Modèles de langage N-gram

TALN semaine 3

Merci à Dan Jurafsky pour certaines des diapos et l'inspiration !

Plan pour aujourd'hui

1. **Modèles Uni-gram ou sac-de-mots**
2. **Modèles de langage N-gram et probabilités conditionnelles**
3. **Évaluation des modèles de langage**
4. **Évaluer vs générer du texte avec les LMs**
5. ***Exercices de groupe***

Qu'est-ce qu'un modèle de langage ?

- Ce sont des modèles de probabilité contextualisé appliqué au langage. Ils peuvent attribuer la probabilité aux mots en contexte et peuvent ainsi faire des choses comme **générer** le mot suivant dans une phrase, ou vous **évaluer** la probabilité d'une phrase entière sous le modèle.

- LLM = Large **language model**

Nous étudierons les modèles de langage, ajoutant à leur complexité à chaque niveau :

MODÈLES DE LANGUE

- **Modèles de langage N-gram**—Modèles probabilistes de langage contextualisé
- **Embeddings** — Sémantiques vectorielles
- **Modèles de langage neuronaux récurrents, LSTMs**—Modèles de langage neuronaux
- **Transformers** — Grand modèle de langage ou LLM
- M**odèles encodeurs** — Tâches de notation
- M**odèles encodeur-décodeurs** — tâches génératives
- **Peaufinage et apprentissage en-contexte**— le paradigme actuel de modèle fondateur

Modèle de langage sac-de-mots (unigram)

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Modèle de langage sac-de-mots (unigram)

- Le modèle de sac de mots (SdM), également connu sous le nom de modèle unigram, est un modèle de langage qui attribue des probabilités aux mots sans contexte préalable.

- Ainsi, la probabilité d'un mot est simplement sa fréquence, ou son compte normalisé dans un document ou un corpus.

Dépendances séquentielles

‘Not only was it the greatest cast’

-> positive

‘Only it was not the greatest cast’

-> negative

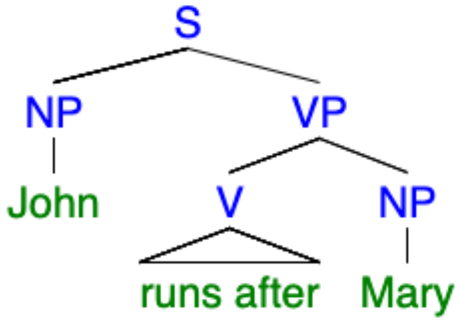
Les caractéristiques unigram sont pareils!

[cast, greatest, it, only, not, the, was]

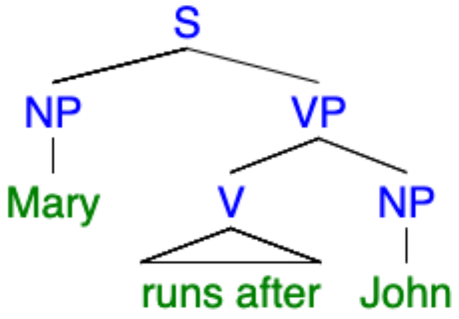
L'ordre des mots est important, le sens vient en partie des dépendances séquentielles.

Les données linguistiques… spéciales?

Il existe une cartographie entre la structure et le sens.

