

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES

PAR
MERY ALASSANE FALL

Détection des erreurs dans des textes écrits par des apprenants du français langue seconde

Septembre 2025

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

Ce mémoire explore l'application de la Grammaire Catégorielle Combinatoire et Applicative (GCCA) à l'analyse syntaxique du français, en particulier pour les apprenants en français langue seconde. L'objectif principal est de concevoir une solution informatique capable d'analyser syntaxiquement des phrases, de détecter les erreurs, de localiser leur position et de proposer des suggestions de correction.

Pour ce faire, nous avons utilisé un corpus de 100 phrases que nous avons nous-mêmes créées combiné aux outils des bibliothèques NLTK et LanguageTool, en adaptant les règles de la GCCA aux particularités morphosyntaxiques du français, notamment les pronoms, déterminants, modaux, adverbes et verbes. Des exemples concrets illustrent la détection de fautes telles que les erreurs de conjugaison, d'accord ou de combinaison d'articles.

AVANT-PROPOS

Ce mémoire est le fruit d'un intérêt croissant pour le traitement automatique du langage naturel et, plus particulièrement, pour les défis que pose l'analyse syntaxique du français langue seconde (Cuq, 2013). Dans un contexte où les outils linguistiques jouent un rôle de plus en plus central dans l'apprentissage et l'assistance à la rédaction, j'ai souhaité explorer une approche rigoureuse et formelle à travers la grammaire catégorielle combinatoire et applicative (GCCA).

Ce travail a été réalisé dans le cadre de ma maîtrise en mathématiques et informatique appliquées, sous la direction de Monsieur Ismaïl Biskri, que je remercie pour ses conseils, sa disponibilité, son écoute attentive, son œil critique et son accompagnement tout au long de cette recherche.

Je tiens également à exprimer ma gratitude envers tous les professeurs du département de mathématique et informatique de l'UQTR qui ont participé de près ou de loin, à la concrétisation de ce projet.

Enfin, je tiens à exprimer ma reconnaissance et ma gratitude à ma famille qui m'a toujours soutenu et encouragé pour la réussite de ce projet.

Table des matières

Résumé.....	ii
AVANT-PROPOS.....	iii
Table illustration des tableaux	vii
Table des illustrations	viii
Chapitre I : Introduction générale.....	1
Chapitre II : Systèmes applicatifs typés	4
I) Introduction.....	4
II) Principe des systèmes applicatifs typés	4
II.1) Les termes	4
II.2) Les types ou catégories	4
II.3) Les propositions	5
II.4) Les opérateurs.....	5
III) La grammaire catégorielle.....	5
III.1) Husserl (1913) : Catégories de significations.....	5
III.2) Lesniewski (1922) : Catégories sémantiques.....	6
III.3) Ajdukiewicz (1935)	6
III.4) Hillel (1953) : Grammaire catégorielles bidirectionnelle.....	8
III.5) Lambek (1958-1961) : Calcul de Lambek	10
III.6) Steedman (1989) : La grammaire catégorielle combinatoire.....	11
III.7) Shaumyann(1998) : La grammaire applicative universelle.....	13
III.8) Descles (1989) : Grammaire applicative et cognitive	14
IV) La logique combinatoire.....	15
IV.1) Cadre historique	15
IV.2) Combinateurs logiques.....	15
V) Conclusion.....	22
Chapitre III : Méthodologie.....	23
I) Introduction.....	23
II) Le Workflow	23
III) La grammaire catégorielle combinatoire et applicative : GCCA.....	25
III.1) Principe de la GCCA.....	25
III.2) Les règles de la GCCA	25
III.3) Les étapes de la GCCA	27
III.4) La coordination de la GCCA.....	29
III.5) Les métarègles.....	29
III.6) La réorganisation structurelle	31

IV) Quelques exemples de phrases avec leurs dérivations syntaxiques	34
IV.1) Dérivation de phrases bien construites	34
IV.2) Dérivation de phrases mal construites	41
V) Outil utilisé pour la suggestion de correction	49
V.1) Qu'est-ce-que LanguageTool ?	49
V.2) Fonctionnement de languageTool	50
VI) Poids des suggestions de corrections.....	51
VII) Application du modèle	52
VIII) Conclusion.....	53
Chapitre IV : Implémentation.....	54
I) Introduction.....	54
II) Présentation et objectifs de la solution proposée.....	54
III) Le langage de programmation utilisé : Python.....	54
IV) L'analyseur syntaxique choisi	55
IV.1) L'analyseur syntaxique C&C CCG Parser	55
IV.2) L'analyseur syntaxique OpenCCg	55
IV.3) L'analyseur syntaxique NLTK	56
IV.4) Pourquoi le choix de NLTK ?	57
V) Les bibliothèques utilisées	58
V.1) NLTKCCG	58
V.2) Csv (Comma Separated Values)	58
V.3) Pandas	58
V.4) Sklearn.metrics	58
VI) La représentation des données.....	58
VI.1) La notation des Catégories.....	59
VI.2) Les règles de combinaison :	62
VII) Fonctionnement de la plateforme	67
VII.1) Ecriture de la phrase à analyser	67
VII.2) Exécution du code d'analyse	67
VII.3) Analyse grammaticale avec GCCA.....	67
VII.4) Détection d'erreurs avec LanguageTool	68
VII.5) Traitement des suggestions.....	68
VIII) Conclusion.....	69
Chapitre V : Expérimentation	70
I) Introduction.....	70
II) Le corpus	70

III) Le lexique.....	70
IV) Captures des résultats des cinquante phrases correctes.....	71
IV.1) Introduction.....	71
IV.2) Conclusion	95
V) Captures des résultats des cinquante phrases incorrectes.....	95
V.1) Introduction.....	95
V.2) Conclusion	146
VI) Performance des suggestions de correction	146
VII) Performance de notre modèle	147
Interprétation	149
VIII) Conclusion.....	151
Chapitre VI : Conclusion générale	152
Bibliographie.....	153
Annexe I : Corpus composé de 100 PHRASES.....	155
Annexe II : Lexique (Dictionnaire).....	158
Annexe III : Fichier csv résultats_suggestions.....	159
Annexe IV : Fichier CSV pour l'évaluation de la performance du code.....	160
Annexe V : Explication des éléments de la matrice de confusion	163

Table illustration des tableaux

Tableau 1: Tableau comparatif de CCg Parser, Openccg et NLTKCCg	57
Tableau 2: Quelques catégories du lexique	158

Table des illustrations

Figure 1 : Propriété de Church-Rosser.....	22
Figure 2 : Workflow	24
Figure 3:Etapes de la GCCA	28
Figure 4: Illustration poids de suggestion de correction	52
Figure 5 : Interface entrée de phrases.....	67
Figure 6 : Bouton d'exécution du code.....	67
Figure 7 : Résultat dérivation CCG d'une phrase correcte.....	68
Figure 8 : Résultat dérivation CCG d'une phrase incorrecte.....	68
Figure 9 : Suggestions de corrections pour une phrase incorrecte	69
Figure 10 : Analyse CCG phrase 1	71
Figure 11 : Analyse CCG phrase 2	72
Figure 12 : Analyse CCG phrase 3	72
Figure 13 : Analyse CCG phrase 4	73
Figure 14 : Analyse CCG phrase 5	73
Figure 15 : Analyse CCG phrase 6	74
Figure 16 : Analyse CCG phrase 7	74
Figure 17 : Analyse CCG phrase 8	75
Figure 18 : Analyse CCG phrase 9	75
Figure 19 : Analyse CCG phrase 10	76
Figure 20 : Analyse CCG phrase 11	76
Figure 21 : Analyse CCG phrase 12	77
Figure 22 : Analyse CCG phrase 13	77
Figure 23 : Analyse CCG phrase 14	78
Figure 24 : Analyse CCG phrase 15	78
Figure 25 : Analyse CCG phrase 16	79
Figure 26 : Analyse CCG phrase 17	79
Figure 27 : Analyse CCG phrase 18	80
Figure 28 : Analyse CCG phrase 19	80
Figure 29 : Analyse CCG phrase 20	80
Figure 30 : Analyse CCG phrase 21	81
Figure 31 : Analyse CCG phrase 22	81
Figure 32 : Analyse CCG phrase 23	82
Figure 33 : Analyse CCG phrase 24	82
Figure 34 : Analyse CCG phrase 25	83
Figure 35 : Analyse CCG phrase 26	83
Figure 36 : Analyse CCG phrase 27	84
Figure 37 : Analyse CCG phrase 28	84
Figure 38 : Analyse CCG phrase 29	85
Figure 39 : Analyse CCG phrase 30	85
Figure 40 : Analyse CCG phrase 31	86
Figure 41 : Analyse CCG phrase 32	86
Figure 42 : Analyse CCG phrase 33	87
Figure 43 : Analyse CCG phrase 34	87
Figure 44 : Analyse CCG phrase 35	88
Figure 45 : Analyse CCG phrase 36	88
Figure 46 : Analyse CCG phrase 37	89

Figure 47 : Analyse CCG phrase 38	89
Figure 48 : Analyse CCG phrase 39	90
Figure 49 : Analyse CCG phrase 40	90
Figure 50 : Analyse CCG phrase 41	91
Figure 51 : Analyse CCG phrase 42	91
Figure 52 : Analyse CCG phrase 43	92
Figure 53 : Analyse CCG phrase 44	92
Figure 54 : Analyse CCG phrase 45	93
Figure 55 : Analyse CCG phrase 46	93
Figure 56 : Analyse CCG phrase 47	93
Figure 57 : Analyse CCG phrase 48	94
Figure 58 : Analyse CCG phrase 49	94
Figure 59 : Analyse CCG phrase 50	95
Figure 60 : Analyse CCG Phrase 51	96
Figure 61 : Validation avec suggestion phrase 51	97
Figure 62 : Analyse CCG Phrase 52	97
Figure 63 : Validation avec suggestion phrase 52	98
Figure 64 : Analyse CCG Phrase 53	98
Figure 65 : Validation avec suggestion phrase 53	98
Figure 66 : Analyse CCG Phrase 54	99
Figure 67 : Validation avec suggestion phrase 54	99
Figure 68 : Analyse CCG Phrase 55	100
Figure 69 : Validation avec suggestion phrase 55	100
Figure 70 : Analyse CCG Phrase 56	101
Figure 71 : Validation avec suggestion phrase 56	101
Figure 72 : Analyse CCG Phrase 57	102
Figure 73 : Validation avec suggestion phrase 57	102
Figure 74 : Analyse CCG Phrase 58	103
Figure 75 : Validation suggestion phrase 58	103
Figure 76 : Analyse CCG Phrase 59	104
Figure 77 : Validation suggestion phrase 59	104
Figure 78 : Analyse CCG Phrase 60	105
Figure 79 : Validation suggestion phrase 60	105
Figure 80 : Analyse CCG Phrase 61	106
Figure 81 : Validation suggestion phrase 61	106
Figure 82 : Analyse CCG Phrase 62	107
Figure 83 : Validation suggestion phrase 62	107
Figure 84 : Analyse CCG Phrase 63	108
Figure 85 : Validation suggestion phrase 63	108
Figure 86 : Analyse CCG Phrase 64	109
Figure 87 : Validation suggestion phrase 64	109
Figure 88 : Analyse CCG Phrase 65	110
Figure 89 : Validation suggestion phrase 65	110
Figure 90 : Analyse CCG Phrase 66	111
Figure 91 : Validation suggestion phrase 66	111
Figure 92 : Analyse CCG Phrase 67	112
Figure 93 : Validation suggestion phrase 67	112
Figure 94 : Analyse CCG Phrase 68	113

Figure 95 : Validation suggestion phrase 68.....	113
Figure 96 : Analyse CCG Phrase 69	114
Figure 97 : Validation suggestion phrase 69.....	114
Figure 98 : Analyse CCG Phrase 70	115
Figure 99 : Validation suggestion phrase 70.....	115
Figure 100 : Analyse CCG Phrase 71	116
Figure 101 : Validation suggestion phrase 71.....	116
Figure 102 : Analyse CCG Phrase 72	117
Figure 103 : Validation suggestion phrase 72.....	117
Figure 104 : Analyse CCG Phrase 73	118
Figure 105 : Validation suggestion phrase 73.....	118
Figure 106 : Analyse CCG Phrase 74	119
Figure 107 : Validation suggestion phrase 74.....	119
Figure 108 : Analyse CCG Phrase 75	120
Figure 109 : Validation suggestion phrase 75.....	120
Figure 110 : Analyse CCG Phrase 76	121
Figure 111 : Validation suggestion phrase 76.....	121
Figure 112 : Analyse CCG Phrase 77	122
Figure 113 : Validation suggestion phrase 77.....	122
Figure 114 : Analyse CCG Phrase 78	123
Figure 115 : Validation suggestion phrase 78.....	123
Figure 116 : Analyse CCG Phrase 79	124
Figure 117 : Validation suggestion phrase 79.....	124
Figure 118 : Analyse CCG Phrase 80	125
Figure 119: Validation suggestion phrase 80.....	126
Figure 120 : Analyse CCG Phrase 81	126
Figure 121 : Validation suggestion phrase 81.....	127
Figure 122 : Analyse CCG Phrase 82	127
Figure 123 : Validation suggestion phrase 82.....	128
Figure 124 : Analyse CCG Phrase 83	128
Figure 125 : Validation suggestion phrase 83.....	129
Figure 126 : Analyse CCG Phrase 84	129
Figure 127 : Validation suggestion phrase 84.....	130
Figure 128 : Analyse CCG Phrase 85.....	130
Figure 129 : Validation suggestion phrase 85.....	131
Figure 130 : Analyse CCG Phrase 86	131
Figure 131 : Validation suggestion phrase 86.....	132
Figure 132 : Analyse CCG Phrase 87	132
Figure 133 : Validation suggestion phrase 87.....	133
Figure 134 : Analyse CCG Phrase 88	133
Figure 135 : Validation suggestion phrase 88.....	134
Figure 136 : Analyse CCG Phrase 89	134
Figure 137 : Validation suggestion phrase 89.....	135
Figure 138 : Analyse CCG Phrase 90	135
Figure 139 : Validation suggestion phrase 90.....	136
Figure 140 : Analyse CCG Phrase 91	136
Figure 141 : Validation suggestion phrase 91.....	137
Figure 142 : Analyse CCG Phrase 92	137

Figure 143 : Validation suggestion phrase 92.....	138
Figure 144 : Analyse CCG Phrase 93	138
Figure 145 : Validation suggestion phrase 93.....	139
Figure 146 : Analyse CCG Phrase 94	139
Figure 147 : Validation suggestion phrase 94.....	140
Figure 148 : Analyse CCG Phrase 95	140
Figure 149 : Validation suggestion phrase 95.....	141
Figure 150 : Analyse CCG Phrase 96	141
Figure 151 : Validation suggestion phrase 96.....	142
Figure 152 : Analyse CCG Phrase 97	142
Figure 153 : Validation suggestion phrase 97.....	143
Figure 154 : Analyse CCG Phrase 98.....	143
Figure 155 : Validation suggestion phrase 98.....	144
Figure 156 : Analyse CCG Phrase 99	144
Figure 157 : Validation suggestion phrase 99.....	145
Figure 158 : Analyse CCG Phrase 100	145
Figure 159 : Validation suggestion phrase 100.....	146
Figure 160:Fichier récapitulatif des performances de LanguageTool en matière de suggestion	147
Figure 161 : Phrases avec position de la première suggestion différente de celle proposée par LanguageTool.....	147
Figure 162: Rapport métrique du jeu de test.....	149
Figure 163: Résultat rapports_erreurs.csv	150
Figure 164 : Corpus composé de 100 phrases.....	157
Figure 165 : Fichier csv résultats_suggestions.....	159
Figure 166: Fichier Csv pour l'évaluation de la performance	162
Figure 167 : Explication éléments de la matrice de confusion	163

Chapitre I : Introduction générale

Le traitement automatique des langues naturelles (TALN) est un domaine de recherche qui vise à permettre aux ordinateurs de comprendre, interpréter et générer le langage humain. Avec l'essor de l'intelligence artificielle et des technologies linguistiques, le TALN est devenu essentiel dans de nombreuses applications telles que la traduction automatique, la reconnaissance de la parole, les assistants virtuels et le traitement de textes ((Muller, Royauté, & Silberztein, 2004); (Cori, 2020) ; (Chaumartin, 2020)).

En effet, dans un monde où les interactions homme-machine deviennent de plus en plus courantes, la nécessité d'avoir des outils capables d'interpréter, d'analyser et de générer des phrases dans diverses langues est cruciale. Le français, avec sa riche morphosyntaxe et ses nombreuses exceptions pose un défi particulier aux modèles d'analyse syntaxique. Les méthodes traditionnelles, souvent basées sur des grammaires contextuelles (EITCI, 2023) montrent leurs limites face à cette complexité.

C'est dans ce contexte que la Grammaire Catégorielle Combinatoire et Applicative (GCCA) s'impose comme une avancée significative dans le domaine de la linguistique computationnelle et du traitement automatique des langues naturelles (TALN). Développée par Ismaïl Biskri (1995) et Jean-Pierre Descles (1996), la GCCA se distingue par sa capacité à décrire de manière rigoureuse les structures syntaxiques des langues naturelles grâce à l'utilisation de catégories lexicales et de règles de combinaison strictes. Cette approche permet non seulement une analyse syntaxique précise, mais aussi une flexibilité dans le traitement de phrases complexes, ce qui la rend particulièrement utile pour les applications en TALN.

Notre mémoire se propose d'explorer en profondeur l'application de la grammaire catégorielle combinatoire à l'analyse syntaxique du français comme langue seconde (Cuq, 2013). Plus précisément, l'objectif principal de notre recherche est de proposer une solution informatique qui permettra de faire l'analyse syntaxique de phrases avec la GCCA, de détecter les erreurs dans les phrases ou dans les textes mais aussi d'indiquer leur position et de proposer des suggestions de correction. Pour ce faire, nous nous concentrerons sur l'adaptation de cette grammaire aux spécificités linguistiques du français, en tenant compte des particularités des pronoms, des déterminants, des modaux, des adverbes et des verbes (qui semblent être les obstacles pour un apprenant du français comme langue seconde (Cuq, 2013)). Prenons quelques exemples pour mieux illustrer l'objectif de notre travail.

Exemple 1 : Mery vais à l'école

Nous remarquons que le verbe « aller » est mal conjugué, car le sujet est à la troisième personne tandis que le verbe est conjugué à la première personne. Notre solution informatique vise à aider un utilisateur peu à l'aise avec la langue française à détecter que le verbe « aller », situé à la position 5, est mal conjugué, et lui proposera des suggestions de correction telles que « va ».

Exemple 2 : Il ne sera jamais malheureu

Ici, nous constatons que « malheureu » est mal orthographié. Notre solution permettra à un utilisateur de repérer que l'adjectif « malheureu », situé à la position 18, est incorrect, et lui proposera des suggestions comme « malheureux ».

Exemple 3 : Ils va travaillé

Nous constatons que le verbe « aller » est mal conjugué, car le sujet est à la troisième personne du pluriel, tandis que le verbe est conjugué à la troisième personne du singulier. Notre solution informatique vise à aider un apprenant en français langue seconde à détecter que le verbe aller, situé à la position 4, est mal conjugué, et à lui proposer des suggestions de correction telles que vont. Cependant, pour ce type d'erreur, notre solution peut suggérer « il va » ou « ils vont », laissant à l'apprenant le choix de la correction appropriée.

De plus, nous relevons que le verbe travailler est employé sous une forme verbale incorrecte : lorsque deux verbes se suivent, le second se met à l'infinitif. Par conséquent, notre solution proposera également de remplacer travaillé par « travailler ».

Exemple 4 : Je veux manger de les soupes

Dans cet exemple, nous identifions une erreur liée à l'utilisation de « de les », une combinaison d'articles qui n'existe pas en français. Notre solution analysera cette faute et suggérera, très probablement, la correction « des » (puisque soupes est au pluriel), qui est grammaticalement correcte.

Ainsi, notre approche ne se limitera pas à la correction des erreurs grammaticales, mais aidera également les utilisateurs à améliorer leurs compétences linguistiques en leur fournissant des retours clairs et précis sur les erreurs relevées et les corrections proposées.

Pour mener à bien ce travail, nous utiliserons un corpus composé de cent phrases (cinquante phrases correctes et cinquante phrases incorrectes que nous aurons sélectionnées nous-même) en français et appliquerons les outils de la bibliothèque Natural Language Toolkit (NLTK) combinés avec les techniques de la GCCA. Notre méthodologie inclura l'analyse de phrases simples et complexes, l'implémentation de règles de combinaison

spécifiques à la GCCA et l'évaluation des performances de notre modèle syntaxique. Nous porterons une attention particulière à la catégorisation des éléments lexicaux et à l'application des règles de combinaison tout en nous assurant que notre approche respecte les principes théoriques de la GCCA.

Notre objectif à terme est de concevoir un outil permettant d'assister un apprenant du français langue seconde (Cuq, 2013) à mieux maîtriser la langue écrite.

Il convient toutefois de préciser que ce travail n'est pas une étude complète sur les langues secondes (Cuq, 2013), mais une exploration puis une implémentation d'une grammaire considérée comme flexible et apte à s'adapter aux spécificités linguistiques du français. Nous sommes conscientes qu'il reste encore beaucoup de travail à accomplir pour analyser et modéliser l'ensemble des cas d'erreurs que peuvent commettre les apprenants de français langue seconde (Cuq, 2013).

Notre mémoire s'articule de la manière suivante :

- Le chapitre 2 introduit les systèmes applicatifs typés.
- Le chapitre 3 est consacré à la présentation de notre workflow, au modèle catégoriel adopté dans notre solution, ainsi qu'à l'illustration de quelques exemples de phrases accompagnés de leur dérivation syntaxique.
- Le chapitre 4 expose en détail notre implémentation, en mettant en avant les technologies employées, les bibliothèques utilisées et la méthode de représentation des données.
- Le chapitre 5 constitue la partie pratique, où nous présenterons notre corpus, notre lexique, nos résultats, et procéderons à l'évaluation des performances de notre solution.
- Enfin, le chapitre 6 conclut notre recherche en ouvrant sur des perspectives pour des travaux futurs.

Chapitre II : Systèmes applicatifs typés

I) Introduction

Les systèmes applicatifs postulent un modèle général dans lequel un opérateur s'applique à un opérande via des règles pour donner un résultat. Ils permettent de décrire la structure des phrases dans une langue donnée et ainsi de voir les relations entre celles-ci et de pouvoir comprendre comment elles interagissent grammaticalement.

Ce chapitre sera axé sur trois points importants à savoir : le principe des systèmes applicatifs typés, la grammaire catégorielle et enfin la logique combinatoire.

Ces derniers constituent les fondements nécessaires à la compréhension des systèmes applicatifs typés et permettrons d'en expliquer clairement le fonctionnement.

II) Principe des systèmes applicatifs typés

Les systèmes applicatifs typés sont composés de :

- Termes
- Types ou catégories
- Propositions
- Opérateurs

II.1) Les termes

Les termes jouent un rôle fondamental dans la représentation de la structure syntaxique des phrases et des expressions linguistiques. Dans un système applicatif typé, les termes représentent généralement les noms, verbes, adjectifs, adverbes, pronoms, etc. Ce sont les unités de base que la grammaire cherche à analyser. Par exemple, dans une phrase comme « Mery prépare le diner », les termes seraient les mots « Mery », « prépare », et « le diner ». Chaque terme est cependant associé à une catégorie grammaticale (type) qui spécifie son rôle syntaxique dans la phrase.

II.2) Les types ou catégories

Les types ou catégories permettent de comprendre le rôle de chaque terme dans une phrase en grammaire catégorielle. En effet, ils désignent la fonction grammaticale d'un mot ou d'une expression. Par exemple, un nom comme « Mery » peut avoir le type N (nom), tandis qu'un verbe transitif comme « prépare » peut avoir un type complexe $(S \setminus N) / N$ (une expression qui attend un complément d'objet à droite et un sujet à gauche pour former une phrase). Les types sont utilisés pour combiner les termes de manière logique et déterminer si une phrase est grammaticalement correcte.

II.3) Les propositions

Elles sont utilisées pour décrire la manière dont les termes s'agencent pour former des phrases grammaticales. Elles indiquent comment un terme donné se combine avec d'autres termes en fonction des types pour former une phrase correcte. Une proposition est comme une règle qui renseigne sur la manière dont les termes de la langue sont structurés selon leurs types. Par exemple, « Mery prépare le diner » serait une proposition qui déclare qu'il existe une manière valide de combiner ces termes pour former une phrase complète.

II.4) Les opérateurs

Les opérateurs indiquent comment les types peuvent se combiner pour former des structures grammaticales valides. En grammaire catégorielle, les deux opérateurs principaux sont la barre oblique droite (/) et la barre oblique gauche (\), qui représentent des types « fonctionnels ». Par exemple, dans le type $S \backslash N$, l'opérateur (\) indique que l'élément de type $S \backslash N$ est une fonction qui attend un terme de type N à sa gauche pour former une phrase complète de type S.

III) La grammaire catégorielle

La grammaire catégorielle a joué un rôle majeur dans le développement des modèles de traitement automatique du langage naturel et des systèmes de génération de langage. En effet, elle a permis de faire une représentation de la structure syntaxique des phrases en utilisant des catégories grammaticales et ceci en combinant des catégories grammaticales de base.

Cependant, pour mieux comprendre la grammaire catégorielle, il est indispensable de connaître son historique (Bourdeau, 2002) et (Second, 1990). Dans cette partie, nous ferons une brève historique des grammaires catégorielles, ce qui nous permettra de mieux expliciter leur rôle dans notre travail.

III.1) Husserl (1913) : Catégories de significations

Edmund Husserl était un mathématicien et un philosophe autrichien. Il est le fondateur de la phénoménologie (De Waelhens , 1954). En effet, cette dernière cherche à décrire et à analyser les structures essentielles de l'expérience consciente sans préjugés ou présuppositions métaphysiques.

Pour illustrer sa théorie, Husserl a pris l'exemple d'une fleur dans un environnement donné donc il s'est posé une seule question : Qu'est-ce qu'une fleur ?

Il se rendra compte de l'existence de plusieurs réponses à cette question seulement en variant son aspect dans sa pensée (Dortier, 2022).

En effet, la fleur peut être un platane, un sapin, un cerisier en fleurs de n'importe quelle couleur. Seulement, il sera difficile de définir la fleur sans les éléments qui entourent l'environnement dans lequel elle se trouve.

La phénoménologie de Husserl a inspiré plusieurs chercheurs qui ont essayé de développer ses idées dans plusieurs domaines notamment celui de la linguistique.

En effet, dans le domaine de la linguistique, cela peut se traduire par une exploration des significations et des structures sémantiques, ainsi que la manière dont les mots et les expressions acquièrent leur sens dans un contexte particulier.

III.2) Lesniewski (1922) : Catégories sémantiques

Stanislaw Lesniewski était un logicien polonais qui a contribué de manière significative au développement de la logique formelle. Il fut l'initiateur de la théorie qui a permis de régler la question des antonymes et qui deviendra les grammaires catégorielles.

En effet, sa théorie (catégories sémantiques (semantischen Kategorien)) se base sur deux types de catégories : d'une part, les catégories de base et, d'autre part, les catégories dérivées. Les seules catégories de base admises sont celles de la proposition (P) et celle du nom (N), les catégories dérivées seront introduites par la suite et formées à partir des catégories déjà présentes dans le système (Rapaille, 2011).

Les catégories de significations de Husserl sont généralement associées à l'analyse phénoménologique de l'expérience consciente, tandis que les catégories sémantiques de Lesniewski sont employées dans un contexte plus formel de la logique formelle et de l'ontologie.

III.3) Ajdukiewicz (1935)

Kazimierz Ajdukiewicz (1890-1963) était un philosophe et logicien polonais, un éminent représentant de l'école de Varsovie-Lemberg membre du Cercle de Vienne et un représentant important du mouvement logique polonais.

Il proposa une nouvelle approche formelle du langage naturel qui reposait sur un calcul conjoint de la grammaticalité et du sens des phrases comme le montre Adam Joly dans son mémoire intitulé « Du document textuel à sa carte sémantique fonctionnelle » (Joly, 2009).

L'une des réalisations les plus influentes de Kazimierz Ajdukiewicz est une idée de description syntaxique au moyen de types logiques, qui sous-tend la théorie de la grammaire appelée plus tard grammaire catégorielle.

Son approche est un prolongement de la théorie des catégories sémantiques de Stanislaw Lesniewski et de Husserl (Joray & Godart-Wendling, 2002).

Les catégories linguistiques se divisent en deux catégories à savoir : les catégories de base et les foncteurs.

Les catégories de base, elles, sont divisées en syntagmes nominaux et les catégories phrases, qui sont représentées respectivement par les notations N (nom) et S (phrase). Les catégories foncteurs sont récursivement obtenues à partir des catégories de base et d'un symbole d'application.

Ces dernières obéissent à ces deux règles :

- Les catégories de base du modèle sont des catégories de ce modèle.
- Si X et Y sont des catégories du modèle alors X/Y est une catégorie du modèle.
X/Y est une catégorie foncteur.

La barre oblique (/) représente l'application. Plus précisément :

- Le dénominateur (ici Y) symbolise le type de l'argument attendu.
- Le numérateur (ici X) représente le type du résultat de l'application de ce foncteur à un argument de type Y.

En résumé, X/Y est une fonction qui prend un élément de type Y et produit un élément de type X.

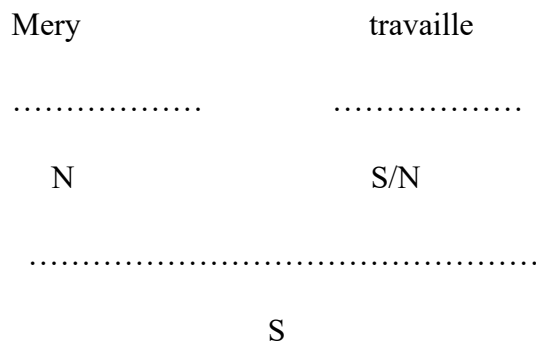
En plus de ces deux règles, Ajdukiewicz a mis en place un calcul formel. Ce dernier se fait d'une manière unique de gauche à droite dans le but d'arriver à un résultat égal à l'une des catégories de base. En effet, il se matérialise par les règles de réduction suivantes :

$$\frac{X}{Y} Y \rightarrow X$$

$$Y \frac{X}{Y} \rightarrow X$$

Partons maintenant du triplet (S, N, –) et appliquons-le sur un cas réel.

Prenons l'exemple suivant : Mery travaille



En appliquant les deux règles précédemment énoncées, nous obtenons (Mery travaille) avec le type S (S/N N).

Dans sa théorie, Ajdukiewicz dit que lorsque le processus de réduction aboutit à une expression qui appartient à l'une des deux catégories de base, cela indique que la structure syntaxique est cohérente et peut être associée à une signification sémantique, autrement dit, nous pouvons conclure que l'expression en question est bien formée.

III.4) Hillel (1953) : Grammaire catégorielles bidirectionnelle

Yehoshua Bar-Hillel était un philosophe, mathématicien et linguiste israélien. Il fut un pionnier dans les domaines de la traduction automatique et de la linguistique formelle. Il fut le développeur des grammaires AB (grammaire catégorielle bidirectionnelle). Ces dernières se basent sur une idée de Ajdukiewicz à savoir attribuer à chaque composant d'un langage donné une fraction indiquant comment ce composant doit être combiné (Moreau, 2004).

Cependant, dans les grammaires AB, toute grammaire est décrite par l'association d'une ou de plusieurs catégories à chaque mot du vocabulaire autrement dit le « lexique ». Les règles utilisées dans les dérivations sont donc universelles (Anoun, 2007). Ces dernières représentent les règles de réduction droite et gauche :

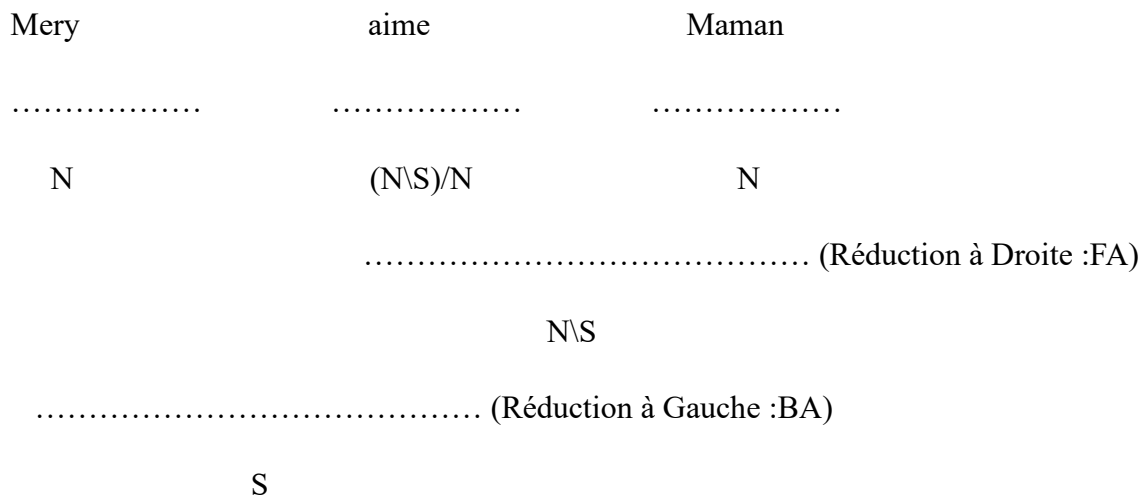
- $A/B, B \rightarrow A$ FA (Forward Application) : une catégorie de la forme A/B se combine avec un argument de type B situé à sa droite pour produire un résultat de type A
- $B, B \setminus A \rightarrow A$ BA (Backward Application) : une catégorie de la forme $B \setminus A$ se combine avec un argument de type B situé à sa gauche pour produire un résultat de type A.

Ces règles permettent de dériver des expressions syntaxiques complexes en suivant des combinaisons déterminées par les catégories assignées aux mots.

Tout comme l'approche de Adjukiewicz, dans les grammaires de AB, nous avons :

- Deux catégories de base S et N.
- Les types composés : désignent l'ensemble des types atomiques formés grâce aux opérateurs bidirectionnels ($/$, \backslash) qui offrent la possibilité de réaliser des réductions dans les deux sens (gauche et droite).
- Les types atomiques : Ces types représentent des expressions complètes.

Prenons un exemple pour illustrer tout ceci :



En appliquant les deux règles de réduction énoncées précédemment dans cet ordre : règle de droite suivie de la règle de gauche nous obtenons le résultat : (Mery aime Maman) de type S et ce qui signifie que la phrase est syntaxiquement bien construite.

En effet, aime est un verbe transitif et nécessite un sujet à sa gauche, ainsi qu'un complément d'objet à sa droite pour constituer une phrase. Ces derniers peuvent être aussi bien un nom propre comme Mery ou un groupe nominal composé, comme un chat (qui doit être de type N/N). Ainsi, dans la grammaire catégorielle combinatoire, le verbe aimer se voit attribuer la catégorie $(N \backslash S) / N$, indiquant qu'il attend un complément d'objet à droite et un sujet à gauche.

À l'inverse, les verbes intransitifs n'exigent pas de complément d'objet ; ils se construisent uniquement avec un sujet. Par exemple, dans la phrase Mery dort, le verbe dort suffit à former une phrase complète. Catégoriellement, un verbe intransitif est donc représenté par la catégorie $N \setminus S$, puisqu'il n'attend qu'un sujet pour produire une phrase.

III.5) Lambek (1958-1961) : Calcul de Lambek

Joachim Lambek était un mathématicien et linguiste canadien. Il a développé le Calcul de Lambek dans les années 1958-1961 comme une extension de la grammaire catégorielle originale d'Yehoshua Bar-Hillel.

Le calcul de Lambek est également connu sous le nom de « Grammaires Catégorielles ».

Comme les grammaires AB, les grammaires catégorielles de Lambek comportent un ensemble de catégories (les catégories de base et les foncteurs) comme le montre (Godart-Wending, 2003) et Sammoud Mohamed Sadok dans son mémoire intitulé « Applications linguistiques multilingues : apport des grammaires catégorielles et de la logique combinatoire », (Sadok, 2010).

Une Grammaire Catégorielle est déterminée par différents types d'unités linguistiques. Toute information pour vérifier la bonne connexion des phrases (plus généralement des syntagmes) est donnée par les types assignés aux unités linguistiques de la grammaire.

Dans le calcul de Lambek, il a été introduit un ensemble de règles facilitant l'analyse des phrases d'un langage naturel. Ces règles comprennent :

- Les types de base sont des types ;
- Si X et Y sont des types alors X/Y et $X \setminus Y$ sont des types ;
- L'opérateur de concaténation, noté par le symbole $+$, permet de combiner deux types. Si X et Y sont des types, alors $X + Y$ est également un type.
- La règle de réduction (la notation $X \rightarrow Y$ signifie que le type X se réduit au type Y . De même, la notation $X \leftrightarrow Y$ signifie à la fois que le type de X se réduit au type Y et que le type Y se réduit au type X).
- La réflexivité : $X \rightarrow X$
- L'associativité : $(X+Y) + Z \rightarrow X + (Y + Z)$
 $X + (Y + Z) \rightarrow (X + Y) + Z$
- La transitivité : Si $X \rightarrow Y$ et $Y \rightarrow Z$ alors : $X \rightarrow Z$
- Si $X + Y \rightarrow Z$ Alors $X \rightarrow Z / Y$
- Si $X + Y \rightarrow Z$ Alors $Y \rightarrow Z \setminus X$
- Si $X \rightarrow Z / Y$ Alors $X + Y \rightarrow Z$
- Si $Y \rightarrow Z \setminus X$ Alors $X + Y \rightarrow Z$

À partir des règles présentées ci-dessus, on peut déduire les théorèmes suivants :

- $X \rightarrow (X+Y) / Y$
- $(Z/Y) + Y \rightarrow Z$
- $Y \rightarrow Z \setminus (Z / Y)$
- $(Z/Y) + (Y/X) \rightarrow (Z / X)$
- $Z/Y \rightarrow (Z/X) / (Y/X)$
- $(Y \setminus X) / Z \rightarrow (Y / X) \setminus Z$
- $(X / Y) / Z \rightarrow X / (Z + Y)$
- Si $X \rightarrow X'$ et $Y \rightarrow Y'$ Alors $X + Y \rightarrow X' + Y'$
- Si $X \rightarrow X'$ et $Y \rightarrow Y'$ Alors $X / Y' \rightarrow X' / Y$

III.6) Steedman (1989) : La grammaire catégorielle combinatoire

Mark Jerome Steedman était un linguiste informaticien connu pour ses travaux en syntaxe, sémantique et traitement automatique des langues. En 1989, l'une de ses contributions notables était son travail sur la Grammaire Catégorielle Combinatoire (CCG, pour Combinatory Categorical Grammar). CCG est un type de grammaire catégorielle qui construit des structures syntaxiques et sémantiques en combinant des catégories lexicales à l'aide d'un ensemble de combinatoires.

Elle adopte les mêmes règles de Grammaires Catégorielles de Ajdukiewicz et de Bar-Hillel ainsi que certaines propriétés du calcul de LAMBEK mais introduit aussi de nouvelles catégories autres que les catégories de base(S/N) :

- La catégorie pp désignant un « Groupe Prépositionnel » (exemple : au Sénégal, dans mon pays).
- La catégorie VP désignant un « Syntagme Verbal » (exemple : range la chambre, prépare le repas)
- La catégorie NP désignant un « Syntagme Nominal » (exemple : le thé, le vin rouge)
- Si X et Y sont des types orientés alors X/Y et $X \setminus Y$ sont des types orientés

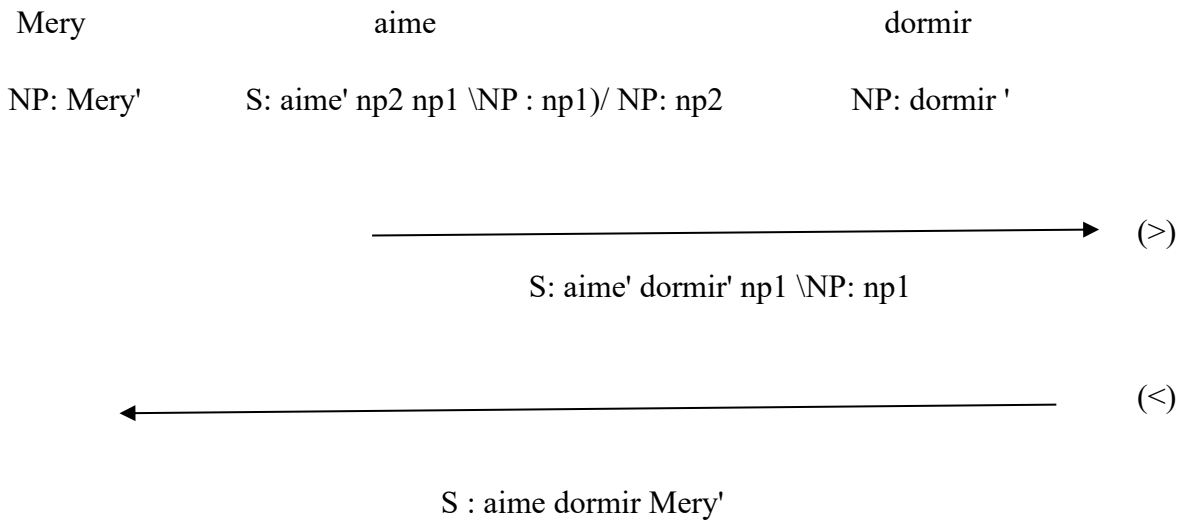
Les verbes sont associés à une catégorie syntaxique qui leur attribue le statut de foncteur. Steedman a donné comme exemple le verbe aimer et lui a attribué comme type syntaxique :

$$(S : \text{aimer}' \text{ np2 } \text{ np1} \setminus \text{NP} : \text{ np1}) / \text{NP} : \text{ np2}$$

Tentons maintenant d'expliquer cette attribution en l'appliquant à la phrase suivante : Mery aime dormir.

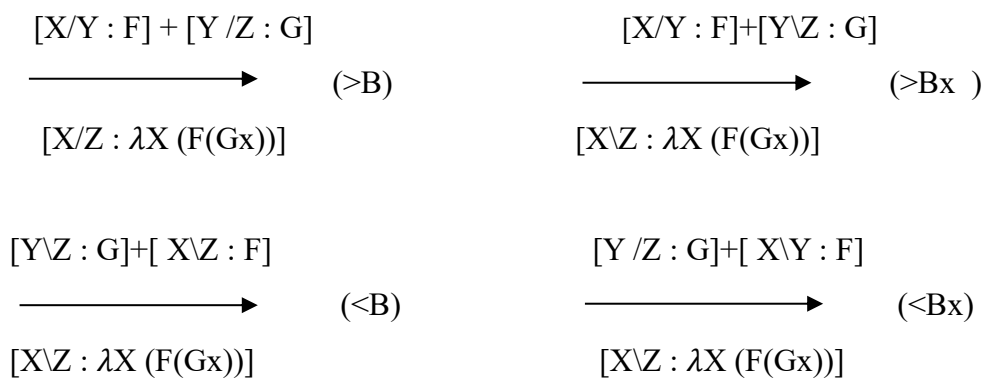
Rappelons qu'avec cette notation de Steedman (1989), les types syntaxiques sont en lettres majuscules, les constantes sémantiques portent des symboles ('), Les lettres minuscules sont des variables sémantiques. Nous allons appliquer les règles de réduction avant et arrière :

- $A/B, B \rightarrow A$ FA (Forward Application) (Forward Application $>$)
- $B, B \backslash A \rightarrow A$ BA (Backward Application) (Backward Application $<$)



Steedman a introduit de nouvelles règles pour enrichir le système des grammaires catégorielles. Ces règles ont été présentées dans le mémoire de Adam Joly intitulé « Du document textuel à sa carte sémantique fonctionnelle », (Joly, 2009) et également soutenue par Amsili dans son étude sur les Grammaires Catégorielles Combinatoires (Amsili, 2015).

- Des règles de composition fonctionnelle qui spécifient comment les catégories syntaxiques peuvent être combinées de manière systématique pour former des structures syntaxiques plus complexes. Elles sont au nombre de quatre :



- Des règles de changement de type qui permettent à des arguments de devenir des fonctions. Elles sont au nombre de quatre :

$$\begin{array}{ccc} [X : x] & \longrightarrow & (>T) \\ \hline [Y / (Y \setminus X) : \lambda F(Fx)] \end{array} \qquad \begin{array}{ccc} [X : x] & \longrightarrow & (>Tx) \\ \hline [Y / (Y/X) : \lambda F(Fx)] \end{array}$$

$$\begin{array}{ccc} [X : x] & \longrightarrow & (<T) \\ \hline [Y \setminus (Y/X) : \lambda F(Fx)] \end{array} \qquad \begin{array}{ccc} [X : x] & \longrightarrow & (<Tx) \\ \hline [Y \setminus (Y \setminus X) : F(Fx)] \end{array}$$

- Des règles de substitution fonctionnelle conçues pour maintenir la transparence de la correspondance entre la syntaxe et la sémantique, ce qui est une caractéristique importante de la grammaire catégorielle combinatoire de Steedman. Elles sont au nombre de quatre :

$$\begin{array}{ccc} ((X/Y)/Z : F) - (Y/Z : G) & \longrightarrow & (>S) \\ \hline (X/Z) : \lambda x (Fx (Gx)) \end{array} \qquad \begin{array}{ccc} ((X/Y) \setminus Z : F) - (Y \setminus Z : G) & \longrightarrow & (>Sx) \\ \hline (X/Z) : \lambda x (Fx (Gx)) \end{array}$$

$$\begin{array}{ccc} (YZ : G) - ((X \setminus Y) \setminus Z : F) & \longrightarrow & (<S) \\ \hline (X \setminus Z) : \lambda x (Fx (Gx)) \end{array} \qquad \begin{array}{ccc} Y/Z : G) - ((X \setminus Y) / Z : F) & \longrightarrow & (<Sx) \\ \hline (X/Z) : \lambda x (Fx (Gx)) \end{array}$$

III.7) [Shaumyann\(1998\) : La grammaire applicative universelle](#)

Sebastian Konstantinovich Shaumyan (ou Chaumian) était un linguiste soviétique et arménien, né le 26 octobre 1935 et décédé le 3 septembre 2002. Il a fait des contributions significatives dans le domaine de la linguistique théorique et de la sémantique.

