugg* as an intransitive verb



(39) VP → V (DP, (↑ OBJ)=↓) { PP*, (↑ OBL-TH) = ↓, MARK1 ∈ *o**, ↓ ∈ (↑ ADJUNCT), MARK2 ∈ *o** }

However, in the context of Wolof, the use of the OT mechanism encounters some essential problems. For instance, the use of preference constraints was frequently faced with exceptions and counterexamples. There are still cases where OT chooses the wrong structure. By way of example, let us consider the sentences in (40).

(40) a. *Mu tontu ca laaj ba.*
3sg reply PREP question cl.DIST
"So, (s)he replies to the question."

b. *Mu tontu ca saa sa.*
3sg reply PREP instant cl.DIST
"So, (s)he replies immediately."

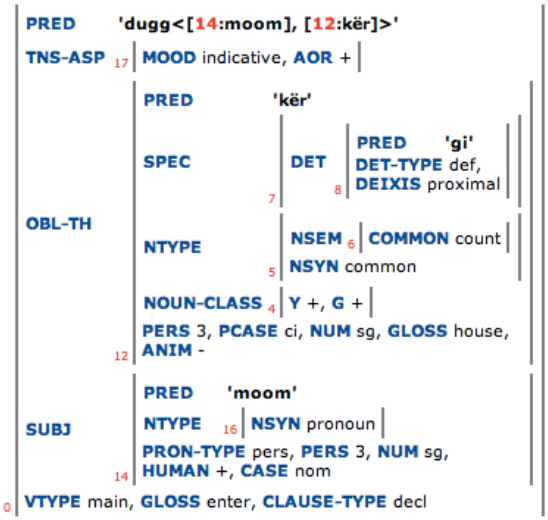


Figure 6:
Analysis of *dugg* as a verb with an oblique argument

These sentences contain the verb *tontu*, which typically selects for the prepositions *ci* and *ca*, meaning, inter alia, 'on', 'to', and 'about' according to the context. Accordingly, the PP *ca laaj ba* in (40a) can be assumed to be subcategorized for by the verb, and therefore bears an argument function (specifically an oblique function) within the clause. As this kind of construction is quite common for this verb, a preference mark for PP obliques was introduced in the Wolof grammar in the early stage of grammar development. This aimed to automatically suppress certain ambiguities due to the adjunct reading. However, as exemplified by (40b), this approach is not successful in all contexts. In (40b), the prepositional phrase *ca saa sa* appears as a modifier rather than an argument of the verb. It modifies the verb *tontu*, but it is not governed by this predicate. Hence, a preference mark for oblique PPs over adjuncts will falsely choose the oblique reading for the PP *ca saa sa* in the sentence (40b).

A similar situation can be observed in English: The preference of oblique PPs over adjuncts may lead to an incorrect analysis of the constituent *for two hours* in the sentence *John waited for two hours*. As Copperman and Segond (1996, pp. 6) pointed out, it is difficult

to find in the literature "a discussion of preferring arguments to adjuncts (via subcategorization frames), which strikes us a valid general preference. Of course, in many cases the question of whether to consider something an argument or an adjunct is no more solved than PP attachment, so this will actually help only in clear cases."

Another disadvantage of this mechanism is that the use of optimality marks requires careful adjusting and experimenting to get certain effects, both in terms of preferences and performance. This reflects the fact that the OT specifications have global interactions and are thus difficult to describe. For instance, for a grammar writer, it may be very difficult to introduce new marks or reorder old ones to get the relatively straightforward outcomes that (s)he is looking for. The indirect consequences of minor adjustments can be hard to understand and predict. Thus, with regard to the Wolof grammar, the idea of using preference marks for PP obliques was abandoned due to the large number of counterexamples.

6.4 *Handling coordination ambiguity*

As discussed in Section 3.2.1, Example (18), repeated in (41),[20] also illustrates asyndetic coordination (i.e., coordinated structures without an explicit conjunction). Such structures are frequently encountered in the Wolof data. A typical syntactic feature of these coordinate structures is that they may exhibit *forward conjunction reduction* (Kempen 1991) involving a subject gap: The subject of the left conjunct is omitted from the second clause and understood to be identical to the first clause's subject. In LFG terms, the fact that the second conjunct seems to be missing a subject raises a particular issue with regard to Completeness (Kaplan and Bresnan 1982): All the governable grammatical functions required by the PRED of the f-structure should have a value in the f-structure.

(41) Xale b-i moom doon ree.
child cl-DFP own IPF.PST laugh
"The child owns (something) and was laughing."

To handle this kind of coordination in Wolof, rules like (42) are used. These allow phrases of same constituents to be coordinated. In

[20] The gloss and translation only retain the reading as asyndetic coordination, which is the relevant one for the discussion.

the associated f-structure, the coordinate phrase is represented as a set-valued f-structure. Each of the conjuncts is represented as an element within the set by the functional annotations $\downarrow \in \uparrow$. To solve the Completeness problem, the symmetric analysis with *asymmetric grammaticalised discourse function (GDF) projection* proposed in Frank (2002) for German subject-gap constructions is adopted for Wolof. For example, (42) defines symmetric S coordination in c-structure, with symmetric projection of the conjunct's f-structures in terms of the classical $\downarrow \in \uparrow$ annotations. The annotation $(\uparrow \text{SUBJ}) = (\downarrow \text{SUBJ})$ defines the first conjunction's subject as the subject of the coordination as a whole.[21] The predicate *e* matches against the empty coordinating conjunction string.[22] The feature (↑ COORD-FORM) specifies the form of the conjunction (e.g., *and* or *or*). In this rule, the form is assumed to be null, since the conjunction is not overtly realized. The annotation *(↑ COORD)* = + indicates that the whole structure is a coordinate phrase.

(42) SCoord → { S: ↓ ∈ ↑ (↑ SUBJ) = (↓ SUBJ);
e: (↑ COORD-FORM) = null (↑ COORD) = +
CWCONJ ∈ o*;
S: ↓ ∈ ↑
}.

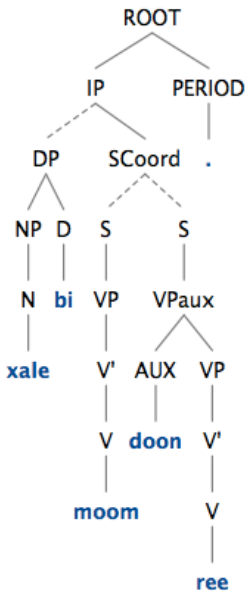
Figure 7 shows the c-structure related to the reading of the sentence (41) as an instance coordination without an explicit conjunction.

