IFT6285 - Devoir 4

Louis-Vincent Poellhuber - Youcef Islam Barkat

23 novembre 2023

1 Question a.

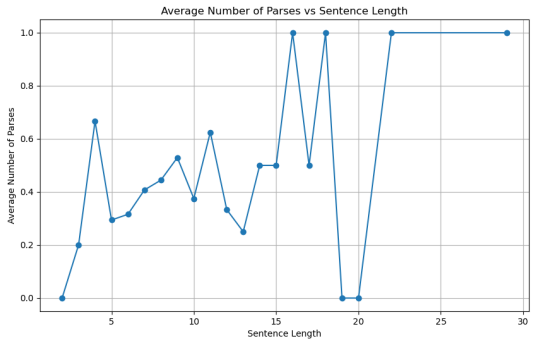
Afin d'analyser les phrases contenant des mots inconnus, nous avons mis en place une solution en remplaçant d'abord les mots inconnus dans l'ensemble de données par un symbole de substitution, 'UNK.' Cette approche nous permet d'intégrer ces phrases de manière transparente dans le processus d'analyse. De plus, nous avons modifié la grammaire pour inclure des règles explicites de gestion du symbole 'UNK', garantissant ainsi sa reconnaissance et son analyse correcte. En prétraitant l'ensemble de données pour remplacer les mots inconnus et en mettant à jour la grammaire en conséquence, nous avons permis au ViterbiParser de gérer efficacement les phrases avec un vocabulaire précédemment non reconnu, contribuant à une capacité d'analyse plus robuste et complète.

2 Question b.

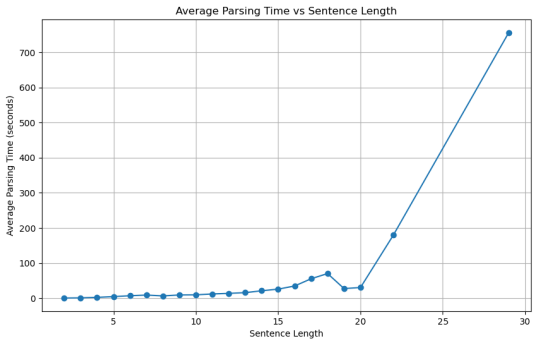
La moyenne des longueurs de phrases dans le corpus Penn Treebank (PTB), distribué dans le Natural Language Toolkit (NLTK), présente une différence significative par rapport aux phrases agrammaticales du jeu de données CoLA dev. Le corpus PTB affiche une longueur moyenne de phrase relativement plus longue, d'environ 25,7 mots. En revanche, les phrases agrammaticales du jeu de données CoLA dev sont nettement plus courtes, avec une longueur moyenne d'environ 8,7 mots. Cette disparité souligne les caractéristiques linguistiques diverses capturées par ces deux ensembles de données : le corpus PTB, reconnu pour sa complexité syntaxique et ses structures linguistiques riches, et le jeu de données CoLA dev, spécifiquement conçu pour évaluer l'acceptabilité, capturant des formes de phrases plus courtes et plus directes.

3 Question c.1

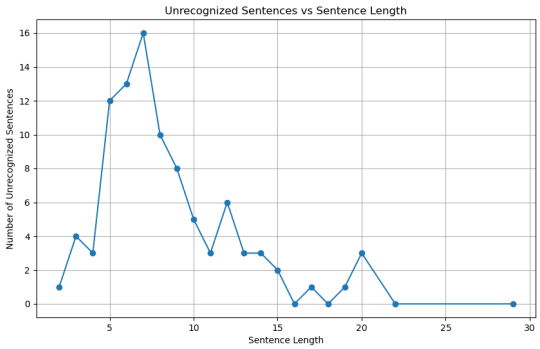
Pour offrir une analyse complète des performances du ViterbiParser, trois aspects clés ont été étudiés : le temps d'analyse et le nombre d'analyses, le temps d'analyse moyen et la reconnaissance des phrases.



(a) Parsing Time and Number of Parses Analysis



(b) Average Parsing Time Analysis



(c) Unrecognized Sentences Analysis

FIGURE 1 – Analysis of Parsing Time, Number of Parses, and Unrecognized Sentences for Different Subset Lengths

L'analyse des performances du ViterbiParser sur des longueurs de phrases variables révèle des tendances notables. Tout d'abord, le temps moyen d'analyse présente une corrélation directe avec la longueur de la phrase, démontrant une augmentation du temps d'analyse à mesure que les phrases deviennent plus longues. De manière similaire, le nombre moyen d'analyses suit une tendance parallèle, augmentant avec la croissance de la longueur de la phrase. En revanche, le nombre de phrases non reconnues affiche une relation inverse avec la longueur de la phrase, indiquant une diminution de la proportion de phrases non reconnues à mesure que leur longueur augmente. Il est important de noter que certaines lacunes dans les graphiques peuvent découler de l'absence de longueurs de phrases spécifiques dans l'ensemble de données CoLA Dev, soulignant la couverture limitée du jeu de données pour certaines longueurs de phrases.

