Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Extraction de fragments syntaxiques en français à partir d'une mesure d'autonomie basée sur l'entropie

Marine Courtin[1]
(1) LPP - Laboratoire de Phonétique et Phonologie - UMR 7018 , 19 rue des Bernardins 75005 Paris, France
marine.courtin@sorbonne-nouvelle.fr

RÉSUMÉ

Dans cet article nous nous intéressons à la prédiction du caractère syntaxique ou non d'une séquence de tokens dans des corpus du français. En particulier, nous comparons une méthode d'extraction de fragments syntaxiques identifiés au moyen d'une mesure d'autonomie basée sur l'entropie à une méthode de référence qui extrait des fragments aléatoires. Les résultats semblent indiquer que les fragments ainsi extraits sont bien plus souvent des unités syntaxiques que les fragments aléatoires. Une telle méthode pourrait être utilisée dans des travaux ultérieurs afin de proposer une induction non-supervisée de structures de dépendances syntaxiques.

ABSTRACT

Mining French syntactic fragments using an entropy-based autonomy measure.

In this paper we investigate how sequences of tokens can be identified as syntactic fragments in French corpora. We compare two methods for extracting syntactic fragments : a random baseline and a method that uses an entropy-based autonomy measure to induce syntactic fragments. Results suggest that the proposed method improves the prediction accuracy of sequences that are syntactic units, as compared to the baseline. These findings could be used in further works for unsupervised syntactic dependency induction.

MOTS-CLÉS : fragments syntaxiques, autonomie, analyse syntaxique non-supervisée.

KEYWORDS: syntactic fragments, autonomy, unsupervised parsing.

1 Objectifs

Cet article est une contribution à l'induction non-supervisée de structures syntaxiques. Nous nous plaçons dans le cadre de la syntaxe de dépendance, une théorie syntaxique introduite par (Tesnière, 1959), et cherchons à décider si des segments contigus à l'intérieur d'un énoncé forment des fragments connexes de l'arbre de dépendance. (Gerdes & Kahane, 2011) ont montré que la structure de connexion d'un énoncé pouvait être entièrement définie à partir de l'ensemble des fragments de cet énoncé, fragments qui sont définis notamment par leur capacité à être autonomisés. Nous proposons donc d'utiliser une mesure d'autonomie basée sur l'entropie afin d'induire des fragments syntaxiques. Cette mesure d'autonomie a été utilisée avec succès par le passé pour identifier des unités plus petites telles que des mots, mais nous cherchons à savoir s'il est possible d'extraire des unités plus grandes pour une tâche d'induction de structure syntaxique. Notre hypothèse est que l'évolution de l'entropie à travers la phrase peut nous permettre de faire des prédictions éclairées sur les frontières entre unités syntaxiques, et pourrait se révéler utile pour décider quelles séquences sont des unités syntaxiques et

Volume 2 : 23e REncontres jeunes Chercheurs en Informatique pour le TAL (RECITAL), pages 15–27.

quelles séquences ne le sont pas.

Ce travail s'inscrit dans la lignée d'autres travaux qui se sont penchés sur des tâches comme l'analyse syntaxique non-supervisée, l'induction de structures syntaxiques ou encore la recherche d'informations syntaxiques dans des représentations vectorielles denses (ou plongements) au moyen de « structural probes » (Hewitt & Manning, 2019).

Computationnellement, la méthode que nous proposons est plus légère que ces derniers, qui nécessitent d'entraîner des modèles plus lourds comme BERT (Devlin *et al.*, 2019) et ELMO (Peters *et al.*, 2018), ce qui requiert de très grand corpus d'entraînement et la mobilisation de lourdes infrastructures de calcul coûteuses en ressources (Strubell *et al.*, 2019). De plus, nous espérons que l'autonomie basée sur l'entropie soit plus facilement interprétable, puisqu'elle est associée à un solide ancrage théorique en linguistique notamment avec l'hypothèse de Harris (Harris, 1955) qui propose qu'un plus grand paradigme de successeurs ou prédecesseurs à une position entre deux tokens (dans son cas des caractères), indique la présence d'une frontière linguistique (pour lui des frontières entre morphèmes). Cette théorie nous semble assez naturellement adaptable aux frontières entre unités syntaxiques, même si nous pouvons nous demander si l'entropie constituera une information suffisante dans ce cas, puisque la variabilité des tokens est bien plus grande que la variabilité des caractères.

D'autres travaux encore cherchent à établir des liens entre des prédicteurs et la présence de relations de dépendance, c'est le cas par exemple de (Futrell *et al.*, 2019) qui remarquent un lien entre l'information mutuelle pour une paire de mots, et la présence d'une relation de dépendance entre eux. L'information mutuelle étant liée à l'entropie, il nous paraît d'autant plus intéressant d'utiliser une mesure d'autonomie qui soit basée sur cette dernière.

