Natural Language Processing

Paul IROLLA

Skills4All
digital-learning certifiant

Votre formateur

- **Paul Irolla**
  - **Docteur en Intelligence Artificielle (Paris Saclay)**
  - **5 Publications scientifiques à comité de lecture**
  - **6 conférences internationales de hacking & sécurité informatique**
  - **Data Scientist**

Skills4
digital-learning certifiant

Partie 1 : Introduction au NLP

Skills4All
digital-learning certifiant

Sommaire

- Introduction au NLP
  - Qu'est ce que le NLP ?
  - Quel est l'impact du NLP ?
  - La langue : différents niveaux d'analyse
  - Les challenges des modèles de NLP
  - Ce que vous apprendrez dans ce cours

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Qu'est ce que le NLP ?

- Un des domaines les plus en pointe du ML
- Vous l'utilisez actuellement sans le savoir
  - Gmail, Google Search
  - Recommandation de produits dans Amazon
  - Traduction de texte en ligne
- La masse de données textuelle permet l'avènement du NLP
  - 10 000 Tweets / seconde
  - 100 000 Recherches Google / seconde
  - 3 000 000 mails envoyés / seconde

Zone d'incrustation vidéo

Skills4 digital-learning certifiant

Quel est l'impact (actuel, potentiel) du NLP ?

- **Notre monde social est textuel**
  - **4.7 milliards d'utilisateurs actifs sur les réseaux sociaux**
  - **6h en moyenne d'utilisation d'internet / jour**
- **L'automatisation et l'optimisation sont une logique interne du capitalisme. Les interactions textuelles sont en train et seront automatisées.**

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

La langue : différents niveaux d'analyse

- Phonologique
- Graphémique
- Morphologique
- Syntaxique
- Sémantique
- Contexte linguistique
- Contexte extra-linguistique

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Les challenges des modèles de NLP

- La langue est un objet difficile à modéliser car :
  - Elle est ambiguë : chaque phrase et terme peuvent avoir plusieurs significations
  - Elle est variable : variabilité de mots, d'expressions, fautes d'orthographes ...
  - Elle est productive : néologismes, création de nouveaux mots, importation de mots étrangers
  - Elle est épars : la plupart des mots sont rares

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Ce que vous apprendrez dans ce cours

- Manipuler du texte avec Python
- Transformer des symboles en vecteurs
- Utiliser le Deep Learning pour le NLP
- Vous réaliserez les projets suivants :
  - Réaliser un détecteur de spams
  - Réaliser un modèle capable de répondre à des questions sur un texte
  - Réaliser un modèle capable de déduire d'un texte une émotion particulière (positif / négatif)
  - Réaliser un bot de conversation
  - Réaliser un modèle capable de résumer un texte

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Partie 2 : Manipuler les textes et les symboles

Skills4All
digital-learning certifiant

Sommaire

- Manipuler les textes et les symboles
  - Segmentation des symboles (tokenization)
  - Détection de récurrences (patterns) et nettoyage de textes
  - Étiqueter des symboles (token tagging)

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Partie 2.1 : Segmentation des symboles

Skills4All
digital-learning certifiant

2.1 - Segmentation des symboles (tokenization)

Définitions

- Un **symbole (token)** est la plus petite unité de texte
- Un **document** est une séquence de symboles de taille arbitraire
- Un **corpus** est un ensemble de documents
- Un **vocabulaire** est l'ensemble de symboles d'un corpus

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

2.1 - Segmentation des symboles (tokenization)

- Un symbole peut être :
  - Un caractère
  - Une chaîne de taille fixe de caractères (bigrams, ngrams)
  - Un mot
  - Un sous-mot (préfixe, suffixe, forme canonique)
  - Une chaîne de mots

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

2.1 - Segmentation des symboles (tokenization)

- Trois stratégies se distinguent pour segmenter et traiter les textes :
  - Les approches statistiques : fréquence d'apparition de symbole, ngrams, co-occurences
  - Les approches linguistiques : forme canonique des mots, POS (Part-Of-Speech)
  - Les approches d'apprentissage non-supervisées : un modèle d'apprentissage non-supervisé apprends à projeter les symboles dans un espace vectoriel
  - Une combinaison des trois

Zone d'incrustation vidéo

Skills4

2.1 - Segmentation des symboles (tokenization)

Quelques notions avant d'aborder une première partie pratique :

- Le but de la segmentation est de construire un vocabulaire.
- La taille du vocabulaire peut être un problème, plus elle est grande, plus les coûts en ressources sont grands.
- Le vocabulaire pourra être muni de symboles spéciaux pour pallier à ce problème : OOV, NUM, EMAIL, URL etc.
- Le vocabulaire pourra être réduit à sa forme canonique (lemmatisation) ou à sa forme radicale (stemming), en solution supplémentaire

Zone d'incrustation vidéo

Skills4 great learning centre

2.1 – Travail pratique

2 - Manipuler les textes et les symboles

2.1 - Segmentation des symboles (tokenization)

Nous allons ici explorer les différentes manières de segmenter les textes avec les librairies les plus communes de NLP. Dans un premier temps il faut installer Python 3 et la librairie NLTK avec la commande suivante :

sudo apt install python3 python3-dev python3-pip ipython3 python-is-python3 build-essential

Le paquet python-is-python3 sert à créer des alias pour les commandes python3 et pip3 en enlevant le numéro de version pour plus de simplicité. Ensuite, installez les paquet nltk et spacy :

pip install nltk spacy

Le paquet nltk a des données optionnelles téléchargables, comme des modèles pré-entrainés, des datasets ainsi que des dictionnaires de mots ou de symboles. Par simplicité et pour éviter d'être bloqué sur certaines parties, nous allons tout télécharger. Pour cela, lancez le terminal Python dans un terminal :

ipython

Cela va lancer le terminal d'interprétation de code Python dans lequel vous taperez :

import nltk

nltk.download()

Une interface graphique va se lancer dans laquelle il vous suffira de télécharger le méta-paquet "all". Si vous utilisez un Colab (ce que je vous encourage à ne pas faire et à utiliser votre propre système Linux), il vaut mieux télécharger un à un les paquets :

nltk.download('stopwords')

nltk.download('wordnet')

nltk.download('punkt')

Concernant Spacy, nous aurons besoin de quatre modèles qui sont eux aussi téléchargeable séparément :

python -m spacy download fr_core_news_sm

python -m spacy download fr_core_news_lg

python -m spacy download en_core_web_sm

python -m spacy download en_core_web_lg

Les versions sm sont des petits modèles rapides et les versions lg sont des plus gros modèles plus lents mais plus performants.

Note : toute librairie istallée pendant l'utilisation de ce notebook ne pourraient être disponible tout de suite. Pour que Jupyter les prenne en compte, il convient de relancer le notebook.

Zone d'incrustation vidéo

2.1 – Travail pratique

Test des différentes manières de segmenter un text

Nous allons travailler sur le dataset de review d'Amazon. Comme il nous aurons besoins d'autres librairies pour travailler sur les datasets et les manipuler, vous pouvez les installer maintenant si ce n'est pas déjà fait :

pip install pandas datasets

Charger le dataset

```
from datasets import load_dataset

df = load_dataset('amazon_reviews_multi', split = 'train').to_pandas()
```

Exploration du dataset

```
df.head()
```

	review_id	product_id	reviewer_id	stars	review_body	review_title	language	product_category
0	de_0203609	product_de_0865382	reviewer_de_0267719	1	Armband ist leider nach 1 Jahr kaputt gegangen	Leider nach 1 Jahr kaputt	de	sports
1	de_0559494	product_de_0678997	reviewer_de_0783625	1	In der Lieferung war nur Ein Akku!	EINS statt ZWEI Akkus!!!	de	home_improvement
2	de_0238777	product_de_0372235	reviewer_de_0911426	1	Ein Stern, weil gar keine geht nicht. Es hande...	Achtung Abzocke	de	drugstore
3	de_0477884	product_de_0719501	reviewer_de_0836478	1	Dachte, das wären einfach etwas festere Binden...	Zu viel des Guten	de	drugstore
4	de_0270868	product_de_0022613	reviewer_de_0736276	1	Meine Kinder haben kaum damit gespielt und nac...	Qualität sehr schlecht	de	toy

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

2.1 – Travail pratique

Sélectionner un commentaire français de plus de 1000 caractères

```
df_fr  = df[df.language == 'fr']
review = None
for _, rev in df_fr['review_body'].items():
    if len(rev) > 1000:
        review = rev
        break

print(review)
```

```
Acheté fin juillet, mis en service il y a environ 4 semaine
nes, la deuxième ampoule est morte ce week-end, ce qui lui
vendus par ""qui que ce soit"", il change constamment, les
es, je reçois en général une utilisation estivale complète
que 2 mois, cela ne fonctionne pas pour moi plus, va devoir
à. Quand ils fonctionnent, ils fonctionnent bien, mais si l
ue les tripes vont survivre grâce à une autre paire d'ampou
à de l'idée d'en acheter une autre .. BTW, environnement de
ur tuer les mouches qui le rendent dans le bureau .. c'est
contrôlée pour le matériel informatique afin qu'il ne soit
météorologiques, ou même beaucoup d'une charge ... ordure!
```

digital-learning certifiant

Zone d'incrustation vidéo

2.1 – Travail pratique

Décomposer le commentaire avec différents tokenizers

```
from nltk.tokenize import (
    TreebankWordTokenizer,
    ToktokTokenizer,
    TweetTokenizer,
    WhitespaceTokenizer
)
from nltk.tokenize import (
    word_tokenize,
    wordpunct_tokenize,
    sent_tokenize
)
```

```
tokens = word_tokenize(review)
print(tokens)
print(len(tokens))
```

```
['Acheté', 'fin', 'juillet', ',', 'mis', 'en', '
'première', 'ampoule', 'est', 'morte', 'il', 'y'
e', 'est', 'morte', 'ce', 'week-end', ',', 'ce',
'un', 'mois', '.', 'Ceux-ci', 'sont', 'vendus',
l', 'change', 'constamment', ',', 'les', 'noms',
us', 'identiques', ',', 'je', 'reçois', 'en', 'g
n", "d'entre", 'eux', 'et', 'les', 'présente', '
'ne', 'fonctionne', 'pas', 'pour', 'moi', 'plus'
i', '.', 'La', 'valeur', "n'est", 'pas', 'là', '
ien', ',', 'mais', 'si', 'les', 'ampoules', 'sor
'que', 'les', 'tripes', 'vont', 'survivre', 'grâ
',', 'ils', 'ont', 'mis', 'au', 'point', 'cette'
utre', '..', 'BTW', ',', 'environnement', 'de',
',', 'juste', 'pour', 'tuer', 'les', 'mouches',
ssi', 'sur', 'une', 'distribution', 'de', "l'ali
'informatique', 'afin', "qu'il", 'ne', 'soit', '
x', 'conditions', 'météorologiques', ',', 'ou',
233
```

Zone d'incrustation vidéo

2.1 – Travail pratique


```
tokens = wordpunct_tokenize(review)
print(tokens)
print(len(tokens))
```

```
tokens = sent_tokenize(review)
print(tokens)
print(len(tokens))
```

```
tokenizer = TreebankWordTokenizer()
tokens    = tokenizer.tokenize(review)
print(tokens)
print(len(tokens))
```

```
tokenizer = ToktokTokenizer()
tokens    = tokenizer.tokenize(review)
print(tokens)
print(len(tokens))
```

```
tokenizer = TweetTokenizer()
tokens    = tokenizer.tokenize(review)
print(tokens)
print(len(tokens))
```

Zone d'incrustation vidéo

2.1 – Travail pratique

Construire un vocabulaire

```
vocab     = set()
tokenizer = TweetTokenizer()

for _, rev in df_fr['review_body'].items():
    vocab.update(tokenizer.tokenize(rev))

print(len(vocab))
```

```
80904
```

Enlever les stopwords

```
from nltk.corpus import stopwords

vocab = vocab - set(stopwords.words('french'))
vocab = vocab - set(stopwords.words('english'))
print(len(vocab))
```

```
80698
```

digital-learning certifiant

Zone d'incrustation vidéo

2.1 – Travail pratique

Stemming and Lemmatizations

```
from nltk.stem import PorterStemmer

stemmer    = PorterStemmer()
stem_vocab = [stemmer.stem(word) for word in vocab]
print(stem_vocab[:50])

stem_vocab = set(stem_vocab)
print(len(stem_vocab))
```

```
['1,63', 'an', 'ticket', "m'endormir", 'habil', 'modu
'ocean', 'teckel', 'malaxé', 'contempl', 'mesur', 'hô
'tro', "l''horlog", 'pousseur', 'axé', 'astral', 'lyo
'destiné', "d'estomac", '1,20', 'retrogam', "d'encras
'decoup', 'absorb', 'froid', "l'identité", 'attach']
46850
```

```
from nltk.stem import SnowballStemmer

stemmer = SnowballStemmer("french")
stem_vocab = [stemmer.stem(word) for word in vocab]
print(stem_vocab[:50])

stem_vocab = set(stem_vocab)
print(len(stem_vocab))
```

```
['1,63', 'an', 'ticket', "m'endorm", 'habill', 'modul
'teckel', 'malax', 'contempl', 'mesur', 'hôtel', 'a.a
g", 'pousseur', 'axé', 'astral', 'lyon', "l'enfonc",
c", '1,20', 'retrogaming', "d'encr", 'x200', 'rhodiol
roid', "l'ident", 'attach']
37902
```

Zone d'incrustation vidéo

2.1 – Travail pratique


```
from nltk.stem import WordNetLemmatizer

lemmatizer       = WordNetLemmatizer()
lemmatizer_vocab = [lemmatizer.lemmatize(word) for word in vocab]
print(lemmatizer_vocab[:50])

lemmatizer_vocab = set(lemmatizer_vocab)
print(len(lemmatizer_vocab))
```

```
['1,63', 'AN', 'ticket', "m'endormir", 'habille', 'modulables', 'b
tilisait", 'Ocean', 'teckel', 'malaxée', 'contemplative', 'mesurab
e', 'concis', '40cm', 'tro', "l''horloge", 'pousseur', 'axés', 'as
'tromperait', 'PLANTAIRES', 'destinée', "d'estomac", '1,20', 'retr
L', 'malsaine', "d'imposer", '5x5', 'Decoupes', 'absorbance', 'Fro
78905
```

```
import spacy

vocab = set()
nlp   = spacy.load('fr_core_news_md')

for _, rev in df_fr['review_body'].items():
    doc = nlp(rev)
    for token in doc:
        vocab.add(token.lemma_)

vocab = vocab - set(stopwords.words('french'))
vocab = vocab - set(stopwords.words('english'))
vocab = list(vocab)

print(vocab[:50])
print(len(vocab))
```

```
['smart', 'lisser', 'rottweiler', 'positionnemer',
ntion', '13/11', 'arsenal', 'resserrement', 'crade'
'38/40', 'legerment', 'Meross', 'flocon.(voir', 's6
que', 'intervention', 'nageoire', 'fouiller', 'blon
e', 're-detecte', 'inspecter', 'neufs', 'équivoque'
e', 'réapprovisionnement']
50841
```

Zone d'incrustation vidéo

Partie 2.2 : Détection de récurrences et nettoyage de texte

Skills4All
digital-learning certifiant

2.2 - Détection de patterns et nettoyage de textes

↗ Objectif : nettoyer, normaliser

```
text_en = """
The packgae has been delivered but is damaged (photo:
https://i.postimg.cc/TPr4LQwY/MEA2-1.jpg). When i openned iy ,
every flask was broke :( I put 1 star because I cannot put
-100 starts #wtf >>>>>>>>>>> dont buy this brand $$$$ \U0001F643
"""

text_fr = """
Le colsi a été livré mais cabossé (photo :
https://i.postimg.cc/TPr4LQwY/MEA2-1.jpg). Quan j'ai ouvert ,
tous les flacons été casser :( abusé je met 1 étoiles parce
que je peux pas mettre -100 étoiles #labus >>>>>>>>>>>
plus jamais cette marque $$$$ \U0001F643
"""
```

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

Encodage des fichiers

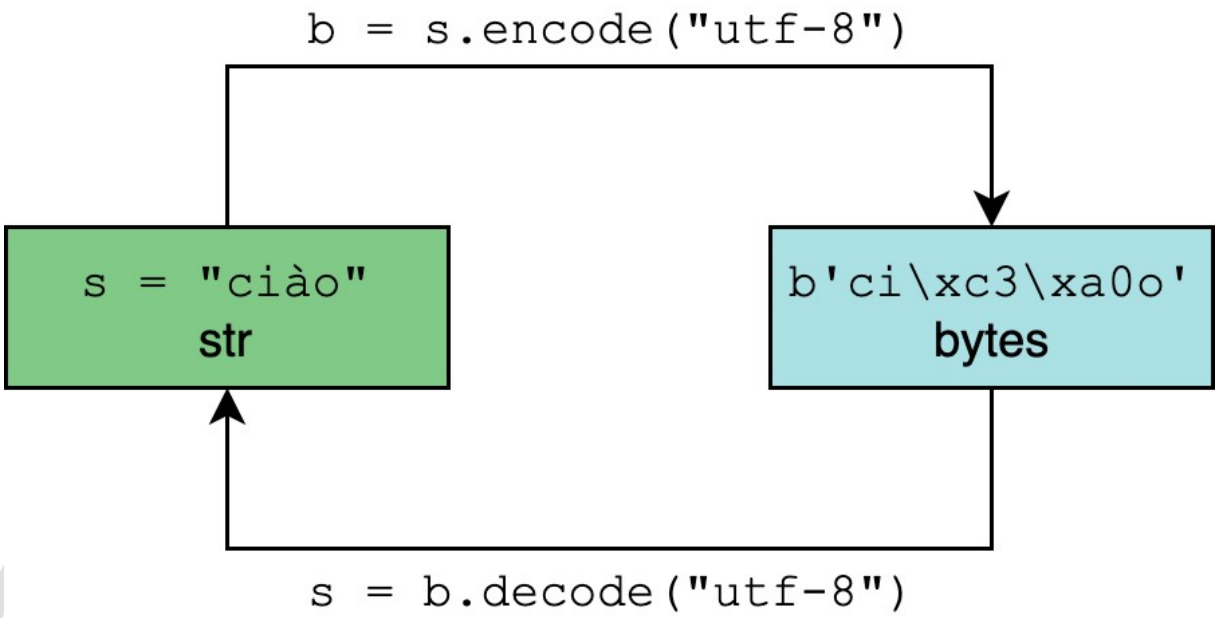
ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Skills4 digital-learning certificat

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes



Zone d'incrustation vidéo

digital-learning certifiant

2.2 - Détection de patterns et nettoyage de textes


```
import urllib.request

url  = 'https://martin.slouf.name/'
page = urllib.request.urlopen(url)

# the website responds with unidentified charset
print('the page advertised encoding: ' + str(page.headers.get_content_charset()))

# so we have bytes with unidentified encoding
byte_content = page.read()

# luckily we use decode_charset to decode the bytes as string
html, encoding = decode_charset(byte_content)
print('the page real encoding: ' + str(encoding))

> the page advertised encoding: None
> the page real encoding: utf_8
```

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes


```
data = b"""
La longueur de l'hypoth\xe9nuse au carr\xe9 dans un triangle
rectangle est \xe9gale \xe0 h\xb2 = a\xb2 + b\xb2
"""
print(data.decode("utf-8")) # error

> UnicodeDecodeError: 'utf-8' codec can't decode byte ...
```

```
# real encoding
data_str, encoding = decode_charset(data, suggested_encoding = 'latin-1')
print(encoding)
print(data_str)

> latin-1
>
> La longueur de l'hypothénuse au carré dans un triangle
> rectangle est égale à h² = a² + b²

# encoding that works
data_str, encoding = decode_charset(data)
print(encoding)
print(data_str)

> cp1252
>
> La longueur de l'hypothénuse au carré dans un triangle
> rectangle est égale à h² = a² + b²
```

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes

Regex : expressions régulières

- Un outil surpuissant pour détecter et manipuler des récurrences
- Un langage en soit, utilisable dans tous les langages de programmation
- Test en ligne : https://regex101.com

```
import re

text = 'Mon adresse email est blabla@gmail.com'
match = re.search(r'[a-z]+@[a-z]+\.com', text)

if match:
  print('found', match.group())
else:
  print('did not find')
```

```
found blabla@gmail.com
```

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes


```
'Mon adresse email est blabla@gmail.com'
```

```
[a-z]+@[a-z]+\.com
```

[a-z] La première partie de l'expression régulière (regex) est une classe de caractères définie par les lettres de *a* à *z*. Cela veut dire que cette partie de la regex détectera n'importe quelle lettre de *a* à *z*. Dans l'exemple cette partie matchera le *b* de l'adresse email. Dans une classe, on peut définir autant de caractères que l'on veut en les énumérant entre crochets ; On eut définir également un intervalle de caractères avec le tiret -. Par exemple *[ab0-9cd]* matchera les chiffres de 0 à 9, et les lettres *a, b, c, d.*

+ Ce symbole indique que le pattern précédent (ici *[a-z]*) se répète une fois ou plus (donc potentiellement une infinité de fois). Dans l'exemple cette partie matchera le *blabla* de l'adresse email.