4 Question c.2

Le traitement de la grammaire, en particulier en conservant uniquement les règles les plus fréquentes, révèle des impacts significatifs sur les performances du ViterbiParser. Le nombre d'analyses générées reste constant entre la grammaire complète et la grammaire filtrée, indiquant que le processus de filtrage n'affecte pas la capacité du parseur à produire des analyses pour les phrases données. Cependant, on observe une réduction notable du temps d'analyse avec la grammaire filtrée (1833,58 secondes) par rapport à la grammaire complète (1955,32 secondes). Cela suggère que le filtrage de la grammaire pour inclure uniquement les règles les plus fréquentes contribue à un processus d'analyse plus efficace, entraînant une diminution substantielle du temps de calcul sans compromettre la précision de l'analyse.

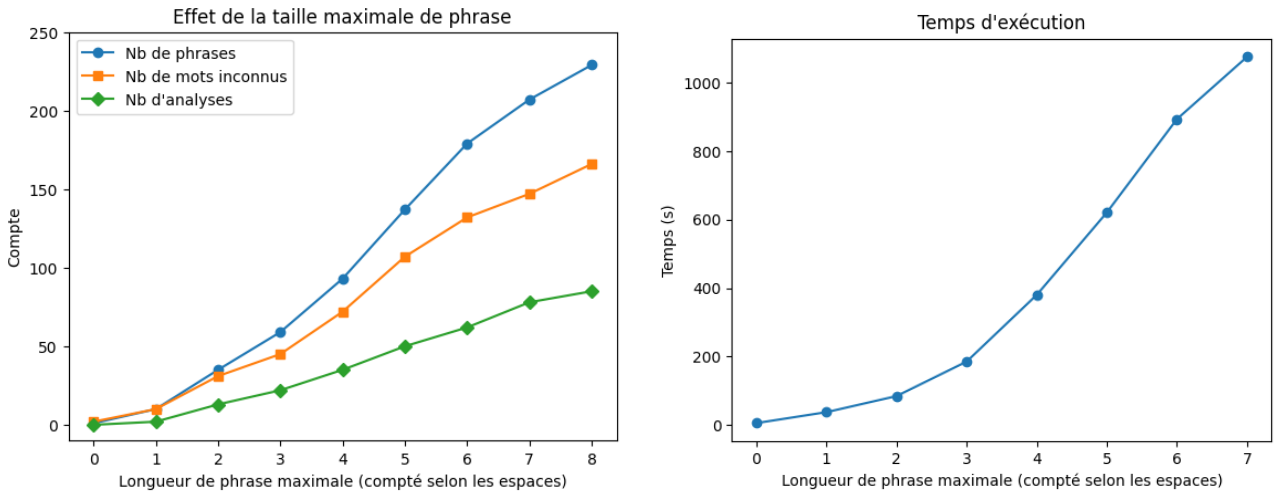


FIGURE 2 – Analyse de l'effet de la longueur de phrase maximale sur différentes métriques

5 Question d.

Nous avons premièrement étudié l'impact de la longueur de phrase maximale sur différentes métriques. En se fiant à la Figure 2, on remarque que le temps de calcul augmente presque exponentiellement, ce qui est en autres pourquoi nous avons choisi d'arrêter après 10 mots. On remarque une évolution semblable quant au nombre de phrases filtrées, nombre de mots inconnus et du nombre d'analyses (parses) effectués par ViterbiParser, quoi que la tendance est plutôt linéaire.

Pour notre taille maximale choisie, étant de 10 mots maximum, nous avons donc observé 229 phrases, contenant 166 mots inconnus uniques. NLTK n'a pu qu'analyser que 85 de ces phrases : c'est une toute petite portion du nombre de phrases totales. Nous avons choisi aléatoirement deux phrases qui sont clairement grammaticalement incorrecte.

