

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

TRAITEMENT AUTOMATIQUE DE LA LANGUE NATURELLE : LA GRAMMAIRE
CATEGORIELLE COMBINATOIRE APPLIQUEE AU WOLOF

MÉMOIRE PRÉSENTÉ COMME EXIGENCE PARTIELLE DE LA

MAÎTRISE EN MATHÉMATIQUES ET
INFORMATIQUE APPLIQUÉES

PAR

DIAGA MBOUP

Février 2026

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

RÉSUMÉ

Le présent mémoire explore l'application de la grammaire catégorielle combinatoire (GCC) à la modélisation syntaxique du wolof. Bien que le wolof soit parlé par plus de dix millions de locuteurs en Afrique de l'Ouest, il demeure relativement peu doté en ressources linguistiques formalisées et en outils d'analyse syntaxique fondés sur des théories grammaticales explicites.

Après un examen des travaux existants sur le wolof et des approches développées pour d'autres langues peu dotées, ce travail propose une première expérimentation fondée sur un jeu de données composé de cent (100) phrases en wolof. L'objectif est d'évaluer la pertinence du formalisme catégoriel combinatoire pour rendre compte de certaines structures morphosyntaxiques caractéristiques de la langue.

Un lexique typé a été conçu et implémenté à l'aide de la bibliothèque NLTK et de son module NLTKCCG en Python. Cette implémentation permet de dériver formellement les structures syntaxiques des phrases analysées et d'observer le comportement du formalisme dans un contexte contrôlé.

Sans prétendre à une couverture exhaustive, ce travail constitue une contribution exploratoire à la formalisation syntaxique du wolof et participe, à son échelle, à l'enrichissement des ressources formelles disponibles pour les langues africaines peu dotées.

Mots-clés : grammaire catégorielle combinatoire, wolof, formalisation syntaxique, NLTK, linguistique computationnelle, langues peu dotées.

ABSTARCT

This thesis explores the application of Combinatory Categorical Grammar (CCG) to the syntactic modeling of Wolof. Although Wolof is spoken by more than ten million speakers in West Africa, it remains relatively under-resourced in terms of formally structured linguistic descriptions and computational tools grounded in explicit grammatical theories.

Following a review of existing work on Wolof and on other low-resource languages, this study presents an exploratory experiment based on a dataset of one hundred (100) Wolof sentences. The objective is to assess the relevance of the CCG formalism in accounting for selected morphosyntactic structures of the language.

A typed lexicon was designed and implemented using the NLTK library and its NLTKCCG module in Python. This implementation enables the formal derivation of syntactic structures and allows for an examination of the behavior of the formalism within a controlled experimental setting.

Without claiming exhaustive coverage, this work constitutes an exploratory contribution to the formal syntactic modeling of Wolof and aims to contribute, at its scale, to the development of formal resources for low-resource African languages.

Keywords: Combinatory Categorical Grammar, Wolof, syntactic formalization, NLTK, computational linguistics, low-resource languages.

REMERCIEMENTS

Tout d'abord, je tiens à remercier mon directeur de recherche, M. Ismail Biskri pour ses conseils et ses orientations qui ont contribué à la réalisation de ce mémoire. Ses indications m'ont permis de mieux structurer ma réflexion et d'avancer dans ce travail.

Je souhaite également remercier l'Université du Québec à Trois-Rivières, du corps professoral au personnel administratif, pour les services rendus et l'encadrement offert. Je leur suis particulièrement reconnaissante pour le soutien financier apporté, sans lequel la réalisation de ce mémoire n'aurait peut-être pas été possible.

Ce mémoire est entièrement dédié à ma famille. Mes parents et mes frères ont toujours été les piliers de ma vie, merci de me soutenir autant, de me montrer votre amour tous les jours ; vous êtes ma plus grande force.

TABLE DES MATIÈRES

| | |
|---|----|
| RÉSUMÉ | 2 |
| ABSTARCT | 3 |
| REMERCIEMENTS | 4 |
| LISTE DES TABLEAUX | 8 |
| LISTE DES FIGURES | 9 |
| Chapitre 1 : Introduction..... | 10 |
| Chapitre 2 : État de l'art..... | 14 |
| 2.1 Travaux linguistiques et ressources numériques pour le wolof..... | 14 |
| 2.2 Traitement automatique des langues africaines peu dotées..... | 15 |
| 2.3 Grammaire catégorielle combinatoire et applications internationales | 15 |
| Chapitre 3 : Le Wolof..... | 17 |
| 1. Syntaxe..... | 18 |
| 1.1. Ordre des mots..... | 18 |
| 1.2. Variantes syntaxiques | 19 |
| Coordination et Subordination | 23 |
| 2. Morphologie | 28 |
| 2.1. Personnes | 28 |
| 2.2. Genre..... | 29 |
| 2.3. Nombre..... | 30 |
| 3. Polysémie | 31 |
| 4. Rôles thématiques | 31 |
| 5. Marqueurs temporels et suffixe "Y" | 32 |
| Chapitre 4 : Les grammaires catégorielles..... | 33 |
| 1. Présentation générale | 33 |
| 2. Principes fondamentaux des grammaires catégorielles..... | 34 |
| Conclusion | 37 |
| 2.2. Les règles combinatoires..... | 37 |
| 2.3. Exemples d'analyse syntaxique..... | 40 |
| 2.4. De la GCC à la GCCA (version applicative) | 44 |
| 2.5. Avantages de la GCCA en TALN | 45 |
| 2.6 Représentation sémantique (λ -calcul)..... | 46 |
| 2.7. Comparaison entre GCC et GCCA..... | 46 |

| | |
|---|----|
| 2.8. Polyvalence linguistique | 47 |
| Chapitre 5 : Analyse du wolof au moyen des grammaires catégorielles..... | 48 |
| 4.1. Phrase déclarative simple | 48 |
| 4.2. Focalisation du sujet..... | 49 |
| 4.3. Topicalisation..... | 50 |
| 4.4. Relative simple..... | 51 |
| 4.5. Coordination | 52 |
| Chapitre 6 : implantation | 55 |
| 1. Choix des technologies | 55 |
| 1.1. Choix du langage de programmation : Python..... | 55 |
| 1.2. Choix de la bibliothèque : Étude comparative entre NLTKCCG et OpenCCG | 56 |
| 1.3. Workflow de l'analyse | 58 |
| 2. Modélisation du lexique wolof selon la Grammaire Catégorielle Combinatoire (GCC) | 58 |
| 3. Développement de l'interface d'analyse | 63 |
| 3.1. Technologies utilisées..... | 63 |
| 3.2. Structure de l'interface..... | 63 |
| 3.3. Exemple d'exécution | 64 |
| 3.4. Pipeline de fonctionnement..... | 66 |
| Chapitre 7 : Expérimentation | 67 |
| 1. Introduction et constitution du corpus | 67 |
| 2. Protocole expérimental..... | 68 |
| 3. Analyse des résultats | 68 |
| 3.1. Corpus utilisé..... | 68 |
| 3.2. Phrases bien construites et acceptées..... | 69 |
| 3.3. Phrases bien construites mais non acceptées | 70 |
| 3.4. Phrases mal construites mais acceptées | 72 |
| 3.5. Phrases mal construites et non acceptées..... | 73 |
| 3.6. Synthèse des résultats | 75 |
| 4. Discussion des résultats..... | 76 |
| 5. Synthèse de l'expérimentation | 77 |
| Chapitre 8 : Conclusion et perspectives..... | 78 |
| REFERENCES..... | 80 |
| DICTIONNAIRE DES TYPES UTILISÉS..... | 84 |

LISTE DES TABLEAUX

- Tableau 1 : Étude comparative entre la topicalisation et la focalisation en wolof 22*
- Tableau 2 : Exemples de catégories fonctionnelles complexes en grammaire catégorielle 35*
- Tableau 3 : Exemple d'analyse syntaxique (1) 40*
- Tableau 4 : Exemple d'analyse syntaxique (2) 41*
- Tableau 5 : Exemple d'analyse syntaxique (3) 42*
- Tableau 6 : Exemple d'analyse syntaxique (4) 43*
- Tableau 7 : Présentation des combineurs principaux 45*
- Tableau 8 : Tableau comparatif entre la GCC et la GCCA 47*
- Tableau 9 ; Tableau comparatif entre NLTKCCG et OpenCCG 57*
- Tableau 10 : Exemple d'analyse syntaxique en wolof selon la GCC(1) 60*
- Tableau 11 : Exemple d'analyse syntaxique en wolof selon la GCC(2) 60*
- Tableau 12 : Exemple d'analyse syntaxique en wolof selon la GCC(3) 61*
- Tableau 13 : Exemple d'analyse syntaxique en wolof selon la GCC(4) 62*
- Tableau 14 : Tableau de synthèse des résultats 76*

LISTE DES FIGURES

Figure 1: Workflow de l'implantation de l'analyse syntaxique 58

Figure 2 : Exemple de structure du fichier JSON 62

Figure 3 : Interface d'analyse syntaxique développée avec Tkinter 64

Figure 4 : Exemple d'exécution de l'analyseur syntaxique via l'interface développée sur la phrase « dama wañi xaliss bi » 65

Figure 5 : Schéma global du fonctionnement du système d'analyse syntaxique 66

Figure 6 : Sélection d'une phrase via le menu déroulant 70

Figure 7 : Exemple d'exécution d'une phrase en wolof bien construite et acceptée par le système 70

Figure 8 : Exemple d'exécution d'une phrase en wolof bien construite mais non acceptée par le système 72

Figure 9 : Exemple d'exécution d'une phrase en wolof mal construite mais acceptée par le système 73

Figure 10 : Exemple d'exécution d'une phrase en wolof mal construite et non acceptée par le système 75

Chapitre 1 : Introduction

Le wolof est une langue parlée principalement au Sénégal, en Mauritanie et en Gambie. Elle appartient à la famille des langues sénégalaises, dans la branche des langues atlantiques. D'après le professeur Cheikh Anta Diop [17] le wolof serait une langue issue de l'égyptien ancien, faisant partie des langues dérivées de celui-ci.

Issu de l'ethnie wolof, le wolof est la langue la plus parlée au Sénégal, où près de 80 % de la population, soit environ 13 millions de personnes sur une population totale d'environ 17 millions, l'utilisent comme première ou seconde langue. Environ 43 % des Sénégalais, soit environ 7,3 millions de personnes, appartiennent à l'ethnie wolof. En plus de son rôle central au Sénégal, le wolof bénéficie du statut de langue nationale en Mauritanie et en Gambie.

Au Sénégal, le wolof est compris par environ 90 % de la population, jouant un rôle de langue véhiculaire interethnique. En Gambie, environ 13 % de la population, soit près de 270 000 personnes sur environ 2,1 millions d'habitants, parlent le wolof, un chiffre légèrement supérieur à la proportion de l'ethnie wolof dans ce pays. En Mauritanie, environ 22 % des locuteurs, soit près de 1,1 million de personnes sur une population de 5,4 millions, parlent le wolof, principalement à Nouakchott et dans la vallée du fleuve Sénégal.

Le wolof est également appris et parlé par des expatriés vivant au Sénégal, notamment des Européens, des Américains et des Asiatiques installés dans le pays. On estime qu'environ 60 000 à 70 000 expatriés étrangers vivent au Sénégal, dont une part croissante adopte le wolof comme langue de communication quotidienne, en particulier à Dakar et dans les grandes villes. Bien que beaucoup utilisent le français comme langue principale, le wolof est souvent appris pour des raisons d'intégration sociale et professionnelle, notamment dans les secteurs du commerce et de l'éducation.

En dehors de ces pays, des communautés wolofophones sont également présentes dans d'autres pays d'Afrique de l'Ouest, notamment en Guinée et au Mali. Le wolof sert aussi de langue de communication au sein des diasporas africaines en Europe et aux États-Unis, où il est couramment utilisé dans les communautés sénégalaises. La diaspora sénégalaise en Europe est estimée à environ 500 000 personnes, principalement en France, en Espagne et en Italie, tandis qu'aux

États-Unis, la communauté sénégalaise regroupe environ 50 000 personnes, principalement concentrées dans des villes comme New York et Atlanta.

Les premiers ouvrages sur la langue wolof ont été rédigés par des colons, notamment parce que la tradition africaine était autrefois majoritairement orale. Jean Dard [11] a publié le premier dictionnaire français-wolof en 1825, suivi d'une grammaire en 1826 [12]. Sa démarche se veut pédagogique, tandis que la grammaire publiée en 1859 par le gouverneur Jacques-François Roger [28] cherche à comprendre la structure profonde de la langue. Ces travaux marquent le début des études systématiques du wolof.

En 1858, David Boilat [9], un prêtre franco-sénégalais, publie une Grammaire de la langue woloffe, récompensée par le prix Volney de l'Institut de France Louis Faidherbe [22], militaire et homme politique français arrivé au Sénégal en 1852, publie en 1864 un lexique français avec des traductions en wolof, soninké et peul

Au fil des ans, de nombreux intellectuels sénégalais ont pris la relève pour étudier et promouvoir leur langue. Parmi eux, Cheikh Anta Diop [17] est l'un des plus célèbres. Historien, anthropologue et linguiste, il a démontré les racines africaines des langues, dont le wolof, et a largement contribué à son étude académique. Birago Diop [16], écrivain et poète sénégalais, a aussi largement utilisé le wolof dans ses écrits, notamment dans *Les Contes d'Amadou Koumba*. Ce travail a permis de préserver de nombreuses traditions et expressions wolof dans la littérature. Le professeur Pathé Diagne [14] a également joué un rôle clé dans la traduction d'œuvres et la normalisation de l'écriture du wolof, facilitant son enseignement et sa diffusion.

Ces intellectuels, parmi d'autres, ont permis au wolof de se positionner comme une langue de culture et d'étude, non seulement au Sénégal, mais aussi dans les cercles académiques internationaux.

Malgré cette reconnaissance, le wolof reste aujourd'hui une langue peu dotée du point de vue du traitement automatique et de la modélisation syntaxique formelle. Les travaux spécifiquement dédiés à l'analyse syntaxique du wolof restent rares, bien que certaines avancées aient été réalisées dans le domaine du TALN, notamment autour de la standardisation, de la correction orthographique ou des assistants vocaux. Ces éléments seront développés plus en détail dans le chapitre consacré à la langue wolof.

Le wolof a des règles sémantiques particulières. Certaines d'entre elles rendent la langue plus difficile à analyser à l'écrit, comme dans le cadre d'une analyse combinatoire catégorielle, qu'à l'oral, par exemple dans une synthèse vocale.

Dans cette étude, une exploration de la grammaire catégorielle appliquée au wolof sera faite. Cette approche linguistique offre une fenêtre intéressante sur la structure profonde de la langue, mettant en lumière la manière dont elle peut être complexe et catégorise ses éléments lexicaux. Il s'agira ici d'une analyse se concentrant sur la richesse morphologique du wolof, examinant les marques spécifiques qui indiquent les fonctions grammaticales associées à la syntaxe, à la morphologie et à la sémantique fonctionnelle.

En plongeant dans la grammaire catégorielle du wolof, l'objectif est de dévoiler les mécanismes sous-jacents qui façonnent la construction des phrases dans cette langue. Une dualité entre l'approche linguistique et l'approche technique sera présente tout au long de ce document. Par souci de clarté, des exemples issus d'autres langues comme le français seront parfois mobilisés.

L'un des premiers défis concerne l'accès limité à de la documentation dédiée à l'analyse formelle du wolof. En effet, dans le domaine du traitement automatique des langues, les ressources linguistiques (corpus annotés, lexiques, modèles...) sont souvent restreintes ou inexistantes pour cette langue. Ce manque de données annotées et de normes standardisées complique l'entraînement des modèles. L'indisponibilité de plateformes adaptées accentue cette difficulté, la majorité des technologies TALN étant développées pour l'anglais ou d'autres langues largement représentées.

Ainsi, l'approche expérimentale de ce mémoire consiste à constituer un jeu de données original de cent (100) phrases en wolof, limité volontairement pour des raisons de faisabilité et de temps. Ces phrases couvrent une diversité de structures grammaticales. La méthodologie se décline en trois étapes principales :

- Constitution du jeu de données : sélection manuelle de phrases variées issues du wolof contemporain ;
- Application du modèle catégoriel : typage et analyse syntaxique via les bibliothèques NLTK et NLTKCCG ;
- Implantation : évaluation des résultats et visualisation à l'aide d'une interface.

Ces étapes permettent d'évaluer les performances du formalisme utilisé, ses forces comme ses limites dans le traitement de la syntaxe wolof.

Ce chapitre qu'est l'introduction expose les fondements de notre travail de recherche.

Le chapitre 2 met en évidence quelques travaux existants en wolof, mais aussi ceux d'autres langues

Le chapitre 3 est consacré à une présentation de la langue wolof. Il met l'accent sur sa structure syntaxique, morphologique et sémantique, tout en évoquant les travaux existants dans le domaine du TALN.

Le chapitre 4 traite de la grammaire catégorielle. Nous y présentons les fondements formels du modèle, ses règles de composition, ses combinateurs, ainsi que ses avantages pour la modélisation linguistique.

Le chapitre 5 est dédié à l'analyse du wolof au moyen des grammaires catégorielles et le chapitre 6 à l'implantation du modèle. Il détaille les choix technologiques, la modélisation du lexique, l'interface d'analyse, et les types syntaxiques retenus.

Le chapitre 7 quant à lui, propose une expérimentation sur un corpus de cent phrases, afin de mesurer la performance du système et de discuter les résultats obtenus.

Le chapitre 8 est la conclusion de ce travail de recherche.

Chapitre 2 : État de l’art

2.1 Travaux linguistiques et ressources numériques pour le wolof

Le wolof bénéficie d’une tradition descriptive ancienne, notamment à travers les travaux grammaticaux de Dard [12] et de Boilat [9], qui ont posé les bases de l’analyse morphologique et syntaxique de la langue. Ces descriptions ont contribué à structurer l’étude des catégories grammaticales et des mécanismes syntaxiques fondamentaux.

Des études plus récentes ont approfondi certains phénomènes spécifiques du wolof. Dione [15], par exemple, propose une analyse des propositions relatives dans le cadre de la « Lexical Functional Grammar » (LFG), mettant en évidence les particularités structurelles de la langue. De même, Nguer [18] examine le système verbal du wolof, en analysant les marqueurs aspectuels et temporels. Ces travaux enrichissent la compréhension théorique de la structure syntaxique du wolof, mais restent principalement descriptifs.

Sur le plan des ressources numériques, des initiatives récentes ont permis la constitution de données exploitables en traitement automatique. Le corpus parallèle SENCORPUS [33] comprend environ 70 000 phrases alignées français–wolof. Ce projet vise à fournir une base pour la traduction automatique et l’apprentissage de représentations bilingues. La disponibilité d’un tel corpus constitue une avancée importante pour une langue peu dotée, en permettant l’entraînement de modèles statistiques ou neuronaux.

Par ailleurs, le développement d’un « treebank » wolof dans le cadre du projet « Universal Dependencies » [32] a conduit à l’annotation syntaxique de 2 107 phrases totalisant 43 584 tokens. Ce travail fournit une représentation en dépendances compatible avec un schéma international standardisé, permettant l’évaluation de parseurs automatiques. Les auteurs soulignent toutefois certaines difficultés spécifiques au wolof, notamment la segmentation morphologique et le traitement des clitiques, qui influencent directement les performances des outils automatiques.

Ces initiatives montrent que le wolof dispose désormais de ressources annotées et de corpus exploitables. Néanmoins, les approches adoptées restent majoritairement orientées vers l’annotation dépendancielle ou l’apprentissage automatique, sans formalisation grammaticale catégorielle explicite.

2.2 Traitement automatique des langues africaines peu dotées

Le développement du traitement automatique des langues africaines s’inscrit dans un contexte de rareté des données et de ressources limitées. Plusieurs projets ont cherché à pallier cette situation en créant des corpus annotés et en adaptant des modèles multilingues.

