Traitement du langage naturel

IFT6758 - Science des données

Sources:

http://demo.clab.cs.cmu.edu/NLP/

http://u.cs.biu.ac.il/~89-680/

And many more …

Mila

Université de Montréal

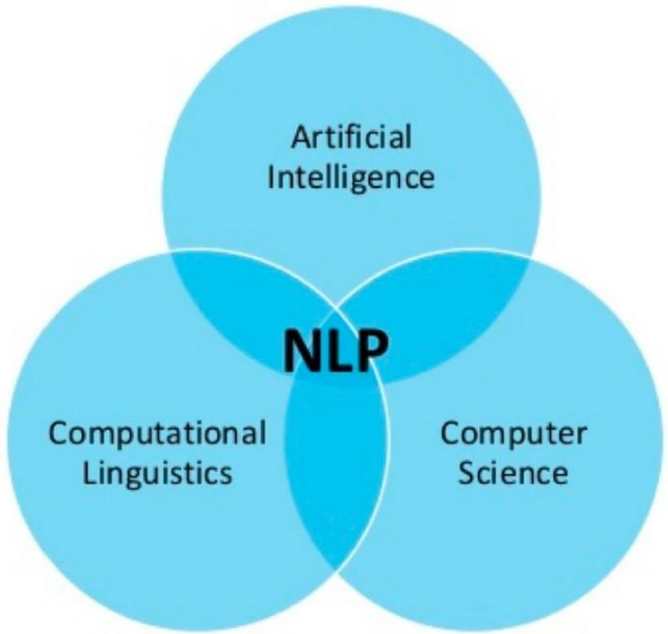
Qu’est-ce que le traitement du langage naturel (NLP) ?

- Automatisation de l’analyse, de la génération et Acquisition du langage humain (« naturel »)

  - **Analyse** (ou « compréhension » ou « traitement ») : l’entrée est le langage, la sortie est une représentation qui soutient une action utile

  - **Génération :** l’entrée est la représentation, la sortie est le langage

  - **Acquisition :** obtention de la représentation et des algorithmes nécessaires, à partir des connaissances et des données

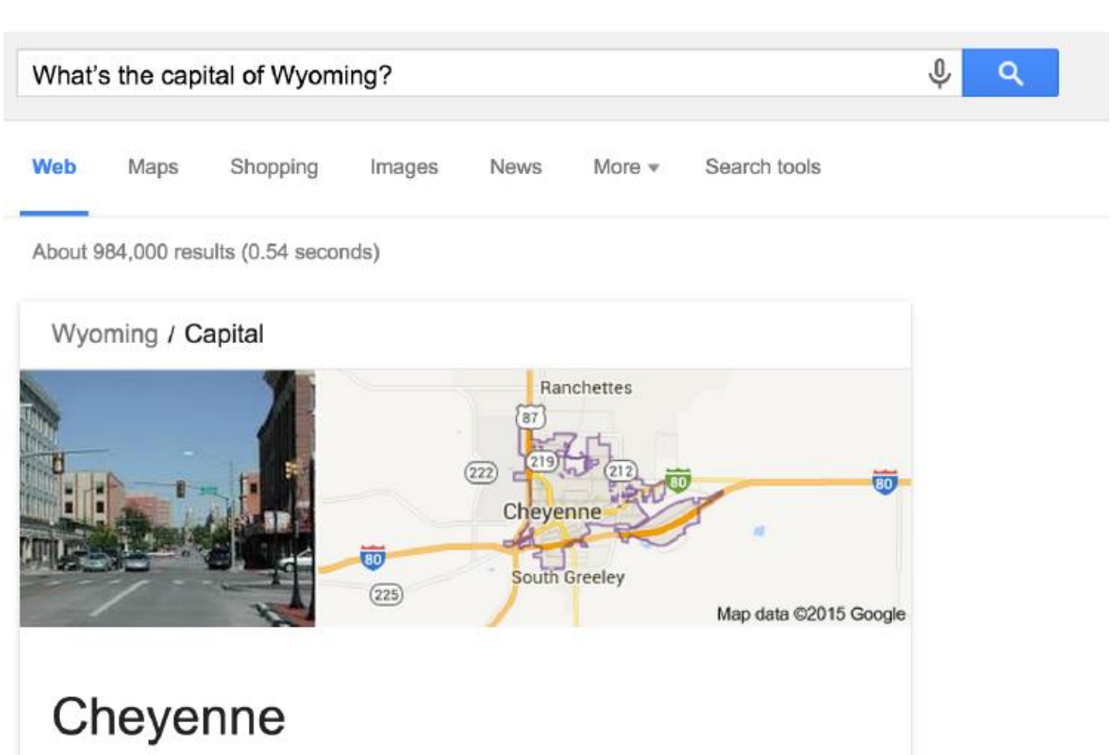


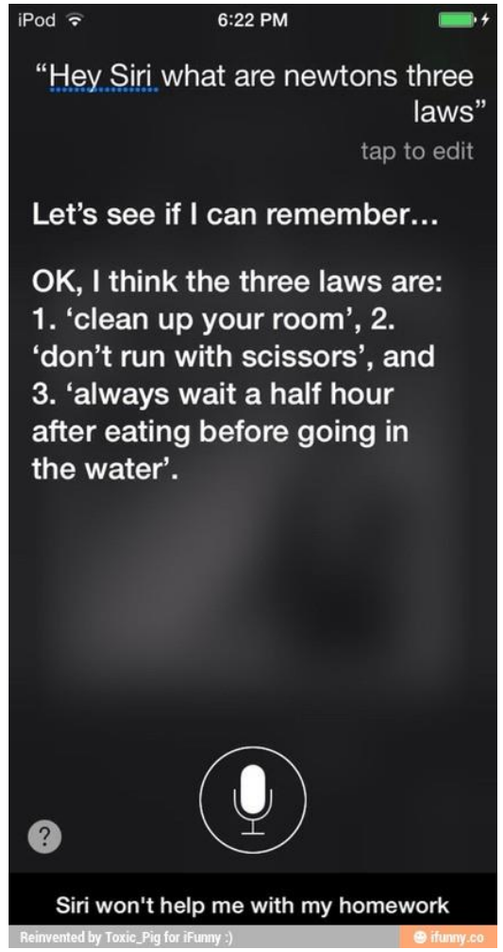
Applications du NLP

Group 1	Group 2	Group 3
Cleanup, Tokenization	Information Retrieval and Extraction (IR)	Machine Translation
Stemming	Relationship Extraction	Automatic Summarization/ Paraphracing
Lemmatization	Named Entity Recognation (NER)	Natural Language Generation
Part of Speech Tagging	Sentiment Analysis/Sentance Boundary Dismbiguation	Reasoning over Knowledge Based
Query Expansion	World sense and Dismbiguation	Quation Answering System
Parsing	Text Similarity	Dialog System
Topic Segmentationand Recognation	Coreference Resolution	Image Captioning & other Multimodel Tasks
Morphological Degmentation (Word/Sentences)	Discourse Analysis	