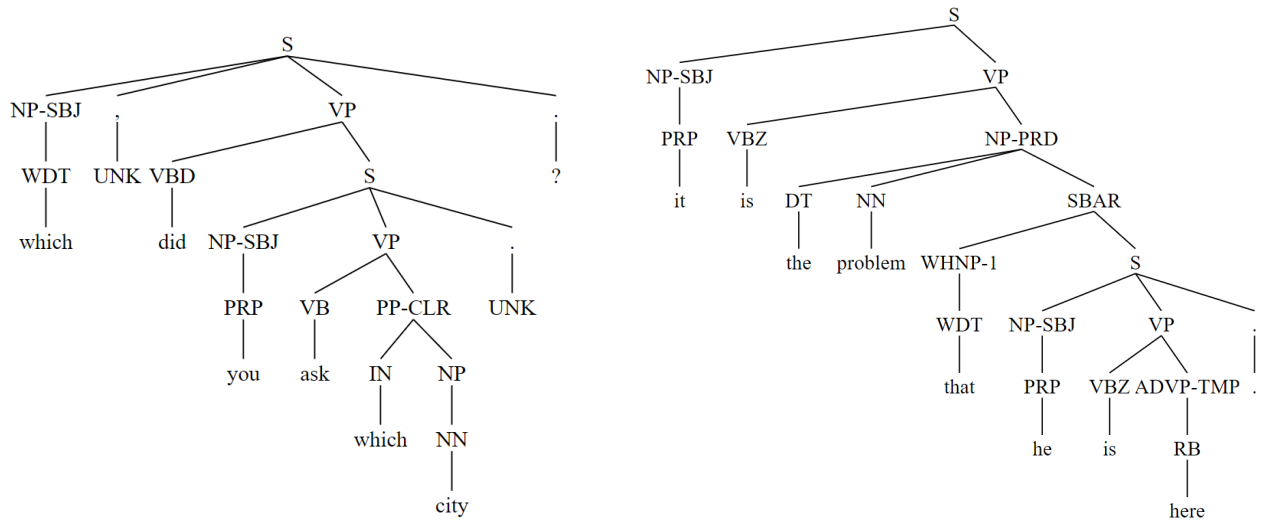


FIGURE 3 – Exemples d'arbres syntaxiques de phrases grammaticalement incorrectes, avec une longueur de phrase maximale de 10 mots (= 9 espaces).

6 Question e.

En soit, la faiblesse du parser de NLTK est qu'il ne fait que combiner des règles déjà vues, il n'y a rien qui détermine si la combinaison de ces règles produit une phrase grammaticalement correcte. Nous avons remarqué que la profondeur de l'arbre syntaxique semble souligner ce problème : si on regarde la Figure 3, on remarque plusieurs mots ayant beaucoup de niveaux sans branchement. Par exemple, le tout premier mot, *which*, est analysé à l'aide de deux règles. Afin de mieux souligner ce problème, nous avons trouvé un exemple d'une courte phrase avec une profondeur anormale. Dans la Figure 4, l'arbre syntaxique a une profondeure maximale de 5 alors que la phrase ne possède que 3 mots ! Cette phrase n'a pu être analysée parce qu'il existe une combinaison complexe de règles la structurant. Il est cependant évident avec l'exemple ci-contre qu'il s'agit persque d'un hasard que de telles règles se combinent correctment.

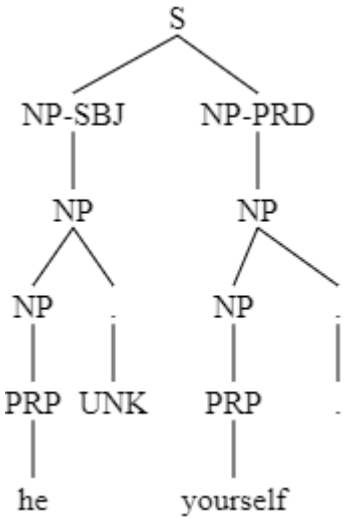


FIGURE 4 – Arbre très profond pour le nombre de mots.

De plus, théoriquement, avec un jeu de règles (productions) assez élevé, nous pourrions analyser n'importe quelle phrase : ceci n'implique toutefois pas que ces phrases sont grammaticalement correctes. Ainsi, la **profondeur de l'arbre**, la **longueur de la phrase** et surtout le **ratio entre les deux** peuvent être de bons features pour estimer la pauvreté grammatiale d'une phrase. Idéalement, il nous faudrait des méta-règles pour déterminer quelles combinaisons de règles sont valides ! Ceci implique toutefois un travail manuel encore plus monumental que ce qui a déjà été requis pour le PTB, car il faudrait soit les lister explicitement ou avoir un nombre de règles assez immense pour apprendre les méta-règles.