Résumé long

Cette thèse intitulée *Des corpus arborés à l'induction de structures syntaxiques partielles* se divise en deux parties principales : une revue théorique qui présente des éléments de contexte nécessaire pour situer le travail, et des contributions originales réalisées au cours de la thèse.

La première partie, intitulée *Revue théorique*, se compose de trois chapitres. Le premier chapitre, *Pourquoi développer des treebanks ?*, pose les bases de notre étude en démontrant l'utilité des treebanks dans de nombreux domaines. Le deuxième, *Retracer l'histoire du développement des treebanks*, explore l'évolution de ces ressources essentielles, depuis les premières analyses syntaxiques systématiques jusqu'aux treebanks modernes. Le troisième chapitre, *Une typologie des pratiques de développement des treebanks*, propose une classification novatrice de ces dernières.

La seconde partie, *Contributions*, comprend quatre chapitres. Le chapitre quatre, *Constitution de guides d'annotations et problèmes rencontrés*, met en avant les difficultés inhérentes à l'activité de développement de treebanks et de leur documentation avec des études de cas pratiques. Le cinquième chapitre, *Développement collaboratif des treebanks*, rend compte d'un travail portant sur l'outillage visant à faciliter l'annotation et l'exploration des treebanks. Le sixième, *Propriétés globales des treebanks*, constate l'existence de certaines propriétés des arbres syntaxiques en lien avec des principes linguistiques théoriques. Enfin, le septième chapitre, *Induction partielle de structures à partir de corpus bruts*, conclut cette partie avec une approche innovante de la génération de structures syntaxiques.

La conclusion récapitule les accomplissements majeurs de la thèse et ouvre des

perspectives pour des travaux futurs. En annexe, un document complémentaire de dix pages est présenté, suivi de la bibliographie exhaustive de la thèse.

Dans le premier chapitre, nous introduisons la notion de *treebank*, ou corpus arborés. Ces corpus de textes sont annotés avec des informations syntaxiques telles que des catégories morphosyntaxiques et une description de la structure au moyen de regroupements en constituants ou par des relations de dépendance. Ils jouent un rôle crucial dans la recherche linguistique, l'enseignement des langues et les technologies du langage. Ce résumé présente leurs divers usages, soulignant leur importance dans le domaine de la linguistique computationnelle et au-delà.

Les treebanks permettent de remettre en question et de réviser la théorie syntaxique à partir de données empiriques. Ils sont particulièrement utiles pour explorer des constructions moins étudiées ou pour examiner la fréquence et la régularité des constructions dans diverses langues. Les treebanks, tout comme la linguistique basée sur les corpus, favorisent une approche plus inclusive de la syntaxe, en mettant en lumière des constructions souvent négligées dans les approches théoriques traditionnelles.

Les treebanks sont bénéfiques non seulement pour la syntaxe mais aussi pour d'autres domaines tels que la typologie linguistique, la prosodie, la sémantique, et la sociolinguistique. Ils offrent des données concrètes pour étudier les variations et les similitudes entre les langues, ainsi que pour comprendre les interactions entre divers niveaux linguistiques, notamment entre la syntaxe, la prosodie et la sémantique. Ils constituent également une ressource précieuse pour l'enseignement des langues étrangères et de la syntaxe. En permettant aux étudiants d'interagir avec des données linguistiques authentiques, ils encouragent à développer une compréhension plus approfondie et nuancée de la langue cible. Les treebanks parallèles, qui alignent des textes dans différentes langues, sont particulièrement utiles pour illustrer les différences syntaxiques et pour faciliter l'apprentissage comparatif.

Les treebanks sont essentiels pour le développement et l'évaluation des systèmes de traitement automatique des langues. Ils sont utilisés pour entraîner des analyseurs syntaxiques et pour évaluer leur performance. De plus, ils contribuent à améliorer la

qualité de diverses applications en TAL, telles que la traduction automatique, l'analyse de sentiments, la résolution de coréférence, et bien d'autres tâches.

Avec l'avènement des méthodes d'apprentissage profond, la question de savoir dans quelle mesure les représentations apprises par les réseaux neuronaux encodent des informations syntaxiques est devenue un sujet de recherche actif. Les treebanks jouent un rôle dans l'évaluation de ces représentations et dans la compréhension de leur potentiel pour capturer des phénomènes syntaxiques complexes. Ces questionnements sont essentiels pour favoriser l'explicabilité des ces nouvelles méthodes.