Nous commencerons par présenter en section 2 la mesure d'autonomie utilisée pour prédire la nature syntaxique ou non d'une unité. En section 3 nous présenterons le corpus qui nous servira à entraîner le modèle d'estimation de l'autonomie, ainsi que les corpus arborés sur lesquels seront évaluées nos prédictions. En 4 nous décrirons brièvement le processus d'extraction des fragments aléatoires qui nous servira de méthode de référence. Enfin, en 5 nous présenterons nos premiers résultats.

2 Autonomie et unités syntaxiques

2.1 Mesure d'autonomie

La mesure d'autonomie que nous utilisons est décrite dans (Magistry, 2013). Elle consiste à considérer comme autonome une unité dont les éléments seraient cohésifs, et dont les frontières seraient difficilement prédictibles car situées à des positions de forte entropie.

La mesure d'autonomie est construite de la façon suivante : tout d'abord **l'entropie de branchement** est évaluée à chaque position inter-mot. Cette entropie de branchement permet de rendre compte de la diversité des tokens qui peuvent succéder ou précéder un certain contexte. On calcule ensuite **la variation d'entropie de branchement** qui est obtenue en soustrayant l'entropie de branchement de la position précédente à l'entropie de branchement de la position actuelle. Cette mesure permet d'observer à quel point l'entropie augmente ou diminue en ajoutant un nouveau token.

L'autonomie d'un n-gramme est ensuite calculée en sommant les variations de l'entropie de branche-

ment [1] depuis un parcours gauche-droite et un parcours droite-gauche du texte. Plus un n-gramme aura une autonomie élevée, plus ses frontières auront de fortes entropies comparées aux positions inter-mots, et plus il sera probable que le n-gramme constitue une unité syntaxique.

Formellement, le calcul pour arriver à cette autonomie est réalisé ainsi (nous reprenons toujours (Magistry, 2013)) :

Étant donné un n-gramme $x_{0..n} = x_{0..1}x_{1..2}...x_{n-1..n}$ avec pour contexte gauche $X_{\rightarrow}$, l'entropie de branchement droite est définie comme :

$$h_{\rightarrow}(x_{0..n}) = H(X_{\rightarrow}|x_{0..n})$$

$$h_{\rightarrow}(x_{0..n}) = -\sum_{x \in X_{\rightarrow}} P(x|x_{0..n}) log P(x|x_{0..n}).$$

Pour l'entropie de branchement gauche on note $X_{\leftarrow}$ le contexte droit de $x_{0..n}$, ce qui nous donne :

$$h_{\leftarrow}(x_{0..n}) = H(X_{\leftarrow}|x_{0..n})$$

.

À partir de l'entropie de branchement pour les n-grammes $x_{0..n}$ et $x_{0..n-1}$ la variation d'entropie dans les deux directions peut être calculée :

$$\delta h_{\rightarrow}(x_{0..n}) = h_{\rightarrow}(x_{0..n}) - h_{\rightarrow}(x_{0..n-1})$$

$$\delta h_{\leftarrow}(x_{0..n}) = h_{\leftarrow}(x_{0..n}) - h_{\leftarrow}(x_{1..n})$$

Après avoir appliqué la normalisation mentionnée dans la note 1, l'autonomie du n-gramme $x_{0..n}$ est formée :

$$a(x_{0..n}) = \widetilde{\delta} h_{\leftarrow}(x_{0..n}) + \widetilde{\delta} h_{\rightarrow}(x_{0..n})$$

Cette méthode assignant une autonomie à chaque n-gramme, il est ensuite possible de calculer le score d'une segmentation en additionnant pour chaque n-gramme le produit de son autonomie et de sa taille (en terme de tokens). On a donc une méthode qui nous permet de classer les différentes segmentations d'après leur score global, et un score d'autonomie pour chaque n-gramme.

2.2 Unités syntaxiques

Cette mesure d'autonomie a été pensée pour identifier des mots, mais nous pensons qu'il est possible d'utiliser la même logique pour identifier d'autres unités plus grandes, qui seraient de nature syntaxique. Plus précisément, nous cherchons à identifier des séquences de tokens qui forment une partie connexe dans la structure de dépendance, c'est-à-dire des catenas (Osborne *et al.*, 2012).

1. Une normalisation est également effectuée pour centrer la mesure sur 0 pour chaque taille de n-gramme, afin que les n-grammes plus courts ne soient pas favorisés.

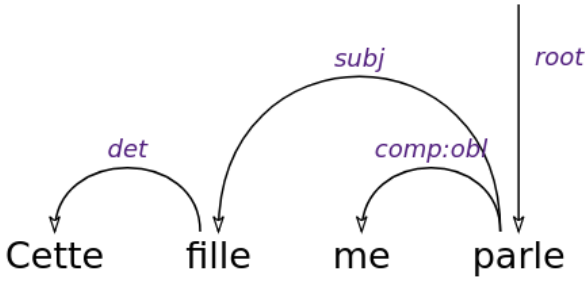

FIGURE 1 – Arbre de dépendance pour l'énoncé « Cette fille me parle »

Ainsi, si nous prenons pour exemple l'énoncé « Cette fille me parle » dont la structure de dépendance peut être représentée comme en figure 1, nous pouvons identifier 10 catenas : (Cette), (fille), (me), (parle), (Cette fille), (fille parle), (me parle), (Cette fille parle), (fille me parle), (Cette fille me parle). En revanche les séquences (fille me) et (Cette fille me) ne sont pas des catenas, puisqu'elles ne constituent pas une partie connexe de la structure de dépendance.

