Modèles de langues - AIMA.

Russell & Norwig - traduction et adaptation Pierre Mercuriali
Avec un appendice explicatif ajouté

Introduction

Ce document est une traduction et adaptation du chapitre 22 - Natural Language Processing, section 22.1 - Language Models, de l'ouvrage *Artificial Intelligence - A Modern Approach* (AIMA) de Stuart Russell et Peter Norvig.

"Traduction", car nous en avons traduit intégralement l'anglais en français ; "adaptation", car nous avons rajouté des explications de termes précédemment expliqués afin que le présent document se suffise à lui-même et, surtout, soit lisible par un public qui n'est pas un public de mathématiciens ou d'informaticiens, mais plutôt un public de linguistes. Puisqu'il n'y en a pas dans le texte original, toutes les notes de bas de page sont des notes de lecture qui relèvent de cette adaptation.

Mots-clés : analyse statistique du texte ; modèles de langues ; probabilités ; traitement automatique des langues

Prérequis : un peu de probabilités fréquentistes : la notion de probabilités conditionnelles ; chaînes de Markov (pour la justification des modèles n-gramme. Voir Appendice A pour une présentation).

Traduction de l'introduction du chapitre 22.

Traitement automatique des langues[1] : où l'on voit comment utiliser la quantité phénoménale de connaissances exprimées en langue naturelle.

Homo sapiens est considéré comme une espèce à part du fait de sa capacité d'utilisation du langage. Il y a environ 100 000 ans, les humains ont appris à parler, et il y a environ 7 000 ans ils ont appris à écrire. Bien que les chimpanzées, les dauphins, et d'autres animaux aient montré qu'ils possèdent des vocabulaires de plusieurs centaines de signes, seuls les humains peuvent communiquer, de manière sûre, un nombre non-borné de messages quantitativement différents sur n'importe quel sujet, en utilisant des signes discrets[2].

Il existe bien entendu d'autres attributs qui sont uniquement humains : aucune autre espèce ne porte d'habits, ne crée d'art figuratif, ou ne regarde la télévision trois heures par jour. Mais,

[1] *NLP* en anglais, c'est-à-dire "Natural Language Processing" : traitement du langage naturel. Comparez les différences de traduction : en français, nous avons rajouté la notion d'*automatisme*, et nous avons enlevé la notion de naturalité. Vous trouverez parfois les termes "traitement de la langue naturelle", "traitement automatique de la langue naturelle", etc. Mais le sigle le plus courant est bien TAL (traitement automatique des langues).

[2] Le terme "discret" fait référence à la distinction "discret/continu" en mathématiques. Ici, cela signifie que les signes sont bien séparés - pensez à la notion de phonèmes ou de paires minimales. Par exemple, un tel ensemble de signes discrets peut être l'alphabet latin.

lorsqu'Alan Turing a proposé son Test (voir Section 1.11), il l'a fondé sur le langage, et non pas l'art ou la télévision . Deux grandes raisons font que nous voudrions nos agents ordinateurs[3] puissent traiter les langages naturels : premièrement, pour communiquer avec les humains, un sujet que nous traitons dans le Chapitre 23, et, deuxièmement, pour acquérir de l'information à partir du langage écrit, qui est le sujet de ce chapitre.

Il y a des milliards de milliards de pages sur le Web, qui sont presque toutes en langage naturel. Un agent qui souhaite *acquérir de l'information* a besoin de comprendre (au moins partiellement) les langages ambigus et approximatifs[4] que les humains utilisent. Nous examinons ce problème du point de vue de tâches spécifiques de recherche d'information : la classification de textes, la récolte d'information[5], et l'extraction d'information. Un des facteurs communs lorsque l'on tente d'accomplir ces tâches est l'utilisation de *modèles de langue*[6] : des modèles qui prédisent la distribution de probabilité des expressions du langage.

1 Language Models

Les langages formels tels que les langages de programmation Java ou Python on des modèles de langue définis précisément. Un *langage* peut être défini comme un ensemble de chaînes de caractères ; "`print( 2 + 2 )`" est un programme légal[7] en Python, mais "`2 ) + ( 2 print`" n'en est pas un. Puisqu'il existe un nombre infini de programmes légaux, il n'est pas possible de tous les énumérer. Au lieu de cela, les programmes légaux sont spécifiés par un ensemble de règles que l'on appelle une *grammaire*. Les langages formels ont aussi des règles qui définissent le sens[8] ou la *sémantique* d'un programme. Par exemple, ces règles stipulent que le "sens" de "`2 + 2`" est 4,[9] et que le sens de "`1 / 0`" est qu'une erreur est signalée.