In the rule in (42), the annotation *CWCONJ* ∈ *o** (in XLE notation: "CWCONJ $ o::*") says that (i) *CWCONJ* is a member of the OT projection; (ii) such a structure is coordinated without an explicit conjunct; and (iii) it should be dispreferred. Allowing any S, IP or NP constituent to be coordinated with another constituent of the same category without an explicit conjunction poses notorious ambiguity and performance problems: It generates a great number of parse possibilities, and sometimes leads to memory, time-out and coverage problems. Thus, the ambiguity caused by this kind of construction had to be addressed during grammar development.

[21] The (↑ SUBJ) = (↓ SUBJ) equation is only needed for asymmetric case where there is no analysis in which the subject can be construed to be outside of the two conjuncts, in which case normal distribution over sets will take care of this.

[22] The symbol *e* is the "epsilon" symbol for an LFG grammar.

Figure 7:
C-structure analysis of coordinated sentences without an explicit conjunction



Currently, the Wolof grammar handles ambiguity caused by coordination without an explicit conjunction by using the STOPPPOINT mark. For performance reasons, all rules dealing with coordination like (42) (ca. 10 rules) are annotated with the OT dispreference mark *CWCONJ*. In the OT configuration of the grammar, *CWCONJ* is inserted to the left of the STOPPOINT mark to consider this expensive construction only when no other analysis is available.

To measure the impact of using this approach, the grammar was run on the test set with and without the application of STOPPOINT. The test runs reveal that the use of the STOPPOINT mark increases the parsing time by 6%. In fact, the approach does not lead to effective advantages in terms of efficiency because XLE has to parse each sentence in several passes. This explains the slight increase in parsing time. However, this approach pays off very well in terms of ambiguity reduction: The comparison of the number of solutions produced by each run reveals that it reduces the ambiguity rate by a factor of 5–6. However, with this approach, there is also a decrease in the pars-

ing quality: In the test set, about 25% of the desired interpretations (relative to coordination without an explicit conjunction) were also eliminated, causing a drop in f-score of ca. 0.94 points.

6.5 *Preserving ambiguity due to pro-drop and impersonal passive*

Unlike constructions discussed in the previous sections, where the main goal was to remove some readings, many syntactically ambiguous utterances can be parsed and assigned ambiguous structures. Section 3.3 discussed constructions that exhibit global ambiguity due to pro-drop and impersonal passive, as illustrated in (19b), repeated in (43).

(43) *Gor* *na-ñu* *garab g-i.*
cut.down +F-3pl tree cl-DFP
"They cut down the tree."
"The tree was cut down."

As other types of ambiguity, referential and lexical ambiguity can interact, resulting in global ambiguity. The referential ambiguity is raised by the subject marker *nañu*, which implies that the subject is a non-arbitrary referential subject or an arbitrary subject used as a third person plural person in impersonal passive constructions. As with ambiguity due to subcategorization frames, the phrase structure in (43) is the same in each case, but the difference lies in the form of the semantic predicates. Thus, while the sentence has a single c-structure, its semantic structure is ambiguous.

The f-structure analysis for the first reading is shown in Figure 8. This analysis follows the standard LFG treatment of pro-drop, in which the verb specifies that its subject has the PRED value 'pro'. In the impersonal construction, however, the subject PRED and PRON-TYPE are assumed to be null in order to reanalyze the null-subject construction as an arbitrary reading. The analysis for the second reading is given in Figure 9.

As an instance of global ambiguity, the sentence in (43) is not disambiguated at the parsing level. The solution considered is to leave the decision to users, which are accustomed to resolving many types of ambiguity in texts subconsciously and efficiently using common-sense knowledge. Without the common-sense knowledge that is necessary to resolve this kind of ambiguity, the parser is not expected to know

Figure 8:
F-structure analysis of sentence (19b) as an instance of pro-drop constructions

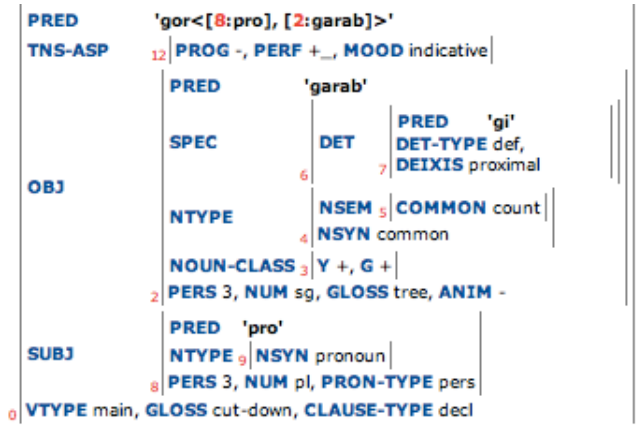
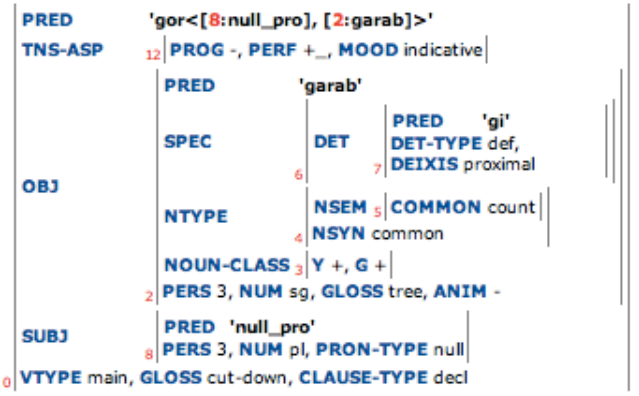


Figure 9:
F-structure of sentence (19b) analyzed as a passivized sentence with a null subject



which of the solutions that it generates can be left ambiguous because they will be disambiguated correctly by the user or because the consequences of misinterpretation are trivial, and which will be genuinely problematic (Chantree 2004). Consequently, automatic resolution of global ambiguity is mostly not desired. It can be very beneficial to allow this kind of ambiguities to remain in the text and to be resolved by users at a later stage (i.e., in post-parsing). For such a task, there are grammar engineering tools available which provide a representation of a set of ambiguous f-structures in a single, packed structure, allow-

ing (non-expert) users to locate specific solutions among the output set straightforwardly and efficiently (e.g., using discriminants).