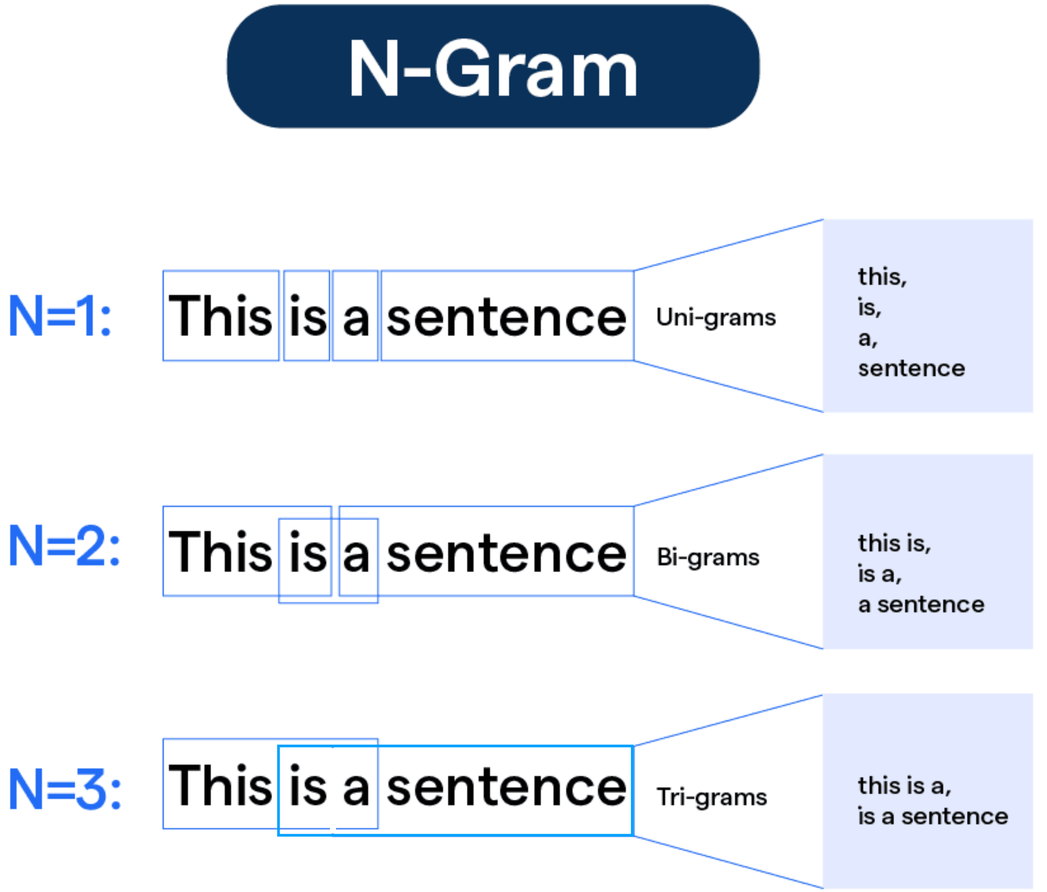


vs



Modèles N-gram

- Les modèles SdM ne représentent pas de dépendances séquentielles dans leurs distributions.
- **La solution : des modèles N-gram, une généralisation des SdM à des tailles de contexte plus larges**
- Le « N » représente la taille du contexte de dépendance :
    - modèles Uni-gram considèrent des contextes de 1 mot a.k.a uni-gram = SdM
    - modèles Bi-gram considèrent des dépendances de 2 mots
    - modèles tri-gram considèrent les dépendances de 3 mots
    - …



modèle de langage

Les modèles de langage sont des systèmes qui peuvent prédire les mots à venir :

1. Ils peuvent attribuer une probabilité à chaque mot suivant potentiel;

2. Ils peuvent attribuer une probabilité à une phrase entière.

modèle de langage

Pourquoi la prédiction de mots ?

C’est ainsi que fonctionnent les grands modèles de langage (LLMs) !

Les LLMs sont **entraînés** à prédire les mots
• Les modèles autorégressifs apprennent à prédire le mot suivant

Les LLMs **génèrent** du texte en prédisant les mots
• En prédisant le prochain mot à répétition

modèle de langage

Objectif : calculer la probabilité d'une phrase ou d'une séquence de mots **W** :

$$\mathbf{P(W) = P(w_1, w_2, w_3, w_4, w_5 \ldots w_n)}$$

Tâche associée : probabilité d'un mot à venir :

$$\mathbf{P(w_5 \mid w_1, w_2, w_3, w_4)} \quad \textbf{ou} \quad \mathbf{P(w_n \mid w_1, w_2 \ldots w_{n-1})}$$

Un modèle de langage calcule l'une ou l'autre de ces probabilités :

$$\mathbf{P(W)} \quad \text{ou} \quad \mathbf{P(w_n \mid w_1, w_2 \ldots w_{n-1})}$$

Distributions conjointes

- Une distribution sur une série de variables ou d'événements aléatoires
- Le support d'une distribution conjointe est le produit cartésien des supports des variables aléatoires.
- La fonction de masse de probabilité dépendra du fait qu'il y ait ou non des hypothèses d'indépendance.

$$P(X_1, \ldots, X_n)$$

Distributions marginales

Compte tenu d'une certaine distribution conjointe : $P(X, Y)$

- Le support de la distribution marginale $P(Y)$ est l'ensemble de valeur possible pour la variable Y

- La probabilité marginale d'une valeur $y \in Y$ est :

$$P(y) = \sum_{x \in X} P(x, y)$$

Distributions conditionnelles

- Le support d'une distribution conditionnelle est le sous-ensemble du support de la distributions conjointe où la condition Y est vraie.