Un exemple représentatif est le projet MasakhaPOS [34], qui propose des jeux de données d’étiquetage morpho-syntaxique pour vingt langues africaines typologiquement diverses (par exemple le Yoruba, le Hausa, l’Amharique ou encore le Swahili). L’objectif du projet est de fournir des données normalisées selon le cadre Universal Dependencies afin de permettre l’entraînement de modèles supervisés. Les résultats montrent que les performances varient fortement selon la taille des données disponibles, mais que le transfert multilingue permet d’atteindre des scores compétitifs, souvent supérieurs à 80 % d’exactitude en étiquetage pour certaines langues disposant de volumes annotés suffisants.

Ces travaux démontrent que l’apprentissage automatique, notamment via des modèles pré-entraînés multilingues, constitue aujourd’hui l’approche dominante pour les langues peu dotées. Cependant, cette méthodologie repose principalement sur la disponibilité de données annotées et sur l’adaptation de modèles existants, plutôt que sur une modélisation grammaticale interne propre à chaque langue.

Ainsi, bien que des progrès significatifs aient été réalisés en matière de performances empiriques, les approches symboliques fondées sur des formalismes grammaticaux explicites restent peu explorées dans le contexte africain.

2.3 Grammaire catégorielle combinatoire et applications internationales

Les grammaires catégorielles trouvent leur origine dans les travaux d’Ajdukiewicz [3], de Bar-Hillel [5] et de Lambek [24], qui ont proposé une formalisation logique des catégories syntaxiques. La Grammaire Catégorielle Combinatoire (CCG), développée notamment par Steedman [30][29], constitue une extension de ce cadre théorique et repose sur un lexique riche en catégories fonctionnelles combinées au moyen d’un ensemble restreint de règles dérivationnelles.

Dans le domaine du parsing automatique, la CCG a été largement appliquée à l’anglais. Clark et Curran [35] ont développé un analyseur statistique à large couverture basé sur la CCG, évalué

sur le corpus Wall Street Journal. Leur système atteint une F-mesure de 87,6 % sur l'ensemble de test standard, tout en maintenant une couverture supérieure à 99 % des phrases analysées. Ces résultats ont contribué à établir la CCG comme un formalisme robuste et compétitif pour l'analyse syntaxique.

Des travaux ultérieurs ont combiné supertagging et parsing dérivationnel afin d'améliorer encore les performances [36]. Ces approches montrent que la CCG permet une modélisation efficace des dépendances longues et des phénomènes syntaxiques complexes, tout en conservant un lien étroit entre syntaxe et interprétation sémantique.

Cependant, si la CCG a démontré son efficacité pour des langues largement dotées comme l'anglais, son application aux langues africaines, et en particulier au wolof, demeure inexistante ou marginale. Cette absence ne reflète pas une inadéquation théorique du formalisme, mais plutôt un manque d'implémentations adaptées aux langues peu dotées.

Synthèse et positionnement

L'examen des travaux existants permet de formuler plusieurs constats :

- Le wolof dispose désormais de ressources annotées et de corpus parallèles exploitables [32][33].
- Les langues africaines peu dotées bénéficient principalement d'approches fondées sur l'apprentissage automatique et le transfert multilingue [34].
- La Grammaire Catégorielle Combinatoire a démontré son efficacité empirique dans des langues internationales, avec des performances élevées en parsing syntaxique [35][36].

Toutefois, aucune étude ne propose une formalisation catégorielle combinatoire appliquée au wolof. Le présent travail s'inscrit donc dans une perspective complémentaire aux approches existantes, en explorant l'adaptation d'un formalisme symbolique à une langue africaine peu dotée et en évaluant son potentiel pour la modélisation syntaxique.

Chapitre 3 : Le Wolof

Le wolof a déjà fait l'objet de quelques études linguistiques et de développements en traitement automatique du langage naturel (TALN). Parmi les contributions notables, le linguiste sénégalais Mamour Dramé [20], chercheur en linguistique à l'Institut Fondamental d'Afrique Noire (IFAN) de l'Université Cheikh Anta Diop, s'est illustré par ses travaux sur la standardisation et l'enseignement du wolof à l'écrit. Il est notamment l'auteur de *Dawal ak bind làmmiñu wolof*, une œuvre en deux volumes visant à améliorer la maîtrise de l'écriture du wolof, souvent mal orthographié dans l'espace public (Dramé, 2020).

Toujours dans une perspective de recherche, Thierno Ibrahima Cissé [10], informaticien et spécialiste en traitement automatique des langues, a développé un correcteur orthographique automatique pour le wolof. Son système repose sur une combinaison de structures de données de type trie, de programmation dynamique et de la distance de Levenshtein pondérée, atteignant une précision remarquable de 98,31 % (Cissé & Sadat, 2023). Ce travail s'inscrit dans une dynamique visant à fournir des outils concrets pour la valorisation numérique des langues africaines (Cissé, 2021).

Le projet DiLAF [21] – Dictionnaires de Langues Africaines, coordonné par une équipe pluridisciplinaire de linguistes et d'informaticiens, a entrepris de numériser des ressources lexicographiques en wolof, en les structurant au format XML et en les rendant accessibles en ligne (Enguehard & Mangeot, 2014). Ce projet participe à la normalisation et à la diffusion du wolof dans l'univers numérique.

Le wolof bénéficie également d'une reconnaissance internationale grâce à son inclusion dans le corpus MasakhaNER [1] [2], une initiative panafricaine regroupant chercheurs et ingénieurs en intelligence artificielle autour de la création de modèles multilingues adaptés au continent (Adelani et al., 2021 ; MasakhaNER v2, 2022). Ce projet consiste à annoter des entités nommées dans plusieurs langues africaines peu dotées, dont le wolof.

Ces travaux illustrent la dynamique actuelle visant à intégrer pleinement les langues comme le wolof dans les technologies du TALN. Cependant, ce mémoire ne prétend pas couvrir tous les aspects du wolof. La présentation qui suit se limite aux éléments syntaxiques et morphologiques nécessaires à notre analyse grammaticale, notamment dans le cadre de la grammaire catégorielle combinatoire. Cette restriction est motivée par notre volonté de modéliser le wolof dans

une approche formelle, sans entrer dans des considérations phonologiques ou sociolinguistiques plus larges.

Ce chapitre présente une description synthétique de certains aspects linguistiques du wolof, en mettant l'accent sur sa structure syntaxique et quelques particularités morphologiques et sémantiques pertinentes dans le cadre de notre étude. Loin d'être une description exhaustive de la langue, cette présentation vise à fournir un aperçu des éléments essentiels qui contribueront à bien comprendre les analyses grammaticales présentées dans les sections suivantes. Nous nous concentrerons notamment sur l'ordre des mots, quelques variantes syntaxiques, ainsi que certains aspects morphologiques tels que la formation des groupes nominaux et l'expression du temps. Ce choix est motivé par notre volonté de modéliser la syntaxe du wolof dans une approche formelle, en particulier à travers la grammaire catégorielle combinatoire.

1. Syntaxe

1.1. Ordre des mots

Le wolof suit principalement un ordre des mots **SVO (Sujet - Verbe - Objet)**. Cela signifie que dans une phrase de base, le sujet précède le verbe, qui précède à son tour l'objet. Cependant, il existe une certaine flexibilité, notamment avec des structures qui permettent de mettre l'accent sur certaines parties de la phrase.

Par exemple :

- **Xale** [nom : enfant] **bi** [article : le/la] **lekk** [verbe : manger] **ceeb bi** [le riz]

L'enfant mange le riz

- **Yayam** [Groupe nominal : sa mère] **joox na** [verbe : a donné] **xale bi** [l'enfant] **mburu** [nom : pain].

La mère a donné du pain à l'enfant (ou à cet enfant)

Toutefois, cet ordre peut être modifié dans certaines constructions syntaxiques comme la **topicalisation** et la **focalisation**.

1.2. Variantes syntaxiques

En wolof, la structure de base SVO peut être modifiée selon l'intention pragmatique du locuteur. Deux procédés majeurs permettent de mettre en relief certains constituants de la phrase : la topicalisation et la focalisation. Bien qu'ils consistent tous deux à déplacer un élément de la phrase, ils remplissent des fonctions informationnelles distinctes.

Topicalisation

La topicalisation consiste à placer un élément en début d'énoncé pour le désigner comme thème principal du discours. Il s'agit généralement d'une information connue ou présumée, sur laquelle porte la suite du message. En wolof, cet élément topicalisé est souvent repris plus loin dans la phrase par un pronom ou un marqueur prédicatif.

Exemples :

- **Ceeb** [riz] **bi** [défini] **mako** [pronom objet indirect] **lekk** [manger].
→ *Le riz, c'est moi qui l'ai mangé.*
- **Xale** [enfant] **bi** [défini], **dafa** [marqueur prédicatif] **xam** [connaître] **xibaar** [nouvelle] **bi** [défini].
→ *L'enfant connaît la nouvelle.*

Dans ces phrases, les groupes nominaux topicalisés (**ceeb bi**, **xale bi**) sont présentés comme point de départ de l'énoncé, tandis que la suite fournit une information à leur sujet. Ce phénomène correspond à la notion de thématisation (Lambrecht, 1994 ; Robert, 2010).

Focalisation

La focalisation, quant à elle, vise à mettre en valeur un élément en créant un contraste implicite avec d'autres possibilités. Il s'agit de marquer une information nouvelle, saillante ou inattendue. En wolof, ce type de mise en relief se réalise au moyen de particules focales spécifiques

1.1. Ordre des mots

Le wolof suit principalement un ordre des mots SVO (Sujet - Verbe - Objet). Cela signifie que dans une phrase de base, le sujet précède le verbe, qui précède à son tour l'objet. Cependant, il

existe une certaine flexibilité, notamment avec des structures qui permettent de mettre l'accent sur certaines parties de la phrase.

Par exemple :

• **Xale** [nom : enfant] **bi** [article : le/la] **lekk** [verbe : manger] **ceeb bi** [le riz]

L'enfant mange le riz

• **Yayam** [Groupe nominal : sa mère] **joox na** [verbe : a donné] **xale bi** [l'enfant] **mburu** [nom : pain].

La mère a donné du pain à l'enfant (ou à cet enfant)

Toutefois, cet ordre peut être modifié dans certaines constructions syntaxiques comme la topicalisation et la focalisation.

1.2. Variantes syntaxiques

En wolof, la structure de base SVO peut être modifiée selon l'intention pragmatique du locuteur, notamment via la topicalisation et la focalisation. Ces deux procédés syntaxiques déplacent un élément en début d'énoncé, mais remplissent des fonctions informationnelles distinctes.

Topicalisation

La topicalisation consiste à mettre en avant un élément déjà connu, présupposé ou donné dans le discours, appelé le thème. Cet élément, placé en début de phrase, sert de point de départ pour le reste de l'énoncé. En wolof, l'élément topicalisé est fréquemment repris plus loin dans la phrase par un pronom ou un marqueur prédicatif, évitant la répétition.

Exemples :

• **Ceeb** [riz] **bi** [défini] **mako** [pronom objet indirect] **lekk**.

→ Le riz, c'est moi qui l'ai mangé.

• **Xale** [enfant] **bi** [défini], **dafa** [marqueur prédicatif] **xam** [connaître] **xibaar** [nouvelle] **bi**.

→ L'enfant connaît la nouvelle.

Dans ces phrases, les groupes nominaux topicalisés (*ceeb bi, xale bi*) sont présentés comme point de départ de l'énoncé, tandis que la suite fournit une information à leur sujet. Ce phénomène correspond à la notion de thématisation (Lambrecht, 1994 ; Robert, 2010).

Ce mécanisme facilite la continuité du discours en soulignant le thème déjà familier.

Focalisation

La focalisation vise à mettre en relief un élément porteur d'information nouvelle, contrastive ou saillante. En wolof, cette mise en valeur s'effectue au moyen de particules focales spécifiques telles que *mo, ngay, la, lañu*, placées immédiatement après l'élément focalisé. Contrairement à la topicalisation, la focalisation ne s'accompagne pas d'une reprise pronominale.

Exemples :

- Focalisation du sujet

Moom [lui/elle] **mo** [FOC] **lekk ceeb bi**.

→ C'est lui qui mange le riz.

- Focalisation de l'objet

Ceeb bi ngay lekk.

→ C'est le riz que tu manges.

- Focalisation du lieu

Fii [ici] **lañu dem**.

→ C'est ici qu'ils sont allés.

Cette procédure attire explicitement l'attention sur l'information principale de la phrase.

Distinction fonctionnelle

Si topicalisation et focalisation partagent une structure similaire (élément en tête de phrase), leur distinction repose sur la nature de l'information mise en avant :

La topicalisation concerne un thème donné, familier dans le discours.

La focalisation concerne une information nouvelle, contrastive ou saillante.

Les indices syntaxiques et pragmatiques, tels que la reprise pronominale (présente en topicalisation) ou l’usage de particules focales (caractéristique de la focalisation), permettent de différencier ces deux phénomènes.

Enjeux linguistiques et computationnels

La distinction entre topicalisation et focalisation est cruciale pour comprendre l’organisation de l’information dans le discours wolof et pour modéliser ses structures syntaxiques et pragmatiques. Elle est également fondamentale pour les applications de traitement automatique des langues, où une bonne représentation de ces phénomènes améliore la génération et l’analyse des textes.

Voici un tableau comparatif qui résume les différences abordées :

| Critère | Topicalisation | Focalisation |
|---------------------------|---|---|
| Définition | Mise en avant d’un thème connu ou présupposé | Mise en valeur d’un élément contrastif ou inattendu |
| Fonction informationnelle | Thématisation : introduit le sujet de discussion | Focus : attire l’attention sur l’information centrale |
| Structure grammaticale | Élément topicalisé + reprise pronominale (dafa, mako, etc.) | Élément focalisé + particule spécifique (mo, ngay, la, lañu, etc.) |
| Exemple 1 | Xale bi, dafa lekk ceeb bi. (L’enfant, il mange le riz.) | Moom mo lekk ceeb bi. (C’est lui qui mange le riz.) |
| Exemple 2 | Ceeb bi, mako lekk. (Le riz, c’est moi qui le mange.) | Ceeb bi ngay lekk. (C’est le riz que tu manges.) |
| Exemple 3 | Fii, lañu dem. (Ici, ils sont allés.) | Fii lañu dem. (C’est ici qu’ils sont allés.) (forme identique, fonction différente) |

Tableau 1 : Étude comparative entre la topicalisation et la focalisation en wolof

En résumé, la topicalisation met l’accent sur le thème, c’est-à-dire ce dont on parle déjà dans le discours, tandis que la focalisation met en lumière l’information nouvelle ou la partie la plus importante du message (le focus). Cette distinction repose sur une hiérarchie informationnelle propre à chaque contexte de communication. Le thème topicalisé est généralement repris dans

l'énoncé, souvent à l'aide de marqueurs tels que *dafa* ou *mako*, tandis que l'élément focalisé est introduit comme nouveau ou contrastif, à l'aide de particules spécifiques. Cette différenciation enrichit la richesse expressive et la dynamique informationnelle de la syntaxe wolof

Coordination et Subordination

Coordination

La coordination en wolof permet de relier des mots, syntagmes ou propositions. Elle utilise différentes conjonctions selon le type de relation exprimée, dans ce travail nous nous limitons à la conjonction *ak* [conjonction : et] utilisée pour coordonner différents éléments, par exemple : deux noms, deux groupes nominaux, ou deux verbes. Elle marque l'addition, comme "et" en français.

- Coordination de deux noms (personnes) :
Góor bi [nom : l'homme] *ak* [conjonction : et] *jigéen bi* [nom : la femme] *ñëw nañu* [verbe : sont venus].
L'homme et la femme sont venus.
- Coordination de deux objets/choses :
Teere bi [nom : le livre] *ak* [conjonction : et] *njàngat bi* [nom : la leçon].
Le livre et la leçon.
- Coordination de deux actions (verbes) :
Damay lekk [verbe : je mange] *ak* [conjonction : et] *naan* [verbe : boire].
Je mange et je bois.
- Coordination de deux groupes nominaux (personnes ou objets) :
Xale yi [nom : les enfants] *ak* [conjonction : et] *góor yi* [nom : les hommes] *dem nañu* [verbe : sont partis].
Les enfants et les hommes sont partis.
- Coordination de deux adjectifs (caractéristiques associées au même nom) :
Jigéen bi [nom : cette femme] *rafet* [adjectif : belle] *ak* [conjonction : et] *baax* [adjectif : bon] *la def* [verbe : être].
Cette femme est belle et gentille.

Éclipse et distribution du verbe en wolof

En wolof, des phénomènes d'élision du verbe et de distribution interviennent dans les constructions coordonnées. Ces phénomènes permettent d'éviter la répétition inutile du verbe tout en conservant la clarté syntaxique et sémantique.

1. Élipse du verbe dans la coordination

Principe :

En Wolof, lorsque deux sujets ou deux propositions sont coordonnés avec la conjonction **ak** (*et*), le verbe de la première proposition peut être sous-entendu dans la seconde. Cela évite de répéter le verbe inutilement.

Exemple détaillé :

- Phrase complète :
Xale bi [*l'enfant, défini*] **dafa tuuti** [*est petit*] **ak papa ji** [*le père, défini*] **tamitt dafa tuuti** [*aussi est petit*]
- Phrase avec élision :
Xale bi dafa tuuti ak papa ji [*le verbe « dafa tuuti » est sous-entendu ici*]

Explication :

- Ici, le verbe **dafa tuuti** (*être petit*) est exprimé explicitement pour **Xale bi**, mais il est sous-entendu pour **papa ji**.
- La conjonction **ak** relie les deux sujets. Le sens du verbe de la première proposition est automatiquement **distribué** à la seconde.
- Ce mécanisme est appelé **élision du verbe**, et il est très courant pour rendre les phrases plus fluides en Wolof.

2. Distribution du verbe sur plusieurs adjectifs

Principe :

Quand un sujet est associé à plusieurs adjectifs coordonnés avec **ak**, le verbe peut apparaître une seule fois, **au début**, et s'appliquer à tous les adjectifs. On parle alors de distribution du verbe.

Exemples :

1. **Dafa rafet, baax ak dal.**

(*Il/elle est beau/belle, gentil et calme.*)

- **dafa** [*il/elle est*] apparaît **une seule fois**.
- Les adjectifs **rafet** [*beau/belle*], **baax** [*gentil*] et **dal** [*calme*] sont coordonnés avec **ak**, et le verbe s'applique à tous.

2. **Raaya senegal bi** [*le drapeau du Sénégal*] **dafa** [*est*] **nëtëx** [*vert*] **ak** [*et*] **mboq** [*jaune*] **ak** [*et*] **xonq** [*rouge*]
(*Le drapeau du Sénégal est vert, jaune et rouge.*)

- Ici, le verbe **dafa** est exprimé une seule fois.
- La conjonction **ak** relie les adjectifs **nëtëx**, **mboq**, **xonq**, et le verbe **dafa** est implicitement distribué sur chacun.

Explication :

- Cette stratégie évite de répéter le verbe devant chaque adjectif.
- Elle rend la phrase plus naturelle et fluide en Wolof.
- On peut noter que l'ordre des adjectifs et la coordination par **ak** sont essentiels pour la compréhension.

Ces exemples montrent comment le wolof exploite l'éllision et la distribution pour des expressions plus fluides et naturelles, similaires aux phénomènes observés dans d'autres langues.

Subordination

En wolof, plusieurs types de subordination existent (temps, cause, condition, etc.). Pour les besoins de ce mémoire, nous nous concentrons uniquement sur les subordonnées relatives et complétives jouant un rôle de modifieur de nom ou de constructeur de sujet. Ces subordonnées relatives correspondent à l'utilisation des pronoms « qui » (sujet) et « que » (objet).

En français, les subordonnées relatives s'introduisent avec des pronoms invariables comme *qui* ou *que*. Ces pronoms ne changent pas selon le nombre ou le type de l'antécédent :

- *La femme **qui** parle est ma tante.*
- *Les femmes **qui** parlent sont sœurs.*
- *Le livre **que** j'ai lu est intéressant.*
- *Les livres **que** j'ai lus sont intéressants.*

Le pronom relatif reste le même (*qui*, *que*), quel que soit le cas.