@ Cette partie matchera uniquement le caractère arobase. Dans l'exemple cette partie matchera le *blabla@* de l'adresse email.

[a-z]+ C'est la même qu'au début et donc cela matchera une fois ou plus n'importe quelle lettre de *a* à *z*. Dans l'exemple cette partie matchera le *blabla@gmail* de l'adresse email.

\. Ici nous voulons matcher le caractère **.** mais c'est un caractère spécial dans les regex. Il s'agit d'un symbole qui matche n'importe quel carctère. Ce n'est pas ce que nous voulons, et donc l'antislash permet d'échaper ce caractère spécial pour matcher le caractère **.**. Dans l'exemple cette partie matchera le *blabla@gmail.* de l'adresse email.

com Ces trois caractères permettent de matcher le *com* à la fin de l'adresse email.

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes


```
text_1 = 'Mon adresse email est blabla@gmail.com'
text_2 = 'blabla@gmail.com est mon adresse mail'

regex_begin = r'\A[a-z]+@[a-z]+\.com'
regex_end   = r'[a-z]+@[a-z]+\.com\Z'

if re.search(regex_begin, text_1):
    print('email found at beginning in text_1')
if re.search(regex_begin, text_2):
    print('email found at beginning in text_2')
if re.search(regex_end, text_1):
    print('email found at the end in text_1')
if re.search(regex_end, text_2):
    print('email found at the end in text_2')

> email found at beginning in text_2
> email found at the end in text_1
```

Zone d'incrustation vidéo

Exercice

Uniquement avec ce que vous avez appris jusqu'à présent, modifiez la regex précédente afin qu'elle puisse matcher les adresses emails suivantes :

```
romain@hotmail.com
jeanyves@yahoo.com
romain@hotmail.fr
jean.yves@yahoo.com
```

Cependant, il ne faut pas que la regex matche les adresses emails malformées suivantes :

```
jean-yves@yahoo.com
romain@hotmail.gmail.fr
roger@yahoo
```

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes

Exercice corrigé

```
\A[a-z\.]+@[a-z]+\.[a-z]+\Z
```

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

Regex : Quantifieurs de répétition

* Le pattern précédent peut apparaitre 0 fois ou plus.

\+ Le pattern précédent peut apparaitre 1 fois ou plus.

? Le pattern précédent peut apparaitre 0 ou 1 fois.

{n,m} Le pattern précédent peut apparaitre entre n et m fois.

{n,} Le pattern précédent peut apparaitre n fois ou plus.

{,n} Le pattern précédent peut apparaitre au plus n fois.

*? Le pattern précédent peut apparaitre 0 fois ou plus. Dès que le pattern suivant apparaitre, ne pas poursuivre la répétition du précédent pattern.

\+? Le pattern précédent peut apparaitre 1 fois ou plus. Dès que le pattern suivant apparaitre, ne pas poursuivre la répétition du précédent pattern.

La regex **[a-z,]*\.? [cC]ependant** matchera aussi bien des phrases qui commencent par *Cependant*, comme dans "*comme au dernier **jour. Cependant***", que des phrases ou *cependant* est un mot comme un autre comme dans "*comme au dernier **jour, cependant***".

digital-learning certifiant

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes

↗️ Exercice

Vous disposez d'un programme qui effectue des recherches de fichiers dans votre disque, selon certains critères. Dans un autre programme, vous voulez exécuter le programme de recherche et agir différemment suivant son résultat. Le programme de recherche peut retourner les messages suivants :

```
do_match = [
    '1 file has been found',
    '34 files has been found',
    '2 directories has been found, 1 file has been found'
]

dont_match = [
    '15 files has been found with similar search parameters, no results with current parameters',
    'No files has been found',
    'The files has not been found',
    '5 files cound have been found, but the disk returned some I/O errors',
    '1 directory has been found, no file has been found'
]
```

Le but est de séparer les cas où au moins un fichier a été trouvé avec les parametres actuels de recherche (matcher cas 1 à 3, sans matcher cas 4 à 8).

digital-learning certifiant

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes

Exercice corrigé

```
[0-9]+ files? has been found\Z
```

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

Regex : classes et intervalles de caractères

\w Classe des caractères de mots. Cette classe contient les lettres a à z, A à Z, 0 à 9 et l'underscore. Attention cette classe ne prends pas compte les caractères accentués.

\W Classe négative de \w. Contient tous les caractères qui ne se trouvent pas dans \w.

\s Classe des espaces. Cette classe contient l'espace, la tabulation et d'autres caractères invisibles qui se présentent comme des espaces.

\S Classe de tout ce qui n'est pas un espace.

\d Classe des nombres.

\D Classe de tout ce qui n'est pas un nombre.

```
[abcd0123]+
```

```
[a-zA-Z0-9]+
```

```
[\w.,-*+]+
```

```
[^<>]*
```

Skills4 Learning

Zone d'incrustation vidéo

Exercice

```
text = """
<dl>
  <dt id="introduction_au_html">Introduction au HTML</dt>
  <dd>
    <p>Vous faites vos premiers pas dans le développement web ? <a
    href="/fr/docs/Learn/Getting_started_with_the_web/HTML_basics">Nos articles
    sur les bases de HTML</a> expliquent ce qu'est HTML et comment
    l'utiliser.</p>
  </dd>
  <dt id="tutoriels_html">Tutoriels HTML</dt>
  <dd>
    <p>Pour plus d'informations sur l'utilisation du HTML, des tutoriels et des
    exemples complets, vous pouvez consulter <a href="/fr/docs/Learn/HTML">notre
    section Apprendre HTML</a>.</p>
  </dd>
  <dt id="référence_html">Référence HTML</dt>
  <dd>
    <p>Dans notre <a href="/fr/docs/Web/HTML/Reference">référence
    exhaustive</a>, vous trouverez le détail de chaque élément et attribut
    constituant HTML.</p>
  </dd>
</dl>
"""
```

1. Avec une regex, matchez les valeurs des *id* des balises *dt*.
2. Avec une regex, matchez les textes des balises *a*.
3. Avec une regex, matchez les urls des balises *a* qui ne sont pas dans l'introduction.

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes

Exercice corrigé

Pour le premier exercice :

```
<dt id="[^"]+"
```

Pour le deuxième exercice :

```
[^<>]+<\/a>
```

Pour le troisième exercice :

```
<dt id="[^i][\w\W]*?href="[^"]+"
```

Zone d'incrustation vidéo

↗ Regex : re.findall et groupes capturants

```
text  = 'Mon adresse email est blabla@gmail.com et pas blublu@gmail.com'
match = re.findall(r'([a-z]+)@[a-z]+\.com', text)

print(match)
> ['blabla', 'blublu']
```

```
text  = 'Mon adresse email est blabla@gmail.com et pas blublu@gmail.com'
match = re.findall(r'([a-z]+)@([a-z]+\.com)', text)

print(match)
> [('blabla', 'gmail.com'), ('blublu', 'gmail.com')]
```

```
text = """
Les paroles de la chanson sont "bla bla bla bla, bla bla bla",
c'est ma préféré. Par contre je déteste celle qui fait
"bla bla bla, bla bla bla bla"
"""

match = re.findall(r'((bla,? ?)+)', text)
print(match)
> [('bla bla bla bla, bla bla bla', 'bla'), ('bla bla bla, bla bla bla bla', 'bla')]
```

Zone d'incrustation vidéo

Regex : groupes non-capturants

```
text = """
Les paroles de la chanson sont "bla bla bla bla, bla bla bla",
c'est ma préféré. Par contre je déteste celle qui fait
"bla bla bla, bla bla bla bla"
"""

match = re.findall(r'((?:bla,? ?)+)', text)
print(match)
> [['bla bla bla bla, bla bla bla'], ['bla bla bla, bla bla bla bla']]
```

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

Regex : conditionnalité des expressions

Une propriété des groupes que nous n'avons pas encore exploré est la conditionnalité. À l'intérieur d'un groupe il est possible de définir le match pour une sous-expression ou une autre, avec l'opérateur tuyau, | :

```
text  = "J'aime les pêches, mais aussi les poires"
match = re.findall(r'p(êche|oire)s', text)

print(match)
> ['êche', 'oire']
```

En combinant avec un groupe non-capturant, nous obtenons :

```
text  = "J'aime les pêches, mais aussi les poires"
match = re.findall(r'(p(?:êche|oire)s)', text)

print(match)
> ['pêches', 'poires']
```

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes

Exercice

- Ré-écrivez les regex de l'exercice précédent (*classes et intervalles de caractères*) pour capturer uniquement l'information demandé

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Exercice corrigé

Pour le premier exercice :

```
<dt id="([^"]+)"
```

Pour le deuxième exercice :

```
([^<>]+)<\/a>
```

Pour le troisième exercice :

```
<dt id="[^i][\w\W]*?href="([^"]+)"
```

Zone d'incrustation vidéo

↗️ Regex : re.sub

```
import re

text     = 'Mon adresse email est blabla@gmail.com et pas blublu@gmail.com'
new_text = re.sub(r'[a-z]+@[a-z]+\.com', '<EMAIL>', text)

print(new_text)
> Mon adresse email est <EMAIL> et pas <EMAIL>
```

```
import re

text     = 'Mon adresse email est blabla@gmail.com et pas blublu@gmail.com'
new_text = re.sub(r'([a-z]+)@([a-z]+\.com)','\\1[at]\\2', text)

print(new_text)
> Mon adresse email est blabla[at]gmail.com et pas blublu[at]gmail.com
```

digital-learning certifiant

Zone d'incrustation vidéo

Regex : tables récapitulatives (1 / 3)

Ancres	
^	Début de la chaine de caractères, ou début d'une ligne dans une regex multi-lignes
\A	Début de la chaine de caractères
$	Fin de a chaine de caractères, ou fin de la ligne dans une regex multi-lignes
\Z	Fin d'une chaine de caractères
\b	Limite d'un mot
\B	Tout ce qui n'est pas la limite d'un mot
\<	Début d'un mot
\>	Fin d'un mot

Classes de caratères	
\c	Caractères de contrôle
\s	Espaces
\S	Tout ce qui n'est pas un espace
\d	Nombres
\D	Tout ce qui n'est pas un nombre
\w	Caractères de mots
\W	Tout ce qui n'est pas un caractère de mots
\x	Nombre hexadécimal
\O	Tout nombre en base 8

Quantifieurs			
*	0 ou plus	{3}	Exactement 3 fois
+	1 ou plus	{3,}	3 ou plus
?	0 ou une fois	{3,5}	3, 4 ou 5

Ajouter ? à un quantifieur pour le rendre non-greedy (il arrête de matcher dès qu'un match est possible avec l'expression suivante)

Regex : tables récapitulatives (2 / 3)

Groupes, classes et intervalles	
.	N'importe quel caractère sauf une nouvelle ligne (\n)
(a|b)	a ou b
(...)	Groupe
(?:...)	Groupe non-capturant
[abc]	Classe (a ou b ou c)
[^abc]	Non (a ou b ou c)
[a-q]	Classe de l'intervalle des lettres de a à q
[A-Q]	Lettres de A à Q
[0-7]	Nombres de 0 à 7
Les intervalles sont inclusifs.	

Assertions	
?=	L'expression en avant existe
?!	L'expression en avant n'existe pas
?<=	L'expression en arrière existe
?!= or ?<!	L'expression en arrière n'existe pas
?>	N'utilise pas le backtracking
?()	Condition [if then]
?()|	Condition [if then else]
?#	Commentaire

Échappement	
\	Échappe le prochain caractère
\Q	Début d'une séquence littérale
\E	Fin d'une séquence littérale
Échapper un caractère signifie qu'il est interprété littéralement	

Caractères spéciaux communs			
^	[	.	$
{	*	(	\
+	)	|	?
<	>		
Ces caractères s'échappent avec \			

ls4
digital-learning c

Regex : tables récapitulatives (3 / 3)

Autre	
\n	Nouvelle ligne
\r	Retour chariot
\t	Tabulation
\v	Tabulation verticale
\f	Caractère spécial demandant à l'imprimante de changer de page
\xxx	Caractère en octal xxx
\xhh	Caractère en hexadécimal hh

Skills
digital-learning

Nettoyage et normalisation de textes

Prenons tout d'abords des textes en français et en anglais qui ne sont pas propres.

```
text_en = """
The packgae has been delivered but is damaged (photo:
https://i.postimg.cc/TPr4LQwY/MEA2-1.jpg). When i openned iy ,
every flask was broke :( I put 1 star because I cannot put
-100 starts #wtf >>>>>>>>>>> dont buy this brand $$$$ \U0001F643
"""

text_fr = """
Le colsi a été livré mais cabossé (photo :
https://i.postimg.cc/TPr4LQwY/MEA2-1.jpg). Quan j'ai ouvert ,
tous les flacons été casser :( abusé je met 1 étoiles parce
que je peux pas mettre -100 étoiles #labus >>>>>>>>>>>
plus jamais cette marque $$$$ \U0001F643
"""
```

Note : dans cette partie vous aurez besoin d'installer les librairies suivantes :

```
pip install emot torch transformers sentencepiece autocorrect clean-text \
textblob wordfreq
```

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes

Nettoyage et normalisation basiques

```python
from cleantext import clean

print(clean(text_en,
    fix_unicode                 = True,
    to_ascii                    = False,
    lower                       = True,
    no_line_breaks              = True,
    no_urls                     = True,
    no_emails                   = True,
    no_phone_numbers            = True,
    no_numbers                  = True,
    no_digits                   = True,
    no_currency_symbols         = True,
    no_punct                    = False,
    replace_with_punct          = "",
    replace_with_url            = "<URL>",
    replace_with_email          = "<EMAIL>",
    replace_with_phone_number   = "<PHONE>",
    replace_with_number         = "<NUMBER>",
    replace_with_digit          = "<DIGIT>",
    replace_with_currency_symbol = "<CUR>",
    lang                        = "en"
))
```

```
> the packgae has been delivered but is damaged (photo: <url>).
> when i openned iy , every flask was broke :( i put <number> star
> because i cannot put <number> starts #wtf >>>>>>>>>>> dont buy
> this brand <cur>
```

Zone d’incrustation vidéo

Normalisation des hashtags

```
import re

hashtag_regex = r'#[a-zà-ÿ_]+'

print(re.sub(hashtag_regex, '<hashtag>', text_en))
print(re.sub(hashtag_regex, '<hashtag>', text_fr))

> [...] -100 starts <hashtag> >>>>>>>>>>>  [...]
> [...] -100 étoiles <hashtag> >>>>>>>>>>>  [...]
```

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

2.2 - Détection de patterns et nettoyage de textes

↗️ Traduction des émojis et émoticons

```
from emot.emo_unicode import EMOTICONS_EMO, UNICODE_EMOJI

def convert_emojis(text):
    for emot in UNICODE_EMOJI:
        emot_regex = r'(' + re.escape(emot) + ')'
        text = re.sub(emot_regex, "_".join(
            UNICODE_EMOJI[emot].replace(",","").replace(":","").replace("-","_").lower().split()), text
        )
    return text

def convert_emoticons(text):
    for emot in EMOTICONS_EMO:
        emot_regex = r'(' + re.escape(emot) + ')'
        text = re.sub(emot_regex, "_".join(EMOTICONS_EMO[emot].replace(",","").lower().split()), text)
    return text

print(convert_emojis(convert_emoticons(text_en)))
print(convert_emojis(convert_emoticons(text_fr)))
```

```
The packgae has been delivered but is damaged (photo:
httpsskeptical_annoyed_undecided_uneasy_or_hesitant/i.postimg.cc/TPr4LQwY/MEA2-1.jpg). When i openned iy ,
every flask was broke frown_sad_andry_or_pouting I put 1 star because I cannot put
-100 starts #wtf >>>>>>>>>>> dont buy this brand $$$$ upside_down_face


Le colsi a été livré mais cabossé (photo :
httpsskeptical_annoyed_undecided_uneasy_or_hesitant/i.postimg.cc/TPr4LQwY/MEA2-1.jpg). Quan j'ai ouvert ,
tous les flacons été casser frown_sad_andry_or_pouting abusé je met 1 étoiles parce
que je peux pas mettre -100 étoiles #labus >>>>>>>>>>>
plus jamais cette marque $$$$ upside_down_face
```

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes

Nettoyage des caractères spéciaux

```
special_char_regex = r'[^_A-zÀ-ÿ<>0-9\'\s.,;?!:%\(\ç)-+]+'
not_tag_regex      = r'(?:<{2,}|>{2,})'
space_norm_regex   = r'\s+'

text_en_norm = re.sub(special_char_regex, '', text_en)
text_fr_norm = re.sub(special_char_regex, '', text_fr)

text_en_norm = re.sub(not_tag_regex, '', text_en_norm)
text_fr_norm = re.sub(not_tag_regex, '', text_fr_norm)

text_en_norm = re.sub(space_norm_regex, ' ', text_en_norm).strip()
text_fr_norm = re.sub(space_norm_regex, ' ', text_fr_norm).strip()

print(text_en_norm)
print(text_fr_norm)
```

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

2.2 - Détection de patterns et nettoyage de textes

- Corrections des fautes de frappes, erreurs orthographiques, de grammaire et de conjugaison
  - Les approches algorithmiques déterministes
  - Les approches de machine learning
  - Les approches linguistiques
  - Une combinaison d'approches