Les langages naturels tels que l'anglais ou l'espagnol ne peuvent pas être caractérisés comme un ensemble de phrases définitif. Tout le monde s'accorde à dire que "Ne pas être invité est triste" est une phrase correcte en français, mais certains ne sont pas d'accord sur la grammaticalité de "Être non-invité est triste". Donc, il est plus utile de définir un modèle du langage naturel comme une distribution de probabilité sur les phrases, plutôt qu'un ensemble définitif. En d'autres termes, plutôt que de se demander si une chaîne de mots *mots* est, ou n'est pas, un membre de l'ensemble qui définit le langage, nous demandons la valeur de

$$\mathbb{P}(S = mots),$$

c'est-à-dire que nous demandons la probabilité qu'une phrase aléatoire S soit constituée de la chaîne *mots*.

Les langages naturels sont, de plus, *ambigus*. "Je vois un éléphant avec mon télescope" peut signifier soit que j'observe un pachiderme à l'aide de mon télescope, soit que l'animal m'a subtilisé mon appareil. Donc, là encore[10], nous ne pouvons pas parler de l'unique sens d'une phrase, mais plutôt d'une distribution de probabilité sur les sens possibles.

[3] "Computer agents" fait référence au modèle principal de l'AIMA, l'*agent*, une entité abstraite qui, au plus simple, peut sentir son environnement et agir (agent) sur cet environnement, à la suite d'un traitement interne de ce qu'il a senti de l'environnement.

[4] "messy".

[5] "Information retrieval".

[6] Quelle est la différence entre "modèle de langue" et "modèle de langage" ? Dans ce texte, nous ne ferons pas de distinction et nous préfèrerons le terme "modèle de langue" parce que ça sonne mieux, mais en linguistique la distinction langue/langage peut être importante.

[7] C'est-à-dire qu'il respecte les règles de la syntaxe de Python et pourra être compilé et lancé par un ordinateur.

[8] "meaning"...

[9] Remarquez la différence de police d'écriture entre '`2`' (symbole du langage formel) et '4' (symbole du "métalangage" sémantique).

[10] C'est-à-dire dans le domaine de la sémantique, et non plus de la grammaire - la suite de mots *mots*.

Enfin, les langages naturels sont difficiles à gérer car ils sont très grands et qu'ils changent constamment. Donc, nos modèles de langue sont, au mieux, des approximations. Nous commençons avec les approximations les plus simples, et puis les complexifions.

1.1 Modèles de caractères n-gramme

Éventuellement, un texte écrit est composé de *caractères* : des lettres, des chiffres, des signes de ponctuation, et des espaces, du moins en français (et des caractères plus exotiques dans d'autres langages). Par conséquent, un des modèles de langue les plus simples est une distribution de probabilité sur les suites de caractères. Comme nous l'avons fait au Chapitre 15, nous écrivons

$$\mathbb{P}(c_{1:N})$$

pour indiquer la probabilité (d'occurrence) d'une suite de N caractères c_1 à c_N. Dans une collection de textes du Web,

$$\mathbb{P}(\text{"the"}) = 0,027, \text{ et}$$
$$\mathbb{P}(\text{"zgq"}) = 0,000000002.$$

Une suite écrite de caractères de taille n est appelée un n-gramme (de la racine grecque qui signifie écriture ou lettre) et, dans certains cas, un "unigramme" pour le cas 1-gramme, un "bigramme" pour le cas 2-gramme, et "trigramme" pour le cas 3-gramme. On appelle donc un modèle de la distribution de probabilité de suites de lettres de taille n un *modèle n-gramme* (mais faites attention : on peut définir des modèles n-grammes sur des suites de mots, de syllabes, ou d'autres unités, pas seulement sur des suites de caractères).

Un modèle n-gramme est défini comme une *chaîne de Markov*[11] d'ordre $n-1$. Souvenez-vous, comme expliqué page 568, que dans une chaîne de Markov la probabilité d'un caractère c_i dépend seulement des caractères qui le précèdent immédiatement, et eux seuls. Donc, dans un modèle trigramme (une chaîne de Markov d'ordre 2), nous avons

$$\mathbb{P}(c_i|c_{1:i-1}) = \mathbb{P}(c_i|c_{i-2:i-1}).$$

Nous pouvons définir la probabilité d'une suite de caractères $\mathbb{P}(c_{1:N})$ en factorisant en chaîne et en utilisant l'hypothèse de Markov :

$$\mathbb{P}(c_{1:N}) = \prod_{i=1}^{N} \mathbb{P}(c_i|c_{1:i-1}) = \prod_{i=1}^{N} \mathbb{P}(c_i|c_{i-2:i-1}).$$

Dans le cas d'un modèle trigramme d'un langage à 100 caractères, l'ensemble $\mathbb{P}(c_i|c_{i-2:i-1})$ pour tous caractères a un million d'entrées[12] et cette distribution peut être estimée précisément à l'aide d'un ensemble de textes de 10 millions de caractères au moins[13]. Nous appelons un tel ensemble de textes un *corpus* (*corpora* au pluriel), d'après le terme latin pour *corps*.

Que pouvons-nous faire avec un modèle n-grammes ? Une tâche pour laquelle les modèles sont particulièrement adéquats est *l'identification de langages* : étant donné un texte, il faut déterminer en quelle langue naturelle le texte est écrit. C'est une tâche relativement aisée

[11] Voir aussi Appendice A à la fin de ce document.