Ce sont donc ces portions connexes de la structure de dépendance, un type d'unités syntaxiques appelées catena, que nous allons chercher à extraire par la suite.

3 Données et méthodologie

3.1 Données

Nous utilisons deux corpus du français : un corpus brut qui est utilisé afin d'entraîner le modèle d'autonomie, et un corpus arboré en dépendance, sur lequel nous poursuivrons l'entraînement du modèle d'autonomie (en utilisant uniquement le texte). Inclure le texte des corpus arborés nous permet d'assurer que le vocabulaire qui y apparaît sera bien couvert. En ce qui concerne les structures de dépendances des corpus arborés, nous les utilisons uniquement afin d'évaluer les prédictions d'unités syntaxiques en comparant les unités prédites avec la structure de référence.

Le premier de ces corpus est constitué d'oeuvres littéraires et segmenté en phrases et en tokens. Nous échantillons des sous-corpus de tailles variées afin d'étudier l'influence de la taille du corpus d'entraînement sur les prédictions. En ce qui concerne les corpus arborés, nous utilisons 6 corpus provenant du projet Universal Dependencies (Zeman *et al.*, 2020), dans la version 2.7 : FQB, GSD, ParTUT, PUD, Sequoia and Spoken. Au total, ces corpus sont constitués de 26 555 phrases et 50 9257 tokens. Ils forment un corpus hétérogène en terme de modalité et de genre, puisqu'on y retrouve de l'écrit et de l'oral, et que les genres couvrent articles de presse, notices de médicaments, wikis, blogs, textes légaux et oral transcrit.

À l'intérieur de ces corpus arborés, des noeuds qui ne correspondent pas directement à des tokens ont été introduits pour rendre compte des amalgames comme « au » à+le, ou « du » de+le. Puisque ces formes désamalgamés n'apparaîtront pas dans notre corpus d'entraînement qui est un corpus brut, nous choisissons d'appliquer une grammaire de réécriture sur ces corpus arborés en utilisant

Grew (Guillaume *et al.*, 2012), afin de rétablir les tokens d'origine en fusionnant les amalgames. [2] Un exemple de cette transformation est présenté en figure 2.

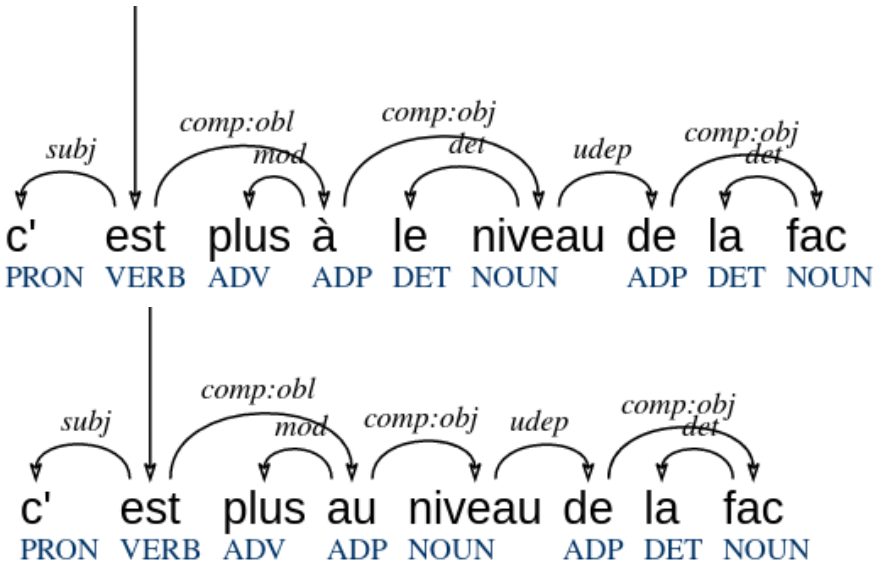

FIGURE 2 – Exemple de transformation pour fusionner un amalgame (à+le → au)

Un autre aspect qui nous semble très intéressant est l'influence du schéma d'annotation sur l'évaluation de notre méthode. Selon le schéma sélectionné, les séquences qui vont être considérées comme des unités syntaxiques vont varier, ce qui signifie que les performances du modèle seront conditionnées par ce schéma. Par exemple, il est possible qu'un fragment extrait par le modèle soit une catena dans un schéma avec des têtes fonctionnelles, mais ne le soit pas dans un schéma avec des têtes lexicales. Afin de tester à quel point ce critère est important, nous évaluons nos prédictions sur 4 versions différentes des corpus arborés, obtenues après application de grammaires de réécriture des graphes de dépendance. Les différences entre ces 4 versions peuvent être décrites de la manière suivante :