- La fonction de masse de probabilité retourne les probabilités renormalisées pour le support de telle sorte qu'elles somment à 1 (nous le faisons en divisant par la probabilité marginale de Y)

$$P(X \mid Y) = \frac{P(X, Y)}{P(Y)}$$

Marginalisation versus conditionnement :

- Dans la marginalisation, nous demandons : quelle est la probabilité que le dernier mot de notre phrase soit "film"
- Dans le conditionnement, nous demandons : étant donné que le dernier mot de la phrase est « film », quelle est la probabilité de l'un des autres mots de la phrase.

Ce sont des opérations complémentaires par rapport à la distribution conjointe.

Indépendance de variable

- **L'indépendance** peut être définie de deux façons
    - Les variables sont indépendantes si leur distribution conjointe est le produit des distributions marginales

$$P(X_1, \ldots, X_n) = P(X_1) \cdot \ldots \cdot P(X_n)$$

    - Les variables sont indépendantes si leurs distributions conditionnelles sont égales à leurs distributions marginales

$$P(X \mid Y) = P(X)$$

Formule des probabilités composées

- Une décomposition de la probabilité conjointe en utilisant des probabilités conditionnelles et marginales
- Fonctionne à la fois pour les variables indépendantes et dépendantes
- Peut être présenté dans n'importe quel ordre!

$$\Pr(X_1, \ldots, X_n) = \Pr(X_1) \cdot \Pr(X_2 \mid X_1) \cdot \ldots \cdot \Pr(X_n \mid X_1, \ldots X_{n-1})$$

$$= \Pr(X_n) \cdot \Pr(X_{n-1} \mid X_n) \cdot \ldots \cdot \Pr(X_1 \mid X_2, \ldots, X_n)$$

Formule des probabilités composées

La formule appliquée pour calculer la probabilité conjointe des mots dans une phrase

$$P(w_{1:n}) = P(w_1)P(w_2|w_1)P(w_3|w_{1:2})\ldots P(w_n|w_{1:n-1})$$
$$= \prod_{k=1}^{n} P(w_k|w_{1:k-1})$$

P(“The water of Walden Pond”) =

P(The) × P(water|The) × P(of|The water)

× P(Walden|The water of) × P(Pond|The water of Walden)

Assumer le propriété de Markov

Andrei Markov

- Les modèles N-gram sont également connus sous le nom de modèles de Markov, car ils suivent la propriété de Markov.

- **propriété de Markov :** La probabilité d'événements futurs dans une distribution conjointe ne dépend que de l'événement présent et est conditionnellement indépendante de celle d'événement passé. On peut donc la décomposée en tant que telle.

Propriété de Markov dans un modèle bi-gram

$$P(\texttt{blue}|\texttt{The water of Walden Pond is so beautifully})$$

$$\approx \quad P(\texttt{blue}|\texttt{beautifully})$$

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1})$$

modèles N-gram

- **Uni-gram LM:**

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i)$$

- **Bi-gram LM:**

$$P(w_1 w_2 \ldots w_n) \approx \prod_{k=1}^{n} P(w_k | w_{k-1})$$

- **Tri-gram LM:**

$$P(w_1 w_2 \ldots w_n) \approx \prod_{k=1}^{n} P(w_k | w_{k-2} w_{k-1})$$

modèles N-gram

P(‘John chased Mary’) =

- **Uni-gram LM:**

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i)$$

P(‘John’) x P(‘chased’) x P(‘Mary’)

- **Bi-gram LM:**

$$P(w_1 w_2 \dots w_n) \approx \prod_{k=1}^{n} P(w_k | w_{k-1})$$

P(‘\PAD John’) x P(‘John chased’) x P(‘chased Mary’) x P(‘Mary \PAD’)

- **Tri-gram LM:**

$$P(w_1 w_2 \dots w_n) \approx \prod_{k=1}^{n} P(w_k | w_{k-2} w_{k-1})$$

P(‘\PAD \PAD John’) x P(‘\PAD John chased’) x P(‘John chased Mary’) x P(‘chased Mary \PAD’) x P(‘Mary \PAD \PAD’)

Évaluations

Comment pouvons-nous évaluer l'efficacité ou la précision d'un modèle de langage ?

