

Deep Learning par la pratique

Jour 3, après midi : texte

NLP

Le deep learning pour le texte adapté dans 2 cas

- énormément de texte (au minimum wikipedia sinon plus)
- Tâche complexe : traduction, résumé, transfer de style, chatbot, génération etc

Pour tout le reste :

- classification, analyse de sentiment (c'est de la classification), topic modeling, NER etc
- NLTK + Spacy suffit la plupart du temps

Le Traitement du langage naturel - NLP

le NLP couvre un très vaste champs d'applications

- Part of speech tagging (POS), analyse sémantique,
- classification : sentiment, sujets, toxicité, ...
- NER : reconnaissance d'entités, information extraction

Puis de façon plus complexe

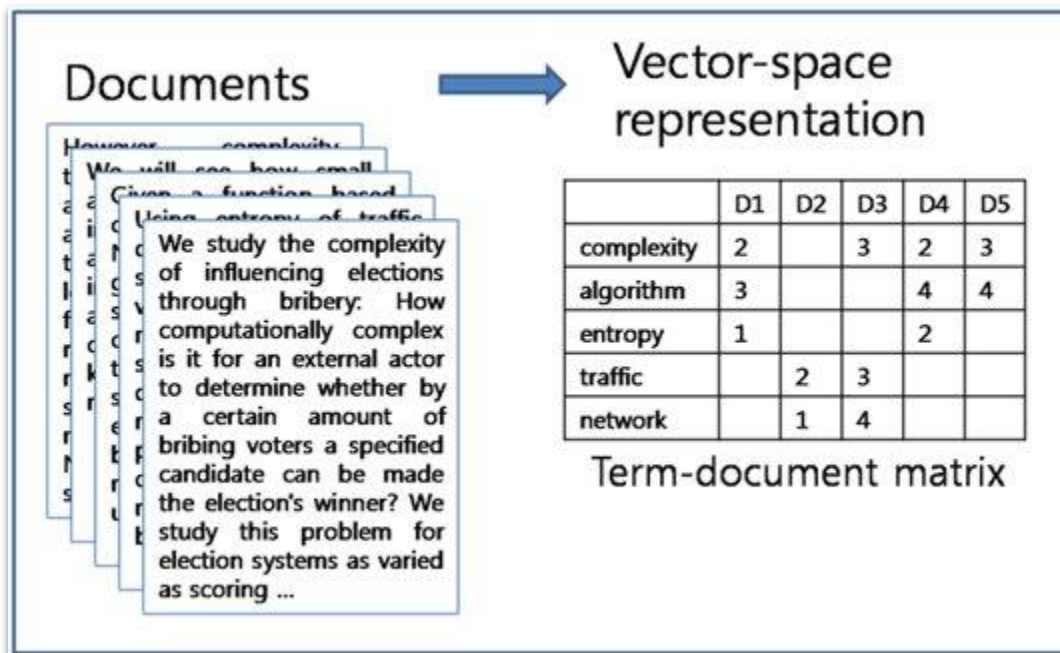
- traduction, résumé, compréhension, génération, chatbots

et le texte comme origine ou cible

- transformations : texte - image - speech - audio - video - SQL
- génération :
 - text to image & image to text
 - text to audio & audio to text
 - text to video & video to text