- version UD : le schéma d'annotation original pour tous les corpus arborés à l'exception de GSD et Spoken qui sont maintenus en version SUD. Dans cette version les têtes sont des éléments lexicaux et les mots fonctions sont dépendants, ce qui créé des structures généralement plus plates. Une description plus complète des différences entre schéma UD et schéma SUD peut être trouvée dans (Gerdes *et al.*, 2018).
- version SUD : le schéma d'annotation natif pour les corpus GSD et Spoken, contrairement au schéma UD, les têtes sont fonctionnelles, ce qui mène généralement à des structures plus profondes.
- version SUD+ : une version plus extrême du schéma SUD, qui lui est identique en tout aspect à l'exception des relations entre noms et déterminants qui sont inversées pour que les déterminants deviennent têtes. Les autres relations restent identiques.
- version SUD++ : une version identique à la précédente, avec en plus les anciens dépendants du nom qui sont rattachés au déterminant, pour que celui-ci domine tous les éléments à l'intérieur d'un groupe nominal.

Nous savons que ces choix sur le schéma d'annotation vont modifier de façon plus ou moins importante les structures rencontrées, ce qui aura un impact sur l'évaluation du modèle. En première observation

2. La grammaire correspondante réalisée par Bruno Guillaume est disponible ici : https://github.com/surfacesyntacticud/tools/blob/master/textform_wordform/remove_amalg_fr.grs

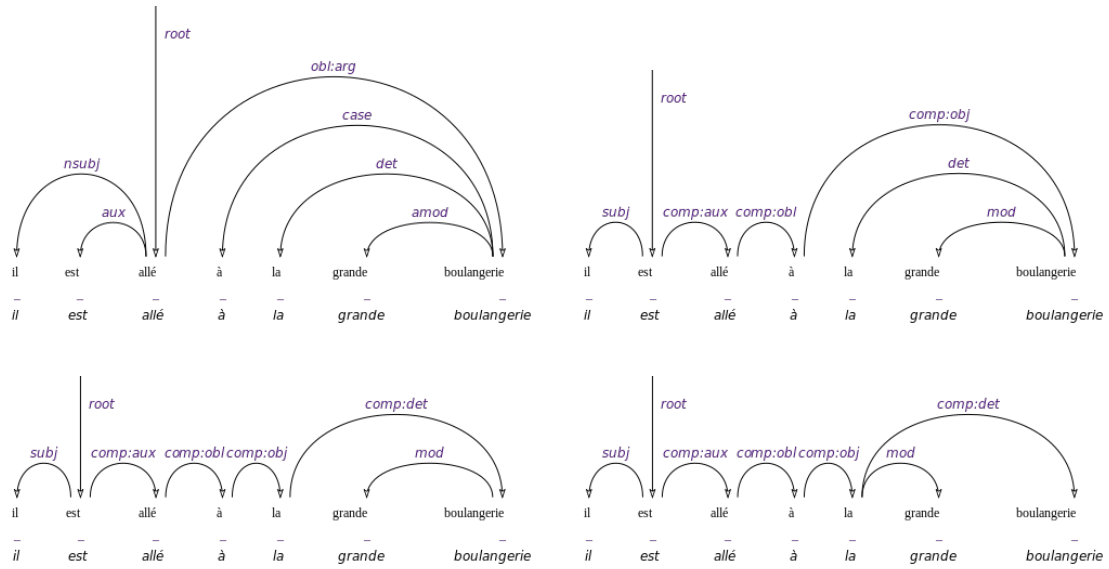

FIGURE 3 – Exemple d'annotation pour les 4 schémas (de gauche à droite puis de haut en bas : UD, SUD, SUD+, SUD++)

nous calculons la proportion de bigrammes, trigrammes et quadrigrammes qui sont des catenas dans les différentes versions des corpus arborés. Plus ces proportions seront élevées, plus il sera probable que le modèle en extrait un nombre important, sans que cela signifie nécessairement qu'il s'améliore. Les résultats dans le tableau 1 nous permettent d'observer que la version SUD+ est de loin la plus riche en catenas observées sur les bigrammes, trigrammes et quadrigrammes, et que la version UD est celle qui en présente le moins, les versions SUD et SUD++ étant similaires en proportions et se positionnant entre la version UD et la version SUD+.

Schéma	Bigrammes	Trigrammes	Quadrigrammes	Tous
UD	0,47	0,39	0,33	0,40
SUD	0,62	0,53	0,46	0,54
SUD+	0,72	0,60	0,51	0,61
SUD++	0,63	0,52	0,46	0,54

TABLE 1 – Proportion des séquences de longueur 2 à 4 qui sont des catenas dans les différents schémas d'annotation des corpus arborés.

3.2 Méthodologie

Dans la section 2 nous avons décrit la mesure d'autonomie que nous utilisons pour extraire des fragments que nous espérons être des unités syntaxiques. Cette mesure est implémentée dans l'outil ELeVE [3] (Magistry & Sagot, 2012) que nous utilisons pour obtenir les fragments.