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Corrections : approches déterministes

```
from autocorrect import Speller

spell_en = Speller('en')
spell_fr = Speller('fr')

print(spell_en("When i openned iy , every flask was broke"))
print(spell_fr("Quan j'ai ouvert , tous les flacons été casser"))

> When i opened iy , every flask was broke
> Quan j'ai ouvert , tous les flacons été casser
```

```
from textblob import TextBlob, Word

print(TextBlob(text_en).correct())

word = Word('falibility')
print(word.spellcheck())
```

```
The package has been delivered but is damaged (photo:
http://i.posting.c/TPr4LQwY/MEA2-1.pg). When i opened in ,
every flask was broke :( I put 1 star because I cannot put
-100 starts #utf >>>>>>>>>>> dont buy this brand $$$$ 🙃

[('fallibility', 1.0)]
```

Zone d'incrustation vidéo

Corrections : approches machine learning

```
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

tokenizer = AutoTokenizer.from_pretrained("Bhuvana/t5-base-spellchecker")
model     = AutoModelForSeq2SeqLM.from_pretrained("Bhuvana/t5-base-spellchecker")

def correct(inputs):
    input_ids = tokenizer.encode(inputs, return_tensors = 'pt')
    sample_output = model.generate(
        input_ids,
        do_sample            = True,
        max_length           = 50,
        top_p                = 0.99,
        num_return_sequences = 1
    )
    res = tokenizer.decode(sample_output[0], skip_special_tokens = True)
    return res

text = "christmas is celbrated on decembr 25 evry ear"

print(correct(text))
print(correct("When i openned iy , every flask was broke"))

> christmas is celebrated on december 25 every ear
> i opened it s every flask was broken
```

Zone d'incrustation vidéo

Corrections : espaces

```
str_en = 'thumbgreenappleactiveassignmentweeklymetaphor'
str_fr = 'lesapprocheslesplusaupointdanscedomainenesontpaspubliques'

print(infer_spaces(str_en, 'en'))
print(infer_spaces(str_fr, 'fr'))

> thumb green apple active assignment weekly metaphor
> les approches les plus au point dansce domaine ne sontpas publiques
```

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

2.2 - Détection de patterns et nettoyage de textes

Exercice

- Avec l'ensemble des outils que nous avons vu précédemment, construisez une fonction par langage qui nettoie et normalise les textes

```
print(clean_en(text_en))

> the package has been delivered but is damaged (photo: <url>). when i opened iy ,
> every flask was broke frown_sad_andry_or_pouting i put <number> star because i
> cannot put <number> starts <hashtag> dont buy this brand <cur> upside_down_face
```

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Exercice corrigé

```python
def clean_en(text):
    cleaned_text = clean(
        text, True, False, True, True, True, True, True,
        True, True, True, False, "", "<URL>", "<EMAIL>",
        "<PHONE>", "<NUMBER>", "<DIGIT>", "<CUR>", "en"
    )

    cleaned_text = convert_emojis(cleaned_text)
    cleaned_text = convert_emoticons(cleaned_text)

    hashtag_regex      = r'#[a-zà-ÿ_]+'
    special_char_regex = r'[^_A-zÀ-ÿ<>0-9\'\s.,;?!:%\(\ç)-+]+'
    not_tag_regex      = r'(?:<{2,}|>{2,})'
    space_norm_regex   = r'\s+'

    cleaned_text = re.sub(hashtag_regex, '<hashtag>', cleaned_text)
    cleaned_text = re.sub(special_char_regex, '', cleaned_text)
    cleaned_text = re.sub(not_tag_regex, '', cleaned_text)
    cleaned_text = re.sub(space_norm_regex, ' ', cleaned_text).strip()

    corrected = ''
    for word in cleaned_text.split():
        if not re.search('[<>_:,.!?\(\])]', word):
            alt_words = Word(word).spellcheck()
            if len(alt_words) > 0:
                alt_word, confidence = alt_words[0]
                if confidence > 0.6:
                    corrected += alt_word + ' '
                    continue

        corrected += word + ' '

    cleaned_text = corrected

    return cleaned_text
```

Zone d'incrustation vidéo

Partie 2.3 : Étiqueter les symboles (token tagging)

Skills4All
digital-learning certifiant

2.3 – Étiqueter les symboles

- Tâches de tagging communes du NLP
  - Part-Of-Speech (POS)
  - Lemmatisation
  - Morphologie flexionnelle (Inflectional Morphology)
  - Analyse des dépendances (Dependency Parsing)
  - Reconnaissance d'entités nommées (NER : Named Entity Recognition)
  - Liaison d'entités (Entity Linking)

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

2.3 – Étiqueter les symboles

- Usages
  - Filtrer certains éléments du texte (nettoyage, normalisation)
  - Utiliser les tags en temps que tel (extraction de connaissance)
  - Utiliser les tags pour comme données additionnelles pour l'entrainement d'un modèle
- Objectif : tagger le texte suivant

```
text = """
NASA is "go" for a post-breakfast launch of the biggest rocket ever—and that
includes the final flight of the Saturn V in 1973. A flight readiness review
this week confirmed that its Artemis-1 mission will launch during a two-hour
window that opens at 8:33 a.m. EDT on Monday, August 29. However, if for any
reason it doesn't launch on schedule then the next time it can go is at
lunchtime on Friday, September 2, 2022.
"""
```

Zone d'incrustation vidéo

2.3 – Étiqueter les symboles

Part-Of-Speech

```python
import spacy

nlp = spacy.load("en_core_web_lg")
doc = nlp(text)

for token in doc:
    print(f'{token.text}\t{token.pos_}')
```

Token	POS tag	Token	POS tag	Token	POS tag
	SPACE	that	PRON	week	NOUN
NASA	PROPN		SPACE	confirmed	VERB
is	AUX	includes	VERB	that	SCONJ
"	PUNCT	the	DET	its	PRON
go	VERB	final	ADJ	Artemis-1	PROPN
"	PUNCT	flight	NOUN	mission	NOUN
for	ADP	of	ADP	will	AUX
a	DET	the	DET	launch	VERB
post	ADJ	Saturn	PROPN	during	ADP
-	ADJ	V	PROPN	a	DET
breakfast	ADJ	in	ADP	two	NUM
launch	NOUN	1973	NUM	-	PUNCT
of	ADP	.	PUNCT	hour	NOUN
the	DET	A	DET		SPACE
biggest	ADJ	flight	NOUN	window	NOUN
rocket	NOUN	readiness	NOUN	that	PRON
ever	ADV	review	NOUN	opens	VERB
—	PUNCT		SPACE	at	ADP
and	CCONJ	this	DET	8 :33	NUM

Zone d'incrustation vidéo

2.3 – Étiqueter les symboles

POS tag	Description	Examples
ADJ	adjective	*big, old, green, incomprehensible, first*
ADP	adposition	*in, to, during*
ADV	adverb	*very, tomorrow, down, where, there*
AUX	auxiliary	*is, has (done), will (do), should (do)*
CONJ	conjunction	*and, or, but*
CCONJ	coordinating conjunction	*and, or, but*
DET	determiner	*a, an, the*
INTJ	interjection	*psst, ouch, bravo, hello*
NOUN	noun	*girl, cat, tree, air, beauty*
NUM	numeral	*1, 2017, one, seventy-seven, IV, MMXIV*
PART	particle	*'s, not,*
PRON	pronoun	*I, you, he, she, myself, themselves, somebody*
PROPN	proper noun	*Mary, John, London, NATO, HBO*
PUNCT	punctuation	*., (,), ?*
SCONJ	subordinating conjunction	*if, while, that*
SYM	symbol	*$, +, -, *, =, :)*
VERB	verb	*run, runs, running, eat, ate, eating*
X	other	*sfpksdpsxmsa*
SPACE	space	

digital-learning certifiant

Zone d'incrustation vidéo

Morphologie flexionnelle

```
for token in doc:
    print(f'{token.text} : {token.tag_} : {token.morph}')

# Token      : Tag : Morph

> NASA       : NNP : Number=Sing
> is         : VBZ : Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin
> "          : ``  : PunctSide=Ini|PunctType=Quot
> go         : VB  : VerbForm=Inf
> "          : ''  : PunctSide=Fin|PunctType=Quot
> for        : IN
> a          : DT  : Definite=Ind|PronType=Art
> post       : JJ  : Degree=Pos
> -          : JJ  : Degree=Pos
> breakfast  : JJ  : Degree=Pos
> launch     : NN  : Number=Sing
> of         : IN
> the        : DT  : Definite=Def|PronType=Art
> biggest    : JJS : Degree=Sup
> rocket     : NN  : Number=Sing
> ever       : RB
> -          :
```

Zone d'incrustation vidéo

2.3 – Étiqueter les symboles

POS	TAG	*DESCRIPTION*	EXAMPLE
ADJ	AFX	*affix*	The Flintstones were a **pre**-historic family.
ADJ	JJ	*adjective*	This is a **good** sentence.
ADJ	JJR	*adjective, comparative*	This is a **better** sentence.
ADJ	JJS	*adjective, superlative*	This is the **best** sentence.
ADJ	PDT	*predeterminer*	Waking up is **half** the battle.
ADJ	PRP$	*pronoun, possessive*	**His** arm hurts.
ADJ	WDT	*wh-determiner*	It's blue, **which** is odd.
ADJ	WP$	*wh-pronoun, possessive*	We don't know **whose** it is.
ADP	IN	*conjunction, subordinating or preposition*	It arrived **in** a box.
ADV	EX	*existential there*	**There** is cake.
ADV	RB	*adverb*	He ran **quickly.**
ADV	RBR	*adverb, comparative*	He ran **quicker.**
ADV	RBS	*adverb, superlative*	He ran **fastest.**
ADV	WRB	*wh-adverb*	**When** was that ?
CONJ	CC	*conjunction, coordinating*	The balloon popped **and** everyone jumped.
DET	DT	*determiner*	**This** is **a** sentence.
INTJ	UH	*interjection*	**Um**, I don't know.
NOUN	NN	noun, singular or mass	This is a **sentence**.
NOUN	NNS	noun, plural	These are **words**.
NOUN	WP	wh-pronoun, personal	**Who** was that ?
NUM	CD	cardinal number	I want **three** things.
PART	POS	possessive ending	Fred**'s** name is short.
PART	RP	adverb, particle	Put it **back** !
PART	TO	infinitival to	I want **to** go.
PRON	PRP	pronoun, personal	**I** want **you** to go.
PROPN	NNP	noun, proper singular	**Kilroy** was here.
PROPN	NNPS	noun, proper plural	The **Flintstones** were a pre-historic family.
VERB	MD	verb, modal auxiliary	This **could** work.
VERB	VB	verb, base form	I want to **go**.
VERB	VBD	verb, past tense	This **was** a sentence.
VERB	VBG	verb, gerund or present participle	I am **going**.
VERB	VBN	verb, past participle	The treasure was **lost**.
VERB	VBP	verb, non-3rd person singular present	I **want** to go.
VERB	VBZ	verb, 3rd person singular present	He **wants** to go.

Zone d'incrustation vidéo

2.3 – Étiqueter les symboles

Analyse des dépendances

```
for token in doc:
    print(f'{token.text} => {token.dep_} => {token.head.text}')

>           => dep    => go
> NASA      => nsubj  => go
> is        => aux    => go
> "         => punct  => go
> go        => ROOT   => go
> "         => punct  => go
> for       => prep   => go
> a         => det    => launch
> post      => amod   => launch
> -         => amod   => launch
> breakfast => amod   => launch
> launch    => pobj   => for
> of        => prep   => launch
> the       => det    => rocket
> biggest   => amod   => rocket
> rocket    => pobj   => of
> ever      => advmod => rocket
> -         => punct  => go
> and       => cc     => go
> that      => nsubj  => includes
```

Zone d'incrustation vidéo

2.3 – Étiqueter les symboles

Dep Tag (link)	Description
acl	clausal modifier of noun (adnominal clause)
acl:relcl	relative clause modifier
advcl	adverbial clause modifier
advmod	adverbial modifier
advmod:emph	emphasizing word, intensifier
advmod:lmod	locative adverbial modifier
amod	adjectival modifier
appos	appositional modifier
aux	auxiliary
aux:pass	passive auxiliary
case	case marking
cc	coordinating conjunction
cc:preconj	preconjunct
ccomp	clausal complement
clf	classifier
compound	compound
compound:lvc	light verb construction
compound:prt	phrasal verb particle
compound:redup	reduplicated compounds
compound:svc	serial verb compounds
conj	conjunct
cop	copula
csubj	clausal subject
csubj:outer	outer clause clausal subject
csubj:pass	clausal passive subject
dep	unspecified dependency
det	determiner
det:numgov	pronominal quantifier governing the case of the noun
det:nummod	pronominal quantifier agreeing in case with the noun
det:poss	possessive determiner
discourse	discourse element
dislocated	dislocated elements
expl	expletive
expl:impers	impersonal expletive
expl:pass	reflexive pronoun used in reflexive passive
expl:pv	reflexive clitic with an inherently reflexive verb
fixed	fixed multiword expression
flat	flat multiword expression
flat:foreign	foreign words
flat:name	names
goeswith	goes with
iobj	indirect object
list	list
mark	marker
nmod	nominal modifier
nmod:poss	possessive nominal modifier
nmod:tmod	temporal modifier

digital-learning certifiant

2.3 – Étiqueter les symboles

Information détaillée, avec exemples appliqués sur : https://universaldependencies.org/u/dep/

nsubj	nominal subject
nsubj:outer	outer clause nominal subject
nsubj:pass	passive nominal subject
nummod	numeric modifier
nummod:gov	numeric modifier governing the case of the noun
obj	object
obl	oblique nominal
obl:agent	agent modifier
obl:arg	oblique argument
obl:lmod	locative modifier
obl:tmod	temporal modifier
orphan	orphan
parataxis	parataxis
punct	punctuation
reparandum	overridden disfluency
root	root
vocative	vocative
xcomp	open clausal complement

2.3 – Étiqueter les symboles

Reconnaissance d'entités nommées (NER)

```
for entity in doc.ents:
    print(f'{entity.text} : {entity.label_}')

> NASA                        : ORG
> Saturn                      : PRODUCT
> 1973                        : DATE
> this week                   : DATE
> two-hour                    : TIME
> 8:33 a.m. EDT               : TIME
> Monday, August 29           : DATE
> Friday, September 2, 2022 : DATE
```

Labels possibles :

- CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART

Zone d'incrustation vidéo

Skills4 Digital Learning Unit

2.3 – Étiqueter les symboles

- Liaison d'entités
  - Lorsqu'il n'y a qu'une seule entité possible pour un nom, alors il s'agit d'un simple match dans une base de données
  - Lorsqu'il y a plusieurs entités possibles, il faut une méthode qui permette de choisir une des entités possibles, en fonction ou non du contexte.

```
pip install spacy-entity-linker
python -m spacy_entity_linker "download_knowledge_base"
```

digital-learning certifiant

Zone d'incrustation vidéo

2.3 – Étiqueter les symboles


```
nlp.add_pipe("entityLinker", last = True)
doc = nlp(text)

print('wikidata entities summary ordered by categories')
doc._.linkedEntities.print_super_entities()

print()
print('wikidata entities details')
for s_index, sentence in enumerate(doc.sents):
    entities = sentence._.linkedEntities
    print(f'sentence {s_index}')
    for e_index, entity in enumerate(entities):
        print(f'  entity {e_index}')
        print(f'    text: {entity.get_span()}')
        print(f'    id: {entity.get_id()}')
        print(f'    label: {entity.get_label()}')
        print(f'    description: {entity.get_description()}')
```

```
>  entity 1
>    text: launch
>    id: 361945
>    label: slipway
>    description: ramp on the shore by which ships or boats can be moved
>    to and from the water
>  entity 2
>    text: rocket
>    id: 41291
>    label: rocket
>    description: pyrokinetic engine used for propulsion; for the
>    incendiary weapon, see Q2037215
```

Zone d'incrustation vidéo

Partie 3 : Représenter les symboles comme des vecteurs numériques

Skills4All
digital-learning certifiant

Sommaire

- Vectorisation des symboles
  - Vectorisation par la présence et la fréquence des tokens
  - Vectorisation par la contingence des tokens
  - Vectorisation par apprentissage auto-supervisé

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Partie 3.1 : Vectorisation par présence / fréquence

Skills4All
digital-learning certifiant

3.1 – Vectorisation par présence / fréquence

- Rappel de vocabulaire
  - Le token est l'élément unitaire
  - Une phrase est séquence de tokens
  - Un document est une séquence de phrase
  - Un corpus est un ensemble de documents
  - Un vocabulaire est l'ensemble de tokens uniques dans un corpus

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

3.1 – Vectorisation par présence / fréquence

One-Hot Encoding

- Représentation la plus simple d'une liste de symboles.
- Chaque index du vecteur corresponds à un mot du vocabulaire
- Le vecteur est nul sauf à l'index du token existant où sa valeur est 1.

```
labels = [
    'chien', 'chat', 'ours', 'loup',
    'chat', 'chat', 'ours', 'chien',
    'chien', 'loup', 'ours', 'ours'
]
```

```
one_hot_labels = [
    [1.0, 0.0, 0.0, 0.0], [0.0, 1.0, 0.0, 0.0],
    [0.0, 0.0, 1.0, 0.0], [0.0, 0.0, 0.0, 1.0],
    [0.0, 1.0, 0.0, 0.0], [0.0, 1.0, 0.0, 0.0],
    [0.0, 0.0, 1.0, 0.0], [1.0, 0.0, 0.0, 0.0],
    [1.0, 0.0, 0.0, 0.0], [0.0, 0.0, 0.0, 1.0],
    [0.0, 0.0, 1.0, 0.0], [0.0, 0.0, 1.0, 0.0]
]
```

Zone d'incrustation vidéo

3.1 – Vectorisation par présence / fréquence

One-Hot Encoding

```
import numpy as np

text = """
Il jouait on ne sait quel effrayant jeu de cache-cache avec la mort ;
chaque fois que la face camarde du spectre s'approchait, le gamin
lui donnait une pichenette. Une balle pourtant, mieux ajustée ou
plus traître que les autres, finit par atteindre l'enfant feu follet.
On vit Gavroche chanceler, puis il s'affaissa.
"""

vocab       = dict()
vocab_index = 0
for word in text.split():
    if word not in vocab:
        vocab[word]  = vocab_index
        vocab_index += 1

def make_one_hot(word, vocab):
    vect = np.zeros(len(vocab))
    if word in vocab:
        vect[vocab[word]] = 1

    return vect

print(make_one_hot('Gavroche', vocab))
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.
 0. 0. 0.]
```

Zone d'incrustation vidéo

3.1 – Vectorisation par présence / fréquence

Fréquence d'apparition de symboles

- Amélioration du one-hot encoding pour représenter un document

```
def vectorize_with_freq(text, vocab):
    vect = np.zeros(len(vocab))

    words = text.split()
    for word in words:
        vect[vocab[word]] += 1

    vect /= len(words)
    return vect

vectorize_with_freq(text, vocab)
```

```
array([0.01886792, 0.01886792, 0.01886792, 0.01886792, 0.01886792,
       0.01886792, 0.01886792, 0.01886792, 0.01886792, 0.01886792,
       0.01886792, 0.03773585, 0.01886792, 0.01886792, 0.01886792,
       0.01886792, 0.03773585, 0.01886792, 0.01886792, 0.01886792,
       0.01886792, 0.01886792, 0.01886792, 0.01886792, 0.01886792,
       0.01886792, 0.01886792, 0.01886792, 0.01886792, 0.01886792,
       0.01886792, 0.01886792, 0.01886792, 0.01886792, 0.01886792,
       0.01886792, 0.01886792, 0.01886792, 0.01886792, 0.01886792,
       0.01886792, 0.01886792, 0.01886792, 0.01886792, 0.01886792,
       0.01886792, 0.01886792, 0.01886792, 0.01886792, 0.01886792,
       0.01886792])
```

