Université de Montréal

IFT6285 – Traitement automatique des langues naturelles

Devoir 2

---

par:

Eugénie Yockell
(20071932)

Date de remise: 25 septembre 2020

Kneser–Ney est une méthode qui permet de calculer la distribution de n-grams dans un document selon l'historique, soit l'ensemble d'entraînement. C'est une méthode qui a été proposé par Reinhard Kneser, Ute Essen et Hermann Ney. Nous allons ici étudier les modèles Kneser-Ney à l'aide du corpus *1Bwc*. Nous entraînons un modèle Kneser-Ney à l'aide du package `KenLM`. Pour commencer, nous avons entraîné un modèle `KenLM` bigramme sur 9 tranches, et nous avons mesuré la perplexité de 1000 phrases sur un ensemble test. Les valeurs de perplexité sont définies à la table 1. On évalue ici la qualité du modèle Kneser-Ney à l'aide de la perplexité qui est mesuré sur une phrase inconnue du modèle. La perplexité correspond à sa difficulté à prédire cette phrase. Ainsi, plus la perplexité est petite, meilleur est le modèle. Les perplexités sont mesurées à l'aide de la librairie Python `kenlm`.

	Minimum	**Maximum**	**Moyenne**
Perplexité	28.74	21 411.13	412.93

Table 1: Perplexité minimum, maximum et moyenne sur 1000 phrases tests sur un modèle Kneser-Ney bigramme entraîner sur 9 tranches.

La perplexité minimale est "I had never seen anything like it." Ce n'est pas étonnant puisque c'est une phrase très simple. C'est aussi une phrase qui contient beaucoup de mots très commun: 'I', 'had', 'anything', 'like' et 'it' et ainsi, le modèle à une grande probabilité de les prédire.

Tandis que la perplexité maximale est à la phrase "Rossi broke Stoner 's 2007 lap record of 2min 02.108sec." Encore une fois, ce résultat n'est pas étonnant. Cette phrase contient beaucoup de nombre, des ponctuations et des mots rares comme 'Rossi' et 'Stoner'. Le modèle a donc de la difficulté à prédire une telle phrase.

Il est intéressant d'étudier les particularités des modèles bigrammes `KenLM` selon la quantité de données d'entraînements. La quantité de données est définie selon le nombre de tranches utilisé du corpus *1Bwc*. À noter que les calculs ont été fait sur Google Colab où la plus grande restriction était l'espace. En effet, les modèles `KenLM` prennent beaucoup d'espace et ne peuvent pas facilement être supporté par l'espace restreint de Google Colab de 107 GB et de 15 Go pour le *drive* personnel. On peut étudier la mémoire que requière chaque modèle bigramme `KenLM` selon la quantité de données d'entraînement. À la figure 1 on visualise la taille en GB selon le nombre de tranches utilisé. Cette fois, la courbe n'est pas toute à fait linéaire. La taille tend à augmenter plus rapidement plus il y a de données d'entraînements. De plus, plus on augmente l'ordre des modèles n-grams `KenLM` plus les modèles sont gros.

À la figure 2 on voit que le temps semble linéaire en fonction du nombre de tranches considéré. La droite n'est pas lisse, car les modèles n'ont pas tous été calculé au même moment, ainsi la ressource pouvait parfois générer différente vitesse. On remarque tout de même la tendance linéaire, ce qui est logique, car chaque tranche contient environ la même quantité de mots.