L'outil nous permet de calculer l'autonomie pour tous les n-grammes, mais aussi d'ordonner les segmentations selon leur score global et éventuellement d'en extraire les n meilleures. Ces informations sont particulièrement intéressantes car le score d'autonomie d'une séquence ne dépend

3. L'outil est disponible en ligne ici : https://github.com/kodexlab/eleve

pas du contexte dans lequel on trouve celle-ci, puisqu'il est calculé à partir de tous ses contextes. En revanche, pour savoir si la séquence constitue bien une unité syntaxique, on aimerait que ce contexte d'apparition soit pris en compte, ce qui est le cas lorsqu'on s'intéresse au score global d'une segmentation. Ainsi une segmentation dans laquelle un seul des segments obtient un score très élevé et tous les autres segments ont des scores médiocres apparaîtra plus bas dans le classement qu'une segmentation qui permet d'obtenir plusieurs segments avec de bons scores, même si individuellement chacun de ces scores est plus faible que celui du très bon segment de la première segmentation.

Ainsi, pour chaque phrase, nous extrayons une liste de fragments, chacun associé à un unique score d'autonomie, et au rang des différentes segmentations dans lesquelles il apparaît. Nous fixons également la taille maximale des n-grammes à comptabiliser à 5 (les estimations pour des segments de longueur supérieures ne seraient pas assez fiables), ce qui nous donnera des fragments de longueur 1 à 4.

En ce qui concerne l'évaluation nous nous intéressons à deux aspects. Le premier consiste à regarder si les fragments sélectionnés constituent bien des catenas dans l'arbre de dépendance du corpus de référence. La proportion de ces fragments sélectionnés qui sont des catenas nous fournira notre score de précision. Nous mesurons aussi à quel point la structure de dépendance est couverte par les fragments extraits, c'est-à-dire quelle proportion des catenas présentes dans la structure nous avons réussi à extraire, ce qui constituera notre rappel.

4 Fragments aléatoires

Nous proposons d'induire aléatoirement des fragments afin de comparer notre méthode à une référence. Si notre hypothèse est vérifiée, nous devrions observer une meilleure compatibilité des fragments induits en utilisant la mesure d'autonomie par rapport aux fragments induits aléatoirement.

Tout d'abord, il nous semble important de préciser la différence entre deux procédés aléatoires permettant d'échantillonner des séquences de tokens :

Une **segmentation aléatoire** consiste à proposer un découpage unique de la phrase, le plus souvent en la parcourant et en attribuant une probabilité d'introduire une frontière à chaque position inter-mot.

Par opposition, une **fragmentation aléatoire** vise à induire plusieurs segmentations, ce qui permettra notamment d'avoir des fragments qui se chevauchent. C'est cette deuxième option que nous privilégions puisque sa sortie ressemblera davantage aux fragments induits.

Parmi les nombreuses façons possibles et imaginables de proposer une fragmentation aléatoire, nous proposons la suivante :

Chaque token dans la phrase est considéré comme noyau d'un fragment aléatoire. Pour ce fragment nous tirons au sort une longueur entre 2 et 4 (puisque ce sont les longueurs que nos fragments candidats peuvent adopter). Une fois la longueur du fragment définie, nous tirons au sort la position du token à l'intérieur du fragment (premier, second, troisième, quatrième). Si la position est incompatible avec la position du token dans la phrase, nous réiterons jusqu'à obtenir une position compatible.

Par exemple pour la phrase « Nous tirons une position au sort », nous pourrions avoir ce type de proposition pour « tirons » :

— longueur du fragment : 3

— position à l'intérieur du fragment : troisième (impossible), premier (possible)

Ce qui nous donnerait un fragment aléatoire « tirons une position ».

Cette première fragmentation aléatoire est appelée « uniforme », puisqu'il n'y a pas de pondération particulière sur la longueur des fragments, celles-ci étant équiprobables.

Nous proposons également une seconde version, appelée « pondérée », avec une pondération sur les longueurs de fragments, afin que la distribution des longueurs dans la fragmentation originale et aléatoire soient similaires. Les poids sélectionnés sont les suivants : 0,77 pour les fragments de longueur 2, 0,15 pour les fragments de longueur 3 et 0,08 pour les fragments de longueur 4.

5 Résultats et discussion

Dans cette partie, nous présentons et analysons des premiers résultats issus des expériences menées, et montrons que l'autonomie pourrait nous permettre d'extraire des fragments syntaxiques.

5.1 Taille du corpus d'entraînement

Afin d'obtenir de bonnes estimations de l'entropie sur les bigrammes, trigrammes et quadrigrammes, il nous faut un corpus d'entraînement suffisamment grand. Nous commençons par regarder la précision sur les fragments extraits pour différentes tailles de corpus : 1000 tokens, 10 000 tokens, 100 000 tokens, 500 000 tokens et 1 million de tokens.