[12] Considérez une suite de trois caractères choisis parmi les 100 caractères ; il y a 100 possibilités pour le premier caractère, 100 pour le second, et 100 pour le troisième, donc $100 \times 100 \times 100$ possibilités pour le tout, c'est-à-dire un million de suites différentes de trois caractères.

[13] Le nombre de 10 millions est (sans doute) le résultat de considérations statistiques, qui permettent de mesurer l'erreur faite en remplaçant la distribution de probabilité "réelle" de la langue par le modèle de trigrammes conçu à partir de l'ensemble de textes.

: même lorsque le texte est aussi court que "Hello, world" ou "Salut les aminches", il est facile d'identifier que le premier est écrit en anglais et le second en français. Les systèmes informatiques peuvent identifier les langages avec une précision de plus de 99% ; parfois, des langues proches très proches telles que le suédois et le norvégien sont mélangées.

Une approche possible pour l'identification de langages est, premièrement, de créer un modèle trigramme de caractères pour chaque langage potentiel

$$\mathbb{P}(c_i|c_{i-2:i-1}, \ell)$$

où la variable ℓ varie sur tous les langages. Pour chaque langage ℓ, le modèle est construit en comptant les trigrammes dans un corpus de ce langage. (Il suffit d'environ 100000 caractères pour chaque langage.) Cela nous donne un modèle de

$$\mathbb{P}(\text{Texte}|\text{Langage})$$

pour tout texte et tout langage, mais nous voulons sélectionner le langage le plus probable [14] étant donné un texte, donc nous appliquons la formule de Bayes puis l'hypothèse de Markov pour obtenir le langage le plus probable :

$$\begin{aligned}\ell* &= \texttt{argmax}_l \mathbb{P}(\ell|c_{1:N}) \\ &= \texttt{argmax}_l \mathbb{P}(\ell)\mathbb{P}(c_{1:N}|\ell) \\ &= \texttt{argmax}_l \mathbb{P}(\ell) \prod_{i-1}^{N} \mathbb{P}(c_i|c_{i-2:i-1}, \ell)\end{aligned}$$

Le modèle trigramme peut être appris à partir d'un corpus, mais que faire pour évaluer la probabilité à priori $\mathbb{P}(\ell)$? Nous pouvons obtenir une estimation de ces valeurs. Par exemple, si nous sélectionnons une page web aléatoirement nous savons que l'anglais sera la langue la plus probable et que la probabilité que la page soit en macédonien sera inférieure à 1%. La valeur exacte que nous sélectionnons à priori n'est pas critique, parce que le modèle trigramme sélectionne habituellement un langage qui est beaucoup plus probable, de plusieurs magnitudes, que tout autre langage.

La correction orthographique, la classification des genres littéraires, et la détection des entitées nommées sont d'autres tâches qui conviennent aux modèles de caractères. La classification des genres littéraires signifie décider si un texte est une nouvelle de journal, un document légal, un article scientifique, etc. Bien que beaucoup de caractéristiques puissent permettre de classifier, le décompte de la ponctuation et d'autres caractéristiques des n-grammes de caractères peuvent aller très loin (Kessler *et al.*, 1997). La reconnaissance des entités nommées est la tâche qui consiste à trouver les noms de choses dans un document et à décider dans quelle classe ils se trouvent. Par exemple, dans le texte

"Monsieur Sopersteen a été prescrit de l'aciphex"

nous reconnaîtrons que "Monsieur Sopersteen" est le nom d'une personne et "aciphex" est le nom d'un médicament. Les modèles qui se situent au niveau des caractères sont bons pour ces tâches parce qu'ils peuvent associer la suite de caractères "ex_" ("ex" suivi par un espace) à un nom de médicament et "steen_" à un nom de personne, et ainsi identifier des noms que le modèle n'a jamais vus auparavant.

[14] Le symbole $\texttt{argmax}_\ell \mathbb{P}(\ell...)$ ("argument max sur ℓ") des formules ci-dessus signifie que nous cherchons quel langage ℓ (l'argument) maximise le terme $\mathbb{P}(\ell...)$.

1.2 Lisser des modèles n-gramme

La plus grosse complication des modèles n-gramme est que le corpus d'entraînement fournit seulement une estimation de la distribution de probabilités réelle. Pour des suites de caractères fréquentes telles que "_th", n'importe quel corpus en anglais donnera une bonne estimation : environ 1,5% de tous les trigrammes. [15] En revanche, "_ht" est beaucoup plus rare : aucun mot du dictionnaire ne commence par "ht". Il est très probable que cette séquence aie un compte de 0 dans un corpus d'entraînement en anglais standard. Cela signifie-t-il qu'il faudrait conclure que $\mathbb{P}(\text{"}th\text{"}) = 0$? Si tel était le cas, alors le texte "The program issues an http request" aurait une probabilité de 0 en anglais, ce qui ne semble pas très juste[16]. Nous avons là affaire à un problème de généralisation : nous voulons que nos modèles de langue puissent être généralisés correctement à des textes qu'ils n'ont pas encore vus. Notre modèle ne devrait pas affirmer que l'on ne peut pas rencontrer "_http" parce qu'il ne l'a jamais vu. Donc, nous allons ajuster notre modèle de langue afin de donner aux suites dont le décompte est zéro une probabilité très faible mais non-nulle (et les autres décomptent seront légèrement baissés afin que la somme des probabiltiés fasse toujours 1).[17] Ce procédé d'ajustage des probabilités des suites dont le décompte est très faible s'appelle le *lissage*[18].