du décompte des mots aux LLMs

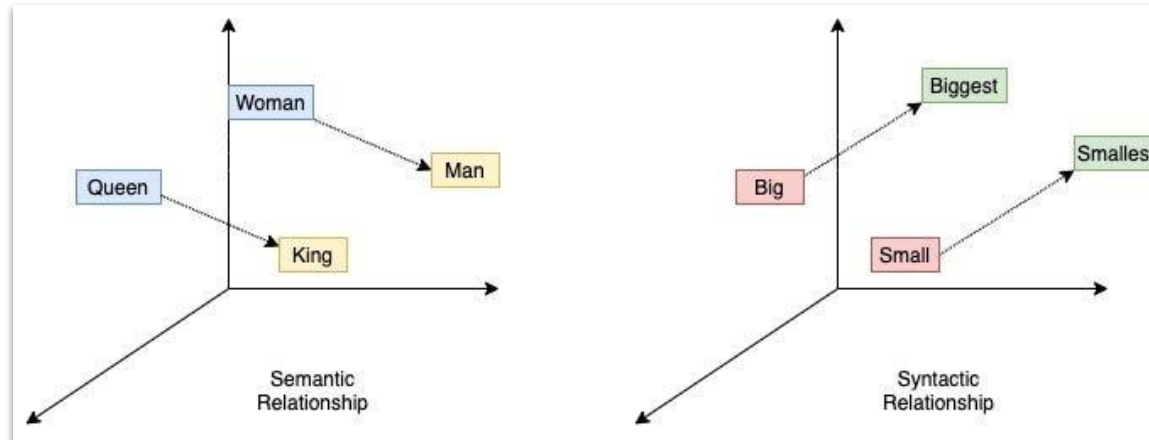
Dans l'ancien temps, on se basait sur la fréquence des mots dans un corpus, l'approche tf-idf



du décompte des mots aux LLMs

2013 embeddings - word2vec

- on entraîne un RNN sur un large corpus pour prédire le mot manquant dans une phrase
- on récupère les poids de la dernière couche comme représentation de chaque mot. Ce vecteur est appelé **embedding**.
- La représentation vectorielle capture la signification du mot



Principales étapes

- **word2vec** : 2013 capture les relations sémantiques. suivi de GloVe, FastText
 - **Seq2Seq** : 2014 encoder-decoder architecture with RNNs
 - **Attention** : 2015 Allows the model to focus on different parts of the input sequence for each output step.
-

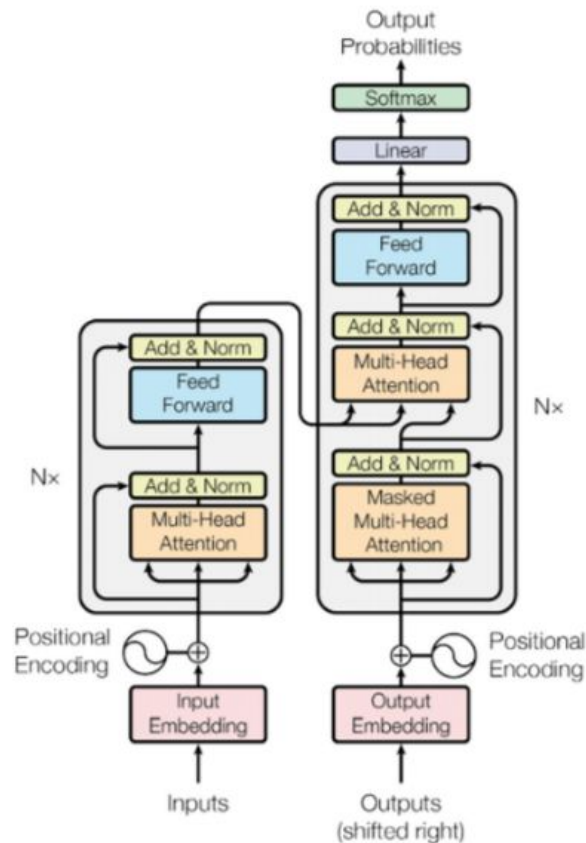
- **Transformers** : 2017 attention mechanisms sans CNN ou RNN
- **BERT** : 2018 evolution de l'architecture Transformers
- **GPT 1, 2, 3, 4**: de 2018 à 2023 : transformers; modèles de plus en plus grand avec des datasets d'entraînement de plus en plus gigantesques
- **2023 - 2024 Les LLMs** : GPT4o, Claude Opus, Mistral, Gemini, LLama,

Du décompte des mots aux LLMs - Ce qui compte c'est l'attention

2017 - Attention Is All You Need - Transformers

<https://arxiv.org/abs/1706.03762>

- mélange de CNN et de RNN
- focus sur certain mots : l'attention
 - multi-head attention : focus sur plusieurs mots à la fois
 - self-attention : lien entre les mots