Nous extrayons les fragments apparaissant dans la meilleure segmentation de chaque phrase, et vérifions leur statut de catena dans la version SUD des corpus arborés. Les résultats correspondant sont présentés en 4 où on note principalement que les meilleures précisions globales (respectivement 0,83 et 0,82) sont obtenues pour les deux plus grandes tailles de corpus. Ce gain de précision vient avant tout de meilleures prédictions sur les trigrammes et les quadrigrammes qui sont probablement trop rares dans les petits corpus pour qu'on puisse réellement estimer leur autonomie.

5.2 Influence du schéma d'annotation

Cette fois-ci, nous nous intéressons aux variations dans l'évaluation de la précision selon le schéma d'annotation des corpus arborés.

Les scores de précision globaux indiquent que les fragments extraits respectent davantage le schéma SUD+ (0,81), que le schéma SUD (0,68), c'est-à-dire qu'en changeant uniquement la relation entre les noms et déterminants pour que les déterminants deviennent gouverneurs nous gagnons 0,13 de précision ce qui est considérable. Il est aussi intéressant de noter qu'il y a au final peu de différences entre les scores pour le schéma SUD et le schéma UD, bien que les structures dans ces deux versions soient très différentes.

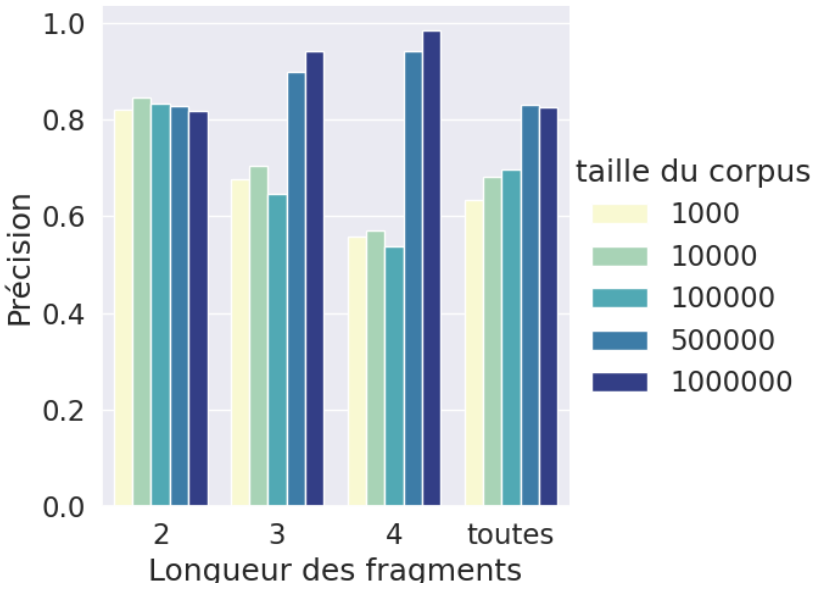

FIGURE 4 – Influence de la taille du corpus d'entraînement sur la précision des fragments extraits (schéma : SUD, n=1)

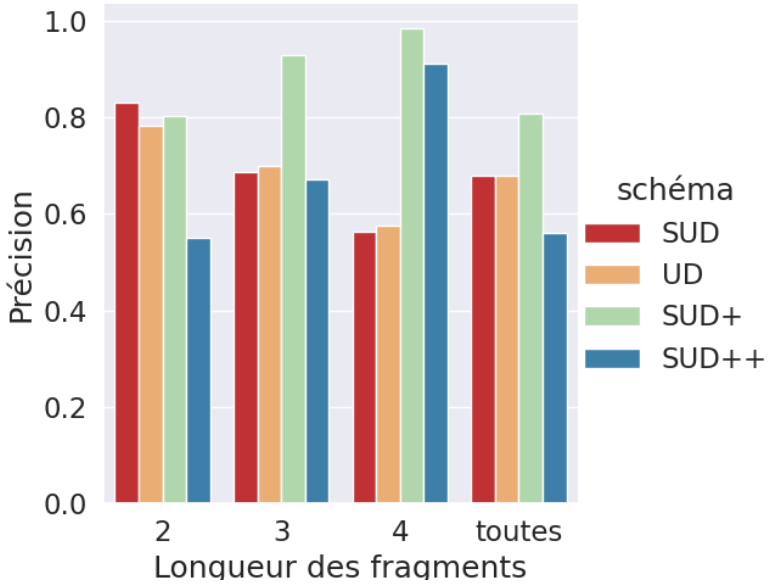

FIGURE 5 – Influence du schéma d'annotation sur la précision des fragments extraits (taille : 1 million de tokens, n=1

5.3 Évolution des scores en fonction des *n* meilleurs

Jusqu'à présent l'évaluation ne concernait que les fragments appartenant à la meilleure segmentation de chaque phrase. Pour ces fragments nous notons une bonne amélioration par rapport à au score de référence sur les fragments aléatoires, en revanche il serait incomplet de s'arrêter ici sans parler de rappel.