Le modèle linguistique de la grammaire applicative universelle (Shaumyan, 1998) insiste de façon très explicite sur la pertinence des niveaux de représentation des langues avec interaction entre ces différents niveaux (Biskri & Descles, 1995). Leur principal objectif étant une étude des propriétés formelles des systèmes langagiers et une analyse de leurs structures logiques. Shaumyan avait comme principal objectif de mettre en œuvre une grammaire applicative universelle.

Pour ce faire, il a permis de faire une distinction entre deux niveaux d'analyse et de représentation des langues:

- le niveau phénotypique qui est un niveau concret, connu sous le nom de « niveau de surface » ou encore « niveau observable ». Il représente l'ordre et la position des mots dans une phrase et il se caractérise par la morphologie, la syntaxe, etc. d'une langue donnée.
- Le niveau génotype qui est un niveau abstrait va permettre de décrire les langues phénotypiques. Il est basé sur le principe « opérateur/opérande ». Il est organisé autour de l'utilisation des outils de la logique combinatoire de Curry (1958). On l'appelle langage génotypique (Descles, 2003).

Shaumyan distingue trois classes d'expressions linguistiques essentielles :

- Les noms des objets qui sont appelés « termes » (terms).
- Les noms des situations qui sont appelés « phrases » (sentences).
- Les moyens nécessaires pour construire les noms des objets et les noms des situations, qui sont appelés « opérateurs ».

III.8) Descles (1989) : Grammaire applicative et cognitive

Jean-Pierre Descles était un linguiste français, spécialisé dans plusieurs domaines de la linguistique, y compris la linguistique théorique, la sémantique, la syntaxe et la pragmatique. Il fut l'initiateur de la Grammaire Applicative et Cognitive qui n'est qu'une version étendue de la grammaire applicative universelle de Shaumyan.

L'apport majeur de la grammaire applicative et cognitive de Descles est l'introduction d'un troisième niveau d'analyse et de représentation des langues : le niveau cognitif (Biskri & Descles, 1995).

En effet, la grammaire applicative et cognitive reprend les niveaux de représentation des langues naturelles, que sont le phénotype et le génotype mais en introduisant un troisième niveau où seront représentées les significations des prédicats par des schémas qui sont les générateurs des représentations sémantiques des phrases : le niveau cognitif (Descles, 1990).

IV) La logique combinatoire

La logique combinatoire a été développée pour analyser logiquement les paradoxes de (Russell) (Irvine, Deutsch, & Marshall, 2025) et la notion de substitution. Dans cette partie, nous parlerons essentiellement de la logique combinatoire. En effet, celle-ci sera nécessaire pour pouvoir présenter notre prochain chapitre qui parlera essentiellement du modèle catégoriel que nous allons utiliser dans notre travail à savoir : la grammaire catégorielle combinatoire applicative.

En ce sens, nous allons parler un peu de l'historique de la logique combinatoire, faire une présentation de quelques notions de celle-ci et finir par la description de quelques combinateurs que nous allons utiliser dans notre travail.

IV.1) Cadre historique

La logique combinatoire est une théorie logique introduite par Moses Schönfinkel en 1920 lors d'une conférence et reprise par Curry et Feys (1958) dans le but d'apporter une solution logique à certains paradoxes comme celui de Russel (Irvine, Deutsch, & Marshall, 2025) et de supprimer les besoins en variable en mathématique.

Elle est aussi en rapport avec le λ -calcul de Church (1941) (Barendregt, 1984). Plus récemment, elle a été utilisée en informatique comme modèle théorique de calcul et comme base pour la conception de langages de programmation fonctionnels (Guilbert, Sauzay, & Descles, 2016).

Actuellement, elle est un système applicatif sans variable, utilisant des opérateurs abstraits appelés combinateurs.

IV.2) Combinateurs logiques

Les combinateurs logiques sont des opérateurs qui prennent des entrées logiques et produisent une sortie en fonction de règles logiques prédéterminées. En effet, ils permettent de construire, à partir d'opérateurs, des opérateurs de plus en plus complexes. L'action d'un combinateur sur un argument est définie par une règle spécifique appelée β -réduction (Goubault-Larrecq, 2001). Cette dernière établit une relation entre une expression avec un combinateur et une expression équivalente sans combinateur.

Nous allons dans ce qui suit présenter quelques combinateurs (Ginisti, 1988) (Joly, 2009), (Hassani, 2016) et (Bitam, 2022).

IV.2.1) Le combinateur d'identité (I)

Le combinateur I est fondamental dans la théorie des combinateurs logiques. En effet, il correspond à la fonction identité, c'est-à-dire une fonction qui retourne exactement son argument, sans lui appliquer aucune

transformation. Il se joint à un opérateur f pour former l'opérateur complexe $I f$. Son action est formellement définie selon la β -réduction qui suit :

$$I f \rightarrow f$$

Il est représenté par l'expression $\lambda x \rightarrow x$.

Autrement dit, appliquer I à un objet f revient simplement à obtenir f .

Imaginons que l'on applique le combinateur I à une valeur numérique :

$$I(5) = 5$$

Ainsi, I ne modifie ni une valeur ni une fonction : il « transmet » simplement ce qu'on lui donne.

IV.2.2) Le combinateur de composition (B)

Le combinateur B est un opérateur logique qui permet de composer deux fonctions. Autrement dit, il prend deux fonctions comme arguments et produit une nouvelle fonction correspondant à la composition de ces deux fonctions.

En effet, il s'associe à deux opérateurs f et g afin de construire l'opérateur complexe $B f g$ tel que pour un argument x on obtienne la règle de réduction suivante :

$$B f g x \rightarrow f(g x)$$

Sa représentation en λ -expression est : $\lambda x y z \rightarrow x (y z)$.

Exemple :

Prenons deux fonctions simples : $f(x)=x+1$ et $g(x)=2x$

Si l'on compose ces fonctions avec le combinateur B :

$$B f g = \lambda(z). f g(z)$$

Appliquons-le à un argument, par exemple 3 :

$$B f g 3 = f(g(3)) = f(6) = 6+1 = 7$$

IV.2.3) Le combinateur de substitution (S)

Le combinateur S prend trois arguments et répartit le troisième entre les deux premiers. Considérons deux opérateurs f et g , f étant binaire et g unaire, associés par le combinateur S afin de former l'opérateur complexe $S f g$. Une fois Appliqué à un opérande X , ce combinateur agit comme suit :

$$S f g x \rightarrow f x (g x)$$

Il est représenté par le λ -expression est $\lambda x y z \rightarrow x z (yz)$.

Exemple :

Prenons deux fonctions simples : $f(x)=x+1$ et $g(x)=2x$ et appliquons le combinateur S

$$S f g 3 = f 3 (g 3)$$

$$S f g 3 = f (3, g (3)) = f(3, 6) = 3+6 = 9$$

IV.2.4) Le combinateur de coordination (ϕ)

Ce combinateur doit son nom à son importance dans l'étude du cas de coordination dans le domaine de la linguistique. C'est un combinateur à quatre arguments, il s'applique à une séquence comme : ABCz. L'action du combinateur de coordination est définie par la règle β -réduction suivante :

$$\phi ABCz \rightarrow A(Bz)(Cz)$$

Considérons la phrase : Mery [mange et parle]

On peut la voir comme une coordination de deux prédicats appliqués au même sujet Mery.

- $B = \lambda x. \text{mange}(x)$
- $C = \lambda x. \text{parle}(x)$
- $A = \lambda p q. p \wedge q$
- $z = \text{Mery}$

Alors :

$$\phi ABCz \rightarrow A(Bz)(Cz) = (\text{mange}(\text{Mery})) \wedge (\text{parle}(\text{Mery}))$$

Donc ϕ permet de factoriser le sujet et de composer correctement la coordination syntaxique et sémantique.

IV.2.5) Le combinateur de distribution ψ

Le combinateur ψ compose avec les opérateurs f et g afin de construire l'opérateur complexe $\psi f g$. Il permet de distribuer un même opérateur unaire g sur deux arguments avant de les passer à un opérateur binaire f. Cet opérateur agit sur les opérands x et y en fonction de la β -réduction suivante :

$$\psi f g x y \rightarrow f(g x) (g y)$$

Supposons que :

- $f(p,q) = p \wedge q$
- $g(x) = \text{lave}(x, \text{Mery})$

Alors:

$$\begin{aligned} \psi fg \text{ papa maman} &\rightarrow f(g(\text{papa}), g(\text{maman})) \\ &= (\text{lave}(\text{papa}, \text{Mery})) \wedge (\text{lave}(\text{maman}, \text{Mery})) \end{aligned}$$

Ce qui correspond à la phrase : Mery lave papa et maman

IV.2.6) Le combinateur de changement de type (C*)

Il permet de changer le statut d'un opérande pour en faire un opérateur. Il sert à échanger l'ordre des arguments.

Ce combinateur est défini par la règle β -réduction suivante :

$$C^*YZ \rightarrow ZY$$

$$C^*XY \rightarrow YX$$

IV.2.7) Le combinateur de permutation (C)

Ce combinateur permet d'exprimer la commutativité d'une opération. Il se traduit par la règle suivante :

$$CFXY \rightarrow FYX$$

Il est principalement utilisé pour la description des verbes symétriques (épouser, rencontrer, croiser, etc.).

Exemple : Mery rencontre Maman.

L'objectif est de montrer que cette proposition est équivalente à Maman rencontre Mery. L'expression combinatoire relative à Mery rencontre Maman est rencontre Mery Maman.

Posons : rencontre = C rencontre.

rencontre Mery Maman \Leftrightarrow C rencontre Mery Maman.

L'application de la règle de réduction associée au combinateur C nous donne : rencontre Maman Mery.

Mery rencontre Maman \Leftrightarrow Maman rencontre Mery

IV.2.8) Le combinateur d'effacement K

Son fonctionnement est de prendre deux arguments en entrées et de rendre le premier comme résultat.

Le combinateur d'effacement K est traduit par la règle β -réduction suivante :

$$K \text{ fx} \rightarrow f$$

Exemple : $f = \text{Fatima}$ et $x = \text{dort}$

$$K \text{ fx} \rightarrow K \text{ Fatima dort} \rightarrow \text{Fatima}$$

IV.2.9) Le combinateur de duplication W

Considérons un opérateur f et un opérande x . Le combinateur de duplication W s'associe à f pour construire le combinateur complexe $W f$. Il agit sur un opérande x comme suit :

$$W \text{ fx} \rightarrow f \text{ xx}$$

W est surtout utilisé dans le cas des prédicats réfléchis (se-laver, se-blesser, se-raser, ...). Prenons l'exemple du prédicat se-laver et appliquons-le à Maman. L'expression résultante sera se-lave Maman.

Posons se-lave = $W \text{ lave}$, le remplacement de cette expression combinatoire dans l'expression se-lave Maman donne $W \text{ lave Maman}$.

La réduction de cette nouvelle expression combinatoire donne comme résultat l'expression : lave Maman Maman. Cela montre que les deux propositions se-lave Maman et lave Maman Maman sont sémantiquement équivalentes.

IV.2.10) Les combinateurs complexes

Ils sont définis par l'association de plusieurs combinateurs élémentaires. On peut par exemple avoir BCC, BC*, SSK, BSK etc.

L'action d'un combinateur complexe est déterminée par l'application successive de ses combinateurs élémentaires de gauche à droite ce qui transforme l'expression pas à pas, en suivant la règle :

Exemple : SSK

On sait que :

- $S \text{ fgx} \rightarrow f \text{ x(gx)}$
- $K \text{ ab} \rightarrow a$

Donc SSK peut se développer en suivant ces réductions :

$$SSK \text{ x} = (SSK) \text{ x} \rightarrow (S \text{ x}(K \text{ x})) \rightarrow \text{x}$$

IV.2.10.1) La puissance d'un combinateur

Il est possible d'attribuer une puissance à un combinateur. Cela aura pour effet de réitérer n fois l'action du combinateur en vertu de la définition qui suit :

Si c est un combinateur alors c^n itère n fois l'action du combinateur c tel que :

$$c^1 = c \text{ et } c^n = B c c^{n-1}$$

Soit le combinateur C défini par :

$$C(x)=x+1$$

Alors :

- $c^1(x)=c(x)=x+1$
- $c^2(x)=B c c^1(x) = c(c(x)) = (x+1)+1 = x+2$
- $c^3(x) = c(c(c(x))) = x+3$

Donc $c^n(x) = x + n$

IV.2.10.2) Les combinateurs à distance

Ils tirent cette appellation du fait qu'ils agissent à distance dans l'expression selon le principe suivant :

Si c est un combinateur alors c_n diffère son action de n pas. $c_n = B^n c$.

Exemples: $C_2 a b c d e \rightarrow B^2 C a b c d e \rightarrow a b c e d$

$B_3 a b c d e \rightarrow B^3 B a b c d e \rightarrow a b c (d e)$

IV.2.11) Théorème sur les combinateurs

Tous les combinateurs peuvent être exprimés en fonction des combinateurs S et K (Curry, 1958).

Ainsi nous avons les exemples suivants :

$$I = S K K$$

$$B = S (K S) K$$

$$W = S S (K (S K K))$$

$$C = S (B B S) (K K)$$

$$\phi = B (B S) B$$

$$\psi = \psi (\psi (\psi B)) B (K K)$$

IV.2.12) La forme normale

IV.2.12.1) Définition

Une expression est considérée comme étant sous forme normale lorsqu'elle ne peut plus être réduite.

Exemple : $a (b c)$. Soit E' une expression sous forme normale et E une expression qui n'est pas sous forme normale. Si E est réduite à E' alors nous dirons que E' est la forme normale de E .

Notons cependant que certaines expressions combinatoires n'ont pas de formes normales.

Exemple 1 : l'expression $W W W$.

En tentant de la réduire, on constate que le combinateur W qui a comme argument un autre argument W est dupliqué. Cela entraîne une situation où la réduction ne pourra jamais se faire, car elle continuera indéfiniment, empêchant ainsi d'atteindre une forme normale : $W W W \implies W W W \implies W W W \implies W W W \implies \dots$

Exemple 2 : $(SII)(SII)$

Rappel des définitions : $I = \lambda x.xI$ et $S = \lambda f g x. fx(gx)$

Donc : $Sfgx \rightarrow fx(gx)$

Appliquons la définition de S avec $f=I$ et $g=I$

$SIIx \rightarrow Ix(Ix)$

Or $Ix=x$ donc : $SIIx \rightarrow xx \rightarrow SII=\lambda x.xx$

Notons $M=SII$ Alors : $M=\lambda x.xx$

Donc : $(SII)(SII) = MM = (\lambda x.xx)(\lambda x.xx)$

En essayant de réduire $(\lambda x.xx)(\lambda x.xx) \rightarrow \dots$, on a une boucle infinie donc l'expression $(SII)(SII)$ n'a pas de forme normale.

IV.2.12.2) Théorème de Church-Rosser

Il pose la question de l'unicité de la forme normale, à savoir si une expression combinatoire peut être réduite à plusieurs formes normales ou bien à une seule.

Si une expression combinatoire X se réduit en une expression combinatoire Y_1 et si X se réduit en une autre expression combinatoire Y_2 alors il existe une expression combinatoire Z tel que Y_1 et Y_2 se réduisent en Z (Desclés, 1988).

Ce théorème révèle le fait que la logique combinatoire possède la propriété de Church-Rosser pour les relations transitives. Nous résumons cette propriété par le diagramme suivant :

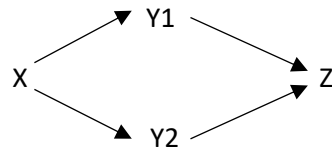


Figure 1 : Propriété de Church-Rosser

A partir de ce théorème, nous pouvons dégager les corollaires ci-dessous :

Corollaire 1 : Une expression combinatoire a au plus une seule forme normale.

Corollaire 2 : Si $X=Y$ alors il existe un Z tel que $X \longrightarrow Z$ et $Y \longrightarrow Z$.

Corollaire 3 : Si $X = Y$ et Y est une forme normale, alors $X \longrightarrow Y$.

Corollaire 4 : Si $X = Y$ alors soit X et Y n'ont pas de forme normale, soit X et Y ont la même forme normale.

Corollaire 5 : Si X et Y sont deux expressions applicatives qui ont des formes normales différentes alors $X \neq Y$

V) Conclusion

Ce chapitre est très intéressant en ce qu'il nous a permis de nous familiariser avec le principe des systèmes applicatifs typés, de mieux comprendre l'évolution des grammaires catégorielles ainsi que leur utilisation dans l'analyse des phrases allant des catégories de signification de Husserl à la grammaire applicative et cognitive de Desclés. Il est aussi très important dans notre travail car il nous a permis de mieux comprendre la logique combinatoire qui nous a aidé à mieux découvrir les combinateurs. En effet, ces derniers seront intégrés dans la grammaire catégorielle combinatoire pour un nouveau modèle catégoriel appelé grammaire catégorielle combinatoire applicative, ce qui constituera l'élément central de notre prochain chapitre.

Chapitre III : Méthodologie

I) Introduction

La grammaire catégorielle est une approche théorique de la syntaxe qui met l'accent sur les catégories lexicales et les règles de combinaison syntaxique associées. Dans le chapitre précédent, nous avons retracé son évolution, en partant de la version de Lesniewski jusqu'à la grammaire applicative cognitive développée par Jean-Pierre Descles.

L'objectif principal de notre mémoire est d'explorer l'analyse syntaxique des phrases à l'aide de la Grammaire Catégorielle Combinatoire Applicative (GCCA). Cette analyse vise à détecter les erreurs présentes dans les phrases ou les textes, à localiser leur position, et à proposer des suggestions de correction adaptées aux apprenants de français langue seconde (Cuq, 2013).

Ce chapitre, essentiel dans le cadre de notre mémoire, sera consacré à :

- Le workflow, détaillant les différentes étapes de notre processus d'analyse ;
- Le modèle catégoriel utilisé, à savoir la Grammaire Catégorielle Combinatoire Applicative (GCCA). Ce modèle enrichit la grammaire catégorielle de Steedman en intégrant une association entre les règles syntaxiques et les combineurs logiques (S, B, C) ainsi qu'une utilisation de métarègles permettant de contrôler les opérations de changement de type ;
- Quelques exemples de phrases dérivées en CCG
- L'outil de suggestion de correction, essentiel pour guider les apprenants dans leur processus d'apprentissage.

II) Le Workflow

Un workflow (flux de travail) représente une séquence d'activités ou d'étapes nécessaires pour accomplir une tâche ou un processus spécifique. Dans le contexte de notre projet, le workflow décrit les étapes nécessaires pour analyser et traiter les phrases. La figure 2 suivante représente une explication générale du workflow de notre solution :

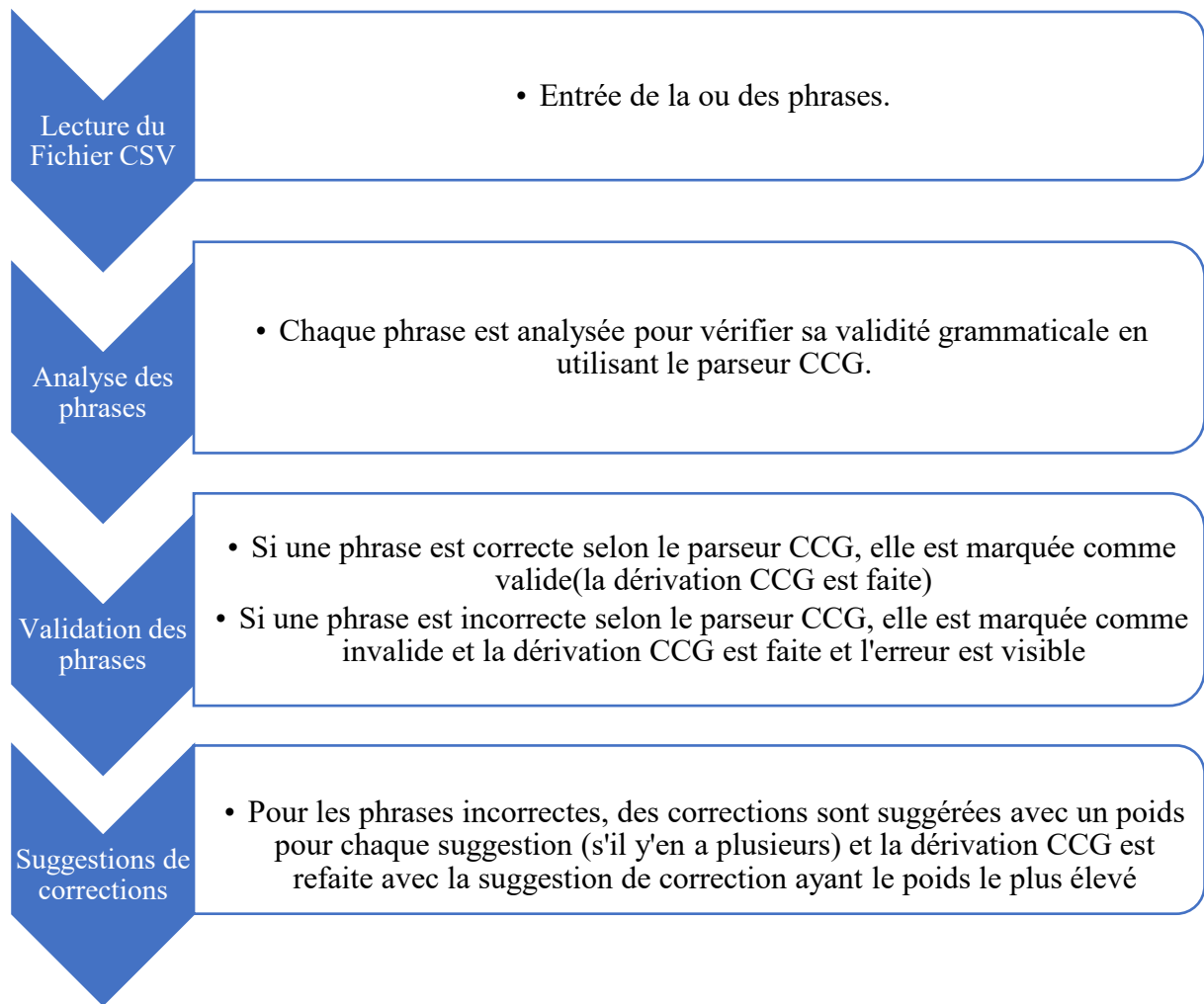


Figure 2 : Workflow

En résumé, un utilisateur soumet les phrases qu'il souhaite analyser. Chaque phrase est d'abord vérifiée pour sa validité syntaxique à l'aide de CCG, puis les erreurs linguistiques sont détectées à l'aide de LanguageTool. Si la phrase est valide, une dérivation CCG est effectuée, en fonction de notre lexique. En revanche, si la phrase est invalide, une dérivation CCG est réalisée et les erreurs identifiées, leur position précise, et des suggestions de correction sont proposées (avec un poids pour chaque suggestion). Ces suggestions seront ensuite utilisées pour analyser à nouveau la phrase.

III) La grammaire catégorielle combinatoire et applicative : GCCA

III.1) Principe de la GCCA

La GCCA a été développée par Ismaïl Biskri (1995) et Jean-Pierre Descles en 1996. Elle s'inspire du modèle général de la Grammaire Applicative et Cognitive qui n'est qu'une version étendue de la grammaire applicative universelle de Shaumyan. Rappelons que dans ces modèles, une analyse du langage doit postuler trois niveaux que sont : le niveau phénotype, le niveau génotype et le niveau cognitif.

La GCCA a été proposée pour :

- Régler le problème de la pseudo-ambiguïté (Biskri & Descles, 1995) (fait référence à une situation où une phrase semble ambiguë du fait de son sens)
- Permettre de reconnaître les phrases syntaxiquement correctes
- Avoir une interprétation sémantique fonctionnelle de l'énoncé

Tout ceci, ne sera possible qu'avec l'introduction d'un système formel permettant de relier la structure morphosyntaxique à la structure applicative et se fera à l'aide des combinateurs logiques de Curry S, B, C* vus dans le chapitre précédent.

Rappelons aussi que la GCCA hérite de toutes les règles des anciennes versions (principalement celles de Steedman) tout en y ajoutant de nouvelles.

III.2) Les règles de la GCCA

Les Grammaires Catégorielles assignent des catégories syntaxiques à chaque unité linguistique.

Dans la GCCA, les mêmes règles de grammaire catégorielle combinatoire de Steedman sont adoptées mais pour passer d'une structure concaténée à une structure applicative, les règles de Steedman introduisent les combinateurs de Curry S, B et C* ((Biskri & Descles, 1995) ; (Biskri, Meunier, & Nault, 1988)).

Les catégories syntaxiques sont des types orientés engendrés à partir de types de base et de deux opérateurs constructifs '/' et '\'.

- (i) N (syntagme nominal) et S (phrase) sont des types de base.
- (ii) Si X et Y sont des types orientés alors X/Y et X\Y sont des types orientés.
- (iii) u1-u2 représentent la concaténation dans le phénotype des unités linguistiques u1, u2.

Les règles de la GCCA sont :

➤ Règles d'application :

$$\frac{[X/Y : u_1] - [Y : u_2]}{\text{-----} >} \quad \frac{[Y : u_1] - [X \setminus Y : u_2]}{\text{-----} <} \\ [X : (u_1 u_2)] \qquad [X : (u_2 u_1)]$$

➤ Règles de changement de type

$$\frac{[X : u]}{\text{-----} > T} \quad \frac{[X : u]}{\text{-----} < T} \\ [Y / (Y \setminus X) : (C^*, u)] \qquad [Y \setminus (Y / X) : (C^* u)]$$

$$\frac{[X : u]}{\text{-----} > T_X} \quad \frac{[X : u]}{\text{-----} < T_X} \\ [Y / (Y / X) : (C^*, u)] \qquad [Y \setminus (Y \setminus X) : (C^* u)]$$

Exemple : La santé est une richesse.

(1) $[N/N : La] - [N : santé] - [(S \setminus N) / N : est] - (N/N : une) - [N : richesse]$

(2) $[N : (La santé)] - [(S \setminus N) / N : est] - [N/N : une] - [N : richesse] \quad >$

(3) $[S / (S \setminus N) : (C^*(La santé))] - [(S \setminus N) / N : est] - [N/N : une] - [N : richesse] \quad > T$

(4) $[S / N : B (C^*La santé) est] - [N/N : une] - [N : richesse] \quad > B$

(5) $[S : ((B (C^*La santé) est) une richesse)] \quad >$

Lors de la troisième étape, un changement de type a été appliqué sur l'opérande la santé pour construire l'opérande C*(La santé) de type S/(S\N). En utilisant la règle de composition B, C*(La santé) est devenue un opérateur plus complexe : (B (C*La santé) est) de type S/N à l'étape 4. Et enfin, à l'étape 5, ce dernier s'applique à l'opérande une richesse afin que l'on obtienne l'expression applicative de type S : S : ((B (C*La santé) est) est) une richesse.

➤ Règles de composition

$$\begin{array}{ccc} [X/Y : u_1] - [Y/Z : u_2] & & [Y\Z : u_1] - [X\Y : u_2] \\ \text{-----} > B & & \text{-----} < B \\ [X/Z : (B u_1 u_2)] & & [X\Z : (B u_2 u_1)] \end{array}$$

$$\begin{array}{ccc} [X/Y : u_1] - [Y\Z : u_2] & & [Y/Z : u_1] - [X/Y : u_2] \\ \text{-----} > Bx & & \text{-----} < Bx \\ [X\Z : (B u_1 u_2)] & & [X/Z : (B u_2 u_1)] \end{array}$$

➤ Règles de substitution fonctionnelle

$$\begin{array}{ccc} [(X/Y)/Z : u_1] - [Y/Z : u_2] & & [Y\Z : u_1] [(X\Y)\Z : u_2] \\ \text{-----} > S & & \text{-----} > S \\ [X/Z : (S u_1 u_2)] & & [X\Z : (S u_2 u_1)] \end{array}$$

$$\begin{array}{ccc} [(X/Y)Z : u_1] - [Y\Z : u_2] & & [Y/Z : u_1] [(X\Y)/Z : u_2] \\ \text{-----} > Sx & & \text{-----} > Sx \\ [X\Z : (S u_1 u_2)] & & [X/Z : (S u_2 u_1)] \end{array}$$

III.3) Les étapes de la GCCA

La GCCA permet de passer d'une structure concaténée à une structure applicative fonctionnelle en traitant les mots et les phrases comme des fonctions et des arguments qui se combinent hiérarchiquement pour former une structure syntaxique complète et cohérente. Ceci se fera selon un processus en deux phases:

- La vérification de la bonne connexion syntaxique et la construction de structures applicatives avec les combinateurs de Curry sont introduites progressivement dans le processus d'analyse. Cette vérification s'effectue dans le phénotype (structure morphosyntaxique), mais l'expression obtenue appartient au langage génotype (structure applicative).
- La construction d'une structure applicative par l'élimination successive des combinateurs introduits à la première étape en utilisant les règles de β -réduction des combinateurs. Cette étape nous permettra d'avoir une forme normale(irréductible). Ce dernier calcul s'effectue entièrement dans le génotype (structure applicative).

Ces étapes sont résumées dans la figure ci-dessous :

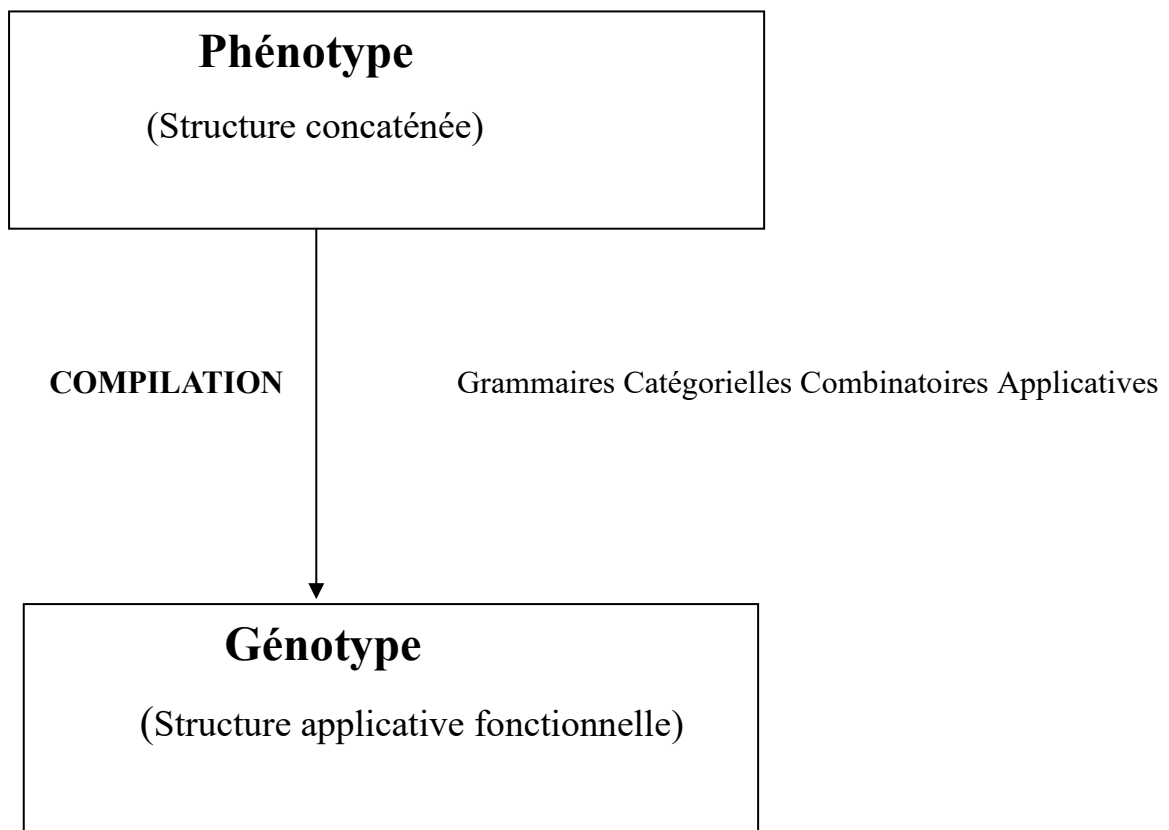


Figure 3:Etapes de la GCCA

III.4) La coordination de la GCCA

La coordination en grammaire catégorielle combinatoire et applicative (GCCA) est la manière dont des éléments similaires (comme des phrases, des noms, etc.) sont combinés pour former des structures coordonnées. En GCCA, la coordination est généralement modélisée en utilisant des catégories spéciales et des combinateurs qui permettent de combiner des éléments de même type.

Deux unités linguistiques peuvent être coordonnées pour donner une unité linguistique de type X si et seulement si chaque unité est de type X (Steedman (1989), Barry et Pickering (1990)).

La coordination est basée sur deux hypothèses:

- Hypothèse 1 : La catégorie construite qui suit immédiatement la conjonction « et » détermine le type de la coordination. Dès que nous rencontrons la conjonction « et », nous interrompons momentanément l'analyse quasi-incrémentale pour construire le second membre de la coordination. Cette procédure permet d'assurer une analyse correcte et cohérente de la phrase, en garantissant que la structure syntaxique prend en compte la nature de chaque élément coordonné, et facilite ainsi une interprétation plus précise et flexible des phrases coordonnées.
- Hypothèse 2 :
Lorsque nous rencontrons une coordination de type X, telle que définie par l'hypothèse 1, la catégorie du premier membre de la coordination détermine la nature de l'ensemble. L'application de cette hypothèse permet d'analyser les phrases coordonnées de manière incrémentale, tout en assurant une structure syntaxique correcte et complète.

III.5) Les métarègles

Les métarègles en grammaire catégorielle combinatoire et applicative permettent de gérer efficacement les structures syntaxiques tout en évitant l'explosion combinatoire. Ces métarègles, en définissant comment et quand appliquer les règles de changement de type, assurent une analyse syntaxique précise et contrôlée. En comprenant et en appliquant ces métarègles quand et où il faut, nous pouvons traiter des phrases complexes de manière systématique et rigoureuse. Il en existe dix que nous allons lister.

Les métarègles 1 à 8 sont utilisées pour la construction des types syntaxiques qui se combineront avec l'utilisation des règles d'application, de composition, de composition distributive

Soient u_1 et u_2 dans l'expression concaténée ' u_1-u_2 ' :

- Métarègle 1 : Si u_1 est de type N et u_2 de type $(Y \setminus N) / Z$, alors nous appliquons le changement de type avant ($>T$) à u_1 :

$$[N : u_1 \implies Y / (Y \setminus N) : (C * u_1)]$$

Exemple : Mery aime l'eau

$$[N : \text{Mery}] - [(S \setminus N) / N : \text{aime}] - (N / N : \text{l'}) - [N : \text{eau}]$$

$$[S / (S \setminus N) : (C * (\text{Mery}))] - [(S \setminus N) / N : \text{aime}] - [N / N : \text{l'}] - [N : \text{eau}]$$

Dans ce cas, $Y = S$ et $Z = N$

- Métarègle 2 : Si u_1 est de type X et u_2 de type $(Y \setminus X) / Z$, alors nous appliquons le changement de type avant ($>T$) à u_1 :

$$[N : u_1 \implies Y / (Y \setminus X) : (C * u_1)]$$

Exemple : Mery dort très profondément

$$[N : \text{Mery}] - [(S \setminus N) : \text{dort}] - [((S \setminus N) \setminus (S \setminus N)) / ((S \setminus N) \setminus (S \setminus N)) : \text{très}] - [(S \setminus N) \setminus (S \setminus N) : \text{profondément}]$$

$$[N : \text{Mery}] - [(S \setminus N) / ((S \setminus N) \setminus (S \setminus N)) : (C * \text{dort})] - [((S \setminus N) \setminus (S \setminus N)) / ((S \setminus N) \setminus (S \setminus N)) : \text{très}] - [(S \setminus N) \setminus (S \setminus N) : \text{profondément}]$$

Dans ce cas, $X = S \setminus N$; $Y = S \setminus N$ et $Z = (S \setminus N) \setminus (S \setminus N)$

- Métarègle 3 : Si u_2 est de type N et u_1 de type $(Y / N) \setminus X$, alors nous appliquons le changement de type arrière ($<T$) à u_2 :

$$[N : u_2 \implies Y \setminus (Y / N) : (C * u_2)]$$

- Métarègle 4 : Si u_2 est de type N et u_1 de type Y / S , alors nous appliquons le changement de type avant ($>T$) à u_2 :

$$[N : u_2 \implies S \setminus (S / N) : (C * u_2)]$$

- Métarègle 5 : Si u_1 et u_2 sont de type N et sont reliés par une conjonction, alors nous appliquons le changement de type avant ($>T$) à u_1 :
 $[N : u_1 \implies S / (S \setminus N) : (C * u_1)]$
- Métarègle 6 : Si u_2 est de type N et u_1 de type Y/X et qu'ils sont reliés par la conjonction et, alors nous appliquons le changement de type arrière ($<T$) à u_2 :
 $[N : u_2 \implies X \setminus (X/N) : (C * u_2)]$
- Métarègle 7 : Si u_1 est de type N et u_2 de type Y\X et qu'ils sont reliés par la conjonction et, alors nous appliquons le changement de type arrière ($<T$) à u_1 :
 $[N : u_1 \implies X \setminus (X/N) : (C * u_1)]$
- Métarègle 8 : Si u_1 est de type N et u_2 de type (X\N) /Y et qu'ils sont reliés par la conjonction et, alors nous appliquons le changement de type avant ($>T$) à u_1 :
 $[N : u_1 \implies X / (X \setminus N) : (C * u_1)]$

Les métarègles 9 et 10 sont utilisées pour le traitement de la coordination distributive de deux opérandes.

Soient u_1 et u_2 et u_3 dans l'expression concaténée ' $u_1-u_2-u_3$ '

- Métarègle 9 : Si u_1 et u_2 sont de type N et reliés par la conjonction et, u_3 de type (S\N) /X, alors nous appliquons le changement de type avant ($>T$) à u_1 et u_2 :
 $[N : u_1 \implies S / (S \setminus N) : (C * u_1)]$
 $[N : u_2 \implies S / (S \setminus N) : (C * u_2)]$
- Métarègle 10 : Si u_2 et u_3 sont de type N et reliés par la conjonction et, u_1 de type Y /N, alors nous appliquons le changement de type arrière ($<T$) à u_2 et u_3 :
 $[N : u_3 \implies (S \setminus N) / ((S \setminus N) / N) : (C * u_3)]$
 $[N : u_2 \implies (S \setminus N) / ((S \setminus N) / N) : (C * u_2)]$

III.6) La réorganisation structurelle

L'analyse syntaxique « de gauche à droite » est un processus où les éléments d'une phrase sont analysés dans l'ordre de leur apparition. Ce processus peut être non-déterministe du fait que des modificateurs postposés (les adverbes, les propositions relatives, ou certaines conjonctions, qui modifient des parties de la phrase après que ces parties ont été analysées) peuvent rendre l'analyse ambiguë. Dès lors, il est crucial de bien gérer ces derniers pour une analyse syntaxique cohérente et éviter l'explosion combinatoire d'où la réorganisation structurelle proposée par Biskri (1995).

Exemple : Mery commande une pizza rapidement.

Dans cet exemple, un changement de type a été appliqué sur l'opérande Mery pour construire l'opérande $C^*(Mery)$ de type $S/(S \setminus N)$. En utilisant la règle de composition B, $C^*(Mery)$ est devenue un opérateur plus complexe : $(B (C^*Mery) commande)$ de type S/N . Ce dernier s'applique à l'opérande une pizza afin que l'on obtienne l'expression applicative de type S : $((B (C^*Mery) commande) une pizza)$. Cependant, il ne peut être combiné avec rapidement ayant comme type $(S \setminus N) \setminus (S \setminus N)$. En effet, rapidement est un modificateur arrière qui a comme opérande (commande une pizza) positionné à sa gauche. Or comme nous l'avons souligné tantôt, l'analyse syntaxique "de gauche à droite" est un processus où les éléments d'une phrase sont analysés dans l'ordre de leur apparition. La solution serait alors de revoir notre analyse en faisant un retour arrière, ce qui ne fera qu'accroître la mémoire et le temps d'exécution. Pour pallier ces limites, le chercheur Biskri a proposé une opération de réorganisation structurelle qui sera un retour arrière « intelligent » qui aura pour but de permettre une analyse correcte et une élimination de la pseudo-ambiguïté.

Cette réorganisation structurelle s'effectuera en deux étapes successives :

La réorganisation du constituant qui consiste à isoler deux sous-catégories au sein de la structure syntaxique déjà construite puis à tester en utilisant les règles de combinaison syntaxiques de la grammaire catégorielle l'application potentielle du modificateur postposé pour voir s'il se combine avec l'une des deux sous-catégories à gauche. Lorsque nous obtiendrons un test positif, nous aurons une nouvelle structure applicative typée « équivalente » à la première.

Exemple : Maman achète une orange rapidement.

Les étapes de la réorganisation sont :

- le constituant construit : $[S : ((B (C^*Maman) achète) une orange)]$.

- Les deux sous-catégories sont : [S/N : (B (C* Maman) achète)] ; [N : une orange]
 - A la phase Test : [S/N : (B (C* Maman) achète)] ne se combine pas à gauche avec [(S\N)\(S\N) : rapidement] et [N : une orange] ne se combine pas à gauche avec [(S\N)\(S\N) : rapidement]
 - Réduction du combinateur B : [S : ((C* Maman) (achète une orange))]
 - Les deux sous-catégories sont : [S/(S\N) : (C* Maman)] ; [S\N : (achète une orange)]
 A la phase Test : [S/(S\N) : (C* Maman)] ne se combine pas à gauche avec [(S\N)\(S\N) : rapidement] et [S\N : (achète une orange)] se combine à gauche avec [(S\N)\(S\N) : rapidement]
- Arrêt du processus de réduction des combinateurs. Nous récupérons en sortie la catégorie : [S : ((C* Maman) (achète une orange))].

a) La décomposition grâce aux deux règles suivantes :

$$\begin{array}{ccc}
 [X : (u_1 u_2)] & & [X : (u_1 u_2)] \\
 \text{----->dec} & ; & \text{----->dec} \\
 [X/Y : u_1]- [Y : u_2] & & [Y : u_2]-[X\Y : u_1]
 \end{array}$$

Nous lisons ces règles comme suit :

- Pour (>dec): Si nous avons une structure applicative $(u_1 u_2)$ de type X, avec u_1 de type X/Y et u_2 de type Y, alors nous pouvons construire une nouvelle expression concaténée formée des deux catégories $[X/Y : u_1]$ et $[Y : u_2]$.
- Pour (<dec): Si nous avons une structure applicative $(u_1 u_2)$ de type X, avec u_1 de type X\Y et u_2 de type Y, alors nous pouvons construire une nouvelle expression concaténée formée des deux catégories $[Y : u_2]$ et $[X\Y : u_1]$.

Ces deux règles sont utilisées pour donner suite à l'obtention des deux catégories, ce qui facilite l'analyse syntaxique.