- **Évaluations extrinsèques :** Évaluer les modèles sur leur comportement externe, tel que leur performance sur les tâches en aval (par ex. Si nous utilisons un LM pour générer des variables pour la classification)

- **Évaluations intrinsèques :** Évaluation du comportement interne des modèles, c'est-à-dire l’exactitude de leur distributions de probabilité contextualisée. Mais comment ?

Perplexité

- C'est la mesure intrinsèque communément utilisée pour évaluer les LM.
- Basé sur l'intuition que les LM devraient imiter nos jugements grammaticals, de sorte que les phrases grammaticales devraient être plus probables que les phrases non grammaticales... ou que les phrases que nous pensons être plus probables devraient également être plus probables pour le modèle.
- **Perplexité:** Une mesure de la probabilité ou *de la surprise* d'un ensemble test sous un modèle.

Perplexité

Perplexity = La probabilité inverse de l'ensemble de tests (1 divisé par probabilité de l'ensemble de tests), normalisée par le nombre de mots (ou de jetons)

$$\begin{aligned} \text{perplexity}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}} \end{aligned}$$

Ou on peut appliquer la formule des probabilités composés a *W:*

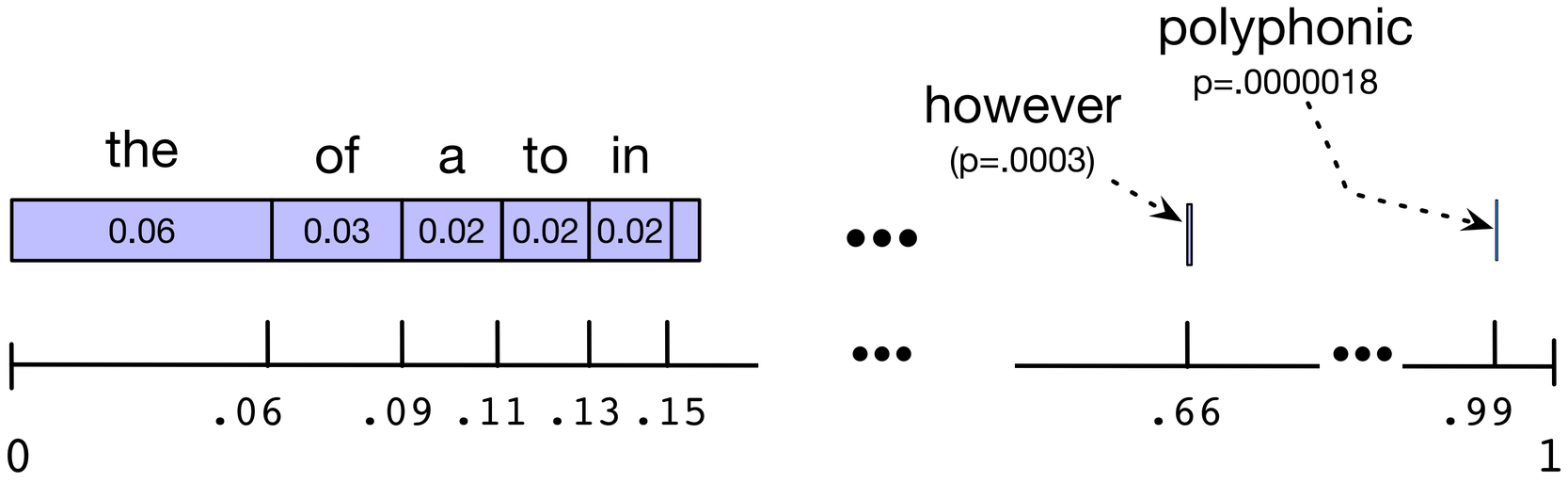
$$\text{perplexity}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}$$

Échantillonner/Générer

Nous avons vu comment **évaluer la probabilité** d'une phrase selon notre modèle de langage, mais comment pouvons-nous utiliser un modèle de langage pour **générer de nouvelles phrases probables** sous notre modèle ?

Échantillonner/Générer

Échantillonner un mot d'une distribution



Échantillonner/Générer

Pour échantillonner une phrase, vous échantillonnez récursivement à partir de la distribution conditionnelle de N-gram. Voici un exemple avec un modèle bi-gram.