7 GRAMMAR ENGINEERING AND AMBIGUITY

7.1 *Ambiguity packing in XLE*

To facilitate ambiguity management, XLE provides a built-in utility for grouping and displaying packed representations of the alternative solutions (King *et al.* 2004). The utility is based on an efficient algorithm for contexted constraint satisfaction that processes ambiguities in a chart-like packed representation (Maxwell III and Kaplan 1996).

Consider Example (43) discussed in Section 6.5. As this ambiguity is global and linguistically appropriate, it will normally be computed and preserved. Accordingly, in the lexicon, the entry for the form *nañu* is provided with two semantic PREDs encoded as disjunctive statements to allow for the two readings of this word. These alternative solutions logically lead to at least two different f-structures. However, with the XLE built-in algorithm for contexted constraint satisfaction, such disjunctive facts are not compiled out and duplicated. The algorithm rather produces a representation as a set of the ambiguous f-structures in a single, packed f-structure, also called f-structure chart, as Figure 10 illustrates.

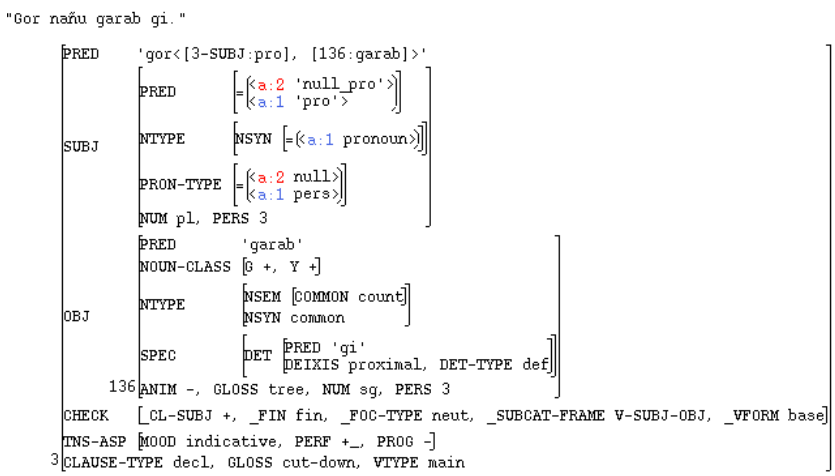


Figure 10: F-structure chart for packed ambiguities in XLE

The f-structure chart window provides a list of choices that are caused by alternative solutions. Hence, this sentence has two analyses,

identical except for the values of the PRED (which may be 'pro' or 'null_pro'), pronoun and noun type features. In the packed f-structure chart, attribute-values are indexed with their corresponding context variables, meaning that the two values are displayed as alternatives, labeled with indices a:1 and a:2.

This XLE tool for ambiguity management helps grammar developers to determine the source of the multiple solutions produced by the parser. As Attia (2008, pp. 221) pointed out, "grouping the solutions in packed representations can effectively speed up the process of detection and revision". For instance, given that the choices in the window in Figure 10 are active, the user can click on a choice and have a solution corresponding to it displayed in the c-structure and f-structure windows. Thus, the use of this facility can avoid the need for the grammar developers to search through the parse forest by examining one solution after the other.

Equally important, the tool provides grammar writers with information on the existence of spurious ambiguities. For instance, vacuous ambiguity of two f-structures, for example, resulting from duplicate lexicon entries for the same word, appears in a specific form (namely, as blank choices) indicating that there is a spurious ambiguity in the grammar with respect to the given sentence. One of the best ways to avoid such vacuous ambiguities is to check the grammar carefully and to make disjunctions exclusive. The elimination of spurious ambiguities proves to be a very effective mechanism for increasing parsing efficiency.

7.2 *Removing spurious ambiguities*

Spurious ambiguities as duplicated solutions may arise from different sources, including the morphology, the lexicon, the c-structure and the f-structure. For instance, c-structures may be duplicated if there are two entries under a word for the same category. This particular problem may also arise when there is no obvious disjunction (Crouch *et al.* 2013).

Disjunctions are the alternative paths that a rule can take. "While disjunctive statements of linguistic constraints allow for a transparent and modular specification of linguistic generalizations, the resolution of disjunctive feature constraint systems is expensive, in the worst case exponential" (King *et al.* 2000, pp. 7). If disjunctions are not clearly

defined in order to be mutually exclusive, they can lead to overgeneration. As the sentence length grows, spurious ambiguity can cause an exponential growth in the number of generated solutions.

Thus, grammar writers need to investigate methods of eliminating spurious ambiguities, for example, by verifying that disjunctions in the grammar are mutually exclusive. For example, if an NP's f-description contains the disjuncts in (44), then this NP is required to receive a nominative case value or a third person value. However, the disjunction in (44) is not mutually exclusive, since both can be satisfied at the same time. A good way to avoid spurious ambiguity in this case is to make the disjunction explicit and mutually exclusive, as shown in (45). While in the first disjunct in (45) the attribute person must have a value other than 3, hence the annotation (↑ PERS) ~= 3, in the second disjunct in this example a third person value is required; thus the two disjuncts cannot be satisfied at the same time.