Ainsi, les treebanks sont des ressources polyvalentes qui influencent profondément la recherche linguistique, l'enseignement des langues et les technologies du langage. Ils offrent des perspectives uniques pour étudier la langue, formuler des théories basées sur les données et développer des outils de traitement automatique des langues plus efficaces. Les efforts mis en oeuvre pour comprendre et exploiter au mieux ces ressources sont essentiels afin de progresser dans la compréhension et le traitement automatique du langage naturel.

Le Chapitre 2 retrace l'histoire du développement des treebanks, mettant en lumière le contexte historique et les progrès dans le domaine des corpus annotés linguistiquement. Il suit les avancées chronologiques et les influences qui ont façonné les treebanks contemporains.

Les premières analyses syntaxiques systématiques, bien avant l'ère numérique, comprenaient des phrases authentiques et des exemples construits par des syntacticiens. Ces travaux soulèvent des questions sur la validité des exemples construits par rapport à ceux extraits des corpus, et les débats entre descriptions textuelles et diagrammatiques. Des chercheurs comme Buffier, Dumarsais, Beauzée, Gaultier, et d'autres ont posé les bases nécessaires pour l'analyse systématique des structures syntaxiques. Leur travail met également en avant les liens historiques entre recherche linguistique et l'enseignement des langues.

Avant l'ère informatique, des corpus annotés manuellement ont été créés pour fournir des informations linguistiques complémentaires aux textes. Avec la numérisa-

tion, les corpus sont devenus scientifiquement échantillonnés et numérisés, permettant une extraction des données plus rapide et efficace. Des œuvres comme l'annotation interlinéaire de Müller sur le Sanskrit et la concordance lemmatisée de Strong pour la Bible constituent des exemples significatifs de cette période. Ces travaux ont eu une importance critique pour la documentation linguistique et l'anthropologie, en facilitant une meilleure compréhension des langues et des cultures, et ont posé les bases des travaux futurs en humanités numériques.

Plus tard, le développement du Brown Corpus et son influence sur la création du Penn Treebank ont marqué un tournant dans l'histoire des treebanks. Le Penn Treebank a introduit une annotation systématique et a servi de base pour de nombreux treebanks ultérieurs. Ce corpus, couvrant plus de 4,5 millions de mots, a constitué une entreprise considérable. Le processus d'annotation a été mené à l'aide de logiciels de traitement automatique et de corrections manuelles. Le Talbanken suédois et le Penn Treebank anglais ont ouvert la voie à l'évolution des treebanks. Ces projets ont combiné le traitement automatique et la correction manuelle pour atteindre un équilibre entre échelle et précision.

Les projets Negra/Tiger et le Prague Dependency Treebank ont été des précurseurs du changement de paradigme vers la dépendance en TAL. Cette évolution a été influencée par plusieurs facteurs, notamment la complexité réduite et l'intuitivité des structures de dépendance, ainsi que par l'efficacité computationnelle accrue du traitement dépendanciel.

Les *shared task*, et la recherche de parsers performants, ont conduit de nombreux chercheurs à formuler des standards pour encoder les structures dans des schémas communs, tendant vers plus d'universalité. Ces standards et schémas ont permis à une large communauté de se constituer et de mobiliser des efforts communs pour développer des treebanks dans de nombreuses langues.

Avec l'émergence des technologies modernes de TAL qui utilisent des représentations continues, le besoin de structures syntaxiques explicites a diminué. Cependant, les treebanks restent pertinents pour l'évaluation des technologies linguistiques et comme ressources importantes pour la recherche linguistique. Ils sont cruciaux pour

de nombreuses langues moins largement couvertes et pour la documentation de genres linguistiques variés. Malgré les avancées technologiques, les treebanks continuent de jouer un rôle crucial en recherche linguistique et en TAL. Ils offrent une richesse de données structurées, et permettent d'améliorer l'explicabilité de nouvelles méthodes.

Dans le chapitre 3 nous nous intéressons aux pratiques de développement des treebanks.

Dans le domaine de la linguistique computationnelle, les treebanks sont des ressources clés, représentant des corpus annotés où les phrases sont liées à leurs structures syntaxiques, généralement sous forme d'arbres. Ce chapitre vise à cartographier ces méthodologies de développement de treebanks, en comprenant leurs bases théoriques et en identifiant les défis rencontrés. Le développement d'un treebank commence souvent par la collecte et la sélection du corpus. Selon l'intégrité des données brutes et les ressources disponibles, il peut être crucial de choisir judicieusement les textes les plus représentatifs. Deux défis majeurs se présentent : la transcription de sources audio en texte écrit, et la conversion de documents écrits stockés dans des formats complexes en texte brut.