Zone d'incrustation vidéo

3.1 – Vectorisation par présence / fréquence

- Fréquence d'apparition des symboles
  - Problèmes
    - La taille d'un document est égale à celle du vocabulaire
    - Toujours pas de représentation sémantique ('vent' et 'vents' sont considérés comme des features différentes)
    - La fréquence d'apparition ne met pas forcément en valeur les tokens importants ('le' a une fréquence plus grande que 'Gavroche')

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

3.1 – Vectorisation par présence / fréquence

Fréquence d'apparition de Ngrams

- Suite de tokens de taille n
- Généralement couplé à une segmentation plus courte (en caractère par exemple)
- Permet de contrôler la taille du vocabulaire

```python
def n_gram_split(text, n = 2):
    sequence = list()
    for i, c in enumerate(text):
        if i + n - 1 == len(text):
            break

        gram = c
        for j in range(1, n):
            gram += text[i + j]

        sequence.append(gram)

    return sequence

print(n_gram_split(text))
```

```
['\nI', 'Il', 'l ', ' j', 'jo', 'ou'
i', 'it', 't ', ' q', 'qu', 'ue', 'e
e', 'eu', 'u ', ' d', 'de', 'e ', '
v', 've', 'ec', 'c ', ' l', 'la', 'a
'qu', 'ue', 'e ', ' f', 'fo', 'oi',
'e ', ' c', 'ca', 'am', 'ma', 'ar',
'e ', ' s', "s'", "'a", 'ap', 'pp',
'ga', 'am', 'mi', 'in', 'n ', ' \n',
u', 'un', 'ne', 'e ', ' p', 'pi', 'j
', ' b', 'ba', 'al', 'll', 'le', 'e
e', 'eu', 'ux', 'x ', ' a', 'aj', 'j
'us', 's ', ' t', 'tr', 'ra', 'aî',
'au', 'ut', 'tr', 're', 'es', 's,',
'tt', 'te', 'ei', 'in', 'nd', 'dr',
'eu', 'u ', ' f', 'fo', 'ol', 'll',
G', 'Ga', 'av', 'vr', 'ro', 'oc', 'c
```

Zone d'incrustation vidéo

3.1 – Vectorisation par présence / fréquence

- Fréquence d'apparition de Ngrams
  - Perte d'une forme simple de sens
  - Gain sur le contrôle de la taille des vecteurs
  - Possibilité de combiner ngrams de taille 1,2,3,4 …
  - Toujours pas de représentation sémantique
  - Ne met toujours pas en valeurs les tokens importants

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

3.1 – Vectorisation par présence / fréquence

- **TF-IDF**
  - Un moyen de mettre en valeur la fréquence d'apparition des tokens importants
  - La fréquence des tokens d'un document (Term Frequence – TD) sont divisés par les fréquence d'apparition des tokens dans le corpus (Inverse Document Frequence – IDF)
  - Plus un token est fréquent dans le document et / ou plus il est rare dans le corpus, plus son TF-IDF sera élevé

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

TF-IDF - Implémentation

```python
from wordfreq import word_frequency

min_freq = 9e999
for word in text.split():
    if word in vocab:
        word_freq = word_frequency(word, 'fr')
        if word_freq != 0.0:
            if word_freq < min_freq:
                min_freq = word_freq

def tf_idf(text, vocab, min_freq):
    vect = np.zeros(len(vocab))

    words = text.split()
    for word in words:
        vect[vocab[word]] += 1

    vect /= len(words)

    for word, index in vocab.items():
        word_freq = word_frequency(
            word    = word,
            lang    = 'fr',
            minimum = min_freq
        )
        vect[index] /= word_freq

    return vect

print(tf_idf(text, vocab, min_freq))
```

```
[1.97569890e+00 1.57232704e+03 3.67795800e+00 3.85059684e+00
 9.02771509e+01 6.69075338e+01 3.93902391e+03 6.10612444e+01
 3.93902391e-01 1.11644524e+03 4.22101220e+00 1.40281967e+00
 4.31760287e+01 5.97086219e+05 4.74068455e+01 2.11761218e+01
 3.22528624e+00 7.17411579e+01 3.43053173e+05 1.93121029e+00
 1.79694519e+03 1.36724091e+04 8.42318059e-01 9.43396226e+02
 1.19417244e+01 1.36724091e+03 1.88679245e+00 8.81678716e+04
 1.88679245e+00 4.74068455e+02 1.19417244e+02 4.03160781e+01
 2.54627861e+04 6.55136268e+00 3.93902391e+00 3.85059684e+03
 1.05999576e+00 1.88679245e+01 4.85036620e+02 3.59389039e+00
 3.51357999e+02 1.02543068e+02 1.30123617e+02 5.82343350e+04
 3.67795800e+00 2.85445152e+02 7.17411579e+04 1.88679245e+05
 4.03160781e+01 1.97569890e+00 5.97086219e+05]
```

Zone d'incrustation vidéo

Partie 3.2 : Vectorisation par contingence

Skills4All
digital-learning certifiant

3.2 – Vectorisation par contingence

- Comment représenter la sémantique ?
  - Une meilleure question : comment nous, en tant qu'humain, pouvons comprendre un nouveau mot ?
  - Que veut dire le mot « amphigourique » ?
  - « Ce qu'elle m'a dit était amphigourique »
  - « Il bafouillait, son propos était amphigourique. Je n'ai pas confiance »
  - « La bravitude, le bruit et l'odeur etc. J'adore l'amphigourisme des policiens »
  - « Il joue à l'idiot avec ses amphigourismes. Hier il m'a dit qu'il était un fan invertébré de Johnny Holiday »

Zone d'incrustation vidéo

Skills4 digital-learning certifiant

3.2 – Vectorisation par contingence

- Comment représenter la sémantique ?
  - Amphigourique : un discours burlesque, volontairement ou involontairement obscur, incohérent, confus ou inintelligible
  - Le sens d'un mot est donné par sa propension à être accompagné par d'autres mots
  - Le sens d'un mot est donné par son contexte

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

3.2 – Vectorisation par contingence

Matrice de co-occurences

```
documents = [
    'un chat a un collier',
    'un chien n\'aiment pas un chat',
    'un chien a une laisse',
    'elle a un chien et un chat'
]

vocab = {
    'chien'   : 0,
    'chat'    : 1,
    'collier' : 2,
    'laisse'  : 3
}

word_vector_matrix = np.zeros((len(vocab), len(vocab)))
for document in documents:
    words = document.split()
    for i in range(len(words)):
        w_1 = words[i]
        if w_1 not in vocab:
            continue

        w_1_index = vocab[w_1]
        for j in range(len(words)):
            if i == j:
                continue

            w_2 = words[j]
            if w_2 not in vocab:
                continue

            w_2_index = vocab[w_2]

            word_vector_matrix[w_1_index][w_2_index] += 1

print(word_vector_matrix)
```

	Chien	Chat	Laisse	Collier
Chien	0	2	0	1
Chat	2	0	1	0
Laisse	0	1	0	0
Collier	1	0	0	0

Zone d'incrustation vidéo

3.2 – Vectorisation par contingence

- Matrice de co-occurences
  - + Donne une représentation rudimentaire de la sémantique
  - - La taille des vecteurs de mots est celle du vocabulaire
  - - Les mots les plus fréquents sont les plus représentés
  - - La représentation est très sensible au changements de mots peu fréquents
  - - La représentation est très sensible au nombre de documents dans le corpus

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

3.2 – Vectorisation par contingence

- **Pointwise Mutual Information (PMI)**
  - **Solution : utiliser la probabilité de co-occurence pondérée par la fréquence de chacun des tokens**

$$PMI(w_1, w_2) = log \frac{P(w_1, w_2)}{P(w_1)P(w_2)}$$

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

3.2 – Vectorisation par contingence

Pointwise Mutual Information (PMI)

```
import math

word_freq          = np.zeros(len(vocab))
word_vector_matrix = np.zeros((len(vocab), len(vocab)))

for document in documents:
    words = document.split()
    for i in range(len(words)):
        w_1 = words[i]
        if w_1 not in vocab:
            continue

        w_1_index = vocab[w_1]
        word_freq[w_1_index] += 1
        for j in range(len(words)):
            if i == j:
                continue

            w_2 = words[j]
            if w_2 not in vocab:
                continue

            w_2_index = vocab[w_2]

            word_vector_matrix[w_1_index][w_2_index] += 1

word_freq          /= len(vocab)
word_vector_matrix /= len(vocab)

for i in range(len(word_freq)):
    for j in range(len(word_freq)):
        if i == j:
            continue

        if word_vector_matrix[i][j] != 0:
            word_vector_matrix[i][j] /= word_freq[i] * word_freq[j]
            word_vector_matrix[i][j]  = math.log(word_vector_matrix[i][j])
```

Zone d'incrustation vidéo

3.2 – Vectorisation par contingence

Pointwise Mutual Information (PMI)

```
print(word_vector_matrix)
print()
print(vocab)
for i, word in enumerate(vocab):
    word_vector = word_vector_matrix[i]
    for j, v in enumerate(word_vector):
        if v < 0:
            word_vector[j] = 0

    print(f'{word} : {word_vector}')
```

```
[[ 0.          -0.11778304  0.          0.28768207]
 [-0.11778304  0.           0.28768207  0.        ]
 [ 0.           0.28768207  0.          0.        ]
 [ 0.28768207  0.           0.          0.        ]]

{'chien': 0, 'chat': 1, 'collier': 2, 'laisse': 3}
chien : [0.         0.         0.         0.28768207]
chat : [0.         0.         0.28768207 0.        ]
collier : [0.         0.28768207 0.         0.        ]
laisse : [0.28768207 0.         0.         0.        ]
```

- La représentation n’est plus sensible aux nombre de documents dans le vocabulaire et au changement du nombre de mots fréquents

Zone d’incrustation vidéo

Skills4 digital-learning certifiant

3.2 – Vectorisation par contingence

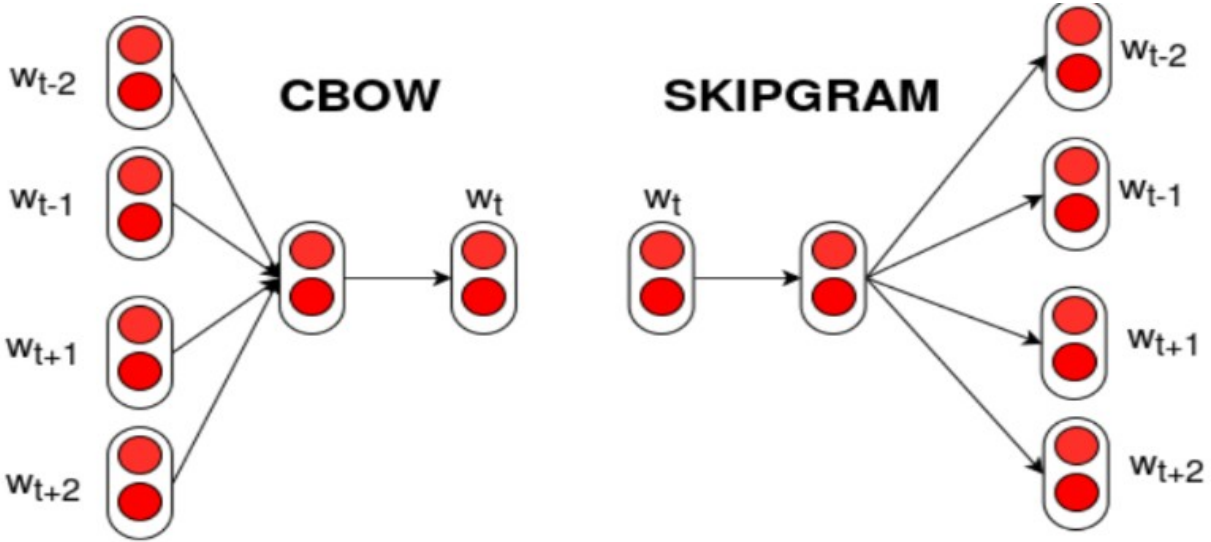
- Limites (PMI)
  - La taille des vecteurs de mots est toujours égale à celle du vocabulaire, ce qui rend la représentation inutilisable pour la plupart des problèmes
- Solution ?
  - Construire une représentation dont la taille est fixée en avance
  - Apprendre par l'apprentissage auto-supervisé une représentation du vocabulaire en fonction des mots environnants

Zone d'incrustation vidéo

skills4
digital-learning certifiant

3.2 – Vectorisation par apprentissage auto-supervisé

- **Word2vec / modèle CBOW et Skip-gram**
  - **Pour chaque phrase : choisir un mot cible, le modèle CBOW essaye de prédire ce mot en fonction des mots environnants ; Le modèle Skip-gram essaye de prédire les mots environnants en fonction du mot cible**



Zone d'incrustation vidéo

3.2 – Vectorisation par apprentissage auto-supervisé

- **Word2vec : MLP**
  - Entraîné pour de la classification
  - Afin de résoudre ce problème le modèle est obligé de construire une représentation interne du langage dans sa couche cachée
  - Une fois le modèle entraîné, on utilise la sortie de sa première couche cachée comme représentation vectorielle du mot
  - Nous contrôlons la taille de la couche cachée et donc la taille de l'embedding

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

3.2 – Vectorisation par apprentissage auto-supervisé

Word2vec

- CBOW est beaucoup plus rapide que le Skip-gram. Il sera adapté sur des grands datasets
- Skip-gram obtient de meilleurs embeddings, notamment sur les mots peu fréquents
- A chaque token corresponds un seul embedding donc la polysémie est impossible
- Les performances de ces modèles sont parmi les meilleurs de l'état de l'art pour représenter des tokens

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Word2vec : implémentation avec Gensim

```python
import gensim.downloader as api

# pre-trained skip-gram
model = api.load('word2vec-google-news-300')

print(f'word #10/{len(model.index_to_key)} is "{model.index_to_key[10]}"')
print(f'word vector for "was" is {model["was"]}')
```

```
word #10/3000000 is "was"
word vector for "was" is [ 2.60009766e-02 -1.89208984e-03  1.85546875e-01 -5.17578125e-02
  5.12695312e-03 -1.09863281e-01 -8.17871094e-03 -8.83789062e-02
  9.66796875e-02  4.83398438e-02  1.10473633e-02 -3.63281250e-01
  8.20312500e-02 -2.12402344e-02  1.58203125e-01  4.41894531e-02
 -1.17797852e-02  2.12890625e-01 -5.73730469e-02  5.66406250e-02
 -1.07421875e-01  1.85546875e-01  7.71484375e-02  1.44958496e-04
  1.52343750e-01 -6.54296875e-02 -1.52343750e-01  2.25585938e-01
  8.10546875e-02  8.88671875e-02  7.32421875e-02 -1.03515625e-01
 -6.68945312e-02  1.76757812e-01  2.12890625e-01  1.40625000e-01
 -3.41796875e-02  1.78222656e-02  5.95703125e-02  2.86102295e-04
  5.88378906e-02  9.27734375e-03  1.66992188e-01 -2.70080566e-03
  1.15722656e-01  1.04492188e-01  5.37109375e-02  1.85546875e-02
  1.06445312e-01  5.05371094e-02 -1.64794922e-02 -1.27929688e-01
  2.16796875e-01  5.15136719e-02  4.78515625e-02  1.52343750e-01
  1.71875000e-01  7.86132812e-02 -5.88378906e-02 -4.29687500e-02
```

digital-learning certifiant

Zone d'incrustation vidéo

↗ Word2vec : que peut on faire avec des embeddings ? (1/2)

```
pairs = [
    ('cherry', 'pear'),
    ('cherry', 'apple'),
    ('cherry', 'cereal'),
    ('cherry', 'cake'),
    ('cherry', 'communism'),
]

for w1, w2 in pairs:
    print(f'sim({w1}, {w2}) = {model.similarity(w1, w2)}')
```

```
sim(cherry, pear) = 0.5022730827331543
sim(cherry, apple) = 0.5369484424591064
sim(cherry, cereal) = 0.17614568769931793
sim(cherry, cake) = 0.3576621115207672
sim(cherry, communism) = -0.005840067286044359
```

```
print(model.most_similar('people', topn = 5))
```

```
[('peole', 0.6058608293533325), ('poeple', 0.5907129049301147), ('individuals'
94459581375122), ('peple', 0.5788743495941162)]
```

```
print(model.doesnt_match(['fire', 'water', 'land', 'sea', 'air', 'car']))
```

```
car
```

Zone d'incrustation vidéo

↗ Word2vec : que peut on faire avec des embeddings ? (2 / 2)

King – Man + Woman = Queen

```
model.most_similar(positive = ['woman', 'king'], negative = ['man'], topn = 20)
```

```
[('queen', 0.7118193507194519),
 ('monarch', 0.6189674139022827),
 ('princess', 0.5902431607246399),
 ('crown_prince', 0.549946129322052),
 ('prince', 0.5377321839332581),
 ('kings', 0.5236844420433044),
 ('Queen_Consort', 0.5235946178436279),
 ('queens', 0.518113374710083),
 ('sultan', 0.5098593831062317),
 ('monarchy', 0.5087411999702454),
 ('royal_palace', 0.5087166428565979),
 ('throne', 0.5005807876586914),
 ('royal', 0.4938204288482666),
 ('Princess_Sikhanyiso', 0.4936617314815521),
 ('ruler', 0.49092763662338257),
 ('empress', 0.4887814223766327),
 ('Prince_Paras', 0.4832944869995117),
 ('princes', 0.48108163475990295),
 ('King_Ahasuerus', 0.47894206643104553),
 ('very_pampered_McElhatton', 0.47636350989341736)]
```

Zone d'incrustation vidéo

3.2 – Vectorisation par apprentissage auto-supervisé

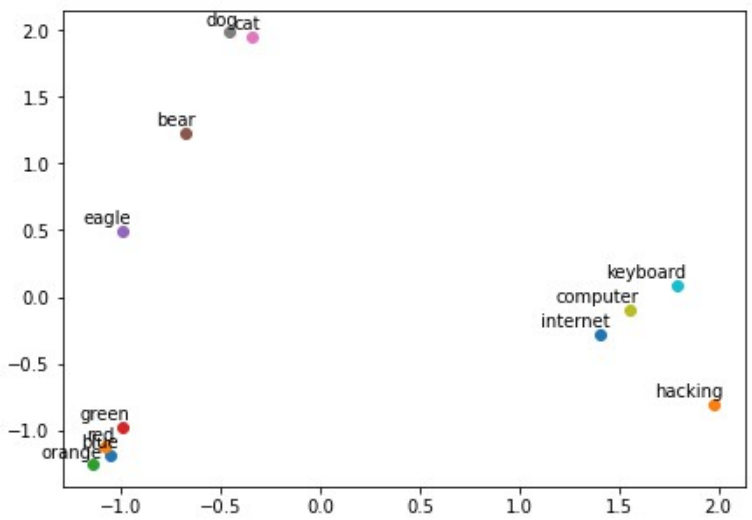
- Remarques sur les embeddings de Word2vec
  - L'embedding capture une part de la sémantique (King – Man + Woman = Queen)
  - La similarité est mesurée par l'angle entre les vecteurs de mots