Dans notre exemple précédent : Maman achète une orange rapidement, nous avons obtenu :
 Les deux sous-catégories : [S/(S\N) : (C* Maman)] ; [S\N : (achète une orange)].

$$[S/(S\N): (C* Maman)]- [S\N: (achète une orange)]-[(S\N)\(S\N) : tendrement] \text{ dec}>$$

[S/(S\N) : (C* Maman)] - [S\N : (rapidement (achète une orange))] (<)

[S : ((C* Maman) (rapidement (achète une orange)))] (>)

[S : (rapidement (achète une orange) (C* Maman))]

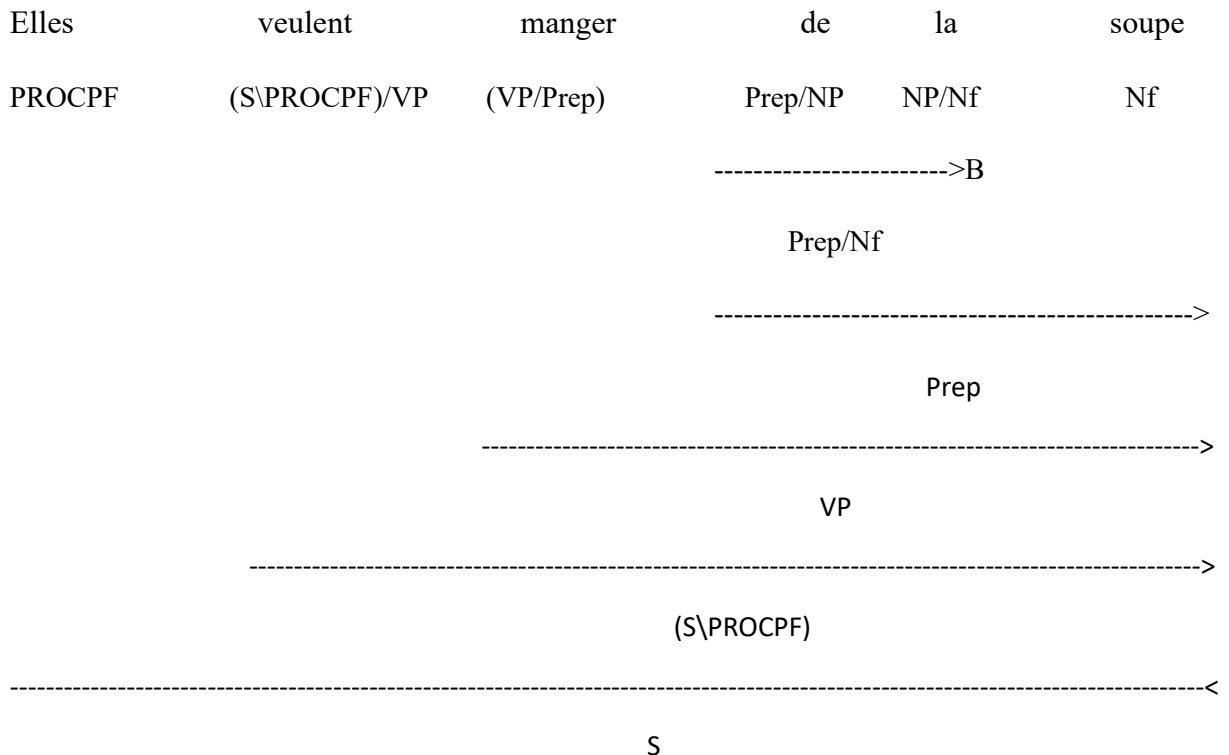
IV) Quelques exemples de phrases avec leurs dérivations syntaxiques

Dans cette partie, nous allons montrer la dérivation catégorielle de dix phrases bien construites (correctes grammaticalement) et de 20 phrases mal construites (incorrectes grammaticalement).

IV.1) Dérivation de phrases bien construites

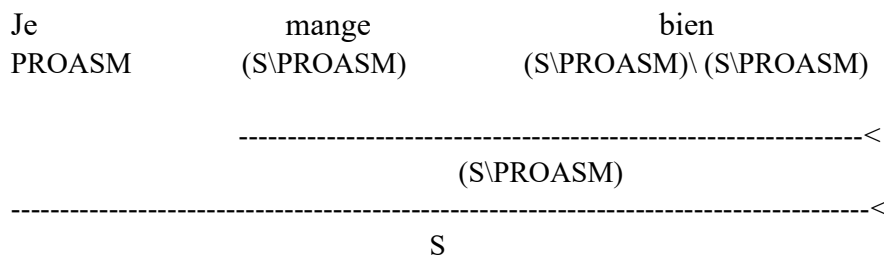
Chaque mot de chaque phrase aura une catégorie qui lui sera attribuée. Nous utiliserons les règles de la grammaire catégorielle combinatoire et applicative pour faire les dérivations catégorielles.

- Elles veulent manger de la soupe



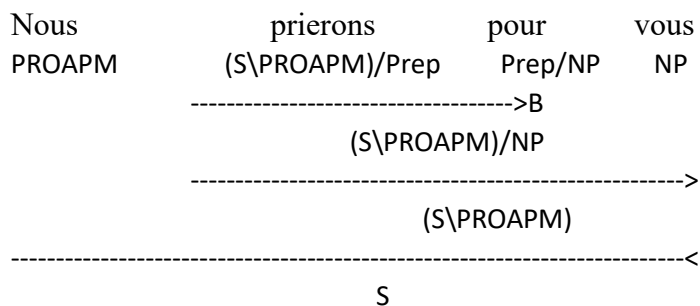
Dans cette phrase, chaque mot est correctement catégorisé et combiné. Le pronom sujet « Elles » est de catégorie PROCPF, ce qui correspond au sujet féminin pluriel. Le verbe conjugué « veulent » accepte une phrase verbale (VP), ce qu'il reçoit via le groupe « manger de la soupe ». L'article défini « la » est combiné à un nom féminin « soupe », formant un syntagme nominal (NP) attendu par la préposition « de ». Ainsi, aucune discordance grammaticale n'entrave la dérivation complète vers la phrase S. La phrase est donc correctement construite selon la CCG.

➤ Je mange bien



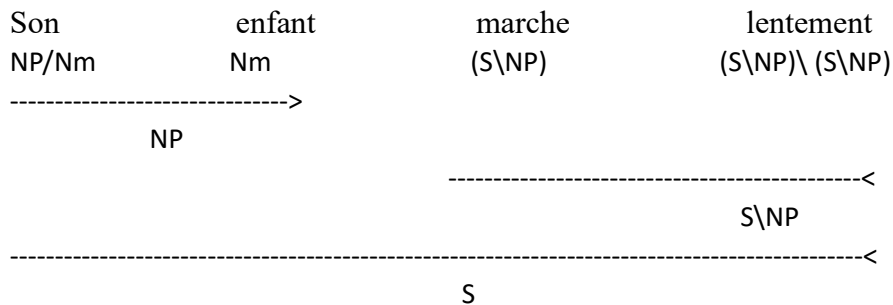
La dérivation aboutit correctement à une phrase complète S, sans conflit syntaxique ni problème de combinaison. L'adverbe « bien » est correctement utilisé comme modificateur de verbe, et le sujet « Je » est compatible avec le verbe conjugué « mange ». La structure est simple, claire, et syntaxiquement valide selon la grammaire catégorielle combinatoire.

➤ Nous prions pour vous



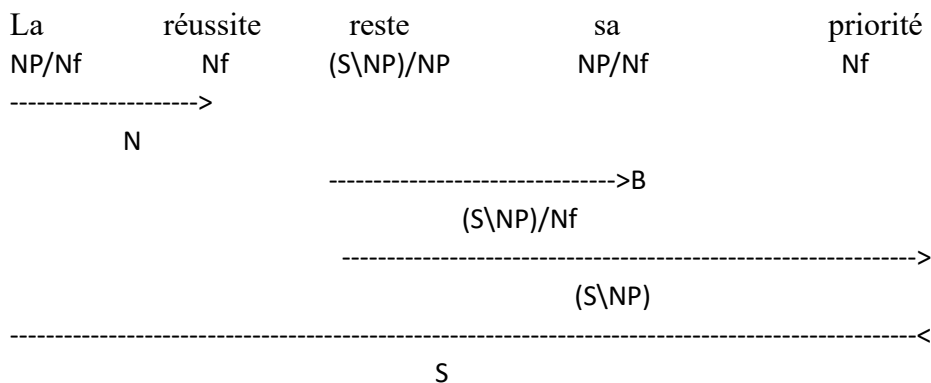
La phrase « nous prions pour vous » est syntaxiquement valide. Il n'y a aucune discordance entre les catégories. Le verbe prier attend un complément prépositionnel (Prep) et un sujet (PROAPM :1^{ère} personne pluriel), qu'il reçoit correctement. La préposition pour combine bien avec vous (qui est considéré comme un nom). La dérivation aboutit à S, conforme à la grammaire catégorielle combinatoire.

➤ Son enfant marche lentement



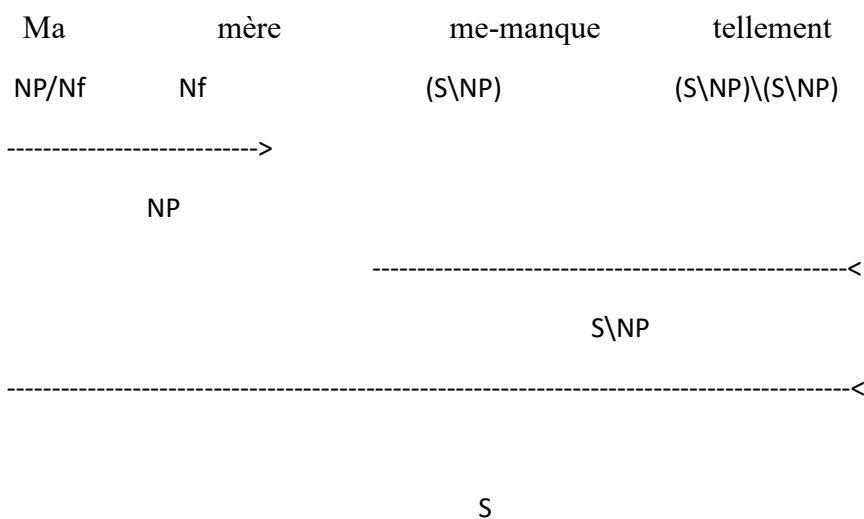
Dans la phrase « Son enfant marche lentement », chaque élément lexical se voit attribuer une catégorie conforme à la grammaire catégorielle combinatoire. Le déterminant possessif « Son » est de catégorie NP/Nm, ce qui signifie qu’il attend un nom masculin pour former un syntagme nominal. Il se combine correctement avec le nom masculin « enfant », aboutissant ainsi à un syntagme nominal (NP). Le verbe conjugué « marche », de catégorie (S\NP), attend quant à lui un syntagme nominal en sujet à gauche. L’adverbe « lentement », catégorisé par (S\NP)\(S\NP), modifie ce verbe en conservant la même catégorie verbale modifiée. La combinaison de ces catégories selon les règles de la grammaire aboutit à la formation d’une phrase complète S. Cette dérivation ne présente aucune discordance, notamment en ce qui concerne l’accord de genre entre « Son » et « enfant », ni dans la modification verbale par l’adverbe. Par conséquent, la phrase est syntaxiquement valide selon la grammaire catégorielle combinatoire.

➤ La réussite reste sa priorité



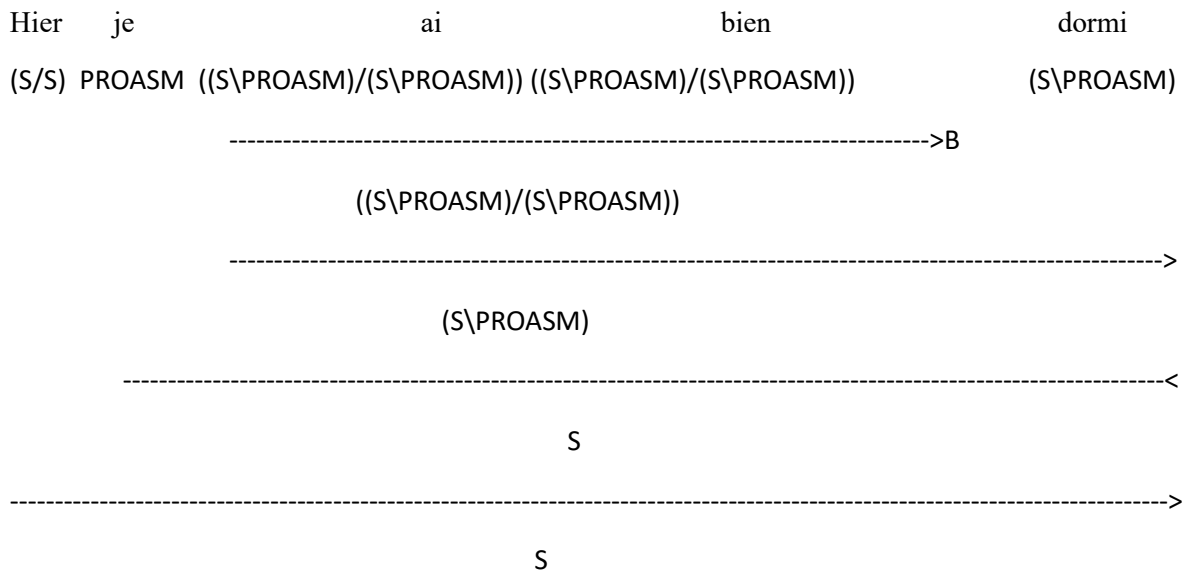
Dans la phrase « La réussite reste sa priorité », chaque mot reçoit une catégorie conforme à la grammaire catégorielle combinatoire. Le déterminant « La » (NP/Nf) se combine avec le nom féminin « réussite » (Nf) pour former un syntagme nominal (NP). De même, le déterminant possessif « sa » (NP/Nf) se combine avec le nom féminin « priorité » (Nf), produisant un autre syntagme nominal (NP). Le verbe « reste », de catégorie (S\NP)/NP, prend en complément le syntagme nominal « sa priorité » (NP) pour former un groupe verbal (S\NP). Enfin, le sujet « La réussite » (NP) se combine avec ce groupe verbal, formant une phrase complète (S). Cette dérivation s’effectue sans aucune incompatibilité grammaticale, ce qui montre que la phrase est syntaxiquement correcte selon la CCG.

➤ Ma mère me-manque tellement



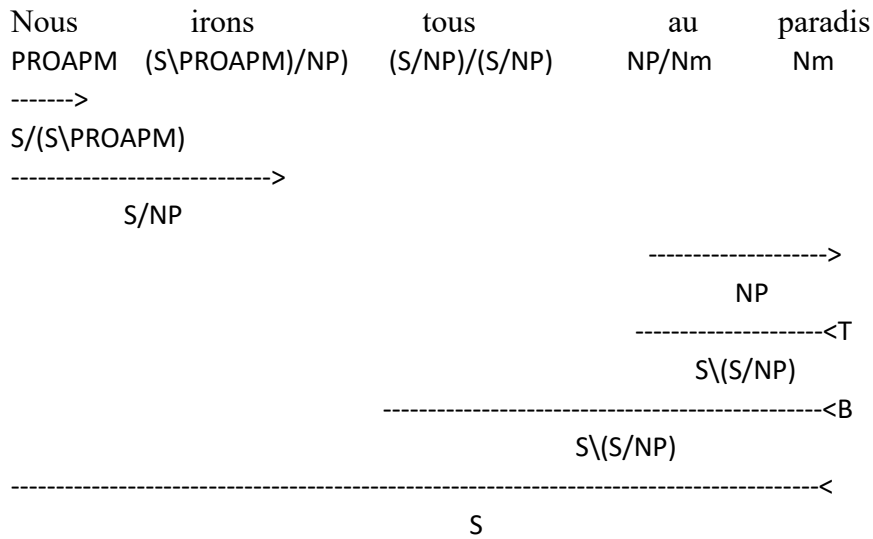
Dans la phrase « Ma mère me-manque tellement », chaque mot est assigné à une catégorie syntaxique conforme à la grammaire catégorielle combinatoire. Le déterminant possessif « Ma » (NP/Nf) se combine avec le nom féminin « mère » (Nf) pour former un syntagme nominal (NP). Le verbe « me-manque » est de catégorie (S\NP), c’est-à-dire qu’il attend un syntagme nominal sujet à gauche. L’adverbe « tellement », de catégorie (S\NP)\(S\NP), modifie ce verbe en conservant la même catégorie verbale modifiée. La combinaison de ces catégories, via les règles de la grammaire, permet de former une phrase complète (S). Cette dérivation est cohérente, notamment en ce qui concerne l’accord de genre entre « Ma » et « mère », ainsi que la fonction modifiante de l’adverbe. Par conséquent, la phrase est syntaxiquement valide selon la grammaire catégorielle combinatoire.

➤ Hier je ai bien dormi



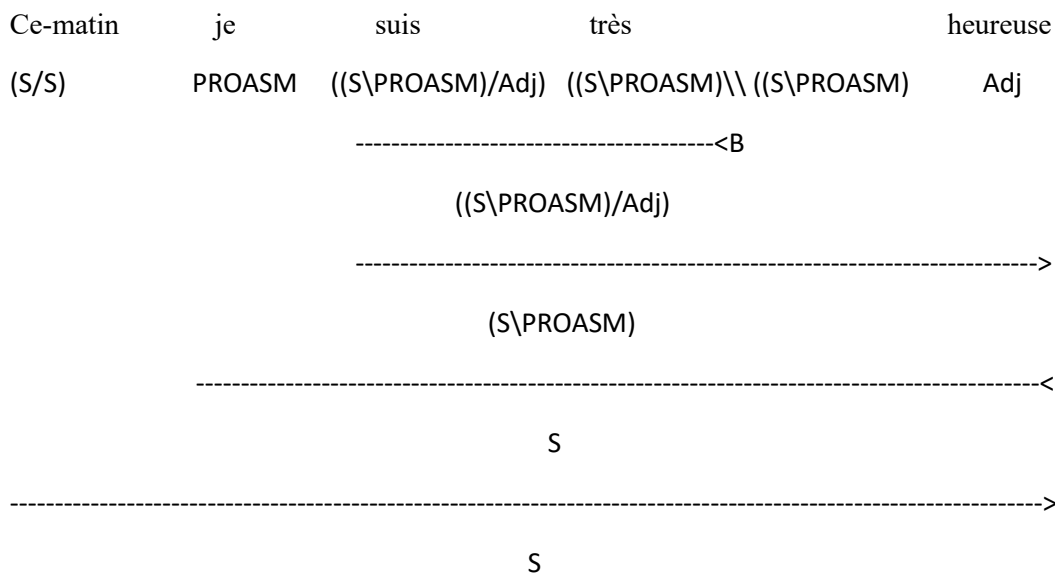
Dans la phrase « Hier je ai bien dormi », chaque élément lexical est associé à une catégorie syntaxique conforme à la grammaire catégorielle combinatoire. L’adverbe de temps « Hier » est de catégorie (S/S), ce qui signifie qu’il modifie une phrase complète en en produisant une autre. Le pronom sujet « je » appartient à la catégorie PROASM. L’auxiliaire « ai » est représenté par la catégorie ((S\PROASM)/(S\PROASM)), indiquant qu’il attend un verbe conjugué pour former un groupe verbal conjugué. De même, l’adverbe « bien » modifie le groupe verbal, possédant la même catégorie ((S\PROASM)/(S\PROASM)). Enfin, le verbe principal « dormi » est de catégorie (S\PROASM). La combinaison successive des catégories selon les règles d’application et composition aboutit à une phrase complète de catégorie S. La dérivation est donc cohérente, ce qui confirme que la phrase est syntaxiquement correcte selon la grammaire catégorielle combinatoire

➤ Nous irons tous au paradis



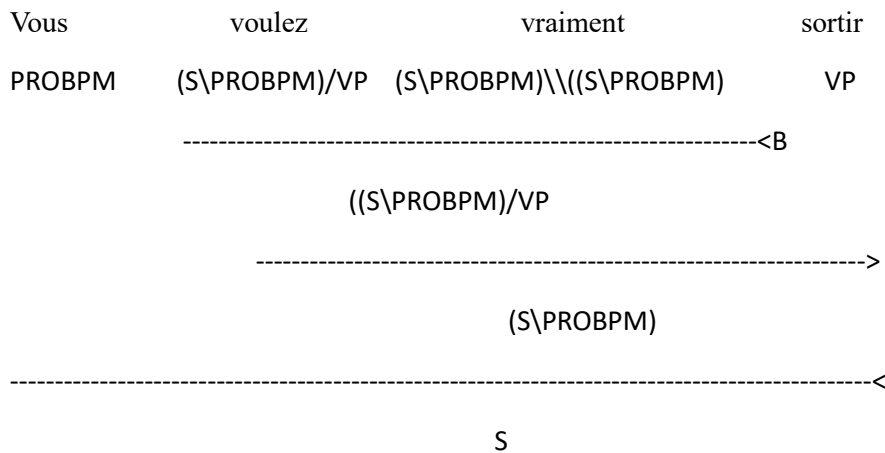
Dans la phrase « Nous irons tous au paradis », chaque mot reçoit une catégorie syntaxique selon la grammaire catégorielle combinatoire. Le pronom « Nous » (PROAPM) se combine avec le verbe conjugué « irons » ((S\PROAPM)/NP), qui attend un complément nominal. L’adverbe « tous » modifie le verbe sans changer sa catégorie. Le groupe nominal « au paradis » est formé par la combinaison du déterminant et du nom. La combinaison successive de ces éléments respecte les règles de la grammaire et aboutit à une phrase complète, valide syntaxiquement selon la CCG.

➤ Ce-matin je suis très heureuse



Dans la phrase « Ce matin je suis très heureuse », chaque élément lexical est associé à une catégorie syntaxique conforme à la CCG. L'expression temporelle « Ce matin » est de catégorie (S/S), ce qui signifie qu'elle modifie une phrase complète. Le pronom sujet « je » est PROASM. Le verbe « suis » attend un adjectif en complément, sa catégorie est ((S\PROASM)/Adj). L'adverbe « très », catégorie ((S\PROASM)\(S\PROASM)), modifie l'adjectif « heureuse » (Adj). La combinaison successive de ces catégories aboutit à une phrase complète S. Cette dérivation est cohérente et valide syntaxiquement selon la grammaire catégorielle combinatoire.

➤ Vous voulez vraiment sortir



Dans la phrase « Vous voulez vraiment sortir », chaque élément reçoit une catégorie syntaxique conforme à la CCG. Le pronom sujet « Vous » est de catégorie PROBPM. Le verbe conjugué « voulez » attend un groupe verbal (VP) à droite, sa catégorie est (S\PROBPM)/VP. L'adverbe « vraiment », de catégorie (S\PROBPM)\(S\PROBPM), modifie ce groupe verbal. Enfin, « sortir » est un verbe à l'infinitif de catégorie VP. Par composition et application, ces catégories se combinent correctement pour former une phrase complète (S). La dérivation est donc syntaxiquement valide selon la grammaire catégorielle combinatoire.

IV.2) Dérivation de phrases mal construites

IV.2.1) Problème liés au genre

- Vous pouvez manger de le soupe

Vous	pouvez	manger	de	le	soupe
PROBPM	(S\PROBPM)/VP	VP/Prep	Prep/NP	NP/Nm	Nf
				----->B	
				Prep/Nf	
				----->	
				Invalid	

Dans cette phrase, nous remarquons un problème de genre, car l'article « le » est censé être combiné avec un nom masculin singulier, mais il est incorrectement associé au nom féminin singulier « soupe ». Cette discordance entre les catégories grammaticales (l'article défini et le nom) engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une combinaison correcte des genres pour que la phrase soit valide.

- Mon famille a honte de moi

Mon	famille	a	honte	de	moi
NP/Nm	Nf	(S\NP)/ADV	(ADV/Prep)	Prep/NP	NP
	----->				
	Invalid				

Dans cette phrase, nous remarquons un problème de genre. En effet, l'article « mon » est censé être combiné avec un nom masculin, mais il est incorrectement associé au nom féminin « famille ». Cette discordance entre les catégories grammaticales (l'article défini et le nom) engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une combinaison correcte des genres pour que la phrase soit valide.

- Je pense avoir la paludisme

Je	pense	avoir	la	paludisme
PROASM	(S\PROASM)/VP	VP/NP	NPf/Nf	Nm
				----->
				Invalid

Dans cette phrase, nous remarquons un problème de genre. En effet, l'article « la » est censé être combiné avec un nom féminin, mais il est incorrectement associé au nom masculin « paludisme ». Cette discordance entre les catégories grammaticales (l'article défini et le nom) engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une combinaison correcte des genres pour que la phrase soit valide.

➤ Mon fils est tout ma vie

Mon	fils	est	tout	ma	vie
NP/Nm	Nm	(S\NP)/NP	NP/NPm	NPf/Nf	Nf
			----->B		
Invalid					

Dans cette phrase, nous remarquons un problème de genre. En effet, l'article « tout » est censé être combiné avec un article masculin, mais il est incorrectement associé à l'article possessif « ma ». Cette discordance entre les catégories grammaticales engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une combinaison correcte des genres pour que la phrase soit valide.

➤ La monsieur est en vie

La	monsieur	est	en	vie
NP/Nf	Nm	(S\NP)/NP	NP/Nf	Nf
				----->
Invalid				

Dans cette phrase, nous remarquons un problème de genre. En effet, l'article « la » est censé être combiné avec un nom féminin, mais il est incorrectement associé au nom masculin « monsieur ». Cette discordance entre les catégories grammaticales (l'article défini et le nom) engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une combinaison correcte des genres pour que la phrase soit valide.

➤ Le chance nous-sourit

Le	chance	nous-sourit
NP/Nm	Nf	S\NP
		----->
Invalid		

Dans cette phrase, nous remarquons un problème de genre. En effet, l'article « le » est censé être combiné avec un nom masculin, mais il est incorrectement associé au nom féminin « chance ». Cette discordance entre les catégories grammaticales (l'article défini et le nom) engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une combinaison correcte des genres pour que la phrase soit valide.

➤ Je dois assister ma frère

Je	dois	assister	ma	frère
PROASM	(S\PROASM)/VP	VP/NP	NP/Nf	Nm
				----->
				Invalid

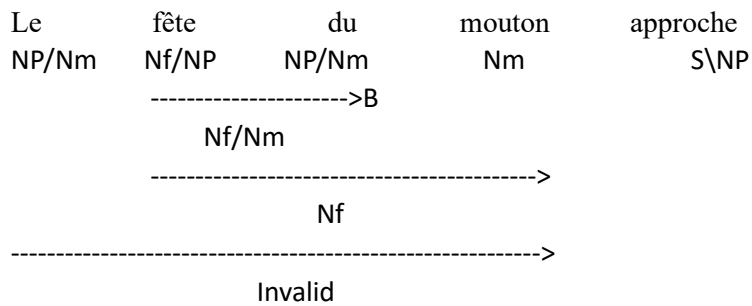
Dans cette phrase, nous remarquons un problème de genre. En effet, l'article « ma » est censé être combiné avec un nom féminin, mais il est incorrectement associé au nom masculin « frère ». Cette discordance entre les catégories grammaticales (l'article défini et le nom) engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une combinaison correcte des genres pour que la phrase soit valide.

➤ Son fille est fière d'elle

Son	fille	est	fière	d'elle
NP/Nm	Nf	(S\NP)/Adj	Adj/NP	NP
				----->
				Invalid

Dans cette phrase, nous remarquons un problème de genre. En effet, l'article « son » est généralement combiné avec un nom masculin, mais il est associé au nom féminin « fille ». Cette discordance entre les catégories grammaticales (l'article défini et le nom) engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une combinaison correcte des genres pour que la phrase soit valide.

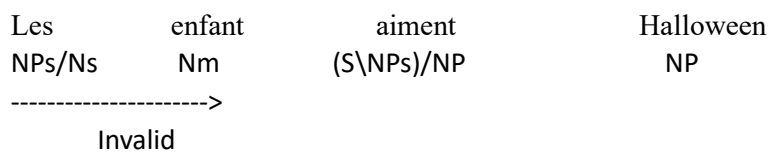
➤ Le fête du mouton approche



Dans cette phrase, nous remarquons un problème de genre. En effet, l'article « le » est censé être combiné avec un nom masculin, mais il est incorrectement associé au nom féminin « fête ». Cette discordance entre les catégories grammaticales (l'article défini et le nom) engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une combinaison correcte des genres pour que la phrase soit valide.

IV.2.2) Problème liés au nombre

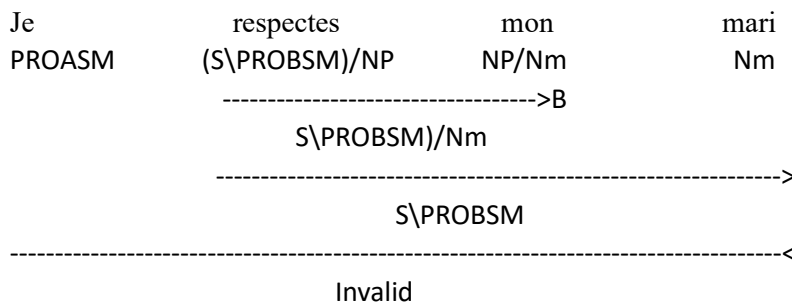
➤ Les enfant aiment Halloween



Pour cette phrase, nous observons un problème d'accord en nombre. En effet, l'article défini « les » est censé être combiné avec un nom au pluriel, mais il est incorrectement associé au nom singulier « enfant ». Cette discordance entre l'article et le nom engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une concordance correcte entre le nombre de l'article et celui du nom pour garantir la validité de la phrase.

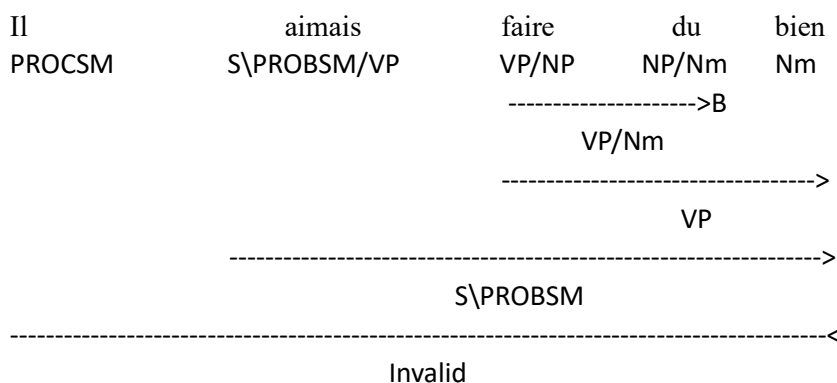
IV.2.3) Problème liés à la conjugaison

➤ Je respectes mon mari



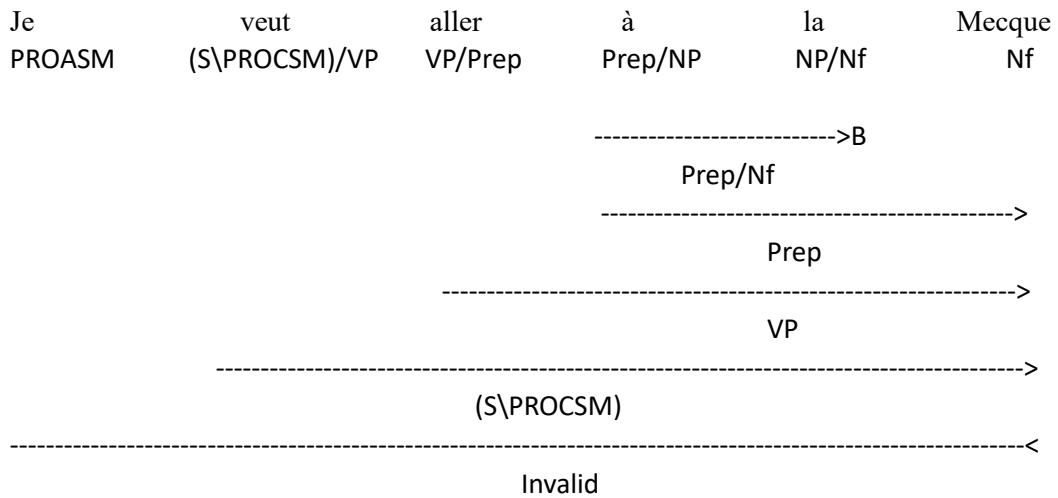
Dans cette phrase, nous rencontrons un problème de conjugaison du verbe. En effet, le verbe « respectes » est conjugué à la deuxième personne du singulier, alors que le sujet « Je » exige une forme verbale à la première personne du singulier, soit « respecte ». Cette incohérence dans la conjugaison engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une correspondance correcte entre le sujet et le verbe pour que la phrase soit valide.

➤ Il aimais faire du bien



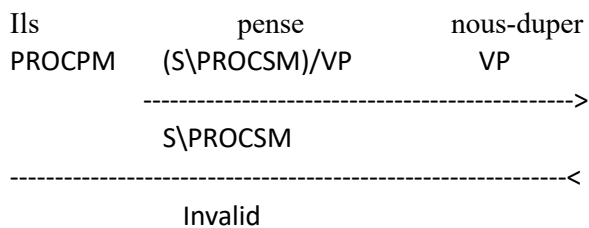
Dans cette phrase, nous rencontrons un problème de conjugaison du verbe. En effet, le verbe « aimais » est conjugué à la deuxième personne du singulier, alors que le sujet « Il » exige une forme verbale à la troisième personne du singulier, soit « aimait ». Cette incohérence dans la conjugaison engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une correspondance correcte entre le sujet et le verbe pour que la phrase soit valide.

➤ Je veut aller à la Mecque



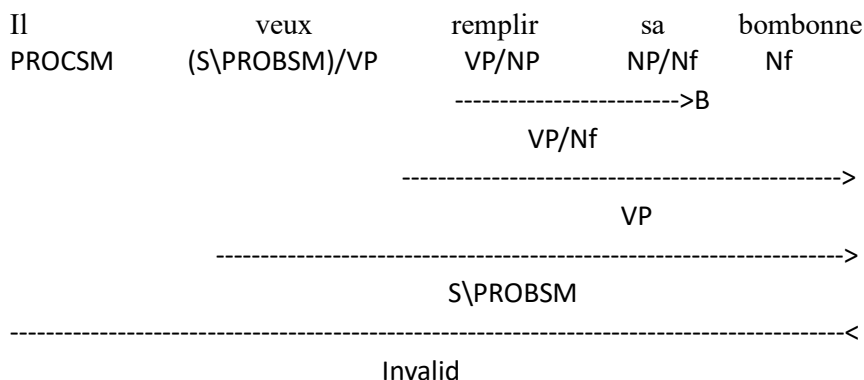
Dans cette phrase, nous rencontrons un problème de conjugaison du verbe. En effet, le verbe « veut » est conjugué à la troisième personne du singulier, alors que le sujet « Je » exige une forme verbale à la première personne du singulier, soit « veux ». Cette incohérence dans la conjugaison engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une correspondance correcte entre le sujet et le verbe pour que la phrase soit valide.

➤ Ils pense nous duper



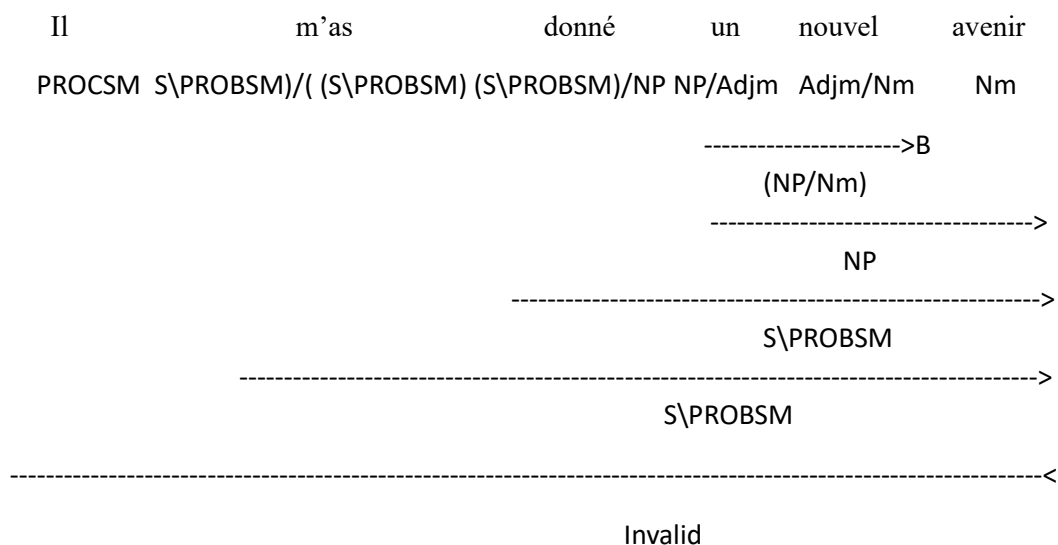
Dans cette phrase, nous rencontrons un problème d'accord en nombre entre le sujet et le verbe. En effet, le verbe « pense » est conjugué à la troisième personne du singulier, alors que le sujet « Ils » exige une forme verbale à la troisième personne du pluriel, soit « pensent ». Cette incohérence dans la conjugaison engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une correspondance correcte entre le sujet et le verbe pour que la phrase soit valide.

➤ Il veut remplir sa bombonne



Dans cette phrase, nous rencontrons un problème de conjugaison du verbe. En effet, le verbe « veut » est conjugué à la deuxième personne du singulier, alors que le sujet « Il » exige une forme verbale à la troisième personne du singulier, soit « veut ». Cette incohérence dans la conjugaison engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une correspondance correcte entre le sujet et le verbe pour que la phrase soit valide.

➤ Il m'as donné un nouvel avenir



Dans cette phrase, nous rencontrons un problème de conjugaison du verbe. En effet, le verbe « m'as » est conjugué à la deuxième personne du singulier, alors que le sujet « Il » exige une forme verbale à la troisième personne du singulier, soit « m'a ». Cette incohérence dans la conjugaison engendre une invalidité syntaxique, car la grammaire catégorielle combinatoire attend une correspondance correcte entre le sujet et le verbe pour que la phrase soit valide.

IV.2.4) Problème liés à la combinaison des catégories syntaxiques

- On sang tellement bon

On	sang	tellement	bon
NP	Nm	((S\NP)\(S\NP))	Adjm

Dans cette phrase, nous rencontrons un problème de combinaison des catégories syntaxiques. Le nom « sang » (catégorie Nm) est associé à une structure syntaxique qui attend un verbe ou un adjectif, mais il est incorrectement combiné avec des éléments qui ne correspondent pas à sa fonction. Le groupe « sang tellement bon » tente d'appliquer une structure de type ((S\NP)(S\NP))/Adjm sur le nom « sang », ce qui est une combinaison invalide dans le cadre de la grammaire catégorielle combinatoire applicative. L'erreur réside dans le fait que la combinaison de ces éléments crée une incompatibilité entre les catégories grammaticales, rendant la phrase syntaxiquement invalide.

- Vous avez peurs pour votre vie

Vous	avez	peurs	pour	sa	vie
PROBPM	(S\PROBPM)/ADV	NPs	(S\S)/NP)	NP/Nf	Nf
----->					
Invalid					

L'erreur réside dans le fait que la combinaison de ces éléments crée une incompatibilité entre les catégories grammaticales, rendant la phrase syntaxiquement invalide. En effet, « avoir » suivi de « peur » (au singulier) est une expression figée (« avoir peur »), mais ici, « peurs » (au pluriel) est utilisé, ce qui empêche la combinaison correcte des catégories syntaxiques. Cette discordance bloque la dérivation syntaxique, rendant la phrase mal formée selon la grammaire catégorielle.

IV.2.5) Problème liés au lexique

- Il ne sera jamais malheureu

Il	ne-sera	jamais	malheureu
PROCSM	(S\PROCSM)	((S\PROCSM)\(S\PROCSM))/Adj	Unknow

- Je ai reçu beaucoup de cadeaux

Je	ai	reçu	beaucoup
----	----	------	----------

PROASM ((S\PROASM)/(S\PROASM))/Prep Unknow (S\PROASM)\(S\PROASM)

de cadeaux
Prep/Ns Ns

Pour ces deux phrases, l'erreur réside dans le fait que les mots 'recu' et 'malheureu' n'existent pas dans le dictionnaire, ce qui rend l'analyse impossible, car le processus échoue systématiquement.

V) Outil utilisé pour la suggestion de correction

Dans le cadre de notre projet, nous utilisons LanguageTool pour la détection des erreurs et la génération de suggestions de correction.

V.1) Qu'est-ce-que LanguageTool ?

LanguageTool est un outil de correction grammaticale et orthographique multilingue développé en open source depuis 2003 par Daniel Naber, Jaume Ortolà et Christopher Blum. Leur ambition était de proposer une solution accessible et performante, capable d'assister aussi bien les professionnels que les non-initiés dans la rédaction de textes sans fautes (Tautou, 2025).

LanguageTool est aujourd'hui largement utilisé dans le domaine du traitement automatique des langues (TAL). Il offre une analyse grammaticale fine des textes, s'appuyant à la fois sur :

- des règles linguistiques formelles (syntaxiques et morphologiques),
- des dictionnaires lexicaux,
- et, dans ses versions les plus récentes, sur des techniques d'intelligence artificielle (notamment pour le contexte sémantique).

Grâce à cette combinaison, LanguageTool permet d'identifier une grande variété d'erreurs : fautes d'orthographe, erreurs grammaticales (accords, conjugaison...), maladresses de style, erreurs de ponctuation ou redondances, incohérences dans les données numériques (dates, formats...) et propose des suggestions adaptées au contexte linguistique.

Compatible avec plus de 30 langues, il est particulièrement performant pour les principales langues d'usage telles que le français, l'anglais, l'allemand, l'espagnol, le néerlandais et le portugais.

Parmi ses fonctionnalités avancées, on peut citer :

- la correction grammaticale et orthographique en temps réel,
- des suggestions de reformulation stylistique,
- la recherche de synonymes,
- la détection des tournures maladroites ou répétitives,
- la sélection d'un registre de langue (soutenu, familier, académique, etc.),
- des outils d'analyse statistique de rédaction,
- un dictionnaire personnel et un guide de style personnalisable,
- un générateur de reformulations (plus formelles, plus simples, plus concises, etc.).

En résumé, LanguageTool constitue un outil précieux pour améliorer la qualité rédactionnelle, en particulier dans les contextes professionnels ou académiques, où la rigueur linguistique est essentielle.

V.2) Fonctionnement de languageTool

Lorsque LanguageTool reçoit une phrase, il procède à plusieurs étapes successives :

V.2.1) Analyse linguistique de la phrase

V.2.1.a) La Tokenisation

La phrase reçue est d'abord segmentée en mots (tokens). Prenons l'exemple de phrase : « ils sont intelligents » qui devient [« ils », « sont », « intelligents »]

V.2.1.b) L'Analyse morphologique

Chaque token est analysé pour déterminer sa catégorie grammaticale et ses caractéristiques morphologiques. Prenons l'exemple précédent : [sont : verbe, 3^e personne pluriel]

V.2.1.c) L'Étiquetage POS (Part-of-Speech Tagging)

Chaque token est associé à un tag grammatical.

En considérant l'exemple précédent, nous aurons : « ils » → Pronom personnel, 3^e personne masculin pluriel

V.2.2) Application de règles linguistiques

LanguageTool applique ensuite des règles prédéfinies (et parfois complexes) sur la structure de la phrase telles que : l'accord (sujet/verbe, nom/adjectif), l'orthographe (erreurs lexicales), la ponctuation (virgules manquantes, majuscules), le style (phrases trop longues, formulations maladroites), les homophones ou confusions fréquentes (c'est / s'est).

V.2.3 Suggestions de correction

Lorsque LanguageTool détecte une erreur, Il précise sa position (le mot ou la portion de texte concernée), il explique sa nature (accord incorrect) et il propose une ou plusieurs suggestions de correction.

- Exemple : Vous pouvez manger un soupe
Erreur détectée sur un, position 19
Suggestion de correction : une soupe

Pour le français, LanguageTool identifie divers types d'erreurs, telles que les fautes d'orthographe, les erreurs grammaticales (accords, conjugaisons, etc.), les maladresses stylistiques et les incohérences lexicales. Dans le cadre de notre projet, LanguageTool a été intégré pour jouer un rôle complémentaire à l'analyse syntaxique basée sur la grammaire catégorielle combinatoire et applicative. Il permet, en amont ou en aval de cette analyse, de signaler les erreurs linguistiques présentes dans les phrases étudiées, et de proposer des suggestions de correction pondérées. Grâce à son interface Python, l'outil peut être facilement automatisé et appliqué à des corpus de phrases, ce qui le rend particulièrement adapté aux tâches de correction automatique et à l'assistance à l'apprentissage du français langue seconde.

VI) Poids des suggestions de corrections

Afin d'assister l'apprenant dans l'identification et la correction des erreurs syntaxiques ou grammaticales, nous avons intégré un système de suggestions basé sur l'outil LanguageTool. Lorsqu'une erreur est détectée dans une phrase, l'outil propose une liste de suggestions possibles. Pour rendre ces dernières plus lisibles et pédagogiques, nous avons introduit un système de pondération linéaire décroissante. Il permet d'attribuer un poids à chaque suggestion reflétant sa position dans la liste proposée. Ainsi, la première suggestion considérée comme la plus pertinente reçoit un poids de 100 %, tandis que les suivantes se voient attribuer un poids décroissant. Ce mécanisme permet de guider l'utilisateur vers les corrections les plus probables, tout en lui laissant la liberté d'examiner d'autres alternatives. Ces suggestions pondérées s'appliquent particulièrement aux mots inconnus ou mal utilisés dans une phrase, renforçant l'aspect didactique de notre approche et facilitant l'apprentissage par la correction contextuelle. La formule de pondération linéaire décroissante utilisée pour attribuer un poids à chaque suggestion est la suivante :

$$D = \left(\frac{N - i}{N} \right) \times 100$$

Où N est le nombre total de suggestions disponibles pour une erreur donnée, *i* est l'indice de la suggestion (en commençant à 0 pour la première suggestion) et le résultat D est exprimé en pourcentage, avec la première suggestion pondérée à 100 %.

➤ Exemple : Je pense avoir la paludisme

Tableau des erreurs détectées		
Phrase	Statut	Erreurs et Suggestions
Je pense avoir la paludisme	Phrase incorrecte	Position 15: Je pense avoir la paludisme -> le paludisme : 100.00%,
Erreurs et Suggestions		
Position 15: Je pense avoir la paludisme -> le paludisme : 100.00%, les paludismes : 50.00%		

Figure 4: Illustration poids de suggestion de correction

VII) Application du modèle

Dans notre modèle, la première étape consiste à créer un lexique qui agira comme un dictionnaire pour l'assignation des catégories grammaticales à chaque mot ou token des phrases. Pour cela, nous utiliserons la bibliothèque NLTKCCG, qui permet d'effectuer une analyse catégorielle des phrases en utilisant les règles de la grammaire catégorielle combinatoire applicative (GCCA).