Pour les corpus oraux, les fichiers audio doivent être transcrits dans un format basé sur les caractères. Cette transcription peut être phonétique ou orthographique et présente de nombreuses difficultés. La tokenisation, étape qui consiste à délimiter les plus petites unités pour l'annotation syntaxique, utilise souvent l'espace blanc et la ponctuation comme indicatifs des frontières des mots. Cette méthode peut varier considérablement selon la langue et présenter différents niveaux de complexités.

Les langues riches en ressources peuvent bénéficier d'arbres syntaxiques existants et de modèles de parsing entraînés sur ces treebanks. Cependant, la qualité de l'annotation humaine reste cruciale pour la cohérence et l'adhésion aux critères syntaxiques. Bien que de nombreuses langues disposent désormais de ce type de ressources, l'influence du genre de texte et du domaine de spécialité présentent d'autres types de difficulté, les représentations apprises n'étant pas toujours généralisables à de nou-

veaux corpus.

Dans les scénarios sans ressources, une stratégie efficace consiste à étudier des langues voisines ou de la même famille linguistique. Les corpus parallèles, où le même texte est disponible dans plusieurs langues, offrent également une voie prometteuse. L'apprentissage par transfert implique l'utilisation de modèles formés sur une langue source pour aider dans des tâches de parsing ou d'annotation dans une langue cible. Cette adaptation peut être réalisée au moyen d'une représentation commune via un transfert délexicalisé, ainsi que par diverses méthodes de sélection de données sources appropriées, dans l'optique de les adapter à la langue cible. La représentation partagée peut également être obtenue au travers d'un alignement au niveau des mots entre les deux corpus, en constituant un corpus parallèle. Des méthodes de projection d'annotation permettent alors de transférer les annotations d'une langue source à une langue cible. Pour chacune de ces techniques, il existe un risque de biaiser l'analyse en la calquant sur celle proposée pour la langue source.

Enfin, de nouvelles méthodes proposent d'identifier des structures syntaxiques de façon non-supervisée. Ces méthodes sont généralement basées sur l'apprentissage de représentations continues, qui constitue une approche prometteuse. Les autoencodeurs, notamment les autoencodeurs récursifs profonds, jouent un rôle clé dans cette stratégie.

Le développement de treebanks peut être facilité par diverses ressources linguistiques, lorsque celles-ci sont disponibles. La lemmatisation, les traits morphologiques, et l'étiquettage morphosyntaxique facilitent une annotation syntaxique. L'intégration de ressources linguistiques existantes, telles que les dictionnaires, les livres de grammaire et les grammaires formelles, peut améliorer considérablement la qualité et la cohérence des annotations des treebanks.

En conclusion, ce chapitre souligne les complexités de l'annotation syntaxique et les défis liés aux différents scénarios quant à la présence ou l'absence de ressources. Il rend compte des nombreux axes de variabilités liés aux corpus notamment la langue, le genre de texte et le domaine, qui sont des facteurs majeurs limitant la généralisation des analyses d'un corpus à l'autres. Il présente certaines des méthodes employées pour

répondre à ces difficultés.

Le chapitre 4 met en avant les difficultés inhérentes à l'activité de développement de treebanks et de leur documentation, en les illustrant au travers de cas pratiques. Il se concentre sur les problématiques liées à la constitution des guides d'annotation, documents techniques décrivant les principes et conventions utiles à l'annotation, en illustrant ces problématiques au travers d'examples de traitement des expressions multi-mots dans le schéma d'annotation d'Universal Dependencies (UD) ainsi que par l'étude du pidgin-créole Naija et son analyse syntaxique réalisée dans le schéma Surface-Syntactic Universal Dependencies (SUD).

Les treebanks, sont des ressources clés en linguistique computationnelle. Leur développement implique des annotations linguistiques précises et cohérentes. Les directives d'annotation jouent un rôle crucial dans ce processus, car elles guident les annotateurs dans l'interprétation et la structuration des données linguistiques. Ces directives, documentées dans des guides, doivent être claires, détaillées au travers d'examples et adaptées aux utilisateurs potentiels, qui sont principalement les annotateurs, et les principles sous-tendant ces directives doivent être présentés afin d'en motiver les objectifs. Ces questions prennent encore davantage d'importance lorsque le schéma d'annotation à une visée multilingue et est appliqués à des projets variés. Les directives doivent également tenir compte des critères tels que la cohérence, la facilité d'apprentissage par des systèmes informatiques et la capacité à gérer les ambiguïtés linguistiques. Le développement de ces directives est souvent itératif et interdépendant de l'annotation à proprement parler.