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Visualiser les mots grâce l'ACP (PCA)

```
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA

word_list = [
    'blue', 'red', 'orange', 'green',
    'eagle', 'bear', 'cat', 'dog',
    'computer', 'keyboard', 'internet', 'hacking'
]
X     = model[word_list]
X_pca = PCA().fit_transform(X)[:,:2]

plt.figure(figsize = (7, 5))
for i in range(X_pca.shape[0]):
    x, y = X_pca[i]
    plt.scatter(x, y)
    plt.annotate(
        word_list[i],
        xy         = (x, y),
        xytext     = (5, 2),
        textcoords = 'offset points',
        ha         = 'right',
        va         = 'bottom'
    )
```



Zone d'incrustation vidéo

Partie 4 : Deep Learning pour NLP

Skills4All
digital-learning certifiant

2.3 – Deep Learning pour NLP

- Deep Learning Pour NLP
  - Rappel de Machine Learning
  - Perceptron multicouches (MLP)
  - CNN, RNN & LSTM
  - Mécanisme d'attention & Transformers

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Partie 4.1 : Rappels de Machine Learning et mesures de performances

Skills4All
digital-learning certifiant

Sommaire

- Rappels de Machine Learning et mesures de performance
  - Types et tâches de Machine Learning
  - Fonctions d'erreur, de distance et de similarité
  - Mesures de performance

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

4.1 – Rappels et mesures de performance

Type de Machine Learning

- **Apprentissage supervisé**

$$X; Y; Loss(model(x), y)$$

- **Apprentissage non-supervisé**

$$X; Similarity(x_1, x_2) || Distance(x_1, x_2)$$

- **Apprentissage par renforcement**

$$X; Cost(model(x))$$

- **Apprentissage antagoniste**

$$X_{real}; Generator() \rightarrow X_{fake}; Discriminator(x) \rightarrow \ is_x_fake\ ?(x)$$

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

4.1 – Rappels et mesures de performance

Tâche de Machine Learning

- **Classification**

$$Model(x) \rightarrow classe\ 1, classe\ 2 \ldots classe\ n \in \mathbb{N}$$

- **Régression**

$$Model(x) \rightarrow y \in \mathbb{R}$$

- **Clustering**

$$Model(X) \rightarrow cluster\ 1, cluster\ 2 \ldots cluster\ n \in \mathbb{P}(\mathbb{X})$$

- **Génération**

$$Model() \rightarrow x$$

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

4.1 – Rappels et mesures de performance

Fonctions d'erreur

- **Erreur quadratique (Squared loss)**

$$e(x, y, f) = \frac{1}{d}\sum_{i=1}^{d}(y_i - f(x))^2$$

- **Erreur absolue (Absolute loss)**

$$e(x, y, f) = \frac{1}{d}\sum_{i=1}^{d}|y_i - f(x)|$$

- **Entropie croisée (Cross-entropy)**

$$e(x, y, f) = -\frac{1}{c}\sum_{i=1}^{c}(y_i.log(f(x)))$$

- **Erreur m**

$$e(X, Y, f) = \frac{1}{n}\sum_{k=1}^{n} e(x_k, y_k, f)$$

Zone d'incrustation vidéo

digital-learning certifiant

4.1 – Rappels et mesures de performance

Fonctions de distance communes

- **Distance Euclidienne**

$$dist(a,b) = \sqrt{\sum_{i=1}^{k}(a_i - b_i)^2}$$

- **Distance de Manhattan**

$$dist(a,b) = \sum_{i=1}^{k}|a_i - b_i|$$

- **Distance de Minkowski**

$$dist(a,b) = \sqrt[n]{\sum_{i=1}^{k}(a_i - b_i)^n}$$

- **Toutes ces distances sont non pertinentes avec des vecteurs en haute dimensions**

Zone d'incrustation vidéo

digital-learning certifiant

4.1 – Rappels et mesures de performance

Fonctions de distance utiles pour le NLP

- Produit scalaire (dot product)

$$dist(a,b) = \vec{a}.\vec{b} = \|a\|.\|b\|.cos(a,b) = \sum_{i=1}^{k} a_i.b_i$$

- Deux vecteurs qui pointent dans la même direction ont un produit scalaire positif
- Deux vecteurs perpendiculaire ont un produit scalaire de zéro
- Deux vecteurs qui pointent dans une direction opposée ont un produit scalaire négatif

Skills4
digital-learning certifiant

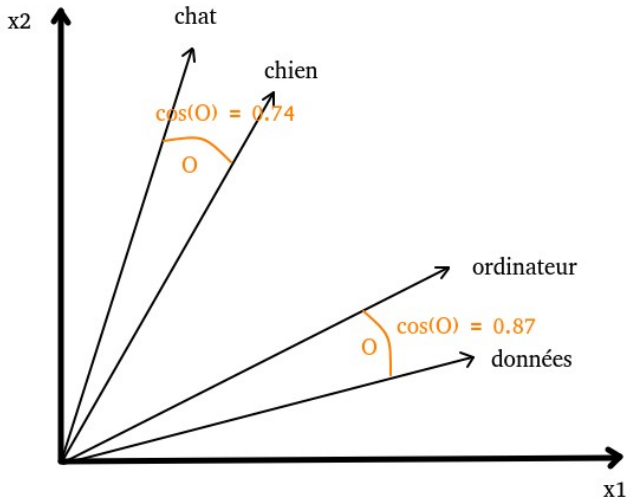
Zone d'incrustation vidéo

4.1 – Rappels et mesures de performance

Similarité cosinus (cosine similarity)

$$sim(a,b) = \frac{\vec{a}.\vec{b}}{\|a\|.\|b\|} = cos(a,b)$$

- **Mesure de similarité entre vecteurs de mots**



Zone d'incrustation vidéo

Mesures de performance pour la classification

		Prédiction du modèle		
		p	n	total
Observations	p′	True positive	False negative	P′
	n′	False positive	True negative	N′
	total	P	N	

digital-learning certifiant

Zone d'incrustation vidéo

4.1 – Rappels et mesures de performance

Le taux de vrais positifs, rappel ou sensibilité (true positive rate, recall, sensibility). C'est la capacité du classifieur à bien détecter la classe positive.

$$TP_{rate} = \frac{TP}{TP + FN} = \frac{TP}{P'}$$

Le taux de vrais négatifs, spécificité ou sélectivité (true negative rate, specificity or selectivity). C'est la capacité du classifieur à bien détecter la classe négative.

$$TN_{rate} = \frac{TN}{TN + FP} = \frac{TN}{N'}$$

Le taux de faux positifs, probablilité de fausse alarme (false positive rate, probability of false alarme). C'est la propension du classifieur à attribuer par erreur la classe positive à un échantillon.

$$FP_{rate} = \frac{FP}{TN + FP} = \frac{FP}{P'}$$

Le taux de faux négatifs (false negative rate, miss rate). C'est la propension du classifieur à attribuer par erreur la classe négtive à un échantillon.

$$FN_{rate} = \frac{FN}{TN + FN} = \frac{FN}{N}$$

Zone d'incrustation vidéo

4.1 – Rappels et mesures de performance

Mesures de performance pour la classification

- **Exactitude (accuracy)**

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Exactitude équilibrée (balanced accuracy)**

$$balanced\ accuracy = \frac{1}{2}.\left(\frac{TP}{TP + FP} + \frac{TN}{TN + FN}\right)$$

- **F1, moyenne harmonique (F1, harmonic mean)**

$$F_1 = \frac{2.TP}{2.TP + FP + FN}$$

Zone d'incrustation vidéo

4.1 – Rappels et mesures de performance

Mesure de performance pour la régression

- Coefficient de détermination

$$R^2(X,Y,f) = 1 - \frac{\sum_{k=1}^{n}(y_k - f(x_k))^2}{\sum_{k=1}^{n}(y_k - E(Y))^2}$$

- Si R2 = 0 : le modèle est équivalent à la moyenne des observations
- Si R2 > 0 : le modèle est meilleur que la moyenne des observations
- Si R2 < 0 : le modèle est vraiment mauvais

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

4.1 – Rappels et mesures de performance

- Mesures de performances pour le NLP
  - Aucune des mesures précédentes n'est utile pour savoir si un texte généré est bon ou pas
  - Aucune des mesures précédentes n'est utile pour savoir si un modèle est en capacité de comprendre un texte
  - La seule manière de procéder est de préparer Q/A sur des textes et d'utiliser des mesures de classifications pour évaluer les performances
  - Le modèle de NLP doit être ré-entrainé (finetuned) pour chacune des tâches de l'ensemble de test

Zone d'incrustation vidéo

Skills4 digital learning certificate

4.1 – Rappels et mesures de performance

Mesures de classification pour le NLP

- GLUE, Super-GLUE, SQuAD, BLEU, ROGUE, MS MACRO, XTREME …

Dataset	Umbrella Term(s)	Data example	Metric
CoLA	Acceptability	They made him angry. 1 = acceptable This building is than that one. 0 = unacceptable	Matthews
SST-2	Sentiment	the movie as a whole is cheap junk and an insult to their death-defying efforts = .11111 the movie is funny , smart , visually inventive , and most of all , alive . = .93056 has both = .5	Accuracy
MRPC	Paraphrase	"A New Castle County woman has become the first Delaware patient to contract the West Nile virus this year , the state 's Department of Health reported ." "A 62-year-old West Babylon man has contracted the West Nile virus , the first human case in Suffolk County this year , according to the county health department .'" label = 0 "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ." "The island reported another 35 probable cases yesterday , taking its total to 418 ." label = 1	Accuracy / F1

digital-learning certifiant

Zone d'incrustation vidéo

4.1 – Rappels et mesures de performance

- Biais de la prise de mesure
  - Les mesures doivent être prise sur des échantillons différents de l'entraînement => partition train / test de la base de données

Zone d'incrustation vidéo

4.1 – Rappels et mesures de performance

Biais de la prise de mesure

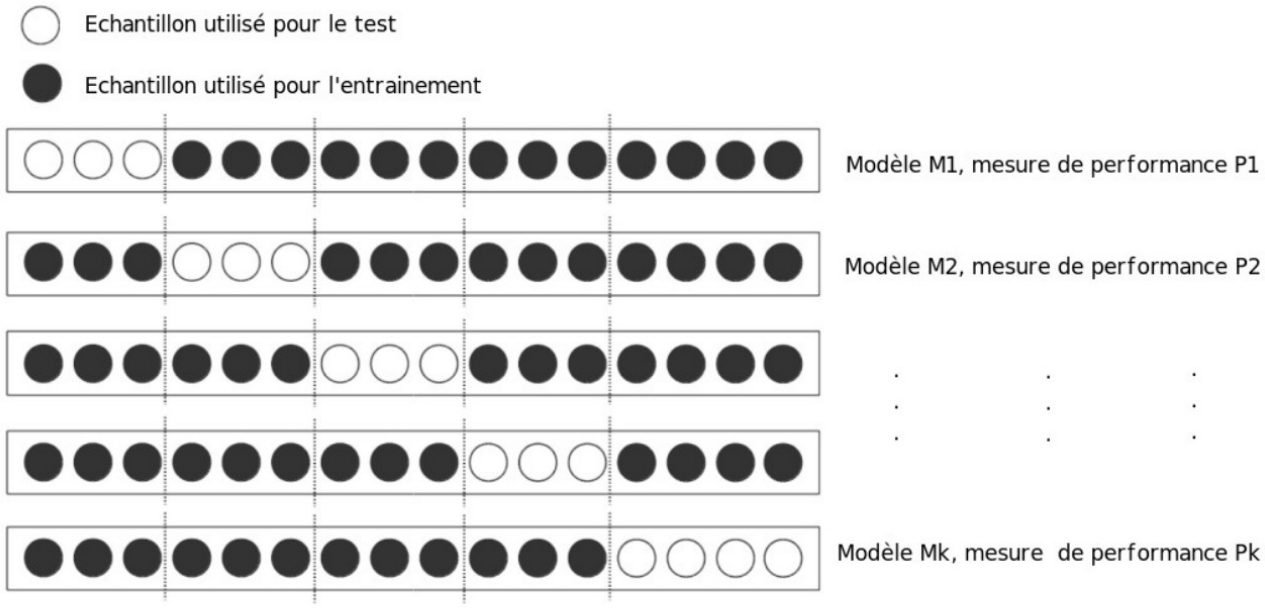
- La partition de la base de données peut être biaisée => validation croisée

```
accuracy = 0.0
while(accuracy < 0.99):
  model = MySuperModel()
  random_init(model)
  X_train, X_test, y_train, y_test = random_split(X, y)
  train(model, X_train, y_train)
  accuracy = test(model, X_test, y_test)

print(f'My super model {model} has an incredible accuracy of {accuracy} (>99\%!!!)')
print('It will convolutionize the world!')
```

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Validation croisée



Performance de la méthode = Moyenne des P0 ... Pk

digital-learning certifiant

Zone d'incrustation vidéo

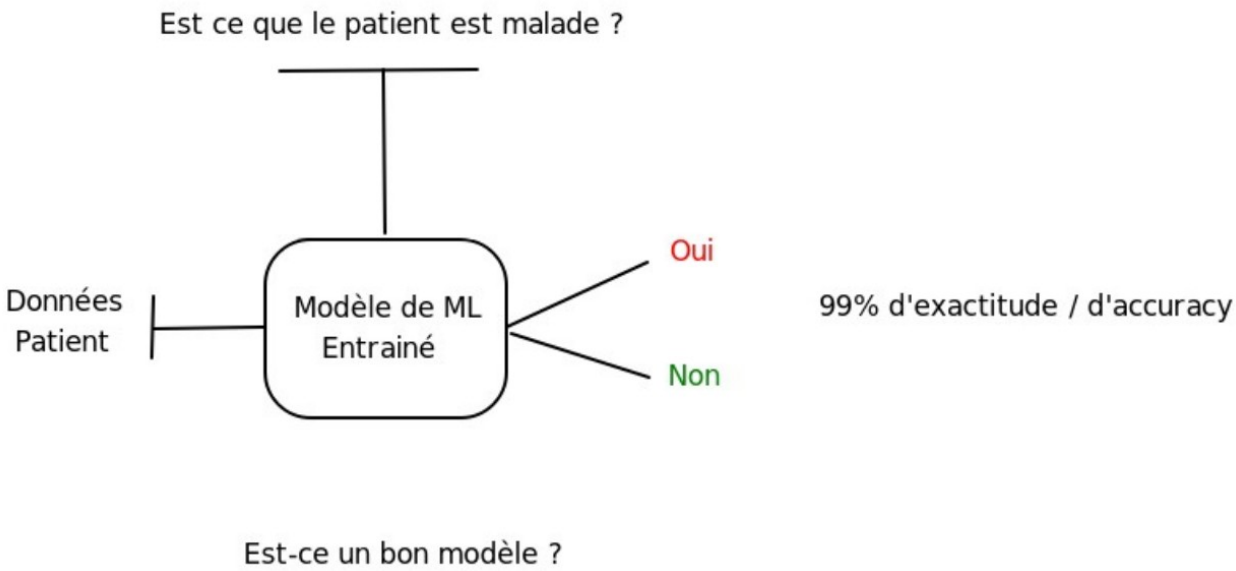
4.1 – Rappels et mesures de performance

- Biais spécifiques au NLP
  - Un modèle de NLP va reproduire les biais sémantiques qui se trouvent dans la base de données.
    - Traduction de « He is a nurse. She is doctor. » de l'anglais vers en l'hongrois puis en anglais : « She is a nurse. He is a doctor »
    - « Black » - « Criminal » = « Caucasian » - « Police »
    - « Christianity » - « Lawful » = « Islamic » - « Terrorist »
  - Solutions => Harmoniser le nombre d'échantillons contenant une certaine propriété | débiaisement des word vectors

Zone d'incrustation vidéo

Skills-4

Mauvais usage des mesures de performance (Etude de cas 1 / 3)



Zone d'incrustation vidéo

digital-learning certifiant

Mauvais usage des mesures de performance (Etude de cas 2 / 3)

Le détecteur est utilisé sur toute la population française (67 millions)

Hypothèse : la maladie a 0,1 % de chances d'apparaître

Observations	Prédictions justes	Prédictions erronées
66 933 000 personnes saines	66 263 670 personnes sont prédites saines	669 330 personnes sont prédites malades (1%)
67 000 personnes malades	66 330 personnes sont prédites malades	670 personnes sont prédites saines (1%)

Zone d'incrustation vidéo

digital-learning certifiant

4.1 – Rappels et mesures de performance

Mauvais usage des mesures de performance (Etude de cas 3 / 3)

- Résultats : 669 330 saines sont envoyées à l'hôpital et traités pour une maladie qu'ils n'ont pas => catastrophe sanitaire
- Conclusions provisoires
  - L'accuracy n'est pas une mesure suffisante
  - Il faut caractériser les erreurs sur chaque classe
  - Le contexte d'application est de première importance pour juger des performances d'un modèle

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

Partie 4.2 : Perceptron multicouches (MLP)

Skills4All
digital-learning certifiant

4.2 – Perceptron multicouches (MLP)

Sommaire

- Le neurone formel
- Perceptron multi-couches
- Algorithme d'apprentissage : descente de gradients
- Implémentation

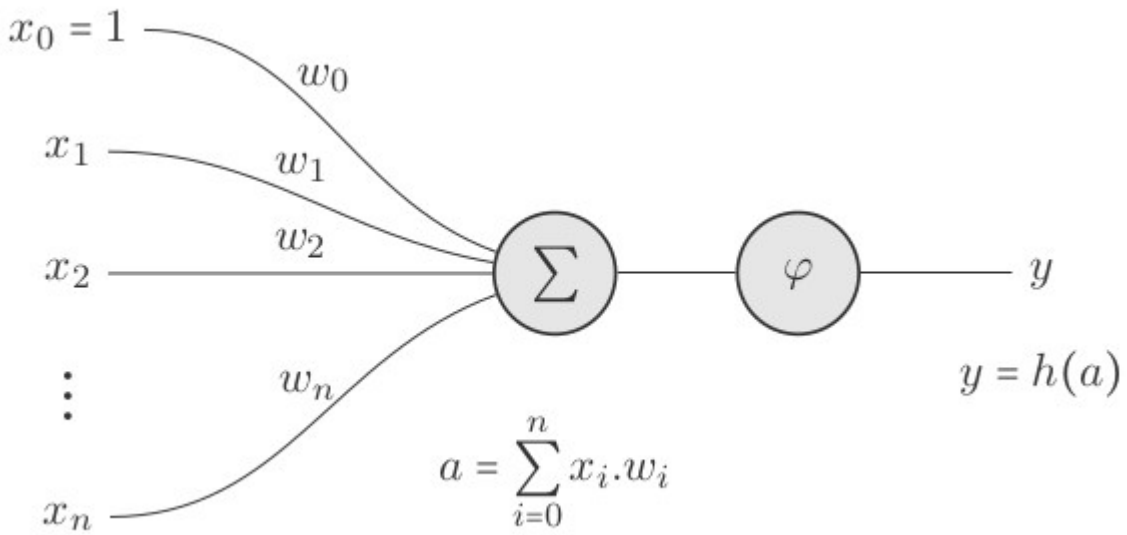
Zone d'incrustation vidéo

Skills4
digital-learning certifiant

4.2 – Perceptron multicouches (MLP)

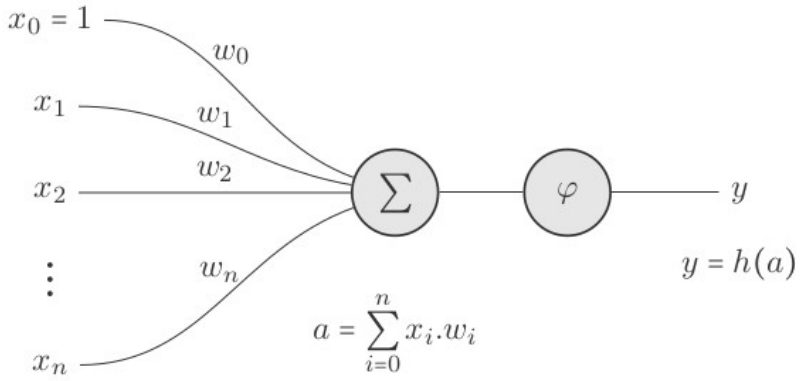
Neurone formel

- Vocabulaire : a → niveau d'activation, W → poids, Phi → fonction d'activation, y → sortie du neurone



Zone d'incrustation vidéo

Comment entrainer un neurone ?