Afin de simplifier notre analyse et d'éviter une segmentation trop fine de certaines unités fréquentes, nous traitons certains mots ou combinaisons de mots comme des entités indivisibles, leur attribuant une seule catégorie grammaticale. Cela est particulièrement pertinent pour les contractions ou les groupes fixes qui apparaissent fréquemment dans les phrases. Par exemple:

- ne-sera, ne-suis, ne-m'as, m'as-donné, me-manque : ces combinaisons de mots seront traitées comme des unités grammaticales et affectées à une catégorie syntaxique spécifique (par exemple, ne-suis sera attribuée à une catégorie verbale). Ces expressions seront combinées avec d'autres éléments selon les règles syntaxiques de la GCCA, facilitant ainsi l'analyse.
- j'ai et qu'il seront traités comme des entités séparés (par exemple, j'ai sera associé à « je » « ai » et qu'il en « que » « il » pour simplifier l'analyse et la détection d'erreurs).
- l' et d': ces formes contractées ne seront pas analysées comme des entités séparées mais intégrées directement dans les groupes nominaux ou les structures pronominales (par

exemple, "l'amitié" sera analysée comme un seul nom). Ce traitement garantit que les contractions ne soient pas divisées en plusieurs jetons, préservant ainsi leur fonction grammaticale dans les constructions.

- ce-matin et chaque-matin : ces combinaisons de mots seront considérés comme des unités figées ou des expressions adverbiales qui sont analysées comme des blocs dans notre lexique. . Cette approche permet d'éviter une analyse trop fine et de conserver l'intégrité de l'expression.

Nous éviterons également de séparer des éléments comme t'a ou m'a, qui seront traités comme des unités complexes (par exemple, m'a sera directement associé à une catégorie verbale pour simplifier l'analyse et la détection d'erreurs).

Cette approche permet de simplifier l'analyse des phrases tout en conservant la précision nécessaire pour détecter des erreurs syntaxiques, tout en traitant les combinaisons de mots les plus courantes de manière cohérente et efficace.

VIII) [Conclusion](#)

En conclusion, ce chapitre a posé les bases théoriques et méthodologiques nécessaires à la réalisation de notre projet. Nous avons décrit le workflow d'analyse, présenté le modèle grammatical sous-jacent, à savoir la Grammaire Catégorielle Combinatoire Applicative (GCCA), fourni des exemples de phrases correctes et incorrectes avec leurs dérivations, discuté de l'application de notre modèle, détaillé l'outil de suggestion de correction et la formule utilisée pour attribuer un poids à chaque suggestion de correction.

Ces éléments, combinés, constituent un cadre robuste pour répondre aux besoins spécifiques des apprenants de français langue seconde (Cuq, 2013). En s'appuyant sur une grammaire rigoureuse et des outils modernes, notre approche permet non seulement d'identifier les erreurs et de proposer des corrections adaptées, mais aussi d'accompagner les apprenants dans une meilleure compréhension des règles syntaxiques sous-jacentes.

Le chapitre suivant sera consacré à l'application concrète de ces concepts, en mettant en œuvre le système dans des situations pratiques, et en évaluant son efficacité à travers des exemples et des expérimentations.

Chapitre IV : Implémentation

I) Introduction

Dans ce chapitre, nous présenterons la solution informatique que nous avons développée dans le cadre de ce travail. Nous détaillerons d'abord le choix des technologies et des bibliothèques utilisées, justifiant leur pertinence au regard des objectifs visés. Nous introduirons ensuite la notation des catégories syntaxiques adoptée, ainsi que les règles de combinaison issues de la Grammaire Catégorielle Combinatoire et Applicative (GCCA) utilisées dans notre lexique. Enfin, nous décrirons le fonctionnement de la plateforme, à travers ses interfaces principales, en expliquant comment l'utilisateur peut soumettre des phrases et obtenir une analyse ou des suggestions de correction.

II) Présentation et objectifs de la solution proposée

Notre solution permettra de faire l'analyse syntaxique de phrases avec la GCCA, de détecter les erreurs dans les phrases mais aussi d'indiquer leur position et de proposer des suggestions de correction. La conception sera réalisée en langage Python sous l'éditeur de code VsCODE et nous utiliserons l'analyseur syntaxique nltk. Il est à noter que nous avons également conçu notre propre corpus pour les besoins de ce projet. Afin de garantir la fiabilité et la précision de notre solution, une évaluation des performances sera effectuée.

III) Le langage de programmation utilisé : Python

Python, créé par Guido van Rossum et sorti en 1991, est un langage orienté objet populaire. Il est connu pour être un langage de programmation généraliste, de haut niveau, très clair et très puissant. Il est généralement utilisé pour le développement web (côté serveur), le développement de logiciels, les mathématiques et les scripts système. Il est souvent décrit comme un langage « piles incluses » en raison de sa bibliothèque standard complète. Python se classe régulièrement parmi les langages de programmation les plus populaires et est largement utilisé dans la communauté de l'apprentissage automatique (Summerfield, 2009).

C'est un langage qui présente plusieurs avantages:

- Python est compatible avec plusieurs plateformes, y compris Windows, Mac, Linux et Raspberry Pi.
- La syntaxe de Python est simple.

- Grâce à sa syntaxe claire, Python permet aux développeurs de créer des programmes en utilisant moins de lignes de code par rapport à d'autres langages de programmation.
- Python utilise un interpréteur, permettant l'exécution immédiate du code, ce qui facilite un prototypage rapide.
- Python peut être utilisé de manière procédurale, orientée objet ou fonctionnelle

Cependant, Nous ne nous attarderons pas sur la présentation du langage python car nombreux sont les ouvrages disponibles pour toutes personnes désirant approfondir ses connaissances en python ((Lutz, 2013) ; (Ramalho, 2022) ; (Downey, 2024)). Dans la section suivante, nous aborderons l'analyseur syntaxique nltk. En effet, nous expliquerons pourquoi nous avons choisi cet analyseur syntaxique vu qu'il en existe beaucoup et, en même temps, justifierons notre choix du langage de programmation Python.

IV) L'analyseur syntaxique choisi

Il existe plusieurs analyseurs syntaxiques pour la grammaire catégorielle combinatoire CCG mais nous allons en présenter trois : C&C CCG Parser, OPenCCG et nltkccg (que nous utiliserons dans notre projet).

IV.1) L'analyseur syntaxique C&C CCG Parser

C&C CCG Parser a été développé par James Curran et Stephen Clark. C'est un analyseur syntaxique à large couverture qui peut traiter efficacement des tâches NLP à grande échelle et qui est implémenté en C++. L'analyseur syntaxique est optimisé par plusieurs techniques. C&C tools offre également un outil de sémantique computationnelle nommé Boxer, qui a été développé par Johan Bros. Boxer prend une analyse syntaxique CCG comme entrée et génère une représentation de discours comme sortie (Clark & Curran, 2007).

IV.2) L'analyseur syntaxique OpenCCg

L'analyseur syntaxique OpenCCG est implémenté en Java et est destiné à être utilisé dans des tâches réelles de TAL. L'analyseur syntaxique applique les extensions multi-modales à CCG ((Baldrige & Kruijff, 2002), (Steedman & Baldrige, 2006)). L'extension multi-modale met les modalités sur la directionnalité et spécifie si elle peut être appliquée ou non avec les règles de composition ou de substitution. L'analyseur syntaxique est très pratique car il a été utilisé dans plusieurs autres projets. L'analyseur OpenCCG propose également VisCCG, qui est un logiciel

de visualisation permettant de créer et de modifier une grammaire écrite au format DotCCG, qui est le format de la grammaire CCG utilisé par l'analyseur OpenCCG (SourceForge.net, 2015).

IV.3) L'analyseur syntaxique NLTK

NLTKCCG est implémenté comme un module de Natural Language Toolkit (Bird, Klein, & Loper, 2009). NLTK est un logiciel libre 3, écrit en Python, qui prend en charge diverses tâches NLP : Balisage POS, analyse syntaxique, interprétation sémantique, balisage de mots, etc. NLTK est largement adopté et a été utilisé comme outil d'enseignement par plus de cinquante universités. En raison de la prééminence de CCG et de l'omniprésence de l'utilisation de NLTK, il est souhaitable que NLTK dispose d'un analyseur syntaxique CCG. L'analyseur syntaxique CCG dans `nlk.ccg` a été écrit pour la première fois par Graeme Gange. C'est un analyseur graphique CYK qui n'est ni basé sur les caractéristiques ni probabiliste. Il analyse simplement une phrase avec des catégories. La dérivation sémantique n'est pas non plus supportée. Les règles combinatoires de (Steedman & Baldridge, 2007) sont supportées. L'analyseur syntaxique de Gange est intégré à la technique de la carte emballée, bien qu'elle ne fonctionne pas de manière efficace. Ceci est dû au fait qu'il a été décidé d'empaqueter les arêtes que chacun d'entre eux possède :

- Catégorie identique
- Portée identique
- Règle combinatoire identique qui est utilisée pour se former elle-même. En fait, si la troisième condition était abandonnée, un plus grand nombre d'arêtes serait emballé, et l'analyseur fonctionnerait plus efficacement. Comme l'analyseur syntaxique est censé être un outil d'exploration pour la GCC, il y a 3 exigences cruciales :
 - Les utilisateurs devraient pouvoir définir leur propre lexique dans lequel une catégorie peut avoir des caractéristiques et un mot est associé à une expression sémantique.
 - Les utilisateurs doivent être en mesure d'examiner chaque étape de l'analyse syntaxique.
 - L'analyseur syntaxique doit être facilement compréhensible et extensible (Bird, Klein, & Loper, 2024).

IV.4) Pourquoi le choix de NLTK ?

Dans cette section, nous établirons un tableau comparatif de nos trois analyseurs afin d'expliquer pourquoi nous avons choisi NLTK comme analyseur syntaxique dans notre projet.

CRITERES	ANALYSEURS		
	C&C CCG Parser	OPENCCG	NLTK
Flexibilité et personnalisation	+	+++	+++
Documentation et communauté	++	+	++++++
Performances	++++++	++	++++
Facilité d'apprentissage	-	-	++++
Transparence du processus d'analyse	-	-	++++

Tableau 1: Tableau comparatif de CCg Parser, Openccg et NLTKCCg

En analysant ce tableau, nous avons choisi NLTKCCG comme analyseur syntaxique pour notre projet, car il se distingue par une grande transparence dans les étapes d'analyse. Cette visibilité facilite le débogage, l'optimisation et l'apprentissage, des éléments essentiels pour un projet où la clarté du processus d'analyse syntaxique est primordiale. C&C Parser et OpenCCG, bien que performants dans leurs domaines respectifs, peuvent offrir une visibilité moins détaillée sur le processus d'analyse, ce qui peut être une considération importante.

V) Les bibliothèques utilisées

V.1) NLTKCCG

La Bibliothèque nltk.ccg (Natural Language Toolkit - CCG) est une partie de NLTK dédiée à l'analyse syntaxique basée sur la grammaire catégorielle combinatoire (CCG). Chart contient des classes et des méthodes pour analyser syntaxiquement des phrases en utilisant les règles CCG spécifiées.

V.2) Csv (Comma Separated Values)

Cette bibliothèque est incluse par défaut avec Python et n'a pas besoin d'être installée séparément. Elle est utilisée pour lire et écrire des fichiers CSV.

V.3) Pandas

Cette bibliothèque est utilisée pour manipuler et analyser des données, notamment pour écrire les résultats dans un fichier CSV.

V.4) Sklearn.metrics

La bibliothèque sklearn.metrics est utilisée pour générer un rapport de classification qui évalue la performance de notre modèle en comparant les statuts attendus des phrases avec les statuts réels détectés par notre système (Dregosa, et al., 2011). Dans notre projet, son utilisation est indispensable, car elle nous permet de produire un rapport détaillé pour mesurer l'efficacité de la détection des phrases correctes et incorrectes.

VI) La représentation des données

La grammaire catégorielle combinatoire applicative est une approche formelle en linguistique et en traitement du langage naturel qui combine des principes de la grammaire catégorielle avec des règles de combinaison spécifiques pour analyser les structures syntaxiques des phrases. En effet, elle permet de décrire un langage en attribuant des types logiques aux unités lexicales.

Dans le cadre de notre analyse syntaxique, la notation des catégories et des règles de combinaison est essentielle pour plusieurs raisons :

- Structuration et Classification : La notation permet de structurer et de classer les différents types de mots et de phrases selon leur rôle syntaxique et

sémantique. Elle aide à comprendre comment les différents éléments d'une phrase interagissent et se combinent.

- Définition des Combinaisons : Elle définit comment les catégories syntaxiques peuvent être combinées pour former des structures plus complexes. Ces notations offrent un cadre formel pour assembler les éléments linguistiques de manière cohérente et systématique.
- Automatisation : Les notations et les règles facilitent l'analyse syntaxique automatique, comme celle réalisée par des systèmes basés sur NLTKccg. Les règles de combinaison, telles que l'application fonctionnelle et la composition, sont cruciales pour automatiser la construction des structures syntaxiques à partir des catégories de base.
- Précision : Une notation précise et des règles de combinaison bien définies permettent aux outils d'analyse de générer des résultats exacts et fiables. Elles diminuent les ambiguïtés en spécifiant clairement comment les différents éléments doivent être assemblés.
- Flexibilité : Les règles de combinaison permettent de traiter une grande variété de structures syntaxiques en établissant des mécanismes généraux pour assembler les catégories. Elles permettent ainsi l'analyse non seulement des phrases simples, mais aussi des constructions plus complexes.

VI.1) La notation des Catégories

Pour les besoins de notre projet, nous utilisons des notations spécifiques pour représenter nos différentes catégories syntaxiques : les catégories de base et les catégories fonctionnelles :

VI.1.1) Catégories de Base

Pour les besoins de notre projet, les catégories de base sont représentées comme suit :

- NP (Groupe Nominal) : Un groupe de mots, généralement constitué d'un nom et de ses déterminants ou adjectifs, qui peut jouer le rôle de sujet, complément d'objet, etc.
Exemple : le livre, une grande maison.
- Nf (Nom féminin) : Cela représente un nom qui désigne une chose ou une personne de genre féminin.

Exemples : maison, femme

- Nm (Nom masculin) : Cela représente un nom qui désigne une chose ou une personne de genre masculin.

Exemples : homme, chien

- Ns (Nom pluriel) : Cela représente un nom au pluriel qui peut être masculin ou féminin.

- NP/Ns (Déterminant possessif/pluriel) : Cela désigne un groupe nominal avec un déterminant possessif au pluriel

Exemples : mes, tes, leurs

- NP/NPf (Déterminant possessif) : Cela désigne un groupe nominal avec un déterminant possessif mais qui doit être suivi d'un nom féminin.

Exemple : ma

- NP/NPm (Déterminant possessif) : Cela désigne un groupe nominal avec un déterminant possessif mais qui doit être suivi d'un nom masculin.

Exemple : mon

- Notons que les catégories pronominales telles que PROCSM, PROCSF, PROAPM, PROBPM, PROCPM, PROCPF, et, dans certains cas, PROASM et PROBSM, entretiennent des relations syntaxiques analogues avec les verbes dans le cadre de l'analyse en Grammaire Catégorielle Combinatoire et Applicative (GCCA).

Ces abréviations désignent des pronoms personnels sujets ou objets, classés selon la personne (1^{re}, 2^e, 3^e), le genre (masculin/féminin), et le nombre (singulier/pluriel).

Par exemple : PROASM correspond au pronom de 1^{re} personne du singulier (masculin ou féminin), PROBSM correspond au pronom de 2^e personne du singulier (masculin ou féminin), PROCSM correspond au pronom de 3^e personne du singulier masculin « il », tandis que PROCSF correspond au pronom de 3^e personne du singulier féminin à « elle ». Dans le cadre de la GCCA, tous ces pronoms partagent généralement la même catégorie syntaxique NP (groupe nominal), ce qui leur permet de se combiner selon le nombre et la personne avec les verbes selon les règles standards d'application fonctionnelle (avant > ou arrière <), de composition et de changement de type. Par exemple, dans les phrases « nous mangeons » et « vous mangez », que « nous : PROAPM : 1^{re} personne du pluriel (masculin ou féminin) » désigne un groupe masculin ou

féminin ou que « vous : PROBPM : 2^e personne du pluriel (masculin ou féminin) » désigne un groupe masculin ou féminin, cela n'affecte ni la catégorie syntaxique (toujours NP), ni la validité de la dérivation. De même, dans les phrases « je veux » et « tu veux », le verbe « veux » (de catégorie (S\NP)) attend un NP sujet à gauche qui soit de la 1^{re} ou de la 2^e personne singulier féminin ou singulier masculin. Ce NP peut être « je » (PROASM) ou « tu » (PROBSM) sans changer la structure syntaxique.

- S (Phrase) : Les verbes sont notés en fonction de leur relation avec les sujets et les objets. Par exemple, (S\PROCSF : pronom personnel 3^e personne singulier féminin) indique un verbe qui requiert un sujet désigné par PROCSF pour former une phrase complète, illustrant ainsi les contraintes syntaxiques associées au verbe.
- Det (Déterminant) et DetS (Déterminant Pluriel) : Les déterminants introduisent des noms dans les groupes nominaux et sont notés en fonction de leur nombre, soit singulier (Det), soit pluriel (DetS). Par exemple, "De" et "Des" sont des déterminants indéfinis qui précisent le nombre des éléments désignés par les noms qu'ils introduisent.
- VP : Les verbes à l'infinitif, tels que "Manger", sont notés comme VP pour illustrer leur rôle dans les constructions verbales complexes.
- ADV(Adverbe) : Mot qui modifie un verbe, un adjectif, ou un autre adverbe, en précisant des informations sur la manière, le temps, le lieu, etc.
Exemples : vite, là, très
- S/S (Adverbe de temps) : Il s'agit d'un adverbe qui indique un moment précis dans le temps.

Exemples : demain, aujourd'hui, hier.

Ces notations permettent de représenter avec précision le rôle et la fonction de chaque élément dans une phrase, ce qui favorise une analyse syntaxique claire, cohérente et systématique.

VI.1.2) Catégories Fonctionnelles

Pour les besoins de notre projet, les catégories fonctionnelles sont représentées comme suit :

- (S/NP) : dans notre lexique, un verbe intransitif est noté S/NP, car il forme une phrase (S) lorsqu'il est combiné avec un groupe nominal (NP). Par exemple, un verbe comme "manger" nécessite un complément nominal pour compléter la phrase.

Exemple : Maman dort

- (S\PROC(SM))/NP : Structure verbale impliquant un sujet à la troisième personne du singulier masculin. Cela pourrait être une construction grammaticale plus complexe dans laquelle un pronom ou un groupe nominal agit en fonction du verbe.

Exemple : Il mange des pommes.

- ModalCSM(S\PROC(SM))/VP) : Bien que cette notation ne suive pas directement le modèle X/Y, les modaux comme ModalCSM exigent un autre verbe pour compléter la phrase. Ils représentent des fonctions spécialisées nécessitant un complément verbal.

Exemple : Ils veulent manger.

En adaptant ces notations fonctionnelles à notre lexique, nous pouvons fournir une description précise de la manière dont les différentes catégories interagissent pour former des structures syntaxiques plus complexes, facilitant ainsi une analyse approfondie et une meilleure compréhension des relations entre les éléments de la phrase.

VI.2) Les règles de combinaison :

Les règles sont définies en fonction des catégories syntaxiques attribuées aux différents mots et expressions dans notre lexique.

VI.2.1) Les déterminants

- Det : NP/Nm

Un déterminant (Det) peut être combiné avec un nom (Nm) pour former un groupe nominal (NP).

Exemple : Det + Nm → NP

- DetS : NPs/Ns

Un déterminant pluriel (DetS) peut être combiné avec un nom pluriel (NS) pour former un groupe nominal (NPS).

Exemple : DetS + NS → NPs

VI.2.2) Les modaux

Les modaux sont notés avec des catégories spécifiques qui indiquent leurs contraintes syntaxiques par rapport aux autres éléments de la phrase :

➤ MODALASM :: S\\PROASM/VP

Un modale (MODALASM) peut être combiné avec un groupe verbal (VP) pour produire une phrase (S) qui nécessite un pronom spécifique (PROASM).

➤ MODALCSM :: S\\PROCSM/VP

Un modale (MODALCSM) peut être combiné avec un groupe verbal (VP) pour produire une phrase (S) qui nécessite un pronom spécifique (PROCSM).

➤ MODALAPM :: S\\PROAPM/VP

Un modale (MODALAP) peut être combiné avec un groupe verbal (VP) pour produire une phrase (S) qui nécessite un pronom spécifique (PROAP).

➤ MODALBPM:: S\\PROBPM/VP

Un modale (MODALBPM) peut être combiné avec un groupe verbal (VP) pour produire une phrase (S) qui nécessite un pronom spécifique (PROBP).

➤ MODALCPM :: S\\PROCPM/VP

Un modale (MODALCPM) peut être combiné avec un groupe verbal (VP) pour produire une phrase (S) qui nécessite un pronom spécifique (PROCPM).

➤ MODALCPF :: S\\PROCPF/VP

Un modale (MODALCPF) peut être combiné avec un groupe verbal (VP) pour produire une phrase (S) qui nécessite un pronom spécifique (PROCPF).

- MODALCSF :: S\PROCSF/VP

Un modale (MODALCSF) peut être combiné avec un groupe verbal (VP) pour produire une phrase (S) qui nécessite un pronom spécifique (PROCSF).

VI.2.3) Verbes Transitifs et Intransitifs

- TV :: VP/NP

Un verbe transitif (TV) peut être combiné avec un groupe nominal (NP) à droite pour produire un groupe verbal (VP). Le groupe nominal sert de complément d'objet direct du verbe transitif.

Exemple : Manger une pomme

« Manger » est un verbe transitif, et « une pomme » est le complément d'objet direct (groupe nominal NP). L'ensemble forme un groupe verbal « manger une pomme ».(VP)

- S//NP => VP

Description : Une phrase (S) avec un groupe nominal (NP) à droite peut être considérée comme un groupe verbal (VP).

VI.2.4) Les adverbes

- hier :: S/S : Il s'agit d'un adverbe qui indique un moment précis dans le temps.

Exemple : demain, aujourd'hui, hier

- rapidement : (S\NP)/(S\NP)

Un adverbe comme rapidement modifie un groupe verbal (VP) en produisant un nouveau groupe verbal (VP).

Exemple : rapidement + VP → VP

VI.2.3) Verbes modaux et auxiliaires

- peux :: (S\PROASM)/NP

Un verbe modal « peut » peut être combiné avec un groupe nominal (NP) pour former une phrase (S) nécessitant un pronom spécifique (PROASM) ou (PROBSM).

- peut : (S\PROCSM)/NP

Un verbe modal peut être combiné avec un groupe nominal (NP) pour former une phrase (S) nécessitant un pronom spécifique (PROCSM).

- veut :: (S\PROCSF)/NP

Un verbe modal « veut » peut être combiné avec un groupe nominal (NP) pour former une phrase (S) nécessitant un pronom spécifique à la 3^e personne du singulier féminin (PROCSF)

En résumé, nos règles de combinaison sont comme suit :

Application Fonctionnelle (à droite)

- $(X/Y) Y \rightarrow X$

Exemple : Det + Nf \rightarrow NP

Application Fonctionnelle (à gauche)

- $Y (X/Y) \rightarrow X$

Exemple : Nf + (NP\Nf) \rightarrow NP

Composition :

- À droite : $(X/Y) (Y/Z) \rightarrow X/Z$
- À gauche : $(X/Y) (Y/Z) \rightarrow XZ$

Exemple : (S/NP) + (NP/S) \rightarrow S/S

Règles de composition

$$\begin{array}{c} [X/Y : u_1] - [Y/Z : u_2] \\ \text{-----} > B \\ [X/Z : (B u_1 u_2)] \end{array}$$

$$\begin{array}{c} [Y \setminus Z : u_1] - [X \setminus Y : u_2] \\ \text{-----} < B \\ [X \setminus Z : (B u_2 u_1)] \end{array}$$

Règles de substitution

$$\begin{array}{c} [(X/Y) \setminus Z : u_1] - [Y \setminus Z : u_2] \\ \text{-----} > S_x \\ [X \setminus Z : (S u_1 u_2)] \end{array}$$

$$\begin{array}{c} [Y \setminus Z : u_1] [(X \setminus Y) / Z : u_2] \\ \text{-----} > S_x \\ [X / Z : (S u_2 u_1)] \end{array}$$

Règles de changement de type

$$\begin{array}{c} [X : u] \\ \text{-----} > T \\ [Y / (Y \setminus X) : (C^*, u)] \end{array}$$

$$\begin{array}{c} [X : u] \\ \text{-----} < T \\ [Y \setminus (Y / X) : (C^* u)] \end{array}$$

Contraction

C'est la simplification de la combinaison en éliminant les catégories intermédiaires redondantes.

$$\text{Exemple : } (S/NP) + (NP/S) \rightarrow S/S$$

Toutes ces règles permettent d'analyser et de générer des structures syntaxiques à partir des catégories définies dans notre lexique. Elles assurent que les éléments sont combinés de manière correcte pour former des phrases grammaticalement valides. Elles seront utilisées dans notre projet pour effectuer une analyse correcte de nos cent phrases.

VII) Fonctionnement de la plateforme

Notre plateforme a été conçue pour permettre une analyse automatique de phrases en français à l'aide de la Grammaire Catégorielle Combinatoire et Applicative (GCCA), enrichie par un système de détection et de suggestion d'erreurs via LanguageTool. Le fonctionnement repose sur les étapes suivantes :

VII.1) Écriture de la phrase à analyser

L'utilisateur commence par écrire sa phrase ou une liste de phrases à analyser comme dans la figure 5 :

```
# Test the function with example sentences
sentences = [
    "Le port du voile est obligatoire",
    "Je respecte mes parents" ,
    "Vous pouvez manger de la soupe"
]
```

Figure 5 : Interface entrée de phrases

VII.2) Exécution du code d'analyse

Une fois les phrases écrites, pour faire l'analyse, l'utilisateur peut directement appuyer sur le bouton exécuter qui se trouve sur la partie gauche du code comme le montre la figure 6 :

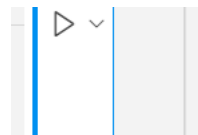


Figure 6 : Bouton d'exécution du code

VII.3) Analyse grammaticale avec GCCA

Chaque mot est associé à une catégorie syntaxique issue d'un lexique structuré, et la phrase est évaluée selon les règles de combinaison (application avant, arrière, composition, changement de type.). Si la phrase est syntaxiquement valide, elle est enregistrée avec sa dérivation comme le montre la figure 7 :

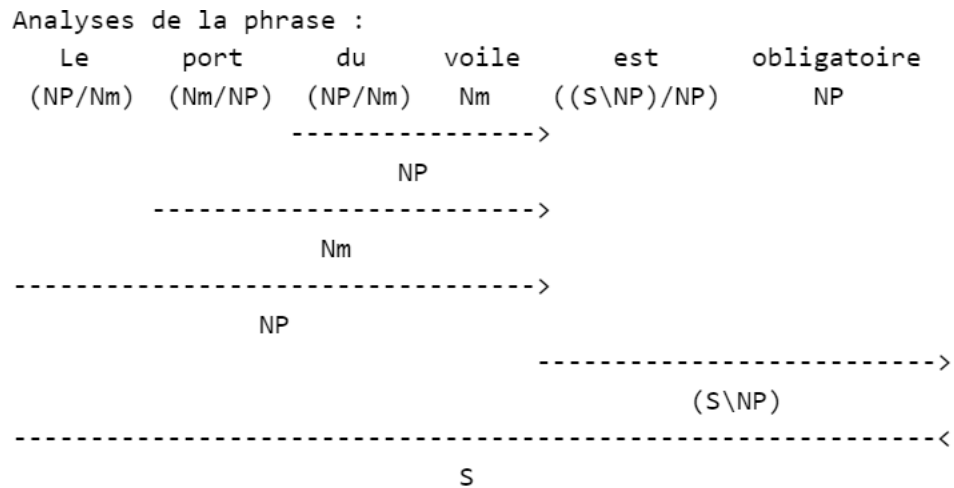


Figure 7 : Résultat dérivation CCG d'une phrase correcte

VII.4) Détection d'erreurs avec LanguageTool

Si la phrase échoue à l'analyse GCCA donc est syntaxiquement invalide, elle est enregistrée avec sa dérivation et est transmise à LanguageTool qui détecte les erreurs et propose des suggestions, pondérées selon des critères de pertinence comme le montre la figure 8 :

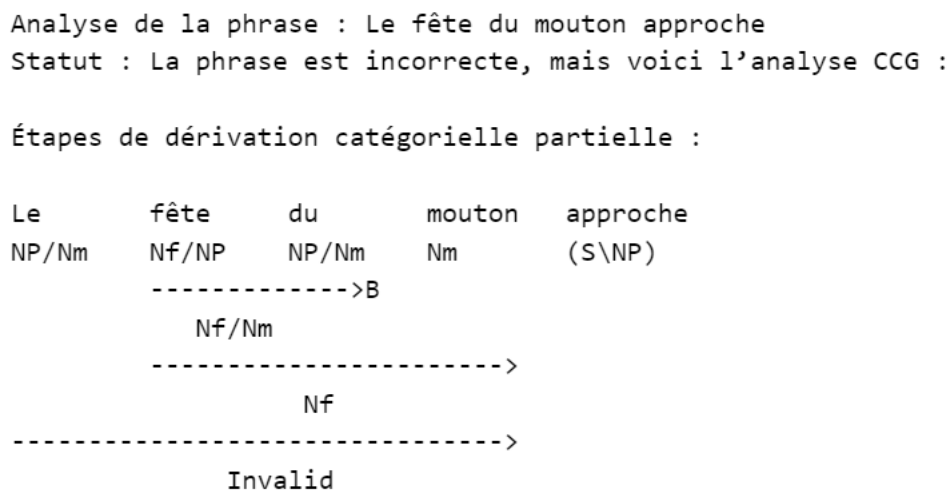


Figure 8 : Résultat dérivation CCG d'une phrase incorrecte

VII.5) Traitement des suggestions

Pour chaque erreur détectée, LanguageTool propose une ou plusieurs suggestions. Nous sélectionnons automatiquement la suggestion la plus pertinente en nous basant sur la position dans la phrase, et le contexte grammatical. Pour rendre les suggestions proposées plus compréhensibles et pédagogiques, nous avons introduit un système de pondération linéaire

décroissante. Il permet d'attribuer un poids à chaque suggestion reflétant sa position dans la liste proposée. Cette approche permet de recommander des corrections cohérentes et adaptées à l'usage du français comme le montre la figure 9 :

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Le fête du mouton approche	Phrase incorrecte	Position 0: Le fête du mouton approche -> La fête : 100.00%,

Erreurs et Suggestions
Position 0: Le fête du mouton approche -> La fête : 100.00%, Les fêtes : 50.00%

Figure 9 : Suggestions de corrections pour une phrase incorrecte

VIII) Conclusion

Ce chapitre a exposé les objectifs de notre projet et la plateforme choisie pour sa mise en œuvre. Nous avons présenté les langages de programmation et les bibliothèques sélectionnés, soulignant leur rôle crucial dans l'atteinte de nos objectifs. Nous avons également détaillé la méthode de représentation des données, essentielle pour garantir une analyse précise. Ces éléments posent les bases pour une compréhension approfondie des résultats et des implications que nous aborderons dans les chapitres suivants.

Chapitre V : Expérimentation

I) Introduction

Dans cette partie, nous décrivons en détail notre expérimentation en suivant différentes étapes clés.

En effet, nous avons constitué un corpus de 100 phrases en français pour tester notre modèle d'analyse syntaxique et de correction grammaticale. Ce corpus inclut des phrases de diverses complexités et erreurs pour évaluer l'efficacité de notre système.

Dans cette partie, nous montrerons comment notre application fonctionne pour faire une bonne analyse grammaticale quand la phrase est correcte et la réponse que notre système renvoie quand les phrases énoncées sont incorrectes.

II) Le corpus

Un corpus est un recueil organisé de textes ou d'enregistrements de la langue parlée soigneusement sélectionnés et structurés pour servir de référence dans des études linguistiques. Il représente une base de données textuelle riche et diversifiée permettant d'analyser des usages réels de la langue.

Les corpus peuvent être constitués de textes provenant de livres, d'articles, de discours, de conversations, ou de toute autre source de communication verbale.

Pour les besoins de notre projet, nous avons-nous même constitué notre propre corpus composé de 100 phrases (50 phrases correctes et 50 phrases incorrectes) de diverses complexités et erreurs pour évaluer l'efficacité de notre système. Ces 100 phrases sont matérialisées dans l'Annexe I.

III) Le lexique

Comme nous l'avons déjà dit plus haut, la grammaire catégorielle est une méthode formelle pour analyser la syntaxe des langues naturelles, basée sur le concept que les mots et les phrases sont assignés à des catégories syntaxiques. Ces catégories sont alors combinées suivant des règles spécifiques pour créer des structures grammaticalement correctes (Gerner & STUMPF, 2007). Dans ce cadre, un lexique catégoriel attribue à chaque mot une ou plusieurs catégories syntaxiques. Dans notre projet, nous avons créé notre propre lexique qui nous permettra de faire l'analyse catégorielle. Quelques exemples sont consignés dans l'annexe 2.

IV) Captures des résultats des cinquante phrases correctes

IV.1) Introduction

Nous avons créé une fonction `is_sentence_valid` qui permet de vérifier si la phrase peut être analysée correctement.

Le test final imprime si la phrase est bien formée ou non selon le lexique et les règles définis.

En résumé, notre code utilise la grammaire catégorielle combinatoire (CCG) pour analyser la structure syntaxique des phrases en associant chaque mot à une catégorie syntaxique spécifique dans notre lexique, puis en utilisant un analyseur syntaxique CCG pour vérifier si la phrase est correctement construite selon ces règles. Les figures ci-dessous montrent l'analyse CCG réalisée des cinquante phrases.

➤ Phrase 1 : Je mange bien

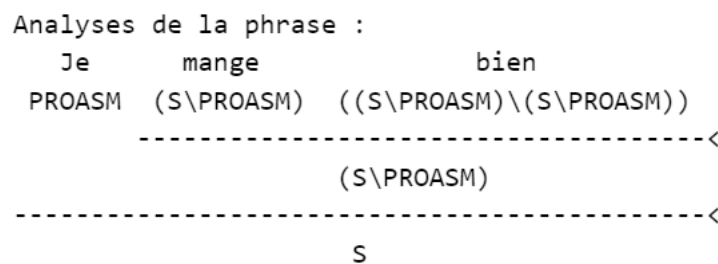


Figure 10 : Analyse CCG phrase 1

Cette dérivation montre que la phrase 1 « Je mange bien » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle, et les combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 2 : Elles veulent manger de la soupe

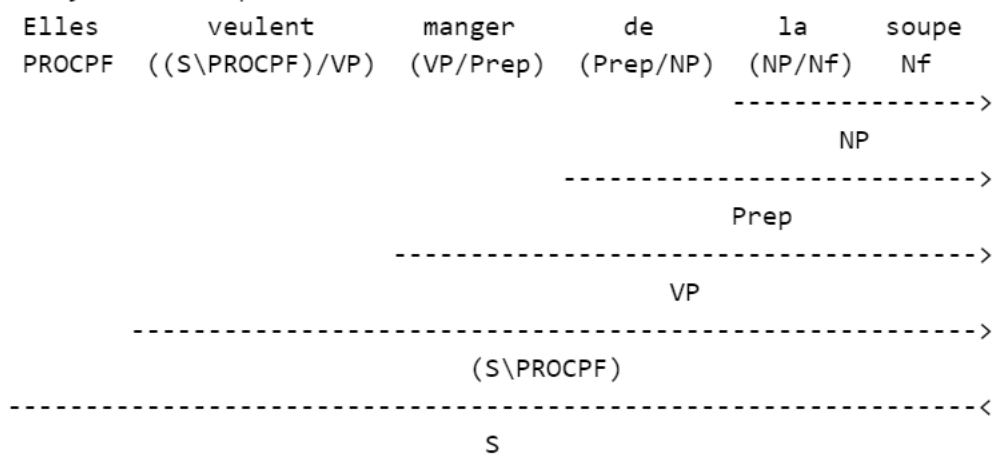


Figure 11 : Analyse CCG phrase 2

Cette dérivation montre que la phrase 2 « Elles veulent manger de la soupe » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle, et les combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 3 : Elles pensent aller au marché

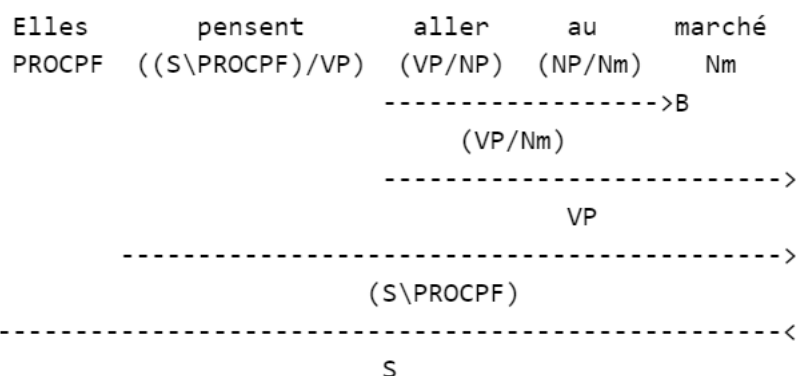


Figure 12 : Analyse CCG phrase 3

Cette dérivation montre que la phrase 3 « Elles pensent aller au marché » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 4: Plus-tard nous irons au salon

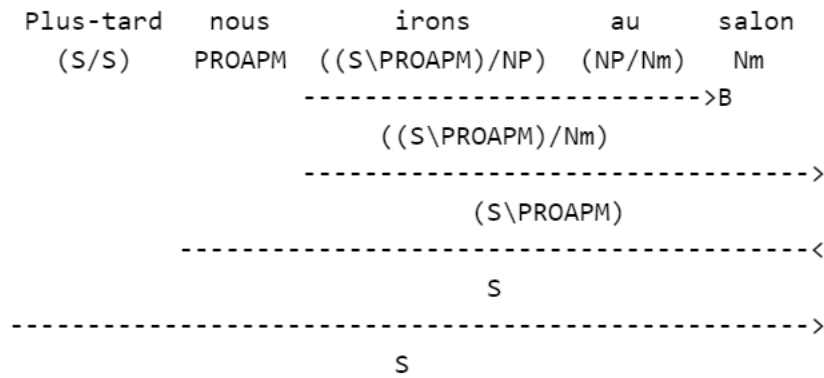


Figure 13 : Analyse CCG phrase 4

Cette dérivation montre que la phrase 4 « Plus-tard nous irons au salon » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 5 : Nous prions pour vous

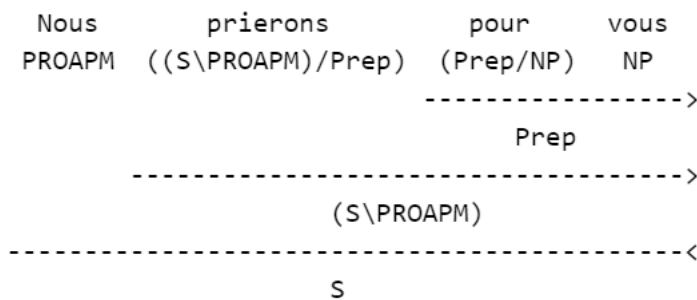


Figure 14 : Analyse CCG phrase 5

Cette dérivation montre que la phrase 5 « Nous prions pour vous » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle, et les combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 6 : Son enfant marche lentement

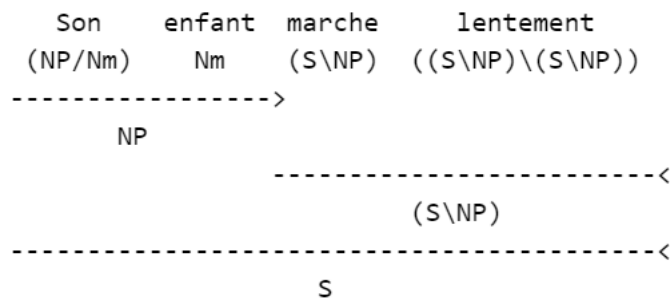


Figure 15 : Analyse CCG phrase 6

Cette dérivation montre que la phrase 6 « Son enfant marche lentement » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle, et les combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 7 : La famille est sacrée

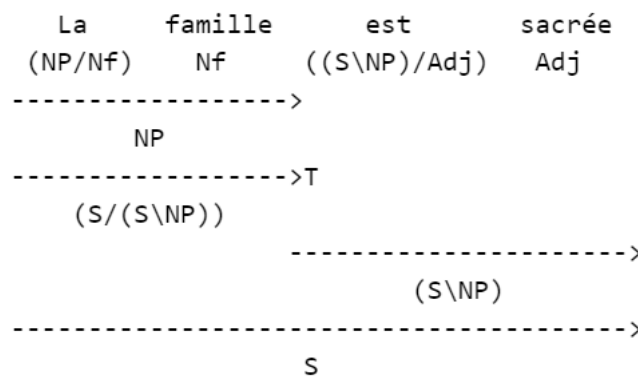


Figure 16 : Analyse CCG phrase 7

Cette dérivation montre que la phrase 7 « La famille est sacrée » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de changement de type permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 8 : La réussite reste sa priorité

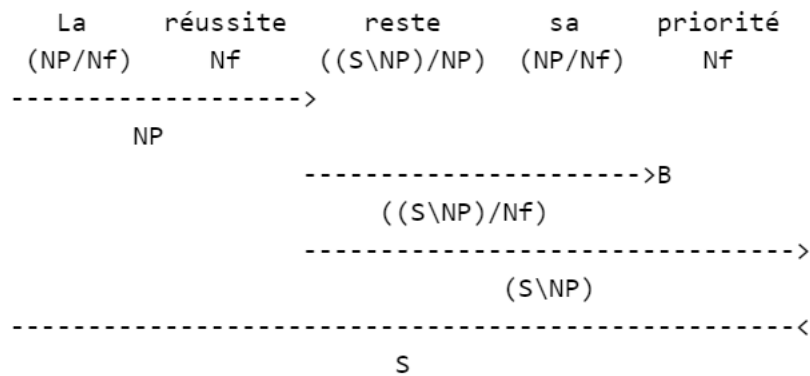


Figure 17 : Analyse CCG phrase 8

Cette dérivation montre que la phrase 8 « La réussite reste sa priorité » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 9 : Il est le plus âgé d'entre nous

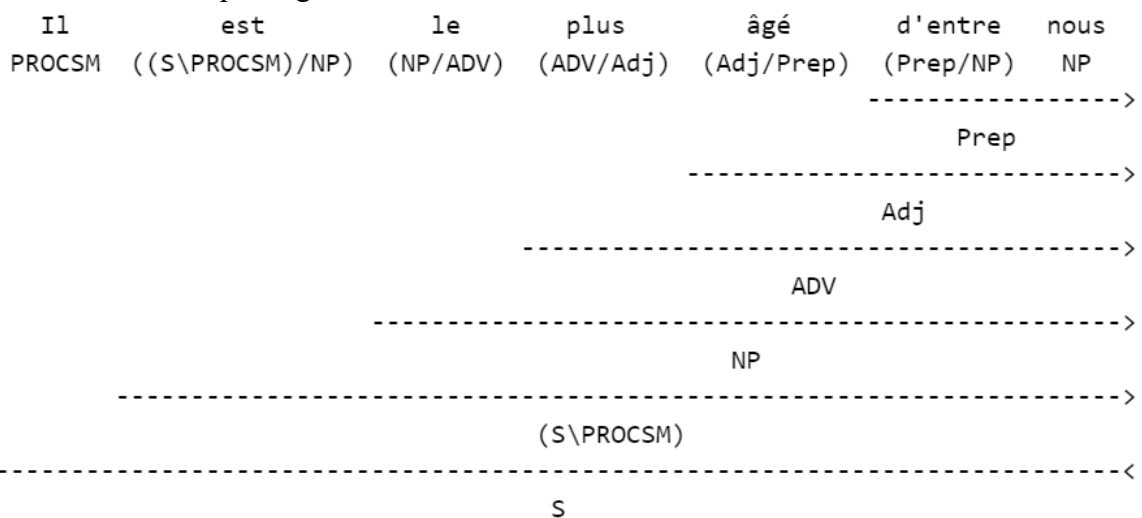


Figure 18 : Analyse CCG phrase 9

Cette dérivation montre que la phrase 9 « Il est le plus âgé d'entre nous » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle, et les combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 10 : Ma mère me-manque tellement

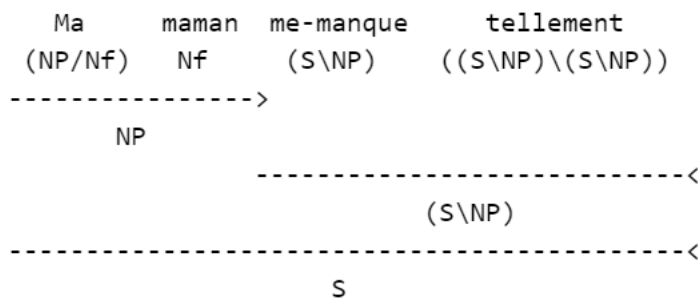


Figure 19 : Analyse CCG phrase 10

Cette dérivation montre que la phrase 10 « Ma mère me-manque tellement » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle, et les combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