Les expressions multi-mots présentent des défis particuliers pour l'annotation. Ces constructions, qui regroupent en fait des unités variées allant de structures syntaxiquement régulières à des unités sémantiquement non compositionnelles, nécessitent une approche d'annotation permettant de capturer à la fois leur nature complexe et leur fonction à l'intérieur de la phrase. Le schéma UD a été ajusté pour mieux traiter ces expressions, en introduisant des étiquettes spécifiques et des règles pour gérer la diversité et la complexité de ces expressions. Cette adaptation permet une meilleure

analyse et compréhension des structures linguistiques complexes, tout en conservant une cohérence avec les principes généraux de UD.

Le pidgin-créole Naija, parlé au Nigeria, offre un terrain d'étude intéressant pour la linguistique computationnelle en raison de sa structure grammaticale unique et de son importance sociolinguistique. Le projet NaijaSynCor vise à développer un treebank en utilisant le schéma SUD, une variante de UD décrivant la structure de surface et adaptée au contexte d'annotation du Naija. Ce travail présente les différentes phases d'annotation du treebank, jusqu'à la descriptions des constructions syntaxiques spécifiques au Naija, telles que les constructions clivées, et les constructions verbales sérielles. L'utilisation de SUD facilite l'annotation et l'analyse de ces structures, tout en permettant une conversion automatique vers UD pour une compatibilité avec les ressources linguistiques existantes. L'annotation est évaluée au travers d'une mesure d'accord inter-annotateur, qui permet d'en mesurer la stabilité, et l'apprentissage d'un modèle d'analyse syntaxique automatique (*parsing*) qui nous donne une idée de la qualité que l'on peut espérer obtenir en analysant de futurs textes en Naija. Ce projet contribue non seulement à la recherche en linguistique computationnelle, mais aussi à la reconnaissance et à l'étude du Naija en tant que langue à part entière.

Les avancées dans le domaine de la linguistique computationnelle, notamment dans le développement des treebanks, l'annotation des expression multi-mots et l'étude de langues moins documentées telles que le Naija, sont essentielles pour la compréhension des structures linguistiques complexes et leur traitement informatique. Ces efforts contribuent non seulement à la recherche scientifique, mais aussi à la valorisation et à la préservation de la diversité linguistique.

Le chapitre 5 aborde le développement collaboratif des treebanks et les outils d'annotation associés. L'évolution constante des directives d'annotation pendant le processus de développement des treebanks nécessite des outils appropriés pour faciliter la mise à jour des annotations. Les campagnes d'annotation impliquent souvent plusieurs annotateurs, avec des annotations parfois divergentes. Les outils d'annotation doivent donc permettre de comparer ces divergences pour en comprendre l'origine et parvenir

à une annotation finale unifiée.

Des fonctionnalités utiles supplémentaires incluent des outils de requête pour trouver des exemples pour la documentation, avoir une vue d'ensemble des structures syntaxiques présentes ou absentes dans le treebank, et repérer les incohérences et les structures qui semblent aberrantes au premeir abord. De nombreux outils d'annotation existent, mais aucun ne semblait regrouper toutes ces fonctionnalités. C'est pourquoi nous avons travaillé à l'intégration de deux outils complémentaires : Arborator et Grew. Arborator est un outil d'annotation de treebanks de dépendances en ligne, collaboratif et graphique, largement utilisé. Grew est conçu pour interroger et réécrire des graphes à l'aide de règles formelles et a été utilisé pour développer des treebanks syntaxiques et des banques de graphes sémantiques.

Nous présentons Arborator-Grew, un outil d'annotation collaboratif pour le développement de treebanks. Sa conception a été guidée par les problèmes communs rencontrés durant le processus d'annotation de treebanks syntaxiques et leur exploitation. Arborator-Grew ouvre de nouvelles perspectives pour créer, réviser et maintenir collectivement des treebanks syntaxiques et des banques de graphes sémantiques.

Arborator-Grew s'inscrit dans une longue tradition de travail dédié à faciliter la visualisation, l'édition, l'interrogation et la réécriture d'arbres de dépendances. Des outils tels que Brat, WebAnno, Annis, et bien d'autres encore ont été développés pour faciliter divers aspects de l'annotation et de la gestion de treebanks. Arborator-Grew se distingue en intégrant à la fois la création collaborative de treebanks et les fonctionnalités de requête et de réécriture de graphes.

L'architecture d'Arborator-Grew se compose de trois parties logicielles interagissant via des interfaces REST. Le stockage des données repose sur le système de stockage Grew basé sur Ocaml, avec une API acceptant des requêtes JSON. Le back-end et la persistance utilisateur sont gérés par une application Flask, qui contrôle l'accès aux données et gère la logique logicielle. L'interface utilisateur est écrite en VueJS, un framework JavaScript qui facilite le développement d'interfaces utilisateur réactives, modulaires et flexibles.