Le type de lissage le plus simple a été proposé par Pierre-Simon Laplace au XVIII^{e} siècle : il soutint que, sans plus d'information, si une variable aléatoire Booléenne[19] X a pris la valeur `Faux` sur n observations, alors l'estimation pour $\mathbb{P}(X = \texttt{Vrai})$ doit être $\frac{1}{(n+2)}$. C'est-à-dire qu'il suppose que, parmi deux nouvelles observations, l'une sera `Vrai` et l'autre `Faux`. Le lissage de Laplace, (aussi appelé lissage-plus-un[20]) est un pas dans la bonne direction, mais ne fonctionne pas si bien que ça. Le *modèle à reculons*[21] est une meilleure approche, et consiste à commencer à décompter les n-grammes ; mais, si une suite a un nombre d'occurrences bas (ou nul), nous reculons et utilisons des $n-1$-grammes.

Le *lissage à interpolation linéaire* est un modèle à reculons qui combine, par interpolation linéaire, un modèle trigramme, un modèle bigramme, et un modèle unigramme. Il définit l'estimation de probabilité par la formule

$$\hat{\mathbb{P}}(c_i|c_{i-2:i-1}) = \lambda_3\mathbb{P}(c_i|c_{i-2:i-1}) + \lambda_2\mathbb{P}(c_i|c_{i:i-1}) + \lambda_1\mathbb{P}(c_i|c_i),$$

où $\lambda_3 + \lambda_2 + \lambda_1 = 1$. Les paramètres λ_i peuvent être fixes ou entraînés à l'aide d'un algorithme de maximisation de l'espérance. Il est également possible de faire en sorte que la valeur de λ_i dépende des décomptes : si l'on a un grand nombre de trigrammes, alors on leur donne un poids plus important ; sinon, on donne un poids plus important aux modèles bigramme et unigramme. Un front de chercheurs a développé des modèles de lissage de plus en plus sophistiqués, tandis qu'un autre front suggère de réunir des corpus de plus en plus larges afin que même des modèles simples puissent fonctionner. Dans tous les cas, les chercheurs ont le même but : réduire la variance du modèle de langue [22].

[15] En français, le résultat est similaire avec le trigramme "_de" : environ 1,15% de tous les trigrammes - vérifié dans la version de la *Bible* de Segond. Le 4-gramme "_et_" est, sans surprise, le plus fréquent avec 0,79%, Le 5-gramme "_et_" est le plus fréquent, etc.

[16] "an http request", une "requête http", est un terme fréquent en informatique des réseaux qui intervient dans les descriptions techniques de connexions à internet.

[17] C'est là une propriété importante des probabilités.

[18] "Smoothing" en anglais.

[19] Booléenne signifie qu'elle peut prendre deux valeurs, `Vrai` ou `Faux` ; c'est équivalent au lancer d'une pièce qui peut tomber soit sur `Pile`, soit sur `Face`.

[20] "Add-one smoothing".

[21] "Backoff model".

[22] En probabilités, la variance peut être comprise comme l'éloignement par rapport à une moyenne. En quelque sorte, les chercheurs essayent de concevoir des modèles stables qui, lorsqu'ils sont face à une situation imprévue telle qu'un mot qui n'existe pas, ne fournissent pas de prédictions insensées.

Il y a une complication : notez que lorsque $i = 1$ dans le terme $\mathbb{P}(c_i|c_{i-2:i-1})$, il faut calculer $\mathbb{P}(c_1|c_{-1:0})$, mais il n'y a pas de caractères avant c_1. Nous pouvons introduire des caractères artificiels, par exemple, en définissant c_0 comme un espace, ou un caractère spécial de "début de texte". Nous pouvons également reculer et revenir à un modèle de Markov d'ordre inférieur, et ainsi définir $c_{-1:0}$ comme la suite de taille nulle, et ainsi définir $\mathbb{P}(c_1|c_{-1:0}) = \mathbb{P}(c_1)$.

1.3 Évaluation d'un modèle

Avec autant de modèles n-gramme différents – unigramme, bigramme, trigramme, lissage interpolé avec différentes valeurs de λ, etc. – comment savoir quel modèle choisir ? Nous évaluons un modèle par évaluation croisée[23]. Séparez le corpus en un corpus d'entraînement et un corpus de validation ; déterminez les paramètres du modèle à partir des données d'entraînement ; puis évaluez le modèle sur le corpus de validation[24].