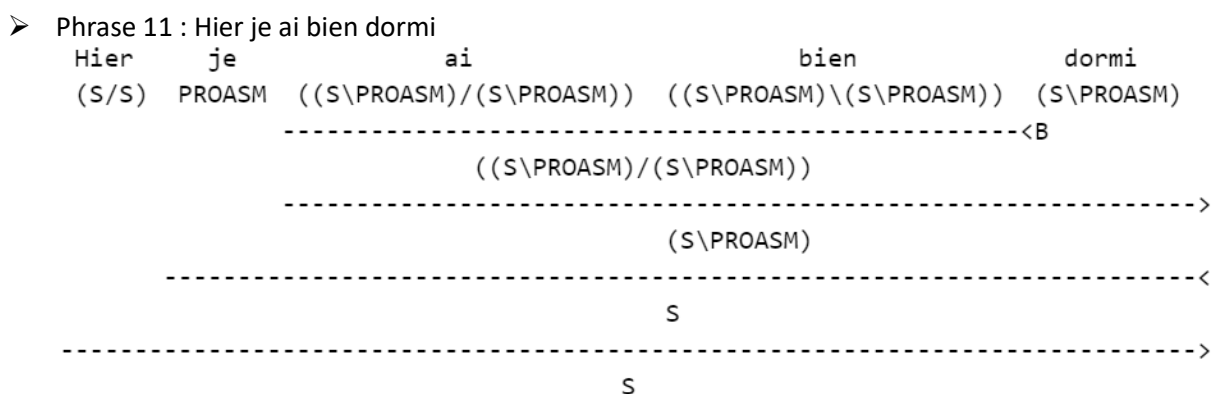


Figure 20 : Analyse CCG phrase 11

Cette dérivation montre que la phrase 11 « Hier je ai bien dormi » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 12 : Nous irons tous au paradis

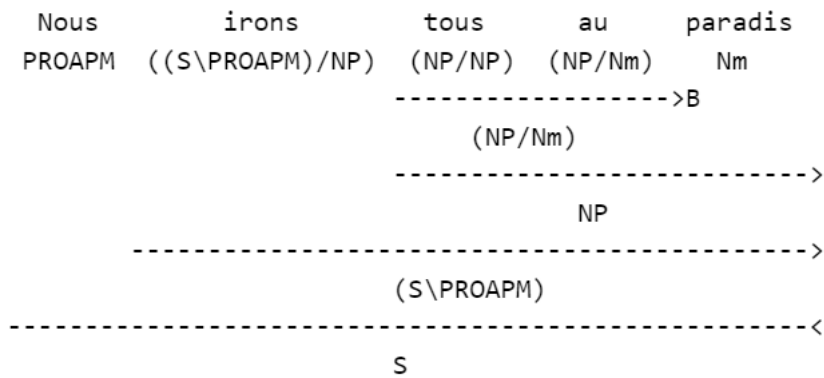


Figure 21 : Analyse CCG phrase 12

Cette dérivation montre que la phrase 12 « Nous irons tous au paradis » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 13 : Nous allons nous-tresser

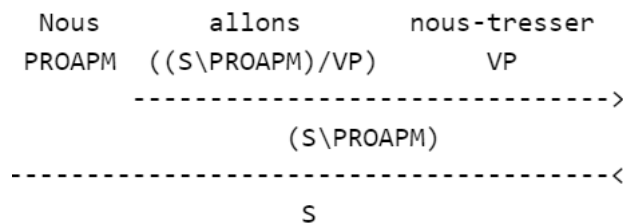


Figure 22 : Analyse CCG phrase 13

Cette dérivation montre que la phrase 13 « Nous allons nous-tresser » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 14 : La vie est tellement courte

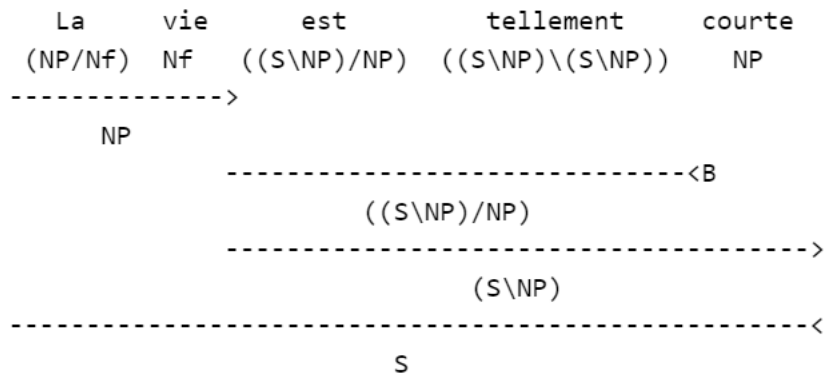


Figure 23 : Analyse CCG phrase 14

Cette dérivation montre que la phrase 14 « La vie est tellement courte » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 15 : Le professeur est généreux

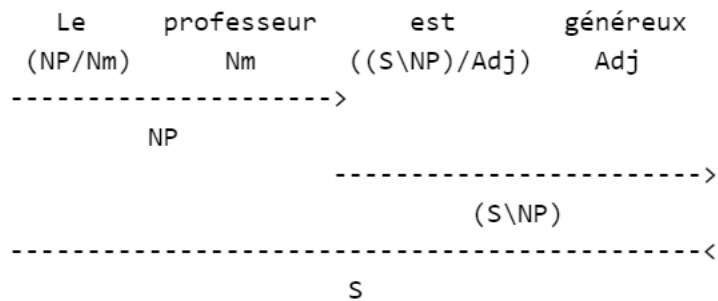


Figure 24 : Analyse CCG phrase 15

Cette dérivation montre que la phrase 15 « Le professeur est généreux » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 16 : Elle a eu la meilleure note

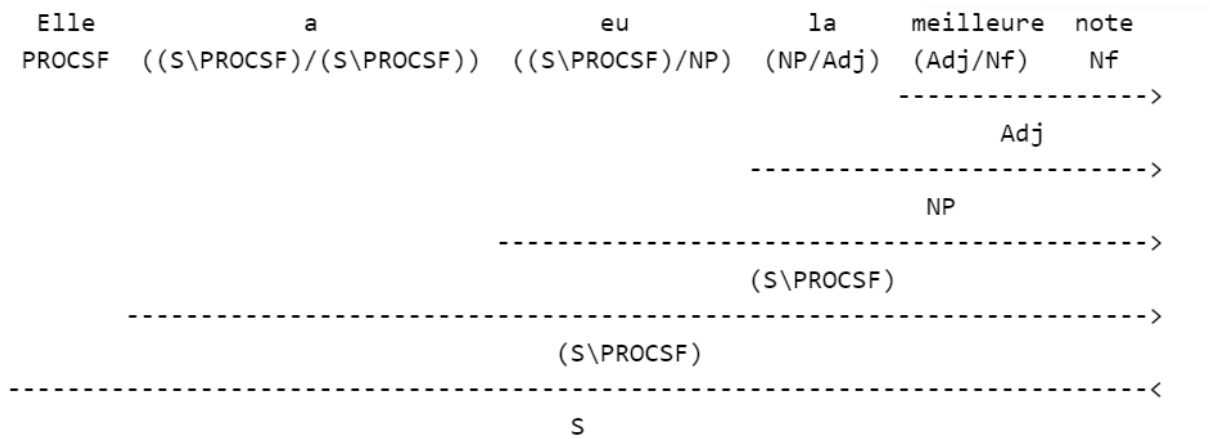


Figure 25 : Analyse CCG phrase 16

Cette dérivation montre que la phrase 16 « Elle a eu la meilleure note » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 17 : Je pense que il est timide

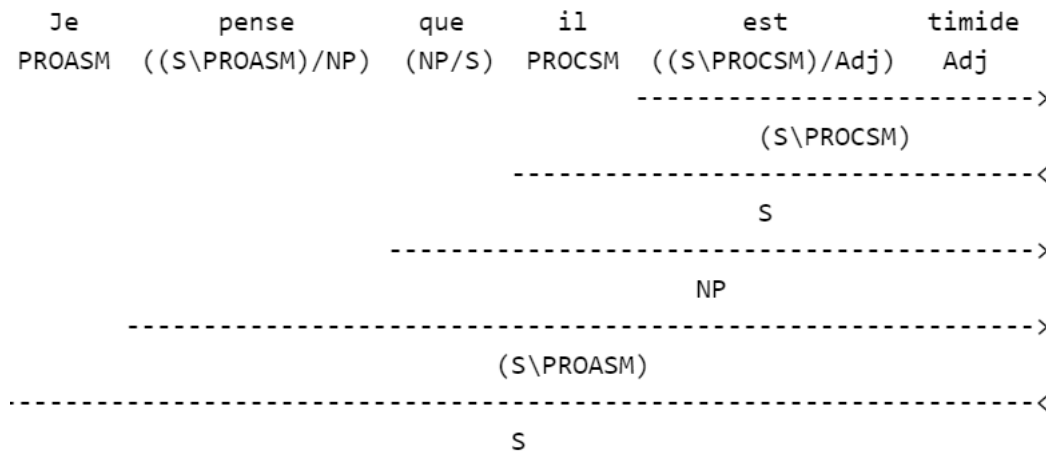


Figure 26 : Analyse CCG phrase 17

Cette dérivation montre que la phrase 17 « Je pense que il est timide » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 18 : Son nom est Maryam

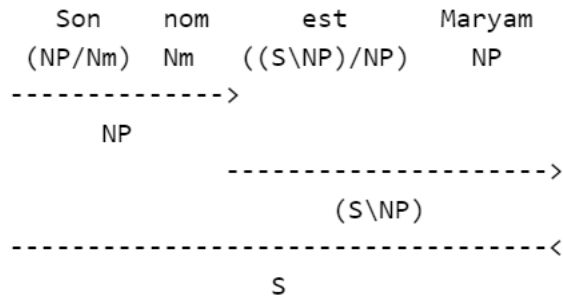


Figure 27 : Analyse CCG phrase 18

Cette dérivation montre que la phrase 18 « Son nom est Maryam » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 19 : Je ne-saurais te dire

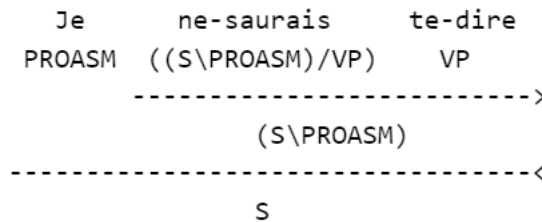


Figure 28 : Analyse CCG phrase 19

Cette dérivation montre que la phrase 19 « Je ne-saurais te-dire » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 20 : Ce-matin je suis très heureuse

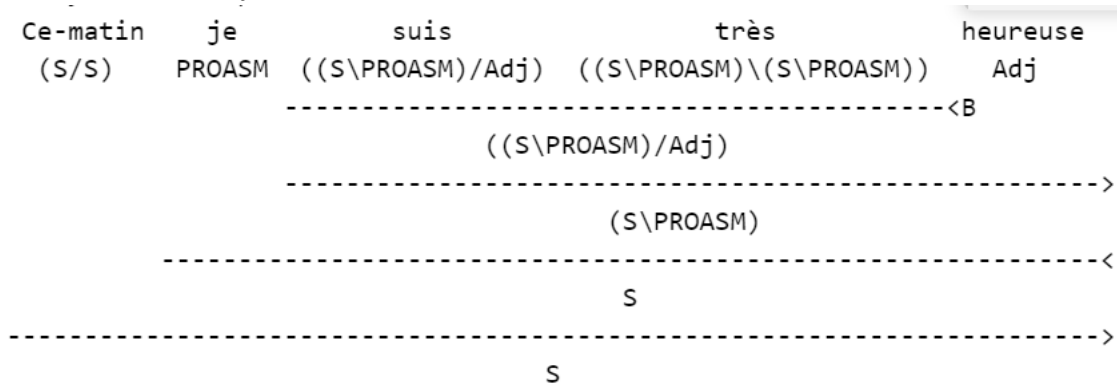


Figure 29 : Analyse CCG phrase 20

Cette dérivation montre que la phrase 20 « Ce-matin je suis très heureuse » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

- Phrase 21 : Ma fille craint de-se-blesser

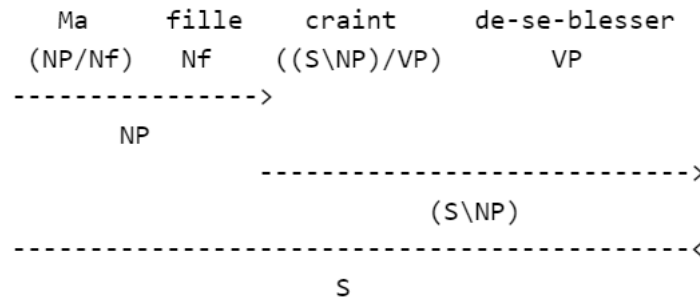


Figure 30 : Analyse CCG phrase 21

Cette dérivation montre que la phrase 21 « Ma fille craint de-se-blesser » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

- Phrase 22 : Elle pensait être seule

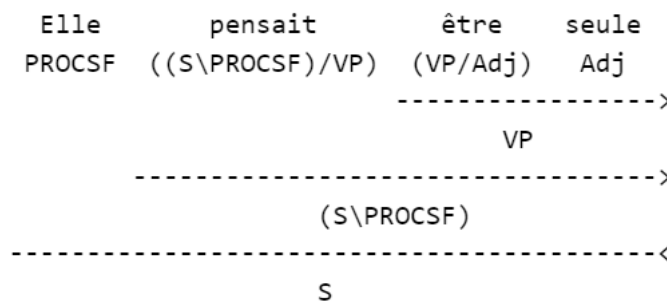


Figure 31 : Analyse CCG phrase 22

Cette dérivation montre que la phrase 22 « Elle pensait être seule » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

- Phrase 23 : Elle mérite vraiment sa promotion

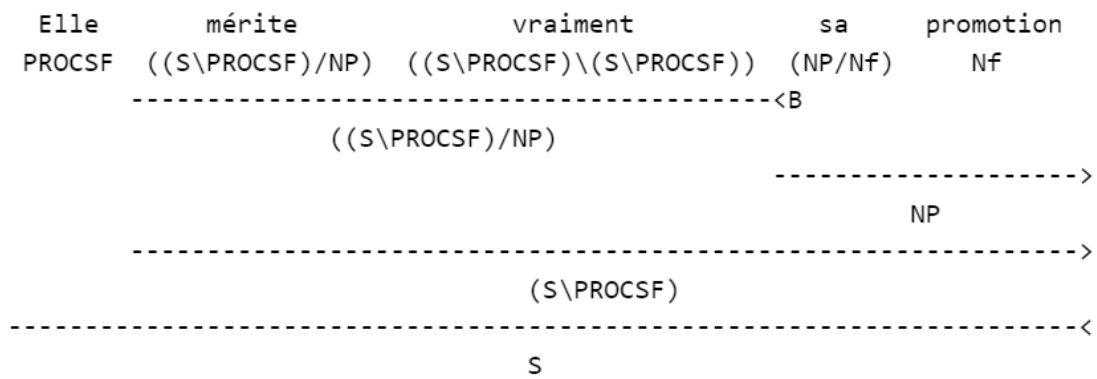


Figure 32 : Analyse CCG phrase 23

Cette dérivation montre que la phrase 23 « Elle mérite vraiment sa promotion » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 24 : Maman est tellement serviable

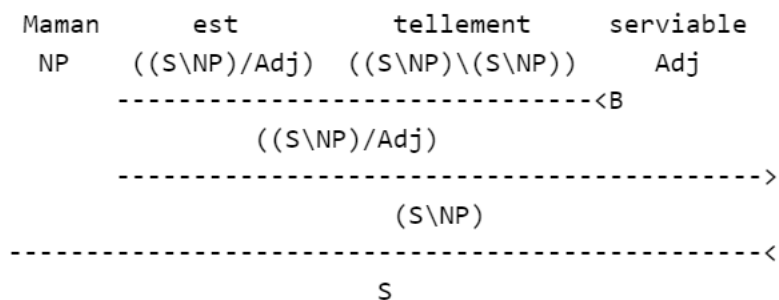


Figure 33 : Analyse CCG phrase 24

Cette dérivation montre que la phrase 24 « Maman est tellement serviable » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 25 : Ma maman reste ma meilleure amie

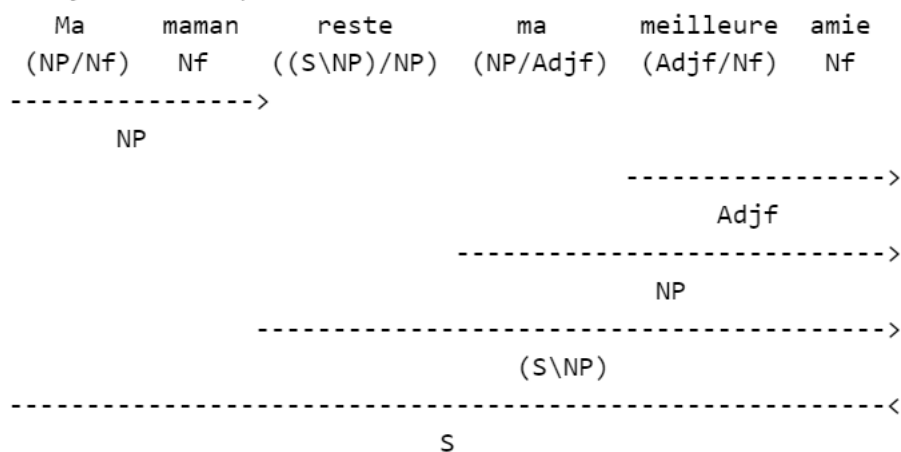


Figure 34 : Analyse CCG phrase 25

Cette dérivation montre que la phrase 25 « Ma maman reste ma meilleure amie » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 26 : Je ai réalisé un excellent projet

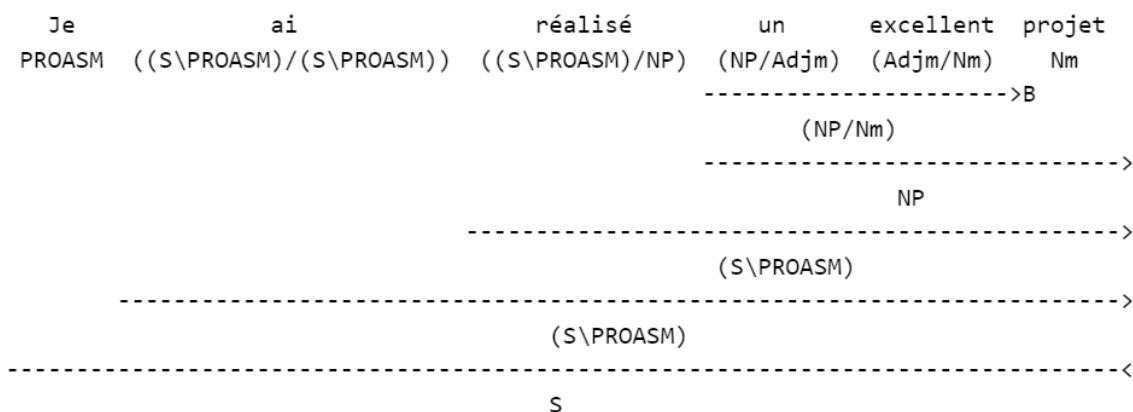


Figure 35 : Analyse CCG phrase 26

Cette dérivation montre que la phrase 26 « Je ai réalisé un excellent projet » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 27 : Le vol est strictement interdit

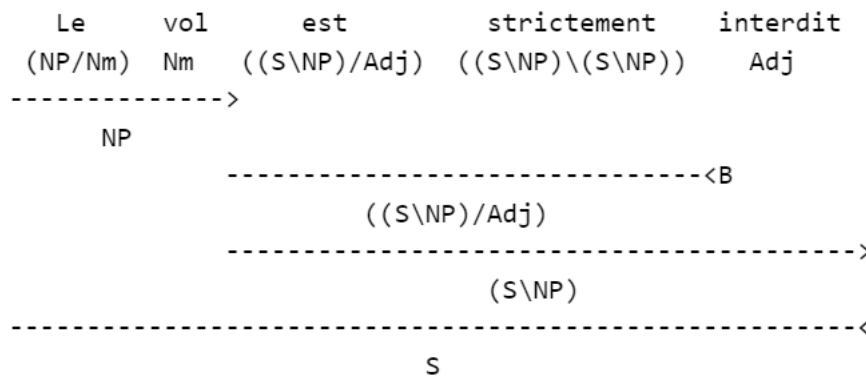


Figure 36 : Analyse CCG phrase 27

Cette dérivation montre que la phrase 27 « Le vol est strictement interdit » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 28 : Nous avons le droit de-manger

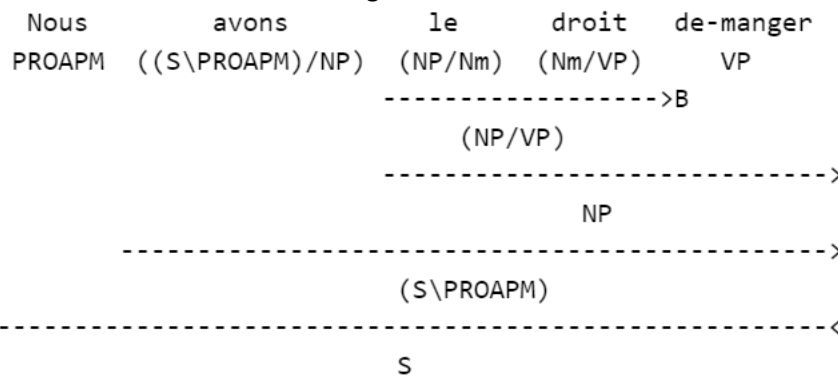


Figure 37 : Analyse CCG phrase 28

Cette dérivation montre que la phrase 28 « Nous avons le droit de-manger » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 29 : Elle t'a donné une dernière chance

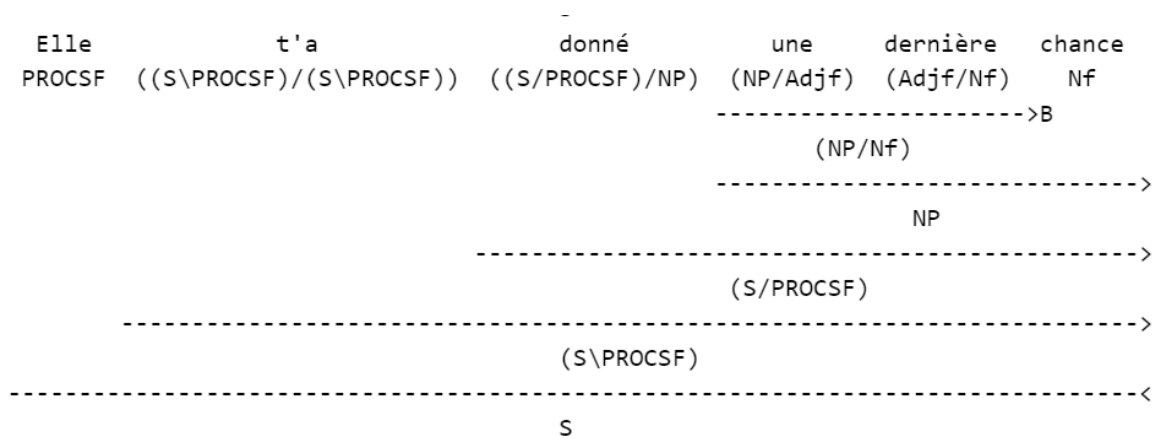


Figure 38 : Analyse CCG phrase 29

Cette dérivation montre que la phrase 29 « Elle t'a donné une dernière chance » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

- Phrase 30 : Nous craignons pour lui

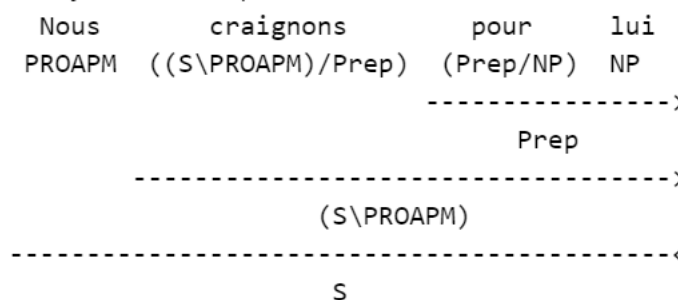


Figure 39 : Analyse CCG phrase 30

Cette dérivation montre que la phrase 30 « Nous craignons pour lui » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

- Phrase 31 : Je ai tant à-offrir

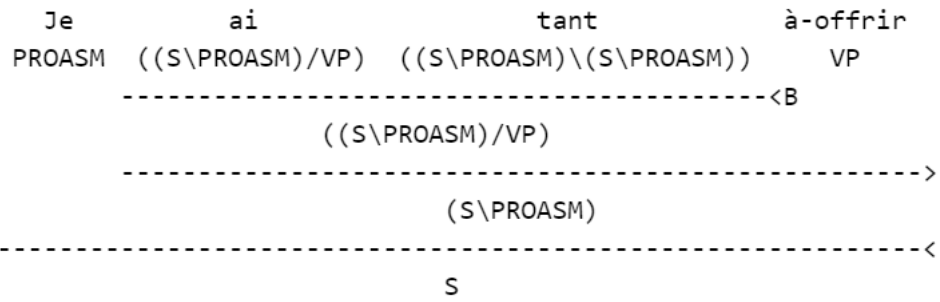


Figure 40 : Analyse CCG phrase 31

Cette dérivation montre que la phrase 31 « Je ai tant à-offrir » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 32 : La confiance en-soi est une qualité

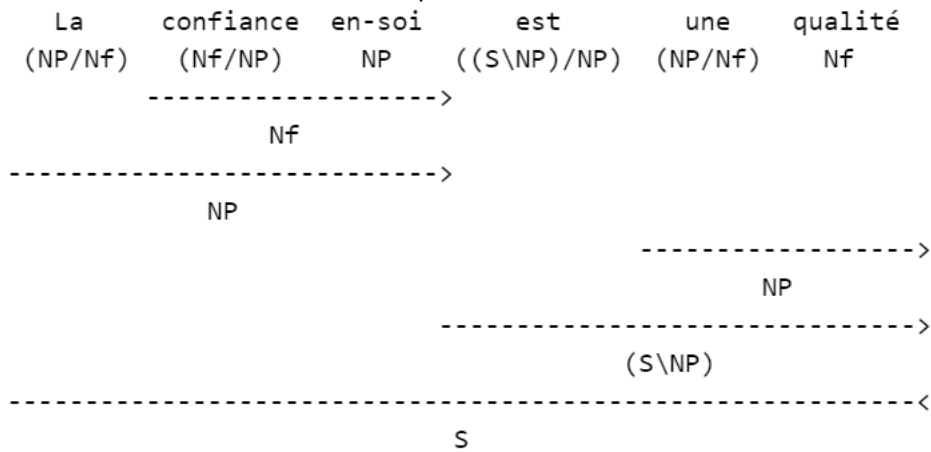


Figure 41 : Analyse CCG phrase 32

Cette dérivation montre que la phrase 32 « La confiance en-soi est une qualité » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 33 : Ils chantent en chœur

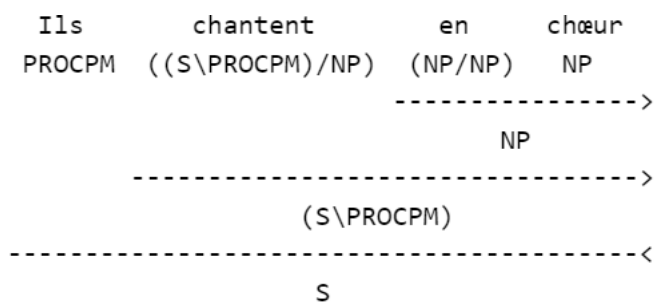


Figure 42 : Analyse CCG phrase 33

Cette dérivation montre que la phrase 33 « Ils chantent en chœur » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 34 : L'eau de la piscine est jaune

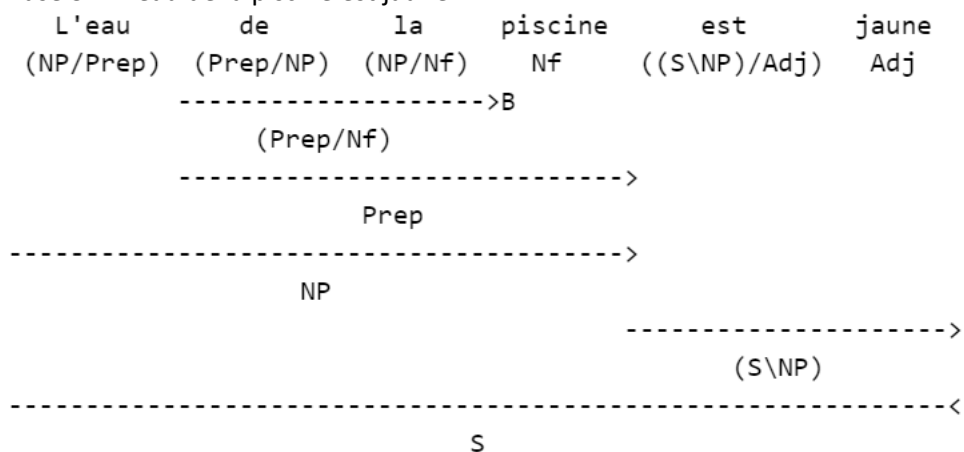


Figure 43 : Analyse CCG phrase 34

Cette dérivation montre que la phrase 34 « L'eau de piscine est jaune » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 35 : Papa a tellement mal au cœur

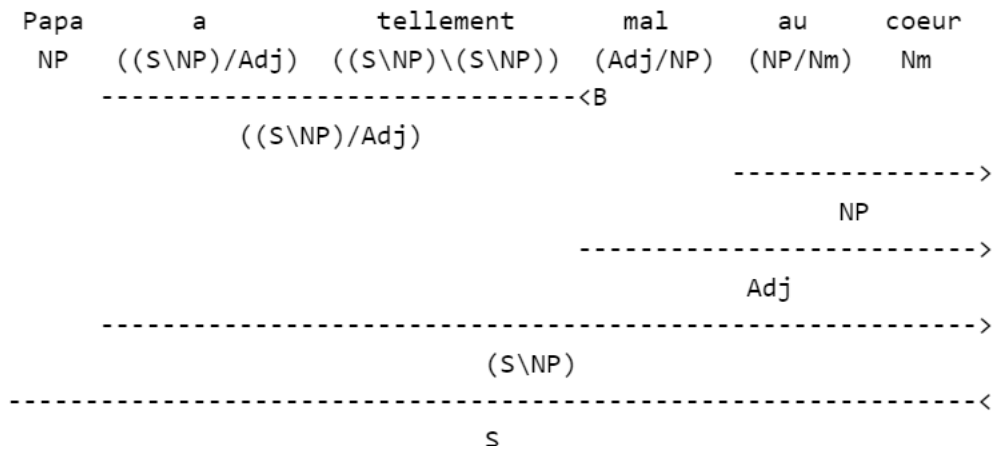


Figure 44 : Analyse CCG phrase 35

Cette dérivation montre que la phrase 35 « Papa a tellement mal au cœur » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 36 : Je ne-suis pas comme toi

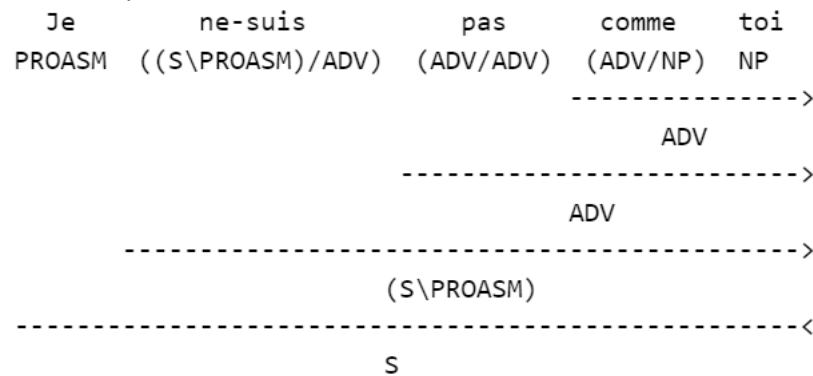


Figure 45 : Analyse CCG phrase 36

Cette dérivation montre que la phrase 36 « Je ne-suis pas comme toi » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 37 : Ma famille a honte de moi

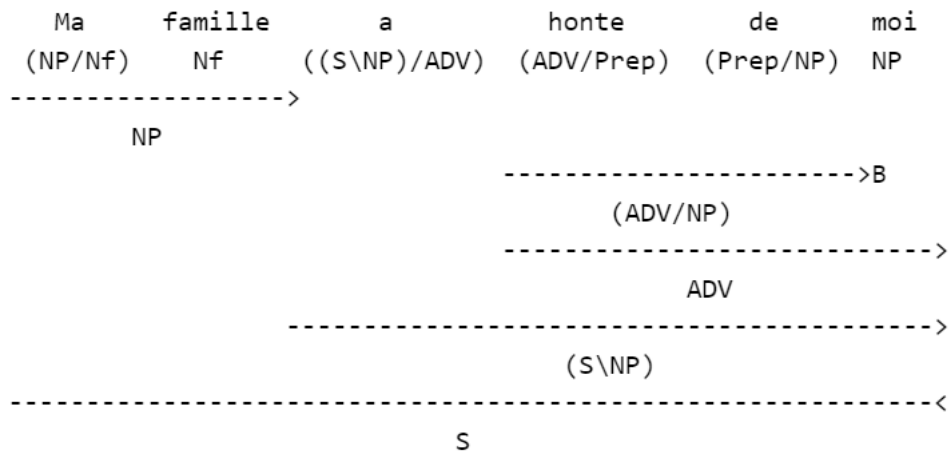


Figure 46 : Analyse CCG phrase 37

Cette dérivation montre que la phrase 37 « Ma famille a honte de moi » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 38 : Je dis non à la violence

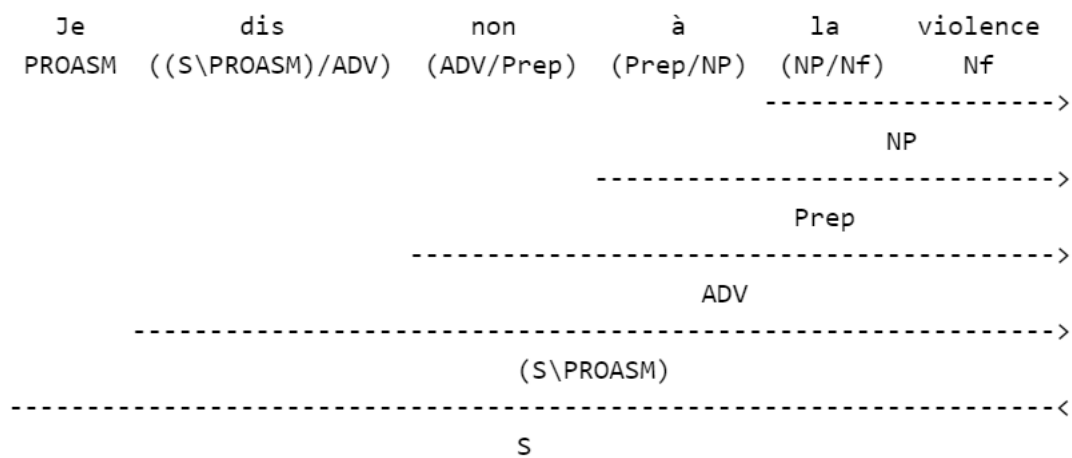


Figure 47 : Analyse CCG phrase 38

Cette dérivation montre que la phrase 38 « Je dis non à la violence » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 39 : Je ai fini la rédaction de mon mémoire

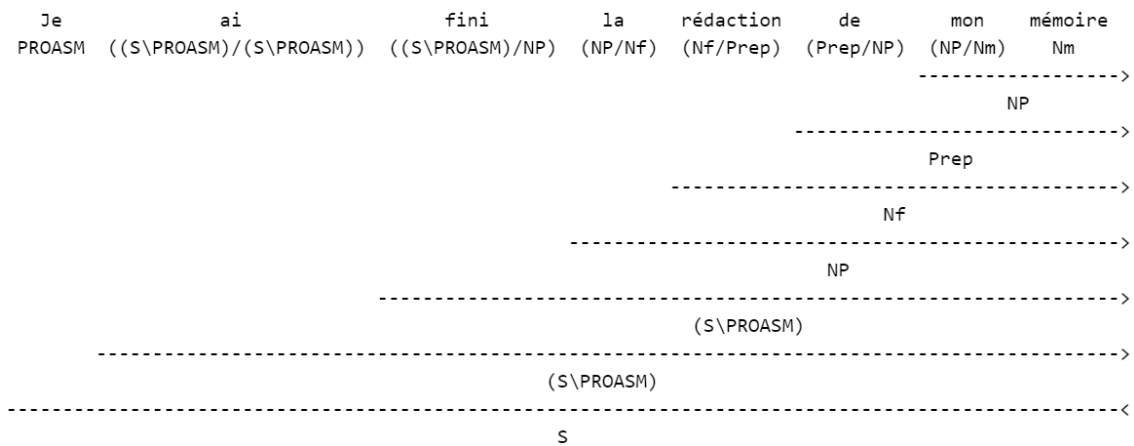


Figure 48 : Analyse CCG phrase 39

Cette dérivation montre que la phrase 39 « Je ai fini la rédaction de mon mémoire » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 40 : Le port du voile est obligatoire

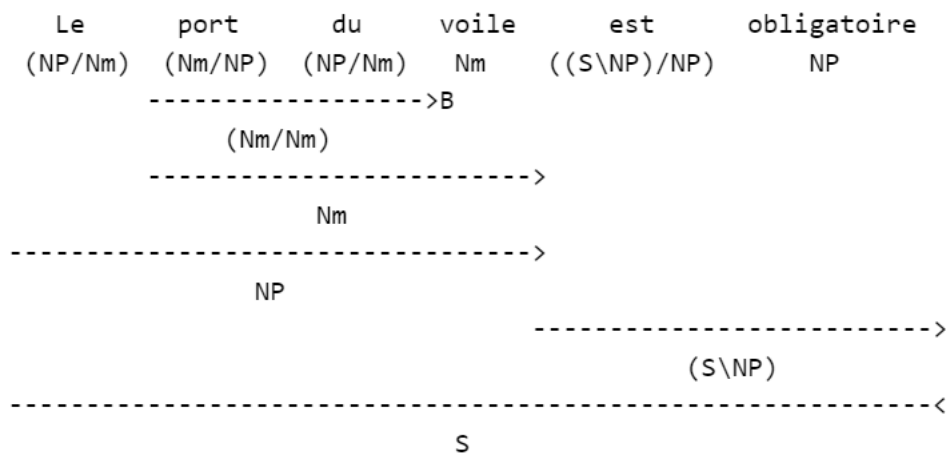


Figure 49 : Analyse CCG phrase 40

Cette dérivation montre que la phrase 40 « Le port du voile est obligatoire » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 41 : Il avait un mal de tête

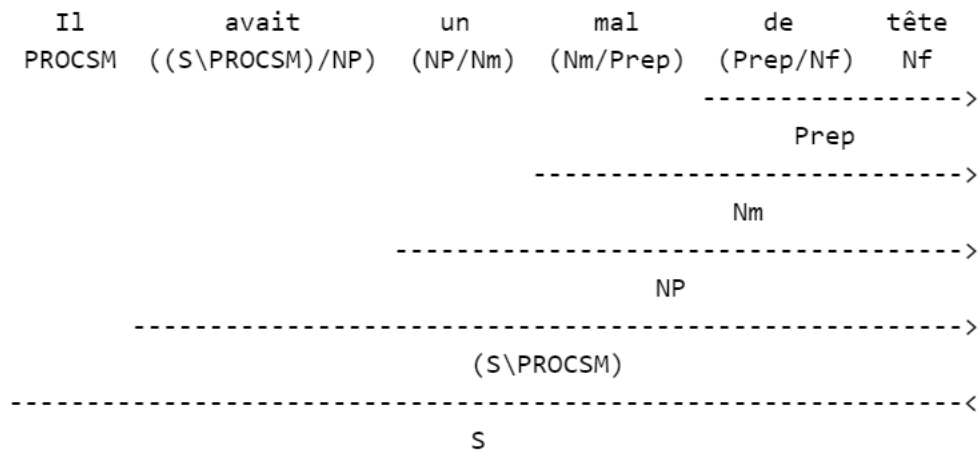


Figure 50 : Analyse CCG phrase 41

Cette dérivation montre que la phrase 41 « Il avait un mal de tête » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 42 : Son parfum sent tellement bon

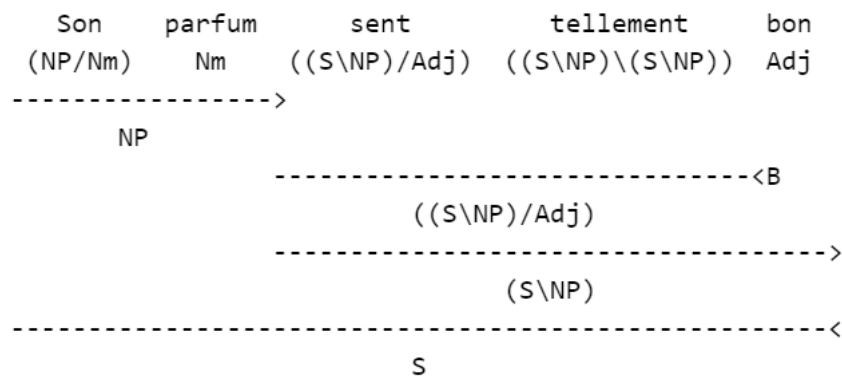


Figure 51 : Analyse CCG phrase 42

Cette dérivation montre que la phrase 42 « Son parfum sent tellement bon » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 43 : Vous voulez vraiment sortir

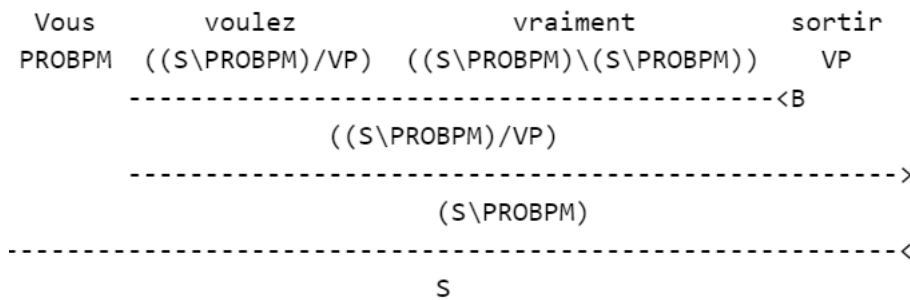


Figure 52 : Analyse CCG phrase 43

Cette dérivation montre que la phrase 43 « Vous voulez vraiment sortir » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 44 : Je me-marie en janvier

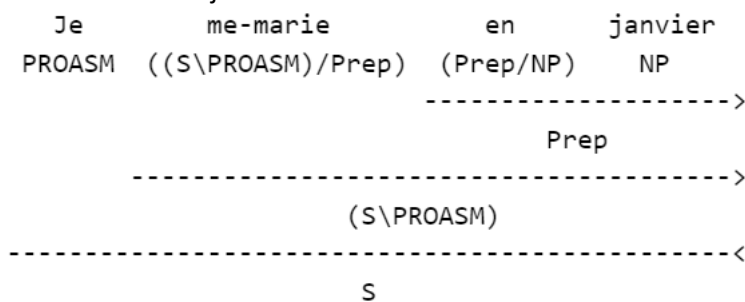


Figure 53 : Analyse CCG phrase 44

Cette dérivation montre que la phrase 44 « Je me marie en janvier » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 45 : Elle a vraiment besoin d'argent

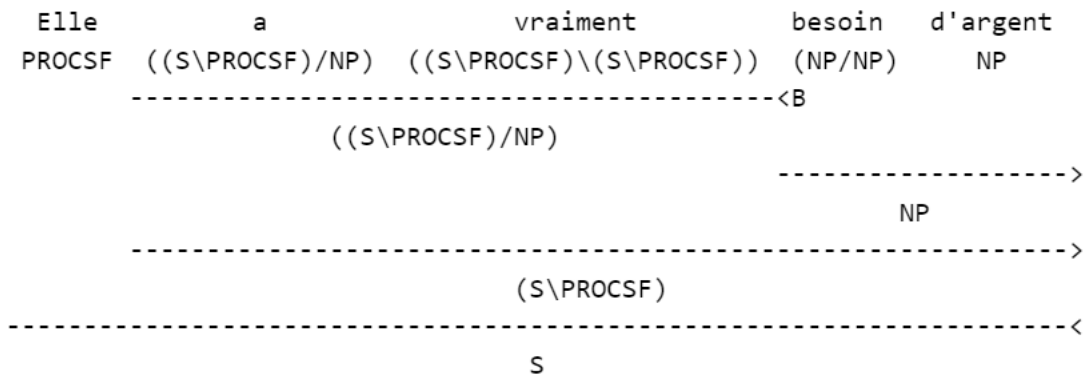


Figure 54 : Analyse CCG phrase 45

Cette dérivation montre que la phrase 45 « Elle a vraiment besoin d'argent » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 46 : L'amitié n'existe plus

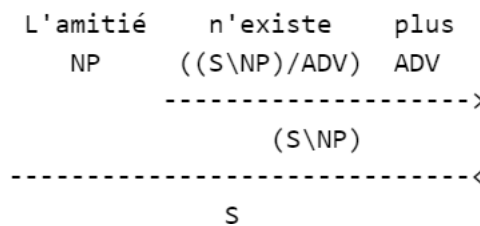


Figure 55 : Analyse CCG phrase 46

Cette dérivation montre que la phrase 46 « L'amitié n'existe plus » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 47 : L'automne approche

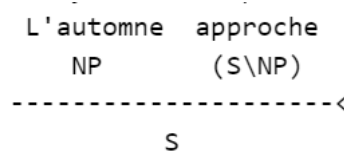


Figure 56 : Analyse CCG phrase 47

Cette dérivation montre que la phrase 47 « L'automne approche » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons

fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 48 : Chaque-matin je rends grâce à Allah

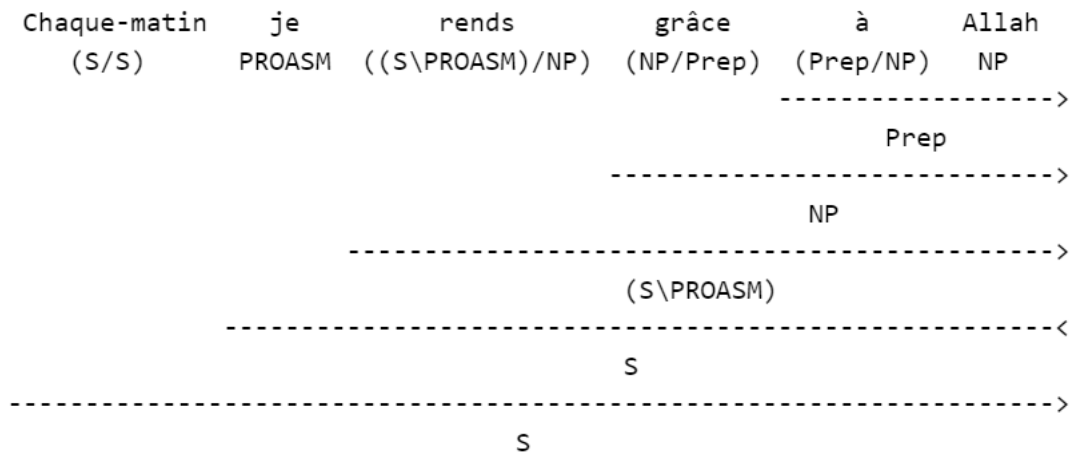


Figure 57 : Analyse CCG phrase 48

Cette dérivation montre que la phrase 48 « Chaque-matin je rends grâce à Allah » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 49 : Je veux me-marier

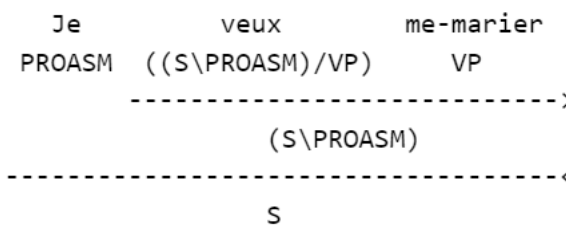


Figure 58 : Analyse CCG phrase 49

Cette dérivation montre que la phrase 49 « Je veux me-marier » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

➤ Phrase 50 : Elle est trop timide

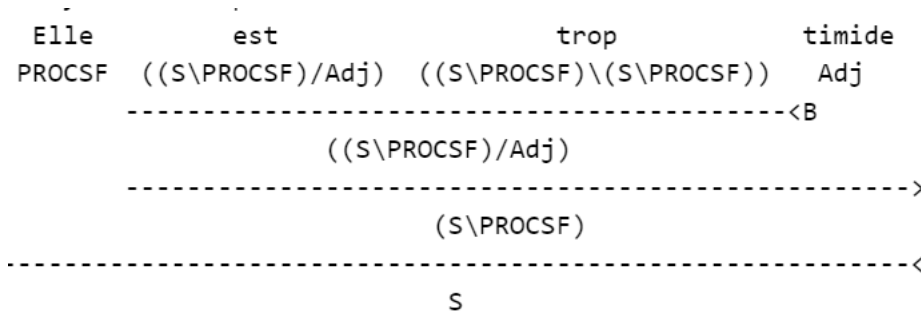


Figure 59 : Analyse CCG phrase 50

Cette dérivation montre que la phrase 50 « Elle est trop timide » est bien formée selon la GCCA. Chaque mot reçoit une catégorie syntaxique formelle. Les règles de combinaisons fonctionnelles (> et <) et de composition permettent de construire progressivement la structure de la phrase. L'analyse aboutit à la catégorie S, confirmant la validité syntaxique de la phrase.