(44) {(↑ CASE) $=_c$ nom | (↑ PERS) $=_c$ 3 }

(45) {(↑ CASE) $=_c$ nom (↑ PERS) ~= 3 | (↑ PERS) $=_c$ 3 }

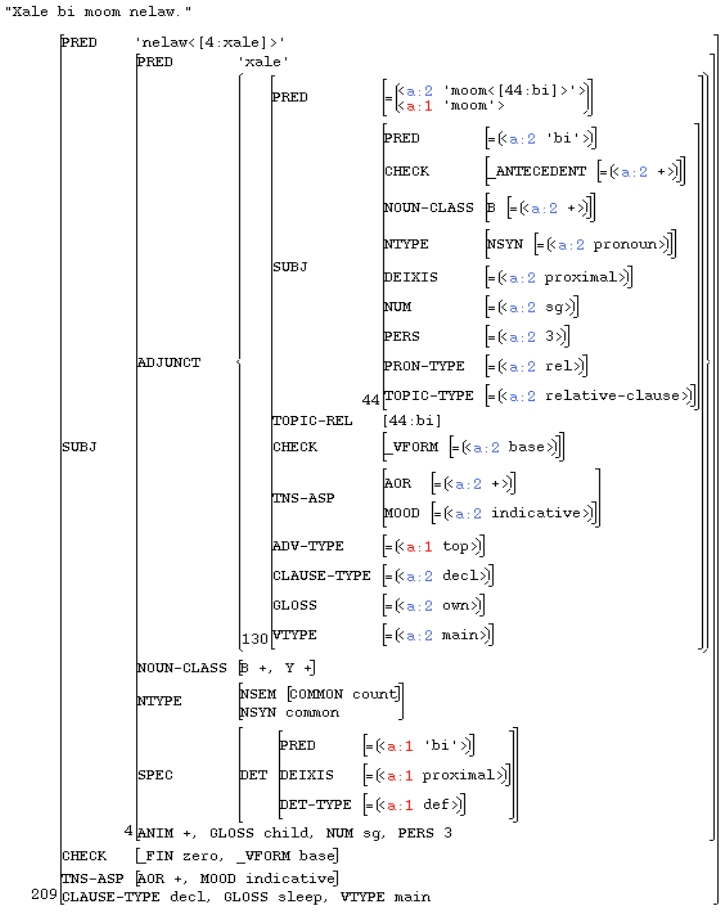
Accordingly, I have thoroughly checked the rules, templates (including verb subcategorization frames) and lexical entries in the Wolof grammar, in order to avoid as many duplicate solutions as possible. This careful review and redesign of the grammar has led to a considerable reduction of spurious ambiguities. Though it might seem like a small detail, removing spurious ambiguities can lead to a great improvement in parsing efficiency: The decrease in parse time observed after taking this measure was more than 50%. Attia (2008, pp. 219) has made a similar observation with respect to the Arabic Grammar, stating that "changing the way a rule was written to avoid a non-exclusive disjunction led to a huge reduction in parse time by 68%. The number of subtrees was reduced then by approximately 10%." These common observations clearly show the effectiveness of spurious ambiguity elimination by making disjunctions mutually exclusive. However, the sources of such ambiguities (e.g., the fact that disjunctions such as the one in Example (44) are not mutually exclusive) are not really obvious for grammar developers and deserve consideration in grammar writing.

8 DISAMBIGUATION WITH DISCRIMINANTS

The facility for packing ambiguity provided by XLE is easy to use for disambiguation when there are only a few choices. However, in some contexts, there are many choices. For some other inputs, more than hundreds of analyses are produced. Such a context is illustrated by Example (46) and its related f-structure in Figure 11.

(46) *Xale bi moom nelaw.*
child the TOP.ADV sleep
"The child, for his part, sleeps."

Figure 11: Partial f-structure chart for sentence (46)



When dealing with sentences with many choices such as (46), using this facility requires expert competence in using XLE and detailed knowledge of the grammar (Rosén *et al.* 2005). Consequently, in addition to packing ambiguities, Rosén *et al.* (2005) have implemented discriminants for LFG to facilitate the disambiguation task.

Discriminants are defined as small independent choices which interact to create dozens of analyses (Carter 1997). The idea is based on Carter's (1997) argument "that disambiguation may be achieved quickly and without expert competence if it is based on elementary linguistic properties which the disambiguator may accept or reject independently of other properties" (Rosén *et al.* 2005, pp. 378). On this basis, Rosén *et al.* (2005) implement discriminants for LFG-based parsers, defining a discriminant in LFG terms as "any local property of a c-structure or f-structure that not all analyses share."

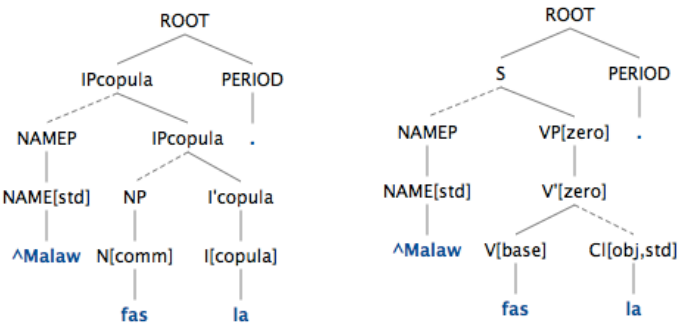
There are four major types of discriminants for LFG grammars (Rosén *et al.* 2007): lexical, morphological, c-structure and f-structure discriminants. Any given discriminant can induce a binary partition on the choice space. The selection of a discriminant (or its complement) amounts to the selection of one of the two partition elements, reducing the choice space accordingly.

The discriminant-based approach provides efficient and elegant support for LFG parse disambiguation. For instance, the Wolof treebank has been established by running test suites through the grammar. To disambiguate the outputs of the parser and to measure the parsing quality, the set of solutions returned by the parser must be manually reviewed, as no gold standard data are available for the language. However, the Wolof parser, like most parsing systems, produced a great number of solutions (tens or hundreds, sometimes even thousands of solutions). In such cases, reviewing the output by hand to see if the intended reading is in the set of the produced parses becomes a time-consuming, tedious and impractical task.

Therefore, I used a semi-automatic, incremental parsebanking approach based on lexical, morphological, c-structure and f-structure discriminants, in order to disambiguate the Wolof data in an efficient and elegant way. By way of illustration, let us consider Example (47). This sentence is associated with two c-structures displayed in Figure 12.

(47) *Malaw* ***fas*** *la.*
Malaw horse/amulet/to.tie COP.3SG/2SG.OBJ
"Malaw is a horse/an amulet." / "Malaw tied you."

Figure 12: Two possible c-structures for the sentence in (47)

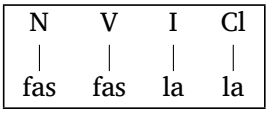


In this sentence, the word form *fas* may be either a verb (V) or a noun (N). Likewise, *la* may be a copular verb (I) or an object clitic (Cl). Because of this, there are two quite different c-structures, as shown in Figure 12. However, choosing the lexical category of either of these words is sufficient to determine which c-structure is the intended one; thus examining these c-structures is no longer necessary. Figure 13 illustrates *lexical discriminants* for the ambiguous words in (47). The traditional part of speech (e.g., N, V, I, Cl) is the lexical category specified in the discriminant. The relevant subtrees containing preterminal and terminal nodes for Example (47) are shown in Figure 14.