L'évaluation peut être une mesure spécifique à une tâche, telle que la mesure de la précision de la détection d'une langue. Alternativement, il est possible d'utiliser une mesure de la qualité d'un modèle qui ne dépend pas d'une tâche spécifique : calculez la probabilité assignée au corpus de validation par le modèle ; plus la probabilité est élevée, mieux c'est. Cette mesure n'est pas très pratique parce que la probabilité d'un grand corpus sera un nombre très petit, et alors les problèmes de précision et de virgule flottante apparaissent. Une manière différente de décrire la probabilité d'une suite est d'utiliser une mesure que l'on appelle la *perplexité*, définie par

$$\texttt{Perplexity}(c_{1:N}) = \mathbb{P}(c_{1:N})^{\frac{1}{N}}.$$

On peut voir la perplexité comme la réciproque de la probabilité, normalisée par la taille de la suite [25]. On peut également la voir comme la moyenne pondérée du facteur de branchement d'un modèle. Supposez qu'il y ait 100 caractères dans notre langage, et que notre modèle dise qu'ils soient tous équiprobables. Alors, quelle que soit la taille d'une suite, sa probabilité sera de 100.[26]

Si certains caractères sont plus probables que d'autres, et que le modèle le reflète, alors le modèle aura une perplexité inférieure à 100.

1.4 Modèles de mots n-gramme

À présent, nous nous intéressons aux modèles de mots n-grammes plutôt qu'aux modèles de caractères. Les mêmes mécanismes s'appliquent de la même manière aux modèles de mots et aux modèles de caractères. La différence principale se situe au niveau du *vocabulaire* – l'ensemble de symboles qui constitue le modèle et le corpus) : il est plus grand. Il n'y a qu'environ 100 caractères dans la plupart des langues et, parfois, nous construisons des modèles de caractères qui sont encore plus restrictifs, en considérant que les symboles "A" et "a" sont les mêmes, ou en considérant toutes les marques de ponctuation comme étant le même symbole. Mais, dans le cas des modèles de mots, nous avons au moins des dizaines de milliers de symboles, parfois des millions. Le nombre faramineux de possibilités est dû au fait que ce qui constitue un mot n'est pas entièrement clair. En anglais une suite de lettres séparée par des espaces constitue un mot,

[23] "Cross-validation".

[24] Aussi appelé "corpus de test".

[25] La perplexité est l'inverse de la racine N-ième de la probabilité. Une formule équivalente de la perplexité est $\frac{1}{\sqrt[N]{\mathbb{P}(c_{1:N})}}$.

[26] En effet, si tous les caractères sont équiprobables (et valent $\mathbb{P}(c_i) = \frac{1}{100}$ pour tout i), alors en particulier ils sont indépendants, et la probabilité d'une suite de caractère est le produit de la probabilité de chaque caractère : $\mathbb{P}(c_{1:N}) = \prod_1^N \mathbb{P}(c_i)$. Donc, $\texttt{Perplexity}(c_{1:N}) = \mathbb{P}(c_{1:N})^{-\frac{1}{N}} = (\prod_1^N \mathbb{P}(c_i))^{-\frac{1}{N}} = ((\frac{1}{100})^N)^{-\frac{1}{N}} = 100$.

mais, dans d'autres langues telles que le chinois, les mots ne sont pas séparés par des espaces ; même en anglais, il faut prendre des décisions pour avoir une politique claire sur ce qui constitue un mot : combien de mots y-a-t'il dans "ne'er-do-well"[27] ? Ou dans "(Tel:1-800-960-5660x123)" ?

Les modèles de mots n-grammes doivent gérer les mots *hors du vocabulaire*. Dans le cas des modèles de caractères, nous n'avions pas à nous inquiéter de l'invention d'une nouvelle lettre de l'alphabet (à l'exception du travail novateur de T. Geisel en 1955). Mais, dans le cas des modèles de mots, il y a toujours le risque d'un mot inconnu qui n'a pas été rencontré dans le corpus d'entraînement, ce que nous devons donc modéliser explicitement dans notre modèle de langue. Pour cela, nous pouvons ajouter un nouveau mot à notre vocabulaire : `<UNK>`, qui signifie "mot inconnu"[28]. Nous pouvons estimer le compte des n-grammes qui contiennent `<UNK>` avec cette astuce : parcourez le corpus d'entraînement et, puisqu'un mot est inconnu la première fois que vous le rencontrez, remplacez-le par `<UNK>`. Gardez les autres occurrences du mot inchangées. Puis, comptez les n-grammes du corpus comme d'habitude, en comptant `<UNK>` comme n'importe quel autre mot. Puis, lorsqu'un mot inconnu apparaît dans un ensemble de test, vous lui donnerez la probabilité de `<UNK>`. Parfois, on utilise plusieurs symboles de mots inconnus correspondant à différentes classes. Par exemple, toute suite de nombres pourra être remplacée par `<NUM>`, ou toute adresse e-mail par `<EMAIL>`.