Until the *learning* is *over*, do

For each couple $(\vec{x}, y_{ex})$, in the learning dataset, do

$a = \sum_{i=0}^{n} x_i.w_i$

$y = h(a)$

$e(t) = y_{ex} - y$

For each $i \in [0, n]$, do

$w_i(t+1) = w_i(t) + \mu.x_i.e(t)$

End for

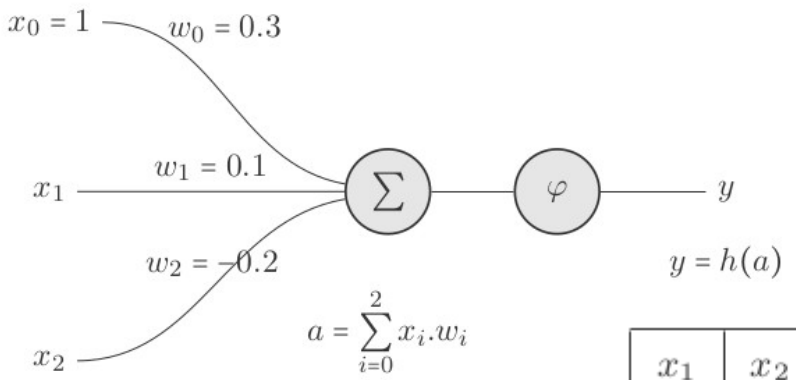
learning is *over* if a *stopping condition* have been *reached*

End for

End until

Zone d'incrustation vidéo

Apprentissage de la porte logique ET (1 / 2)



x_1	x_2	y_{ex}
0	0	0
1	0	0
0	1	0
1	1	1

Où :

- $w_0 = 0.3$, $w_1 = 0.1$, $w_2 = -0.2$
- $h(X) = \begin{cases} 1 & \text{if } X > 0 \\ 0 & \text{else} \end{cases}$
- $\mu = 0.5$
- L'apprentissage est arrêté quand les valeurs des poids convergent, c'est à dire quand un passage sur l'ensemble des échantillons de la base de données ne changent pas les poids.

Zone d'incrustation vidéo

Apprentissage de la porte logique ET (2 / 2)

```python
import numpy as np

def h(x):
  if x > 0:
    return 1
  else:
    return 0

def neuron_train(X, Y, w, μ):
  it      = 0
  is_over = False
  while is_over is False:
    is_over = True

    for x, y_ex in zip(X, Y):
      a = np.inner(w, x) # vector product
      y = h(a)
      e = y_ex - y

      w = w + μ * e * x

      if e != 0:
        is_over = False

    it += 1

  return w, it

w = np.array([0.3, 0.1, -0.2])
X = np.array([ # we add 1 to represent the bias
  [1, 0, 0],
  [1, 1, 0],
  [1, 0, 1],
  [1, 1, 1]
])
y = np.array([0, 0, 0, 1])
μ = 0.5

w, it = neuron_train(X, y, w, μ)
print(w, it)
```

```
[-1.2  0.6  0.8] 6
```

```python
def neuron_predict(X, w):
  Y = list()
  for x in X:
    a = np.inner(w, x)
    y = h(a)
    Y.append(y)

  return Y

print(neuron_predict(X, w))
```

```
[0, 0, 0, 1]
```

ls4
ning certifiant

Zone d'incrustation vidéo

4.2 – Perceptron multicouches (MLP)

- Exercice : apprentissage de la porte logique OU
  - Ré-implémenter le code précédent avec la porte logique OU

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

4.2 – Perceptron multicouches (MLP)

↗️ Exercice : apprentissage de la porte logique OU

```
w = np.array([0.3, 0.1, -0.2])
y = np.array([1, 1, 1, 0])

w, it = neuron_train(X, y, w, µ)
print(w, it)
print(neuron_predict(X, w))

> [ 1.3 -0.4 -1.2] 6
> [1, 1, 1, 0]
```

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

Apprentissage de la porte logique XOR (1 / 2)

x_1	x_2	y_{ex}
0	0	0
1	0	1
0	1	1
1	1	0

Entrée [*]:
```
w = np.array([0.3, 0.1, -0.2])
y = np.array([0, 1, 1, 0])
w, it = neuron_train(X, y, w, μ)

print(w, it)
print(neuron_predict(X, w))
```

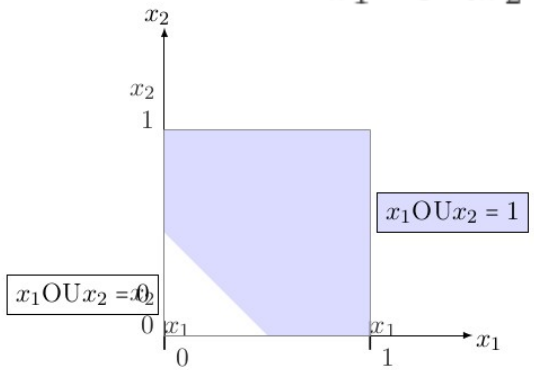
SKILS4
digital-learning certifiant

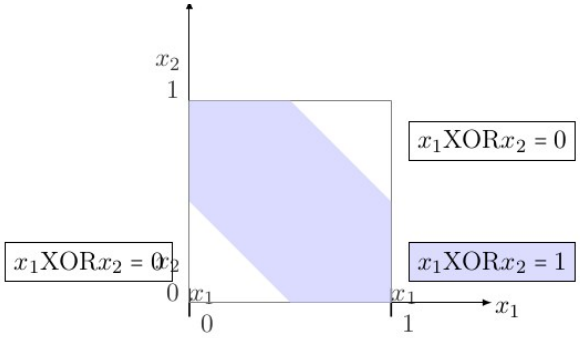
Zone d'incrustation vidéo

Apprentissage de la porte logique XOR (2 / 2)

- OU est linéairement séparable, XOR ne l'est pas

$$x_1 \mathrm{XOR} x_2 = (x_1 \mathrm{OR} x_2) \mathrm{AND} (x_1 \mathrm{NAND} x_2)$$

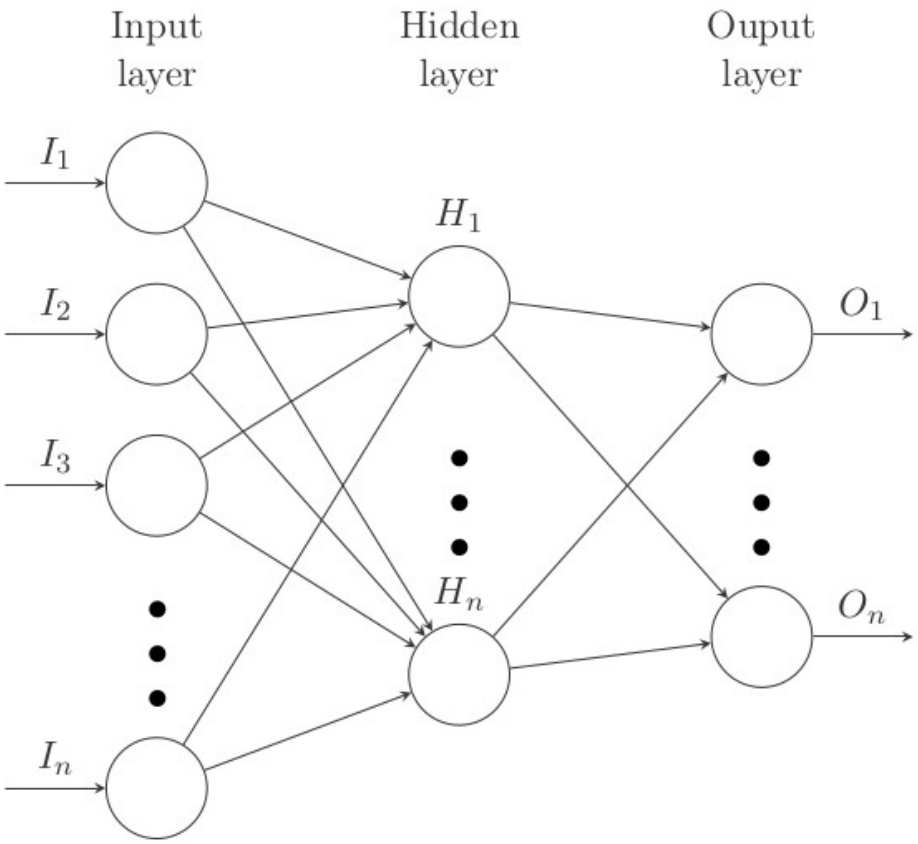




- Solution : utiliser au moins 3 neurones
- Généralisation : connecter un nombre arbitraire de neurones => Perceptron multicouches

Zone d'incrustation vidéo

Perceptron multicouches (MLP)

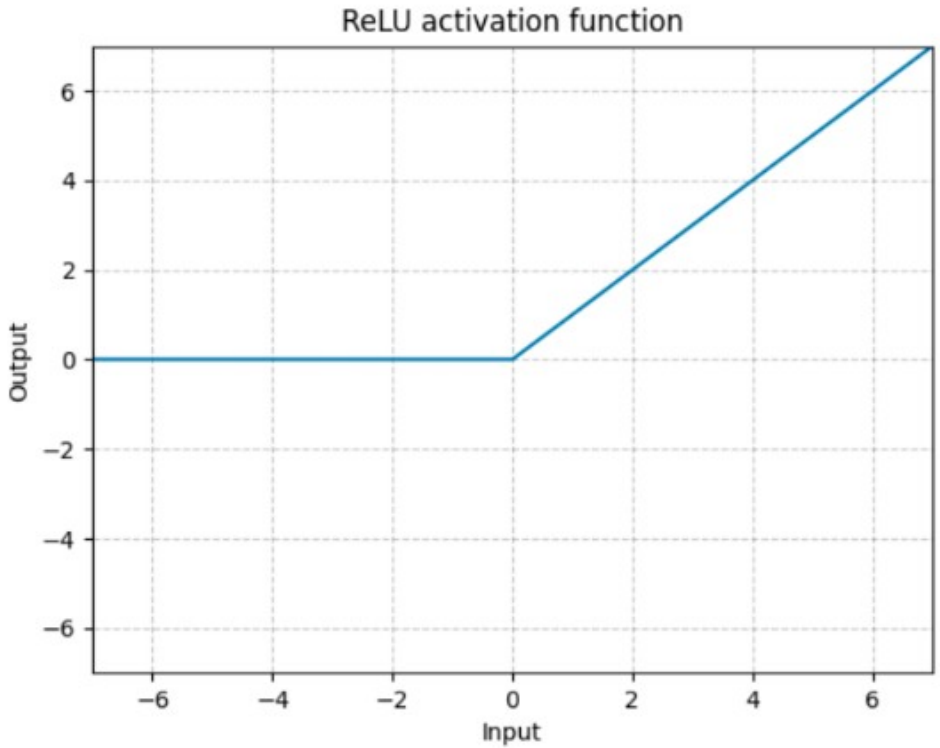


Zone d'incrustation vidéo

4.2 – Perceptron multicouches (MLP)

Fonction de transfert : torch.nn.ReLU

$$h(x) = max(0, x)$$

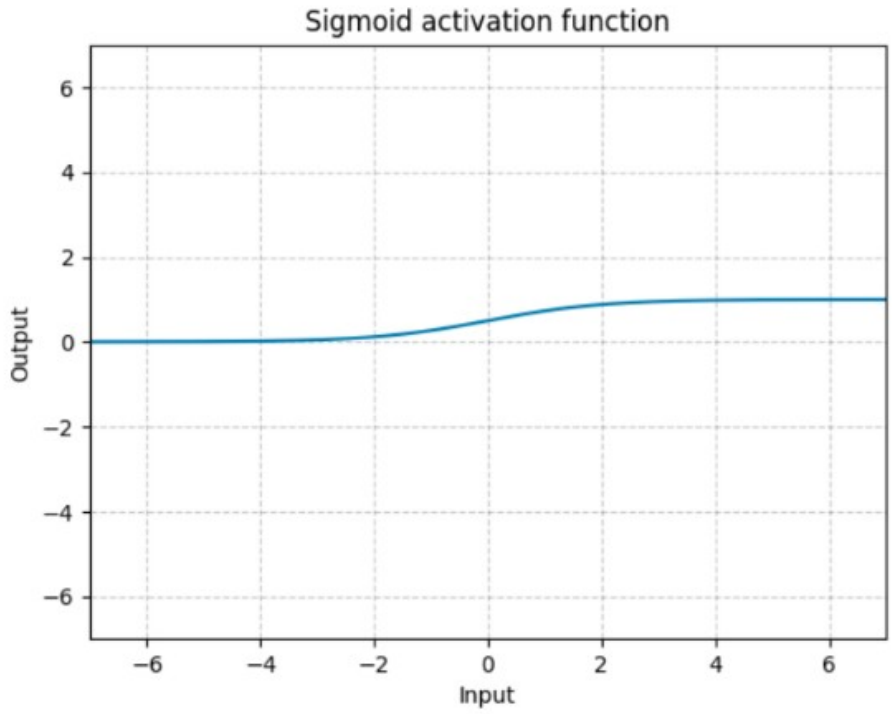


Zone d'incrustation vidéo

4.2 – Perceptron multicouches (MLP)

Fonctions de transfert : torch.nn.Sigmoid

$$h(x) = \frac{1}{1 - e^{-x}}$$

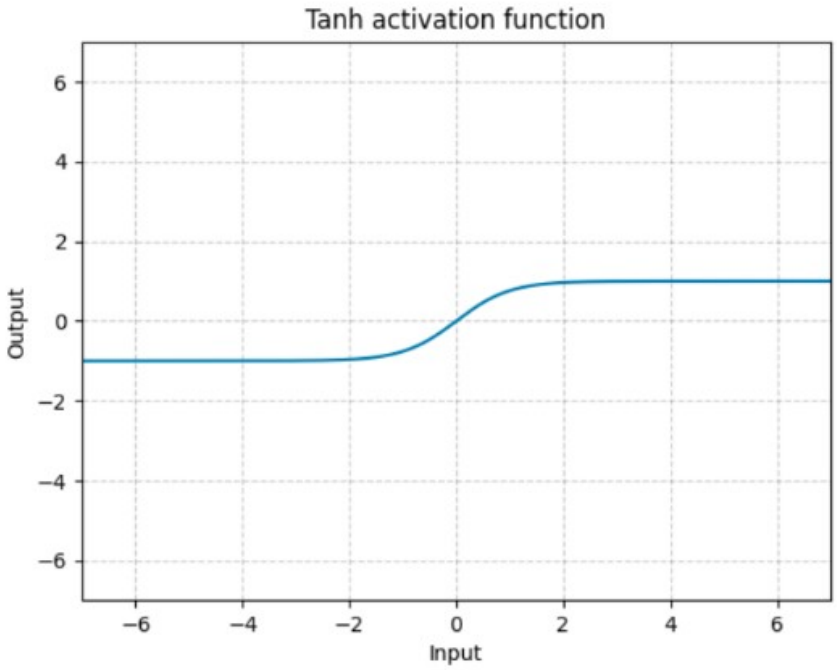


Zone d'incrustation vidéo

4.2 – Perceptron multicouches (MLP)

Fonctions de transfert : torch.nn.Tanh

$$h(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}$$



Zone d'incrustation vidéo

4.2 – Perceptron multicouches (MLP)

Fonction de transfert : softmax & cross-entropy

torch.nn.Softmax [7]

$$h(x) = \{\frac{e^{x_i}}{\sum_{\forall j} e^{x_j}}, \ \forall i\}$$

torch.nn.LogSoftmax [8]

$$h(x) = \{log(\frac{e^{x_i}}{\sum_{\forall j} e^{x_j}}), \ \forall i\}$$

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

Fonctions d'erreur : MSE

```
# problème de régression
y = torch.tensor([ # y -> ce qui est produit par le modèle
  [-0.23, 0.44, 1.2], # échantillon 1
  [-0.36, 0.73, 0.9], # échantillon 2
])
y_ex = torch.tensor([ # y_ex -> ce que l'on veut obtenir
  [-0.11, 0.16, 1.5], # échantillon 1
  [-0.63, 0.42, 0.8], # échantillon 2
])
loss_funct = nn.MSELoss() # erreur moyenne du lot
loss       = loss_funct(y, y_ex)
print(loss)

# problème de classification (3 classes)
y = torch.tensor([
  [0.63, 0.12, 0.25], # le modèle prédit la classe 1
  [0.004, 0.5, 0.495],# le modèle prédit la classe 2
])
y_ex = torch.tensor([
  [1.0, 0.0, 0.0], # la vrai classe est la classe 1
  [0.0, 0.0, 1.0], # la vrai classe est la classe 3
])
loss = loss_funct(y, y_ex)
print(loss)

> tensor(0.0603)
> tensor(0.1198)
```

torch.nn.MSELoss[9]

$$e(x,y) = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - y_i)^2$$

Zone d'incrustation vidéo

Fonctions d'erreur : erreur absolue et cross-entropy

torch.nn.L1Loss

$$e(x,y) = \frac{1}{n} \cdot \sum_{i=1}^{n} |x_i - y_i|$$

torch.nn.CrossEntropyLoss

$$e(x,y) = -log(\frac{e^{x_y}}{\sum_{\forall i} e^{x_i}})$$

```
# problème de classification (3 classes)

y = torch.tensor([ # les sorties n'ont pas à être normalisées
  [7.24, -0.2, 3.25], # le modèle prédit la classe 1
  [1.1, 5.0, 4.95],   # le modèle prédit la classe 2
])
y_ex = torch.tensor([ # on donne directement l'indice de la classe
  0, # la vrai classe est la classe 1
  2, # la vrai classe est la classe 3
])
loss_funct = nn.CrossEntropyLoss()
loss       = loss_funct(y, y_ex)
print(loss)

> tensor(0.3738)
```