En wolof, au contraire, les relatives s'introduisent avec des particules comme **bu**, **gi**, **ju**, **yi**, **etc.** Ces particules :

- S'accordent avec l'antécédent en nombre : singulier (*bu*, *gi*, *ju*) vs pluriel (*yi*, *ñi*, *etc.*).

- Varient selon la classe nominale du nom (par ex. certains noms prennent *b-*, d'autres *g-*, d'autres encore *j-*).

Cela signifie que le marqueur relatif en wolof concentre plusieurs informations : la fonction de l'antécédent dans la subordonnée (sujet = « qui », objet = « que »), le nombre de l'antécédent et sa classe nominale.

Exemple 1 : relative sujet (équivalent de « qui »)

- **Soxna bu xam Xadim**

→ « La femme qui connaît Xadim »

Analyse :

- *soxna* = femme
- *bu* = relatif sujet, singulier, classe b-
- *xam* = connaître
- *Xadim* = nom propre

Ici, l'antécédent est au singulier et le relatif prend la forme **bu**, équivalent de « qui ».

- **Soxna yi xam Xadim**

→ « Les femmes qui connaissent Xadim »

Au pluriel, la particule relative change : **bu** devient **yi**. C'est une différence importante avec le français, où *qui* resterait inchangé.

Exemple 2 : relative objet (équivalent de « que »)

- **Njàngat gi ma def**

→ « La leçon que j'ai faite »

Analyse :

- *njàngat* = leçon
- *gi* = relatif objet, singulier, classe g-
- *ma def* = que j'ai faite

Ici, l'antécédent est l'objet du verbe *def* (« faire »). Le relatif **gi** correspond à « que ».

- **Njàngat yi ma def**

→ « Les leçons que j'ai faites »

Au pluriel, le relatif devient **yi**. En français, *que* ne varierait pas.

Exemple 3 : autre classe nominale (j-)

- **Jàmm ju ñu gën a bëgg**

→ « La paix qu'ils préfèrent »

Analyse :

- *jàmm* = paix (classe j-)
- *ju* = relatif singulier de la classe j-
- *ñu gën a bëgg* = ils préfèrent

Ici, on voit clairement l'influence de la classe nominale : l'antécédent appartient à une autre classe, donc le relatif prend la forme **ju**.

Exemple 4 : relatif avec un pluriel de personnes

- **Xale ñi ñu xam**

→ « Les enfants qui savent »

Analyse :

- *xale* = enfant
- *ñi* = relatif pluriel (classe des humains)
- *ñu xam* = ils savent

Ce cas montre que le système ne repose pas uniquement sur *yi* au pluriel, mais varie selon la classe et le type de nom (ici, les humains → ñi).

Discussion

Le système des relatives en wolof se distingue de celui du français par sa **flexibilité morphologique** :

- En français, *qui* et *que* sont **invariables**.
- En wolof, les particules relatives **s'accordent** avec l'antécédent en **nombre** (singulier/pluriel) et en **classe nominale**.

Ce mécanisme rend la langue :

- plus précise morphologiquement, puisque le relatif reflète directement le type et le nombre de l'antécédent ;
- plus économique, car un seul morphème concentre la fonction relative, le nombre et la classe.

En résumé :

- **bu** = « qui » au singulier (classe b-)
- **yi** = « qui / que » au pluriel (classe b-)
- **gi** = « que » au singulier (classe g-)
- **ju** = relatif singulier pour la classe j-
- **ñi** = relatif pluriel pour les humains

Ce système montre que, même si on peut traduire directement *bu* par « qui » et *gi* par « que », le wolof intègre aussi des informations morphologiques qui lui sont propres.

2. Morphologie

2.1. Personnes

En wolof, les noms désignant des personnes (humains) peuvent être des noms communs ou des noms propres.

- Les noms communs désignent des catégories générales de personnes.
Exemple : **xale** [nom : enfant], **góor** [nom : homme], **jigéen** [nom : femme].

- Les noms propres désignent des individus spécifiques.
Exemple : **Youssou** [nom propre], **Aïda** [nom propre].

Dans une phrase, les noms communs sont généralement accompagnés d'un déterminant, tandis que les noms propres peuvent apparaître sans déterminant.

- Exemple (nom commun) : **Xale bi** [nom : l'enfant].
L'enfant.
- Exemple (nom propre) : **Youssou ñëw na** [verbe : Youssou est venu].
Youssou est venu.

Les noms désignant des personnes jouent un rôle fondamental dans la structure des phrases en wolof, où l'accord avec le verbe (en nombre surtout) est essentiel. Cependant, contrairement à des langues comme le français, ces noms ne portent pas de marque spécifique indiquant le **genre** (masculin ou féminin).

2.2. Genre

Le wolof ne distingue pas le masculin du féminin comme c'est le cas en français. Cependant, certaines distinctions peuvent être marquées lexicalement, c'est-à-dire par le choix du mot ou par le contexte. Dans la majorité des cas, le même adjectif, article ou verbe est utilisé pour les deux genres.

- **Exemple d'adjectif : rafet** [adjectif : beau / belle]
- **Góor bi dafa rafet / Jigéen bi dafa rafet**
(L'homme est beau / La femme est belle)
- **Góor bi dafa soon / Jigéen bi dafa soon**
(L'homme est fatigué / La femme est fatiguée)
- **Xale bi, góor bi, jigéen bi, teere bi** → même article **bi** utilisé pour tous

- **Exemple avec un article :**

xale bi [nom : l'enfant].

L'enfant (fille ou garçon) est parti(e).

góor bi [nom : l'homme].

L'homme est venu.

jigéen bi [nom : la femme].

La femme est venue.

teere bi [nom : le livre].

Le livre est sur la table.

Dans tous ces cas, l'article bi reste invariable, indépendamment du genre du nom qu'il accompagne. Cela illustre bien que le wolof n'intègre pas la notion de genre grammatical dans son système d'articles, contrairement au français où l'article change entre "le" et "la" selon le genre du nom.

En résumé, la gestion du genre en wolof repose davantage sur le contexte que sur la morphologie. C'est une différence majeure avec le français, où le genre est systématiquement marqué dans les noms, adjectifs, articles et accords grammaticaux

2.3. Nombre

Le wolof ne marque pas le pluriel par des suffixes ajoutés aux noms, mais utilise des déterminants et des pronoms pluriels pour signaler la pluralité.

- Exemple avec un nom de personne :

Góor bi [nom : l'homme], **Góor yi** [nom : les hommes].

L'homme / Les hommes.

- Exemple avec un verbe :

Ñëw na [verbe : il/elle est venu(e)], **Ñëw nañu** [verbe : ils/elles sont venus].

Il/elle est venu(e) / Ils/elles sont venus.

- Exemple avec un objet :

teere bi [nom : le livre], **teere yi** [nom : les livres].

Le livre / Les livres.

- Exemple avec un aliment :

ceeb bi [nom : le riz], **ceeb yi** [nom : les plats de riz].

Le riz / Les plats de riz.

- Exemple avec un nom commun :

xale bi [nom : l'enfant], **xale yi** [nom : les enfants].

L'enfant / Les enfants.

- Exemple avec une action collective :
liggéey na [verbe : il/elle a travaillé], **liggéey nañu** [verbe : ils/elles ont travaillé].
Il/elle a travaillé / Ils/elles ont travaillé.

Dans chaque cas, c'est le déterminant (**bi/yi**) ou le pronom (**na/nañu**) qui indique le nombre (singulier/pluriel), alors que le nom lui-même reste invariable.

3. Polysémie

Certains mots en wolof peuvent avoir plusieurs significations selon le contexte.

- Exemple : **Soxna** [nom : femme / épouse].
- Exemple : **xale** [verbe : enfant / jeune].

4. Rôles thématiques

Le wolof exprime les relations entre les éléments d'une phrase à travers la structure de la phrase et l'utilisation de particules spécifiques. Les rôles thématiques les plus courants incluent l'agent, le patient et le bénéficiaire.

- **Agent** : Il désigne celui ou celle qui réalise l'action dans la phrase.
 - Exemple : **Ñoom** [pronom : eux] **dañuy joxe** [verbe : ils/ elles donnent] **xaliss** [nom : argent]. (*Eux, ils donnent de l'argent.*)
- **Patient** : Il représente l'entité qui subit l'action.
 - Exemple : **Xaliss bi** [nom : argent] **lañu jox** [verbe : ont donné]. (*C'est l'argent qu'ils ont donné.*)
- **Bénéficiaire** : Il indique celui ou celle qui bénéficie de l'action.
 - Exemple : **Kër gi** [nom : maison] **lañuko defaral** [verbe : ont réparé pour]. (*C'est la maison qu'ils ont réparée pour lui/elle.*)

Dans ces constructions, l'ordre des mots ainsi que l'utilisation de particules comme **lañu** ou **dañuy** permettent de bien identifier les différents rôles thématiques dans la phrase.

5. Marqueurs temporels et suffixe "Y"

Le wolof utilise le **suffixe "y"** pour marquer l'aspect progressif (action en cours) dans les verbes d'action.

- Exemple : **Damay jang** [je suis en train d'étudier].
- Exemple : **Mangi jang** [j'étudie].

En conclusion, la morphologie du wolof repose sur des mécanismes relativement simples et réguliers, ce qui en fait un terrain particulièrement intéressant pour toute analyse linguistique formelle. La distinction entre noms propres et noms communs est claire, sans variations morphologiques complexes. L'absence de marquage grammatical du genre simplifie également la catégorisation, en réduisant le nombre de traits morphosyntaxiques à prendre en compte. Enfin, la gestion du nombre par le biais de déterminants et de pronoms, plutôt que par des flexions sur le nom, rend les structures morphologiques du wolof prévisibles et compatibles avec une modélisation computationnelle dans le cadre d'un travail en traitement automatique des langues.

Chapitre 4 : Les grammaires catégorielles

1. Présentation générale

Une des applications concrètes du TALN est l'analyse morphosyntaxique des langues. Plusieurs approches existent, dont un modèle qui substitue un calcul mathématique inferentiel logique à une analyse purement linguistique fondée sur l'intrication linéaire des unités lexicales. Ce modèle fait partie des grammaires catégorielles, et son avantage est sa capacité à s'adapter à divers contextes.

Les grammaires catégorielles trouvent leurs origines dans les travaux fondateurs de Kazimierz Ajdukiewicz [3] et Yehoshua Bar-Hillel [5], à l'origine de la AB-grammar, fondée sur l'application fonctionnelle. Cette base a été considérablement enrichie par Joachim Lambek [24], qui introduisit le calcul des résidus, ouvrant la voie à une approche plus puissante et déductive des structures syntaxiques.

Dans les années 1970, Richard Montague [25] a montré que l'on pouvait relier syntaxe et sémantique de façon compositionnelle, en s'appuyant sur la logique formelle, ce qui a contribué à poser les fondements de la sémantique formelle. À partir des années 1980, Mark Steedman [29] [30] a développé la Grammaire Catégorielle Combinatoire (Combinatory Categorical Grammar, CCG), une extension de la grammaire catégorielle intégrant des règles de combinaison plus souples, adaptées aux constructions syntaxiques naturelles.

En 1995, Jean-Pierre Desclés et Ismaïl Biskri [13] ont proposé une version dite applicative de ce formalisme, appelée Grammaire Catégorielle Combinatoire Applicative (GCCA). Cette approche s'inscrit dans le cadre plus général de la Grammaire Applicative et Cognitive (GAC), qui articule différents niveaux de représentation du langage (phénotype, génotype, cognitif) et vise à modéliser les opérations mentales impliquées dans la production et l'interprétation linguistique. La GCCA repose sur la logique combinatoire de Curry et Schönfinkel, intégrant des combinateurs logiques (comme B, C, S) pour formaliser les règles de dérivation syntaxique.

Grâce à sa forte lexicalisation, sa modularité et sa capacité à représenter simultanément la structure syntaxique et la composition sémantique, la GCCA constitue un outil particulièrement adapté aux besoins du TALN, notamment pour les langues peu dotées. C'est dans ce cadre que nous l'utiliserons pour l'analyse syntaxique du wolof dans le présent mémoire.

2. Principes fondamentaux des grammaires catégorielles

Ce chapitre introduit les principes fondamentaux des grammaires catégorielles, en mettant l'accent sur la Grammaire Catégorielle Combinatoire (GCC) et sa version applicative (GCCA), dans une perspective orientée vers le traitement automatique des langues (TALN). La GCCA constitue une version applicative de la CCG développée par Mark Steedman [29] [30], intégrant la logique combinatoire pour une modélisation computationnelle renforcée.

Nous présenterons les fondements formels, les règles de dérivation, les combinateurs logiques, ainsi que leur lien avec la représentation sémantique. Cependant, une attention particulière sera portée à l'intérêt de la GCCA pour les langues peu dotées comme le wolof.

2.1.1. Principe de base

En GCC, chaque mot lexical reçoit une **catégorie** qui indique son comportement syntaxique. Il existe deux grands types de catégories :

- **Catégories atomiques (primitives) :**
 - **S** : phrase
 - **NP** : syntagme nominal
 - **N** : nom
 - **PP** : syntagme prépositionnel
 - **Adj** : adjectif
 - **Adv** : adverbe
 - **INF** : infinitif
- **Catégories complexes** : elles sont formées à partir des catégories primitives à l'aide d'opérateurs directionnels / et \.

Elles indiquent qu'un mot se comporte comme une fonction prenant un argument à gauche ou à droite.

Exemples de catégories complexes :

| Élément | Catégorie | Explication |
|--------------|--------------|--|
| Le | NP/N | attend un nom à droite pour former un NP |
| grand | N/N | modifie un nom à droite |
| mange | (S\NP)/NP | attend un complément objet (NP) à droite, puis un sujet à gauche |
| dort | S\NP | attend un sujet à gauche |
| dans | (PP/NP) | attend un nom à droite pour former un PP |
| vite | (S\NP)(S\NP) | modifie un verbe à sa gauche (adverbe) |

Tableau 2 : Exemples de catégories fonctionnelles complexes en grammaire catégorielle

2.1.2. Les opérateurs / et \

En grammaire catégorielle, les opérateurs permettent de définir la direction d'attente d'un constituant :

- / : attend un argument à **droite**.
(A/B signifie : prends un B à ta droite et deviens un A).
- \ : attend un argument à **gauche**.
(A\B signifie : prends un B à ta gauche et deviens un A).

Ces opérateurs transforment la composition syntaxique en une véritable opération fonctionnelle, proche du raisonnement mathématique, où chaque élément est une fonction cherchant son argument dans une direction donnée.

Exemple 1 – opérateur / (attente à droite)

Phrase :

Pierre mange une pomme.

Typage simplifié :

- **mange** : (S\NP)/NP
(un verbe transitif attend d'abord un complément d'objet NP à droite, puis un sujet à gauche).

- **une pomme** : NP
- **Pierre** : NP

Dérivation :

1. **Mange** ((S\NP)/NP) prend **une pomme** (NP) à sa droite → donne S\NP.
2. **Pierre** (NP) combine ensuite avec S\NP → donne une phrase complète S.

Discussion : Ici, l'opérateur / indique que le verbe *mange* doit d'abord trouver son complément à droite avant de pouvoir s'associer à son sujet. Sans cet opérateur, il serait impossible de distinguer l'ordre des constituants dans la phrase.

Exemple 2 – opérateur \ (attente à gauche)

Phrase :

Marie dort.

Typage simplifié :

- **dort** : S\NP
(un verbe intransitif attend uniquement un sujet NP à gauche pour former une phrase).
- **Marie** : NP

Dérivation :

1. **Marie** (NP) se combine avec **dort** (S\NP).
2. Le schéma (S\NP) + NP → S donne une phrase complète S.

Discussion : Ici, l'opérateur \ reflète la dépendance du verbe intransitif envers un sujet **placé à gauche**. Le prédicat « dort » ne peut pas exister seul : il nécessite un argument qui le précède, ce qui correspond bien à la structure française.

Exemple 3 – combinaison des deux opérateurs

Phrase :

Paul lit un livre.

Typage simplifié :

- **lit** : (S\NP)/NP
- **un livre** : NP
- **Paul** : NP

Dérivation :

1. **lit** ((S\NP)/NP) prend **un livre** (NP) → devient S\NP.

2. **Paul** (NP) s'associe à gauche → donne la phrase complète **S**.

Discussion : Cet exemple montre comment un verbe transitif combine les deux opérateurs :

- / pour intégrer son objet à droite,
- \ pour intégrer ensuite son sujet à gauche.

Cela illustre la nature « fonctionnelle » des constituants : le verbe agit comme une fonction qui s'applique d'abord à son objet, puis à son sujet.

Exemple 4 – adjectif attributif (illustration de \)

Phrase :

L'enfant est fatigué.

Typage simplifié :

- **fatigué** : S\NP
- **l'enfant est** (sujet + copule) : NP

Dérivation :

1. **fatigué** (S\NP) attend un sujet à gauche.
2. **l'enfant est** (NP) fournit ce sujet → la phrase devient complète (S).

Discussion : Ici encore, l'opérateur \ montre que l'adjectif attributif ou le prédicat dépend toujours d'un sujet placé à gauche.

Conclusion

Ces exemples montrent clairement :

- que l'opérateur / organise les arguments attendus **à droite** (objets, compléments),
- tandis que l'opérateur \ organise les arguments attendus **à gauche** (sujets, attributs).

Cette distinction permet de formaliser rigoureusement la structure des phrases françaises en termes de **fonctions et d'arguments**, ce qui est au cœur de la grammaire catégorielle.

2.2. Les règles combinatoires

2.2.1. Application fonctionnelle

Cette règle exprime la manière la plus simple de combiner une fonction et son argument. Elle existe en deux variantes :

- **Application fonctionnelle à droite (>) :**

$$(A/B) + B \Rightarrow A (>)$$

Une catégorie fonctionnelle (A/B) attend un argument de type B à sa droite et produit une expression de type A.

- **Application fonctionnelle à gauche (<) :**

$$B + (A\B) \Rightarrow A (<)$$

Une catégorie fonctionnelle (A\B) attend un argument de type B à sa gauche et produit une expression de type A.

2.2.2. Composition fonctionnelle

La composition permet de créer une nouvelle fonction à partir de deux fonctions compatibles, sans saturer immédiatement l'argument.

- **Composition fonctionnelle à droite (>B) :**

$$(A/B) + (B/C) \Rightarrow A/C (>B)$$

Une fonction (A/B) rencontre une fonction (B/C). Au lieu de donner directement un résultat, elles se combinent pour former une nouvelle fonction de type A/C.

- **Composition fonctionnelle à gauche (<B) :**

$$(B\C) + (A\B) \Rightarrow A\C (<B)$$

De manière symétrique, la fonction (A\B) rencontre une fonction (B\C). Leur combinaison produit une nouvelle fonction de type A\C.

2.2.3. Élévation de type (type raising)

L'élévation de type transforme un argument en fonction afin de permettre des combinaisons plus souples, par exemple dans les cas d'inversion syntaxique, de focalisation ou de questions.

- **Élévation de type à droite (>T) :**

$$NP \Rightarrow S/(S\NP) (>T)$$

Un syntagme nominal (NP) est retypé en fonction, attendant une prédication verbale (S/NP) à sa droite pour former une phrase (S).

- **Élévation de type à gauche (<T) :**

$NP \Rightarrow S(S/NP) (<T)$

De manière symétrique, NP est transformé en fonction qui attend une prédication verbale (S/NP) à sa gauche pour former une phrase (S).

2.2.4. Coordination

La coordination est un mécanisme essentiel permettant de combiner deux constituants de même type syntaxique. En grammaire catégorielle, elle est formalisée par une catégorie générique :

- **et** : $(X \backslash X) / X$

Cette règle exprime que la conjonction prend un constituant X à sa droite, et attend ensuite un autre X à sa gauche, pour donner un constituant X complet.