Afin de visualiser l'évolution de la précision et du rappel en fonction des *n* meilleures segmentations sélectionnées, nous choisissons de nous intéresser à des phrases de longueur fixe, ce qui nous permettra de fixer un *n* maximum qui corresponde au nombre total de segmentations possibles. [4] Nous sélectionnons toutes les phrases de longueur 10 et faisons varier *n* entre 1 et 401 afin de couvrir toutes les segmentations possibles.

La précision démarre assez haute avec 86% des fragments extraits qui sont des catenas, puis décroit rapidement dans les 25 meilleures segmentations. Elle décroit ensuite plus lentement jusqu'à atteindre 0,57 lorsque toutes les segmentations sont prises en compte. Côté rappel on observe 3 phases, une augmentation très forte dans les 25 premières segmentations, où l'on atteint 0,48, puis une augmentation forte jusqu'aux alentours de la 250e segmentation (0,96) et une augmentation beaucoup plus lente sur la fin.

Pour pouvoir fixer un *n* qui permettrait d'avoir à la fois une bonne précision et un rappel suffisant, il faudrait regarder à quel point certaines catenas peuvent être déduites d'autres catenas qui se combineraient ensemble (par exemple une catena de longueur 2 qui se combinerait avec une catena de longueur 3, avec l'un des noeuds en commun pourrait permettre de déduire la catena de longueur 4 qui englobe les deux).

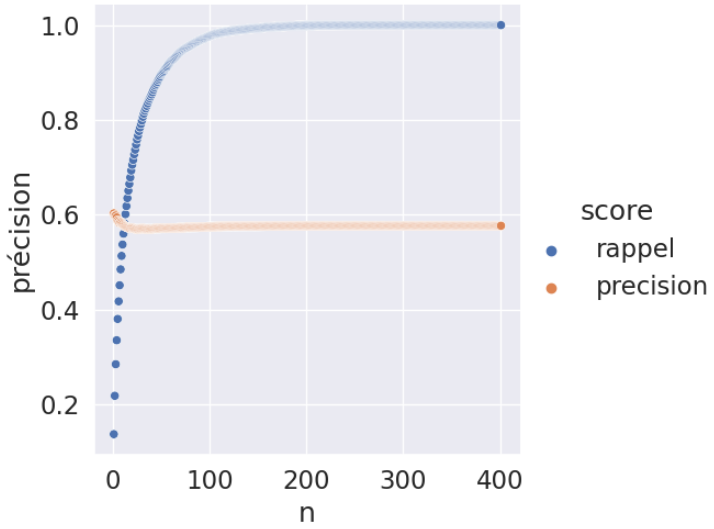

FIGURE 6 – Précision et rappel sur les fragments extraits en fonction des *n* meilleures segmentations sélectionnées (schéma : SUD, taille : 1 million de tokens)

4. Le nombre de segmentations possible pour une phrase de longueur *m* avec des segments de longueurs comprises entre 1 et *p* peut être obtenu à partir de la p-suite de Fibonacci (Olaiju & Taiwo, 2015)

5.4 Comparaison entre fragments extraits et fragments aléatoires

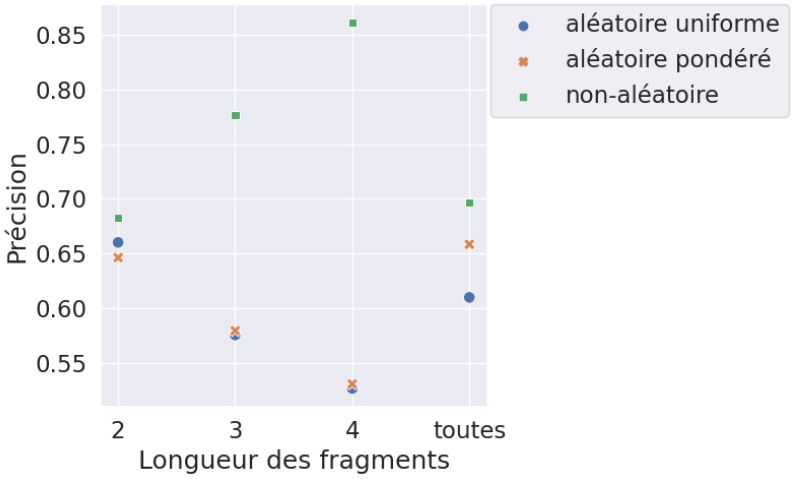

FIGURE 7 – Précision sur les fragments aléatoires et candidats extraits (parmi les 10 meilleures segmentations), en fonction de leur longueur (schéma : SUD, taille : 1 million de tokens)

Dans la figure 7 nous pouvons observer à quel point les fragments extraits (qu'ils soient extraits aléatoirement ou suivant notre méthode) sont effectivement des catenas dans le corpus arboré de référence. Pour ce qui est des fragments aléatoires, nous avons des précisions similaires pour les deux méthodes, avec respectivement pour la méthode uniforme et la méthode pondérée : une précision de 0,66 et 0,65 pour les fragments de longueur 2, 0,58 pour les fragments de longueur 3, 0,53 pour les fragments de longueur 4 et 0,61 contre 0,66 si on ne tient pas compte de la longueur. Plus le fragment est long moins celui-ci a de chances d'être effectivement une catena, ce qui correspond bien aux fréquences décrites dans la table 1.