Traduction automatique

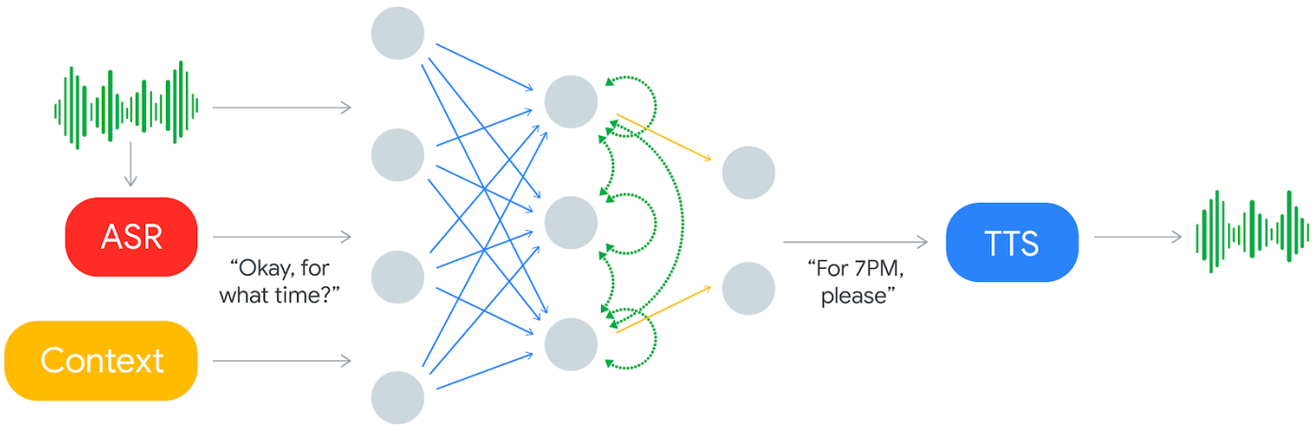
Réponses aux questions

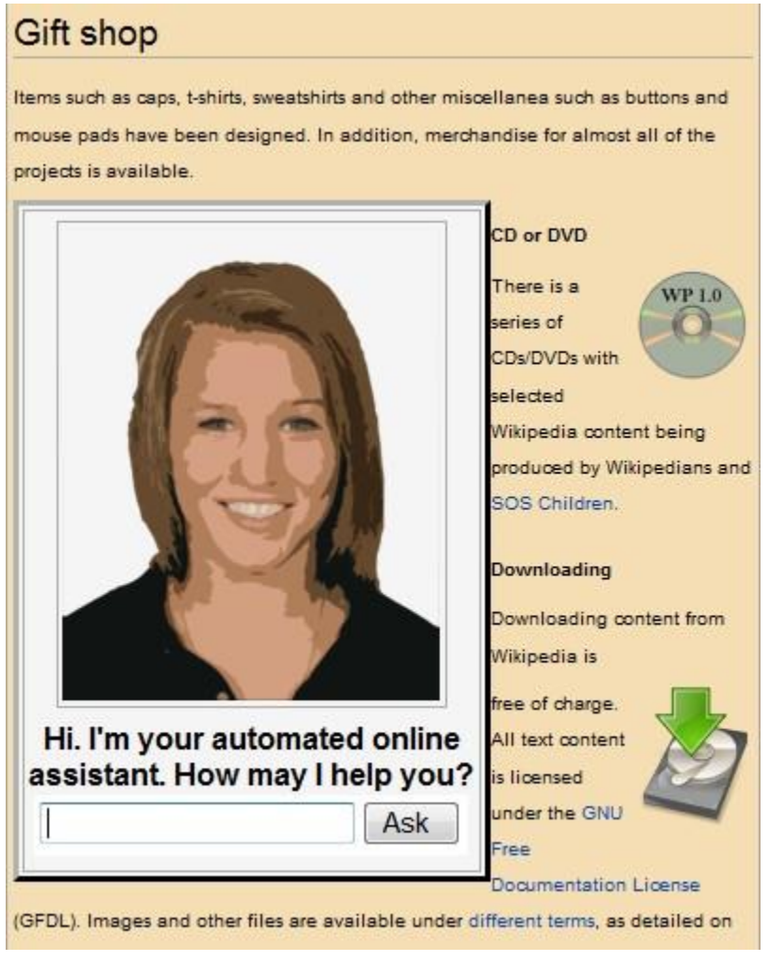




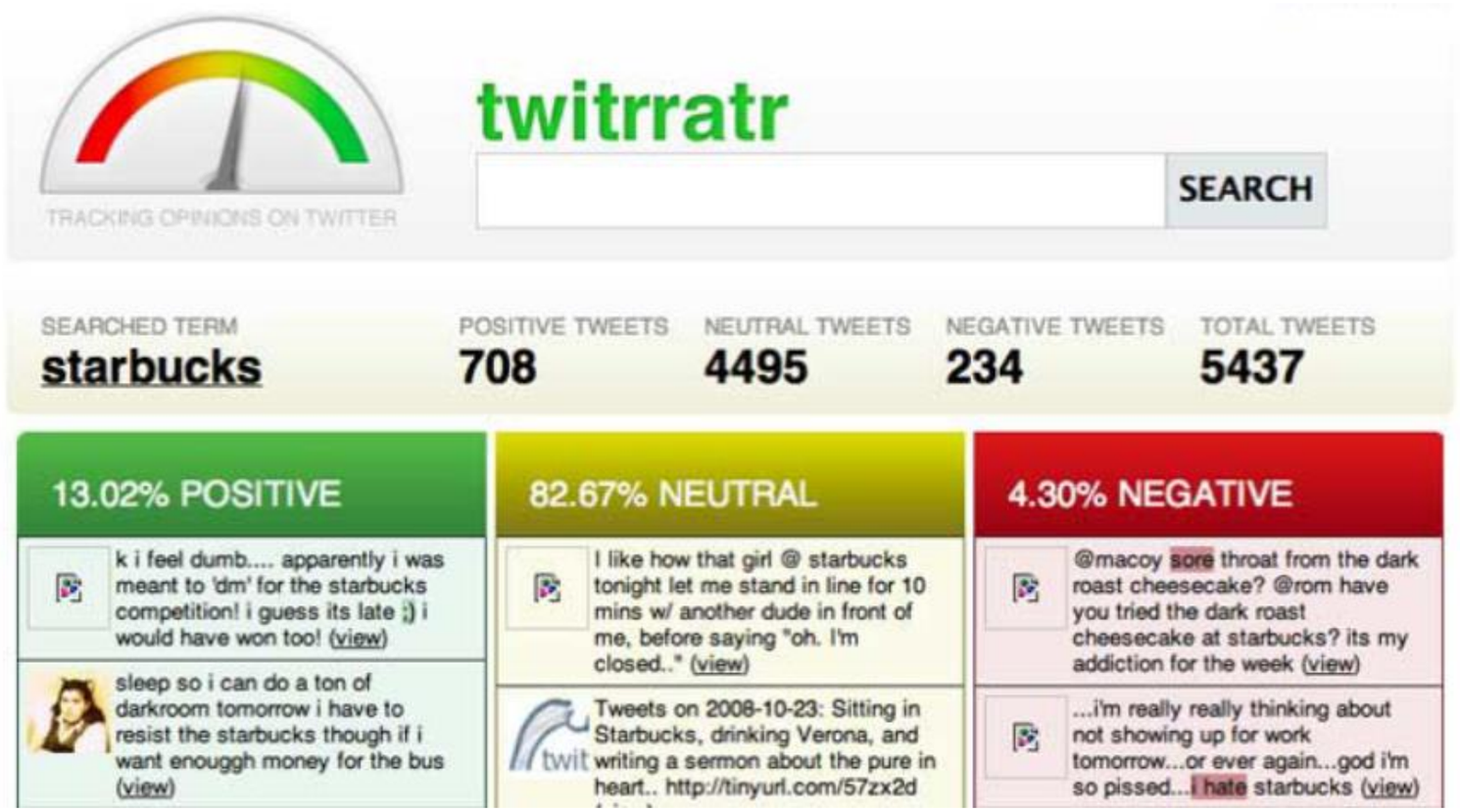
credit: ifunny.com

Système de dialogue

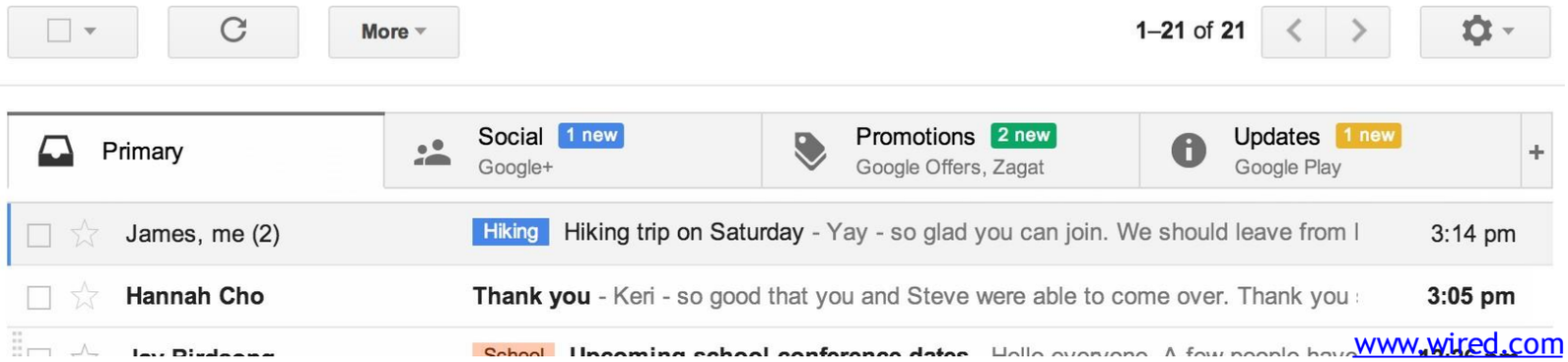




Analyse des sentiments et des opinions



Classification de texte



www.wired.com

To cela est fait grâce à des modèles auto-récursifs!!!