Ressources sur les transformers

- simple =>

<https://galaxyinferno.com/3-lessons-from-the-paper-attention-is-all-you-need-as-a-beginner/>

- The Illustrated Transformer : <https://jalammar.github.io/illustrated-transformer/>
- Analyse de l'architecture : <https://kikaben.com/transformers-encoder-decoder/>
- <https://neptune.ai/blog/bert-and-the-transformer-architecture>
- <https://towardsdatascience.com/attention-is-all-you-need-discovering-the-transformer-paper-73e5ff5e0634>
- transformers 2022.02 Lucas Beyers <https://www.youtube.com/watch?v=UpfcyzoZ644>
- Attention is all you need <https://arxiv.org/abs/1706.03762>

Préparer les données textuelles

Préparer les données

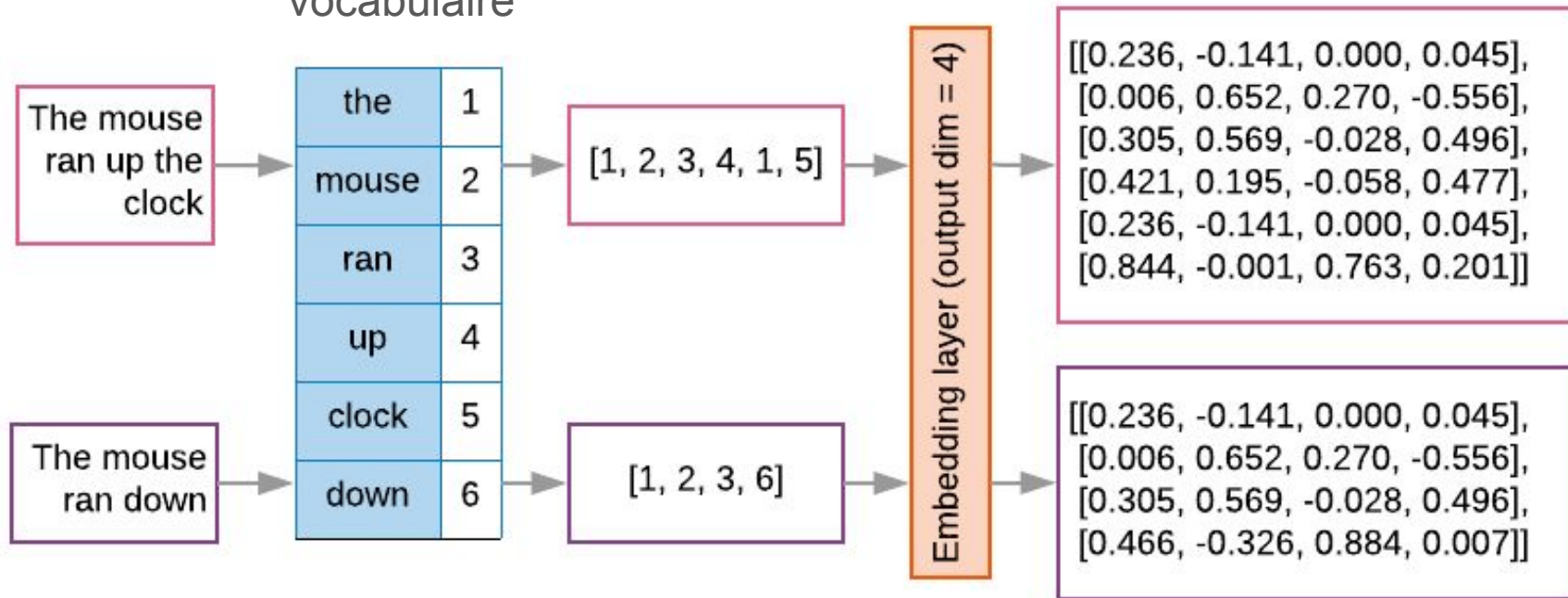
Comment passer d'un texte de dimension variable à une représentation numérique que l'on peut utiliser pour entraîner un modèle ?

1. **Tokenization** : processus de découpage d'un texte en unités élémentaires appelées "tokens".