Exemple : *Jean et Marie dorment.*

- Jean : NP
- Marie : NP
- et : $(NP \backslash NP) / NP$
- dorment : S/NP

La combinaison conduit à une phrase complète (S). Cette règle illustre la souplesse de la GCC et de la GCCA dans la gestion de structures non triviales.

2.2.5. Subordination

La GCC permet également de traiter des subordinées relatives ou circonstancielles, en attribuant des catégories spécifiques aux introducteurs (relatifs, conjonctions).

Exemple : *l'homme qui dort*

- qui dort : $(NP \backslash NP)$
- l'homme : NP
→ NP

Ce mécanisme montre la capacité des grammaires catégorielles à intégrer la hiérarchie syntaxique des phrases complexes.

Toutes ces règles constituent le noyau de la grammaire catégorielle combinatoire. Elles permettent de combiner les catégories lexicales de manière systématique afin d'obtenir des structures

syntaxiques bien formées. Les exemples d'application concrète de ces règles seront présentés dans la section suivante.

2.3. Exemples d'analyse syntaxique

Exemple 1 : Le garçon mange une pomme.

| Mot | Catégorie |
|--------|-----------|
| Le | NP/N |
| garçon | N |
| mange | (S\NP)/NP |
| une | NP/N |
| pomme | N |

Tableau 3 : Exemple d'analyse syntaxique (1)

Dérivation :

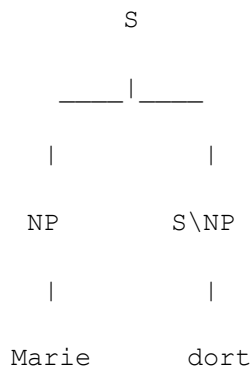
Le garçon = NP

une pomme = NP

mange (une pomme) = S\NP

(NP) (S\NP) \Rightarrow S

Arbre syntaxique



Ici, le verbe intransitif « dort » a pour catégorie S\NP : il attend uniquement un sujet nominal à gauche. Le constituant « Marie » fournit cet argument, ce qui suffit à former une phrase complète. Cet exemple illustre la simplicité de l'analyse des verbes intransitifs dans la grammaire catégorielle,

Exemple 3 : L'enfant lit un livre.

| Mot | Catégorie |
|--------|-----------|
| L' | NP/N |
| enfant | N |
| lit | (S\NP)/NP |
| un | NP/N |
| livre | N |

Tableau 5 : Exemple d'analyse syntaxique (3)

Dérivation :

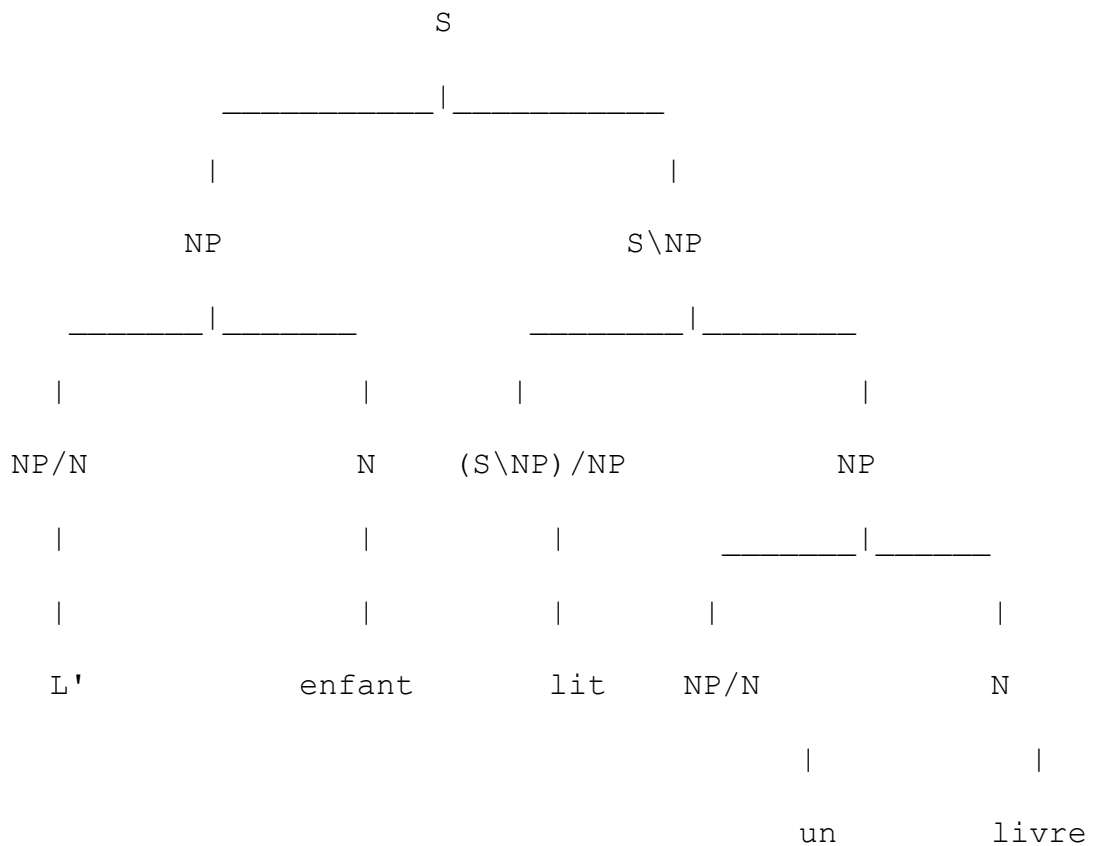
L'enfant = NP

un livre = NP

lit un livre = S\NP

NP (S\NP) ⇒ S

Arbre syntaxique :



Cet exemple reprend la structure de la phrase simple avec verbe transitif. Le verbe « lit » agit comme une fonction binaire : il attend un complément d’objet à droite (« un livre »), puis un sujet à gauche (« l’enfant »). La composition conduit à une phrase complète S. L’arbre montre le double mouvement fonctionnel des opérateurs (/** et **).

Exemple 4 : Le grand chien aboie.

| Mot | Catégorie |
|-------|-----------|
| Le | NP/N |
| grand | N/N |
| chien | N |
| aboie | S\NP |

Tableau 6 : Exemple d’analyse syntaxique (4)

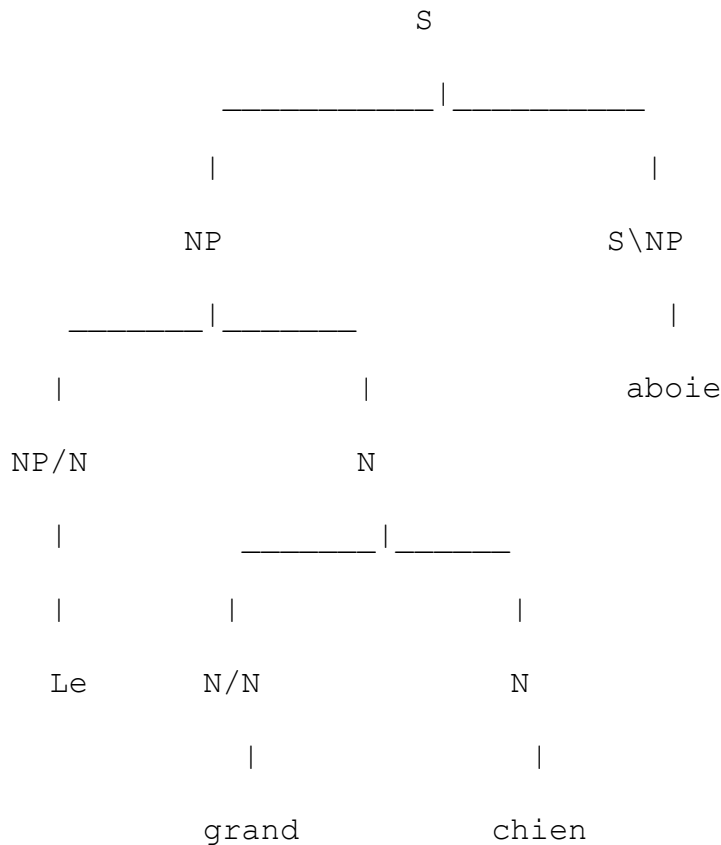
Dérivation :

grand chien = N

Le (grand chien) = NP

NP (S\NP) ⇒ S

Arbre syntaxique :



Cet exemple montre l'intégration d'un adjectif dans un syntagme nominal. « grand » est de catégorie N/N, ce qui signifie qu'il attend un nom à sa droite pour donner un nom. Ainsi, « grand » + « chien » forment le nom complexe « grand chien », qui devient un groupe nominal grâce à l'article « le ». Ce groupe nominal fournit l'argument au verbe intransitif « aboie » (S\NP). L'arbre illustre clairement le rôle des « modificateurs » dans la structure hiérarchique.

2.4. De la GCC à la GCCA (version applicative)

2.4.1. Motivation

La GCCA (Grammaire Catégorielle Combinatoire Applicative) introduit une modélisation plus fonctionnelle adaptée aux outils de TALN. Elle repose sur l'usage de combinateurs logiques, inspirés de la logique combinatoire.

2.4.2 Les combinateurs principaux

| Combinateur | Rôle | Forme | Fonction |
|-------------|--------------|-----------------------|--|
| B | Composition | $B f g x = f (g x)$ | Associe deux fonctions successives |
| C | Permutation | $C f x y = f y x$ | Inverse deux arguments |
| S | Distribution | $S f g x = f x (g x)$ | Applique deux fonctions sur un même argument |
| I | Identité | $I x = x$ | Ne change rien |

Tableau 7 : Présentation des combinateurs principaux

2.4.3. Utilisation des combinateurs en GCCA

Les combinateurs B, C, S et I ne sont pas seulement des abstractions logiques : en GCCA, ils généralisent et systématisent les règles de dérivation.

- B (composition) : formalise la composition fonctionnelle ($>B$, $<B$).
- C (permutation) : permet de changer l'ordre des arguments, utile dans les langues à ordre libre.
- S (distribution) : capture le mécanisme de type raising et d'application multiple.
- I (identité) : reflète les cas où un constituant se transmet tel quel.

Cette intégration rapproche la GCCA de la logique combinatoire de Curry et Schönfinkel, en renforçant son potentiel computationnel.

2.5. Avantages de la GCCA en TALN

- Lexicalisation forte : la grammaire est portée par le lexique, pas par des règles syntaxiques lourdes.
- Analyse modulaire et transparente.
- Adaptée aux outils informatiques comme NLTK, OpenCCG.
- Modélisation multilingue, adaptée aux langues peu dotées comme le wolof.

2.6 Représentation sémantique (λ -calcul)

La GCCA permet une composition syntaxique et sémantique en parallèle, notamment via le λ -calcul.

Exemple : Pierre aime Marie.

- Pierre : $\lambda P. P(\text{pierre})$
- aime : $\lambda x. \lambda y. \text{aimer}(y, x)$
- Marie : marie

→ Interprétation finale : aimer (pierre, marie)

Lien syntaxe-sémantique :

La GCCA ne s'arrête pas à la structure syntaxique : chaque étape de la dérivation correspond à une étape de calcul sémantique. Cette propriété de compositionnalité (héritée de Montague) permet de relier syntaxe et sémantique de manière transparente.

Exemple :

- Jean : $\lambda P. P(\text{jean})$
- mange : $\lambda x. \lambda y. \text{manger}(y, x)$
- une pomme : pomme

→ Résultat : manger (jean, pomme)

Cette double lecture syntaxe-sémantique constitue un des atouts majeurs de la GCCA dans le TALN.

2.7. Comparaison entre GCC et GCCA

| Aspect | GCC (Steedman) | GCCA (Desclés & Biskri) |
|------------|--|---|
| Fondements | Application fonctionnelle et composition | Intégration de la logique combinatoire (B, C, S, I) |
| Portée | Principalement syntaxique | Syntaxe + calcul logique |

| | | |
|------------------|---|---|
| Orienta- tion | Grammaire catégorielle combinatoire (CCG) | Version applicative, intégrée à la Grammaire Applicative et Cognitive (GAC) |
| Usage TALN | Parsing et dérivation syntaxique | Parsing + représentation cognitive et computationnelle |
| Souplesse | Gestion des constructions SVO, coordination | Adaptée aux langues à ordre libre, focalisation, inversion |

Tableau 8 : Tableau comparatif entre la GCC et la GCCA

2.8. Polyvalence linguistique

Un des points forts de la GCCA est sa capacité à s'adapter à des langues très différentes.

- En anglais ou en français, elle modélise efficacement les structures SVO.
- Dans des langues à ordre plus libre (latin, wolof, japonais), les mécanismes de type raising, composition et coordination permettent de conserver une analyse robuste.
- La GCCA est donc un outil privilégié pour le TALN multilingue et pour l'étude des langues peu dotées.

Chapitre 5 : Analyse du wolof au moyen des grammaires catégorielles

Le wolof, langue nigéro-congolaise parlée principalement au Sénégal, se caractérise par une relative régularité de son ordre des mots (principalement SVO), l'usage systématique de déterminants postposés, ainsi que par des mécanismes d'organisation informationnelle tels que la focalisation et la topicalisation. Ces propriétés en font une langue particulièrement intéressante pour une formalisation en Grammaire Catégorielle Combinatoire (GCC).

Le présent chapitre illustre, à travers des exemples, comment la GCC permet de modéliser les principales structures syntaxiques du wolof. Nous montrerons notamment comment sont analysées les phrases déclaratives simples, la focalisation, la topicalisation, les relatives, ainsi que la coordination.

4.1. Phrase déclarative simple

Pour l'application de la Grammaire Catégorielle Combinatoire (GCC) au wolof nous avons choisi d'analyser à partir d'exemples simples et représentatifs. Le wolof est relativement régulier dans son ordre des mots et dans l'usage des déterminants, il se prête bien à une formalisation compositionnelle.

Les analyses qui suivent illustrent la manière dont la GCC permet de décrire les principales structures syntaxiques du wolof.

Exemple 1 : phrase déclarative simple

Xale bi lekk ceeb bi

→ « L'enfant qui a mangé le riz »

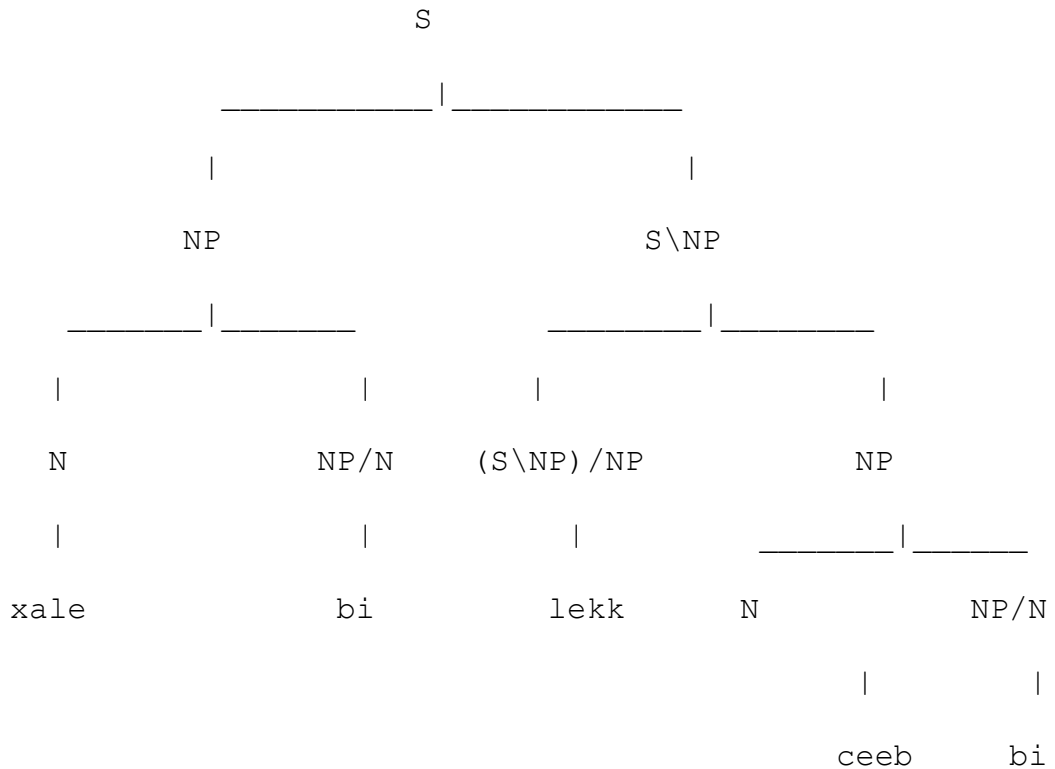
Catégories lexicales :

- *xale* : N
- *bi* : NP/N
- *ceeb* : N
- *bi* : NP/N
- *lekk* : (S\NP)/NP

Dérivation :

- *xale bi* = NP
- *ceeb bi* = NP
- *lekk ceeb bi* = S\NP
- *NP (S\NP)* ⇒ S

Arbre :



Discussion :

La structure est de type SVO, parfaitement régulière. Le déterminant postposé (*bi*) s'analyse comme une fonction NP/N, ce qui permet de dériver directement un syntagme nominal. La GCC modélise donc efficacement cette structure de base.

4.2. Focalisation du sujet

Exemple :

Moom mo lekk ceeb bi

→ « C'est lui qui a mangé le riz »

Catégories lexicales :

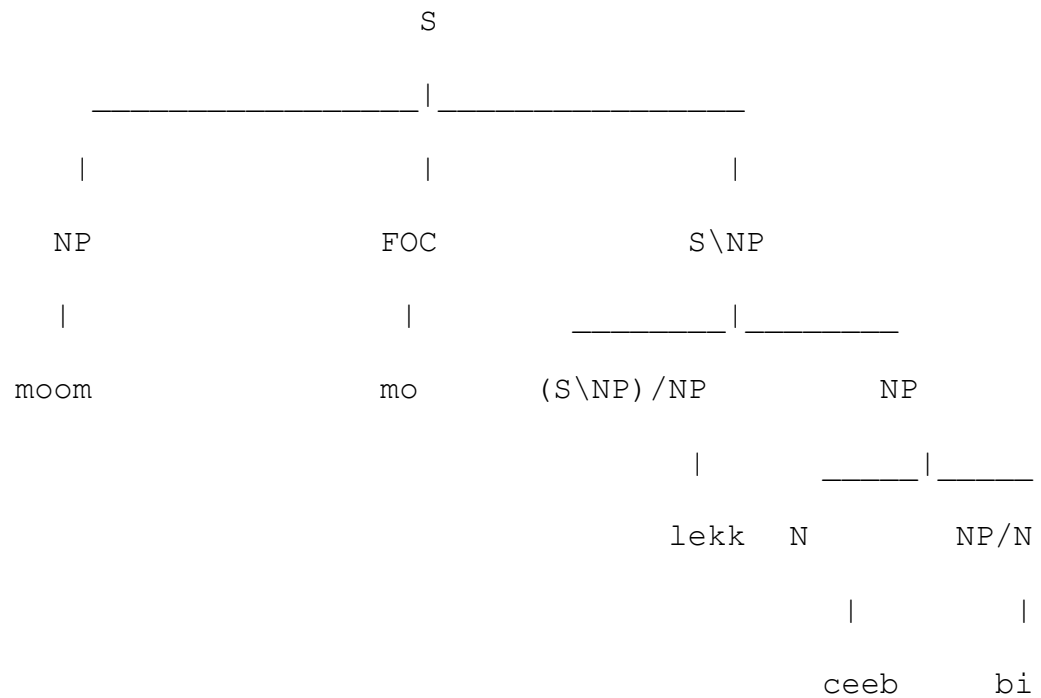
- *moom* : NP
- *mo* : FOC (S/(S\NP))
- *lekk* : (S\NP)/NP
- *ceeb bi* : NP

Dérivation simplifiée :

- *moom* = NP
- *mo moom* = S/(S\NP) appliqué à NP

- *lekk ceeb bi* = S\NP
- Application \rightarrow S

Arbre :



La focalisation est représentée par l'ajout de la particule *mo*, qui reconfigure la structure informationnelle de la phrase. La GCC, grâce à des catégories comme $S/(S\NP)$, permet de formaliser cette mise en relief de manière systématique.