Mila

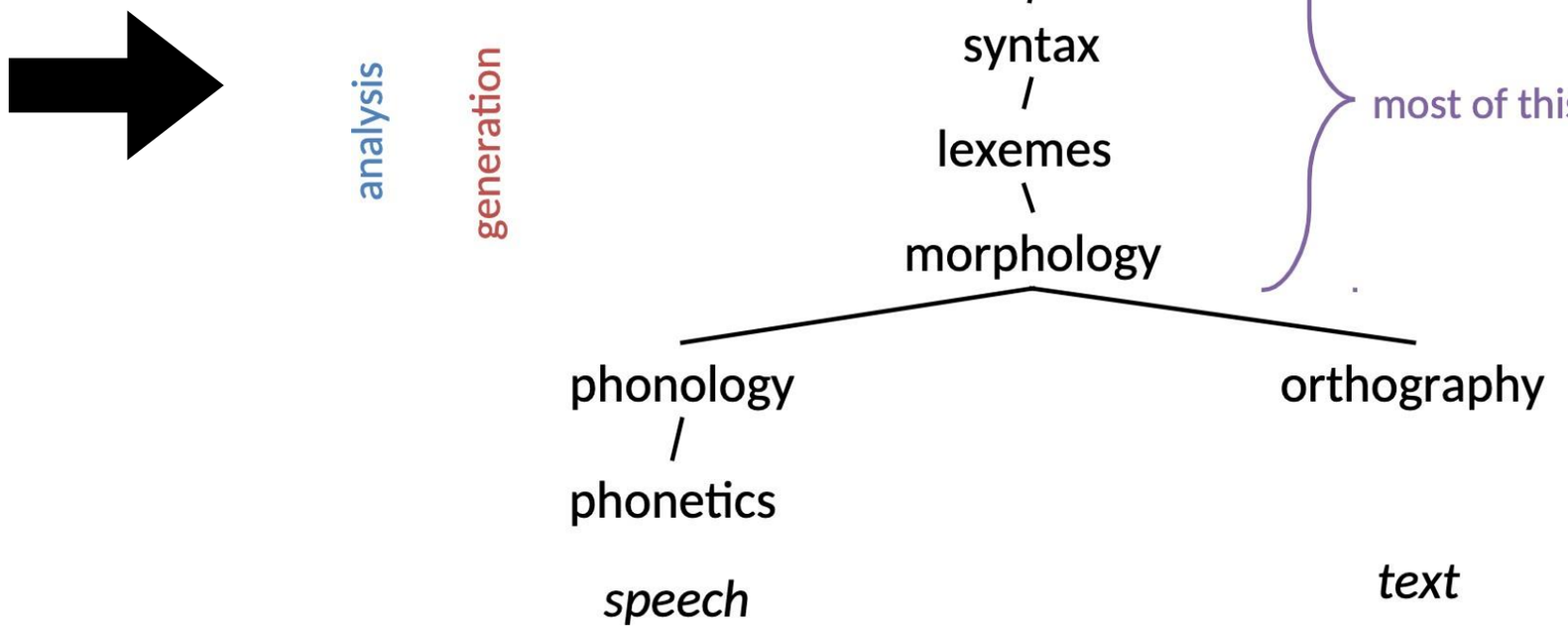
Université de Montréal

Pourquoi le NLP est difficile

Modèle de langage naturel

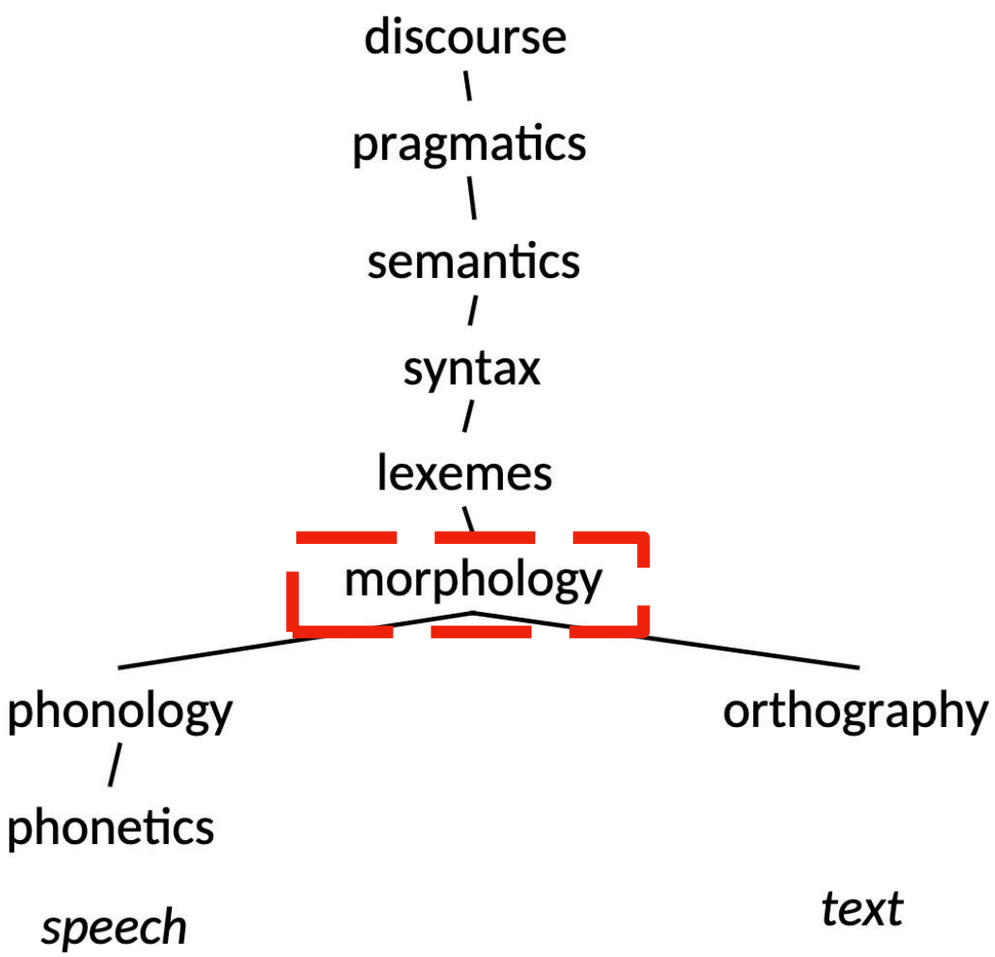
Pourquoi le NLP est difficile

Representation



- Les fonctions entre les niveaux sont extrêmement complexes.
- La pertinence d'une représentation dépend de la tâche.

Pourquoi le NLP est difficile



- **Morphologie :** représentation des mots en composants significatifs

- Spectre de complexité à travers les langues :

  Langues analytiques ou isolantes (p. ex., anglais, chinois)
  Langues synthétiques (p. ex., finnois, turc, hébreu)

TIFGOSH ET HAYELED BAGAN
"you will meet the boy in the park"

Puedes dármelo
"You can give it to me"

uygarlaştıramadıklarımızdanmışsınızcasına
"(behaving) as if you are among those whom we could not civilize"

Mots vides

- **Mots vides :** Les mots vides sont des mots qui ne contiennent pas de signification importante. Ce sont des mots très courants dans une langue (par exemple a, an, the en anglais. 的, 了en Chinois et え, も en japonais). Cela n'aide pas sur la plupart des problèmes tels que l'analyse sémantique, la classification, etc.

- Habituellement, ces mots sont filtrés du texte car ils renvoient une grande quantité d'informations **inutiles**.

- Chaque langue donnera sa propre liste de mots vides à utiliser. Par exemple, les mots couramment utilisés dans la langue anglaise sont as, the, be, are.

- Dans NLTK, vous pouvez utiliser des mots vides prédéfinis, ou vous pouvez utiliser d'autres mots définis par une autre partie tels que Stanford NLP et Rank NL.
  Stanford NLP :
  https://github.com/stanfordnlp/CoreNLP/blob/master/data/edu/stanford/nlp/patterns/surface/videmots.txt
  Rang NL : https://www.ranks.nl/stopwords
  Jieba : https://github.com/fxsjy/jieba/blob/master/extra_dict/stop_words.txt

Résolution de co-référence

- **La résolution de co-référence** est la tâche de trouver toutes les expressions qui font référence à la **même entité dans un texte**.
- C’est une étape importante pour de **nombreuses tâches de haut niveau** en NLP qui impliquent la compréhension du langage naturel, telles que la synthèse de documents, la réponse aux questions et l’extraction d’informations.