Arborator-Grew offre de nouvelles fonctionnalités pour la construction et l'amélioration

des treebanks. Nous nous concentrons sur deux aspects principaux : la création collaborative de treebanks en classe et la fouille d'erreurs dans un treebank existant. Ces deux applications sont facilitées par le requêtage du treebank au moyens de patrons. L'intégration de Grew permet de requêter des patrons syntaxiques dans un treebank et de corriger directement les arbres dans l'interface graphique. De plus, des règles peuvent être mises en place pour transformer automatiquement certaines erreurs systématiques. Cette fonctionnalité facilite la correction automatique des erreurs, l'adaptation des treebanks aux directives mises à jour et la conversion des treebanks dans différents schémas d'annotation.

Arborator-Grew a été utilisé avec succès dans divers contextes, tels que l'enseignement et pour réaliser des campagnes d'annotation. Plusieurs treebanks sont maintenant développés et maintenus en utilisant Arborator-Grew, démontrant qu'il répond aux besoins de la communauté. Le code source est ouvert, permettant aux chercheurs de l'adapter à leurs besoins spécifiques. Des initiatives comme Arborator-Grew-NILC, développé par une équipe travaillant sur le treebank Porttinari, ont introduit des améliorations axées sur l'expérience utilisateur, telles que des raccourcis pour augmenter l'efficacité des annotateurs et des avertissements pour signaler les annotations non conformes aux critères de validation. Ces initiatives fournissent un retour précieux sur ce qui pourrait être amélioré dans Arborator-Grew et mettent en lumière l'importance croissante de la maintenance active pour proposer des treebanks de qualités et à jour.

Les améliorations futures d'Arborator-Grew pourraient inclure des fonctionnalités ciblant la correction des erreurs, l'adaptation aux nouvelles directives d'annotation, et la validation des treebanks, qui sont devenues des pratiques courantes dans le développement de treebanks et nécessitent des outils appropriés.

En conclusion, Arborator-Grew représente une avancée significative dans le domaine de l'annotation collaborative de treebanks, intégrant des outils de requête et des fonctionnalités de réécriture de graphes pour faciliter la création, la maintenance et l'amélioration des treebanks syntaxiques et sémantiques. Son approche innovante et ses fonctionnalités avancées répondent aux besoins évolutifs des linguistes et des

développeurs de treebanks, tout en ouvrant de nouvelles voies pour l'annotation et l'analyse linguistique.

Dans le chapitre 6, nous examinons les propriétés globales des arbres de dépendance syntaxique, en mettant l'accent sur la loi de Menzerath-Altmann (MAL) et son interaction avec le phénomène de déplacement des constituants lourds (*Heavy Constituent Shift* or HCS). L'étude se base sur des analyses quantitatives menées sur diverses langues dans le cadre des projets Universal Dependencies (UD) et Surface-Syntactic Universal Dependencies (SUD).

Dans la prmeière section, nous examinons les interactions entre la Loi de Menzerath-Altmann (MAL) et le Heavy Constituent Shift (HCS), phénomène de déplacement des constituants longs. En utilisant les données multilingues des treebanks SUD (Surface-Syntactic Universal Dependencies), nous évaluons l'hypothèse selon laquelle les constituants plus lourds tendent à être déplacés vers la fin des phrases, conformément au HCS, et comment cette hypothèse intéragi avec la loi de Menzerath-Altmann, qui stipule que dans une construction linguistique, plus la construction est grande, plus ses unités constituantes sont petites. Nous analysons les structures syntaxiques dans une gamme variée de langues pour comprendre comment ces deux principes interagissent et influencent la structure des phrases.

La Loi de Menzerath-Altmann (MAL) et le principe de Heavy Constituent Shift (HCS) sont deux concepts fondamentaux en linguistique qui traitent de l'organisation des structures linguistiques. La MAL stipule que dans une construction linguistique, plus la construction est grande, plus ses constituants sont courts. Le HCS, quant à lui, observe que dans plusieurs langues, notamment en anglais, les constituants plus lourds (plus longs ou complexes) sont souvent repositionnés vers la fin de la phrase. Les études sur la loi de Menzerath-Altmann se sont principalement concentrées sur la vérification de cette loi dans différentes constructions linguistiques et langues, ainsi que sur l'interprétation de ses paramètres. Des recherches ont examiné comment la longueur des mots, des syllabes ou des clauses est influencée par la taille globale

de la construction linguistique. De plus, la MAL a attiré l'attention dans d'autres disciplines, telles que la biologie. Le Heavy Constituent Shift (HCS)est un phénomène syntaxique bien connu, basé sur le concept de constituants lourds qui contiennent plus de mots que les constituants légers. Selon le HCS, ces constituants lourds tendent à être déplacés vers la fin de la clause. Cette tendance a été observée pour la première fois dans les langues SVO comme le français et l'anglais.