IV.2) Conclusion

L'analyse des 50 phrases selon les règles de la Grammaire Catégorielle Combinatoire et Applicative(GCCA) a permis de confirmer, pour chaque énoncé bien formé, sa validité syntaxique à travers une dérivation structurée. En effet, chaque mot a été associé à une catégorie syntaxique formelle issue du lexique, et les règles de combinaison fonctionnelle standard (application avant > et application arrière <) ont permis d'aboutir à la catégorie S, confirmant la validité de la phrase.

Lorsque nécessaire, certaines dérivations ont impliqué des règles de changement de type ou des règles de composition, afin d'assurer la compatibilité entre catégories non directement combinables. Ces ajustements syntaxiques, restent conformes aux principes de la GCCA et permettent de traiter des constructions plus complexes.

V) Captures des résultats des cinquante phrases incorrectes

V.1) Introduction

Nous avons également intégré l'outil LanguageTool afin de détecter automatiquement les erreurs grammaticales ou lexicales dans une phrase. En cas de phrase incorrecte, le système fournit une analyse catégorielle partielle, accompagnée de suggestions de correction pondérées pour les mots mal utilisés ou inconnus du lexique. Les suggestions sont classées par pertinence selon leur position dans la liste proposée par l'outil.

Le programme affiche ensuite un tableau synthétique des erreurs détectées, indiquant leur position, le contexte, et les corrections possibles. Pour les mots inconnus (catégorie (Unknow)), des suggestions spécifiques sont proposées afin de guider l'apprenant.

Les figures ci-dessous illustrent le résultat obtenu avec l'analyse de 50 phrases que nous considérons comme incorrectes.

➤ **Phrase 1 : Nous pouvez manger de la soupe**

Analyse de la phrase : Nous pouvez manger de la soupe
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

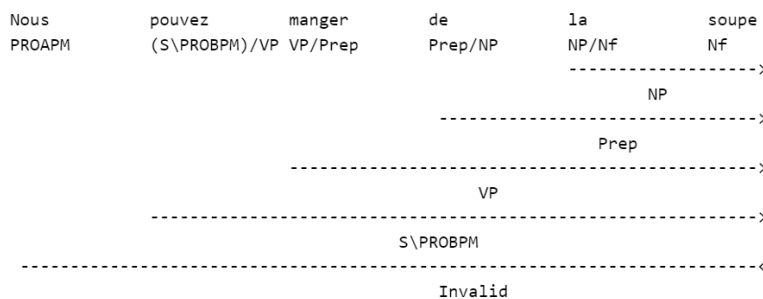


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Nous pouvez manger de la soupe	Phrase incorrecte	Position 5: Nous pouvez manger de la soupe -> pouvons : 100.00%

Figure 60 : Analyse CCG Phrase 51

La dérivation montre que la phrase 51 « Nous pouvez manger de la soupe » est incorrecte selon la GCCA.

L'erreur porte sur le mot « pouvez ». En effet, le pronom personnel « nous » ne peut pas être suivi d'un verbe conjugué à la deuxième personne du pluriel. Ce désaccord de personne rend la combinaison syntaxique incorrecte et bloque l'analyse. Le correcteur propose « pouvons » avec un score de 100 %, ce qui permettrait de corriger la phrase et de compléter la dérivation jusqu'à la catégorie S comme le montre la figure 61.

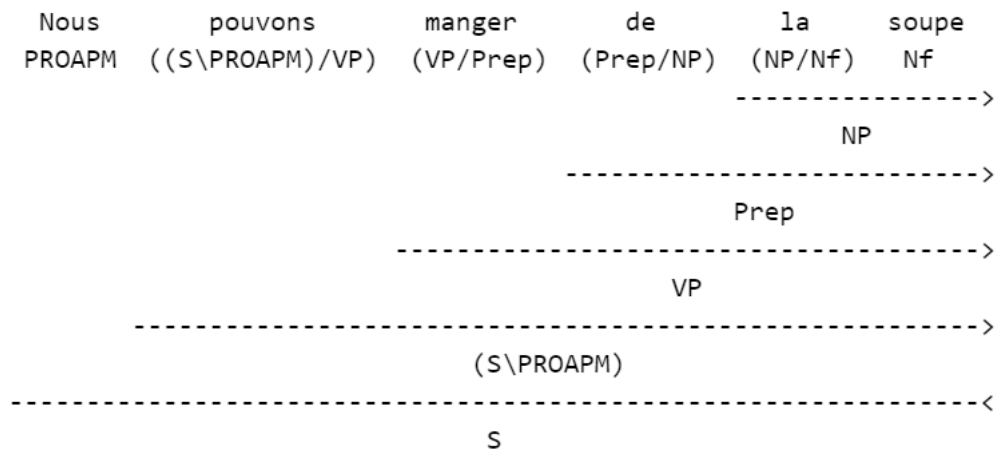


Figure 61 : Validation avec suggestion phrase 51

➤ Phrase 2 : Il parlent beaucoup

Analyse de la phrase : Il parlent beaucoup
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

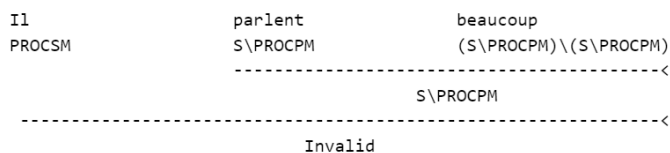


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il parlent beaucoup	Phrase incorrecte	Position 0: Il parlent beaucoup -> Ils parlent : 100.00%, Il parle : 50.00

Phrase	Statut	Erreurs et Suggestions
Il parlent beaucoup	Phrase incorrecte	Position 0: Il parlent beaucoup -> Ils parlent : 100.00%, Il parle : 50.00%

Figure 62 : Analyse CCG Phrase 52

La dérivation montre que la phrase 52 « Il parlent beaucoup » est incorrecte selon la GCCA. L'erreur porte sur le mot « parlent ». En effet, le pronom personnel « il », au singulier, ne peut pas être suivi d'un verbe conjugué à la troisième personne du pluriel. Ce désaccord de nombre rend la combinaison syntaxique incorrecte et bloque l'analyse. Le correcteur propose « ils parlent » avec un score de 100 % et « il parle » avec un score de 50 %, ce qui permettrait de corriger la phrase et de compléter la dérivation jusqu'à la catégorie S comme l'illustre la figure 63.

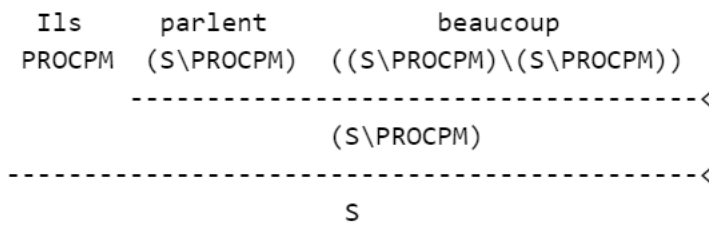


Figure 63 : Validation avec suggestion phrase 52

➤ Phrase 3 : Tu prend l'argent

Analyse de la phrase : Tu prend l'argent
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

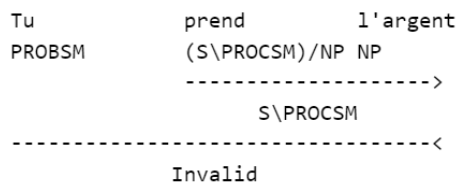


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Tu prend l'argent	Phrase incorrecte	Position 3: Tu prend l'argent -> prends : 100.00%

Figure 64 : Analyse CCG Phrase 53

La dérivation montre que la phrase 53 « Tu prend l'argent » est incorrecte selon la GCCA. L'erreur porte sur le mot « prend ». En effet, le pronom personnel « tu », 2ème personne au singulier, ne peut pas être suivi d'un verbe conjugué à la troisième personne du singulier. Ce désaccord de personne rend la combinaison syntaxique incorrecte et bloque l'analyse. Le correcteur propose « prends » avec un score de 100 %, ce qui permettrait de corriger la phrase et de compléter la dérivation jusqu'à la catégorie S, comme l'illustre la figure 65.

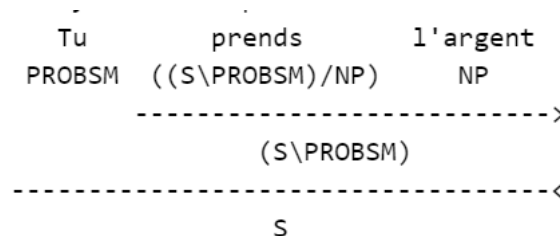


Figure 65 : Validation avec suggestion phrase 53

Phrase 4: La monsieur est en vie

Analyse de la phrase : La monsieur est en vie
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

La      monsieur  est      en      vie
NP/Nf   Nm        (S\NP)/Prep Prep/Nf  Nf
----->
Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
La monsieur est en vie	Phrase incorrecte	Position 0: La monsieur est en vie -> Le monsieur : 100.00%

Figure 66 : Analyse CCG Phrase 54

La dérivation montre que la phrase 54 « La monsieur est en vie » est incorrecte selon la GCCA.

L'erreur porte sur le déterminant « la ». En effet, le mot « monsieur » est un nom masculin et ne peut donc pas être précédé du déterminant féminin « la ». Ce désaccord de genre bloque la combinaison syntaxique attendue et interrompt l'analyse.

Le correcteur propose « Le monsieur » avec un score de 100 %, ce qui permettrait de corriger la phrase et de compléter la dérivation jusqu'à la catégorie S, comme l'illustre la figure 66.

```

Le      monsieur  est      en      vie
(NP/Nm) Nm        ((S\NP)/Prep) (Prep/Nf) Nf
----->
      NP
                                ----->
                                Prep
                                ----->
                                (S\NP)
-----<
      S
  
```

Figure 67 : Validation avec suggestion phrase 54

➤ Phrase 5 : Je vais rendre visit à ma maman

Analyse de la phrase : Je vais rendre visit à ma maman
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Je      vais      rendre  visit  à      ma      maman
PROASM (S\PROASM)/VP VP/NP  Unknow Prep/NP NP/Nf   Nf
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Je vais rendre visit à ma maman	Phrase incorrecte	Position 15: Je vais rendre visit à ma maman -> visite : 100.0

```

-----
| Erreurs et Suggestions
-----
| Position 15: Je vais rendre visit à ma maman -> visite : 100.00%, visité : 88.89%, visait : 77.78%, visita : 66.67%
-----
visait : 77.78%, visita : 66.67%, visio : 55.56%, visité : 44.44%, vissait : 33.33%, vissât : 22.22%, visât : 11.11%
-----

```

Figure 68 : Analyse CCG Phrase 55

La dérivation montre que la phrase 55 « Je vais rendre visit à ma maman » est incorrecte selon la GCCA.

L'erreur porte sur le mot « visit », qui est une forme incorrecte en français. Le mot attendu est « visite », un nom féminin, nécessaire à la construction correcte du syntagme verbal « rendre visite ». Ce mot mal orthographié bloque la dérivation syntaxique et interrompt l'analyse.

Le correcteur propose « visite » avec un score de 100 %, « visité » (88,88 %), « visait » (77,78 %), jusqu'à « visât » (11,11 %). La correction en « visite » permettrait de compléter la phrase et d'aboutir à la catégorie S comme l'illustre la figure 69.

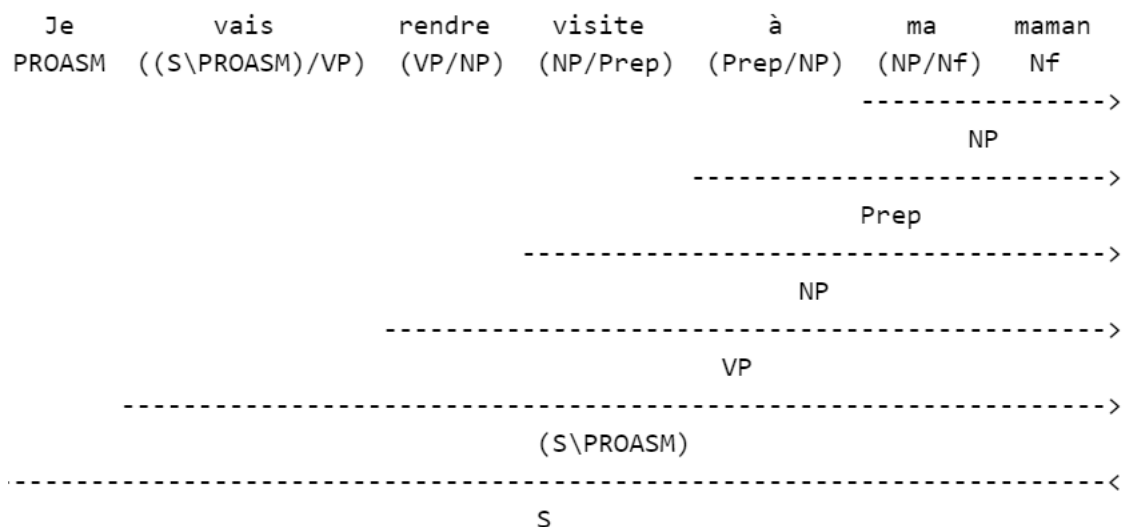


Figure 69 : Validation avec suggestion phrase 55

- Phrase 6 : Il sens tellement bon

Analyse de la phrase : Il sens tellement bon
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Il          sens          tellement          bon
PROCSM     (S\PROBSM)/Adj  (S\PROBSM)\(S\PROBSM) Adj
-----<B
              (S\PROBSM)/Adj
              ----->
                S\PROBSM
-----<
              Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il sens tellement bon	Phrase incorrecte	Position 3: Il sens tellement bon -> sent : 100.00%

Figure 70 : Analyse CCG Phrase 56

La dérivation montre que la phrase 56 « Il sens tellement bon » est incorrecte selon la GCCA.

L'erreur porte sur le mot « sens », qui correspond à la première ou à la deuxième personne du singulier, alors que le sujet « Il » exige une conjugaison à la troisième personne du singulier. Ce désaccord bloque la combinaison syntaxique attendue et interrompt l'analyse.

Le correcteur propose « sent » avec un score de 100 %, ce qui permettrait de corriger la phrase et de compléter la dérivation jusqu'à la catégorie S comme l'illustre la figure 71.

```

Il          sent          tellement          bon
PROCSM     ((S\PROCSM)/Adj)  ((S\PROCSM)\(S\PROCSM)) Adj
-----<B
              ((S\PROCSM)/Adj)
              ----->
                (S\PROCSM)
-----<
              S
  
```

Figure 71 : Validation avec suggestion phrase 56

Phrase 7 : La mariage est prévu pour demain

Analyse de la phrase : La mariage est prévu pour demain
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

La	mariage	est	prévu	pour	demain
NP/Nf	Nm	(S\NP)/(S\NP)	(S\NP)/Prep	Prep/ADV	ADV
----->					
Invalid					

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
La mariage est prévu pour demain	Phrase incorrecte	Position 0: La mariage est prévu pour demain -> Le mariage :
----->		
Erreurs et Suggestions		
----->		
Position 0: La mariage est prévu pour demain -> Le mariage : 100.00%, Les mariages : 50.00%		
----->		

Figure 72 : Analyse CCG Phrase 57

La dérivation montre que la phrase 57 « La mariage est prévu pour demain » est incorrecte selon la GCCA.

L'erreur porte sur le déterminant « la », utilisé devant le nom « mariage », qui est un nom masculin. Ce désaccord de genre bloque la combinaison syntaxique attendue entre le déterminant et le nom, ce qui interrompt l'analyse. Le correcteur propose « le mariage » avec un score de 100 % et « les mariages » avec un score de 50 %. La correction en « le mariage » permettrait de compléter la dérivation et d'aboutir à la catégorie S comme l'illustre la figure 73.

Le	mariage	est	prévu	pour	demain
(NP/Nm)	Nm	((S\NP)/(S\NP))	((S\NP)/Prep)	(Prep/ADV)	ADV
----->					
NP					
----->					
Prep					
----->					
(S\NP)					
----->					
(S\NP)					
----->					
S					
-----<					

Figure 73 : Validation avec suggestion phrase 57

➤ Phrase 8 : Le vie est tellement courte

Analyse de la phrase : Le vie est tellement courte
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Le         vie         est         tellement  courte
NP/Nm     Nf          (S\NP)/Adj  (S\NP)\(S\NP) Adj
----->
Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Le vie est tellement courte	Phrase incorrecte	Position 0: Le vie est tellement courte -> La vie : 100.00%,
Erreurs et Suggestions		
Position 0: Le vie est tellement courte -> La vie : 100.00%, Les vies : 50.00%		

Figure 74 : Analyse CCG Phrase 58

La dérivation montre que la phrase 58 « Le vie est tellement courte » est incorrecte selon la GCCA.

L'erreur porte sur le déterminant « le », utilisé devant le nom « vie », qui est un nom féminin. Ce désaccord de genre entre le déterminant et le nom empêche la formation correcte du syntagme nominal, ce qui bloque la dérivation syntaxique. Le correcteur propose « la vie » avec un score de 100 % et « les vies » avec un score de 50 %. La correction en « la vie » permettrait de compléter la dérivation et d'aboutir à la catégorie S comme l'illustre la figure 75.

```

La     vie     est     tellement  courte
(NP/Nf) Nf  ((S\NP)/Adj) ((S\NP)\(S\NP)) Adj
----->
      NP
          -----<B
              ((S\NP)/Adj)
                  ----->
                      (S\NP)
                          -----<
                              S
  
```

Figure 75 : Validation suggestion phrase 58

Phrase 9 : Je veux aller à le Mecque

Analyse de la phrase : Je veux aller à le Mecque
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

Je	veux	aller	à	le	Mecque
PROASM	(S\PROASM)/VP	VP/Prep	Prep/NP	NP/Nm	Nf
				----->	
				Invalid	

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Je veux aller à le Mecque	Phrase incorrecte	Position 16: Je veux aller à le Mecque -> la Mecque : 100.00%

Figure 76 : Analyse CCG Phrase 59

La dérivation montre que la phrase 59 « Je veux aller à le Mecque » est incorrecte selon la GCCA.

L'erreur porte sur le déterminant « le », utilisé devant le nom « Mecque », qui est un nom féminin propre. Ce désaccord de genre empêche la formation correcte du syntagme nominal introduit par la préposition « à », ce qui bloque la dérivation syntaxique. Le correcteur propose « la Mecque » avec un score de 100 %. La correction en « la Mecque » permettrait de compléter la dérivation et d'aboutir à la catégorie S comme l'illustre la figure 77.

Je	veux	aller	à	la	Mecque
PROASM	((S\PROASM)/VP)	(VP/Prep)	(Prep/NP)	(NP/Nf)	Nf
				----->	
				NP	
			----->		
			Prep		
		----->			
		VP			
	----->				
	(S\PROASM)				
-----<					
S					

Figure 77 : Validation suggestion phrase 59

- Phrase 10 : Je pense avoir la paludisme

Analyse de la phrase : Je pense avoir la paludisme
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Je           pense      avoir      la           paludisme
PROASM      (S\PROASM)/VP  VP/NP      NP/Nf        Nm
----->
Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Je pense avoir la paludisme	Phrase incorrecte	Position 15: Je pense avoir la paludisme -> le paludisme : 100.00%,

Erreurs et Suggestions
Position 15: Je pense avoir la paludisme -> le paludisme : 100.00%, les paludismes : 50.00%

Figure 78 : Analyse CCG Phrase 60

La dérivation montre que la phrase 60 « Je pense avoir la paludisme » est incorrecte selon la GCCA.

L'erreur porte sur le déterminant « la », utilisé devant le nom « paludisme », qui est un nom masculin. Ce désaccord de genre empêche la formation correcte du syntagme nominal, ce qui bloque la dérivation syntaxique. Le correcteur propose « le paludisme » avec un score de 100 % et « les paludismes » avec un score de 50 %. La correction en « le paludisme » permettrait de compléter la dérivation et d'aboutir à la catégorie S comme l'illustre la figure 79.

```

Je           pense      avoir      le           paludisme
PROASM      ((S\PROASM)/VP)  (VP/NP)   (NP/Nm)      Nm
----->
NP
----->
VP
----->
(S\PROASM)
-----<
S
  
```

Figure 79 : Validation suggestion phrase 60

➤ Phrase 11 : Son enfant est tout sa vie

Analyse de la phrase : Son enfant est tout sa vie
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Son      enfant  est      tout     sa      vie
NP/Nm   Nm       (S\NP)/NP NP/NPm  NPF/Nf  Nf
----->
                          NPf
----->
                          Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Son enfant est tout sa vie	Phrase incorrecte	Position 15: Son enfant est tout sa vie -> toute : 100.00%

Figure 80 : Analyse CCG Phrase 61

La dérivation montre que la phrase 61 « Son enfant est tout sa vie » est incorrecte selon la grammaire catégorielle combinatoire (CCG). L'erreur porte sur le déterminant « tout », qui ne s'accorde pas en genre avec le nom « vie », un nom féminin. Ce désaccord empêche la formation correcte du syntagme nominal, bloquant ainsi la dérivation syntaxique. Le correcteur propose « toute » avec un score de 100 %. La correction en « toute sa vie » permettrait de compléter la dérivation et d'aboutir à la catégorie S, signalant une phrase syntaxiquement correcte comme l'illustre la figure 81.

```

      Son      enfant  est      toute     sa      vie
      (NP/Nm)   Nm       ((S\NP)/NP) (NP/NPf)  (NPf/Nf)  Nf
----->
              NP
                                     ----->
                                     NPf
----->
                                     NP
----->
              (S\NP)
-----<
              S
  
```

Figure 81 : Validation suggestion phrase 61

- Phrase 12 : Vous avez peur pour son vie

Analyse de la phrase : Vous avez peur pour son vie
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

Vous	avez	peur	pour	son	vie
PROBPM	(S\PROBPM)/ADV	ADV/Prep	Prep/NP	NP/Nm	Nf
				----->	
				Invalid	

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Vous avez peur pour son vie	Phrase incorrecte	Position 20: Vous avez peur pour son vie -> sa : 100.00%

Figure 82 : Analyse CCG Phrase 62

La dérivation montre que la phrase 62 « Vous avez peur pour son vie » est incorrecte selon la GCCA.

L'erreur porte sur le déterminant « son », qui ne s'accorde pas en genre avec le nom « vie », un nom féminin. Ce désaccord empêche la formation correcte du syntagme nominal, ce qui bloque la dérivation syntaxique. Le correcteur propose « sa » avec un score de 100 %. La correction en « sa vie » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 83.

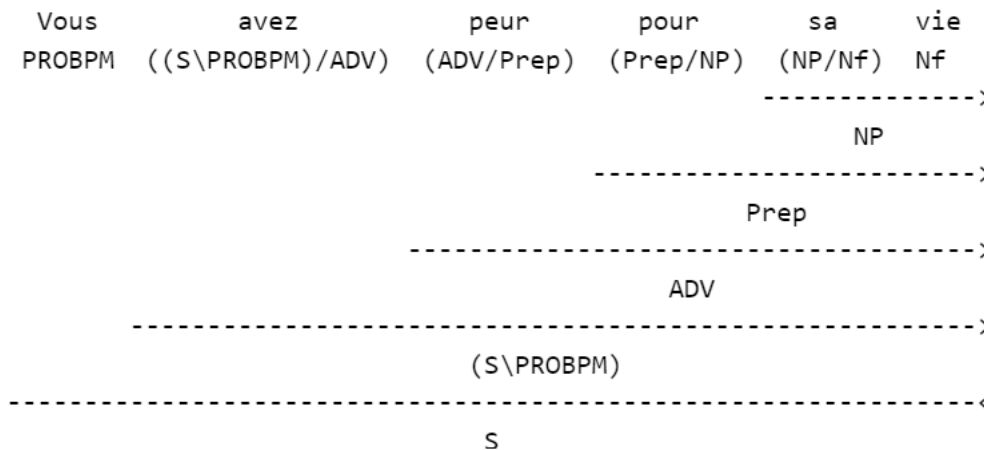


Figure 83 : Validation suggestion phrase 62

➤ Phrase 13 : Sa père est fier d'elle

Analyse de la phrase : Sa père est fier d'elle
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Sa      père      est      fier      d'elle
NP/Nf   Nm        (S\NP)/Adj Adj/NP   NP
----->
      Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Sa père est fier d'elle	Phrase incorrecte	Position 0: Sa père est fier d'elle -> Son père : 100.00%,
Statut	Erreurs et Suggestions	
Phrase incorrecte	Position 0: Sa père est fier d'elle -> Son père : 100.00%, Ses pères : 50.00%	

Figure 84 : Analyse CCG Phrase 63

La dérivation montre que la phrase 63 « Sa père est fier d'elle » est incorrecte selon la GCCA. L'erreur porte sur le déterminant « sa », qui ne s'accorde pas en genre avec le nom « père », un nom masculin. Ce désaccord empêche la formation correcte du syntagme nominal, ce qui bloque la dérivation syntaxique.

Le correcteur propose « son père » avec un score de 100 % et « ses pères » avec un score de 50 %.

La correction en « son père » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 85.

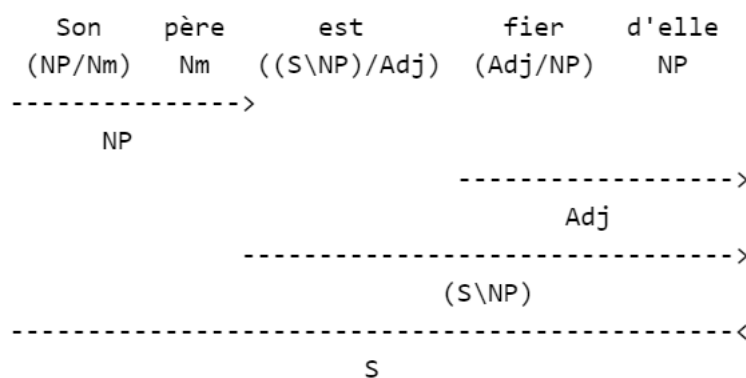


Figure 85 : Validation suggestion phrase 63

➤ Phrase 14 : Ils pense nous duper

Analyse de la phrase : Ils pense nous duper
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

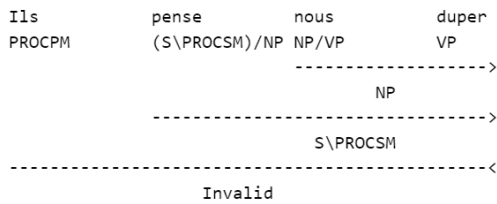


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Ils pense nous duper	Phrase incorrecte	Position 0: Ils pense nous duper -> Il pense : 100.00%, Ils pensent : 50.

Erreurs et Suggestions
Position 0: Ils pense nous duper -> Il pense : 100.00%, Ils pensent : 50.00%

Figure 86 : Analyse CCG Phrase 64

La dérivation montre que la phrase 64 « Ils pense nous duper » est incorrecte selon la GCCA. L'erreur porte sur l'accord entre le sujet « ils » (3^e personne du pluriel) et le verbe « pense », conjugué à la 3^e personne du singulier. Ce désaccord sujet-verbe empêche la formation correcte du syntagme verbal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « il pense » avec un score de 100 %, et « ils pensent » avec un score de 50 %.

La correction en « Il pense nous duper » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 87.

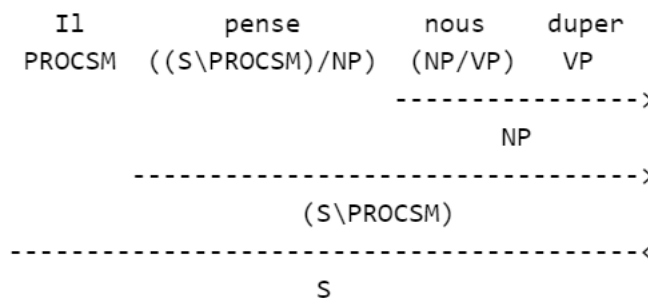


Figure 87 : Validation suggestion phrase 64

- Phrase 15 : Il veut remplir sa bombonne

Analyse de la phrase : Il veut remplir sa bombonne
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

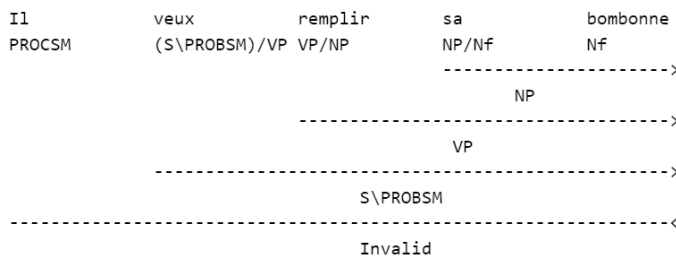


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il veut remplir sa bombonne	Phrase incorrecte	Position 3: Il veut remplir sa bombonne -> veut : 100.00%

Figure 88 : Analyse CCG Phrase 65

La dérivation montre que la phrase 64 « Il veut remplir sa bombonne » est incorrecte selon la grammaire catégorielle combinatoire (CCG). L'erreur porte sur l'accord entre le sujet « Il » (3^e personne du singulier) et le verbe « veut », conjugué à la 2^e personne du singulier ou à la première personne. Ce désaccord sujet-verbe empêche la formation correcte du syntagme verbal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « veut » avec un score de 100 %.

La correction en « Il veut remplir sa bombonne » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 89.

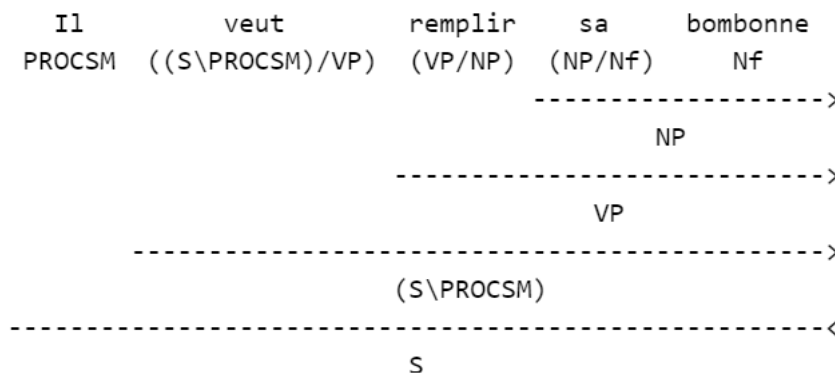


Figure 89 : Validation suggestion phrase 65

➤ Phrase 16 : Tu as trente enfant

Analyse de la phrase : Tu as trente enfant
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

Tu	as	trente	enfant
PROBSM	(S\PROBSM)/NP	NP/Ns	Nm
		----->	
		Invalid	

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Tu as trente enfant	Phrase incorrecte	Position 6: Tu as trente enfant -> trente enfants : 100.00%

Figure 90 : Analyse CCG Phrase 66

La dérivation montre que la phrase 66 « Tu as trente enfant » est incorrecte selon la GCCA.

L'erreur porte sur le nom « enfant », utilisé au singulier alors qu'il est précédé du numéral « trente », qui exige un nom au pluriel. Ce désaccord empêche la formation correcte du syntagme nominal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « trente enfants » avec un score de 100 %.

La correction en « Tu as trente enfants » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 91.

Tu	as	trente	enfants
PROBSM	((S\PROBSM)/NP)	(NP/Ns)	Ns
		----->	
		NP	
	----->		
	(S\PROBSM)		
-----<			
S			

Figure 91 : Validation suggestion phrase 66

- Phrase 17 : Je sens que je va tomber

Analyse de la phrase : Je sens que je va tomber
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

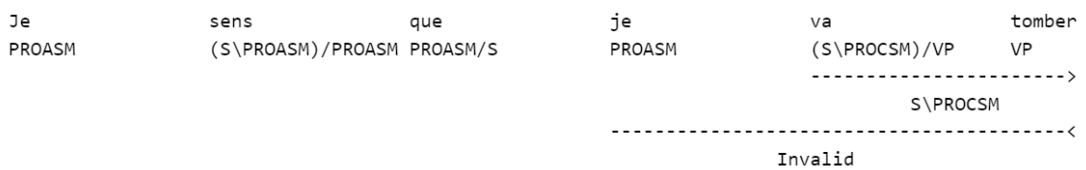


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Je sens que je va tomber	Phrase incorrecte	Position 15: Je sens que je va tomber -> vais : 100.00%

Figure 92 : Analyse CCG Phrase 67

La dérivation montre que la phrase 67 « Je sens que je va tomber » est incorrecte selon la GCCA.

L'erreur porte sur le verbe « va », conjugué à la 3^e personne du singulier, alors que le sujet « je » exige la forme « vais », conjuguée à la 1^{re} personne du singulier. Ce désaccord sujet-verbe empêche la formation correcte du syntagme verbal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « vais » avec un score de 100 %.

La correction en « Je sens que je vais tomber » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 93.

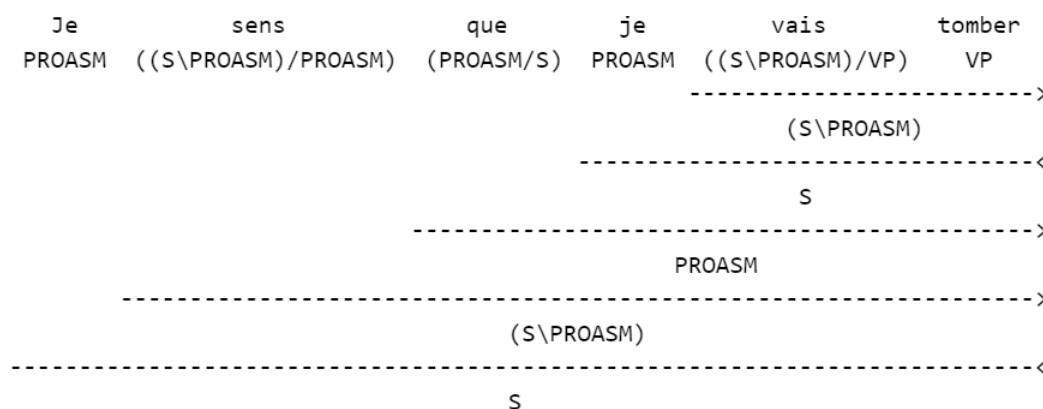


Figure 93 : Validation suggestion phrase 67

- Phrase 18 : Son fils aime Halloween

Analyse de la phrase : Son fils aiment Halloween
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

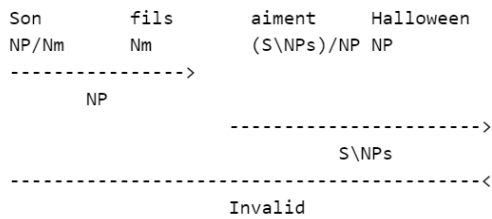


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Son fils aiment Halloween	Phrase incorrecte	Position 9: Son fils aiment Halloween -> aime : 100.00%

Figure 94 : Analyse CCG Phrase 68

La dérivation montre que la phrase 68 « Son fils aiment Halloween » est incorrecte selon la GCCA.

L'erreur porte sur le verbe « aiment », conjugué à la 3^e personne du pluriel, alors que le sujet « son fils » est au singulier. Ce désaccord sujet-verbe empêche la formation correcte du syntagme verbal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « aime » avec un score de 100 %.

La correction en « Son fils aime Halloween » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 95.

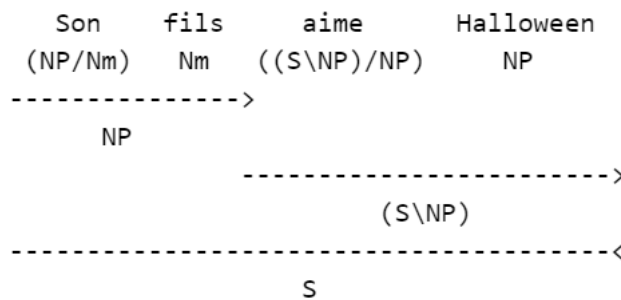


Figure 95 : Validation suggestion phrase 68

- Phrase 19 : Tu as reçu un ventilater

Analyse de la phrase : Tu as reçu un ventilater
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

Tu	as	reçu	un	ventilater
PROBSM	(S\PROBSM)/(S\PROBSM)	(S\PROBSM)/NP	NP/Nm	Unknow

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Tu as reçu un ventilater	Phrase incorrecte	Position 14: Tu as reçu un ventilater -> ventilateur : 100.00%,
Erreurs et Suggestions		
Position 14: Tu as reçu un ventilater -> ventilateur : 100.00%, ventilâtes : 66.67%, ventila ter : 33.33%		

Figure 96 : Analyse CCG Phrase 69

La dérivation montre que la phrase 69 « Tu as reçu un ventilater » est incorrecte selon la grammaire catégorielle combinatoire (CCG). L'erreur porte sur le nom « ventilater », qui est une faute d'orthographe du mot « ventilateur ». Ce mot mal formé empêche la reconnaissance correcte du syntagme nominal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « ventilateur » avec un score de 100 %, « ventilâtes » avec un score de 66,67 %, et « ventila ter » avec un score de 33,33 %.

La correction en « Tu as reçu un ventilateur » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 97.

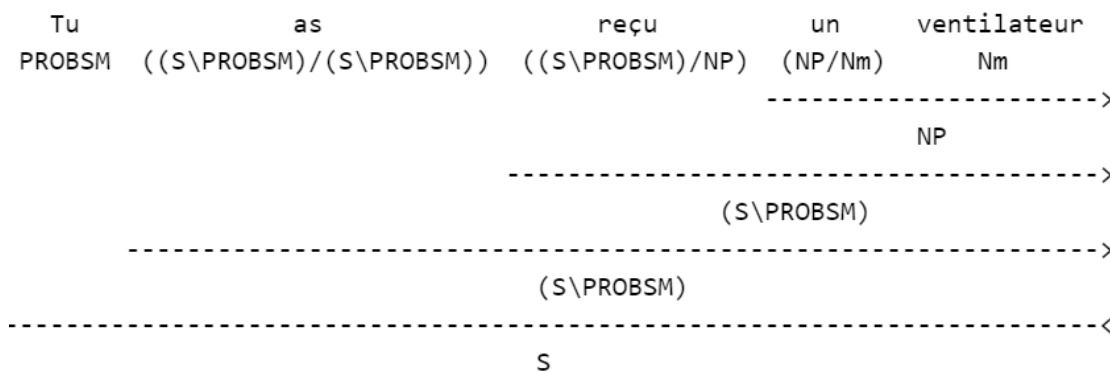


Figure 97 : Validation suggestion phrase 69

- Phrase 20 : Il m'as donné un nouvel avenir

Analyse de la phrase : Il m'as donné un nouvel avenir
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

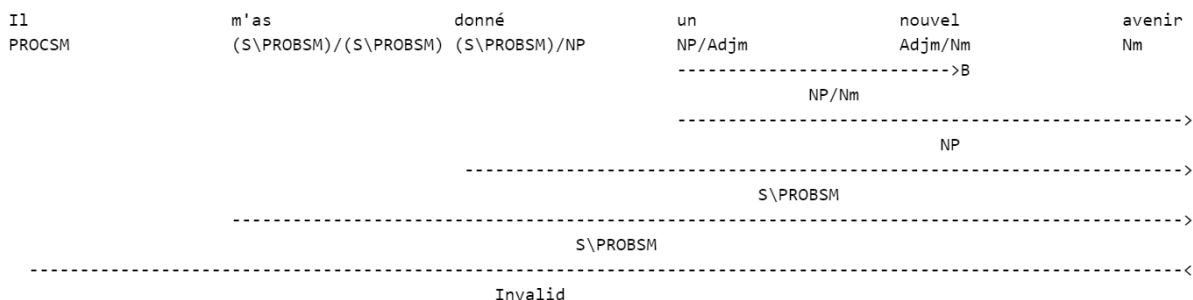


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il m'as donné un nouvel avenir	Phrase incorrecte	Position 5: Il m'as donné un nouvel avenir -> a : 100.00%

Figure 98 : Analyse CCG Phrase 70

La dérivation montre que la phrase 70 « Il m'as donné un nouvel avenir » est incorrecte selon la GCCA.

L'erreur porte sur la forme verbale « m'as », où « as » (2^e personne du singulier) est incorrectement utilisé avec le sujet « il » (3^e personne du singulier). Ce désaccord sujet-verbe empêche la formation correcte du syntagme verbal, ce qui bloque la dérivation syntaxique.

Le correcteur propose « a » avec un score de 100 %.

La correction en « Il m'a donné un nouvel avenir » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 99.

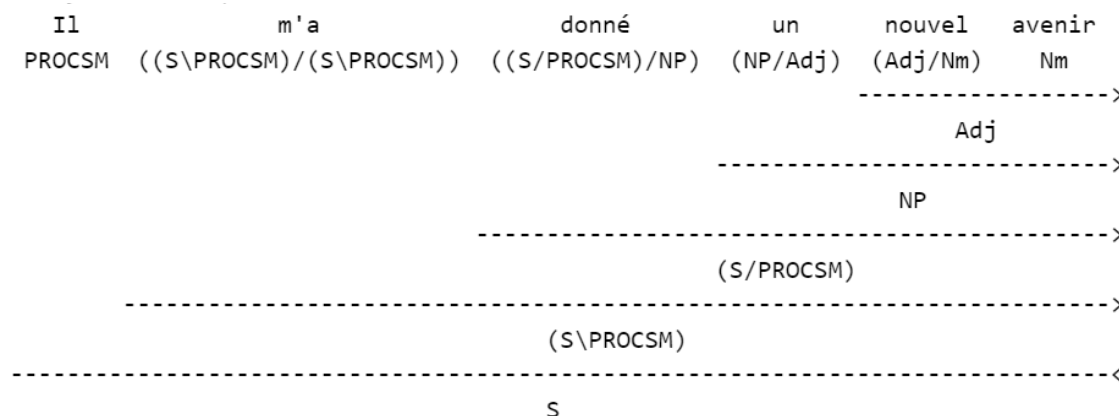


Figure 99 : Validation suggestion phrase 70

➤ Phrase 21 : Je dois assister ma frère

Analyse de la phrase : Je dois assister ma frère
Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Je          dois      assister   ma          frère
PROASM     (S\PROASM)/VP VP/NP      NP/Nf      Nm
----->
Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Je dois assister ma frère	Phrase incorrecte	Position 17: Je dois assister ma frère -> mon frère : 100.00%,

Erreurs et Suggestions

Position 17: Je dois assister ma frère -> mon frère : 100.00%, mes frères : 50.00%

Figure 100 : Analyse CCG Phrase 71

La dérivation montre que la phrase 71 « Je dois assister ma frère » est incorrecte selon la GCCA. L'erreur porte sur le déterminant possessif « ma », qui ne s'accorde pas en genre avec le nom « frère », un nom masculin. Ce désaccord de genre empêche la formation correcte du syntagme nominal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « mon frère » avec un score de 100 %, et « mes frères » avec un score de 50 %.