Christopher Robin is alive and well. **He** is the same person that you read about in the book, **Winnie the Pooh**. As a **boy**, **Chris** lived in a pretty home called **Cotchfield Farm**. When **Chris** was three years old, **his father** wrote a poem about **him**. The poem was printed in a magazine for others to read. **Mr. Robin** then wrote a book

Complexité du NLP

- Les représentations linguistiques sont des constructions théorisées -> **Nous ne pouvons pas les observer directement.**
- **Données :** L'entrée est susceptible d'être bruitée.

- **Ambiguïté:** Chaque chaîne de caractère peut avoir de nombreuses interprétations possibles à tous les niveaux. La résolution correcte de l'ambiguïté dépendra du sens voulu, qui est souvent inférable à partir du contexte.
    - Les gens sont bons pour la résolution d'ambiguïté linguistique, mais les ordinateurs ne sont pas si bons pour cela:
      Q : Comment représentons-nous des ensembles d'alternatives possibles?
      Q : Comment représentons-nous le contexte ?

- **Variabilité:** Il existe de nombreuses façons d'exprimer le même sens, et infiniment de significations à exprimer.

    - Beaucoup de mots/phrases. Chaque niveau interagit avec les autres. Il y a une grande diversité dans les langues humaines. Les langues expriment le même genre de sens de différentes manières. Certaines langues expriment certaines significations plus facilement/souvent
    -

défis du NLP

- Problèmes liés aux données :

  - Beaucoup de données : dans certains cas, nous traitons d'énormes quantités de données Besoin de trouver des modèles capables de traiter efficacement beaucoup de données

  - Manque de données: De nombreux problèmes dans NLP souffrent du manque de données:

    - Plates-formes non standard (traduction de code)

    - Annotation coûteuse (désambiguïsation du sens des mots, reconnaissance des entités nommées)

- Besoin d'utiliser des méthodes pour surmonter ce défi (apprentissage semi-supervisé, apprentissage multitâche...)