Notre étude vise à combiner la MAL et le HCS pour analyser leur interaction dans les langues naturelles. Nous utilisons les données des treebanks SUD pour examiner des structures syntaxiques spécifiques et évaluer notre hypothèse qu'il existe une interaction entre ces deux phénomènes.

Nous avons filtré des clauses spécifiques dans les treebanks SUD, en nous concentrant sur les verbes et leurs compléments. Les mesures de la taille des constituants ont été prises en compte pour analyser l'interaction entre la MAL et le HCS. Nos résultats montrent que dans les 80 langues étudiées, l'hypothèse $a < c$ (où a et c représentent la taille moyenne des constituants) est vérifiée, indiquant une interaction régulière entre la MAL et le HCS. D'après cette études, cette interaction semble être un universel linguistique.

Cette étude pilote montre que l'interaction entre la MAL et le HCS semble être une caractéristique universelle des langues naturelles. Notre approche offre de nouvelles perspectives pour connecter la MAL à des discussions linguistiques traditionnelles telles que le HCS et pourrait s'étendre à d'autres phénomènes linguistiques. Des recherches futures pourraient explorer davantage cette interaction dans divers contextes linguistiques et avec différents types de données.

Le deuxième volet du chapitre six explore l'application de la Loi de Menzerath-Altmann (MAL) à des unités linguistiques plus précises d'un point de vue syntaxique. Nous introduisons le concept de segment linéaire de dépendance (en anglais *Linear Dependency Segment* ou LDS) comme une unité intermédiaire entre la clause et le mot. Cette unité tient compte de la structure linéaire et de la structure de dépendance au sein de la phrase. Nous examinons si cette nouvelle unité est pertinente pour appliquer la loi de Menzerath-Altmann au niveau syntaxique, en utilisant pour cela des données

provenant de deux treebanks tchèques.

La Loi de Menzerath-Altmann (MAL) établit une relation inverse entre la taille des unités linguistiques et celle de leurs constituants. Cependant, cette loi présente des difficultés lorsqu'on l'applique aux niveaux supérieurs de la hiérarchie linguistique, tels que les relations entre la longueur des phrases (en clauses) et la longueur des clauses (en mots). Pour résoudre ces problèmes, nous proposons une nouvelle unité, le segment linéaire de dépendance, qui se situe entre la clause et le mot.

Nous définissions le Segment Linéaire de Dépendance (SLD) comme la plus longue séquence possible de mots au sein d'une même clause, où chaque mot est à la fois un voisin linéaire (adjacent dans la phrase) et un voisin syntaxique (relié par un lien de dépendance syntaxique). Cette définition permet de segmenter de manière univoque chaque clause en SLD.

Selon la MAL, on s'attend à ce que les phrases plus longues (mesurées en nombre de clauses) contiennent des clauses plus courtes (mesurées en nombre de SLDs). Cette hypothèse s'appuie sur le fait que les liens de dépendance qui ne respectent pas la linéarité de la phrase sont plus difficiles à traiter.

Nous avons utilisé deux treebanks en tchèque, le Czech-PDT du corpus Universal Dependencies et le FicTree, tout deux convertis selon le schéma d'annotation Surface-Syntactic Universal Dependencies (SUD), un schéma voisin mais davantage centré sur la syntaxe de surface. Nous avons analysé un total de 86 266 phrases, en excluant les phrases sans prédicat. Les données montrent une bonne adéquation avec la MAL, en particulier pour le treebank combiné (PDT et FicTree), avec un coefficient de détermination $R^2 = 0.9803$. Nous observons également que la valeur du paramètre a est proche de la longueur moyenne des clauses en nombre de segment linéaire de dépendances pour les phrases constituées d'une seule clause, offrant ainsi une interprétation claire de ce paramètre.

Les résultats indiquent que cette nouvelle unité, le segment linéaire de dépendance, peut être considérée comme une unité linguistique significative pour modéliser la loi de Menzerath-Altmann au niveau syntaxique. L'unité permet d'éviter certains problèmes préalablement rencontrés avec la mesure de la longueur des clauses en mots. De plus,

la longueur des clauses mesurée avec cette unité est en accord avec la capacité de la mémoire à court terme, ce qui est l'une des explications théoriques de la loi de Menzerath-Altmann.