La correction en « Je dois assister mon frère » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 101.

```

je          dois      assister   mon      frère
PROASM     ((S\PROASM)/VP) (VP/NP)  (NP/Nm)  Nm
----->
NP
----->
VP
----->
(S\PROASM)
-----<
S
  
```

Figure 101 : Validation suggestion phrase 71

➤ Phrase 22 : Je respectes mon ami

Analyse de la phrase : Je respectes mon ami
Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Je      respectes  mon      ami
PROASM  (S\PROBSM)/NP NP/Nm    Nm
----->
                        NP
----->
                    S\PROBSM
-----<
Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Je respectes mon ami	Phrase incorrecte	Position 3: Je respectes mon ami -> respecte : 100.00%

Figure 102 : Analyse CCG Phrase 72

La dérivation montre que la phrase 72 « Je respectes mon ami » est incorrecte selon la GCCA. L'erreur porte sur la forme verbale « respectes », conjuguée à la 2^e personne du singulier, alors que le sujet « je » est à la 1^{re} personne du singulier. Ce désaccord sujet-verbe empêche la formation correcte du syntagme verbal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « respecte » avec un score de 100 %.

La correction en « Je respecte mon ami » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 103.

```

Je      respecte  mon      ami
PROASM  ((S\PROASM)/NP) (NP/Nm) Nm
----->
                        NP
----->
                    (S\PROASM)
-----<
                        S
  
```

Figure 103 : Validation suggestion phrase 72

➤ Phrase 23 : Le fête du mouton approche

Analyse de la phrase : Le fête du mouton approche
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Le      fête    du      mouton  approche
NP/Nm   Nf/NP    NP/Nm   Nm      (S\NP)
----->B
      Nf/Nm
----->
      Nf
----->
      Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Le fête du mouton approche	Phrase incorrecte	Position 0: Le fête du mouton approche -> La fête : 100.00%,
Erreurs et Suggestions		
Position 0: Le fête du mouton approche -> La fête : 100.00%, Les fêtes : 50.00%		

Figure 104 : Analyse CCG Phrase 73

La dérivation montre que la phrase 73 « Le fête du mouton approche » est incorrecte selon la grammaire catégorielle combinatoire (CCG). L'erreur porte sur le déterminant « le », qui ne s'accorde pas en genre avec le nom « fête », un nom féminin. Ce désaccord empêche la formation correcte du syntagme nominal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « la fête » avec un score de 100 % et « les fêtes » avec un score de 50 %.

La correction en « La fête du mouton approche » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 105.

```

La      fête    du      mouton  approche
(NP/Nf) (Nf/NP)  (NP/Nm)   Nm      (S\NP)
----->
      NP
----->
      Nf
----->
      NP
-----<
      S
  
```

Figure 105 : Validation suggestion phrase 73

- Phrase 24 : Il aimais faire du bien

Analyse de la phrase : Il aimais faire du bien
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

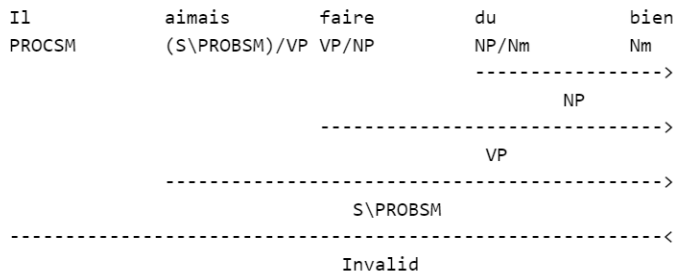


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il aimais faire du bien	Phrase incorrecte	Position 3: Il aimais faire du bien -> aimait : 100.00%

Figure 106 : Analyse CCG Phrase 74

La dérivation montre que la phrase 74 « Il aimais faire du bien » est incorrecte selon la GCCA. L'erreur porte sur le verbe « aimais », qui est conjugué à la première personne du singulier (imparfait) alors que le sujet est « Il », à la troisième personne du singulier. Ce désaccord sujet-verbe empêche la formation correcte du syntagme verbal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « aimait » avec un score de 100 %.

La correction en « Il aimait faire du bien » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 107.

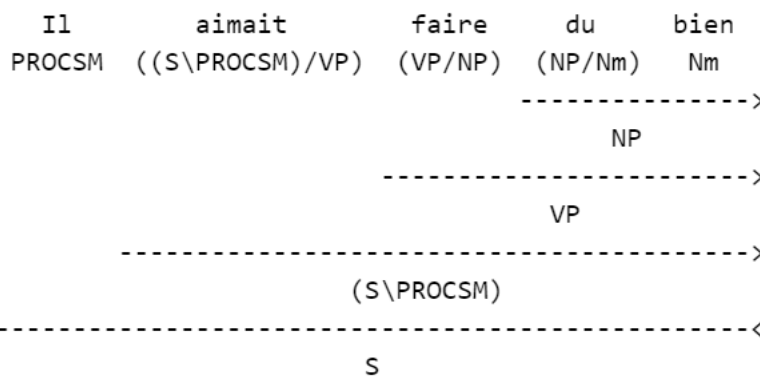


Figure 107 : Validation suggestion phrase 74

➤ Phrase 25 : Mon sœur a grandi

Analyse de la phrase : Mon sœur a grandi
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Mon      sœur      a      grandi
NP/Nm   Nf        (S\NP)/(S\NP) (S\NP)
----->
      Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Mon sœur a grandi	Phrase incorrecte	Position 0: Mon sœur a grandi -> Ma : 100.00%

Figure 108 : Analyse CCG Phrase 75

La dérivation montre que la phrase 75 « Mon sœur a grandi » est incorrecte selon la GCCA. L'erreur porte sur le déterminant possessif « mon », qui ne s'accorde pas en genre avec le nom « sœur », un nom féminin. En français, on utilise « ma » devant un nom féminin commençant par une consonne. Ce désaccord empêche la formation correcte du syntagme nominal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « ma » avec un score de 100 %.

La correction en « Ma sœur a grandi » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 109.

```

      Ma      sœur      a      grandi
      (NP/Nf)  Nf        ((S\NP)/(S\NP)) (S\NP)
----->
      NP
                                ----->
                                (S\NP)
-----<
                                S
  
```

Figure 109 : Validation suggestion phrase 75

- Phrase 26 : Nous avons un secrets

Analyse de la phrase : Nous avons un secrets
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Nous      avons      un      secrets
PROAPM    (S\PROAPM)/NP  NP/Nm    Nms
----->
Invalid
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Nous avons un secrets	Phrase incorrecte	Position 11: Nous avons un secrets -> des secrets : 100.00%, un secret : 75.00%, une secrète : 50.00%, des secrètes : 25.00%

Figure 110 : Analyse CCG Phrase 76

La dérivation montre que la phrase 76 « Nous avons un secrets » est incorrecte selon la GCCA. L'erreur porte sur le déterminant indéfini « un », qui ne s'accorde pas en nombre avec le nom « secrets », un nom pluriel. En français, on utilise « des » devant un nom pluriel indéfini. Ce désaccord empêche la formation correcte du syntagme nominal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « des secrets » avec un score de 100 %, « un secret » avec un score de 50 %, et « des secrètes » avec un score de 25 %.

La correction en « Nous avons des secrets » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 111.

```

      Nous      avons      un      secret
      PROAPM    ((S\PROAPM)/NP)  (NP/Nm)    Nm
      ----->
                        NP
      ----->
                    (S\PROAPM)
      -----<
                        S
  
```

Figure 111 : Validation suggestion phrase 76

➤ Phrase 27 : Grand-mère préparent une soupe

Analyse de la phrase : Grand-mère préparent une soupe
Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

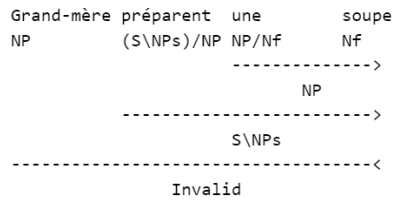


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Grand-mère préparent une soupe	Phrase incorrecte	Position 11: Grand-mère préparent une soupe -> prépare : 100.00%

Figure 112 : Analyse CCG Phrase 77

La dérivation montre que la phrase 77 « Grand-mère préparent une soupe » est incorrecte selon la GCCA. L'erreur porte sur le verbe « préparent », conjugué à la 3^e personne du pluriel, alors que le sujet « Grand-mère » est au singulier. Ce désaccord sujet-verbe empêche la formation correcte du syntagme verbal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « prépare » avec un score de 100 %.

La correction en « Grand-mère prépare une soupe » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 113.

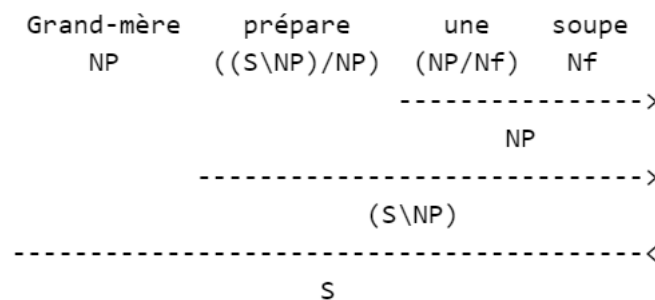


Figure 113 : Validation suggestion phrase 77

➤ Phrase 28 : Je dégustes de la glace

Analyse de la phrase : Je dégustes de la glace
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

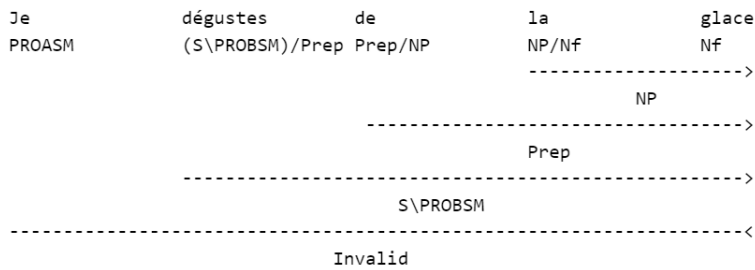


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Je dégustes de la glace	Phrase incorrecte	Position 3: Je dégustes de la glace -> déguste : 100.00%

Figure 114 : Analyse CCG Phrase 78

La dérivation montre que la phrase 78 « Je dégustes de la glace » est incorrecte selon la GCCA. L'erreur porte sur le verbe « dégustes », conjugué à la 2^e personne du singulier, alors que le sujet « Je » exige la 1^{re} personne du singulier, soit « déguste ». Ce désaccord sujet-verbe empêche la formation correcte du syntagme verbal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « déguste » avec un score de 100%.

La correction en « Je déguste de la glace » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 115.

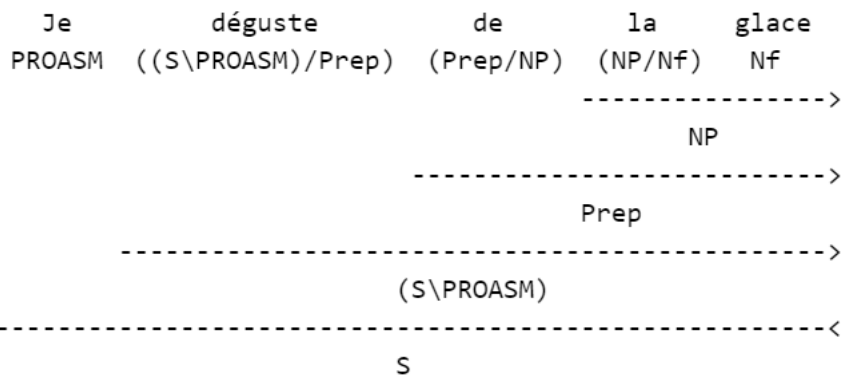


Figure 115 : Validation suggestion phrase 78

- Phrase 29 : Elle veut la bague au doigt

Analyse de la phrase : Elle veut la bague au doigt
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

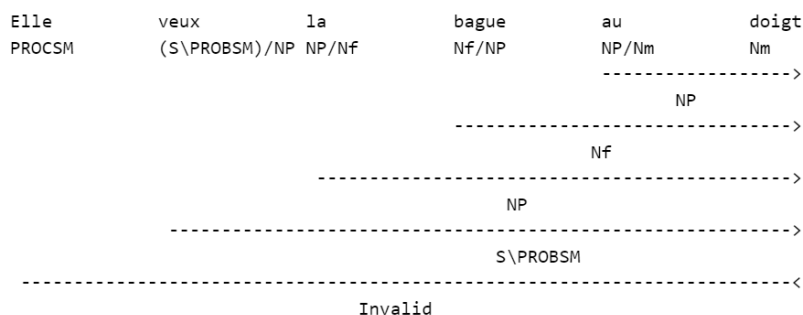


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Elle veut la bague au doigt	Phrase incorrecte	Position 5: Elle veut la bague au doigt -> veut : 100.00%

Figure 116 : Analyse CCG Phrase 79

La dérivation montre que la phrase 79 « Elle veut la bague au doigt » est incorrecte selon la GCCA.

L'erreur porte sur le verbe « veut », conjugué à la 2^e personne du singulier, alors que le sujet « Elle », à la 3^e personne du singulier, exige la forme « veut ». Ce désaccord sujet-verbe empêche la formation correcte du syntagme verbal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « veut » avec un score de 100 %.

La correction en « Elle veut la bague au doigt » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 117.

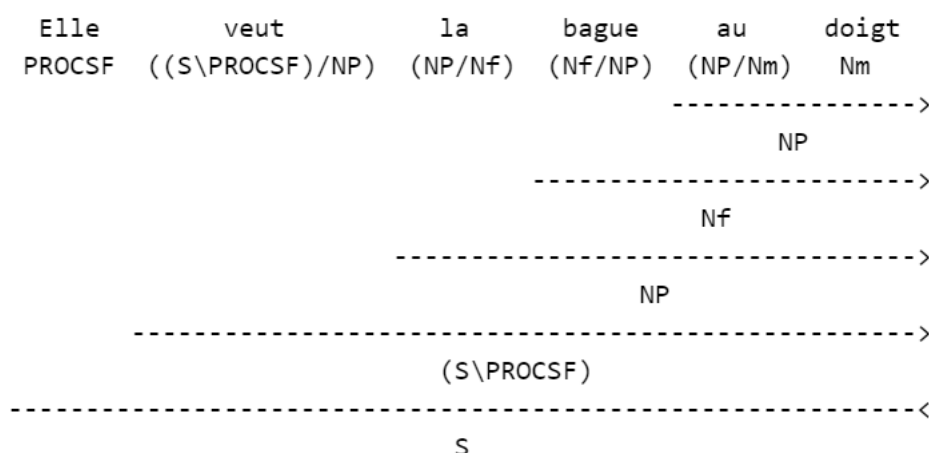


Figure 117 : Validation suggestion phrase 79

➤ Phrase 30 : Tout âme goûtera à la mort

Analyse de la phrase : Tout âme goûtera à la mort
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```
Tout      âme      goûtera   à      la      mort
NP/Nm    Nf      (S\NP)/Prep Prep/NP  NP/Nf    Nf
----->
      Invalid
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Tout âme goûtera à la mort	Phrase incorrecte	Position 0: Tout âme goûtera à la mort -> Toute âme : 100.00%,
Erreurs et Suggestions		
Position 0: Tout âme goûtera à la mort -> Toute âme : 100.00%, Toutes âmes : 50.00%		

Figure 118 : Analyse CCG Phrase 80

La dérivation montre que la phrase 80 « Tout âme goûtera à la mort » est incorrecte selon la GCCA.

L'erreur porte sur le déterminant « Tout », qui est au masculin singulier, alors que le nom « âme » est féminin singulier. En français, on doit utiliser « Toute » devant un nom féminin singulier. Ce désaccord empêche la formation correcte du syntagme nominal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « Toute âme » avec un score de 100 % et « Toutes âmes » avec un score de 50 % également.

La correction en « Toute âme goûtera à la mort » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 119.

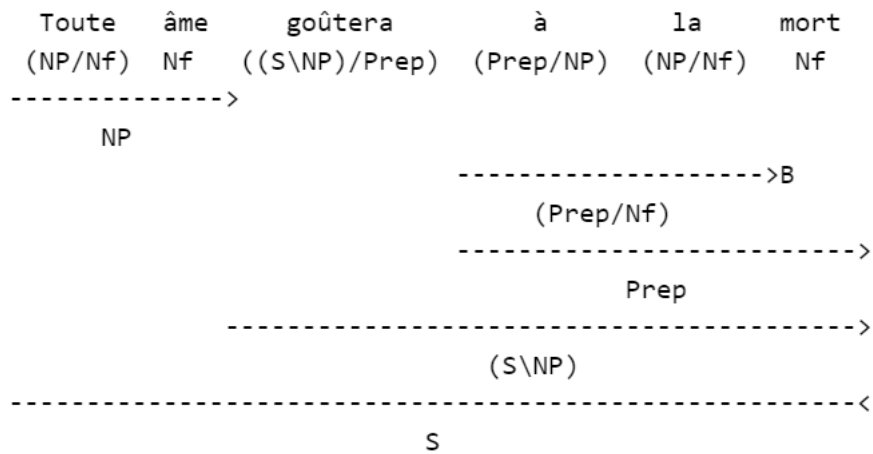


Figure 119: Validation suggestion phrase 80

➤ Phrase 31 : Il pleurent

Analyse de la phrase : Il pleurent
Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Il      pleurent
PROCSM S\PROCPM
-----<
      Invalid

```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il pleurent	Phrase incorrecte	Position 0: Il pleurent -> Ils pleurent : 100.00%, Il pleure : 50.00%

Figure 120 : Analyse CCG Phrase 81

La dérivation montre que la phrase 81 « Il pleurent » est incorrecte selon la GCCA. L'erreur porte sur un désaccord sujet-verbe : le pronom « Il » est au singulier, tandis que le verbe « pleurent » est conjugué à la troisième personne du pluriel. Pour que la dérivation soit valide, il faut que le sujet et le verbe s'accordent en nombre.

Le correcteur propose « Ils pleurent » avec un score de 100 % et « Il pleure » avec un score de 50 %.

La correction en « Ils pleurent » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 121.

```

      Ils      pleurent
      PROCPM  (S\PROCPM)
-----<
      S

```

Figure 121 : Validation suggestion phrase 81

➤ Phrase 32 : Aujourd'hui elle parlent

Analyse de la phrase : Aujourd'hui elle parlent
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Aujourd'hui elle      parlent
S/S      PROCsf      S\PROCPF
-----<
      Invalid

```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Aujourd'hui elle parlent	Phrase incorrecte	Position 12: Aujourd'hui elle parlent -> elles parlent : 100.00%,
Erreurs et Suggestions		
Position 12: Aujourd'hui elle parlent -> elles parlent : 100.00%, elle parle : 50.00%		

Figure 122 : Analyse CCG Phrase 82

La dérivation montre que la phrase 82 « Aujourd'hui elle parlent » est incorrecte selon la GCCA.

L'erreur porte sur un désaccord sujet-verbe : le pronom « elle » est au singulier, tandis que le verbe « parlent » est conjugué à la troisième personne du pluriel. Pour que la dérivation soit valide, il faut que le sujet et le verbe s'accordent en nombre.

Le correcteur propose « elles parlent » avec un score de 100 % et « elle parle » avec un score de 50 %.

La correction en « Ils pleurent » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 123.

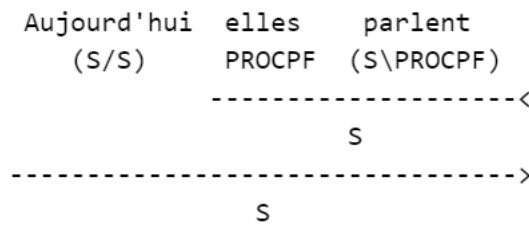


Figure 123 : Validation suggestion phrase 82

➤ Phrase 33 : Personne n'aiment la prison

Analyse de la phrase : Personne n'aiment la prison
Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

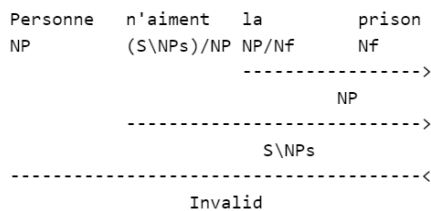


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Personne n'aiment la prison	Phrase incorrecte	Position 11: Personne n'aiment la prison -> aime : 100.00%

Figure 124 : Analyse CCG Phrase 83

La dérivation montre que la phrase 83 « Personne n'aiment la prison » est incorrecte selon la grammaire catégorielle combinatoire (CCG). L'erreur porte sur un désaccord sujet-verbe : le mot « Personne » est un sujet singulier, mais le verbe « aiment » est conjugué à la troisième personne du pluriel. Ce désaccord empêche la formation correcte du syntagme verbal, bloquant ainsi la dérivation syntaxique.

Le correcteur propose « aime » avec un score de 100 %.

La correction en « Personne n'aime la prison » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 125.

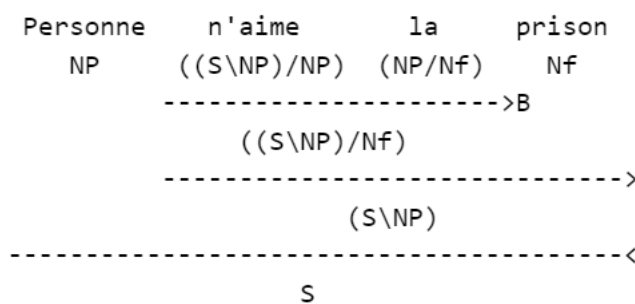


Figure 125 : Validation suggestion phrase 83

➤ Phrase 34: Il est monté dan sa chambre

Analyse de la phrase : Il est monté dan sa chambre
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

Il	est	monté	dan	sa	chambre
PROCSM	(S\PROCSM)/(S\PROCSM)	(S\PROCSM)/Prep	Unknow	NP/Nf	Nf

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il est monté dan sa chambre	Phrase incorrecte	Position 13: Il est monté dan sa chambre -> dans : 100.00%

Figure 126 : Analyse CCG Phrase 84

La dérivation montre que la phrase 84 « Il est monté dan sa chambre » est incorrecte selon la GCCA.

L'erreur porte sur la préposition mal orthographiée « dan », qui devrait être « dans ». Cette erreur empêche la bonne interprétation du complément circonstanciel de lieu, ce qui bloque la dérivation syntaxique à ce niveau.

Le correcteur propose « dans » avec un score de 100 %.

La correction en « Il est monté dans sa chambre » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 127.

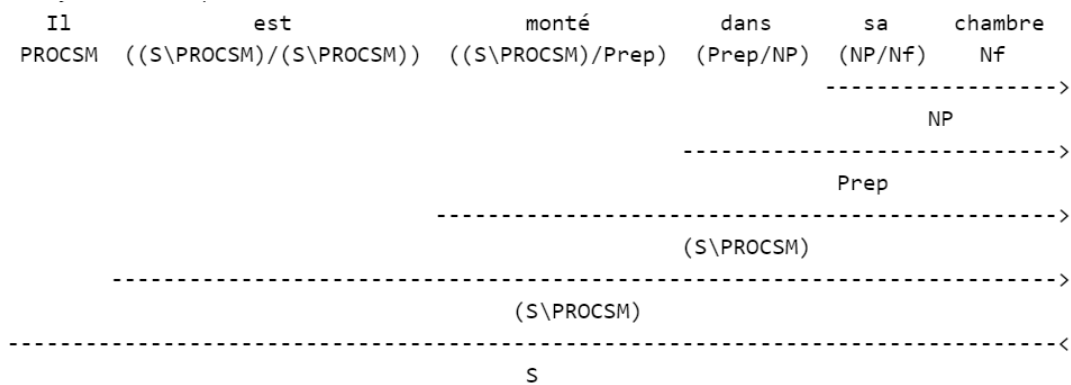


Figure 127 : Validation suggestion phrase 84

➤ Phrase 35 : Il m'as choisi

Analyse de la phrase : Il m'as choisi

Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

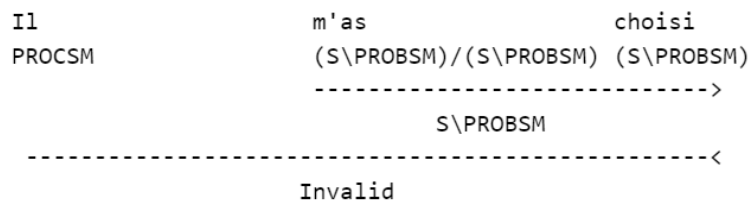


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il m'as choisi	Phrase incorrecte	Position 5: Il m'as choisi -> a : 100.00%

Figure 128 : Analyse CCG Phrase 85

La dérivation montre que la phrase 85 « Il m'as choisi » est incorrecte selon la grammaire GCCA.

L'erreur porte sur le verbe auxiliaire « as », qui est incorrectement conjugué à la 2^e personne du singulier, alors que le sujet « Il » est à la 3^e personne du singulier. La forme correcte est « a », ce qui permet de former correctement le temps composé.

Le correcteur propose « a » avec un score de 100 %.

La correction en « Il m'a choisi » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 129.

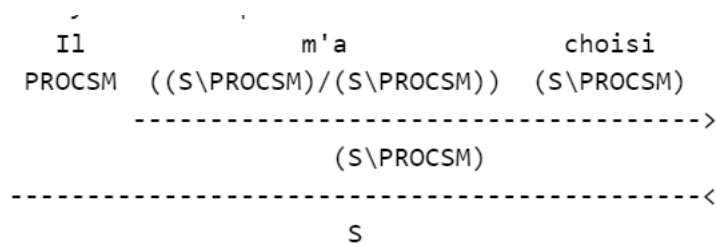


Figure 129 : Validation suggestion phrase 85

➤ Phrase 36 : Fatou a une diplôme

Analyse de la phrase : Fatou a une diplôme
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

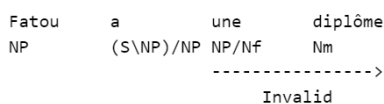


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Fatou a une diplôme	Phrase incorrecte	Position 8: Fatou a une diplôme -> un diplôme : 100.00%, des diplômes : 50

Erreurs et Suggestions
Position 8: Fatou a une diplôme -> un diplôme : 100.00%, des diplômes : 50.00%

Figure 130 : Analyse CCG Phrase 86

La dérivation montre que la phrase 86 « Fatou a une diplôme » est incorrecte selon la GCCA. L'erreur porte sur le déterminant « une », utilisé devant « diplôme », un nom masculin. En français, le déterminant doit s'accorder en genre avec le nom. Il faut donc utiliser « un » au lieu de « une ».

Le correcteur propose « un diplôme » avec un score de 100 %, et « des diplômes » avec un score de 50 %.

La correction en « Fatou a un diplôme » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 131.

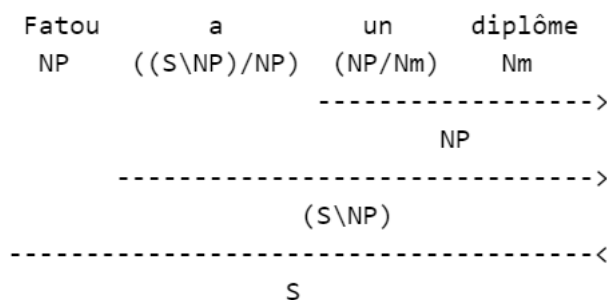


Figure 131 : Validation suggestion phrase 86

➤ Phrase 37 : Je veux veni à Paris

Analyse de la phrase : Je veux veni à Paris
Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Je      veux      veni      à      Paris
PROASM  (S\PROASM)/VP Unknow  Prep/NP  NP

```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Je veux veni à Paris	Phrase incorrecte	Position 8: Je veux veni à Paris -> venir : 100.00%, vent : 90.91%, vend : 81.82%, venu : 72.73%, venin : 63.64%, béni : 54.55%, déni : 45.45%, verni : 36.36%, nenni : 27.27%, henni : 18.18%, : 72.73%, venin : 63.64%, béni : 54.55%, déni : 45.45%, verni : 36.36%, nenni : 27.27%, henni : 18.18%, Venn : 9.09%

Figure 132 : Analyse CCG Phrase 87

La dérivation montre que la phrase 87 « Je veux veni à Paris » est incorrecte selon la GCCA. L'erreur porte sur le verbe « veni », une forme fautive de l'infinitif « venir ». En français, après un verbe modalisateur comme « veux », le verbe qui suit doit être à l'infinitif. L'usage de « veni » empêche ainsi la formation correcte du syntagme verbal, bloquant la dérivation syntaxique.

Le correcteur propose « venir » avec un score de 100 %, suivi de « vent » (90,91 %), « vend » (81,82 %), « venu » (72,73 %), et « Venn » (9,09 %), entre autres.

La correction en « Je veux venir à Paris » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 133.

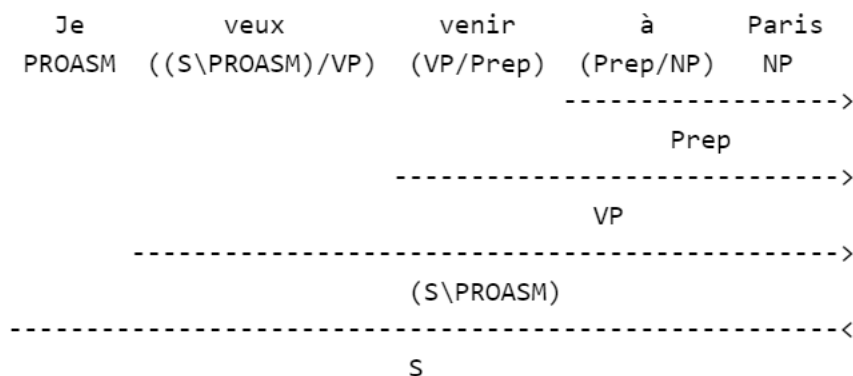


Figure 133 : Validation suggestion phrase 87

➤ Phrase 38 : Mon maman est fidèle

Analyse de la phrase : Mon maman est fidèle

Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

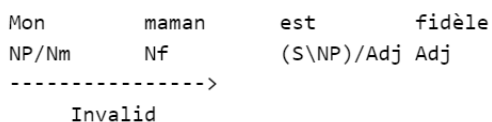


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Mon maman est fidèle	Phrase incorrecte	Position 0: Mon maman est fidèle -> Ma : 100.00%

Figure 134 : Analyse CCG Phrase 88

La dérivation montre que la phrase 88 « Mon maman est fidèle » est incorrecte selon la GCCA. L'erreur porte sur le déterminant possessif « mon », utilisé devant « maman », un nom féminin. En français, on utilise « ma » devant un nom féminin commençant par une consonne. L'emploi de « mon » entraîne donc un désaccord en genre, ce qui empêche la formation correcte du syntagme nominal et bloque ainsi la dérivation syntaxique.

Le correcteur propose « ma » avec un score de 100 %.

La correction en « Ma maman est fidèle » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 135.

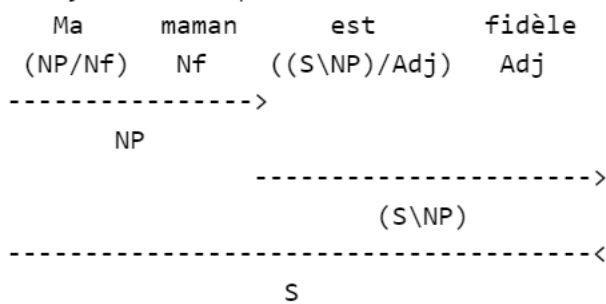


Figure 135 : Validation suggestion phrase 88

➤ Phrase 39 : Ma sœur aime donner l'aumone

Analyse de la phrase : Ma sœur aime donner l'aumone
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

Ma	sœur	aime	donner	l'aumone
NP/Nf	Nf	(S\NP)/VP	VP/NP	Unknow

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Ma sœur aime donner l'aumone	Phrase incorrecte	Position 22: Ma sœur aime donner l'aumone -> aumône : 100.00%, aumône : 100.00%, aumônes : 85.71%, ozone : 71.43%, saumoné : 57.14%, aurone : 42.86%, ozonè : 28.57%, ozoné : 14.29%

Figure 136 : Analyse CCG Phrase 89

La dérivation montre que la phrase 89 « Ma sœur aime donner l'aumone » est incorrecte selon GCCA.

L'erreur porte sur le nom « aumone », qui est mal orthographié : il manque l'accent circonflexe sur le « o ». En français, l'orthographe correcte est « aumône ». Cette faute empêche la reconnaissance correcte du mot comme nom, ce qui bloque la dérivation syntaxique à ce niveau.

Le correcteur propose « aumône » avec un score de 100 %, suivi de « aumônes » (85,71 %), « ozone » (71,43 %), « saumoné » (57,14 %) et « aurone » (42,86 %) entre autres.

La correction en « Ma sœur aime donner l'aumône » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 137.

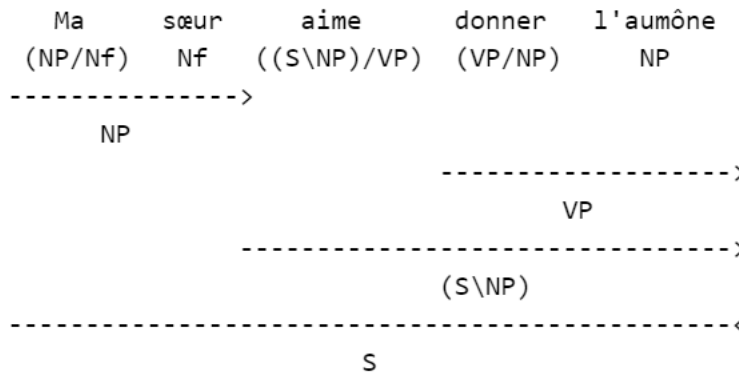


Figure 137 : Validation suggestion phrase 89

➤ Phrase 40 : Il seras malheureux

Analyse de la phrase : Il seras malheureux
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

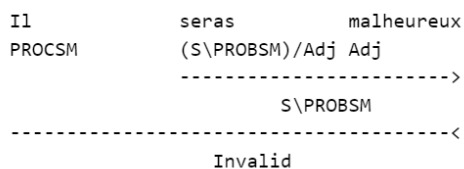


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il seras malheureux	Phrase incorrecte	Position 3: Il seras malheureux -> sera : 100.00%

Figure 138 : Analyse CCG Phrase 90

La dérivation montre que la phrase 90 « Il seras malheureux » est incorrecte selon la GCCA. L'erreur porte sur le verbe « seras », conjugué à la 2^e personne du singulier (« tu seras »), alors que le sujet « Il », à la 3^e personne du singulier, exige la forme verbale « sera ». Ce désaccord sujet-verbe empêche la composition correcte du syntagme verbal, bloquant la dérivation syntaxique.

Le correcteur propose « sera » avec un score de 100 %.

La correction en « Il sera malheureux » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 139.

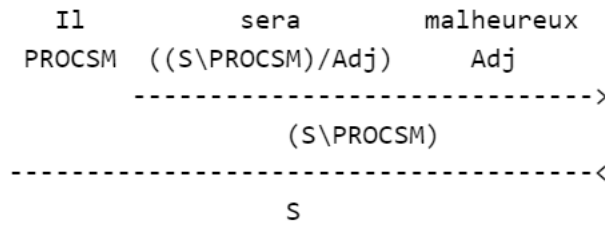


Figure 139 : Validation suggestion phrase 90

➤ Phrase 41 : Tu m'ont aimé

Analyse de la phrase : Tu m'ont aimé

Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

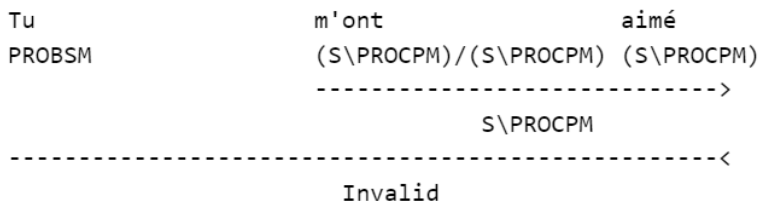


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Tu m'ont aimé	Phrase incorrecte	Position 5: Tu m'ont aimé -> as : 100.00%

Figure 140 : Analyse CCG Phrase 91

La dérivation montre que la phrase 91 « Tu m'ont aimé » est incorrecte selon la GCCA. L'erreur porte sur l'auxiliaire « ont », conjugué à la 3^e personne du pluriel, alors que le sujet « Tu », à la 2^e personne du singulier, exige la forme verbale « as ». Ce désaccord sujet-verbe empêche la composition correcte du syntagme verbal, bloquant la dérivation syntaxique.

Le correcteur propose « as » avec un score de 100 %.

La correction en « Tu m'as aimé » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 141.

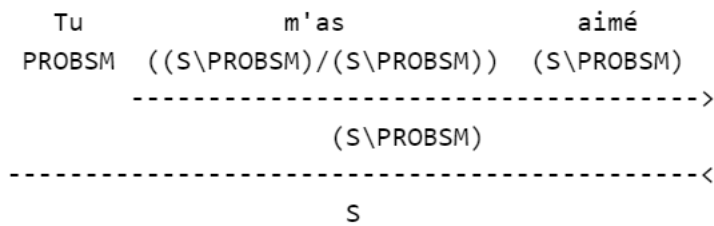


Figure 141 : Validation suggestion phrase 91

➤ Phrase 42 : Ils ont échoué à lexamen

Analyse de la phrase : Ils ont échoué à lexamen
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

Ils	ont	échoué	à	lexamen
PROCPM	(S\PROCPM)/(S\PROCPM)	(S\PROCPM)/Prep	Prep/NP	Unknow

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Ils ont échoué à lexamen	Phrase incorrecte	Position 17: Ils ont échoué à lexamen -> l'examen : 100.00%

Figure 142 : Analyse CCG Phrase 92

La dérivation montre que la phrase 92 « Ils ont échoué à lexamen » est incorrecte selon la GCCA.

L'erreur porte sur le nom « lexamen », qui est mal orthographié. En français, il faut écrire « l'examen » avec une apostrophe pour marquer l'élosion de l'article défini « le » devant un mot commençant par une voyelle. Cette faute bloque la reconnaissance correcte du syntagme prépositionnel « à l'examen », empêchant ainsi la bonne construction syntaxique de la phrase.

Le correcteur propose « l'examen » avec un score de 100 %.

La correction en « Ils ont échoué à l'examen » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 143.

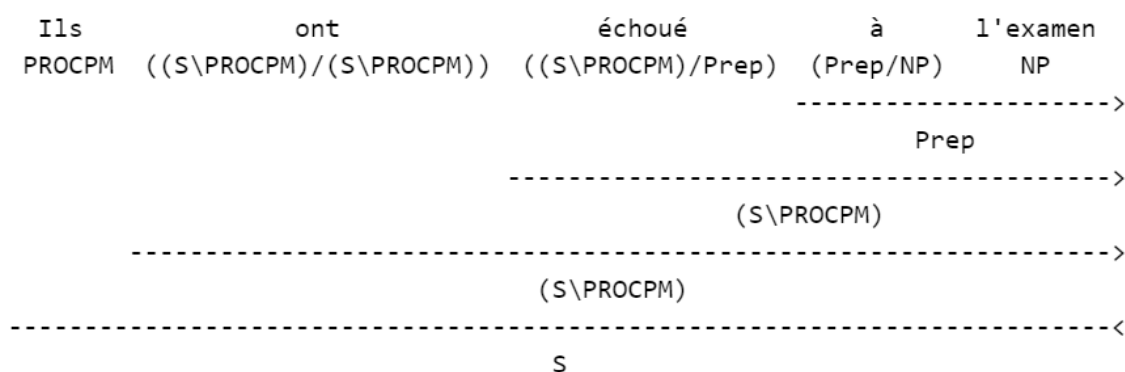


Figure 143 : Validation suggestion phrase 92

➤ **Phrase 43 : La cour est très propr**

Analyse de la phrase : La cour est très propr
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

La	cour	est	très	propr
NP/Nf	Nf	(S\NP)/Adj	(S\NP)\(S\NP)	Unknow

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
La cour est très propr	Phrase incorrecte	Position 17: La cour est très propr -> propre : 100.00%

Figure 144 : Analyse CCG Phrase 93

La dérivation montre que la phrase 93 « La cour est très propr » est incorrecte selon la GCCA. L'erreur porte sur l'adjectif « propr », qui est mal orthographié. En français, la forme correcte est « propre », avec un e final».

Cette faute empêche la reconnaissance correcte de l'adjectif et bloque ainsi la complétion du prédicat « est très propre », nécessaire à la dérivation syntaxique complète.

Le correcteur propose « propre » avec un score de 100 %.

La correction en « La cour est très propre » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 145.

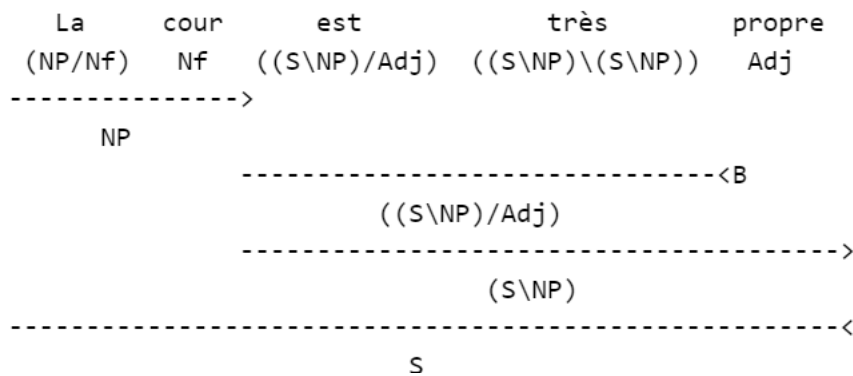


Figure 145 : Validation suggestion phrase 93

➤ Phrase 44 : Il a sourient

Analyse de la phrase : Il a sourient

Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

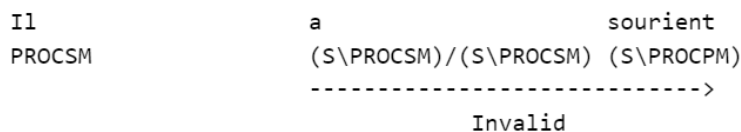


Tableau des erreurs détectées

+-----+-----+-----+
Phrase Statut Erreurs et Suggestions
+-----+-----+-----+
Il a sourient Phrase incorrecte Position 5: Il a sourient -> souri : 100.00%
+-----+-----+-----+

Figure 146 : Analyse CCG Phrase 94

La dérivation montre que la phrase 94 « Il a sourient » est incorrecte selon la GCCA.

L'erreur porte sur le mot « sourient », utilisé ici à la place du participe passé « souri ». En français, après l'auxiliaire « a », le verbe doit être au participe passé pour former le passé composé. Or, « sourient » est une forme conjuguée à la 3^e personne du pluriel du présent de l'indicatif, ce qui rend la construction incorrecte.

Cette faute empêche la composition correcte du syntagme verbal « a souri », bloquant ainsi la dérivation syntaxique à ce niveau.

Le correcteur propose « souri » avec un score de 100 %.

La correction en « Il a souri » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 147.

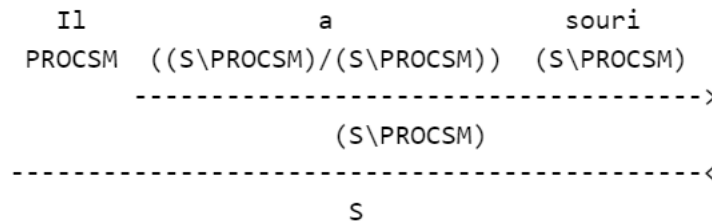


Figure 147 : Validation suggestion phrase 94

➤ Phrase 45 : Je veux connaître le vérité

Analyse de la phrase : Je veux connaître le vérité
Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

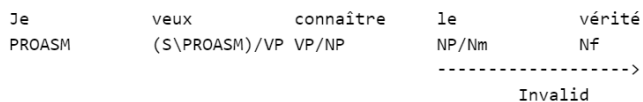


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Je veux connaître le vérité	Phrase incorrecte	Position 18: Je veux connaître le vérité -> la vérité : 100.00%,
Erreurs et Suggestions		
Position 18: Je veux connaître le vérité -> la vérité : 100.00%, les vérités : 50.00%		

Figure 148 : Analyse CCG Phrase 95

La dérivation montre que la phrase 95 « Je veux connaître le vérité » est incorrecte selon la GCCA.

L'erreur porte sur le déterminant « le », utilisé devant « vérité », un nom féminin. En français, le déterminant doit s'accorder en genre avec le nom qu'il détermine. Il faut donc employer « la » au lieu de « le ».

Cette faute d'accord empêche la reconnaissance correcte du groupe nominal « la vérité », ce qui bloque la dérivation syntaxique complète du syntagme verbal « connaître la vérité ».

Le correcteur propose « la vérité » avec un score de 100 %, et « les vérités » avec un score de 50 %.

La correction en « Je veux connaître la vérité » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte. comme l'illustre la figure 149.

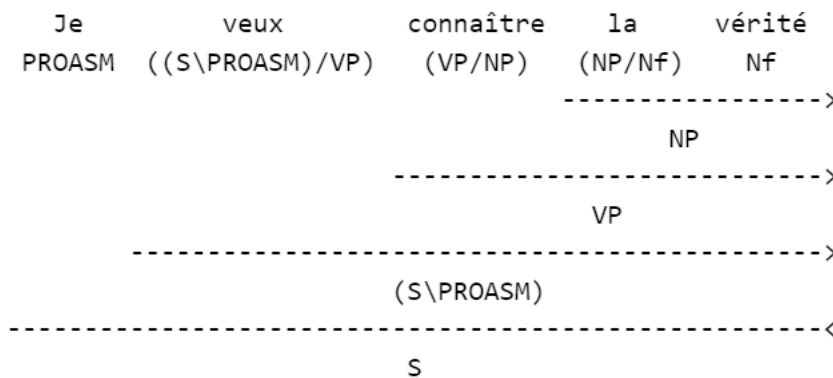


Figure 149 : Validation suggestion phrase 95

➤ Phrase 46 : Il vais à l'école

Analyse de la phrase : Il vais à l'école
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

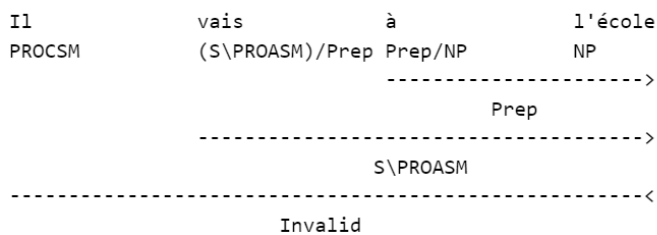


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il vais à l'école	Phrase incorrecte	Position 3: Il vais à l'école -> va : 100.00%

Figure 150 : Analyse CCG Phrase 96

La dérivation montre que la phrase 96 « Il vais à l'école » est incorrecte selon la GCCA.

L'erreur porte sur le verbe « vais », qui est la forme conjuguée de la 1^{re} personne du singulier du verbe « aller ». Or, le sujet « Il », à la 3^e personne du singulier, exige la forme correcte « va ».

Cette faute d'accord sujet-verbe empêche la construction correcte du syntagme verbal « va à l'école », ce qui bloque la dérivation syntaxique de la phrase.

Le correcteur propose « va » avec un score de 100 %.

La correction en « Il va à l'école » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 151.

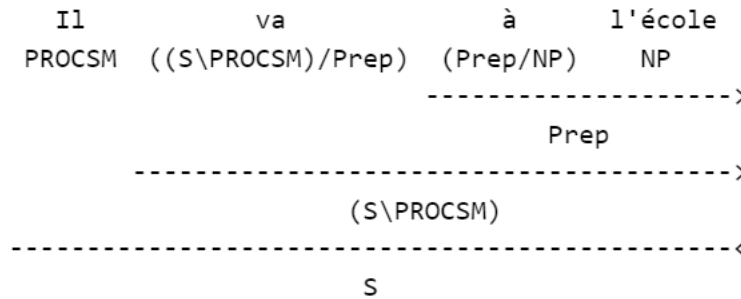


Figure 151 : Validation suggestion phrase 96

➤ Phrase 47 : Papa vont travailler

Analyse de la phrase : Papa vont travailler
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

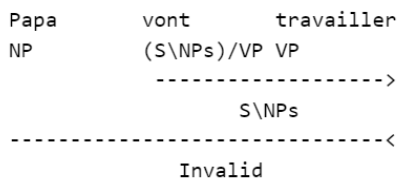


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Papa vont travailler	Phrase incorrecte	Position 5: Papa vont travailler -> va : 100.00%

Figure 152 : Analyse CCG Phrase 97

La dérivation montre que la phrase 97 « Papa vont travailler » est incorrecte selon la GCCA.

L'erreur porte sur le verbe « vont », qui est la forme conjuguée du verbe « aller » à la 3^e personne du pluriel. Or, le sujet « Papa », au singulier, exige la forme verbale « va ».

Cette faute d'accord sujet-verbe empêche la construction correcte du syntagme verbal « va travailler », ce qui bloque la dérivation syntaxique complète de la phrase.

Le correcteur propose « va » avec un score de 100 %.

La correction en « Papa va travailler » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 153.

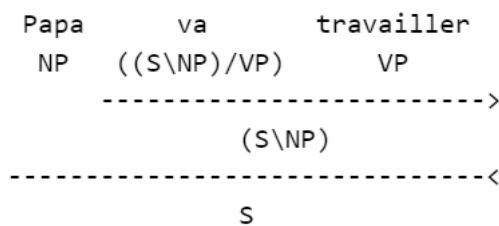


Figure 153 : Validation suggestion phrase 97

➤ Phrase 48 : Je mange un tomate

Analyse de la phrase : Je mange un tomate
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

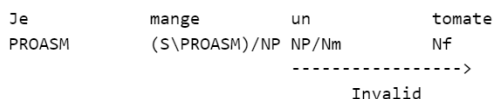


Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Je mange un tomate	Phrase incorrecte	Position 9: Je mange un tomate -> une tomate : 100.00%, des tomates : 50.00
Je mange un tomate	Phrase incorrecte	Position 9: Je mange un tomate -> une tomate : 100.00%, des tomates : 50.00

Figure 154 : Analyse CCG Phrase 98

La dérivation montre que la phrase 98 « Je mange un tomate » est incorrecte selon la GCCA.

L'erreur porte sur le déterminant « un », qui est de genre masculin, alors que le nom « tomate » est féminin. En français, le déterminant doit s'accorder en genre avec le nom qu'il introduit.

Cette faute d'accord empêche la formation correcte du syntagme nominal « une tomate », bloquant ainsi la dérivation syntaxique complète.

Le correcteur propose « une tomate » avec un score de 100 %, suivi de « des tomates » (50 %).

La correction en « Je mange une tomate » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 155.

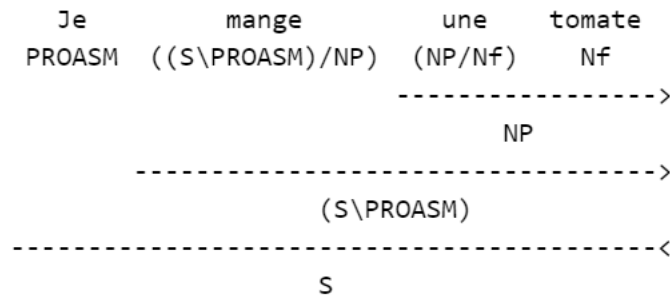


Figure 155 : Validation suggestion phrase 98

➤ Phrase 49 : Le bonne prépare le repas

Analyse de la phrase : Le bonne prépare le repas
Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

Le	bonne	prépare	le	repas
NP/Nm	Nf	(S\NP/NP	NP/Nm	Nm
----->				
Invalid				

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Le bonne prépare le repas	Phrase incorrecte	Position 0: Le bonne prépare le repas -> La bonne : 100.00%,
Erreurs et Suggestions		
Position 0: Le bonne prépare le repas -> La bonne : 100.00%, Le bon : 75.00%, Les bons : 50.00%, Les bonnes : 25.00%		

Figure 156 : Analyse CCG Phrase 99

La dérivation montre que la phrase 99 « Le bonne prépare le repas » est incorrecte selon la GCCA.

L'erreur porte sur le déterminant « le », qui est de genre masculin, alors que l'adjectif « bonne » est de genre féminin. En français, le déterminant doit s'accorder en genre avec le nom qu'il introduit — ici, « bonne » est probablement un nom féminin (« la bonne », désignant une personne), ou bien un adjectif modifiant un nom féminin implicite.

Cette faute d'accord empêche la formation correcte du syntagme nominal « la bonne », ce qui bloque la construction du groupe sujet et empêche la dérivation syntaxique complète.

Le correcteur propose « la bonne » avec un score de 100 %, suivi de « le bon » (75 %), suivi de « les bons » (50 %) et « les bonnes » (25 %).

La correction en « La bonne prépare le repas » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 157.

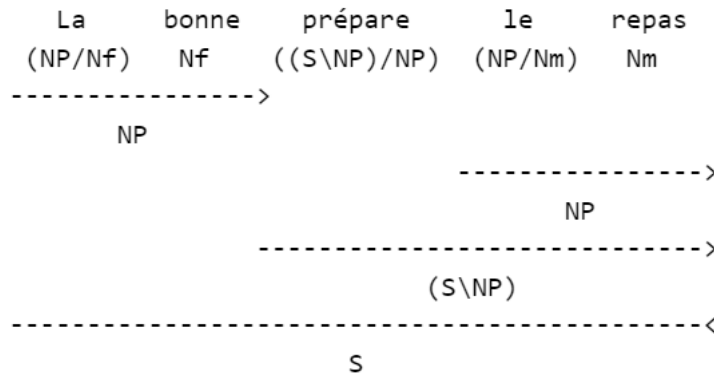


Figure 157 : Validation suggestion phrase 99

➤ Phrase 100 : Il a perdu du poids

Analyse de la phrase : Il a perdu du poids
 Statut : La phrase est incorrecte, mais voici l'analyse CCG :

Étapes de dérivation catégorielle partielle :