Figure 13: Representation of lexical discriminants for *fas* and *la*

'*fas*': V
'*fas*': N
'*la*': I
'*la*': Cl

Figure 14: Subtrees defining lexical discriminants for Example (47)



In (47), the word *fas* is ambiguous between different forms within the same POS (N). It can either mean 'horse' and therefore fits to

the noun class *w* or 'amulet' which belongs to the *g* class. This morphological ambiguity evidences the fact that, in some cases, lexical discriminants are not sufficient for disambiguation. For this reason, Rosén *et al.* (2007) decided to further define a morphological discriminant as "a word with the tags it receives from morphological preprocessing." A *morphological discriminant* for *fas* is illustrated in Figure 15.[23]

fas + Noun + Common + g + Inanim
fas + Noun + Common + w + Anim

Figure 15:
Morphological discriminants for *fas*

Discriminants can be displayed along c- and f-structures using the XLE Web Interface (XLE-Web),[24] as shown in Figure 16.

Originally developed in the TREPIL project (Rosén *et al.* 2009) and now in use for many of the ParGram grammars, the interface is a web-based tool for interactive sentence analysis with XLE. It allows to visualize the mapping from c- to f-structure, and to compactly display packed representations that combine the c- and f-structures of all analyses of a given parse into one c- and one f-structure graph.

Discriminants are presented in a user-friendly form with a sentence and all the parses identified by the parser. In Figure 16, the XLE-Web interface shows possible c- and f-structures for the sentence in (47) as well as lexical, morphological, and syntactic features, allowing binary choices for efficiently selecting the intended discriminant. This example has morphological and lexical discriminants that are reflected in the f-structure. The discriminants are the different values of the NOUN-CLASS, ANIM and GLOSS features. When a discriminant is selected, parses not consistent with that selection are removed from the choice space (and suppressed in the display). Discriminants are not completely independent. Some discriminants are redundant and others eliminate dependent discriminants when selected. Table 5 displays LFG discriminant statistics for the Wolof treebank.

[23] Note that this example refers to the initial analysis of Wolof noun classes prior to the underspecification approach discussed in Section 5.1.

[24] See `http://iness.uib.no/iness/`.

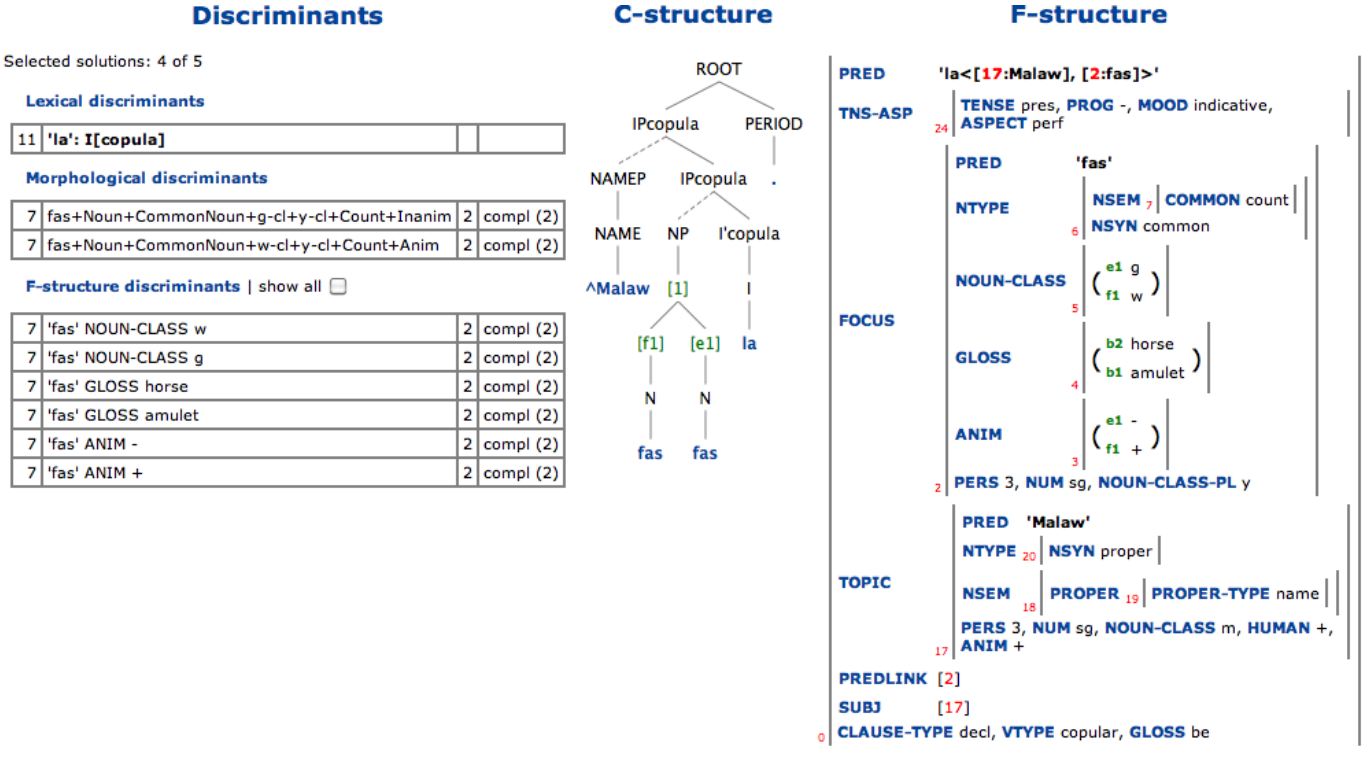


Figure 16: XLE-Web interface showing discriminants

Discriminant Type	Frequency
M: Morphological discriminant	522
L: Lexical discriminant	1432
C(R): C-structure rule discriminant	131
C(C): C-structure constituent discriminant	568
F: F-structure discriminant	606
total	3259

Table 5:
LFG discriminants statistics

9 CONCLUSION

This work shows that natural languages, with a particular focus on Wolof, are rich in ambiguities of many kinds. It also shows that the wide range of possible interpretations of natural languages and the interaction between the different ambiguity types pose a particular challenge for large-scale, linguistically motivated grammars. In the context of Wolof, the most productive sources of ambiguity in the grammar include noun class syncretism, the use of coverbal ideophones, lexically ambiguous words, lexical ambiguity due to subcategorization frames, structural ambiguity, coordination ambiguity, and ambiguity between pro-drop and impersonal passive constructions. Accordingly, I explored several ambiguity management approaches at various parsing levels. This includes systematic ways of dealing with ambiguity, CG-based disambiguation, c-structure pruning, the application of OT constraints, packing ambiguities and discriminant-based disambiguation.