Zone d'incrustation vidéo

Fonctions d'erreur : Null Log Likelihood Loss

torch.nn.NLLLoss

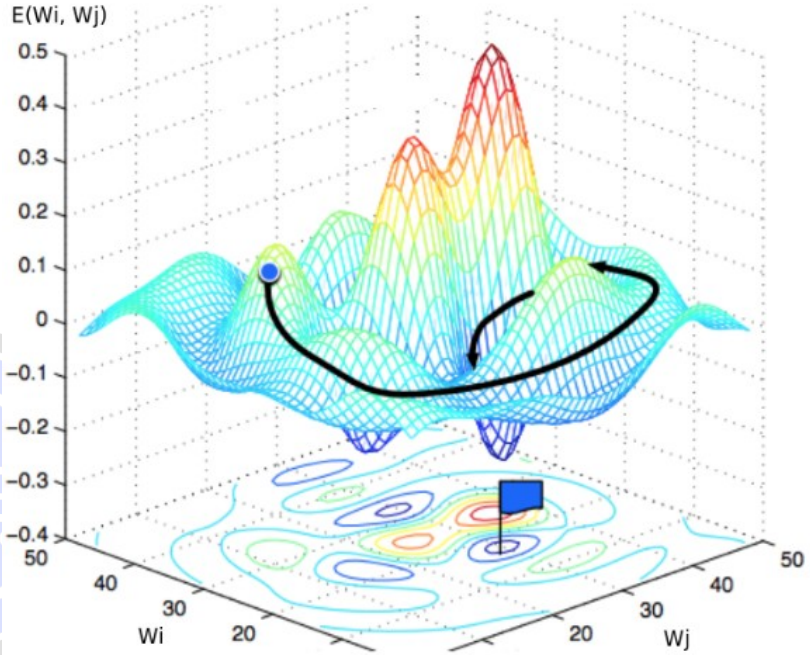
$$e(x, y) = -x_y$$

```
# problème de classification (3 classes)

y = torch.tensor([ # les sorties n'ont pas à être normalisées
  [7.24, -0.2, 3.25], # le modèle prédit la classe 1
  [1.1, 5.0, 4.95],   # le modèle prédit la classe 2
])
y_ex = torch.tensor([ # on donne directement l'indice de la classe
  0, # la vrai classe est la classe 1
  2, # la vrai classe est la classe 3
])

output_transfer = nn.LogSoftmax(dim = 1)
loss_funct      = nn.NLLLoss()
y_lsmax         = output_transfer(y)
loss            = loss_funct(y_lsmax, y_ex)

print(loss)
print(torch.exp(y_lsmax)) # probabilités

> tensor(0.3738)
> tensor([[9.8127e-01, 5.7629e-04, 1.8153e-02],
          [1.0267e-02, 5.0724e-01, 4.8250e-01]])
```

Zone d'incrustation vidéo

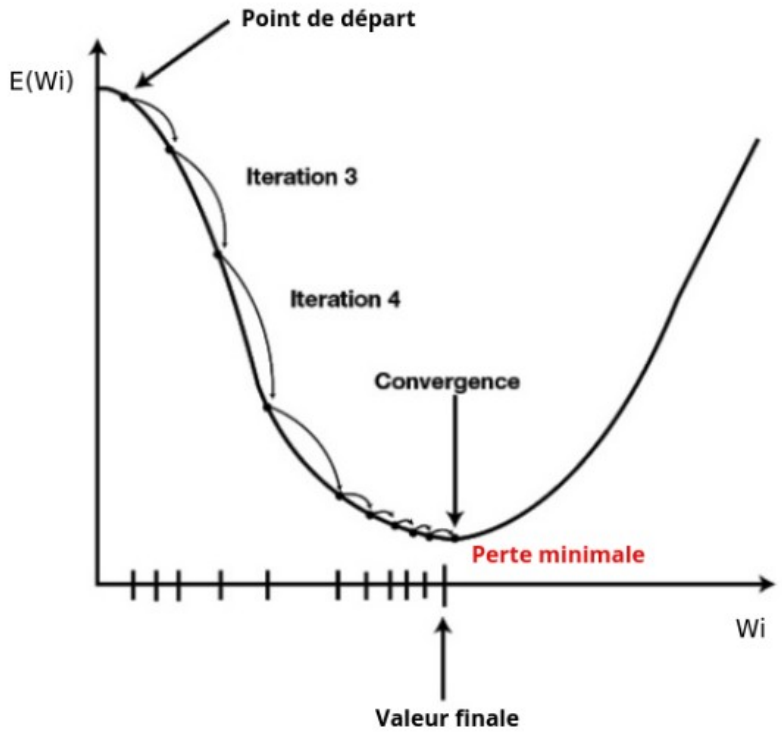
Algorithme d'apprentissage : descente de gradients

Objectif de la descente de gradient



Zone d'incrustation vidéo

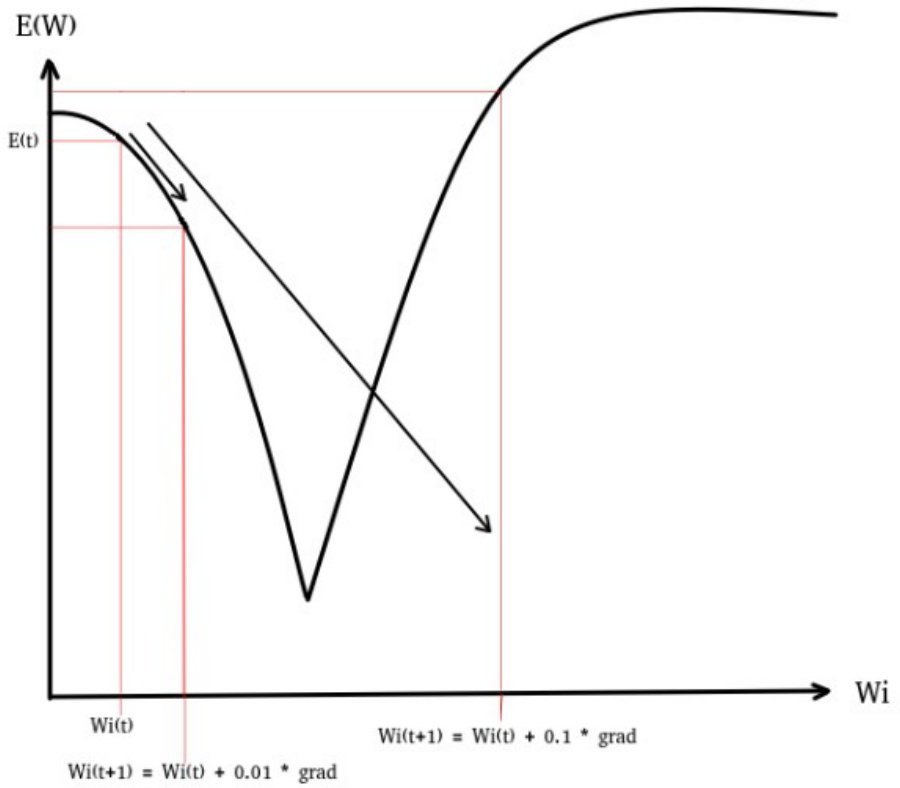
Descente de gradients : convergence



Zone d'incrustation vidéo

4.2 – Perceptron multicouches (MLP)

↗ Descente de gradients : coefficient d'apprentissage



Zone d'incrustation vidéo

SGD : comment obtenir la formule de modification de poids ?

$$E(t) = E(\overrightarrow{w(t)})$$

$$E(t+1) = E(\overrightarrow{w(t)} + \overrightarrow{\Delta w(t)})$$

Le *Théorème de Taylor-Young* à l'ordre 1 nous donne une approximation de $E(t+1)$ autour de $\overrightarrow{w(t)}$:

$$E(t+1) = E(\overrightarrow{w(t)}) + \overrightarrow{\nabla E(w(t))}^T.\overrightarrow{\Delta w(t)} + o(||\overrightarrow{\Delta w(t)}||)$$

En choisissant la direction $\overrightarrow{\Delta w(t)} = -\mu.\overrightarrow{\nabla E(w(t))}, \mu > 0$, notre but est de construire $E(t+1)$ tel que $E(t+1) < E(t)$. Nous obtenons :

$$E(t+1) \approx E(\overrightarrow{w(t)}) - \mu.||\overrightarrow{\nabla E(w(t))}||^2$$

Comme $\mu.||\overrightarrow{\nabla E(w(t))}||^2 > 0$, en choisissant cette direction cela implique que $E(t+1) < E(t)$. Maintenant, nous devons évaluer chaque terme :

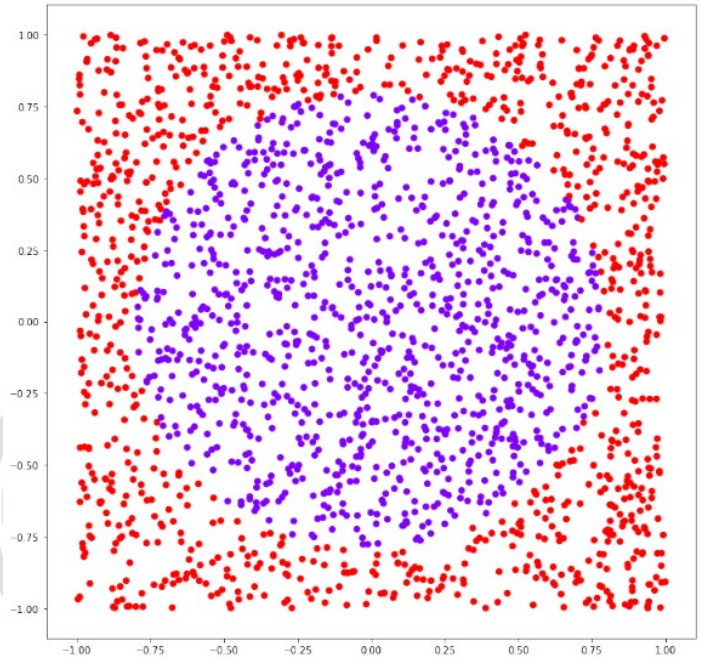
$$\Delta w_{ij}^k = -\mu.\frac{\partial E(t)}{\partial w_{ij}^k}$$

Zone d'incrustation vidéo

4.2 – Perceptron multicouches (MLP)

Implémentation d'un MLP (1 / 2)

- **Objectif : entrainer un MLP à différencier les points à l'intérieur du cercle des points à l'extérieur**




```python
def generate_dataset(n_samples)
```

```python
class MLP(nn.Module):
```

```python
def train(
  model, loss_func, optimizer, train_dataset,
  test_dataset, epochs, batch_size
):
```

```python
def evaluate(model, dataset, batch_size):
```

tifiant

Zone d'incrustation vidéo

Implémentation d'un MLP (2 / 2)

```
n_samples     = 3000
epochs        = 300
batch_size    = 32
learning_rate = 1e-3

train_dataset = generate_dataset(n_samples)
test_dataset  = generate_dataset(n_samples)
model         = MLP()
loss_func     = nn.NLLLoss()
optimizer     = optim.SGD(
  params = model.parameters(),
  lr     = learning_rate
)
train(
  model, loss_func, optimizer, train_dataset,
  test_dataset, epochs, batch_size
)
```

```
250 -> 96.600% train accuracy 250 -> 96.200% test accuracy
260 -> 96.700% train accuracy 260 -> 96.300% test accuracy
270 -> 96.900% train accuracy 270 -> 96.433% test accuracy
280 -> 96.900% train accuracy 280 -> 96.500% test accuracy
290 -> 97.000% train accuracy 290 -> 96.600% test accuracy
```

Zone d'incrustation vidéo

4.2 – Perceptron multicouches (MLP)

Exercice : détection de la langue d'un texte

- Détecter la langue de reviews Amazon
- 10 000 samples
- 6 langues possibles
- ~3000 caractères possibles
- Features : fréquence d'apparition des caractères

```
# construction du vocabulaire
for _, row in samples:
    lang   = row['language']
    review = row['review_body']

    if lang not in langs:
        langs[lang]  = langs_index
        langs_index += 1

    for c in review:
        if c not in vocab:
            vocab[c]     = vocab_index
            vocab_index += 1
```

```
# construction du dataset
X           = list()
y           = list()
feature_len = len(vocab)
label_len   = len(langs)

for _, row in samples:
    lang   = row['language']
    review = row['review_body']
    x      = [0.0] * feature_len

    for c in review:
        x[vocab[c]] += 1.0

    for i, _ in enumerate(x):
        x[i] /= len(review)

    X.append(x)
    y.append(langs[lang])
```

Zone d'incrustation vidéo

Exercice corrigé

```python
class MLP(nn.Module):
  def __init__(self, input_len, output_len):
    super(MLP, self).__init__()
    self.hidden_layer_1    = nn.Linear(input_len, 64)
    self.hidden_layer_2    = nn.Linear(64, 64)
    self.output_layer      = nn.Linear(64, output_len)
    self.hidden_transfer_1 = nn.ReLU()
    self.hidden_transfer_2 = nn.ReLU()
    self.output_transfer   = nn.LogSoftmax(dim = 1)

  def forward(self, x):
    x = self.hidden_layer_1(x)
    x = self.hidden_transfer_1(x)
    x = self.hidden_layer_2(x)
    x = self.hidden_transfer_2(x)
    x = self.output_layer(x)
    x = self.output_transfer(x)

    return x
```

```python
epochs        = 1000
batch_size    = 32
learning_rate = 1e-3

model         = MLP(feature_len, label_len)
loss_func     = nn.NLLLoss()
optimizer     = optim.SGD(
  params = model.parameters(),
  lr     = learning_rate
)
train(
  model, loss_func, optimizer, train_dataset,
  test_dataset, epochs, batch_size
)
```

```
900 -> 86.044% train accuracy 900 -> 85.000% test accuracy
910 -> 86.133% train accuracy 910 -> 85.200% test accuracy
920 -> 86.244% train accuracy 920 -> 85.000% test accuracy
930 -> 86.400% train accuracy 930 -> 85.100% test accuracy
940 -> 86.522% train accuracy 940 -> 85.200% test accuracy
950 -> 86.589% train accuracy 950 -> 85.400% test accuracy
960 -> 86.667% train accuracy 960 -> 85.400% test accuracy
970 -> 86.789% train accuracy 970 -> 85.400% test accuracy
980 -> 86.878% train accuracy 980 -> 85.500% test accuracy
990 -> 86.833% train accuracy 990 -> 85.500% test accuracy
```

Zone d'incrustation vidéo

Partie 4.3 : CNN, RNN, LSTM

Skills4All
digital-learning certifiant

4.3 – CNN, RNN, LSTM

Sommaire

- Réseau de neurones convolutif (CNN)
- Réseau de neurones récurrent (RNN)
- Réseau de neurones à mémoire à court-terme étendue (LSTM)

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Réseau de neurones convolutifs (CNN)

- Point de départ : le MLP n'est pas adapté pour traiter des séquences

```
s_1 = np.array([0, 1, 2, 4, 2, 0, 0, 1, 1, 0])
s_2 = np.array([0, 0, 3, 0, 2, 4, 2, 0, 1, 0])
s_3 = np.array([0, 1, 0, 3, 1, 2, 0, 4, 0, 0])
```

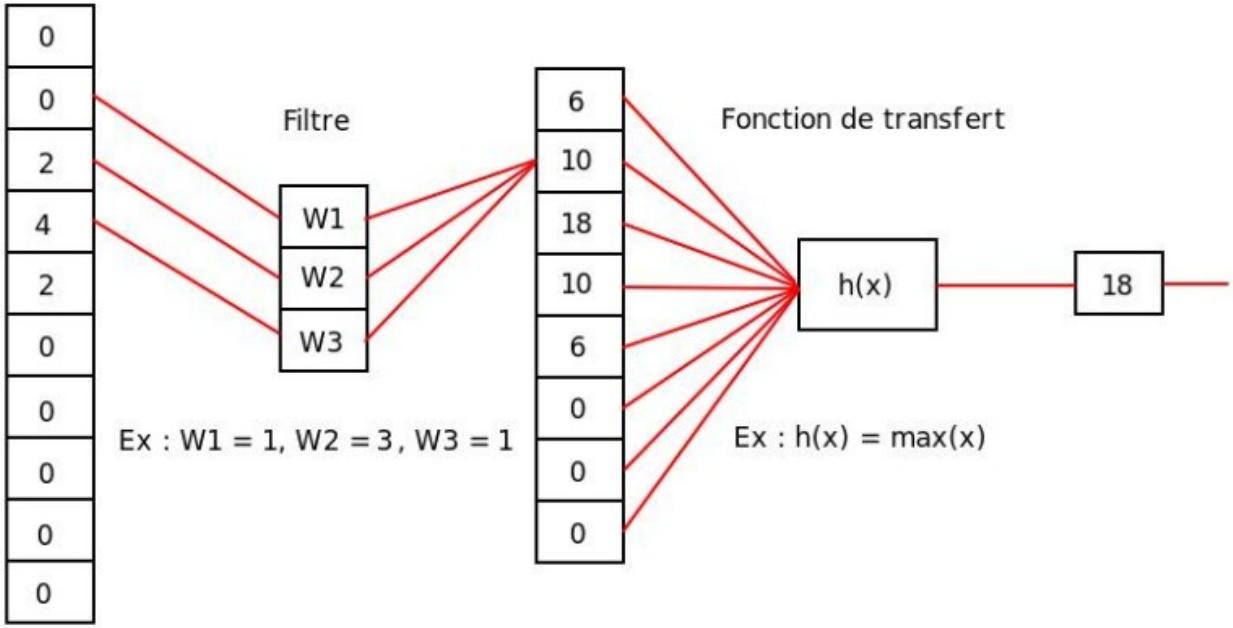
```
def h(x):
  if x == 6:
    return 1
  else:
    return 0

w = np.array([0, 0, 1, 1, 1, 0, 0, 0, 0, 0])
a = w.inner(s_1)
print(f'Valeur de sortie du neurone : {h(a)}')

> Valeur de sortie du neurone : 1
```

```
a = w.inner(s_2)
print(f'Valeur de sortie du neurone : {h(a)}')

> Valeur de sortie du neurone : 0
```

Zone d'incrustation vidéo

Réseau de neurones convolutif

- Filtre à convolution à une dimension



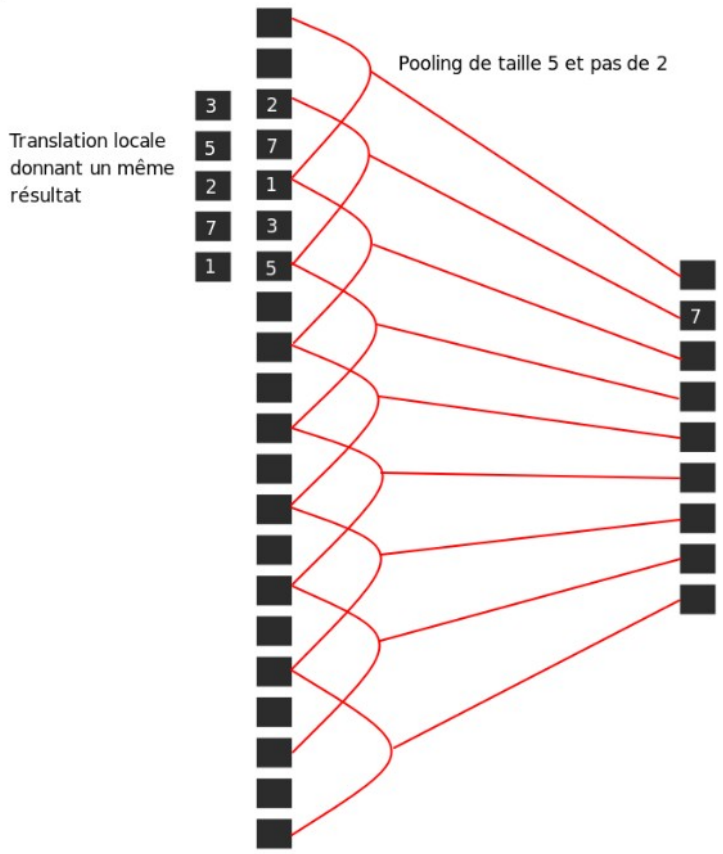
Zone d'incrustation vidéo

- Qualités des filtres à convolution avec pooling
  - Invariance locale par translation
  - Un filtre sera entrainé à extraire les caractéristiques importantes => fin du feature engineering
  - Machine Learning => Deep Learning