Cette étude pilote, bien que limitée dans sa portée, ouvre la voie à de futures recherches. Elle démontre l'intérêt d'analyser les segments linéaires de dépendance comme *construct* dans la loi de Menzerath-Altmann, et invite à examiner leurs fréquences et longueurs en relation avec les lois de distribution couramment utilisées pour modéliser des propriétés linguistiques similaires. De plus, une correspondance potentielle entre les segments linéaires de dépendance et le phénomène observé de minimisation de la distance de dépendance mérite une inspection plus approfondie.

Dans la section 3 du chapitre 6 nous nous intéressons à la comparaison entre arbres de dépendance syntaxiques et artificiels, au travers de diverses propriétés. L'étude proposée présente une comparaison entre des arbres de dépendance syntaxique issus de données linguistiques réelles et des arbres générés artificiellement. Nous explorons les contraintes linguistiques et formelles sur les arbres de dépendance syntaxique pour comprendre ce qui rend certaines structures plus probables que d'autres. En utilisant des arbres générés de manière aléatoire et à l'aide de contraintes, nous cherchons à distinguer les contraintes formelles des contraintes linguistiques ou cognitives. L'étude se concentre sur cinq métriques : longueur, hauteur, arité maximale, distance moyenne de dépendance et poids moyen du flux, et analyse également la distribution des configurations locales de nœuds à l'intérieur des arbres.

Dans cette étude, nous comparons les arbres de dépendance syntaxique, qui représentent la structure des langues naturelles, à des arbres générés artificiellement. L'objectif est de comprendre quelles structures sont linguistiquement plausibles et lesquelles ne le sont pas. En observant les arbres syntaxiques issus de données réelles, nous identifions les caractéristiques uniques de ces structures. En parallèle, la génération d'arbres aléatoires sur lesquelles sont progressivement ajoutées des contraintes, nous permet d'étudier les effets produits par l'ajout de ces contraintes.

Nous utilisons cinq métriques pour analyser les propriétés des arbres de dépendance : longueur, hauteur, arité maximale, distance moyenne de dépendance (DMD) et poids moyen de flux. Ces propriétés interviennent chacune dans les stratégies de linéarisation, c'est-à-dire la façon dont les mots sont ordonnés dans les phrases. De nombreux travaux quantitatifs récents se sont concentrés sur la longueur de dépendance et sa minimisation dans de nombreuses langues naturelles. En complément, nous utilisons le poids moyen du flux, une métrique qui capture l'importance des dépendances disjointes dans la phrase, généralement utilisé comme facteur de complexité.

Nous examinons également les configurations locales dans les arbres de dépendance, en extrayant et comparant la proportion de toutes les configurations possibles de bigrammes et trigrammes.

Les propriétés des arbres syntaxiques sont comparées à celles de trois types d'arbres, des arbres originaux aléatoirement linéarisés, des arbres originaux linéarisés de façon à minimiser la distance de dépendance, et des arbre entièrement aléatoires, que ce soit du point de vue de la structure, ou du point de vue de la linéarisation. Ces procédures de génération d'arbres artificiels nous donnent des outils pour analyser comment différentes stratégies de génération affectent les propriétés des arbres observés.

Pour chaque paire de propriétés, nous avons mesuré le coefficient de corrélation de Pearson. Les résultats montrent que certaines propriétés sont fortement corrélées, indépendamment du fait que les arbres soient syntaxiques ou artificiels.

Pour la distance moyenne de dépendance et la poids moyen du flux, nous notons une corrélation importante, et ce quel que soit le type d'arbre observé, ce qui suggère que l'augmentation du poids moyen de flux tend à créer des dépendances plus longues. Cette corrélation est toutefois encore plus marquée dans les arbres artificiels, surtout ceux optimisés pour minimiser la distance de dépendance. En outre, la corrélation entre la longueur et la hauteur des arbres est forte, tant dans les structures originales que dans les structures aléatoires. Cela indique que la relation entre ces deux propriétés n'est pas uniquement motivée par des facteurs linguistiques. Concernant la corrélation entre la distance moyenne de dépendance et la hauteur de l'arbre, nous observons que

la corrélation est forte dans les arbres artificiels, tandis qu'elle l'est moins pour les arbres syntaxiques. Cela suggère qu'une relation plus complexe entre la hauteur et la distance moyenne de dépendance pourrait exister dans les données réelles, indiquant qu'il existe une relation qui ne peut pas être capturée linéairement.

Nous avons ensuite examiné la distribution des configurations locales en extrayant des trigrammes et en analysant leurs relations de dépendance. Des configurations telles que b←a→c et a→b→c ont été analysées afin de constater les différences entre les structures locales des arbres syntaxiques et des arbres générés aléatoirement.