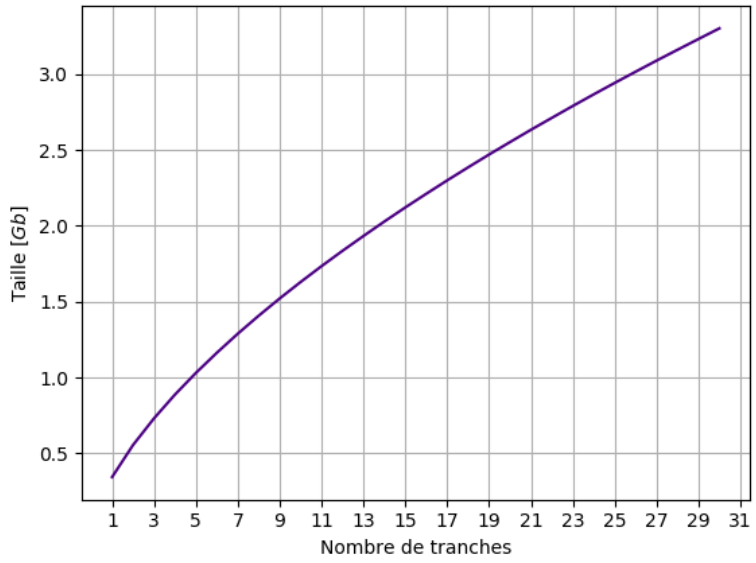


Figure 1: Évolution de la taille du modèle binaire `KenLM` en fonction du nombre de tranche utilisé pour sa création.

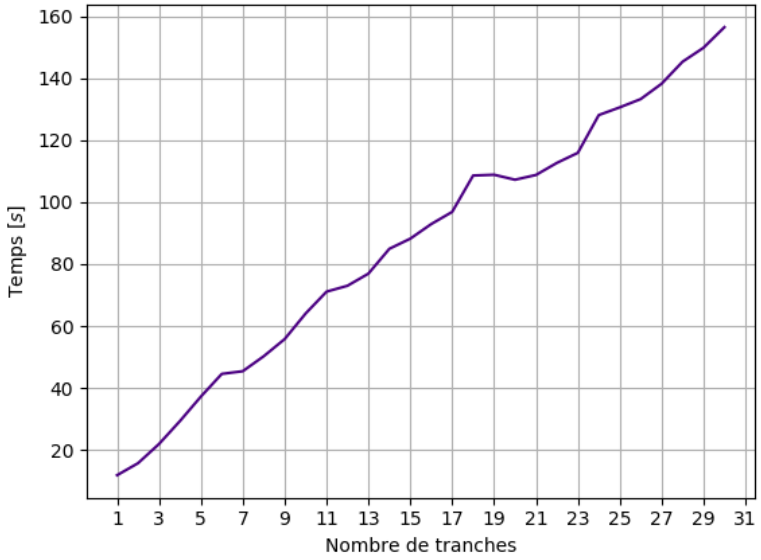
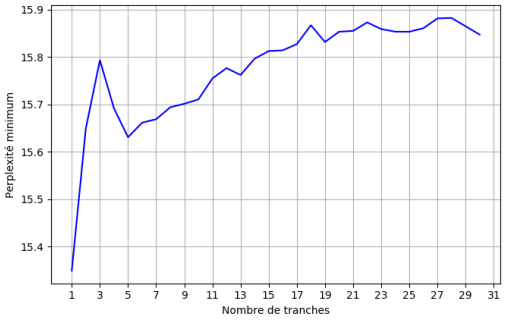
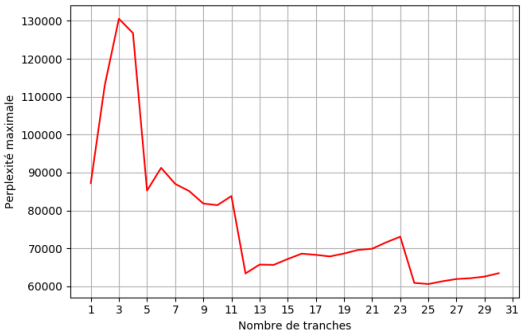


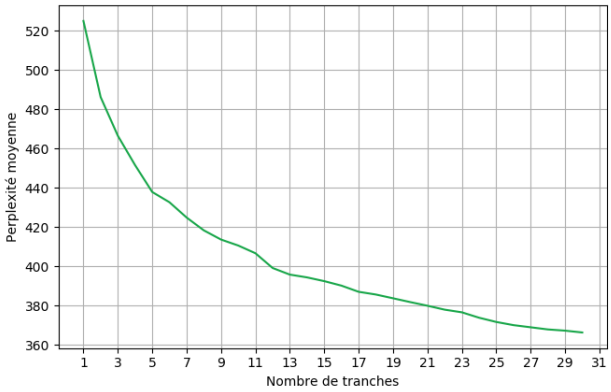
Figure 2: Évolution du temps nécessaire pour créer le modèle binaire `KenLM` en fonction du nombre de tranche utilisé pour sa création.