Pour nous faire une idée de ce que peuvent faire les modèles de langues, nous avons construit des modèles à unigrammes, bigrammes, et trigrammes à partir de tous les mots de cet ouvrage[29] et avons tiré au hasard des suites de mots à partir des modèles[30]. Nous obtenons

- Unigramme : logical are as are confusion a may right tries agent goal the was . . .
- Bigramme : systems are very similar computational approach would be represented . . .
- Trigramme : planning and scheduling are integrated the success of naive bayes model is . . .

Même avec ce petit extrait, il est clair que le modèle à unigrammes est une approximation assez pauvre à la fois de l'anglais et du contenu d'un livre sur l'IA, tandis que les modèles à bigrammes et trigrammes sont bien meilleurs. Les modèles sont d'accord avec ce sentiment : la perplexité du modèle à unigrammes est de 891, celle du modèle à bigrammes de 142, et celle du modèle à trigrammes de 91.

Maintenant que nous avons vu les bases des modèles n-grammes – à caractères ou à mots –, nous pouvons nous tourner vers des tâches liées au langage.

2 Classification du texte

(Non traduit ici)

[27] "Bon à rien".

[28] "**Unk**nown word" en anglais.

[29] Les auteurs faisant référence aux termes de l'AIMA tout entier, je n'ai pas traduit les résultats de leurs tirages. À la place, j'ai entraîné des modèles à unigrammes, bigrammes, et trigrammes sur la *Bible* de Segond pour obtenir, après tirage aléatoire :

- ' , aï ' là , - le souper je
- lèpre , en osée à mon vérité je parce que tes jours de shallum pas vers - t béer -
- lévi , avait , des lyres celui qui laissera n ' apporteras mais maintenant je la surface de et l ' afin qu ' , si nous méchants périront ,

[30] Il est donc sous-entendu ici que les suites les plus probables dans les modèles de langues sont aussi les suites les plus probables lors des tirages.

3 Recherche d'information

(Non traduit ici)

4 Extraction d'information

(Non traduit ici)

5 Résumé

(traduction à venir)

A Chaînes de Markov pour les linguistes

Cette section consiste en une courte introduction aux probabilités et aux chaînes de Markov dans le but de pouvoir comprendre les modèles de langues des sections ci-dessus.

Pour plus de détails, examinez l'ouvrage *Probabilités, variables aléatoires, lois classiques* de Virginie Delsart et Nicolas Vaneecloo, en particulier le Chapitre 1 qui discute de la notion de probabilités. Vous pouvez également lire le tome *Probabilités* de la collection Que-Sais-je.

A.1 Une courte introduction aux probabilités

Comme souvent en sciences, nous pouvons expliquer le même domaine à différents degrés de précision suivant l'audience à laquelle on s'adresse. Parfois même, certains détails sont omis au profit de la compréhension immédiate et aux dépends des dernières avancées scientifiques. Couper court aux débats philosophiques ou techniques contemporains peut être une nécessité en classe étant données les contraintes temporelles. Toutefois, la solution qui consiste à ne présenter qu'une unique facette d'un domaine cache hélas la complexité qui en fait son intérêt et, pire encore peut-être, transmet l'impression trompeuse que les théories scientifiques sont figées et que la vérité existe, une fois pour toute, transmise par le professeur, ce qui peut être frustrant lorsque l'on découvre qu'il faut mettre ses connaissances à jour. (Le problème n'étant pas d'avoir été frustré, mais de n'avoir pas été entraîné à cette "frustration").

C'est le cas en mathématique : on vous a peut-être appris qu'on ne peut pas prendre la racine carrée d'un nombre négatif mais, plus tard, on vous a montré les *nombres imaginaires* et, en particulier, $\mathtt{i}$, défini par la relation $\mathtt{i}^2 = -1$. En physique atomique, on vous a montré le modèle planétaire de Bohr avant de vous parler de modèles plus complexes fondés sur la mécanique quantique. En anglais, cette méthode pédagogique porte le nom de "lie-to-children" (mensonge-aux-enfants).

Un petit cours ou un ouvrage peut aussi cacher à la fois les années de recherches qu'ont nécessité un résultat (chose très fréquente en mathématiques, où les résultats et les preuves sont "nettoyées" de toute sueur humaine), et un ensemble de suppositions théoriques qui font que les termes utilisés, les notions utilisées, les résultats obtenus sont "permis" et sont construits sur des bases solides.

Sachez donc que cette courte introduction aux probabilités repose sur une vision particulière des probabilités, certes populaire, applicable, et utilisée par de très sérieux chercheurs, mais qui n'est ni indiscutable ni la seule vision des probabilités. Sachez également que derrière la pratique se cache une théorie très sérieuse (l'axiomatisation de Kolmogorov).