Dans les arbres syntaxiques, la configuration b←a→c est préférée, ce qui peut aider à minimiser les distances de dépendance. Cette configuration conduit souvent à des sous-arbres *équilibrés*, qui sont optimaux pour minimiser les distances de dépendance. Cependant, les arbres syntaxiques montrent une forte préférence pour les configurations en *bouquet*, qui sont moins fréquentes dans les arbres artificiels.

Dans le cas de la langue japonaise, une proportion élevée de configurations en *zigzag* a été observée, probablement due à la segmentation utilisée dans les treebanks japonais. Ces résultats suggèrent que certaines stratégies de linéarisation pourraient être influencées par des contraintes linguistiques ou cognitives spécifiques.

Cette étude offre un aperçu précieux des contraintes qui influencent la formation des arbres de dépendance syntaxique. En comparant les arbres syntaxiques à des arbres générés artificiellement avec différentes contraintes, nous avons pu identifier des relations entre différentes propriétés des arbres et comment ces relations varient en fonction de la nature des arbres. Ces découvertes ouvrent la voie à de futures recherches pour explorer plus en détail les contraintes linguistiques et cognitives qui sous-tendent la structure des langues naturelles. La mise à disposition des algorithmes de générations d'arbres aléatoires permettra à d'autres de s'emparer des méthodes pour prolonger ce travail.

Le chapitre 7 explore un domaine fascinant de la linguistique computationnelle : l'induction de structures syntaxiques à partir de corpus de textes bruts, c'est-à-dire sans annotations. Cette tâche, consiste à identifier et formaliser les relations entre les

mots dans des phrases sans s'appuyer sur des données annotées préalablement. C'est une sorte de déchiffrage de la structure cachée du langage, en utilisant des méthodes statistiques et informatiques.

Pour aborder ce défi, le chapitre présente deux perspectives principales : la définition des unités syntaxiques basées sur les frontières, et celle basées sur les relations. Les frontières ici désignent les points dans un texte où une unité syntaxique (comme un mot ou un constituant) commence ou se termine. Les liens, quant à eux, concernent les relations entre les mots, comme dans les arbres de dépendance où chaque mot est connecté à son gouverneur dans la phrase.

Un point clé du chapitre est la notion de *frontiéritude*, qui décrit le degré de séparation entre les unités syntaxiques. Cette idée est cruciale car dans une langue, les frontières entre les mots ou les phrases peuvent être plus ou moins marquées, certaines unités étant plus ou moins autonomes ou cohésives. Nous discutons également des moyens permettant de traduire ces concepts théoriques en mesures concrètes, qui peuvent être utilisés pour extraire des treebanks ces informations.

L'un des aspects les plus innovants abordés est l'utilisation d'une mesure d'autonomie basée sur l'entropie pour identifier les unités syntaxiques, ce qui correspond à une vision des unités syntaxiques basée sur la notion de frontière. L'entropie permet de mesurer l'incertitude ou la diversité dans les choix des mots qui suivent ou précèdent un point donné dans une phrase. Nous proposons d'utiliser les points de haute entropie comme indication de frontières d' unités syntaxiques, une hypothèse inspirée par des théories linguistiques antérieures appliquées à la segmentation en mots.

Le chapitre présente ensuite une étude de cas sur le français, en utilisant cette mesure d'entropie pour extraire des fragments syntaxiques à l'intérieur des phrases. Ces fragments syntaxiques sont comparés à une baseline (une méthode simple utilisée comme point de comparaison dans une évaluation) dans laquelle les fragments sont générés de façon aléatoire, afin d'en évaluer la qualité. Nous constatons que les fragments extraits avec cette méthode, correspondent plus étroitement à certains types de structures syntaxiques qu'à d'autres, selon la manière dont les arbres de dépendance sont annotés.

Les résultats sont prometteurs, mais le chapitre reconnaît aussi un certain nombre de limites et de défis rencontrés. Par exemple, afin d'estimer l'entropie, la méthode requière un corpus de grande taille, ce qui affecte la précision des prédictions. De plus, les fragments obtenus ne sont pas suffisants pour reconstituer une structure syntaxique complète, mais ne représente qu'une structure partielle. Cependant, le chapitre soulève des questions intéressantes sur la nature des unités syntaxiques et propose des pistes pour de futures recherches.

En conclusion, ce chapitre formule un pont entre la théorie linguistique et l'application pratique dans le domaine de la linguistique computationnelle. Il offre une perspective intéressante sur la façon dont nous pouvons utiliser des méthodes statistiques et informatiques pour découvrir des structures à l'intérieur des textes, ouvrant ainsi la voie à de nouvelles découvertes dans la compréhension des langues humaines.