défis du NLP

- Défi de l'ambiguïté : **Polysémie :** un mot, plusieurs significations

Book

Verb: **Book** a flight

Noun: He says it's a very good **book**

Bank

The edge of a river: He was strolling near the river **bank**

A financial institution: He works at the **bank**

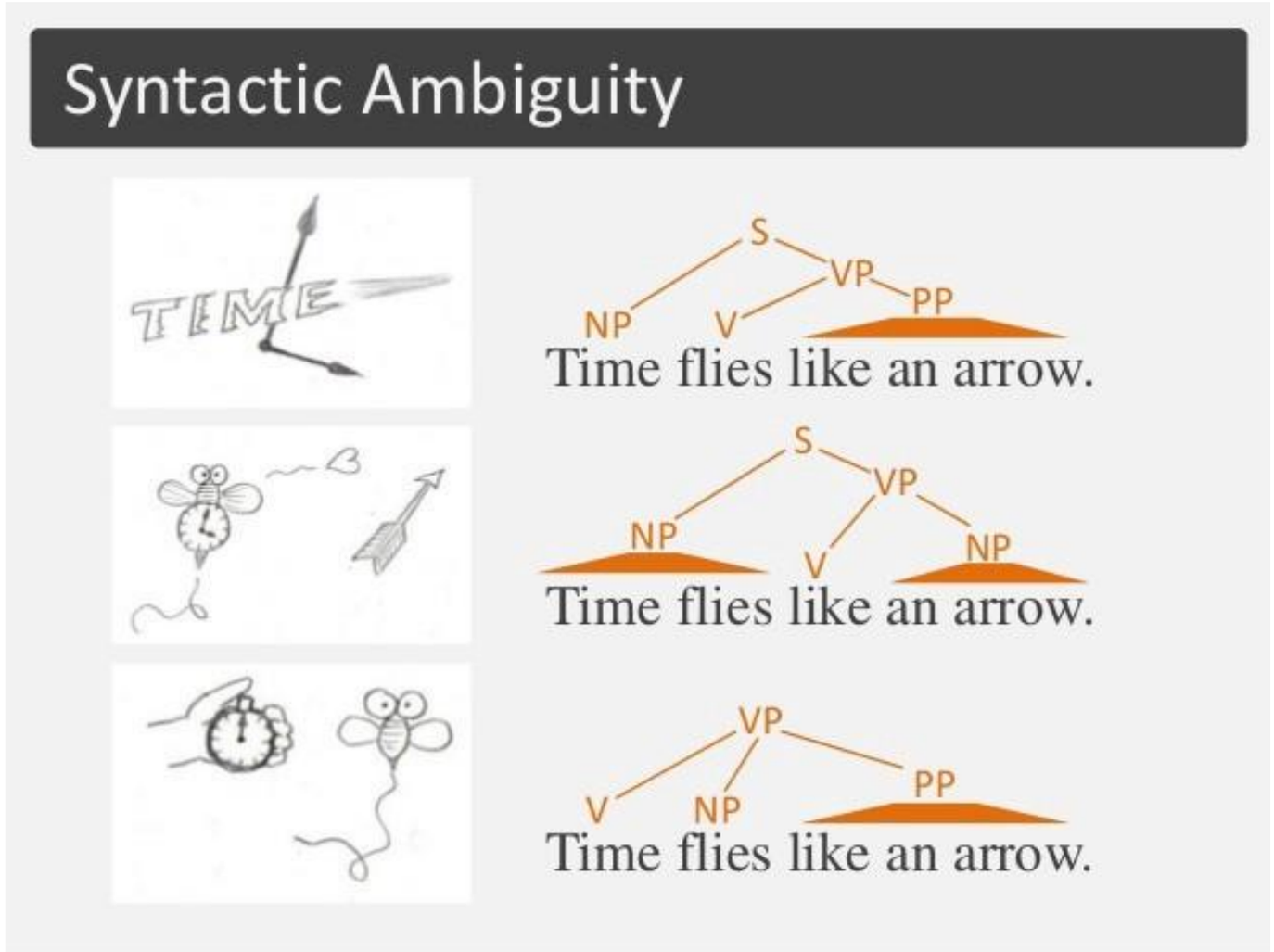
Solution

An answer to a problem: Work out the **solution** in your head

From Chemistry: Heat the **solution** to 75° Celsius

défis du NLP

- Défi de l'ambiguïté : **Ambiguïté syntaxique** : mêmes mots, plusieurs significations



défis du NLP

- **Variabilité :** mots différents, même sens

They allowed him to...

They let him ...

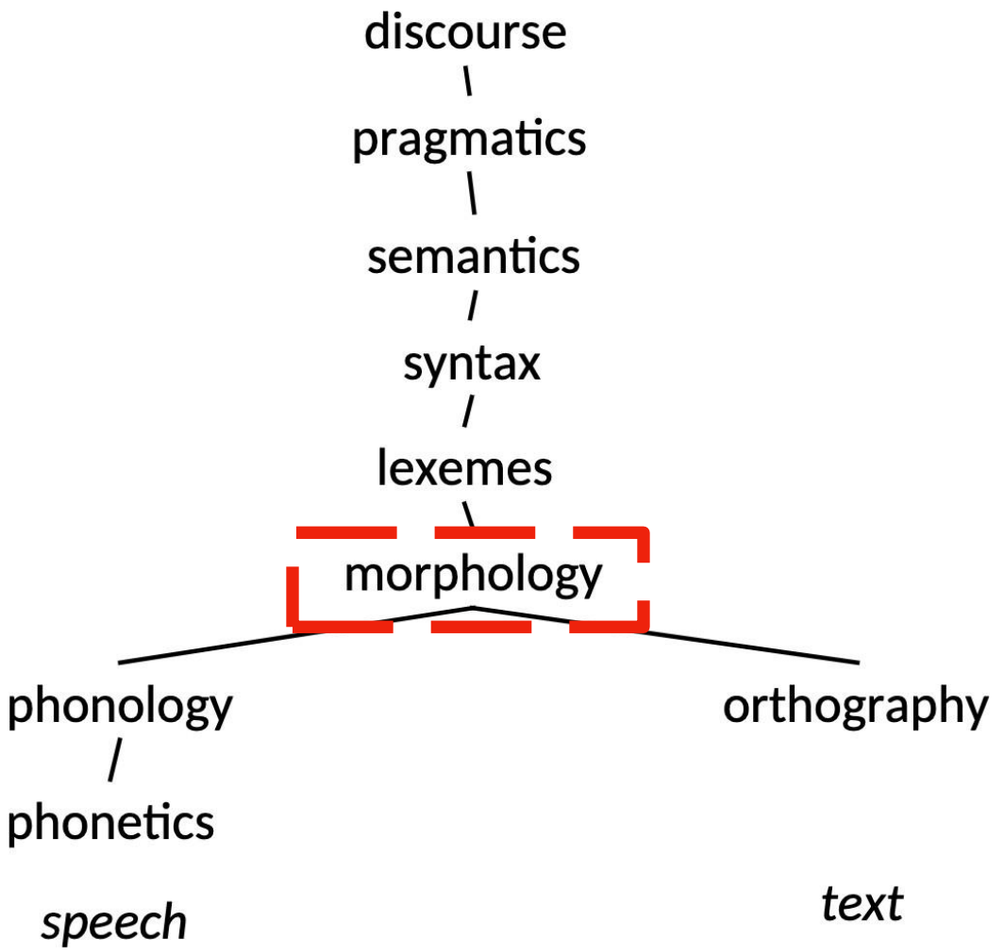
He was allowed to...

He was permitted to...

modèle de langage naturel

Pourquoi le NLP est difficile

Ingénierie des caractéristiques



- **Tokenisation:** séparation du texte en composantes significatives qui seront les caractéristiques d'entrée)
- (des mots mais pas toujours!)

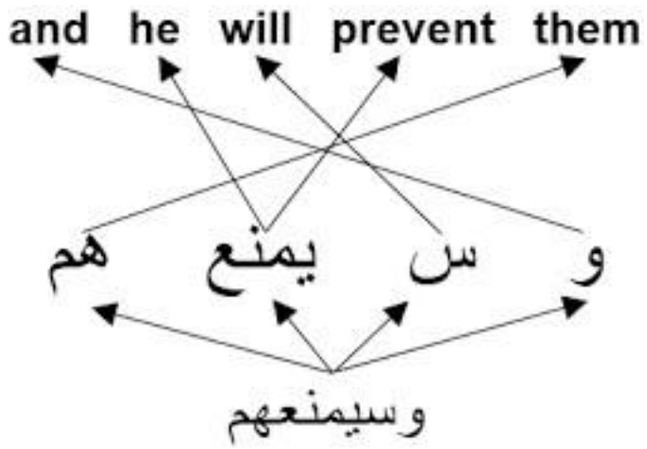
TIFGOSH ET HAYELED BAGAN
"you will meet the boy in the park"

Puedes dármelo
"You can give it to me"

uygarlaştıramadıklarımızdanmışsınızcasına
"(behaving) as if you are among those whom we could not civilize"

Tokenisation

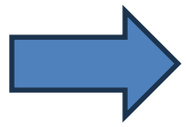
- La tâche la plus courante en NLP est la tokenisation.
- **La tokenisation** est le processus de décomposition d'un document en mots, signes de ponctuation, chiffres numériques, etc.



Tokenization

De la chaîne de caractères à un suite de chiffre (tokens). Exemple:

«La tokenisation est au cœur de bien des bizarreries des modèles de langage de grande taille (LLMs). Ne l'ignorez pas.»