4.3. Topicalisation

Exemple :

Ceeb bi, mako lekk

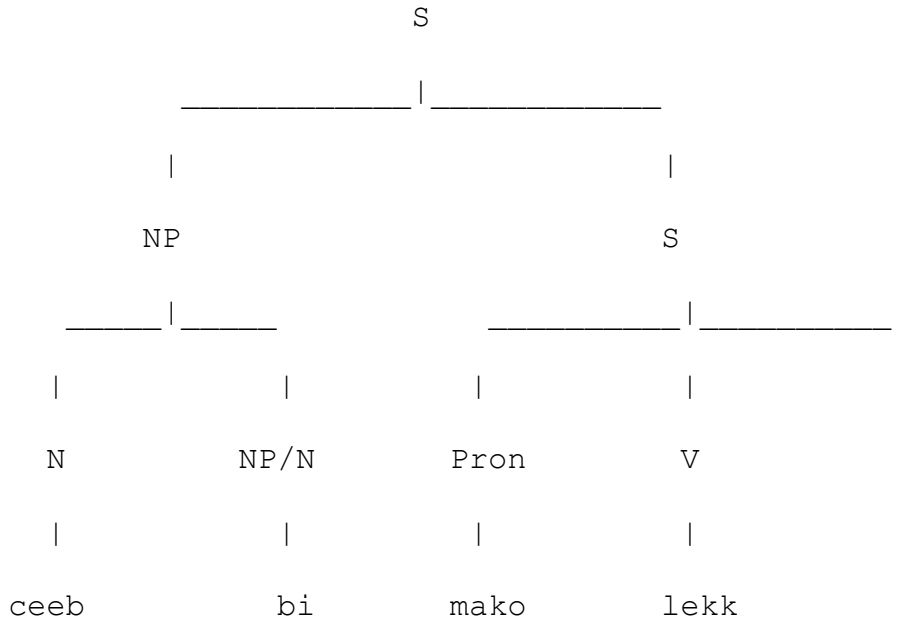
\rightarrow « Le riz, c'est moi qui l'ai mangé »

Catégories lexicales :

- *ceeb bi* : NP
- *mako* : pronom objet = $(S\NP)/V$
- *lekk* : V

(Dérivation simplifiée – le pronom reprend l'élément topicalisé.)

Arbre :



La topicalisation met en avant un thème déjà connu. Le groupe nominal *ceeb bi* est déplacé en tête, et repris par le pronom *mako*. La GCC rend compte de ce mécanisme en considérant la reprise pronominale comme un élément syntaxique obligatoire pour maintenir la cohérence.

4.4. Relative simple

Exemple :

Soxna su xam Xadim

→ « Une femme qui connaît Xadim »

Catégories lexicales :

- soxna : N
- su : REL = (NP\NP)/S
- xam : (S\NP)/NP
- Xadim : NP

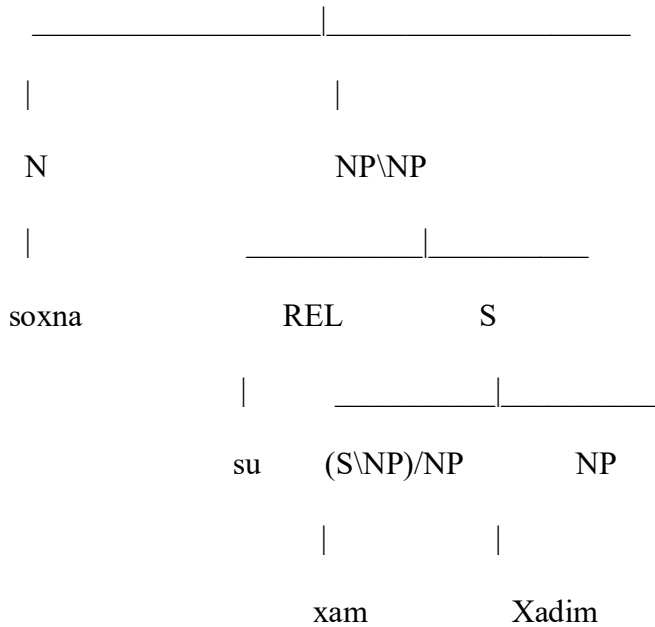
Dérivation :

- *xam Xadim* = S\NP
- *su xam Xadim* = NP\NP
- *soxna* = N
- *soxna su xam Xadim* = NP

- $NP + (NP \backslash NP) \Rightarrow NP$

Arbre :

NP



Le marqueur relatif *su* fonctionne comme un équivalent de « qui » en français. Il introduit une relative où l'antécédent est sujet. La GCC formalise ce lien via la catégorie $(NP \backslash NP)$, qui enrichit un nom initial par une proposition relative.

4.5. Coordination

La coordination est un mécanisme productif en wolof, permettant de relier deux constituants de même type. Comme en français, elle est représentée par une catégorie générique :

- ak (« et ») : $(X \backslash X) / X$

Exemple:

Momar ak Alassane jelun xaliss bi

→ « Momar et Alassane n'ont pas pris l'argent »

Catégories lexicales :

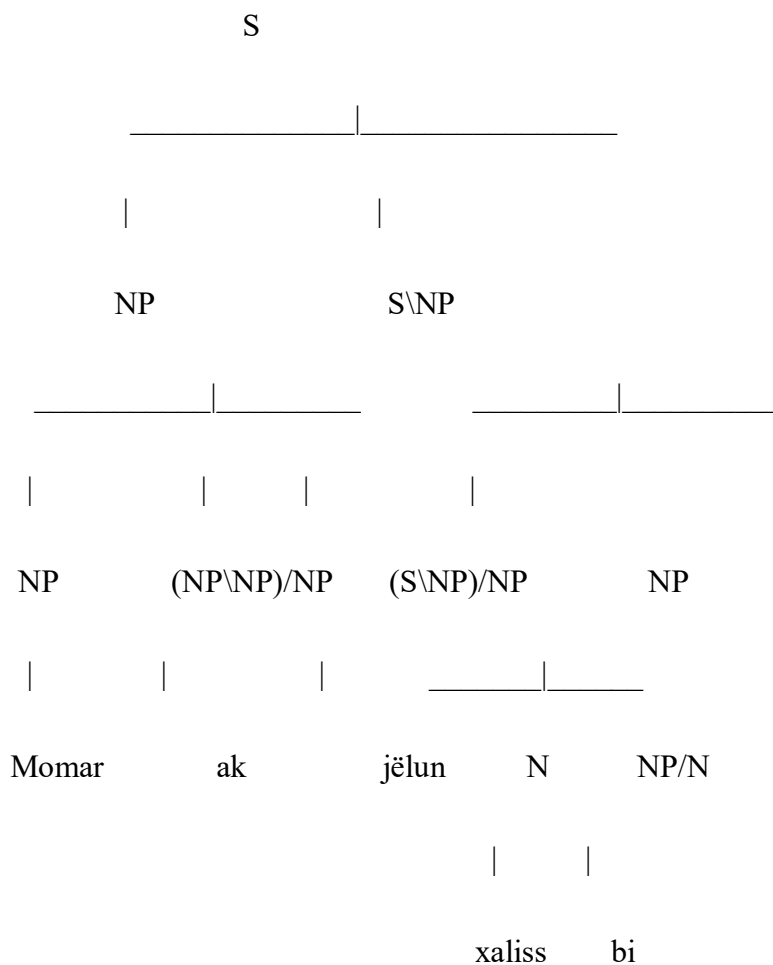
- Momar : NP

- Alassane : NP
- ak : (NP\NP)/NP
- jëlun : (S\NP)/NP (verbe transitif, « prendre » à la forme négative)
- xaliss : N
- bi : NP/N

Dérivation :

- xaliss bi = NP
- jëlun xaliss bi = S\NP
- ak + Alassane = (NP\NP)
- Momar + (ak Alassane) = NP (NP coordonné)
- NP + (S\NP) \Rightarrow S

Arbre syntaxique :



|
Alassane

La coordination illustre la capacité de la GCC à combiner des arguments multiples de manière flexible, tout en maintenant une dérivation compositionnelle.

Ces exemples montrent que le wolof, grâce à sa régularité morphosyntaxique (ordre SVO, déterminants postposés, particules focales, marqueurs relatifs, coordination), se prête particulièrement bien à une analyse en GCC. Ce formalisme permet de modéliser aussi bien la syntaxe de base (phrases déclaratives), les phénomènes informationnels (focalisation, topicalisation), que les structures complexes (relatives, coordination). La GCC constitue ainsi un outil rigoureux, compositionnel et adapté au TALN pour l'étude du wolof, en mettant en évidence ses spécificités tout en exploitant un cadre théorique général

Chapitre 6 : implantation

1. Choix des technologies

¹1.1. Choix du langage de programmation : Python

Le choix du langage Python [18] pour l'implémentation de ce travail s'explique par ses avantages techniques, scientifiques et communautaires dans le domaine du Traitement Automatique des Langues (TAL). Python s'est imposé comme une référence incontournable grâce à sa syntaxe épurée, son accessibilité, et sa polyvalence dans les tâches de modélisation linguistique.

L'objectif principal de ce mémoire étant de modéliser et d'analyser des structures syntaxiques du wolof à l'aide de la Grammaire Catégorielle Combinatoire Applicative (GCC), Python s'avère être un choix pertinent à plusieurs niveaux :

- Il permet de manipuler aisément des structures de données linguistiques complexes (comme les catégories CCG) via des structures de données natives (dictionnaires, listes, objets, etc.).
- Il propose des bibliothèques spécialisées, notamment NLTK (Natural Language Toolkit), qui intègre un sous-module dédié à la grammaire catégorielle combinatoire (`nltk.ccg`), facilitant l'écriture, l'analyse et la dérivation de catégories syntaxiques.
- Il offre une intégration native avec les formats standardisés comme JSON, utilisé dans ce projet pour représenter dynamiquement le lexique wolof de manière extensible et modulaire.
- Il est compatible avec des outils de visualisation simples comme Tkinter, permettant de construire une interface utilisateur graphique favorisant l'interaction humaine-machine et la visualisation des dérivations syntaxiques sous forme d'arbres.
- Il bénéficie d'une communauté scientifique active dans les domaines du TAL, de la linguistique computationnelle, et du deep learning, offrant ainsi un écosystème en constante évolution.

¹ <https://docs.python.org/3/library/tkinter.html>

Enfin, la compatibilité de Python [18] avec d'autres outils tels que Jupyter, NumPy, pandas ou matplotlib offre une flexibilité analytique importante pour la visualisation des résultats ou l'évaluation de la performance du système linguistique. C'est donc un choix stratégique, aussi bien pour le prototypage rapide que pour le déploiement d'un système expérimental fiable.

1.2. Choix de la bibliothèque : Étude comparative entre NLTKCCG et OpenCCG

Présentation des deux bibliothèques

Dans ce projet, deux bibliothèques fondées sur la Grammaire Catégorielle Combinatoire (CCG) ont été évaluées : NLTKCCG [26] et OpenCCG [31].

NLTKCCG (Combinatory Categorical Grammar module de NLTK) est une extension expérimentale développée dans le cadre du projet NLTK [26] (Université de Pennsylvanie), qui permet la définition de lexiques, l'attribution de catégories, et l'analyse syntaxique selon les principes de la CCG. Il est possible d'écrire des grammaires dans des formats simples, et d'obtenir des dérivations syntaxiques composées.

OpenCCG, développé à l'Université d'Édimbourg, est une implémentation plus avancée, dédiée principalement à la génération de langage naturel (NLG), et permet des analyses riches sémantiquement. Il repose cependant sur un écosystème plus complexe, basé sur Java et Scala, et vise davantage des applications industrielles de génération textuelle.

Tableau comparatif :

| Critère | NLTKCCG | OpenCCG |
|--------------------------|-------------------------------------|-------------------------------------|
| Langage | Python | Java (avec spécifications en Scala) |
| Objectif principal | Analyse syntaxique CCG | Génération de langage naturel |
| Public cible | Linguistes, chercheurs, étudiants | Développeurs en NLP/NLG |
| Facilité d'apprentissage | Élevée (grâce à NLTK) | Moyenne à faible (Java + Scala) |
| Multilinguisme | Oui (via grammaires personnalisées) | Principalement orienté anglais |

| | | |
|------------------------------|------------------------------------|---|
| Spécification des grammaires | Fichiers simples (texte ou JSON) | Fichiers Scala, XML plus complexes |
| Visualisation de dérivation | Arbres générés via NLTK ou Tkinter | Interfaces personnalisées ou fichiers XML |

Tableau 9 : Tableau comparatif entre NLTKCCG et OpenCCG

Justification du choix de NLTKCCG

Plusieurs éléments motivent l'adoption de NLTKCCG dans ce projet :

- Il est **entièrement écrit en Python**, ce qui assure une cohérence technologique avec le reste de l'implémentation.
- Il permet une **intégration directe du lexique wolof sous forme de fichier JSON**, sans dépendance complexe.
- Il facilite la **visualisation et l'annotation des arbres syntaxiques**, particulièrement utiles pour l'analyse linguistique.
- Il est **adapté à la création de prototypes linguistiques personnalisés**, notamment pour des langues peu dotées.
- Sa **syntaxe de définition grammaticale** est plus souple et accessible que celle d'OpenCCG, ce qui permet une évolution incrémentale du système.

Compte tenu de ces critères, NLTKCCG [26] s'avère être le choix optimal pour un projet de recherche exploratoire sur la GCC appliquée au wolof, dans une perspective d'analyse syntaxique plutôt que de génération de texte.

En conclusion, Le choix de Python comme langage d'implémentation, associé à la bibliothèque NLTKCCG, repose sur une double exigence : accessibilité technique et pertinence scientifique. Ce binôme technologique permet d'assurer une modélisation fidèle des structures syntaxiques du wolof dans le cadre rigoureux de la grammaire catégorielle, tout en offrant une flexibilité pour la visualisation, l'expérimentation et l'évaluation. La décision de ne pas recourir à des outils plus complexes comme OpenCCG s'inscrit dans une logique de prototypage rapide, adapté à une langue peu dotée nécessitant des adaptations progressives et contrôlées.

L'efficacité de l'analyse syntaxique dans le cadre de la Grammaire Catégorielle Combinatoire Applicative (GCC) repose en grande partie sur la définition précise des catégories grammaticales attribuées aux mots. C'est pourquoi la modélisation rigoureuse du lexique wolof constitue une étape centrale dans notre démarche. La section suivante sera donc consacrée à la construction d'un lexique typé, où chaque mot sera analysé en fonction de son comportement syntaxique, c'est-à-dire en fonction du type d'arguments qu'il attend à gauche ou à droite dans une phrase, conformément aux principes de la GCC.

1.3. Workflow de l'analyse

Après avoir présenté le choix du langage et de la bibliothèque, il est nécessaire de décrire le processus général suivi lors de l'analyse syntaxique. Ce processus peut être représenté sous forme de workflow, qui illustre les différentes étapes allant de la phrase en entrée jusqu'à la validation ou au rejet de son analyse. Le schéma ci-dessous met en évidence les interactions entre le dictionnaire typé, les règles catégorielles et le module d'analyse, ainsi que les sorties possibles (phrase correcte ou incorrecte).

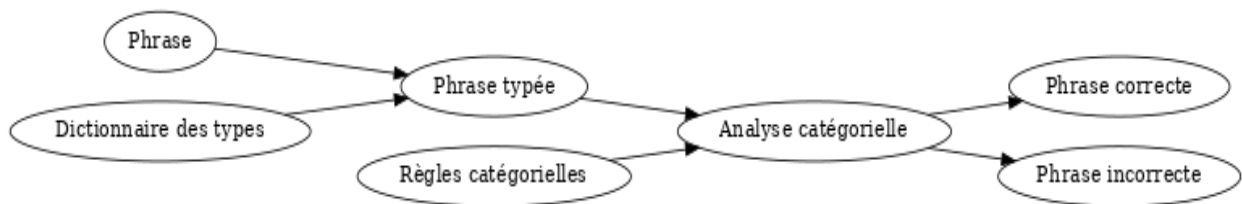


Figure 1: Workflow de l'implantation de l'analyse syntaxique

2. Modélisation du lexique wolof selon la Grammaire Catégorielle Combinatoire (GCC)

Principes de typage

Dans une grammaire catégorielle combinatoire (GCC), chaque mot d'une langue est vu comme une expression logique typée : soit un argument (catégorie atomique), soit une fonction qui attend un ou plusieurs arguments pour produire une nouvelle expression linguistique. Le typage permet de modéliser de façon formelle les propriétés combinatoires des mots et la structure syntaxique des phrases.

Le système repose sur :

- des **catégories atomiques** :
 - N (nom),
 - NP (Pour **Noun Phrase** - syntagme nominal),
 - S (phrase),
 - PP (syntagme prépositionnel), etc.
- des **catégories fonctionnelles orientées** :
 - X/Y : fonction qui attend un argument de type Y à sa droite,
 - X\Y : fonction qui attend un argument de type Y à sa gauche.

Ainsi, un verbe transitif comme “manger” aura pour type (S\NP)/NP : il prend un **objet direct** à droite, puis un **sujet** à gauche pour produire une phrase S.

Dans le cadre de ce mémoire, nous utilisons ce formalisme pour modéliser le lexique de base du **wolof**, une langue SVO (Sujet–Verbe–Objet) caractérisée par :

- des **marqueurs aspectuo-temporels fusionnés** au sujet (ex. *dama, danga, dafa*),
- des **déterminants postposés** (ex. *bi, mi*) qui s’appliquent au nom,
- des **prépositions et locatifs** (ex. *ci, fii*) dont le typage dépend de la structure argumentale.

Constitution du lexique typé à partir d’exemples

a) Phrase 1 : *Dama wañi xaliss bi*

Traduction : j’ai diminué l’argent

| Mot | Traduction | Catégorie syntaxique | Type GCC | Justification |
|--------|-------------|----------------------|-----------|--|
| dama | je | Sujet pronominal | NP | Sujet autonome |
| wañi | diminuer | Verbe transitif | (S\NP)/NP | Attend un objet à droite (xaliss bi), puis un sujet à gauche |
| xaliss | argent | Nom commun | N | Nom simple |
| bi | le (défini) | Déterminant postposé | NP/N | Attend un nom à droite pour former un NP |

Tableau 10 : Exemple d'analyse syntaxique en wolof selon la GCC (1)

Analyse compositionnelle :

- xaliss : N
- bi :NP/N
→ bi xaliss : NP
- wañi:(S\NP)/NP
→ wañi (bi xaliss) : S\NP
- dama:NP
→ dama (wañi bi xaliss) : S

b) Phrase 2 : *Dangay jaang ?*

Traduction : tu étudies ?

| Mot | Traduction | Catégorie syntaxique | Type GCC | Justification |
|--------|---------------------------------|------------------------------|----------|--------------------------|
| dangay | tu (marqueur de sujet + aspect) | Pronom sujet avec auxiliaire | NP | Sujet au présent |
| jaang | étudier | Verbe intransitif | S\NP | Attend un sujet à gauche |

Tableau 11 : Exemple d'analyse syntaxique en wolof selon la GCC (2)

Analyse compositionnelle :

1. jaang : S\NP
2. dangay : NP
→ dangay jaang : S

c) Phrase 3 : *Dafa tok ci bay bi*

Traduction : il/elle est assis(e) sur le banc

| Mot | Traduction | Catégorie syntaxique | Type GCC | Justification |
|------|------------------------------|---------------------------|-----------|---|
| dafa | il/elle (sujet + auxiliaire) | Sujet autonome au présent | NP | Sujet implicite au présent |
| tok | être assis | Verbe locatif | (S\NP)/PP | Attend un complément locatif à droite |
| ci | sur/dans | Préposition | PP/NP | Attend un syntagme nominal à droite pour former un PP |
| baŋ | banc | Nom | N | Nom commun |
| bi | le (article défini) | Déterminant postposé | NP/N | Forme un NP avec un nom |

Tableau 12 : Exemple d'analyse syntaxique en wolof selon la GCC (3)

Analyse compositionnelle :

1. baŋ : N, bi : NP/N → bi baŋ : NP
2. ci : PP/NP, bi baŋ : NP → ci bi baŋ : PP
3. tok : (S\NP)/PP + ci bi baŋ : PP → tok ci baŋ bi : S\NP
4. dafa : NP → dafa tok ci baŋ bi : S

d) Tableau synthétique des nouveaux mots typés

| Mot wolof | Traduction | Catégorie | Type GCC | Remarque |
|-----------|-------------------|-------------------|-----------|------------------------------|
| dama | je | Pronom sujet | NP | Sujet autonome |
| dangay | tu (présent) | Pronom sujet | NP | Forme fléchie du sujet |
| dafa | il/elle (présent) | Pronom sujet | NP | Utilisé avec verbes à l'état |
| wañi | diminuer | Verbe trans. | (S\NP)/NP | Attend objet puis sujet |
| jaang | étudier | Verbe intransitif | S\NP | Sujet uniquement |
| tok | être assis | Verbe locatif | (S\NP)/PP | Attend un locatif |
| ci | sur/dans | Préposition | PP/NP | Forme un PP avec un NP |
| xaliss | argent | Nom | N | Nom abstrait |
| baŋ | banc | Nom | N | Objet concret |

| | | | | |
|----|-------------|-------------|------|----------------------|
| bi | le (défini) | Déterminant | NP/N | Déterminant postposé |
|----|-------------|-------------|------|----------------------|

Tableau 13 : Exemple d'analyse syntaxique en wolof selon la GCC (4)

Enregistrement dans un fichier JSON exploitable

Le fichier JSON utilisé dans le système d'analyse regroupe chaque entrée lexicale avec son mot, sa traduction, sa catégorie grammaticale, son type logique GCC, et un exemple d'utilisation. Ce format facilite l'intégration automatique dans le programme Python.