Systematic disambiguation approaches involve classical ways of underspecification. With the assumption that Wolof nouns typically show no class distinction and often have forms that can be attributed 'indeterminately' to different noun classes, I applied an underspecification approach based on feature indeterminacy to the Wolof noun class system. Following this analysis, Wolof nouns were assigned a feature structure containing a noun class attribute whose value allows specification by means of a separate Boolean-valued attribute. The proposed approach correctly identifies the linguistic aspects triggered by the noun class attribute, allowing to substantially reduce both ambiguity and parse time.

Likewise, ambiguity caused by Wolof ideophones were dealt with in a systematic way. For this word class, I introduced a special c-

structure category based on the main assumption that ideophones behave like verb particles. Using lexical specification, collocational verbs that subcategorize for ideophones were then constrained to specify the lexical form of the particles they select for. Following on from this, a functional template was used to concatenate the arguments represented by the collocational verb and the ideophonic particle. In addition, optimality marks were used to state a preference for the ideophonic reading, when ideophones co-occur with the collocational verb. This helped to control ambiguity caused by the collocational verb and ideophones, resulting in a substantial improvement in parse efficiency.

Disambiguation at the pre-parsing stage includes handling morphological and lexical ambiguities using CG-based approaches. The application of these approaches showed that, with a modest number of CG rules, the average number of readings per token and therefore the large number of lexical and morphological ambiguities can be reduced significantly. Also, the CG-based model proved to be very useful when dealing with less-resourced languages, as it avoids the requirement for a large training corpus. Equally interesting, the CG-based techniques were combined with the c-structure pruning mechanism to tackle ambiguities that arise both at the pre-parsing and at the parsing stages. The application of the c-structure pruning mechanism led to a considerable reduction of structural ambiguity in the grammar. It caused a great decrease of the number of the c-structures built in the XLE chart parser, allowing for a significant improvement in parsing efficiency. In terms of ambiguity reduction and efficiency, techniques based on CG and c-structure pruning proved to be the most effective ones. The experiment results show that the combination of c-structure pruning with CG-based disambiguation can greatly reduce the ambiguity rate by ca. 80% and increase parsing efficiency by 58%, however at the expense of the accuracy of the overall system. The parsing quality decreased by about 3.62 points in f-score. With more training data for c-structure pruning, better results could be expected.

To provide a high-level comparison of disambiguation options, this work has also experimented with optimality marking. The mechanism is used to manage ambiguity caused by ideophonic expressions, asyndetic coordination and verb subcategorization frames. Although OT filtering was originally intended to be effective in filtering syn-

tactic ambiguity, the current findings suggest that the preference constraints are frequently faced with exceptions and counterexamples. Optimality marking seems to have only occasional effects, for instance when disambiguating clear cases like ideophonic expressions or when taking advantage of STOPPOINT effects in certain cases. The results show that, with constructions like asyndetic coordination, using the STOPPOINT mark can prove very beneficial in terms of ambiguity reduction, but also eliminates a substantial number of desired interpretations and decreases parsing efficiency. In the Wolof grammar, the latter approach is currently used as a default option, selected from the explored possible approaches failing an optimal solution.

In addition, this work has discussed grammar engineering utilities that facilitate ambiguity management at the parsing and post-parsing levels. It has shown that XLE provides useful built-in tools that allow for automatic packing of representations of the alternative parse solutions. Such tools are valuable in dealing with global ambiguities where appropriate readings of a construction need to be preserved. For large-scale grammars, one particularly interesting feature of these tools is that they provide grammar writers with very useful information on spurious ambiguities. As the Wolof case has shown, it is crucially important to develop strategies for avoiding vacuous ambiguities, including ways of mending non-exclusive disjunctions originating from duplicate solutions. With a relatively simple disambiguation technique consisting in preventing spurious explorations of the grammar, the performance of the parser could significantly be improved.

More interestingly, the paper shows how efficient and elegant LFG parse disambiguation can be achieved using discriminants. Presented in a user-friendly way, discriminants are easy for humans to judge and are prominent in the XLE-web interface display. The user can disambiguate the sentence by selecting or rejecting discriminants and thereby retaining or rejecting sets of corresponding analyses. The efficiency of this method, as compared to presenting all the full analyses to the user, can be appreciated from the fact that a combination of a small number of local ambiguities can result in a large number of analyses. Applied on the Wolof LFG treebank, this method allowed to efficiently deal with ambiguity using lexical, morphological, c-structure and f-structure discriminants. As with CG, this disambiguation method is particularly attractive, as it does not require much training data.

Table 6: Comparison of the impact of five different disambiguation methods on the Wolof test data

Disambiguation method	Ambiguity reduction	Parse time	Drop in parsing accuracy
Underspecification	≈ 8%	reduced by 4%	None
Lexical specification	≈ 4%	reduced by 16%	None
CG pre-filtering	≈ 77%	reduced by 30%	2.5
C-structure pruning	≈ 72%	reduced by 36%	0.97
Optimality marking	≈ 80%	increased by 6%	0.94

As noted earlier, the various disambiguation methods were applied on different parsing levels and parser versions. The methods have interactions which are very difficult to control systematically. In the same way, it is very tedious to measure all combinations of all techniques. Table 6 gives estimates of the gain or drop that would result from adding some of the techniques.

The table shows the impact of using underspecification, lexical specification for coverbal ideophones, CG-based disambiguation, c-structure pruning and optimality marking. Disambiguation techniques based on the formal encoding of noun class indeterminacy via underspecification and lexical specification apply alternative descriptive devices that reduced both ambiguity and parse time, but otherwise leave the space of analyses unchanged (i.e., they did not lead to a drop in parsing accuracy). CG pre-filtering greatly reduced both ambiguity and parse time, but caused a drop of 2.5 points in f-score. Likewise, with c-structure pruning, the ambiguity rate as well as the parse time were reduced significantly, but the accuracy also decreased by about 0.97 points. Finally, with optimality marking, ambiguity dropped significantly, but there was a slight increase in parse time and a substantial drop in parsing accuracy.

10 ACKNOWLEDGEMENTS

This research has received support from the EC under FP7, Marie Curie Actions SP3-People-ITN 238405 (CLARA).

REFERENCES

Mohammed ATTIA (2008), *Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation*, Ph.D. thesis, University of Manchester.

Mariyaama BA (2007), *Bataaxal bu gudde nii (So long a letter)*, Nouvelles Editions Africaines du Sénégal (NEAS), Dakar, Sénégal.

Matthew BAERMAN, Dunstan BROWN, and Greville G. CORBETT (2005), *The syntax-morphology interface: A study of syncretism*, Cambridge University Press.