[14772, 11241, 5612, 1556, 35851, 269, 129, 241, 333, 390, 275, 2013, 748, 13699, 1678, 748, 953, 14064, 829, 390, 42392, 496, 390, 4490, 68, 7894, 293, 357, 3069, 10128, 737, 3169, 300, 6, 46430, 89, 38836, 13, 198]

Extraction de caractéristiques (Encodage one-hot) pour le texte!

Beaucoup de choix possible pour cet encodage d'une chaîne de caractère vers une liste de nombre!

Il va être important de bien **compresser** l'information.

Étape clef! Une fois la tokenization fixée on ne peut plus la changer!

Mila

Université de Montréal

Compréhension de la langue

- Tout dépend de la probabilité du prochain mot/**token**!

$P(\text{obama}|\text{president of U.S.})$

$P(\text{Good morning}|\text{Buenos días})$

$P(\text{I saw a bus}) \gg$
$P(\text{eyes awe a boss})$

$P(\text{a man eating a sandwich})$

Modèles de langage probabilistes

- **Objectif :** Calculer la probabilité d'une phrase ou de séquences de tokens

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

- Tâche connexe : probabilité d'un token à venir :

$$P(w_5 \mid w_1, w_2, w_3, w_4)$$

- Un modèle qui calcule l'un des éléments ci-dessus est appelé **modèle de langage.**

Compréhension de la langue

A sentence $(x_1, x_2, \ldots, x_T)$

- Ex: (the, cat, is, eating, a, sandwich, on, a, couch)

How likely is this sentence?

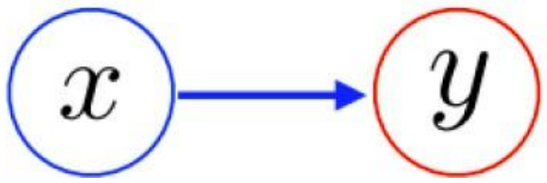
In other words, what is the probability of $(x_1, x_2, \ldots, x_T)$?

- i.e: $P(x_1, x_2, \ldots, x_T) = ?$

Rappel
(probabilité 101)

- Joint probability $p(x, y)$
- Conditional probability $p(x|y)$
- Marginal probability $p(x)$ and $p(y)$

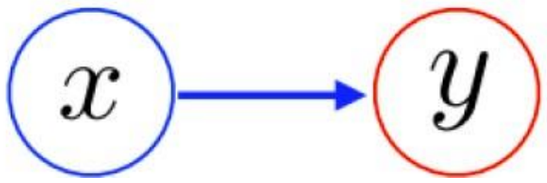
- They are related by $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$



Rappel
(probabilité 101)

- Joint probability $p(x, y)$ Probabilité jointe
- Conditional probability $p(x|y)$ Probabilité conditionnelle
- Marginal probability $p(x)$ and $p(y)$ Probabilité marginale

- They are related by $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$



Chain Rule: Formule des probabilités composées

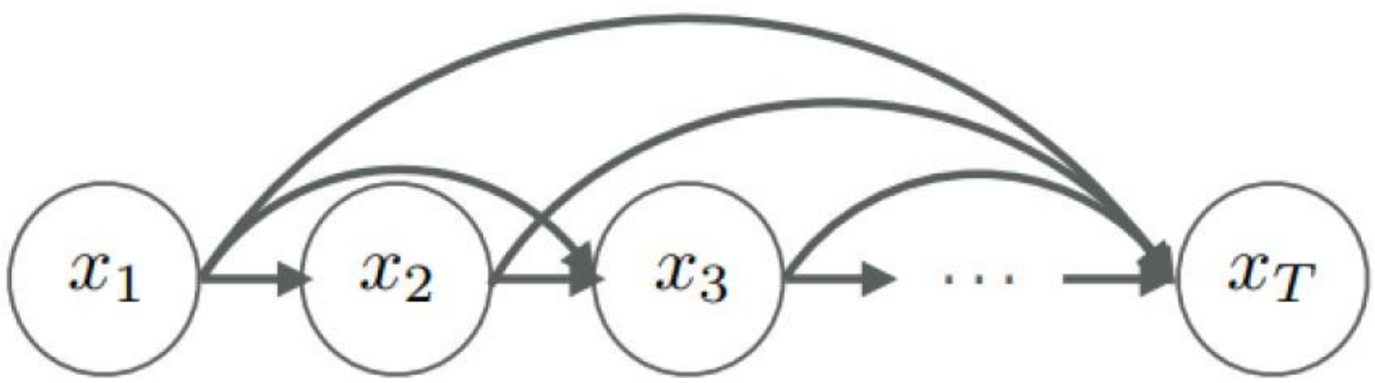
$$P(x_1,x_2,x_3,\ldots,x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2)\ldots P(x_n|x_1,\ldots,x_{n-1})$$

Modèles de langage probabilistes

- Rewrite $p(x_1, x_2, \ldots, x_T)$ into

$$p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

- Graphically,

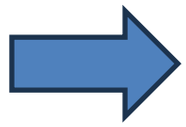


Perspective: Suite de tâche de classification (prédire le mot suivant en fonction des précédents)

Retour à la tokenisation

De la chaîne de caractères à un suite de chiffre (tokens). Exemple:

«La tokenisation est au cœur de bien des bizarreries des modèles de langage de grande taille (LLMs). Ne l'ignorez pas.»



[14772, 11241, 5612, 1556, 35851, 269, 129, 241, 333, 390, 275, 2013, 748, 13699, 1678, 748, 953, 14064, 829, 390, 42392, 496, 390, 4490, 68, 7894, 293, 357, 3069, 10128, 737, 3169, 300, 6, 46430, 89, 38836, 13, 198]

Extraction de caractéristiques (Encodage one-hot) pour le texte!

Beaucoup de choix possible pour cet encodage d’une chaîne de caractère vers une liste de nombre!

Il va être important de bien **compresser** l’information.

Étape clef! Une fois la tokenization fixée on ne peut plus la changer!

Mila

Université de Montréal

Tiktokenizer

gpt2

```
La tokenisation est au cœur de bien des bizarreries des
modèles de langage de grande taille (LLMs). Ne l'ignorez pas.

127 + 677 = 804
1275 + 6773 = 8041

Œuf.
J'ai un œuf.
œuf.
ŒUF.

Enchanté de vous rencontrer. Je suis ChatGPT, un grand modèle
linguistique développé par OpenAI. Si vous avez des questions,
n'hésitez pas à me les poser.

python
Copy code
for i in range(1, 101):
    if i % 3 == 0 and i % 5 == 0:
        print("FizzBuzz")
    elif i % 3 == 0:
        print("Fizz")
    elif i % 5 == 0:
        print("Buzz")
    else:
        print(i)
```

Token count
252

La tokenisation est au cœur de bien des bizarreries de
s modèles de langage de grande taille (LLMs). Ne l'ign
orez pas.

127 + 677 = 804
1275 + 6773 = 8041

Œuf.
J'ai un œuf.
œuf.
ŒUF.