Exemple de structure JSON :

```
{
  "mot": "dama",
  "traduction": "je",
  "categorie": "pronom sujet",
  "type": "NP",
  "exemple": "dama wañi xaliss bi"
},
{
  "mot": "wañi",
  "traduction": "diminuer",
  "categorie": "verbe transitif",
  "type": "(S\\NP)/NP",
  "exemple": "dama wañi xaliss bi"
},
{
  "mot": "bi",
  "traduction": "le",
  "categorie": "déterminant défini",
  "type": "NP/N",
  "exemple": "xaliss bi"
},
{
  "mot": "jaang",
  "traduction": "étudier",
  "categorie": "verbe intransitif",
  "type": "S\\NP",
  "exemple": "dangay jaang"
}
]
```

Figure 2 : Exemple de structure du fichier JSON

En conclusion, Cette section a permis de mettre en œuvre les principes de la GCC pour la modélisation syntaxique du wolof, à travers le typage formel de mots courants. Chaque mot a été attribué à une catégorie grammaticale précise, avec des justifications linguistiques détaillées. Cette base lexicale constituée et structurée en JSON constitue le socle nécessaire pour le traitement automatique des phrases dans les sections suivantes.

3. Développement de l'interface d'analyse

3.1. Technologies utilisées

Afin de permettre une interaction directe avec le système d'analyse syntaxique, une interface utilisateur a été développée à l'aide de Tkinter, la bibliothèque graphique native de Python. Ce choix s'explique par sa légèreté, sa compatibilité immédiate avec l'environnement Python, et sa capacité à générer des interfaces simples sans dépendances externes.

Tkinter a été combiné avec le module `nltk.ccg.chart` afin d'interfacer dynamiquement la sélection ou la saisie d'une phrase avec le processus d'analyse syntaxique. L'architecture du code est organisée de manière modulaire autour de quatre fichiers principaux :

- `lexique_loader.py` : module de chargement et formatage du fichier JSON contenant le lexique wolof typé.
- `parser.py` : module d'analyse syntaxique utilisant le parseur CCG de NLTK.
- `ui.py` : interface Tkinter (interface utilisateur graphique).
- `main.py` : point d'entrée du programme, qui orchestre le système.

Cette séparation des responsabilités permet une meilleure lisibilité du code, ainsi qu'une maintenance facilitée en cas d'évolution future du système.

3.2. Structure de l'interface

L'interface graphique développée permet une interaction simple avec le système, sans connaissances techniques préalables. Elle se compose de quatre éléments fonctionnels :

- **Une liste déroulante** contenant les phrases en wolof définies dans le lexique JSON. Elle permet de sélectionner des exemples typés connus du système.
- **Un champ de saisie libre**, pour introduire manuellement de nouvelles phrases à tester.
- **Un bouton "Analyser"**, qui déclenche le processus de dérivation syntaxique.
- **Une zone de résultats**, qui affiche l'arbre de dérivation syntaxique ou un message d'erreur.

L'interface a été pensée dans une optique de démonstration linguistique et pédagogique, facilitant la visualisation des dérivations dans le cadre de phrases simples en wolof.

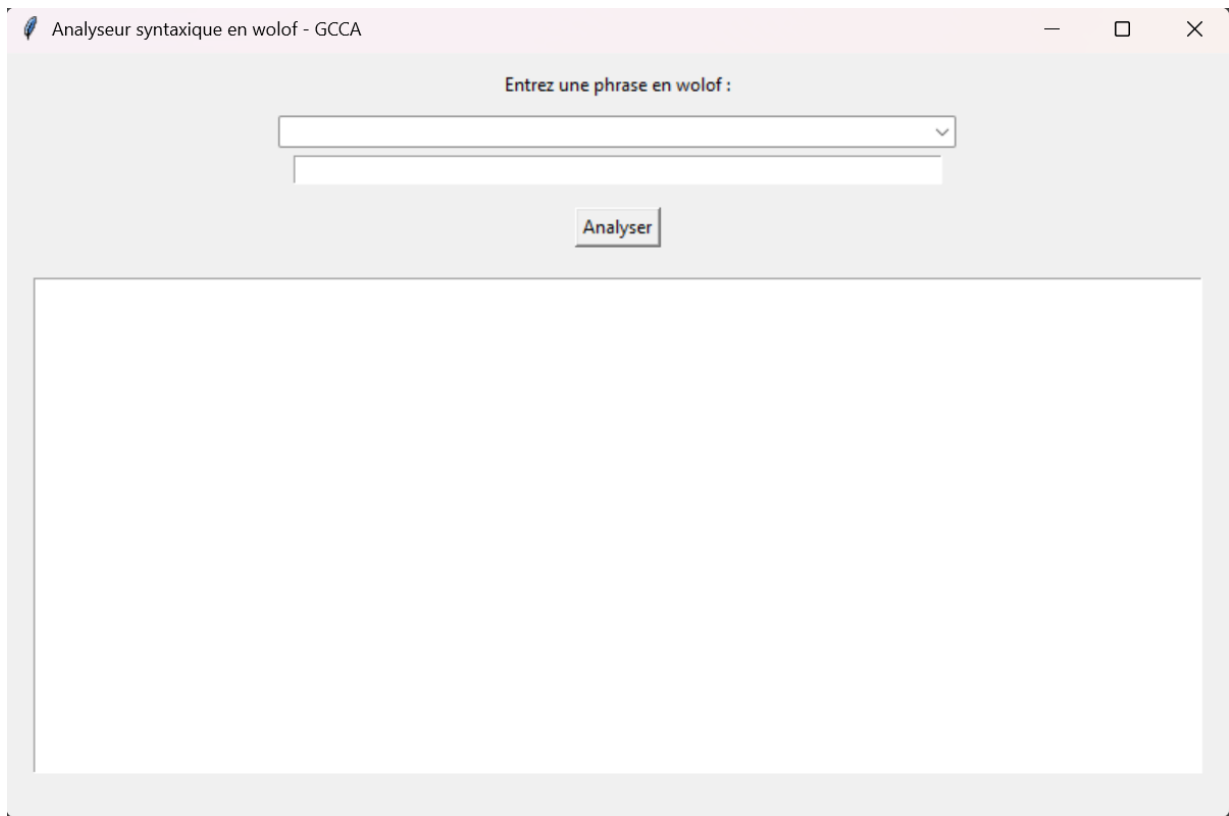


Figure 3 : Interface d'analyse syntaxique développée avec Tkinter

3.3. Exemple d'exécution

Prenons par exemple la phrase suivante :

dama wañi xaliss bi
(*"j'ai diminué l'argent"*)

Une fois cette phrase sélectionnée dans la liste ou saisie manuellement, le système exécute les étapes suivantes :

1. Vérifie la présence de tous les mots dans le lexique.
2. Construit une grammaire CCG dynamique à partir des types définis.
3. Applique le parseur CCGChartParser de NLTK.
4. Affiche, si possible, une dérivation syntaxique cohérente.

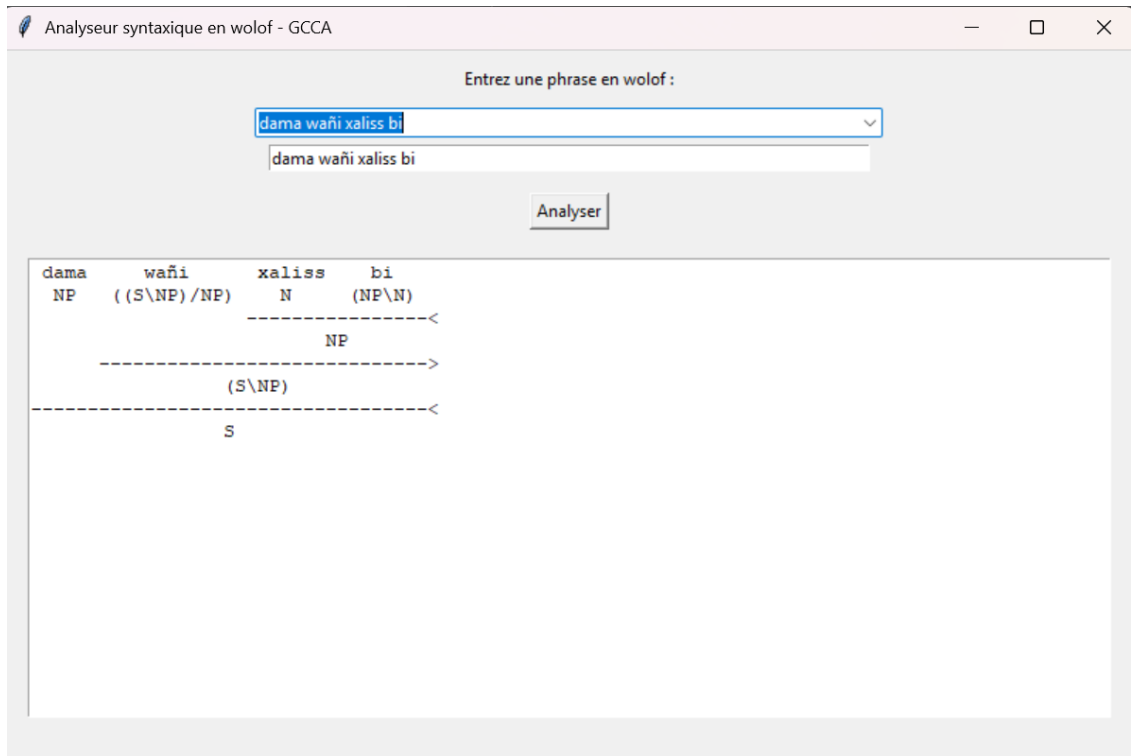
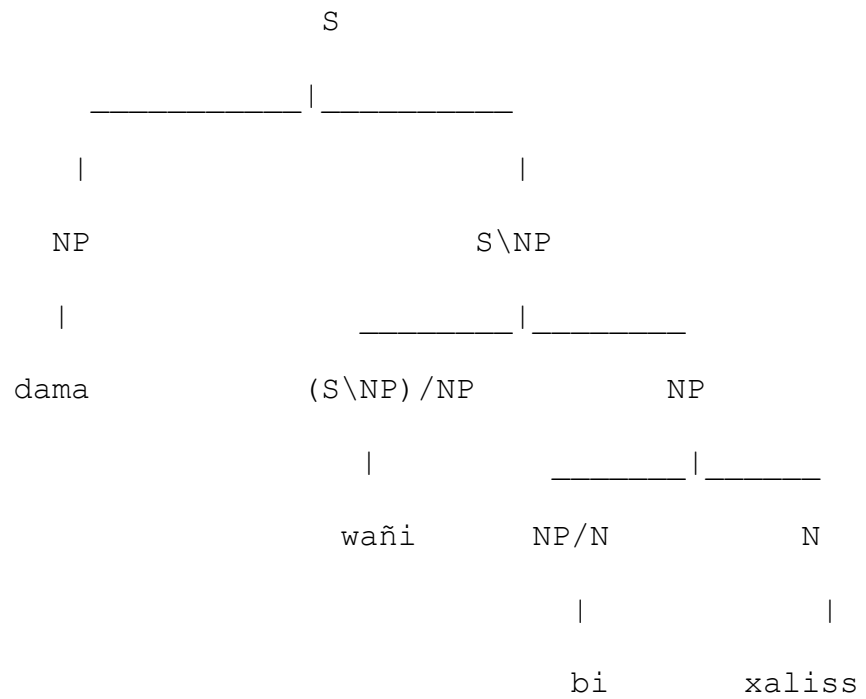


Figure 4 : Exemple d'exécution de l'analyseur syntaxique via l'interface développée sur la phrase « dama wañi xaliss bi »

Extrait de la dérivation produite :



Cette représentation montre que :

- Le déterminant postposé **bi** s'applique au nom **xaliss** pour former un syntagme nominal (**NP**),
- Le verbe transitif **wañi** prend ce **NP** comme complément objet,
- Le pronom **dama** combine ensuite avec le verbe pour donner une phrase complète (S).

3.4. Pipeline de fonctionnement

Le fonctionnement global du système suit un pipeline en cinq étapes :

1. **Chargement du lexique** (via `lexique_loader.py`) : lecture du fichier JSON contenant les mots en wolof et leurs types GCC.
2. **Saisie de la phrase** (via l'interface Tkinter).
3. **Vérification lexicale** : détection de mots inconnus.
4. **Parsing syntaxique** (via `parser.py`) : application des règles de dérivation.
5. **Affichage du résultat** : visualisation textuelle de l'arbre de dérivation ou message d'erreur.

Le schéma suivant résume ce processus :

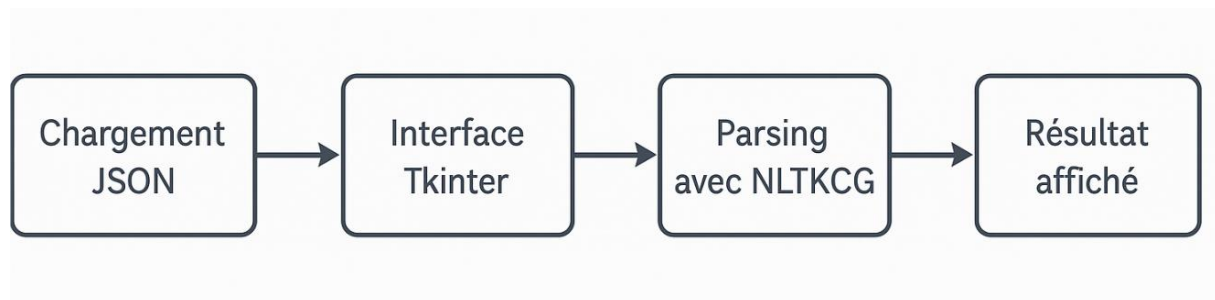


Figure 5 : Schéma global du fonctionnement du système d'analyse syntaxique

Ce chapitre a détaillé l'implantation de l'analyseur syntaxique du wolof en s'appuyant sur la Grammaire Catégorielle Combinatoire (GCC). À travers le choix de Python, l'utilisation de NLTKCCG et la modélisation rigoureuse d'un lexique typé, nous avons construit un prototype opérationnel, capable de générer des dérivations syntaxiques pour différentes phrases en wolof. Cette implantation offre désormais un socle fonctionnel qui servira de base aux expérimentations présentées dans le chapitre suivant, où seront évaluées la pertinence et les limites du modèle sur un ensemble de phrases tests.

Chapitre 7 : Expérimentation

1. Introduction et constitution du corpus

L'objectif principal de cette expérimentation est d'évaluer empiriquement la performance du système d'analyse syntaxique développé dans le cadre de ce mémoire, en vérifiant sa capacité à analyser correctement des phrases en wolof selon le formalisme de la Grammaire Catégorielle Combinatoire (GCC). Cette étape permet non seulement de vérifier la validité des hypothèses de typage préalablement formulées, mais aussi d'identifier les limites linguistiques, formelles ou techniques du prototype.

Le système repose sur un lexique typé structurant les mots en wolof dans un fichier JSON, et utilise un parseur GCC implémenté avec la bibliothèque NLTK. L'évaluation s'appuie sur un corpus de cent phrases construites manuellement, classées selon leur grammaticalité attendue. Chaque phrase est regroupée dans l'une des quatre catégories suivantes :

- Phrases bien construites et acceptées : grammaticalement correctes et analysées avec succès ;
- Phrases bien construites mais non acceptées : grammaticalement correctes, mais rejetées par le système ;
- Phrases mal construites mais acceptées : contenant des anomalies syntaxiques mais validées par le système ;
- Phrases mal construites et non acceptées : incorrectes en wolof et correctement rejetées.

Deux critères ont guidé la constitution du corpus :

(1) La diversité des structures morphosyntaxiques : le corpus inclut des phrases simples (SVO) et des structures impliquant des locatifs, des auxiliaires, des postpositions, ou des constructions verbales composées, de manière à refléter les spécificités syntaxiques du wolof contemporain.

(2) L'évaluation préalable de la grammaticalité : chaque phrase a été jugée par des locuteurs natifs et confrontée à des descriptions linguistiques pour assurer la validité des jugements. Les propriétés de saturation, d'ordre canonique, de postposition déterminative, ou encore de régime prépositionnel ont été prises en compte.

2. Protocole expérimental

Chaque phrase a été introduite dans le système via l'interface d'analyse. Le traitement suit quatre étapes :

1. Chargement des types lexicaux depuis le fichier JSON ;
2. Vérification de la présence de tous les mots dans le lexique ;
3. Tentative d'analyse syntaxique par le parseur GCC ;
4. Affichage du résultat : si une dérivation est trouvée, la phrase est acceptée ; sinon, elle est rejetée.

Chaque résultat est alors confronté à la grammaticalité attendue, ce qui permet de juger de la précision du système et d'identifier les causes d'échec (erreur de typage, absence de mot, composition incorrecte, etc.).

3. Analyse des résultats

3.1. Corpus utilisé

Le corpus utilisé dans ce travail a été entièrement construit par nos soins, car il n'existe pas, à notre connaissance, de base de données ou de corpus en wolof suffisamment adapté aux besoins d'une analyse syntaxique formelle en grammaire catégorielle. Les ressources existantes sont limitées et souvent orientées vers des applications lexicales ou morphologiques (par exemple, dictionnaires numériques ou lexiques bilingues), ce qui ne permettait pas de disposer d'un ensemble de phrases typées pour la modélisation syntaxique.

Nous avons donc élaboré un corpus de 100 phrases originales en wolof, réparties équitablement dans les quatre catégories retenues pour l'expérimentation : phrases bien construites acceptées, bien construites non acceptées, mal construites acceptées, et mal construites non acceptées. Chaque groupe comporte des phrases variées, incluant des structures simples à l'ordre canonique SVO, des verbes transitifs et intransitifs, des constructions prépositionnelles, ainsi que des formes conjuguées typiques du wolof (par ex. *damay*, *dangay*, *dafa*).