Échantillonnez un bigram aléatoire (<s>, w) de la distribution conditionnelle P(w | <s>)

Ensuite, échantillonnez un bigram aléatoire (w, x) de P(x | w)

Et ainsi de suite jusqu'à ce que <\s> soit échantillonné

<s> I
I want
want to
to eat
eat Chinese
Chinese food
food </s>

I want to eat Chinese food

Échantillonner/Générer

Les échantillons de différents modèles N-gram correspondent au corpus du Wall Street Journal

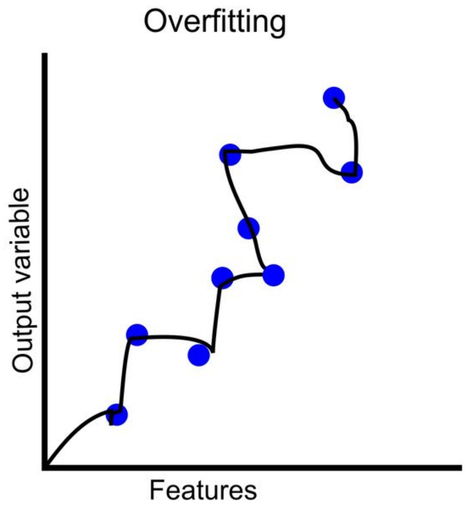
1 gram	Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives
2 gram	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her
3 gram	They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

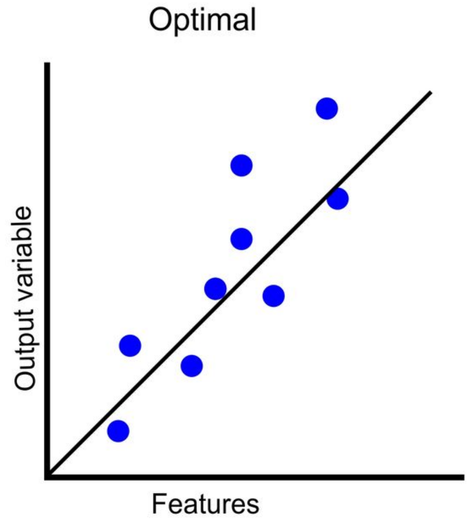
Surapprentissage versus généralisation

- Les modèles de langage sont entraînés/ajustés à un ensemble de données d’entraînement - c'est-à-dire qu'ils apprennent à représenter la distribution contextuelle des ces données.

- Si la vraie distribution de données d'entraînement diffère de la vraie distribution de données test :

➡**Le modèle peut être surajusté à la distribution des données d’entraînement et être incapable de généraliser aux données de test non vues.**



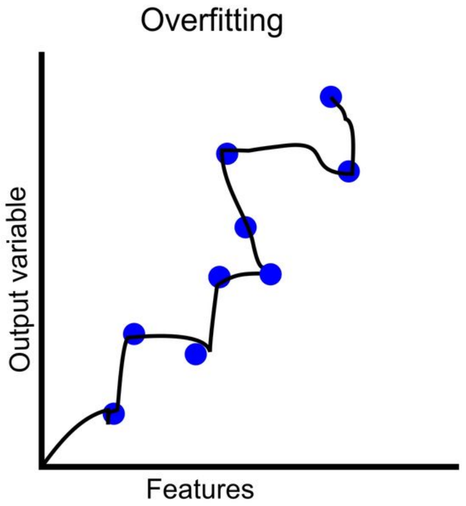


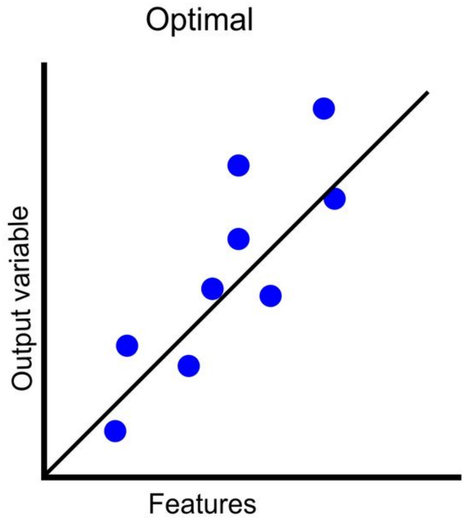
Surapprentissage versus généralisation

- **Pour une bonne généralisation, nous voulons deux choses :**

    1. **Les données d'entraînement et de test qui sont représentatives les unes des autres ;**

    2. **Le modèle est entraîné de manière optimale à ne pas surajuster aux données.**





[pause - 15 minutes]

Utiliser les modèles N-gram !

Mettez-vous en équipe !

Ouvrez les exercices/week 3 dans votre repo du cours et commencez à écrire/exécuter du code !