Enchanté de vous rencontrer. Je suis ChatGPT, un grand
modèle linguistique développé par OpenAI. Si vous avez
des questions, n'hésitez pas à me les poser.

python
Copy code

```
for i in range(1, 101):
    if i % 3 == 0 and i % 5 == 0:
        print("FizzBuzz")
    elif i % 3 == 0:
        print("Fizz")
    elif i % 5 == 0:
        print("Buzz")
    else:
        print(i)
```

```
14772, 11241, 5612, 1556, 35851, 269, 129, 241, 333, 3
90, 275, 2013, 748, 13699, 1678, 748, 953, 14064, 829,
```

Étape 1: Unicode

Une idée simple est de convertir notre chaîne de caractère en Unicode.

```
[ord(x)for x in "안녕하세요 👋 (hello in Korean!)"]
```

```
[50504,
 45397,
 54616,
 49464,
 50836,
 32,
 128075,
 32,
 40,
 104,
 101,
 108,
 108,
 111,
 32,
 105,
 110,
 32,
 75,
 111,
 114,
 101,
 97,
 110,
 33,
 41]
```

Unicode

119 langues

Article Discussion Lire Modifier Modifier le code Voir l'historique Outils

Unicode est un standard informatique qui permet des échanges de textes dans différentes langues, à un niveau mondial. Il est développé par le Consortium Unicode, qui vise au codage de texte écrit en donnant à tout caractère de n'importe quel système d'écriture un nom et un identifiant numérique, et ce de manière unifiée, quelle que soit la plateforme informatique ou le logiciel utilisé.

Logo Unicode.[1]

Ce standard est lié à la norme ISO/CEI 10646 qui décrit une table de caractères équivalente. La dernière version, Unicode 15.1, a été publiée en septembre 2023[2].

Totalement compatible avec le jeu universel de caractères (JUC) de l'ISO/CEI 10646, le standard Unicode l'étend en lui ajoutant un modèle complet de représentation et de traitement de textes, en conférant à chaque caractère un jeu de propriétés (qui peuvent être soit pour certaines, standardisées et stabilisées dans toutes les versions d'Unicode où le caractère a été encodé, soit informatives avec seulement une recommandation sur leur usage, qui peut évoluer en fonction des nouveaux besoins trouvés). Ces propriétés décrivent avec précision les relations sémantiques qui peuvent exister entre plusieurs caractères successifs d'un texte, et permettent de standardiser ou recommander des algorithmes de traitement qui préservent au maximum la sémantique des textes transformés. Unicode a pour objet de rendre un même texte utilisable à l'identique sur des systèmes informatiques totalement différents.

Le standard Unicode est constitué d'un répertoire de 149 186 caractères, couvrant plus de 150 écritures, d'un ensemble de tableaux de codes pour référence visuelle, d'une méthode de codage et de plusieurs codages de caractères standard, d'une énumération des propriétés de caractère (lettres majuscules, minuscules, APL, symboles, ponctuation, etc.) d'un ensemble de fichiers de référence des données informatiques, et d'un certain nombre d'éléments liés, tels que des règles de normalisation, de décomposition, de tri, de rendu et d'ordre d'affichage bidirectionnel (pour l'affichage correct de texte contenant à la fois des caractères d'écritures de droite à gauche, comme l'arabe et l'hébreu, et de gauche à droite).

Problème: déjà ~150000 tokens…. Solution???

Mila

Université de Montréal

Étape 2: UTF-8

Idée: utiliser une chaîne à longueur variable pour encoder les caractères moins fréquents.

```
list("안녕하세요 👋 (hello in Korean!)".encode("utf-8"))
```

```
[236,
 149,
 136,
 235,
 133,
 149,
 237,
 149,
 152,
 236,
 132,
 184,
 236,
 154,
 148,
 32,
 240,
 159,
 145,
 139,
```

Définition du nombre d'octets utilisés dans le codage (attention

Caractères codés	Représentation binaire UTF-8
U+0000 à U+007F	0bbb·bbbb
U+0080 à U+07FF	110b·bbbb 10bb·bbbb
U+0800 à U+FFFF	1110·bbbb 10bb·bbbb 10bb·bbbb
U+10000 à U+10FFFF	1111·0bbb 10bb·bbbb 10bb·bbbb 10bb·bbbb
	1111·0100 1000·bbbb 10bb·bbbb 10bb·bbbb

Mila

Université de Montréal

Étape 3:

Compression en augmentant la taille du vocabulaire: *Byte pair encoding*

Étant donnée un jeu de données d'entraînement:

1121123311
Vocabulaire: 1,2,3

- On trouve la paire d'éléments la plus fréquente (ici, *11)*
- On crée un nouvel élément pour la remplacer (ici, *4)*

4242334
Vocabulaire: 1,2,3,4=11

- On recommence… (jusqu'à que l'on n'ai plus de paires fréquentes ou qu'on arrive à une taille trop grande pour le Vocabulaire)

Étape 3:

Compression en augmentant la taille du vocabulaire: *Byte pair encoding*

É

1

V

-

-

4

V

Élement important: la tokenisation est apprise sur un jeu de donnée d'entraînement. (la paire la plus fréquente dépend du jeu de données)

- On recommence… (jusqu'à que l'on n'ai plus de paires fréquentes ou qu'on arrive à une taille trop grande pour le Vocabulaire)

Mila

Université de Montréal

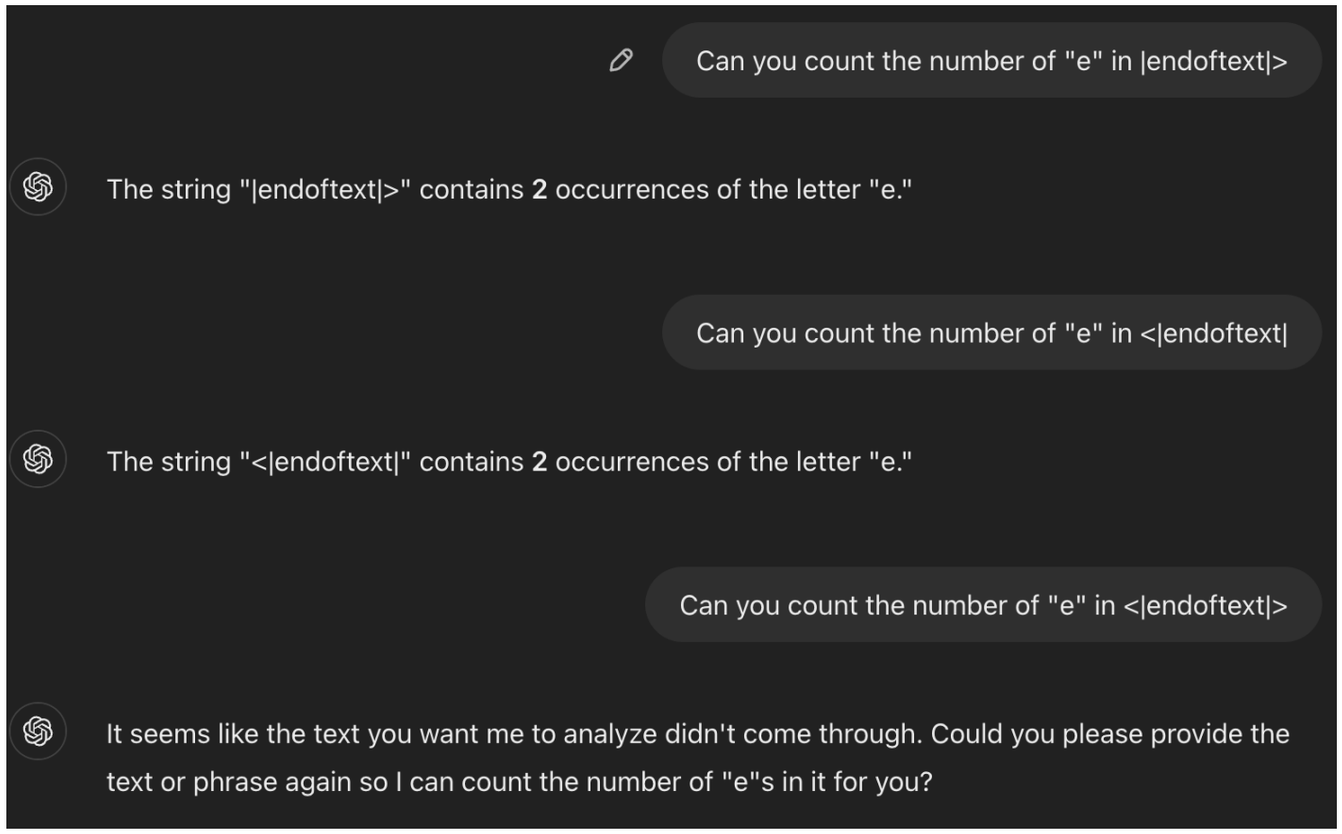
Étape 4:

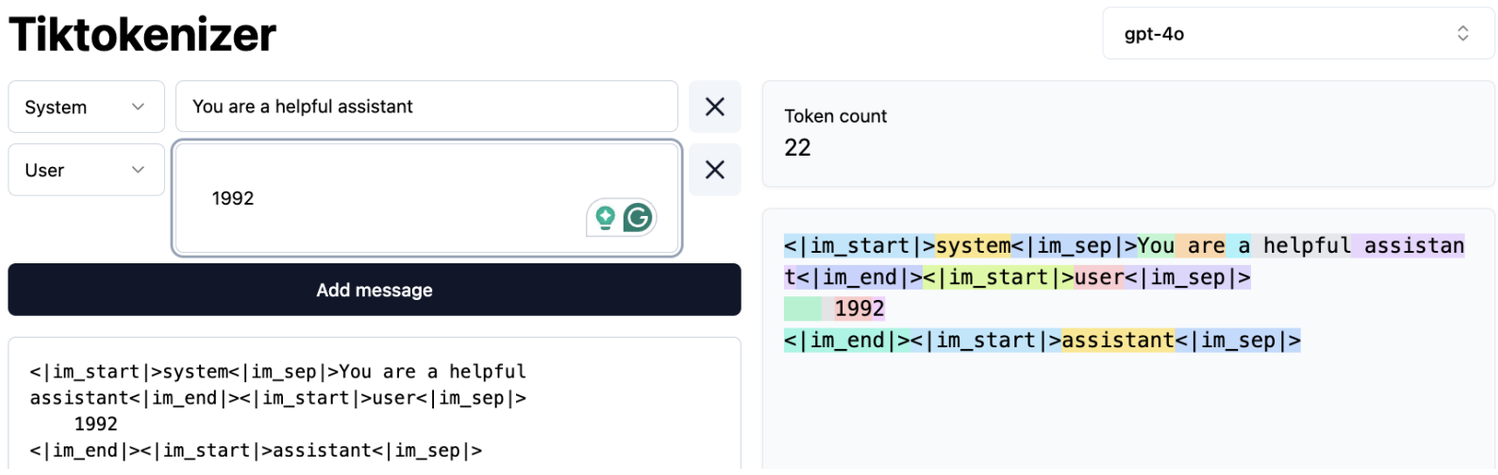
Beaucoup d’astuces (non-couvertes ici):

- Importance des token speciaux
  <|endoftext|>,

- Certaines fusions sont rendues impossibles
(par exemples entre mot et mots vides)

- Prétraitement des nombres et des espaces





Mila

Université de Montréal

Tiktokenizer

gpt2

```
La tokenisation est au cœur de bien des bizarreries des
modèles de langage de grande taille (LLMs). Ne l'ignorez pas.

127 + 677 = 804
1275 + 6773 = 8041

Œuf.
J'ai un œuf.
œuf.
ŒUF.

Enchanté de vous rencontrer. Je suis ChatGPT, un grand modèle
linguistique développé par OpenAI. Si vous avez des questions,
n'hésitez pas à me les poser.

python
Copy code
for i in range(1, 101):
    if i % 3 == 0 and i % 5 == 0:
        print("FizzBuzz")
    elif i % 3 == 0:
        print("Fizz")
    elif i % 5 == 0:
        print("Buzz")
    else:
        print(i)
```

Token count

252

La tokenisation est au cœur de bien des bizarreries de
s modèles de langage de grande taille (LLMs). Ne l'ign
orez pas.

127 + 677 = 804
1275 + 6773 = 8041

Œuf.
J'ai un œuf.
œuf.
ŒUF.

Enchanté de vous rencontrer. Je suis ChatGPT, un grand
modèle linguistique développé par OpenAI. Si vous avez
des questions, n'hésitez pas à me les poser.

python
Copy code

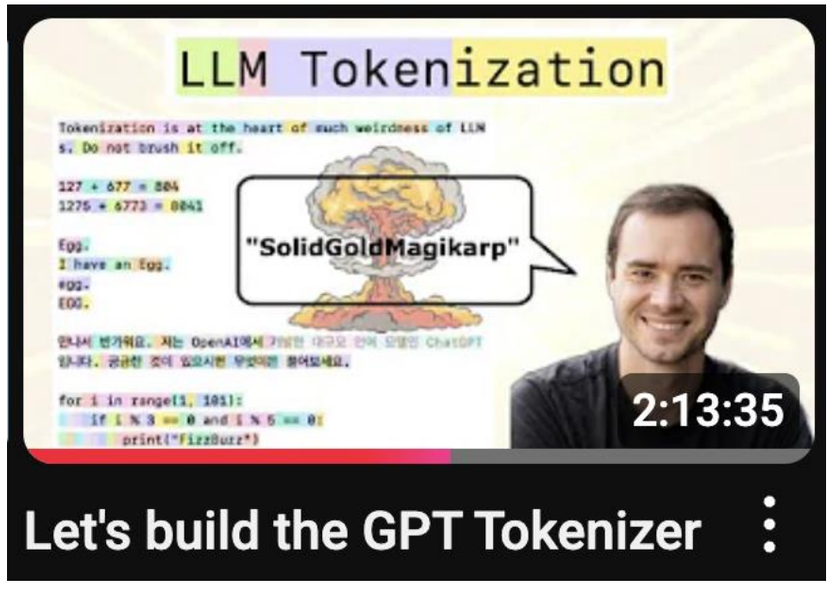
```
for i in range(1, 101):
    if i % 3 == 0 and i % 5 == 0:
        print("FizzBuzz")
    elif i % 3 == 0:
        print("Fizz")
    elif i % 5 == 0:
        print("Buzz")
    else:
        print(i)
```

```
14772, 11241, 5612, 1556, 35851, 269, 129, 241, 333, 3
90, 275, 2013, 748, 13699, 1678, 748, 953, 14064, 829,
```

Plus de détails sur la tokenization

Excellente vidéo d’Andrej Karpathy (une partie de ce cours est inspiré de la vidéo ci-dessous).

Avec des exercices et plus de détails!



Conclusion

<u>Prétraitement:</u>

Étape importante de NLP!

- Tokenisation fondamentale pour les grands modèles de langage

    - Par exemple, GPT utilise le codage de paire d'octets (BPE) (Sennrich et al., 2015)

- La suppression des mots vides peut être importante pour réduire le nombre de jetons et la taille des phrases.

- La compréhension du contexte et les dépendances à long terme sont deux grands défis de NLP.