Afin de ne pas alourdir le texte, nous ne présentons pas l'intégralité du corpus, mais plutôt un échantillon représentatif de phrases pour chaque catégorie. L'ensemble du corpus, structuré et

annoté, est stocké sous format JSON afin de faciliter son exploitation informatique dans le système d'analyse.

3.2. Phrases bien construites et acceptées

Ci-dessous quelques phrases grammaticalement correctes et analysées sans erreurs :

- damay lekk ceeb bi [je suis en train de manger le riz]
- dama wañi xaliss bi [je réduis/diminue l'argent]
- dangay jaang [tu étudies]
- danga xiif [tu as faim]
- dafa tok ci ban bi [il/elle est assis(e) sur le banc]
- dafaY lekk ceeb bi [il/elle mange le riz]
- dama wañi ndox mi [je réduis l'eau]
- dama mar [j'ai soif]
- daño setani lamb ji [Ils sont partis regarder le match]
- dañoy tek tere bi [ils/elles posent/placent le livre]
- damay tek kass bi [je pose/je place la tasse]
- dafay seet kër gi [il/elle cherche la maison]
- dangay naan mew mi? [est-ce que tu bois le lait ?]
- dafay liggeey ci kër gi [il/elle travaille dans la maison]
- dañuy jaang wolof [ils/elles apprennent/étudient le wolof]
- dama bëgg xale bi [j'aime cet enfant]
- danga tok ci kër gi [tu es resté dans la maison]
- dafay naan ndox mi [il/elle boit l'eau]
- dafa jaay xaliss bi [il/elle a dépensé l'argent]
- dama guiss benn tere fii [j'ai vu un livre ici]
- dangay liggeey ci waxtu wi? [tu travailles à cette heure/pendant ce temps ?]
- damay jaang wolof [je suis en train d'étudier le wolof]
- danga xiif rekk [tu as juste faim]
- dama bëgg tok ci kër gi [je veux rester à la maison]

Ces phrases représentent des constructions standard, avec une structure SVO, postposition du déterminant ("bi", "ji", "mi"), et combinaison correcte des types fonctionnels. L'analyseur identifie avec précision les combinaisons syntaxiques attendues.

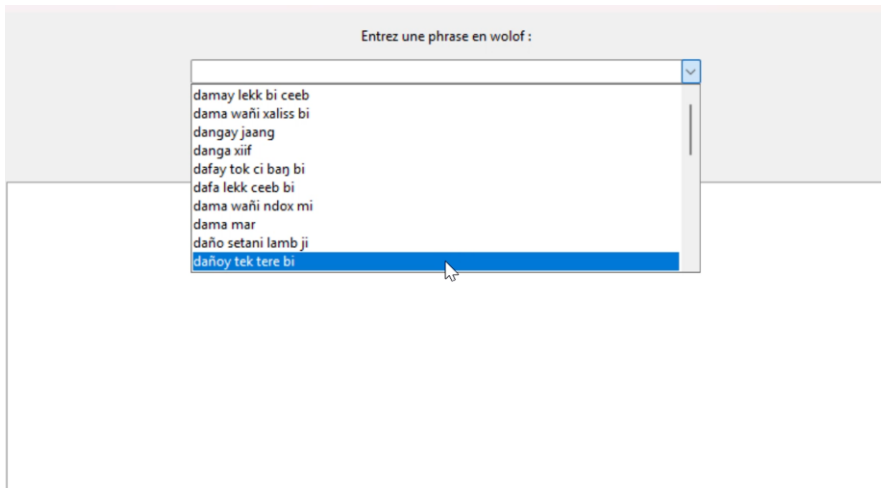


Figure 6 : Sélection d'une phrase via le menu déroulant

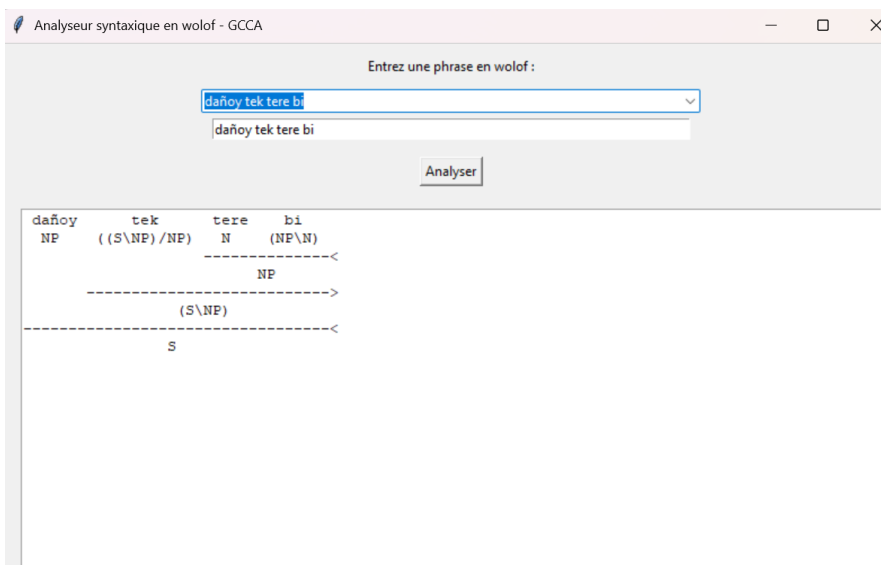


Figure 7 : Exemple d'exécution d'une phrase en wolof bien construite et acceptée par le système

3.3. Phrases bien construites mais non acceptées

Ces phrases sont grammaticalement correctes en wolof, mais ne sont pas analysées ou reconnues par l'analyseur. La raison principale tient au fait que certaines structures ou mots ne figurent pas dans le lexique typé JSON, ou bien n'ont pas reçu un type logique assez précis pour être dérivés par la grammaire catégorielle. Trois difficultés principales se dégagent :

- l'absence ou le mauvais typage de certains éléments fréquents du wolof (*ngi*, *moo*, *naata*, *bëgg*) ;

- la complexité des structures focalisées, interrogatives ou négatives, qui exigent des catégories fonctionnelles supplémentaires (FOC, NEG, Q, etc.) non encore modélisées ;
- l'existence d'expressions figées ou de locutions verbales comme *ma ngi* (« je suis en train de »), qui ne peuvent pas être traitées si elles ne sont pas enregistrées comme unités lexicales.

Exemples :

- *ndax dëgg nga wolof* [Est-ce que tu comprends le wolof ?]
→ non reconnu car la particule interrogative *ndax* et la combinaison avec *dëgg + nga* nécessitent des catégories spécifiques absentes du lexique.
- *ma ngi dem liggeey* [je pars travailler / je vais au travail]
→ non reconnu car *ma ngi* est une locution aspectuelle (« je suis en train de ») qui combine pronom et auxiliaire, absente du lexique.
- *ndax dëgg nga wolof* [est-ce vrai que tu parles wolof ?]
→ non reconnu à cause de la particule interrogative *ndax* ($Q = S/(S \setminus NP)$), qui n'a pas été incluse dans le lexique ou correctement reliée au reste de la phrase.
- *ban waxtu moo jot* [Il est quelle heure ?]
→ non reconnu car *moo* est une particule de focalisation complexe et *ban* est un interrogatif non typé.
- *naata lay jar* [Ça coûte combien ?]
→ non reconnu car *naata* (« combien ») est un quantificateur interrogatif (NP/N) non encore modélisé.
- *baal ma fan moo am wanak* [Excusez-moi, où sont les toilettes ?]
→ non reconnu car la phrase combine impératif (*baal ma* « excuse-moi »), interrogatif (*fan* « où »), et particule *moo*, dépassant la couverture actuelle du modèle.
- *waaw dëgguma* [Oui, je ne comprends pas.]
→ non reconnu à cause de la forme négative *dëgguma* qui n'a pas été intégrée comme entrée lexicale.
- *dangay bëgg jaang* [Tu veux étudier ?]
→ non reconnu car *bëgg* (« vouloir ») n'a pas été traité comme un verbe modal (attendant une proposition).
- *damay seet benn jàngalekat bi* [Je cherche un enseignant.]
→ non reconnu car *seet* (« chercher, voir ») est un verbe transitif bien formé, mais la combinaison avec le quantifieur *benn* (« un ») et le déterminant *bi* n'a pas été correctement modélisée dans le lexique.

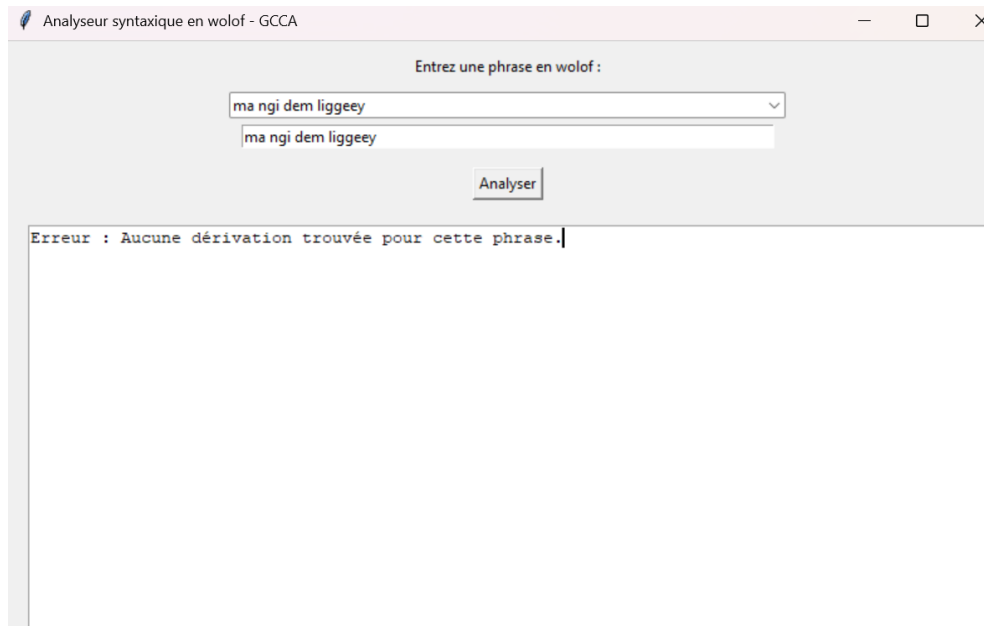


Figure 8 : Exemple d'exécution d'une phrase en wolof bien construite mais non acceptée par le système

En résumé, ces phrases échouent non pas parce qu'elles sont mal formées, mais parce que l'implantation actuelle ne couvre pas encore toutes les subtilités morphosyntaxiques et pragmatiques du wolof.

3.4. Phrases mal construites mais acceptées

Les phrases suivantes ont été acceptées par le système alors qu'elles enfreignent les règles du wolof. Leur reconnaissance par l'analyseur s'explique par le fait que la GCC (telle qu'implémentée ici) valide avant tout la cohérence formelle des catégories syntaxiques, sans intégrer un filtre sémantique ou pragmatique. Autrement dit, le système « croit » qu'une phrase est correcte dès lors que les types se combinent jusqu'à donner une phrase complète (S), même si le résultat est agrammatical ou incohérent en wolof.

- dafa wolof lekk bi [il – wolof – nourriture – la]
- dafay lamb setani ji [il/elle – lutte/match – regarde – le]
- ceeb bi setani dafa [riz – le – regarder – il/elle]
- dama tere tek bi [je – livre – pose – le]
- dafay wañi baŋ gi [il/elle – diminue – banc – le]
- bi xalei tek dafa [enfant – l' – pose – il/elle]
- dafay baŋ wañi bi [il/elle – banc – diminue – le]
- danga setani baŋ bi [tu – regarder – banc – le]

- dafa wolof lekk bi [il/elle – wolof – mange – le]
- banj bi wañi dama [banc – le – diminue – je]

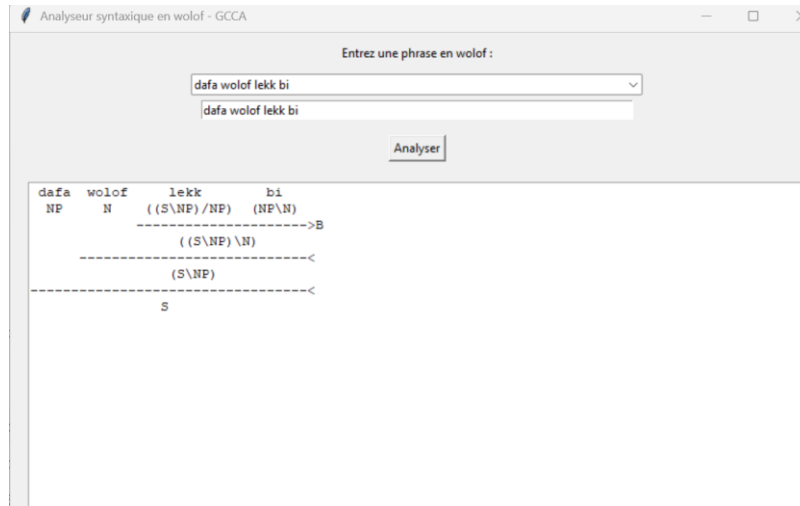


Figure 9 : Exemple d'exécution d'une phrase en wolof mal construite mais acceptée par le système

Ces phrases sont rejetées par la grammaire naturelle du wolof mais acceptées par le parseur, car :

- Le système se limite à la syntaxe formelle des catégories (NP, N, S\NP, etc.),
- Il ne détecte pas les erreurs de position des déterminants postposés (« bi », « ji »),
- Il ne tient pas compte de la plausibilité sémantique (ex. *diminuer un banc*),
- Il ne gère pas encore les contraintes pragmatiques propres au wolof.

En conclusion, ces exemples montrent que la GCC, appliquée de manière purement formelle, peut produire des analyses correctes sur le plan catégoriel, tout en validant des phrases incorrectes en wolof. Cela souligne l'importance d'intégrer à terme des contraintes sémantiques et pragmatiques pour affiner la validité des phrases acceptées.

3.5. Phrases mal construites et non acceptées

Ces phrases illustrent les limites structurelles que la grammaire catégorielle détecte efficacement. Elles montrent que le système rejette automatiquement les séquences qui violent les règles de **composition syntaxique** (ordre des constituants, arguments attendus mais absents,

déterminants mal placés, répétitions incohérentes). Ce rejet est important, car il prouve que l'analyseur ne se contente pas de combiner des catégories formellement, mais respecte aussi la logique combinatoire sous-jacente à la langue.

Exemples représentatifs :

- **Répétitions non structurées :**

- *lekk lekk lekk* [manger – manger – manger]
- *tok tok* [être assis – être assis]

Ces phrases sont rejetées car elles ne contiennent aucun sujet ni objet : les verbes se répètent sans combinaison valide.

- **Ordre des mots illégitime :**

- *ceeb lekk bi* [riz – manger – le]
- *danga danga* [tu es – tu es]

Ici, le déterminant *bi* est mal placé (*ceeb lekk bi*) et la répétition du marqueur verbal ne produit aucune phrase (*danga danga*).

- **Arguments absents ou mal saturés :**

- *xaliss tok ci* [argent – être assis – sur]
- *tok ci bi ban* [être assis – sur – le – banc]

Dans ces cas, les prépositions ou verbes locatifs attendent un complément saturé, mais l'argument reste incomplet, rendant la phrase infaisable.

- **Déterminants ou combinaisons illégitimes :**

- *wañi bi bi* [diminuer – le – le]
- *damay bi ceeb lekk* [je – le – riz – manger]

Ces séquences utilisent plusieurs déterminants pour un seul nom, ce qui viole la structure syntaxique du wolof.

- **Ordre inversé ou sémantiquement incohérent :**

- *waxtu jot bi daño* [temps – atteindre – le – ils]
- *gi kër dangay* [le – maison – tu es]

Ici, l'ordre des constituants produit une phrase grammaticalement absurde : le sujet et le prédicat sont inversés ou mal alignés.

- **Séquences incomplètes ou tronquées :**

- *wanak dafa* [seulement – il/elle est]
- *fi tok bi* [ici – être assis – le]

Ces phrases semblent amorcer une construction mais s'arrêtent avant d'atteindre une structure complète.

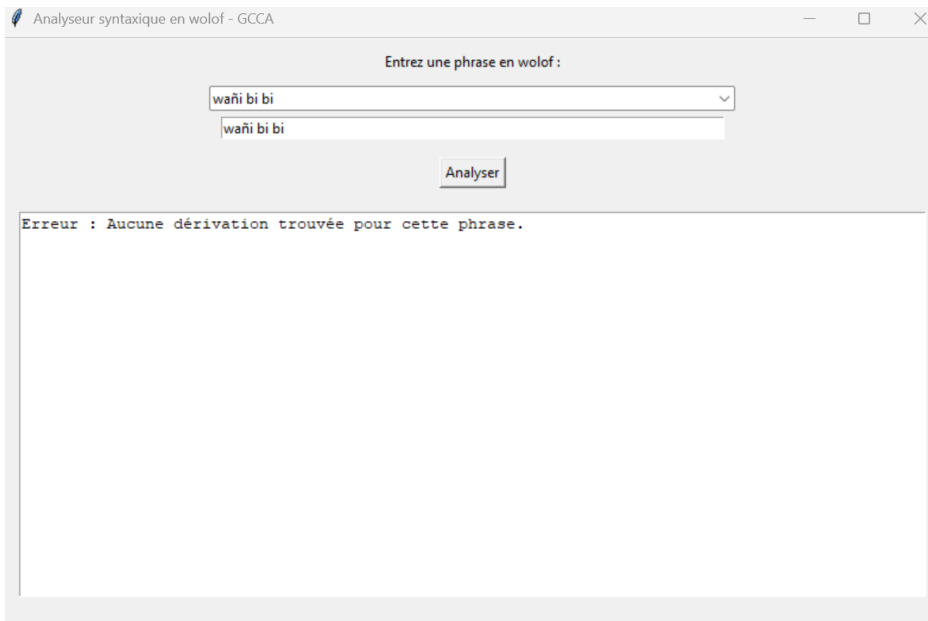


Figure 10 : Exemple d'exécution d'une phrase en wolof mal construite et non acceptée par le système

Ces résultats montrent que le système rejette correctement les phrases sans structure syntaxique valide, qu'il s'agisse de répétitions absurdes, d'arguments manquants, d'ordre des mots illégitime ou de séquences inachevées. Cette capacité de filtrage confirme la robustesse de l'approche par GCC : le parseur ne valide pas mécaniquement toute suite de mots, mais impose un respect strict des contraintes combinatoires. Cela fournit une base solide pour comparer, dans le chapitre suivant, les limites réelles du système face aux phrases grammaticalement plus complexes.

3.6. Synthèse des résultats

L'analyse a porté sur un corpus de cent (100) phrases. Parmi celles-ci, cinquante (50) ont été acceptées par le système et cinquante (50) ont été rejetées. Plus précisément :

- Parmi les phrases acceptées, quarante (40) étaient grammaticalement bien construites, tandis que dix (10) étaient mal construites.
- Parmi les phrases rejetées, dix (10) étaient grammaticalement correctes, tandis que quarante (40) étaient mal construites.

Cette répartition est synthétisée dans le tableau suivant :

| | Bien construites | Mal construites | Total |
|------------------|-------------------------|------------------------|--------------|
| Acceptées | 40 | 10 | 50 |
| Rejetées | 10 | 40 | 50 |
| Total | 50 | 50 | 100 |

Tableau 14 : Tableau de synthèse des résultats

Ces résultats montrent que, sur les 100 phrases, 80 ont été correctement évaluées par l'analyseur, ce qui correspond à un taux de réussite global de **80 %**. Autrement dit, toutes les phrases bien construites acceptées (40/40) et toutes les phrases mal construites rejetées (40/40) ont été correctement identifiées. Les erreurs proviennent essentiellement :

- Du rejet de dix phrases grammaticalement correctes, non reconnues en raison de limites lexicales ou de structures non couvertes par le système,
- Et de l'acceptation de dix phrases incorrectes, dues à des combinaisons syntaxiques formellement valides mais sémantiquement fautives.