(a) Perplexité minimale en fonction du nombre de tranches. Maximum à 15.88.



(b) Perplexité maximale en fonction du nombre de tranches. Maximum à 130 564.



(c) Perplexité moyenne en fonction du nombre de tranches. Maximum à 525.1 et minimum à 366.32.

Figure 3: Différentes études de perplexité sur les 10 000 premières phrases de l'ensemble test pour un modèle bigramme `KenLM` entraîner sur différents ensemble d'entraînement de 1 tranches à 30 tranches.

Étudions maintenant la qualité d'un modèle bigramme `KenLM` selon la quantité de données d'entraînements. À la figure 3a on remarque que la perplexité minimale tend plutôt à augmenter tandis que la perplexité maximale tend à diminuer. Par contre, on remarque que la perplexité moyenne diminue rapidement et semble avoir une forme $f(x) = \frac{1}{x}$. On peut conclure que la perplexité moyenne tend vers une asymptote. Dans tel cas, même si on augmente le nombre de tranche, la perplexité moyenne restera autour de 360.

Essayons maintenant de définir le meilleur modèle en étudiant des modèles n-grams d'ordre différent. Cette étude est présentée à la table 2. On s'attendait à voir la perplexité moyenne augmenter avec un modèle à plus grand n, mais elle diminue. En effet, un modèle n-grams de trop grand ordre perd de la généralité qui est nécessaire pour prédire des phrases inconnues. Par contre, ce n'est pas nécessairement le cas si le texte d'entraînement est assez grand et diversifié pour que, par exemple, un modèle 6-grams reste général. Il se pourrait aussi que le texte de test est semblable au texte d'entraînement et ainsi le modèle reste précis même s'il est complexe. Il se pourrait aussi simplement que la perplexité moyenne n'est pas la meilleur

métrique pour mesurer la qualité d'un modèle. On reste alors incertain sur la qualité réel du modèle `KenLM` à 6-grams.

n-grams	**Perplexité moyenne**
2-grams	410.6
3-grams	314.1
4-grams	301.3
5-grams	299.0
6-grams	298.6

Table 2: Perplexité moyenne pour différentes modèles n-grams entraînés sur 10 tranches. On mesure la moyenne sur 10 000 phrases tests.

Afin de générer le meilleur modèle, il serait intéressant d'étudier aussi les conséquences de prétraitement sur le modèle. Nous allons appliqué ce prétraitement au texte d'entraînement de 10 tranches et le texte de test de 10 000 phrases pour un modèle bigramme sur 10 tranches. Nous avons tout mis en minuscule, nous avons lemmatisé et placer tous les nombres sous le nom: "NOMBRE". On remarque que la perplexité moyenne augmente en comparaison à un modèle non prétraité. Notre hypothèse est que le modèle utilise les majuscules en début de phrase. Le fait de garder les majuscules permet de donner a certain mot une plus grand probabilité de débuter une phrase.

On peut alors conclure que notre meilleur modèle est un modèle non prétraité. Comme nous avons remarqué à la figure 3c la perplexité moyenne tend vers une asymptote pour un grand nombre de texte. Nous allons alors utiliser 20 tranches. Nous avons aussi choisis de conserver un modèle 4-grams, car les modèles de plus grand ordre sont de qualités incertaines. Les perplexités du meilleur modèle sont présentés à la table 3.

	Minimum	**Maximum**	**Moyenne**
Perplexité	2.61	66545.57	263.14

Table 3: Perplexité minimum, maximum et moyenne sur 10 000 phrases tests sur un modèle Kneser-Ney 4-grams entraîner sur 20 tranches.

Il aurait été intéressant d'étudier des modèles de plus grand ordre et mesurer l'impact du nombre de tranches considéré. Avec de plus grande capacité de calcul, il aussi été intéressant de montrer l'effet du surapprentissage pour le modèle de Kneser-Ney de grand ordre sur l'ensemble de texte disponible. De plus, une étude plus approfondie de l'effet du prétraitement sur l'ensemble d'entraînement serait importante.