Le concept même de probabilité porte à discussion, de même que sa relation au monde réel - comme tout modèle mathématique. Ici, dans cette introduction aux probabilités, et puisque

nous travaillons sur du texte, des suites finies de caractères, des symboles... nous pouvons nous contenter de la définition des probabilités proposée par Pascal :

> La probabilité d'un événement est le rapport du nombre de cas qui correspondent à l'événement sur le nombre de cas possibles.

Vous pouvez l'en voir décrire l'application dans sa Seconde Lettre à Fermat.

Par exemple, considérons l'ensemble des lettres de l'alphabet. L'expérience probabiliste consiste à choisir une lettre au hasard. La probabilité de l'événement "obtenir la lettre A", que l'on peut noter $\mathbb{P}("A")$, est calculée par

$$\begin{aligned}\mathbb{P}("A") &= \frac{\text{cas "obtenir la lettre A"}}{\text{cas "obtenir un A", "obtenir un B", etc.}} \\ &= \frac{1}{26}.\end{aligned}$$

La probabilité de l'événement "obtenir une voyelle", étant donné que les voyelles en français sont au nombre de 5 selon certains décomptes (A, E, I, O, et U), vaut

$$\mathbb{P}(\text{obtenir une voyelle}) = \frac{5}{26}.$$

Notez, ce qui est très important (et l'une des choses qui sont souvent cachées), que lorsque l'on mène l'expérience, il faut que chaque tirage soit *équiprobable*, c'est-à-dire que chaque lettre aie la même probabilité d'être piochée $(\frac{1}{26})$ [31].

Nous retrouvons là des questions et des calculs que nous avons l'habitude d'entendre dans la vie courante : quelle est la probabilité de gagner au loto ? À la loterie ? De piocher un as dans un jeu de cartes ? Que la pièce qu'on a lancée tombe sur pile ?

A.2 Chaînes de Markov

Au début du XX^e, le mathématicien russe A.A. Markov étudie la succession des voyelles et des consonnes et formalise la notion de probabilités en chaînes. Il peut ainsi, étant donné un certain texte, déterminer la probabilité qu'une voyelle suive une consonne

$$\mathbb{P}(V|C),$$

qui se lit "probabilité de V *sachant* C". Ce qui est sous-entendu est que l'on révèle, l'une après l'autre, les lettres du texte, et que l'on cherche à savoir la probabilité que l'on obtienne une voyelle ou une consonne.

Suffit-il d'évaluer la probabilité d'obtenir une voyelle $\mathbb{P}(V)$? Non : ce qui est important est que l'on *sait* que l'on a déjà lu une consonne préalablement, ce qui est une *condition* pour le calcul de la probabilité que l'on obtienne une voyelle ; on dit que $\mathbb{P}(V|C)$ est une *probabilité conditionnelle*. Si l'on tirait les lettres au hasard, de manière équiprobable, alors il suffirait en effet de calculer

$$\mathbb{P}(\text{obtenir une voyelle}) = \frac{5}{26}$$

comme nous l'avons fait précedemment. Mais un texte en langue naturelle suit certaines "règles" qui l'écartent d'une pure génération aléatoire de lettres et d'espaces telle que la séquence

[31] Cela soulève un problème logique, lorsque l'on tente de définir la notion de probabilité *en supposant qu'on la calcule à l'aide d'événements équiprobables* : pour définir l'équiprobabilité, nous avons besoin de la notion de probabilité, ce qui amène à une définition circulaire. Les écoles de pensée probabiliste dites "fréquentistes" ou "subjectivistes", puis plus tard l'axiomatisation de Komogorov au début du XX^e offrent des réponses à ce problème, mais, ici, il nous suffit de savoir que l'on peut calculer cette probabilité.

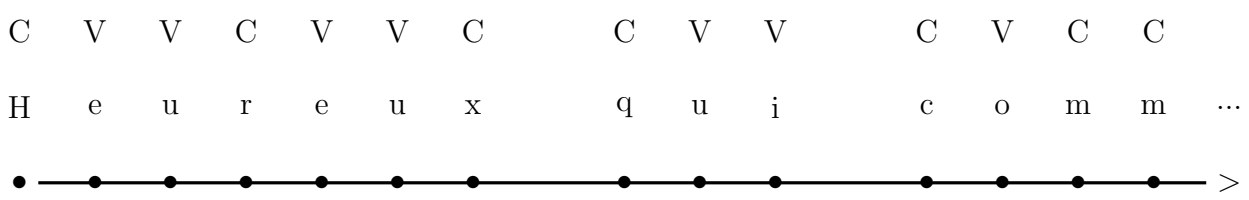

Figure 1: Succession de lettres dans l'axe du texte.

```
lsxmhyhui wzw fzznsrlscngzhiewrw
```

de 32 lettres (et un espace) équiprobables. Dans un texte en langue naturelle, si vous avez déjà lu les lettres `chevau`, vous allez vous attendre à lire la lettre `x`, *avant même de l'avoir lue.* Si vous avez déjà lu les lettres `cheva`, vous pourrez vous attendre à lire la lettre `l` (pour cheval, chevalin, etc.), ou la lettre `u` (pour chevaux, etc.). Donc, *sachant* que Markov a lu une consonne dans son texte, il cherche à évaluer la probabilité de lire une voyelle.