Marlyse BAPTISTA (1995), On the nature of pro-drop in Capeverdean Creole, Technical report, Harvard Working Papers in Linguistics.

Eckhard BICK (2000), *The parsing system "Palavras": Automatic grammatical analysis of Portuguese in a Constraint Grammar framework*, Aarhus University Press, Denmark.

Eckhard BICK (2009), *Basic Constraint Grammar tutorial for CG-3 (Vislcg3)*, Southern Denmark University, Copenhagen, Denmark, `http://beta.visl.sdu.dk/cg3_howto.pdf`.

Miriam BUTT, Helge DYVIK, Tracy Holloway KING, Hiroshi MASUICHI, and Christian ROHRER (2002), The parallel grammar project, in *Proceedings of the COLING 2002 Workshop on Grammar Engineering and Evaluation*, pp. 1–7, Taipei, Taiwan.

Miriam BUTT, Tracy Holloway KING, María-Eugenia NIÑO, and Frédérique SEGOND (1999), *A grammar writer's cookbook*, CSLI Publications, Stanford, CA, USA.

Aoife CAHILL, Tracy Holloway KING, and John T. MAXWELL III (2007), Pruning the search space of a hand-crafted parsing system with a probabilistic parser, in *Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing*, pp. 65–72, Prague, Czech Republic.

Aoife CAHILL, John T. MAXWELL III, Paul MEURER, Christian ROHRER, and Victoria ROSÉN (2008), Speeding up LFG parsing using c-structure pruning, in *Proceedings of the Workshop on Grammar Engineering Across Frameworks*, pp. 33–40, Manchester, UK.

David CARTER (1997), The TreeBanker. A tool for supervised training of parsed corpora, in *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, pp. 9–15, Madrid, Spain.

Francis CHANTREE (2004), Ambiguity management in Natural Language Generation, in *Proceedings of the Seventh Annual CLUK Research Colloquium*, pp. 23–28, Birmingham, UK.

Noam CHOMSKY (1981), *Lectures on Government and Binding*, Foris, Dordrecht, The Netherlands.

Mamadou CISSÉ (1994), *Contes Wolof modernes (Modern Wolof tales)*, L'harmattan, Paris, France.

Ann COPESTAKE and Dan FLICKINGER (2000), An open source grammar development environment and broad-coverage English grammar using HPSG, in *Proceedings of the Second International Conference on Language Resources and Evaluation*, Athens, Greece.

Max COPPERMAN and Frédérique SEGOND (1996), Computational grammars and ambiguity: the bare bones of the situation, in *Proceedings of the LFG '96 Conference*, Grenoble, France.

Denis CREISSELS (2001), Setswana ideophones as uninflected predicative lexemes, *Typological Studies in Language*, 44:75–86.

Dick CROUCH, Mary DALRYMPLE, Ron KAPLAN, Tracy Holloway KING, John T. Maxwell III, and Paula NEWMAN (2013), XLE documentation, On-line, Palo Alto Research Center (PARC), `http://www2.parc.com/isl/groups/nltt/xle/doc/xle_toc.html`.

Berthold CRYSMANN (2005), Syncretism in German: a unified approach to underspecification, indeterminacy, and likeness of case, in *Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar*, pp. 91–107, Lisbon, Portugal.

Mary DALRYMPLE (2001), *Lexical-Functional Grammar*, volume 34 of *Syntax and Semantics*, Emerald Group Publishing Limited, Bingley, West Yorkshire, UK.

Mary DALRYMPLE and Ronald M. KAPLAN (1997), A set-based approach to feature resolution, in *Proceedings of the LFG '97 Conference*, San Diego, CA, USA.

Mary DALRYMPLE, Tracy Holloway KING, and Louisa SADLER (2009), Indeterminacy by underspecification, *Journal of Linguistics*, 45(01):31–68.

Tino DIDRIKSEN (2003), Constraint Grammar manual, `http://beta.visl.sdu.dk/cg3/vislcg3.pdf`, apS, GrammarSoft.

Cheikh M. Bamba DIONE (2012a), An LFG approach to Wolof cleft constructions, in *Proceedings of the LFG '12 Conference*, pp. 157–176, Stanford, CA, USA.

Cheikh M. Bamba DIONE (2012b), A morphological analyzer for Wolof using finite-state techniques, in *Proceedings of the Eigth International Conference on Language Resources and Evaluation*, Istanbul, Turkey.

Cheikh M. Bamba DIONE (2013a), Handling Wolof clitics in LFG, in Christine Meklenborg SALVESEN and Hans P. HELLAND, editors, *Challenging Clitics*, pp. 87–118, John Benjamins, Amsterdam, The Netherlands.

Cheikh M. Bamba DIONE (2013b), Valency change and complex predicates in Wolof: an LFG account, in *Proceedings of the LFG '13 Conference*, pp. 232–252, Stanford, CA, USA.

Cheikh M. Bamba DIONE (2014), Pruning the search space of the Wolof LFG grammar using a probabilistic and a constraint grammar parser, in *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, Reykjavik, Iceland, ISBN 978-2-9517408-8-4.

Clement M. DOKE (1935), *Bantu Linguistic Terminology*, Longmans, Green and Company, London, England.

Helge DYVIK (2000), Nødvendige noder i norsk. Grunntrekk i en leksikalsk-funksjonell beskrivelse av norsk syntaks. [Necessary nodes in Norwegian. Basic properties of a lexical-functional description of Norwegian syntax.], in Øivin ANDERSEN, Kjersti FLØTTUM, and Torodd KINN, editors, *Menneske, språk og felleskap*, Novus forlag, Oslo, Norway.

Dan FLICKINGER (2000), On building a more efficient grammar by exploiting types, *Natural Language Engineering*, 6(1):15–28.

Anette FRANK (2002), A (discourse) functional analysis of asymmetric coordination, in *Proceedings of the LFG '02 Conference*, pp. 174–196, Athens, Greece.

Anette FRANK, Tracy Holloway KING, Jonas KUHN, and John T. MAXWELL III (1998), Optimality Theory style constraint ranking in large-scale LFG grammars, in *Proceedings of the LFG '98 Conference*, Stanford, CA, USA.

Anette FRANK, Tracy Holloway KING, Jonas KUHN, and John T. MAXWELL III (2001), Optimality Theory style constraint ranking in large-scale LFG grammars, in Peter SELLS, editor, *Formal and Empirical Issues in Optimality Theoretic Syntax*, pp. 367–398, CSLI Publications, Stanford, CA, USA.