Ce taux de 80 % constitue une base encourageante, car il reflète la capacité du système à reconnaître correctement une large majorité des constructions standards du wolof. Il met cependant en lumière certaines limites liées à la couverture lexicale, au typage incomplet ou au manque de filtrage sémantique. Ces points ouvrent des perspectives claires d'amélioration : affiner le typage des mots, enrichir le lexique, ou encore intégrer une dimension sémantique. Dans cette optique, il est raisonnable d'envisager qu'une extension de ces ressources permette d'atteindre un taux de réussite supérieur (85 % ou plus) dans des travaux futurs.

4. Discussion des résultats

L'analyse montre que le système est performant pour des phrases simples et bien typées. Il reconnaît correctement les structures SVO avec déterminants postposés et les prépositions. Toutefois, il présente des limites importantes :

- D'une part, l'absence de certains mots empêche l'analyse de phrases correctes.
- D'autre part, le système accepte des phrases syntaxiquement valides mais sémantiquement incorrectes.
- Enfin, il n'intègre pas les opérateurs de focus, de subordination ou de négation.

Ces résultats démontrent la solidité du formalisme GCC pour modéliser la syntaxe de base du wolof, tout en soulignant la nécessité d'un enrichissement lexical et d'une modélisation plus fine des phénomènes complexes (phrases relatives, coordination, etc.).

5. Synthèse de l'expérimentation

L'évaluation expérimentale conduite sur un corpus de cent phrases en wolof a permis de mesurer, de manière qualitative, les performances du système d'analyse syntaxique fondé sur la Grammaire Catégorielle Combinatoire Applicative (GCC). Malgré le nombre limité de phrases dans le jeu de données, les résultats montrent que le parseur présente une capacité satisfaisante à dériver correctement des phrases simples et bien typées, notamment celles reposant sur l'ordre canonique SVO et sur les structures élémentaires du wolof contemporain.

Toutefois, plusieurs limites importantes ont été identifiées : l'incapacité à analyser certaines constructions grammaticalement correctes, en raison de lacunes lexicales ou d'un typage insuffisamment raffiné ; ainsi que la dérivation erronée de phrases mal construites, liées à l'absence de contraintes sémantiques dans le modèle. Ces observations confirment à la fois la validité de l'approche formelle retenue dans un cadre contrôlé, et la nécessité d'un enrichissement progressif du système pour couvrir la diversité morphosyntaxique de la langue.

Ce bilan permet d'établir un diagnostic clair sur l'état actuel du prototype, tout en posant les bases des améliorations futures envisagées dans la conclusion générale du mémoire.

Chapitre 8 : Conclusion et perspectives

Ce mémoire a exploré la modélisation syntaxique du wolof à travers le prisme de la Grammaire Catégorielle Combinatoire (GCC), un formalisme logique et lexicalisé qui s'est révélé particulièrement adapté à la description de langues naturelles. Partant du constat du faible outillage linguistique et computationnel pour le wolof, nous avons entrepris une démarche de formalisation linguistique simple mais pratique, associée à une implémentation expérimentale basée sur la bibliothèque NLTK.CCG en Python.

La première partie du travail a été consacrée à l'étude théorique de la GCC, de ses fondements logiques à ses opérations combinatoires, en passant par les principes de typage et les combinateurs fonctionnels. Cette base formelle a permis d'aborder avec précision la structuration grammaticale du wolof, langue à postpositions, à ordre SVO et aux déterminants postposés, en identifiant les régularités syntaxiques compatibles avec les catégories de la GCC.

Dans la seconde partie, un lexique typé a été élaboré à partir d'exemples représentatifs du wolof moderne, puis enregistré sous forme de fichier JSON. Ce lexique a servi de socle à l'analyseur syntaxique, intégré dans une interface interactive, permettant de dériver ou rejeter des phrases selon leur structure formelle. L'évaluation expérimentale, conduite sur un corpus de cent phrases réparties en quatre catégories (bien construites acceptées, bien construites rejetées, mal construites acceptées, mal construites rejetées), a permis d'identifier les points forts du système, notamment sa capacité à traiter les structures simples SVO, mais aussi ses limites, en particulier dans la gestion des constructions complexes (interrogatives, négatives, focalisées) et dans le rejet de certains faux positifs syntaxiques.

Les résultats obtenus soulignent la pertinence du formalisme GCC pour la modélisation syntaxique du wolof, tout en mettant en lumière les défis propres aux langues peu dotées : lacunes lexicales, traitement des formes conjuguées, absence de contraintes sémantiques.

Plusieurs perspectives d'amélioration peuvent être envisagées pour prolonger ce travail. On peut citer : l'intégration d'un affichage graphique des arbres syntaxiques (via `nlk.tree.Tree` ou `matplotlib`), la prise en compte des multi-dérivations lorsque plusieurs analyses sont possibles, l'enrichissement du lexique avec des verbes complexes et des expressions idiomatiques, l'ajout de fonctions d'export des résultats dérivationnels, ou encore le développement d'une version web interactive avec Streamlit ou Gradio, accessible sans installation préalable.

En définitive, ce mémoire constitue une première contribution à la formalisation grammaticale du wolof en TALN, en combinant approche théorique, modélisation linguistique et implémentation logicielle. Il illustre la fécondité des approches logiques pour le traitement des langues africaines, et appelle à de futurs travaux alliant langues, informatique et sciences du langage pour renforcer la présence du wolof dans les technologies du langage.

REFERENCES

- [1] Adelani, David Ifeoluwa ; Abbott, Jade ; Neubig, Graham ; D'souza, Daniel ; Kreutzer, Julia ; Lignos, Constantine ; Palen-Michel, Chester. MasakhaNER: Named Entity Recognition for African Languages. Transactions of the Association for Computational Linguistics, vol. 9, 2021, p. 1116-1131. <https://aclanthology.org/2021.tacl-1.66.pdf>
- [2] Adelani, David Ifeoluwa ; Neubig, Graham ; Ruder, Sebastian ; Rijhwani, Shruti ; Beukman, Michael ; Palen-Michel, Chester ; Lignos, Constantine. MasakhaNER 2.0: Africa-centric Transfer Learning for Named Entity Recognition. Proceedings of EMNLP, 2022. <https://aclanthology.org/2022.emnlp-main.298/>
- [3] Ajdukiewicz, Kazimierz. Die syntaktische Kategorie der Phrase. Studia Philosophica, vol. 1, 1935, p. 1-27.
- [4] Baldridge, Jason. Lexically Specified Derivational Control in Combinatory Categorical Grammar. Ph.D. Dissertation, University of Edinburgh, 2002. 312 p. https://www.asc.ohio-state.edu/pollard.4/syntax2/readings/Baldridge_dissertation.pdf
- [5] Bar-Hillel, Yehoshua. A Quasi-Arithmetical Notation for Syntactic Description. Language, vol. 29, no 1, 1953, p. 47-58.
- [6] Bird, Steven ; Klein, Ewan ; Loper, Edward. Natural Language Processing with Python. Beijing, O'Reilly Media, 2009. 504 p.
- [7] Biskri, Ismail. « Grammaires catégorielles combinatoires et applications à la représentation syntaxique des langues naturelles ». Recherches Linguistiques, vol. 20, 1999, p. 61-84.
- [8] Biskri, Ismail. « De la logique à la syntaxe : vers une formalisation catégorielle des langues ». Revue T.A.L., vol. 43, no 2, 2002, p. 115-138.
- [9] Boilat, David. Grammaire de la langue woloffe. Paris, Librairie Poussielgue-Rusand, 1858. 224 p.

- [10] Cissé, Thierno Ibrahima ; Sadat, Fatiha. Vérification et correction orthographique automatique pour les langues parlées et très peu dotées : étude de cas sur le wolof. Mémoire de maîtrise, Université du Québec à Montréal, 2023. <https://archipel.uqam.ca/17331/>
- [11] Dard, Jean. Dictionnaire français-wolof. Paris, Imprimerie Royale, 1825. <https://catalog.hathitrust.org/Record/008675500>
- [12] Dard, Jean. Grammaire wolofe. Paris, Imprimerie Royale, 1826. https://books.google.ca/books/about/Grammaire_Wolof.html?hl=fr&id=2iB-KAAAAcAAJ&redir_esc=y
- [13] Desclés, Jean-Pierre ; Biskri, Ismail. Logique combinatoire et linguistique : grammaire catégorielle combinatoire applicative. Paris, Hermès Science, 1995. 248 p.
- [14] Diagne, Pathé. Glossaire scientifique français-wolof. Dakar, Nouvelles Éditions africaines, 1980. http://wolofresources.org/language/download/diagne_grammaire_wolof.pdf
- [15] Dione, Cheikh M. Bamba. Relative clauses in Wolof: An LFG account. Stanford University, 2021. <https://web.stanford.edu/group/cslipublications/cslipublications/LFG/LFG-2021/lfg2021-dione.pdf>
- [16] Diop, Birago. Les Contes d'Amadou Koumba. Paris, Présence Africaine, 1947.
- [17] Diop, Cheikh Anta. Nations nègres et culture. Paris, Présence Africaine, 1979. 440 p.
- [18] Nguer, El Hadji Malick. Contribution à l'étude du système verbal du wolof. Dakar, Université Cheikh Anta Diop, 2015.
- [19] Dramé, Mamour. Dawal ak bind làmmiñu wolof. Vol. 1 : Orthographe, graphèmes et règles. Dakar, IFAN/UCAD, 2023.
- [20] Dramé, Mamour. Dawal ak bind làmmiñu wolof. Vol. 2 : Exercices et corrigés. Dakar, IFAN/UCAD, 2023.
- [21] Enguehard, Chantal ; Mangeot, Mathieu. « DiLAF : informatisation de dictionnaires langues africaines-français ». Actes de conférence, 2014.

- [22] Faidherbe, Louis. Essai sur la langue peul et sur un certain nombre d'autres idiomes de l'Afrique occidentale. Paris, Imprimerie Impériale, 1864.
- [23] Kuhlmann, Marco ; Satta, Giorgio. Parsing Techniques for Combinatory Categorical Grammar. *Computational Linguistics*, vol. 40, no 1, mars 2014, p. 1-63.
- [24] Lambek, Joachim. The Mathematics of Sentence Structure. *American Mathematical Monthly*, vol. 65, no 3, 1958, p. 154-170.
- [25] Montague, Richard. Universal Grammar. *Theoria*, vol. 36, no 3, 1970, p. 373-398.
- [26] Müller, Stefan. Grammatical theory: From transformational grammar to constraint-based approaches. Berlin, Language Science Press, 2016. <https://langsci-press.org/catalog/book/25>
- [27] Nouvel, Jean. « Typologie des langues d'Afrique de l'Ouest ». In Durand, Jacques (dir.), *Introduction à la linguistique africaine*. Paris, CNRS Éditions, 2011, p. 95-120.
- [28] Roger, Jacques-François. *Recherches philosophiques sur la langue ouolofe*. Paris, Dondey-Dupré, 1829.
- [29] Steedman, Mark. *Surface Structure and Interpretation*. Cambridge (Massachusetts), MIT Press, 1996. 288 p.
- [30] Steedman, Mark. Combinatory Grammars and Parasitic Gaps. *Natural Language & Linguistic Theory*, vol. 5, no 3, 1987, p. 403-439.
- [31] Dione, Cheikh M. Bamba ; Kuhn, Jonas ; Zarriß, Sina. *Design and Development of Part-of-Speech-Tagging Resources for Wolof (Niger-Congo)*. Proceedings of LREC, 2010.
- [32] Dione, Cheikh M. Bamba. *Developing Universal Dependencies for Wolof*. Proceedings of the Fourth Workshop on Universal Dependencies (UDW), 2019. <https://aclanthology.org/W19-8003.pdf>
- [33] Nguer, Elhadji Mamadou ; Lo, Alla ; Dione, Cheikh M. Bamba ; Ba, Sileye O. ; Lo, Moussa. *SENCORPUS: A French–Wolof Parallel Corpus*. Proceedings of LREC, 2020. <https://aclanthology.org/2020.lrec-1.341.pdf>

[34] Dione, Cheikh M. Bamba; Adelani, David I. ; et al. *MasakhaPOS: Part-of-Speech Tagging for Typologically Diverse African Languages*. Proceedings of ACL, 2023.

<https://aclanthology.org/2023.acl-long.609.pdf>

[35] Clark, Stephen; Curran, James R. *Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models*. Computational Linguistics, vol. 33, no 4, 2007, p. 493–552.

<https://www.cs.ox.ac.uk/people/stephen.clark/papers/cl07parser.pdf>

[36] Lewis, Mike; Steedman, Mark. *A CCG Parsing with a Supertag-Factored Model*.* Proceedings of EMNLP, 2014.

<https://aclanthology.org/D14-1107.pdf>

DICTIONNAIRE DES TYPES UTILISÉS

```
{
"categories_primitives": ["S", "NP", "N", "PP"],
"lexique": [
{ "mot": "damay", "categorie": "NP" },
{ "mot": "daño", "categorie": "NP" },
{ "mot": "dama", "categorie": "NP" },
{ "mot": "dangay", "categorie": "NP" },
{ "mot": "dafay", "categorie": "NP" },
{ "mot": "daño", "categorie": "NP" },
{ "mot": "danga", "categorie": "NP" },
{ "mot": "dafa", "categorie": "NP" },
{ "mot": "lekk", "categorie": "(S\\NP)/NP" },
{ "mot": "tek", "categorie": "(S\\NP)/NP" },
{ "mot": "wañi", "categorie": "(S\\NP)/NP" },
{ "mot": "setani", "categorie": "(S\\NP)/NP" },
{ "mot": "jaang", "categorie": "S\\NP" },
{ "mot": "tok", "categorie": "(S\\NP)/PP" },
{ "mot": "ceeb", "categorie": "N" },
{ "mot": "xaliss", "categorie": "N" },
{ "mot": "lamb", "categorie": "N" },
{ "mot": "bañ", "categorie": "N" },
{ "mot": "bi", "categorie": "NP\\N" },
{ "mot": "ji", "categorie": "NP\\N" },
{ "mot": "ci", "categorie": "PP/NP" },
{ "mot": "ndox", "categorie": "N" },
{ "mot": "mi", "categorie": "NP\\N" },
{ "mot": "bëgg", "categorie": "(S\\NP)/NP" },
{ "mot": "naan", "categorie": "(S\\NP)/NP" },
{ "mot": "mew", "categorie": "N" },
{ "mot": "dem", "categorie": "S\\NP" },
{ "mot": "liggeey", "categorie": "S\\NP" },
{ "mot": "Ndax", "categorie": "S/(S\\NP)" },
{ "mot": "dëgg", "categorie": "S\\NP" },
{ "mot": "waxtu", "categorie": "N" },
{ "mot": "jot", "categorie": "(S\\NP)/NP" },
{ "mot": "kër", "categorie": "N" },
{ "mot": "gi", "categorie": "NP\\N" },

{ "mot": "mar", "categorie": "S\\NP" },
{ "mot": "bëgg", "categorie": "S\\NP" },
{ "mot": "ma", "categorie": "NP" },
{ "mot": "ngi", "categorie": "NP\\NP" },
{ "mot": "nga", "categorie": "NP" },
{ "mot": "woloƴ", "categorie": "N" },
```

```

{ "mot": "tere", "categorie": "N" },
{ "mot": "fi", "categorie": "PP" },
{ "mot": "rekk", "categorie": "S\\S" },
{ "mot": "ban", "categorie": "NP/N" },
{ "mot": "moo", "categorie": "(S\\NP)/S" },
{ "mot": "naata", "categorie": "NP/N" },
{ "mot": "lay", "categorie": "(S\\NP)/NP" },
{ "mot": "baal", "categorie": "(S\\NP)/NP" },
{ "mot": "fan", "categorie": "PP" },
{ "mot": "am", "categorie": "(S\\NP)/NP" },
{ "mot": "wanak", "categorie": "S\\S" },
{ "mot": "xiif", "categorie": "S\\NP" },
{ "mot": "waaw", "categorie": "S" },
{ "mot": "dëgguma", "categorie": "S" },
{ "mot": "ndax", "categorie": "S/(S\\NP)" },
{ "mot": "jar", "categorie": "(S\\NP)/NP" },
{ "mot": "seet", "categorie": "(S\\NP)/NP" },
{ "mot": "benn", "categorie": "NP/N" },
{ "mot": "jàngalekat", "categorie": "N" },
{ "mot": "cebbi", "categorie": "N" },
{ "mot": "kass", "categorie": "N" },
{ "mot": "guiss", "categorie": "(S\\NP)/NP" },
{ "mot": "dox", "categorie": "S\\NP" },
{ "mot": "fii", "categorie": "PP" },
{ "mot": "xale", "categorie": "N" },
{ "mot": "sabar", "categorie": "N" },
{ "mot": "ndam", "categorie": "N" },
{ "mot": "jaangi", "categorie": "S\\NP" },
{ "mot": "cebb", "categorie": "N" }

```

],

```

"phrases": [
"damay lekk bi ceeb ",
"dama wañi xaliss bi",
"dangay jaang",
"danga xiif",
"dafay tok ci bañ bi",
"dafa lekk ceeb bi",
"dama wañi ndox mi",
"dama mar",
"daño setani lamb ji",
"dañooy tek tere bi",

```

"dama bëgg naan ndox",
"ma ngi dem liggeey",
"ndax dëgg nga wolof",
"ma ngi fi rekk",
"ban waxtu moo jot",
"naata lay jar",

"damay tek kass bi",
"dafay seet kër gi",
"dangay naan mew mi?",
"dafay liggeey ci kër gi",
"daño jaang wolof",
"dama bëgg xale bi",
"danga tok ci kër gi",
"dafay naan ndox mi",
"dafa jaay xaliss bi",
"dama guiss benn tere fii",
"ceeb bi danga lekk",
"kër gi dafay tok",
"danga ci ban bi tok",
"gi lamb danga tek",
"dama wañi gi xaliss",
"ndox gi naan danga",
"wolof bi danga jaang",
"ban bi ci tok danga",
"dafa seet gi jàngalekat",
"dangay jar naata",
"dafa am",
"dama baal",
"Ndax danga?",
"dama jaang wolof bi",
"dafa mar ndax dox bi",
"dangay liggeey ci waxtu wi?",
"damay jaang wolof",
"danga xiif rekk",
"daño dem ci kër gi",
"dama bëgg tok ci kër gi",

"damay bëgg liggeey ci kër gi",
"danga jaang ci benn waxtu",
"ceeb gi lamb lekk danga",
"damay bi ceeb lekk",
"daño lamb tek bi",
"dangay bi gi kër",
"dafay ndox naan bi",
"danga wolof liggeey",

"dama tok bi lamb",
"ndox mi dem",
"waxtu jot bi dañó",
"gi kër dangay",
"naan ceeb gi danga",
"lamb dafay tek ji",
"liggeey wolof dangay",
"fi tok bi",
"damey gi jaang",
"wanak dafa",

"baal ma fan moo am wanak",
"waaw dëgguma",
"dangay bëgg jaang",
"dama seet benn jàngalekat bi",

"danga ceeb lekk bi",
"dafay lamb setani ji",
"ceeb bi setani dafa",
"dama tere tek bi",
"dafay wañi bañ gi",
"ceeb bi tek dafa",
"dafay bañ wañi gi",
"danga setani bañ gi",
"dafa wolof lekk bi",
"bañ bi wañi dama",

"lekk lekk lekk",
"tok tok",
"ceeb lekk bi",
"danga danga",
"xaliss tok ci",
"tok ci bi bañ",
"setani xiif lekk",
"wañi bi bi",
"danga cebbi xiif",
"dafa lamb bi",

"dangay dem ci xew bi",
"dafa setani sabar",
"dafa am xaliss",
"baal ma",
"Ndax dangay dem?",
"ma ngi jaangi",
"dafay tok ci bañ bi",
"danga tek cebb bi ",

"dama jar waxtu bi",
"dañoy liggeey rekk",
"dafa wañi xaliss bi",
"dama seet jàngalekat bi",
"dangay jot waxtu bi",
"dafa tek lamb ji",

"dina am ndam"

] }
}