Évaluons cette probabilité dans le texte suivant de Joachim du Bellay :

```
Heureux qui, comme Ulysse, a fait un beau voyage,
Ou comme cestuy-là qui conquit la toison,
Et puis est retourné, plein d'usage et raison,
Vivre entre ses parents le reste de son âge !

Quand reverrai-je, hélas, de mon petit village
Fumer la cheminée, et en quelle saison
Reverrai-je le clos de ma pauvre maison,
Qui m'est une province, et beaucoup davantage ?

Plus me plaît le séjour qu'ont bâti mes aïeux,
Que des palais Romains le front audacieux,
Plus que le marbre dur me plaît l'ardoise fine :

Plus mon Loire gaulois, que le Tibre latin,
Plus mon petit Liré, que le mont Palatin,
Et plus que l'air marin la doulceur angevine.
```

Reprenant la figure de Micheline Petruszewycz, nous pouvons voir l'expérience aléatoire de lecture du texte comme le parcours d'une chaîne perlée de lettres, dans le sens de lecture du texte (Figure 1).

La particularité du calcul de Markov est qu'à chaque perle, ou à chaque lettre, Markov *suppose* que la probabilité de l'événement suivant (voyelle ou consonne) *ne dépend que* de ce qui vient de se passer, c'est-à-dire la lecture d'une voyelle ou d'une consonne. Cette supposition très forte permet de simplifier énormément les calculs mais a déjà permis à Markov d'obtenir des résultats intéressants (par la suite, il étendra son modèle en regardant *deux* lettres au lieu d'une).

Évaluons à présent la probabilité $\mathbb{P}(V|C)$. Pour cela, nous pouvons utiliser la définition des probabilités de Pascal, c'est-à-dire

$$\mathbb{P}(V|C) = \frac{\text{nombre de cas où V suit C}}{\text{nombre de cas où on observe C}}$$

Quelles sont les successions que l'on *observe* dans notre corpus ? Si l'on remplace les consonnes et les voyelles par les symboles 'C' et 'V', on obtient la suite :

CVVCVVCCVVCVCCVVCCCCVVCVVCVCCVVVCVCVCVVVCVCCVCVCCVCCVCVVCVCCVVCCVCVVCVC
VCCVVCVCCCVCVVCCVCCVVCCVCVCVVCCVVCVCCVCCVVCCCVCVCCVCVCCCCVCVCCVCVCVCCVC
VVCCCVCVCCVVCVCVCVCCVCVCCVCVCCVCCVCVCVCVCCVCCVCVCVVVCVCCVVCCVCVVCVCCVCV
CCVVCVCVCCVCCVCVCVVCCVCVVCVCCVVCVCCVCVCCVCVCCVVCCVVVCVVCCVCVCCVCVCCVCCV
CCVVCCVCVCVVCCVVCCCVCVCVCVVVVCCVVCVCCVCVVCCVCVVCCCVCCVCCVVCVCVVVCCCVCCV
VCVCVCCCVCVCCVCCVCCVCCVVCVCVCVCCVCCVCCVVCVCVVCVVCCVVCVCVCCVCVCVCCCVCCVC
CVCVCCVCVCVVCVCVCCCVCVCVCVCCCVCCVVCVVCCVCVCCVCVVCCVVCVCCVCVCV.

Comptons maintenant le nombre de couples $(C, \cdot)$, où $\cdot$ peut être une consonne ou une voyelle. Par la suite (vous pouvez le vérifier à la main),

$$\mathbb{P}(V|C) = \frac{\text{nombre de cas où V suit C}}{\text{nombre de cas où on observe C}} = \frac{\text{nombre de suites (C,V)}}{\text{nombre de suites (C,}\cdot\text{)}} = \frac{169}{250},$$

c'est-à-dire qu'il y a environ 67,6% de chances qu'une voyelle suive une consonne.

Question subsidiaire : quelle est la probabilité d'obtenir une voyelle, c'est-à-dire $\mathbb{P}(V)$? Ici, pas de probabilités conditionnelles. Pour évaluer cette probabilité, il suffit de calculer

$$\begin{aligned}\mathbb{P}(V) &= \frac{\text{nombre de voyelles}}{\text{nombre total de lettres}} \\ &= \frac{225}{487}\end{aligned}$$

c'est-à-dire qu'il y a environ 46,2% de chances d'obtenir une voyelle en général.

Observez que $\mathbb{P}(V|C)$ et $\mathbb{P}(V)$ sont différents ! Savoir que l'on a déjà lu une consonne apporte donc une information supplémentaire qui nous permet de prédire la lettre suivante. Notez bien que les objets que l'on compte sont très différents : dans le premier cas nous avons compté des suites de deux lettres C et V (CV et CC), alors que dans le second nous avons seulement compté les lettres C et V.