Nataali Dominik GARROS, editor (1997), *Bukkeek "perigam" bu xonq: teeñ yi (Hyena and its red wig: the lice)*, Dakar, Senegal: SIL; Paris, France: EDICEF, dr. Moren ak mbootayu "xale dimbale xale". trad. du français en wolof par Momar Touré.

Gerald GAZDAR and Chris MELLISH (1989), *Natural Language Processing in {LISP}*, Addison-Wesley, Boston, MA, USA.

Talmy GIVÓN (1979), *On understanding grammar*, Academic Press, New York, NY, USA.

Pascual Cantos GÓMEZ (1996), *Lexical ambiguity, dictionaries and corpora*, Services de Publicaciones, Universidad de Murcia, Spain.

Ron KAPLAN and Joan BRESNAN (1982), Lexical-Functional Grammar: a formal system for grammatical representation, in Joan BRESNAN, editor, *The Mental Representation of Grammatical Relations*, pp. 173–281, MIT Press, Cambridge, MA, USA.

Ronald M. KAPLAN, John T. MAXWELL III, Tracy Holloway KING, and Richard CROUCH (2004), Integrating finite-state technology with deep LFG grammars, in *Proceedings of the ESSLLI 2004 Workshop on Combining Shallow and Deep Processing for NLP*, Nancy, France.

Fred KARLSSON (1990), Constraint Grammar as a framework for parsing running text, in *Proceedings of the 13th Conference on Computational Linguistics*, pp. 168–173, Helsinki, Finland.

Gerard KEMPEN (1991), Conjunction reduction and gapping in clause-level coordination: an inheritance-based approach, *Computational Intelligence*, 7(4):357–360.

Tracy Holloway KING, Stefanie DIPPER, Anette FRANK, Jonas KUHN, and John MAXWELL (2000), Ambiguity management in grammar writing, in *Proceedings of the ESSLLI 2000 Workshop on Linguistic Theory and Grammar Implementation*, pp. 5–19, Birmingham, UK.

Tracy Holloway KING, Stefanie DIPPER, Anette FRANK, Jonas KUHN, and John T. MAXWELL III (2004), Ambiguity management in grammar writing, *Research on Language and Computation*, 2(2):259–280.

Ekaterini KLEPOUSNIOTOU (2002), The processing of lexical ambiguity: homonymy and polysemy in the mental lexicon, *Brain and Language*, 81(1):205–223.

Nobo KOMAGATA (2004), A solution to the spurious ambiguity problem for practical Combinatory Categorial Grammar parsers, *Computer Speech & Language*, 18(1):91–103.

Maryellen C. MACDONALD, Neal J. PEARLMUTTER, and Mark S. SEIDENBERG (1994), Lexical nature of syntactic ambiguity resolution, *Psychological Review*, 101(4):676–703.

Christopher D. MANNING and Hinrich SCHÜTZE (1999), *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA, USA.

William MARTIN, Kenneth CHURCH, and Ramesh PATIL (1987), *Preliminary analysis of a breadth-first parsing algorithm: theoretical and experimental results*, Springer, Berlin/Heidelberg, Germany.

John T. MAXWELL and Ronald M. KAPLAN (1993), The interface between phrasal and functional constraints, *Computational Linguistics*, 19(4):571–590.

John T. MAXWELL III and Ronald M. KAPLAN (1996), Unification-based parsers that automatically take advantage of context-freeness, in *Proceedings of the LFG '96 Conference*, Grenoble, France.

Fiona MCLAUGHLIN (2004), Is there an adjective class in Wolof?, in Robert M. W. DIXON and Alexandra Y. AIKHENVALD, editors, *Adjective Classes: A Cross-Linguistic Typology*, pp. 242–262, Oxford University Press, Oxford, UK.

Fiona MCLAUGHLIN (2010), Noun classification in Wolof: When affixes are not renewed, *Studies in African Linguistics*, 26(1):1–28.

Marjorie J. MCSHANE (2005), *A theory of ellipsis*, Oxford University Press, Oxford, UK.

Alan PRINCE and Paul SMOLENSKY (1993), Optimality Theory: constraint interaction in Generative Grammar, Technical report, Rutgers University Center for Cognitive Science, Cambridge, MA, USA.

Stefan RIEZLER, Tracy Holloway KING, Ronald M. KAPLAN, Richard CROUCH, John T. MAXWELL III, and Mark JOHNSON (2002), Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques, in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 271–278, Philadephia, PA, USA.

Victoria ROSÉN, Paul MEURER, and Koenraad DE SMEDT (2005), Constructing a parsed corpus with a large LFG grammar, in *Proceedings of the LFG '05 Conference*, pp. 371–387, Stanford, CA, USA.

Victoria ROSÉN, Paul MEURER, and Koenraad DE SMEDT (2007), Designing and implementing discriminants for LFG grammars, in *Proceedings of the LFG '07 Conference*, pp. 397–417, Stanford, CA, USA.

Victoria ROSÉN, Paul MEURER, and Koenraad DE SMEDT (2009), LFG Parsebanker: a toolkit for building and searching a treebank as a parsed corpus, in *Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories*, pp. 127–133, Utrecht, The Netherlands.

David J. SAPIR (1971), West Atlantic: an inventory of the languages, their noun class systems and consonant alternation, *Current Trends in Linguistics*, 7(1):43–112.

William H. TORRENCE (2005), *On the distribution of complementizers in Wolof*, Ph.D. thesis, University of California, Los Angeles, CA, USA.

Hans USZKOREIT, Rolf BACKOFEN, Stephan BUSEMANN, Abdel K. DIAGNE, Elizabeth A. HINKLEMAN, Walter KASPER, Bernd KIEFER, Hans-Ulrich KRIEGER, Klaus NETTER, Günter NEUMANN, *et al.* (1994), DISCO: an HPSG-based NLP system and its application for appointment scheduling, in *Proceedings of the 15th Conference on Computational Linguistics*, pp. 436–440, Kyoto, Japan.

Erhard F. K. VOELTZ and Christa KILIAN-HATZ, editors (2001), *Ideophones*, Typological Studies in Language, John Benjamins, Amsterdam, The Netherlands.

Sylvie VOISIN-NOUGUIER (2002), *Relations entre fonctions syntaxiques et fonctions sémantiques en Wolof (Relations between syntactic functions and semantic functions in Wolof)*, Ph.D. thesis, Université Lumière (Lyon 2), Lyon, France.