Pour ce qui est des fragments extraits en suivant notre méthode, on a des scores globalement plus élevés, à savoir 0,68 pour les fragments de longueur 2, 0,78 pour les fragments de longueur 3, 0,86 pour les fragments de longueur 4 et 0,70 si on ne tient pas compte de la longueur. Il est particulièrement intéressant de voir que contrairement aux fragments aléatoires, la précision augmente ici avec la longueur du fragment. Nous pensons que c'est un signe encourageant, car ces catenas sont importantes si on veut pouvoir espérer induire une structure de dépendance, du fait de leur chevauchement avec les autres catenas.

La performance de notre modèle dépend fortement de quel n nous choisissons ici, plus le n sera élevé plus les prédictions seront bruitées et se rapprocheront de la méthode aléatoire qui nous sert de référence. En revanche avec un petit n, on aura de bien meilleures prédictions qu'avec l'aléatoire, mais au détriment du rappel.

6 Conclusion et perspectives

Avec cet article, nous avons voulu montrer qu'il est possible de faire de prédictions sur la nature syntaxique ou non d'une séquence de tokens en français, en nous basant sur l'entropie.

Nous proposons d'extraire des fragments en utilisant une mesure d'autonomie basée sur l'entropie, et montrons que ceux-ci sont plus souvent des unités syntaxiques (plus précisément des catenas) qu'avec une méthode de référence aléatoire. Nous montrons également que le corpus d'entraînement doit atteindre une certaine taille pour pouvoir espérer extraire de plus longs fragments.

Les expériences sur le français semblent indiquer que la structure induite de cette façon se rapproche davantage du schéma SUD+ avec des têtes fonctionnelles et des déterminants têtes des noms avec lesquels ils se combinent, puisque c'est celui-ci qui obtient la meilleure précision.

Il reste encore de nombreuses pistes à explorer, notamment pour savoir quelle serait la couverture minimale permettant d'induire de bonnes structures à partir d'un nombre limité d'unités identifiées, ce qui nous permettrait de sélectionner seulement une partie des meilleurs fragments et d'éviter d'introduire des fragments trop bruités.

Une autre piste consisterait à s'intéresser aux séquences qui semblent les moins probables d'être des unités syntaxiques. Identifier ces « mauvaises » unités pourrait nous permettre d'éliminer d'office un certain nombre de connexions, ce qui réduirait la complexité du problème d'induction de la structure.

Une telle méthode pourrait être utilisée dans des travaux ultérieurs afin de proposer une induction non-supervisée de structures de dépendance syntaxiques.

Références

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DIAS G., Éd. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

FUTRELL R., QIAN P., GIBSON E., FEDORENKO E. & BLANK I. (2019). Syntactic dependencies correspond to word pairs with high mutual information. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, p. 3–13, Paris, France : Association for Computational Linguistics. DOI : 10.18653/v1/W19-7703.

GERDES K., GUILLAUME B., KAHANE S. & PERRIER G. (2018). SUD or surface-syntactic Universal Dependencies : An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, p. 66–74, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-6008.

GERDES K. & KAHANE S. (2011). Defining dependencies (and constituents). In *International Conference on Dependency linguistics (Depling 2011)*, p. 17–27.

GUILLAUME B., BONFANTE G., MASSON P., MOREY M. & PERRIER G. (2012). Grew : un outil de réécriture de graphes pour le TAL (Grew : a graph rewriting tool for NLP) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 5 : Software Demonstrations*, p. 1–2, Grenoble, France : ATALA/AFCP.

HARRIS Z. S. (1955). From morpheme to phoneme. *Language*, **31**(2), 190–222.

HEWITT J. & MANNING C. D. (2019). A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association*

for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers), p. 4129–4138, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1419.

MAGISTRY P. (2013). *Unsupervised Word Segmentation and Wordhood Assessment*. Thèse de doctorat, Paris Diderot ; Inria.

MAGISTRY P. & SAGOT B. (2012). Unsupervized word segmentation : the case for Mandarin Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 383–387, Jeju Island, Korea : Association for Computational Linguistics.

OLAIJU S. & TAIWO A. (2015). Steps problem : the link between combinatoric and k-bonacci sequences. *European Journal of Statistics and Probability*, **3**(4), 10–19.

OSBORNE T., PUTNAM M. & GROSS T. (2012). Catenae : Introducing a novel unit of syntactic analysis. *Syntax*, **15**, 354–396. DOI : https ://doi.org/10.1111/j.1467-9612.2012.00172.x.

PETERS M., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 2227–2237, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1202.

STRUBELL E., GANESH A. & MCCALLUM A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 3645–3650, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1355.

TESNIÈRE L. (1959). *Éléments de syntaxe structurale*.

ZEMAN D., NIVRE J. *et al.* (2020). Universal dependencies 2.7. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.