```

Il          a          perdu          du          poids
PROCSM      (S\PROCSM/(S\PROCSM)) (S\PROCSM)/NP  NP/Nm      Unknow
  
```

Tableau des erreurs détectées

Phrase	Statut	Erreurs et Suggestions
Il a perdu du poids	Phrase incorrecte	Position 14: Il a perdu du poids -> poids : 100.00%, poil : 85.71%, poids : 100.00%, poil : 85.71%, pois : 71.43%, pond : 57.14%, poix : 42.86%, poud : 28.57%, po id : 14.29%

Figure 158 : Analyse CCG Phrase 100

Dans la phrase 100 « Il a perdu du poids », le mot « poids » est mal orthographié. Cette erreur empêche de reconnaître correctement le groupe nominal, ce qui bloque l'analyse syntaxique. Le mot est considéré comme inconnu par le système. Le correcteur suggère « poids » avec un score de 100 %, suivi de « poil » (85.71%), suivi de « pois » (71.43 %) et de « pond » (57.14 %), etc.

La correction en « Il a perdu du poids » permettrait de compléter la dérivation et d'aboutir à la catégorie S, indiquant une phrase syntaxiquement correcte comme l'illustre la figure 159.

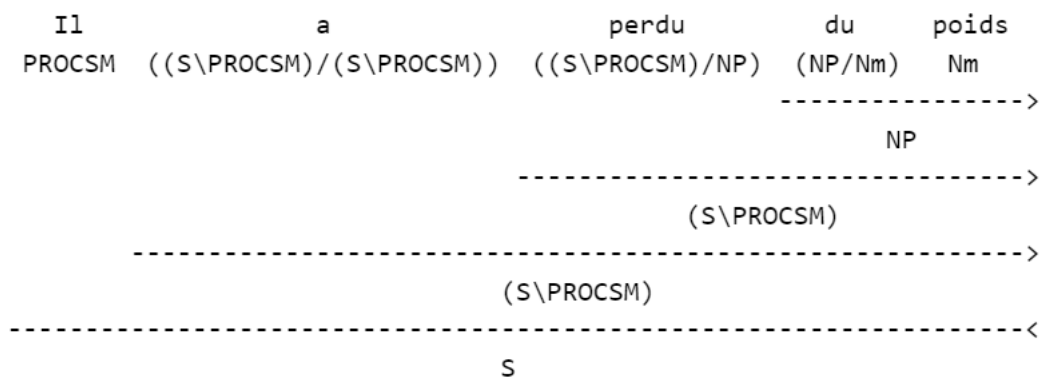


Figure 159 : Validation suggestion phrase 100

V.2) Conclusion

L'analyse des cinquante phrases présentées, selon les principes de la Grammaire Catégorielle Combinatoire et Applicative (GCCA), a permis de confirmer leur non-conformité syntaxique. Chaque mot a été associé à une catégorie formelle issue d'un lexique prédéfini, et les règles standard de combinaison fonctionnelle application avant (>) et application arrière (<) ont mis en évidence les ruptures de dérivation.

Dans certains cas, des tentatives de dérivation partielle ont nécessité l'usage de règles de composition pour explorer d'éventuelles compatibilités. Ces ajustements, bien que syntaxiquement autorisés, n'ont pas permis d'aboutir à une catégorie finale de type S, confirmant ainsi l'invalidité des phrases selon les critères de la GCCA. Cette méthode permet donc non seulement de détecter les erreurs, mais aussi de comprendre leur origine structurelle, ouvrant la voie à une correction ciblée et à un apprentissage grammatical plus rigoureux.

VI) Performance des suggestions de correction

Pour évaluer les performances de notre code en matière de suggestion de correction, nous avons utilisé un fichier CSV contenant 50 phrases. Ce fichier comprend trois colonnes : la phrase, le label (incorrecte) et la suggestion attendue.

Nous avons ensuite comparé les suggestions générées par LanguageTool aux suggestions attendues. Deux colonnes supplémentaires ont été ajoutées : Rang, qui indique la position de la suggestion attendue dans la liste des suggestions fournies, et Poids, qui représente un pourcentage pondérant cette position. Un rang de 1 correspond à une suggestion attendue proposée en première position (Voir Annexe III).

À partir de cette analyse, nous avons calculé le pourcentage de cas où la suggestion attendue apparaissait en première position, ce qui nous donne un indicateur direct de la pertinence des suggestions proposées. Ce taux a été de 90 %, comme le montre la figure suivante.

1	Total des suggestions attendues	En première position	Hors première position	Pourcentage en rang 1
2	50	45	5	90.00%

Figure 160: Fichier récapitulatif des performances de LanguageTool en matière de suggestion

Par ailleurs, nous avons extrait dans un fichier (figure 160) à part les phrases pour lesquelles la suggestion attendue n'était pas en première position sur la liste de suggestion proposée par LanguageTool .

1	phrase	label	suggestion_attendue	Rang	Poids
2	Il parlent beaucoup	Incorrecte	Il parle	2.0	50.0
3	Ils pense nous duper	Incorrecte	Ils pensent	2.0	50.0
4	Nous avons un secrets	Incorrecte	un secret	2.0	75.0
5	Il pleurent	Incorrecte	Il pleure	2.0	50.0
6	Aujourd'hui elle parlent	Incorrecte	elle parle	2.0	50.0

Figure 161 : Phrases avec position de la première suggestion différente de celle proposée par LanguageTool

Ces résultats nous permettent d'avoir une réelle appréciation des performances de LanguageTool, non seulement en termes de détection d'erreurs, mais aussi en termes de pertinence des suggestions, ce qui est essentiel pour l'assistance à l'apprentissage du français langue seconde.

VII) Performance de notre modèle

Pour évaluer les performances de notre code, nous avons créé un fichier CSV dans lequel sont insérés 100 phrases. Ce fichier comporte deux colonnes : une colonne phrase et une colonne label (indiquant si la phrase est correcte ou incorrecte), comme présenté dans l'Annexe IV.

Nous comparons ensuite les prédictions de LanguageTool (indiquant si une phrase est correcte ou non) aux statuts que nous avons attribués à chaque phrase. Deux éléments sont alors générés :

- Une matrice de confusion TP, FP, FN, TN (Voir Annexe V)
- Un rapport de classification (précision, rappel, F1-score, l'Accuracy.).

La Précision qui est le nombre de vrais positifs divisé par le nombre total de phrases classées comme positives (vrais positifs + faux positifs). Elle mesure la capacité du modèle à ne pas étiqueter comme positif un échantillon négatif.

$$\text{Précision(Phrase Correcte)} = \frac{TP}{TP+FP}$$

$$\text{Précision}(\text{Phrase Incorrecte}) = \frac{TN}{TN+FN}$$

Le Recall(rappel) qui est le nombre de vrais positifs divisé par le nombre total de phrases positives (vrais positifs + faux négatifs). Il mesure la capacité du modèle à trouver tous les échantillons positifs.

$$\text{Rappel}(\text{Phrase Correcte}) = \frac{TP}{TP + FN}$$

$$\text{Rappel}(\text{Phrase Incorrecte}) = \frac{TN}{TN + FP}$$

Le F1-score qui représente la moyenne harmonique de la précision et du rappel. Il prend en compte à la fois les faux positifs et les faux négatifs et donne une idée de l'équilibre entre précision et rappel.

$$F1(\text{Phrase Correcte}) = 2 \cdot \frac{\text{Précision} \cdot \text{Rappel}(\text{Phrase Correcte})}{(\text{Précision} + \text{Rappel})(\text{Phrase Correcte})}$$

$$F1(\text{Phrase Incorrecte}) = 2 \cdot \frac{\text{Précision} \cdot \text{Rappel}(\text{Phrase Incorrecte})}{(\text{Précision} + \text{Rappel})(\text{Phrase Incorrecte})}$$

- Le support qui est le nombre de phrases réelles dans chaque catégorie.
- L'Accuraccy qui est le pourcentage de prédictions correctes parmi toutes les prédictions

$$\text{Accuraccy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Ces éléments nous permettent d'évaluer les performances du système, notamment sa capacité à identifier correctement les phrases correctes et incorrectes selon les critères que nous avons définis.

Les résultats obtenus sont ainsi comparés aux statuts de référence pour évaluer la précision du modèle, et le rapport de classification fournit une mesure globale de sa performance.

```

+-----+-----+
| Type           | Nombre |
+-----+-----+
| Vrai positif (TP) | 41 |
+-----+-----+
| Faux négatif (FN) | 9 |
+-----+-----+
| Faux positif (FP) | 0 |
+-----+-----+
| Vrai négatif (TN) | 50 |
+-----+-----+

```

Rapport de classification :

	precision	recall	f1-score	support
Incorrecte	0.85	1.00	0.92	50
Correcte	1.00	0.82	0.90	50
accuracy			0.91	100
macro avg	0.92	0.91	0.91	100
weighted avg	0.92	0.91	0.91	100

Figure 162: Rapport métrique du jeu de test

Interprétation

Les phrases correctes

- La précision est de 1.00, cela signifie que, parmi toutes les phrases classées comme correctes par le modèle, 100% étaient effectivement correctes.
- Le Recall (Rappel) est de 0.82, cela signifie que, parmi toutes les phrases effectivement correctes, 82% ont été correctement identifiées par le modèle.
- Le F1-score est de 0.90 : c'est une mesure combinée de la précision et du rappel, indiquant un bon équilibre dans la classification des phrases correctes.
- Le Support est de 50 et cela signifie qu'il y avait 50 phrases correctes sur les 100 phrases que contenaient le jeu de test.

Les phrases incorrectes

- La précision est de 0.85 et cela signifie que, parmi toutes les phrases classées comme incorrectes par le modèle, 85% étaient effectivement incorrectes.
- Le Rappel est de 1.00 et cela signifie que, parmi toutes les phrases effectivement incorrectes, 100% ont été correctement identifiées par le modèle.

- Le F1-score est de 0.92 est une mesure combinée de la précision et du rappel, indiquant un bon équilibre dans la classification des phrases incorrectes.
- Le Support est de 50 et cela signifie qu'il y avait 50 phrases incorrectes sur les 100 phrases que contenaient le jeu de test
- L' Accuracy (Exactitude) est de 0.91 et cela signifie que 91% des phrases (correctes et incorrectes) ont été correctement classées par le modèle.
- Le Macro avg (Moyenne macro) est de 0.91 et représente la moyenne arithmétique des précisions, rappels et F1-scores pour toutes les classes, sans tenir compte du support.
- Le Weighted avg (Moyenne pondérée) est de 0.91, ce qui correspond à la moyenne pondérée des précisions, rappels et F1-scores pour toutes les classes, en tenant compte du support.

En résumé, notre modèle a une bonne performance générale avec une précision et un rappel élevé, indiquant qu'il est capable de bien distinguer les phrases correctes des phrases incorrectes.

Pour mieux comprendre les résultats de notre métrique, nous avons aussi créé un fichier `rapports_erreurs.csv` qui représente un rapport des discordances entre les résultats attendus et les résultats obtenus par notre code lors de l'analyse des phrases. Il contient toutes les phrases pour lesquelles il y a eu une différence entre le statut attendu (correct ou incorrect) et le statut déterminé par le code comme le montre la figure

1	Phrase	Statut(Fichier CSV d'entrée)	Statut prédite avec Language_Tool	Type d'erreur
2	Hier je ai bien dormi	correcte	incorrecte	Faux négatif (FN)
3	Nous allons nous-tresser	correcte	incorrecte	Faux négatif (FN)
4	Je pense que il est timide	correcte	incorrecte	Faux négatif (FN)
5	Ma fille craint de se-blesser	correcte	incorrecte	Faux négatif (FN)
6	Je ai réalisé un excellent projet	correcte	incorrecte	Faux négatif (FN)
7	Je ai tant à-offrir	correcte	incorrecte	Faux négatif (FN)
8	Je ne-suis pas comme toi	correcte	incorrecte	Faux négatif (FN)
9	Je ai fini la rédaction de mon mémoire	correcte	incorrecte	Faux négatif (FN)
10	Je veux me-marier	correcte	incorrecte	Faux négatif (FN)

Figure 163: Résultat `rapports_erreurs.csv`

En d'autres termes, ce fichier sert à identifier les cas où notre code a commis des erreurs de classification, soit en marquant une phrase correcte comme incorrecte, soit en marquant une phrase incorrecte comme correcte. Cela nous permet de cibler les zones où le code pourrait nécessiter des améliorations ou des ajustements.

Ces erreurs détectées par LanguageTool proviennent d'écart entre l'orthographe standard attendue par l'outil et les conventions de représentation propres à notre lexique, comme l'usage du trait d'union dans des constructions verbales telles que « Je veux me-marier ».

Ces erreurs, bien que considérées comme incorrectes par LanguageTool, sont en réalité conformes à notre modèle syntaxique. Elles sont donc attendues et ne remettent pas en cause le niveau de performance de notre modèle d'analyse grammaticale.

D'ailleurs, si l'on exclut ces cas particuliers du calcul des performances, notre système obtiendrait un F1-Score de 100 %, ce qui témoigne de la fiabilité de notre analyse dans le cadre défini par notre lexique et nos règles de combinaison.

VIII) Conclusion

Dans ce chapitre, nous avons présenté en détail nos résultats, éléments fondamentaux pour l'évaluation de notre modèle. Nous avons ensuite analysé les résultats obtenus, en mettant en lumière les performances de notre solution dans différents contextes d'application. Les résultats montrent que notre approche catégorielle est capable de traiter efficacement les structures syntaxiques complexes, tout en offrant une précision notable dans l'analyse des phrases, dans la détection d'erreurs et dans la suggestion de correction. Dans la conclusion, nous discuterons des perspectives d'évolution de notre travail et des pistes de recherche qui pourraient être explorées pour surmonter certains défis identifiés.

Chapitre VI : Conclusion générale

Dans ce travail, nous avons exploré en profondeur l'application de la grammaire catégorielle combinatoire et applicative (GCCA) à l'analyse syntaxique du français. Notre travail s'est concentré sur l'adaptation de cette grammaire aux spécificités linguistiques du français, en tenant compte des particularités des pronoms, des déterminants, des modaux, des adverbes et des verbes.

Grâce à l'utilisation d'un corpus de 100 phrases en français, aux outils de la bibliothèque Natural Language Toolkit (NLTK) et aux outils de la bibliothèque python LanguageTool, nous avons pu mettre en place une solution informatique permettant de faire l'analyse syntaxique de phrases mais aussi, de détecter les erreurs syntaxiques dans les phrases, d'indiquer leur position et de proposer des suggestions de correction.

Les résultats obtenus montrent que la GCCA est capable de fournir une analyse syntaxique précise et flexible, même pour des phrases complexes. La catégorisation des éléments lexicaux et l'application des règles de combinaison spécifiques à la GCCA se sont avérées essentielles pour atteindre une telle précision. Les résultats obtenus dans le fichier métrique présenté montrent aussi que notre solution a une bonne performance générale avec une précision et un rappel élevé, indiquant qu'il est capable de bien distinguer les phrases correctes des phrases incorrectes.

Cependant, notre recherche a également révélé certains défis, notamment la nécessité d'adapter plus précisément les règles de combinaison pour certaines structures syntaxiques complexes du français. En outre, il serait pertinent de développer un lexique capable de s'adapter à une grande variété de phrases et d'évaluer notre solution sur un corpus plus étendu, tel qu'un fichier CSV contenant par exemple 2000 phrases, afin de tester le temps de réponse.

De plus, l'intégration de techniques d'apprentissage automatique pourrait considérablement améliorer la précision des corrections proposées par notre solution. Enfin, il serait également intéressant de concevoir une application web pour notre solution, afin de rendre son utilisation plus accessible et pratique.

Bibliographie

- Barendregt, H. (1984). *The lambda calculus : its syntax and semantics*. Amsterdam: North-Holland.
- Lutz, M. (2013). *Learning python*. Sebastopol: O'Reilly Media.
- Summerfield, M. (2009). *Programming in python 3*. Boston: Addison-Wesley Professional.
- Amsili, P. (2015). *Grammaires catégorielles combinatoires*.
- Anoun, H. (2007). Approche logique des grammaires pour les langues naturelles. Bordeaux: Université des Sciences et Technologies de Bordeaux I.
- Baldrige, J., & Kruijff, G.-J. (2002). *Multi-modal combinatory categorial grammar*. THESE DE DOCTORAT, University of Edinburgh / Charles University, Edinburgh, Prague.
- Bird, S., Klein, E., & Loper, E. (2024, 08 19). Retrieved from Natural language toolkit.
- Biskri, I., & Descles, J.-P. (1995). *Logique combinatoire et linguistique: grammaire catégorielle combinatoire applicative*. Paris: EHESS.
- Biskri, I., Meunier, J.-G., & Nault, G. (1988). Extraction des connaissances terminologiques au moyen des grammaires catégorielles : un modèle hybride. *Actes de 4e Jour. Int. d'Analyse Statistique des Données Textuelles*, (pp. 123-135). Nice.
- Bitam, M. R. (2022). *Les Systèmes Applicatifs Typés : Vers Une Mise En Oeuvre Réelle*. Mémoire de Maitrise, UQTR.
- Bourdeau, M. (2002). *la genèse des grammaires catégorielles et leur arrière plan logico-philosophique: quelques remarques*. Paris: Revue language.
- Chaumartin, F.-R. (2020). *Le traitement automatique des langues : comprendre les textes grâce à l'intelligence artificielle*. Paris: Dunod.
- Clark, S., & Curran, J. (2007). Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics*, 3(4), pp. 493-552.
- Cori, M. (2020). *Le traitement automatique des langues en question : des machines qui comprennent le français ?* Paris: Éditions Cassini.
- Cuq, J.-P. (2013). *Le FLS: un concept en question*. Montpellier: Tréma.
- De Waelhens , A. (1954). Phénoménologie husserlienne et phénoménologie hégélienne. *Revue Philosophique de Louvain*(134), pp. 234-249.
- Desclés, J.-P. (1988). Théorème de Church-Rosser et structuration des langues naturelles. *mathématiques et sciences humaines*, pp. 67-92.
- Descles, J.-P. (2003). La grammaire applicative et cognitive construit-elle des représentations universelles ?
- Dortier, J.-F. (2022, 08 14). Husserl, les arbres en fleur et la phénoménologie. *Toute la philosophie en quatre questions*, p. 49 à 53.
- Downey, A. (2024). *Think Python*. O'Reilly Media: Sebastopol.

- Dregosa, F., Gramfort, A., Varoquaux, G., Prettenhofer, P., Weiss, R., Michel, V., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, pp. 2825-2830.
- EITCI. (2023, 08 02). *Grammaire contextuelle*.
- Gerner, H., & STUMPF, B. (2007). *Lexiques scientifiques et techniques : constitution et approche historique*. Palaiseau: Éditions de l'École Polytechnique.
- Ginisti, J.-P. (1988). Présentation de la logique combinatoire en vue de ses applications. *Mathématiques et sciences humaines*, pp. N° 103, p. 45-66.
- Godart-Wending, B. (2003). Les grammaires catégorielles lambekiennes : le traitement des constituants discontinus. *Lynx*(48), 123-137.
- Goubault-Larrecq, J. (2001). *Lambda-calcul et langages fonctionnels*. Paris: Dunod.
- Guilbert, G., Sauzay, B., & Descles, J.-P. (2016). *Logique combinatoire et lambda-calcul : des logiques d'opérateurs*. Toulouse: Éditions Cépaduès.
- Hassani, M. (2016). *Vers un modèle formel applicatif pour une nouvelle ingénierie de la langue et de l'information*. Mémoire de Maîtrise, UQTR, Trois Rivières.
- Irvine, A. D., Deutsch, H., & Marshall, O. (2025, Mars 21). Russell's Paradox. Stanford, Californie, USA.
- Joly, A. (2009). *Du document textuel à sa carte sémantique fonctionnelle*. mémoire de maitrise, UQTR.
- Joray, P., & Godart-Wending, B. (2002). De la théorie des catégories sémantiques de Lesniewski à l'analyse de la quantification dans la syntaxe d'Ajdukiewicz. *Languages*(148), 28-50.
- Moreau, E. (2004). Apprentissage partiel de grammaires catégorielles. *TALN 2004*. FES.
- Muller, C., Royauté, J., & Silberztein, M. (2004). *INTEX : Pour la linguistique et le traitement automatique des langues*. Besançon: Presses universitaires de Franche-Comté.
- Ramalho, L. (2022). *Fluent Python*. Sebastopol: O'Reilly Media.
- Rapaille, T. (2011). *Ni husserlienne, ni russellienne : la théorie des catégories sémantiques de Lesniewski*. Liège: Presses universitaires de Liège.
- Sadok, S. M. (2010). *Applications Linguistiques Multilingues: Apport Des Grammaires Catégorielles Et De la Logique Combinatoire*. Mémoire de Maitrise, UQTR.
- Second, F. (1990). Approches des grammaires catégorielles. *Mathématiques et sciences humaines*, pp. 47-60.
- SourceForge.net. (2015, 03 16). Retrieved from SourceForge.net.
- Steedman, M., & Baldridge, J. (2006). Combinatory Categorical Grammar. *Encyclopedia of Language and Linguistics*, 2, pp. 610–620.
- Tautou, A.-S. (2025, 06 16). *LanguageTool, votre correcteur d'orthographe multilingue* . Retrieved from LangageTool Insights.

Annexe I : Corpus composé de 100 PHRASES

1	phrase	label
2	Je mange bien	Correcte
3	Elles veulent manger de la soupe	Correcte
4	Elles pensent aller au marché	Correcte
5	Plus-tard nous irons au salon	Correcte
6	Nous prierons pour vous	Correcte
7	Son enfant marche lentement	Correcte
8	La famille est sacrée	Correcte
9	La réussite reste sa priorité	Correcte
10	Il est le plus âgé d'entre nous	Correcte
11	Ma mère me-manque tellement	Correcte
12	Hier je ai bien dormi	Correcte
13	Nous irons tous au paradis	Correcte
14	Nous allons nous-tresser	Correcte
15	La vie est tellement courte	Correcte
16	Le professeur est généreux	Correcte
17	Elle a eu la meilleure note	Correcte
18	Je pense que il est timide	Correcte

19	Son nom est Maryam	Correcte
20	Je ne saurais te-dire	Correcte
21	Ce-matin je suis très heureuse	Correcte
22	Ma fille craint de se-blesser	Correcte
23	Elle pensait être seule	Correcte
24	Elle mérite vraiment sa promotion	Correcte
25	Maman est tellement serviable	Correcte
26	Ma maman reste ma meilleure amie	Correcte
27	Je ai réalisé un excellent projet	Correcte
28	Le vol est strictement interdit	Correcte
29	Nous avons le droit de-manger	Correcte
30	Elle t'a donné une dernière chance	Correcte
31	Nous craignons pour lui	Correcte
32	Je ai tant à-offrir	Correcte
33	La confiance en-soi est une qualité	Correcte
34	Ils chantent en chœur	Correcte
35	L'eau de piscine est jaune	Correcte

36	Papa a tellement mal au cœur	Correcte
37	Je ne-suis pas comme toi	Correcte
38	Ma famille a honte de moi	Correcte
39	Je dis non à la violence	Correcte
40	Je ai fini la rédaction de mon mémoire	Correcte
41	Le port du voile est obligatoire	Correcte
42	Il avait un mal de tête	Correcte
43	Son parfum sent tellement bon	Correcte
44	Vous voulez vraiment sortir	Correcte
45	Elle a vraiment besoin d'argent	Correcte
46	Je me marie en janvier	Correcte
47	L'amitié n'existe plus	Correcte
48	L'automne approche	Correcte
49	Chaque-matin je rends grâce à Allah	Correcte
50	Je veux me-marier	Correcte
51	Elle est trop timide	Correcte
52	Nous pouvez manger de la soupe	Incorrecte
53	Il parlent beaucoup	Incorrecte

54	Tu prend l'argent	Incorrecte
55	La monsieur est en vie	Incorrecte
56	Je vais rendre visit à ma maman	Incorrecte
57	Il sens tellement bon	Incorrecte
58	La mariage est prévu pour demain	Incorrecte
59	Le vie est tellement courte	Incorrecte
60	Je veux aller à le Mecque	Incorrecte
61	Je pense avoir la paludisme	Incorrecte
62	Son enfant est tout sa vie	Incorrecte
63	Vous avez peur pour son vie	Incorrecte
64	Sa père est fier d'elle	Incorrecte
65	Ils pense nous duper	Incorrecte
66	Il veux remplir sa bombonne	Incorrecte
67	Tu as trente enfant	Incorrecte
68	Je sens que je va tomber	Incorrecte
69	Son fils aiment Halloween	Incorrecte
70	Tu as reçu un ventilater	Incorrecte
71	Il m'as donné un nouvel avenir	Incorrecte

72	Je dois assister ma frère	Incorrecte
73	Je respectes mon ami	Incorrecte
74	Le fête du mouton approche	Incorrecte
75	Il aimais faire du bien	Incorrecte
76	Mon sœur a grandi	Incorrecte
77	Nous avons un secrets	Incorrecte
78	Grand-mère préparent une bonne soupe	Incorrecte
79	Je dégustes de la glace	Incorrecte
80	Elle veux la bague au doigt	Incorrecte
81	Tout âme goûtera à la mort	Incorrecte
82	Il pleurent	Incorrecte
83	Aujourd'hui elle parlent	Incorrecte
84	Personne n'aiment la prison	Incorrecte
85	Il est monté dan sa chambre	Incorrecte
86	Il m'as choisi	Incorrecte
87	Fatou a une diplôme	Incorrecte
88	Je veux veni à Paris	Incorrecte
89	Mon maman est fidèle	Incorrecte

90	Ma sœur aime donner l'aumone	Incorrecte
91	Il seras malheureux	Incorrecte
92	Tu m'ont aimé	Incorrecte
93	Ils ont échoué à lexamen	Incorrecte
94	La cour est très propr	Incorrecte
95	Il a sourient	Incorrecte
96	Je veux connaître le vérité	Incorrecte
97	Il vais à l'école	Incorrecte
98	Papa vont travailler	Incorrecte
99	Je mange un tomate	Incorrecte
100	La bonne prépaes le repas	Incorrecte
101	Il a perdu du poid	Incorrecte

Figure 164 : Corpus composé de 100 phrases

Annexe II : Lexique (Dictionnaire)

Il	PROCSM (3ème pronom personnel singulier masculin)
Elle	PROCSF (3ème pronom personnel singulier féminin)
Soupe	Nf(nom)
Soupes	Ns (nom au pluriel)
De	Prep
Des	NP/Ns
Veut	S\PROCSF(verbe ayant comme sujet le pronom elle et n'ayant pas de complément d'objet)
Peux	ModalCSM(verbe(3ème personne singulier masculin)qui doit être suivi par un verbe
Peut	(S\PROCSM) /NP(verbe(3ème personne singulier masculin)qui doit être suivi par un nom
Manger	TV((verbe a l'infinifit)
Rapidement	(S\NP)\(S\NP)

Tableau 2: Quelques catégories du lexique

Annexe III : Fichier csv résultats_suggestions

	phrase	label	suggestion_attendue	Rang	Poids
1	phrase				
2	Nous pouvez manger de la soupe	Incorrecte	pouvons	1.0	100.0
3	Il parlent beaucoup	Incorrecte	Il parle	2.0	50.0
4	Tu prend l'argent	Incorrecte	prends	1.0	100.0
5	La monsieur est en vie	Incorrecte	Le monsieur	1.0	100.0
6	Je vais rendre visit à ma maman	Incorrecte	visite	1.0	100.0
7	Il sens tellement bon	Incorrecte	sent	1.0	100.0
8	La mariage est prévu pour demain	Incorrecte	Le mariage	1.0	100.0
9	Le vie est tellement courte	Incorrecte	La vie	1.0	100.0
10	Je veux aller à le Mecque	Incorrecte	la Mecque	1.0	100.0
11	Je pense avoir la paludisme	Incorrecte	le paludisme	1.0	100.0
12	Son enfant est tout sa vie	Incorrecte	toute	1.0	100.0
13	Vous avez peur pour son vie	Incorrecte	sa	1.0	100.0
14	Sa père est fier d'elle	Incorrecte	Son père	1.0	100.0
15	Ils pense nous duper	Incorrecte	Ils pensent	2.0	50.0
16	Il veux remplir sa bombonne	Incorrecte	veut	1.0	100.0
17	Tu as trente enfant	Incorrecte	trente enfants	1.0	100.0
18	Je sens que je va tomber	Incorrecte	vais	1.0	100.0
19	Son fils aime Halloween	Incorrecte	aime	1.0	100.0
20	Tu as reçu un ventilater	Incorrecte	ventilateur	1.0	100.0
21	Il m'as donné un nouvel avenir	Incorrecte	a	1.0	100.0
22	Je dois assister ma frère	Incorrecte	mon frère	1.0	100.0
23	Je respectes mon ami	Incorrecte	respecte	1.0	100.0
24	Le fête du mouton approche	Incorrecte	La fête	1.0	100.0
25	Il aimais faire du bien	Incorrecte	aimait	1.0	100.0
26	Mon sœur a grandi	Incorrecte	Ma	1.0	100.0
27	Nous avons un secrets	Incorrecte	un secret	2.0	75.0
28	Grand-mère préparent une bonne so	Incorrecte	prépare	1.0	100.0
29	Je dégustes de la glace	Incorrecte	déguste	1.0	100.0
30	Elle veux la bague au doigt	Incorrecte	veut	1.0	100.0
31	Tout âme goûtera à la mort	Incorrecte	Toute âme	1.0	100.0
32	Il pleurent	Incorrecte	Il pleure	2.0	50.0
33	Aujourd'hui elle parlent	Incorrecte	elle parle	2.0	50.0
34	Personne n'aime la prison	Incorrecte	aime	1.0	100.0
35	Il est monté dan sa chambre	Incorrecte	dans	1.0	100.0
36	Il m'as choisi	Incorrecte	a	1.0	100.0
37	Fatou a une diplôme	Incorrecte	un diplôme	1.0	100.0
38	Je veux veni à Paris	Incorrecte	venir	1.0	100.0
39	Mon maman est fidèle	Incorrecte	Ma	1.0	100.0
40	Ma sœur aime donner l'aumone	Incorrecte	aumône	1.0	100.0
41	Il seras malheureux	Incorrecte	sera	1.0	100.0
42	Tu m'ont aimé	Incorrecte	as	1.0	100.0
43	Ils ont échoué à lexamen	Incorrecte	l'examen	1.0	100.0
44	La cour est très propr	Incorrecte	propre	1.0	100.0
45	Il a sourient	Incorrecte	sourit	1.0	100.0
46	Je veux connaître le vérité	Incorrecte	la vérité	1.0	100.0
47	Il vais à l'école	Incorrecte	va	1.0	100.0
48	Papa vont travailler	Incorrecte	va	1.0	100.0
49	Je mange un tomate	Incorrecte	une tomate	1.0	100.0
50	La bonne prépaes le repas	Incorrecte	prépare	1.0	100.0
51	Il a perdu du poids	Incorrecte	poids	1.0	100.0

Figure 165 : Fichier csv résultats_suggestions

Annexe IV : Fichier CSV pour l'évaluation de la performance du code

1	phrase	label
2	Je mange bien	Correcte
3	Elles veulent manger de la soupe	Correcte
4	Elles pensent aller au marché	Correcte
5	Plus-tard nous irons au salon	Correcte
6	Nous prierons pour vous	Correcte
7	Son enfant marche lentement	Correcte
8	La famille est sacrée	Correcte
9	La réussite reste sa priorité	Correcte
10	Il est le plus âgé d'entre nous	Correcte
11	Ma mère me-manque tellement	Correcte
12	Hier je ai bien dormi	Correcte
13	Nous irons tous au paradis	Correcte
14	Nous allons nous-tresser	Correcte
15	La vie est tellement courte	Correcte
16	Le professeur est généreux	Correcte
17	Elle a eu la meilleure note	Correcte
18	Je pense que il est timide	Correcte

19	Son nom est Maryam	Correcte
20	Je ne saurais te-dire	Correcte
21	Ce-matin je suis très heureuse	Correcte
22	Ma fille craint de se-blesser	Correcte
23	Elle pensait être seule	Correcte
24	Elle mérite vraiment sa promotion	Correcte
25	Maman est tellement serviable	Correcte
26	Ma maman reste ma meilleure amie	Correcte
27	Je ai réalisé un excellent projet	Correcte
28	Le vol est strictement interdit	Correcte
29	Nous avons le droit de-manger	Correcte
30	Elle t'a donné une dernière chance	Correcte
31	Nous craignons pour lui	Correcte
32	Je ai tant à-offrir	Correcte
33	La confiance en-soi est une qualité	Correcte
34	Ils chantent en chœur	Correcte
35	L'eau de piscine est jaune	Correcte

36	Papa a tellement mal au cœur	Correcte
37	Je ne-suis pas comme toi	Correcte
38	Ma famille a honte de moi	Correcte
39	Je dis non à la violence	Correcte
40	Je ai fini la rédaction de mon mémoire	Correcte
41	Le port du voile est obligatoire	Correcte
42	Il avait un mal de tête	Correcte
43	Son parfum sent tellement bon	Correcte
44	Vous voulez vraiment sortir	Correcte
45	Elle a vraiment besoin d'argent	Correcte
46	Je me marie en janvier	Correcte
47	L'amitié n'existe plus	Correcte
48	L'automne approche	Correcte
49	Chaque-matin je rends grâce à Allah	Correcte
50	Je veux me-marier	Correcte
51	Elle est trop timide	Correcte
52	Nous pouvez manger de la soupe	Incorrecte

53	Il parlent beaucoup	Incorrecte
54	Tu prend l'argent	Incorrecte
55	La monsieur est en vie	Incorrecte
56	Je vais rendre visit à ma maman	Incorrecte
57	Il sens tellement bon	Incorrecte
58	La mariage est prévu pour demain	Incorrecte
59	Le vie est tellement courte	Incorrecte
60	Je veux aller à le Mecque	Incorrecte
61	Je pense avoir la paludisme	Incorrecte
62	Son enfant est tout sa vie	Incorrecte
63	Vous avez peur pour son vie	Incorrecte
64	Sa père est fier d'elle	Incorrecte
65	Ils pense nous duper	Incorrecte
66	Il veux remplir sa bombonne	Incorrecte
67	Tu as trente enfant	Incorrecte
68	Je sens que je va tomber	Incorrecte
69	Son fils aiment Halloween	Incorrecte
70	Tu as reçu un ventilater	Incorrecte

71	Il m'as donné un nouvel avenir	Incorrecte
72	Je dois assister ma frère	Incorrecte
73	Je respectes mon ami	Incorrecte
74	Le fête du mouton approche	Incorrecte
75	Il aimais faire du bien	Incorrecte
76	Mon sœur a grandi	Incorrecte
77	Nous avons un secrets	Incorrecte
78	Grand-mère préparent une bonne soupe	Incorrecte
79	Je dégustes de la glace	Incorrecte
80	Elle veux la bague au doigt	Incorrecte
81	Tout âme goûtera à la mort	Incorrecte
82	Il pleurent	Incorrecte
83	Aujourd'hui elle parlent	Incorrecte
84	Personne n'aiment la prison	Incorrecte
85	Il est monté dan sa chambre	Incorrecte
86	Il m'as choisi	Incorrecte
87	Fatou a une diplôme	Incorrecte
88	Je veux veni à Paris	Incorrecte
89	Mon maman est fidèle	Incorrecte
90	Ma sœur aime donner l'aumone	Incorrecte
91	Il seras malheureux	Incorrecte
92	Tu m'ont aimé	Incorrecte
93	Ils ont échoué à lexamen	Incorrecte
94	La cour est très propr	Incorrecte
95	Il a sourient	Incorrecte
96	Je veux connaître le vérité	Incorrecte
97	Il vais à l'école	Incorrecte
98	Papa vont travailler	Incorrecte
99	Je mange un tomate	Incorrecte
100	La bonne prépaes le repas	Incorrecte
101	Il a perdu du poid	Incorrecte

Figure 166: Fichier Csv pour l'évaluation de la performance

Annexe V : Explication des éléments de la matrice de confusion

1	Statut de la phrase(fichier CSV d'entrée)	Statut de la phrase après détection de Language_Tool	Type
2	Correcte	Correcte	TP
3	Incorrecte	Incorrecte	TN
4	Incorrecte	Correcte	FP
5	Correcte	Incorrecte	FN

Figure 167 : Explication éléments de la matrice de confusion