Un token peut être un mot, un signe de ponctuation ou un autre élément atomique.

2. Sequencage (prochain slide)
3. **Embedding** : remplacer chaque token par son équivalent vectoriel

vocabulaire



tokenisation des phrases

représentation vectorielle

Tokenization => split(' ') : 90% du boulot

```
"bonjour tout le monde".split(" ")
```

Tokenizer

NLTK

```
1 from nltk import ngrams
2 from nltk.tokenize import WordPunctTokenizer
3
4 text = "How much wood would a woodchuck chuck if a woodchuck could chuck wood?"
5
6 # Tokenize
7 tokens = WordPunctTokenizer().tokenize(text)
```

Spacy.io

Pour du texte, avant de se lancer dans les réseaux de neurones : Spacy

```
Edit the code & try spaCyspaCy v3.7 · Python 3 · via Binder

# pip install -U spacy
# python -m spacy download en_core_web_sm
import spacy

# Load English tokenizer, tagger, parser and NER
nlp = spacy.load("en_core_web_sm")

# Process whole documents
text = ("When Sebastian Thrun started working on self-driving cars at "
        "Google in 2007, few people outside of the company took him "
        "seriously. "I can tell you very senior CEOs of major American "
        "car companies would shake my hand and turn away because I wasn't "
        "worth talking to," said Thrun, in an interview with Recode earlier "
        "this week.")
doc = nlp(text)

# Analyze syntax
print("Noun phrases:", [chunk.text for chunk in doc.noun_chunks])
print("Verbs:", [token.lemma_ for token in doc if token.pos_ == "VERB"])

# Find named entities, phrases and concepts
for entity in doc.ents:
    print(entity.text, entity.label_)
```

RUN

<https://spacy.io/>

Annoter un corpus

- manuel : <https://prodi.gy/>
- LLM : few shots learning

Preparer les données

Sequencage : découper le texte en séquence de même longueurs avec du padding

on ajoute un token de marquage

"Dynamic Padding"

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	_Eh	_bien	_c	'	_est	_un	_bon	_indicateu	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
2	_Ouais	_je	_suis	_un	_coureur	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
3	_Ils	_ne	_sont	_pas	important	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
4	_Il	_y	_a	_de	nombreux	condition	_qui	_ne	_sont	_pas	_visibles	_	.	
5	_Chaque	_zone	_de	_l	'	_île	_offre	_quelque	_chose	_de	_différent	_	.	
6	_Mais	_tu	_peux	_vivre	_avec	_eux	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
7	_Un	_grand	_homme	_	,	_dit	-	il	_	.	[PAD]	[PAD]	[PAD]	
8	_Elle	_a	_été	_menée	_en	_silence	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	
9	_Tu	er	beaucou	_de	_fourmis	_de	_feu	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
10	_La	_question	_est	_de	_savoir	_si	_clin	ton	_a	_le	_cul	ot	_	.
11	_C	'	_est	_vrai	_	.	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]
12	_Dans	_ce	_domaine	_	,	_seuls	_les	_sa	ther	i	_le	_savent	_	.

Batch Length:

13

Batch Length:

13

Batch Length:

14

Total Tokens: 160

Text classification

Implémentons les tutoriaux suivants

- https://keras.io/examples/nlp/text_classification_from_scratch/
- https://www.geeksforgeeks.org/sentiment-analysis-with-an-recurrent-neural-networks-rnn/?ref=ml_lbp
- <https://www.geeksforgeeks.org/sentiment-classification-using-bert/?ref=lbp>

Ecrire comme Molière

On va s'inspirer du notebook d'écriture de Shakespeare pour écrire comme Molière

dataset :

<https://www.kaggle.com/datasets/guillaumegrosjean/moliere-plays-dataset>

- sur la base des textes; enlever les noms des personnages

with the invention of the transformer architecture are LSTM still relevant ?

<https://chatgpt.com/c/e53f96b6-388f-403c-90f3-bd6bb2890b1e>
<https://claude.ai/chat/d52fa833-e313-441a-ba55-72bd3a76e223>