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

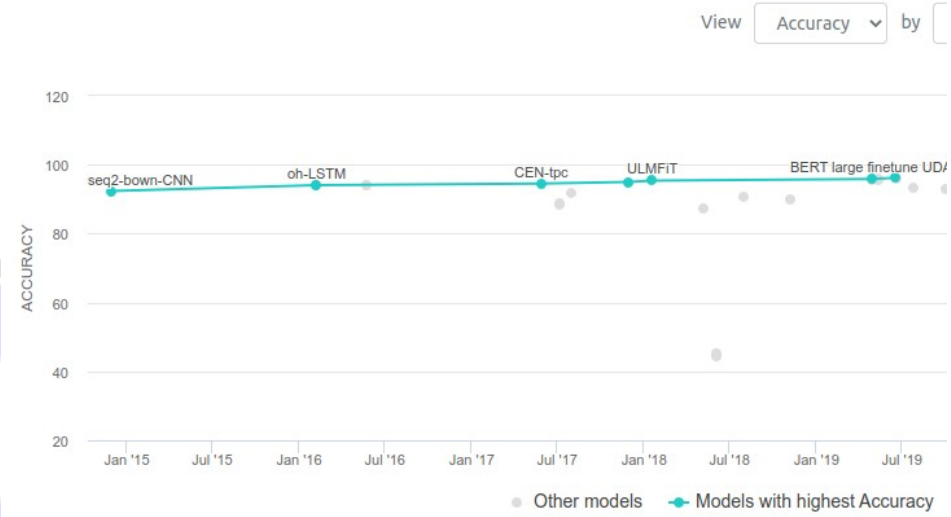
CNN : illustration de l'invariance locale



Zone d'incrustation vidéo

CNN : Implémentation (1/8)

- Détection de sentiment sur le « Large Movie Review Dataset » --- ACLImdb
- 25000 reviews positives / négatives pour le trainset et pour le testset



https://ai.stanford.edu/~amaas/data/sentiment/

Zone d'incrustation vidéo

CNN : Implémentation (2/8)

```
some_files = [
    './aclImdb_v1/aclImdb/train/pos/0_9.txt',
    './aclImdb_v1/aclImdb/train/pos/115_10.txt',
    './aclImdb_v1/aclImdb/train/pos/161_8.txt',
    './aclImdb_v1/aclImdb/train/pos/4002_8.txt',
    './aclImdb_v1/aclImdb/train/pos/3940_9.txt',
    './aclImdb_v1/aclImdb/train/pos/4888_8.txt',
    './aclImdb_v1/aclImdb/train/pos/10736_10.txt',
    './aclImdb_v1/aclImdb/train/pos/10696_7.txt'
]

for filepath in some_files:
    with open(filepath, 'r') as fs:
        comment = fs.read()
        print(comment)
        print('----')
```

```
Bromwell High is a cartoon comedy. It ran at the same time as so
hers". My 35 years in the teaching profession lead me to believe
lity than is "Teachers". The scramble to survive financially, th
eir pathetic teachers' pomp, the pettiness of the whole situatio
tudents. When I saw the episode in which a student repeatedly tr
d ......... at .......... High. A classic line: INSPECTOR: I'm h
to Bromwell High. I expect that many adults of my age think that
isn't!
----
Piece of subtle art. Maybe a masterpiece. Doubtlessly a special
fka style about impossibility of victory or surviving in a perpe
y exercise of adaptation. Lesson about limits and original sin,
```

Zone d'incrustation vidéo

CNN : Implémentation (3/8)

```
from cleantext import clean
import re
from nltk.corpus import stopwords

en_stopwords = set(stopwords.words('english'))

def clean_text(text):
    composed_word_regex = r'-'
    br_regex            = r'<[^<>\/]+\/>'
    anon_regex          = r'\.{4,}'
    normalise_point     = r'\.'
    space_regex         = r'\s+'
    normalise_tags      = r'><'
    unslash             = r'\/'

    cleaned_text = re.sub(composed_word_regex, ' ', text)
    cleaned_text = re.sub(anon_regex, '<ANON>', cleaned_text)
    cleaned_text = re.sub(br_regex, '', cleaned_text)
    cleaned_text = re.sub(normalise_point, '. ', cleaned_text)
    cleaned_text = re.sub(unslash, ' ', cleaned_text)

    cleaned_text = clean(cleaned_text,
```

Zone d'incrustation vidéo

CNN : Implémentation (4/8)

```
import gensim.downloader as gensim_api
import numpy as np
from os import listdir
from os.path import isfile, join
import pickle

embedding_model = gensim_api.load('word2vec-google-news-300')
```

```
from torch.utils.data import TensorDataset
import torch

def get_vector_with_oov(word):
      try:
        return embedding_model.get_vector(word)
      except KeyError:
        return np.zeros(300)
```

```
def process_file(filepath):
    with open(filepath, 'r') as fs:
        comment = fs.read()

    comment  = clean_text(comment)
    sequence = list()
    for word in comment.split():
        sequence.append(get_vector_with_oov(word))

    return sequence
```

Zone d'incrustation vidéo

CNN : Implémentation (5/8)

```python
def make_dataset(is_train = True, seq_len = 100, max_files_per_class = 2000000):
    if is_train:
        pos_seqs, pos_labels = process_dir(
            './aclImdb_v1/aclImdb/train/pos',
            max_files_per_class
        )
        neg_seqs, neg_labels = process_dir(
            './aclImdb_v1/aclImdb/train/neg',
            max_files_per_class
        )
    else:
        pos_seqs, pos_labels = process_dir(
            './aclImdb_v1/aclImdb/test/pos',
            max_files_per_class
        )
        neg_seqs, neg_labels = process_dir(
            './aclImdb_v1/aclImdb/test/neg',
            max_files_per_class
        )

    X = pos_seqs
    X.extend(neg_seqs)
    y = np.concatenate([pos_labels, neg_labels])

    # set the sequence len
    for i, seq in enumerate(X):
        if len(seq) > seq_len:
            X[i] = np.array(seq[:seq_len])
        elif len(seq) < seq_len:
            padded_seq = list(seq)
            padded_seq.extend(list(np.zeros((seq_len - len(seq), 300))))
            padded_seq = np.array(padded_seq)
            X[i]       = padded_seq

    X = np.array(X)
    return TensorDataset(
        torch.as_tensor(X, dtype = torch.float),
        torch.as_tensor(y, dtype = torch.long)
    )
```

Zone d’incrustation vidéo

CNN Implémentation (6/8)

```
class CNN(nn.Module):
    def __init__(
        self, seq_len = 100, filters = 300,
        window_size = 5, stride = 2, emb_len = 300
    ):
        super(CNN, self).__init__()

        self.seq_len     = seq_len
        self.filters     = filters
        self.kernel_size = window_size
        self.stride      = stride
        self.emb_len     = emb_len

        self.conv = nn.Conv1d(
            in_channels  = emb_len,
            out_channels = filters,
            kernel_size  = self.kernel_size,
            stride       = stride
        )

        self.pool = nn.MaxPool1d(
            kernel_size = self.kernel_size,
            stride      = stride
        )

        self.conv_transfer   = nn.ReLU()
        self.hidden_layer    = nn.Linear(self.conv_output_count(), 64)
        self.output_layer    = nn.Linear(64, 2)
        self.hidden_transfer = nn.ReLU()
        self.output_transfer = nn.LogSoftmax(dim = 1)
```

Zone d'incrustation vidéo

CNN Implémentation (7/8)

```
def conv_output_count(self):
    out_conv = ((self.seq_len - 1 * (self.kernel_size - 1) - 1) / self.stride) + 1
    out_conv = math.floor(out_conv)
    out_pool = ((out_conv - 1 * (self.kernel_size - 1) - 1) / self.stride) + 1
    out_pool = math.floor(out_pool)

    return out_pool * self.filters

def forward(self, x):
    x = x.permute(0, 2, 1)
    x = self.conv(x)
    x = self.pool(x)
    x = self.conv_transfer(x)
    x = x.reshape(len(x), -1)
    x = self.hidden_layer(x)
    x = self.hidden_transfer(x)
    x = self.output_layer(x)
    x = self.output_transfer(x)

    return x
```

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

CNN Implémentation (8/8)

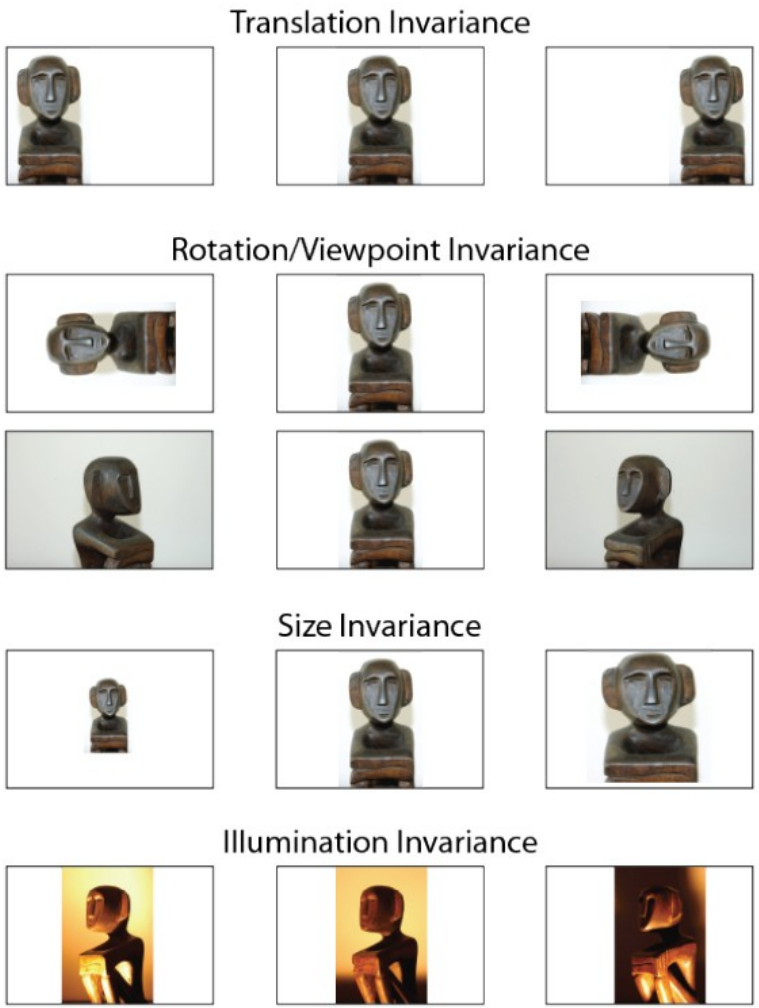
```
epochs        = 50
learning_rate = 1e-3
batch_size    = 64
model         = CNN()
loss_func     = nn.NLLLoss()
optimizer     = optim.Adam(
  params = model.parameters(),
  lr     = learning_rate
)

train(
  model, loss_func, optimizer,
  train_dataset, test_dataset,
  epochs, batch_size
)
```

```
  0 -> 79.625% train accuracy   0 -> 74.725% test accuracy
  1 -> 94.050% train accuracy   1 -> 77.800% test accuracy
  2 -> 99.650% train accuracy   2 -> 75.850% test accuracy
  3 -> 100.000% train accuracy   3 -> 78.450% test accuracy
  4 -> 100.000% train accuracy   4 -> 78.875% test accuracy
  5 -> 100.000% train accuracy   5 -> 78.900% test accuracy
```

digital-learning certifiant

Zone d'incrustation vidéo

4.3 – CNN, RNN, LSTM



Zone d'incrustation vidéo

4.3 – CNN, RNN, LSTM

Réseau de neurones récurrent (RNN)

- Point de départ : les CNN sont incapable de prendre en compte les relations positionnelles des patterns
- De plus les éléments lointains d'une séquence ne peuvent avoir de l'influence sur les éléments actuels

S1 : "Alice a donné un livre à Bob"
S2 : "Bob a donné un livre à Alice"
S3 : "donné Alice à livre un Bob a"

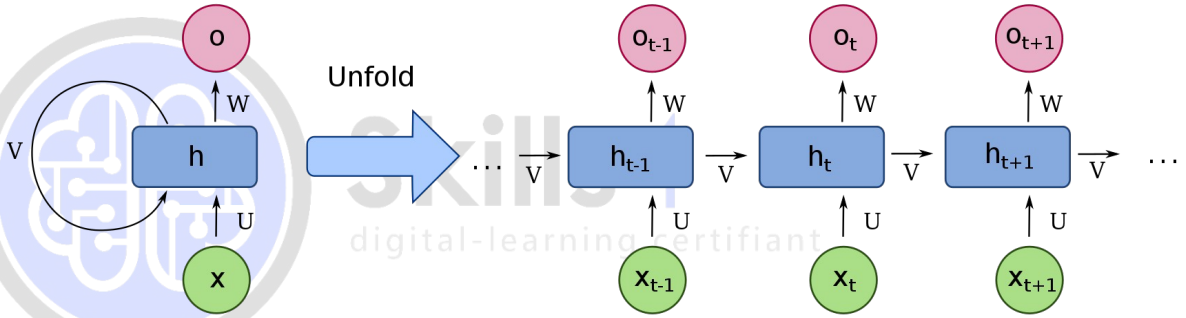
=> CNN(S1) =~ CNN(S2) =~ CNN(S3)

digital-learning certifiant

Zone d'incrustation vidéo

RNN : couche récurrente

- Chaque neurone possède un ensemble de poids connectés aux sorties précédentes de la couche cachée
- Le réseau peut réagir et apprendre par rapport aux entrées précédentes de la séquence



Zone d'incrustation vidéo

RNN : définition

- N_i est le nombre de neurones dans la couche i.
- Les entrées du neurones $x_i^k(t), k \in [0, 1, ..., N_{i-1}]$ où N_{i-1} est le nombre de neurones à la couche précédente. t est l'itération actuelle de l'exécution du RNN.
- La sortie du neurone, y_{ij}
- $w_{ij}^k, k \in [0, 1, ..., N_{i-1}]$ sont les poids associés à chacune des entrées du neurone.
- $v_{ij}^l, l \in [0, 1, ..., N_i]$ sont les poids associés à chacune des sorties des neurones de la même couche mais pour l'itération précédente (et donc l'élément précédent).
- a_{ij} est le niveau d'activation du neurone. Nous viendrons à sa définition ci-après.
- $h(x)$ est la fonction de transfert du neurone.

$$a_{ij} = \sum_{k=0}^{N_{i-1}} w_{ij}^k . x_i^k(t) + \sum_{l=0}^{N_i} v_{ij}^l . y_{il}(t-1)$$

Zone d'incrustation vidéo

RNN : entraînement

Pour la première itération ($t = 0$), les sorties des couches cachées précédentes sont initialisées à 0.

Le RNN, tel que défini, produit une séquence à partir d'une séquence. Nous pouvons utiliser et entrainer ce réseau de plusieurs manières :

- Attribuer des erreurs à chaque élément de la séquence. Soit nous avons une séquence attendue avec laquelle nous pouvons comparer directement chaque élément, soit nous pouvons ajouter d'autres couches qui vont pouvoir faire ce travail comme un CNN.
- Attribuer des erreurs uniquement sur le dernier élément de la séquence. Cet élément pouvant être utilisé pour prédire la classe de la séquence, par exemple.

SKILS4
digital-learning certifiant

Zone d'incrustation vidéo

4.3 – CNN, RNN, LSTM

- Cas pratique : classification de news avec RNN
  - AG's news topic classification dataset
  - 4 classes : Word, Sports, Business, Sci/Tech
  - Des articles collectés depuis plus de 2000 sources
  - Chaque classe contient 30000 articles pour l'entraînement et 1900 articles pour le test

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Cas pratique RNN (1/6)

```python
import torchdata
import torchtext
from torch.utils.data import DataLoader
from torchtext.data import get_tokenizer
from torchtext.vocab import build_vocab_from_iterator

train_dataset, test_dataset = torchtext.datasets.AG_NEWS()
tokenizer = get_tokenizer("basic_english")

def build_vocabulary(dataset):
    for _, text in dataset:
        yield tokenizer(text)

vocab = build_vocab_from_iterator(
    build_vocabulary(train_dataset),
    min_freq = 1,
    specials = ["<UNK>"]
)
vocab.set_default_index(vocab["<UNK>"])
```

```python
print(len(vocab))
tokens  = tokenizer('RNN is a neural network that is able to understand sequences')
indexes = vocab(tokens)
print(tokens)
print(indexes)
```

```
95811
['rnn', 'is', 'a', 'neural', 'network', 'that', 'is', 'able', 'to', 'understand', 'sequences']
[0, 21, 5, 79788, 200, 17, 21, 1226, 4, 4859, 86985]
```

Zone d'incrustation vidéo

Cas pratique RNN (2/6)

```python
from torchtext.data.functional import to_map_style_dataset
import torch

train_dataset  = to_map_style_dataset(train_dataset)
test_dataset   = to_map_style_dataset(test_dataset)
target_classes = ["World", "Sports", "Business", "Sci/Tech"]
max_words      = 25

def vectorize_batch(batch):
    y, X = list(zip(*batch))
    X = [vocab(tokenizer(text)) for text in X]

    for index, tokens in enumerate(X):
        if len(tokens) < max_words:
            X[index] = tokens + ([0] * (max_words-len(tokens)))
        else:
            X[index] = tokens[:max_words]

    return torch.tensor(X, dtype = torch.int32), torch.tensor(y) - 1

train_loader = DataLoader(
    train_dataset,
    batch_size = 1024,
    collate_fn = vectorize_batch,
    shuffle    = True
)
test_loader = DataLoader(
    test_dataset,
    batch_size = 1024,
    collate_fn = vectorize_batch
)
```

Zone d’incrustation vidéo

Cas pratique RNN (3/6)

```python
from torch import nn
from torch.nn import functional as F

embed_len  = 50
hidden_dim = 50

class RNN(nn.Module):
    def __init__(self):
        super(RNN, self).__init__()
        self.embedding_layer = nn.Embedding(
            num_embeddings = len(vocab),
            embedding_dim  = embed_len
        )
        self.rnn = nn.RNN(
            input_size   = embed_len,
            hidden_size  = hidden_dim,
            num_layers   = 1,
            nonlinearity = 'relu',
            batch_first  = True
        )
        self.linear   = nn.Linear(
            hidden_dim,
            len(target_classes)
        )
        self.output_transfer = nn.LogSoftmax(dim = 1)

    def forward(self, x):
        x = self.embedding_layer(x)
        output, hidden = self.rnn(x)
        x = output[:,-1]
        x = self.linear(x)
        return self.output_transfer(x)
```

Zone d'incrustation vidéo

Cas pratique RNN (4/6)

```
epochs        = 20
learning_rate = 1e-3
model         = RNN()
loss_func     = nn.NLLLoss()
optimizer     = optim.Adam(
  params = model.parameters(),
  lr     = learning_rate
)
train(
  model, loss_func, optimizer,
  train_loader, test_loader, epochs
)
```

```
  0 -> 36.111% train accuracy   0 -> 34.961% test accuracy
  1 -> 66.685% train accuracy   1 -> 65.026% test accuracy
  2 -> 76.779% train accuracy   2 -> 74.079% test accuracy
  3 -> 82.857% train accuracy   3 -> 79.934% test accuracy
  4 -> 86.445% train accuracy   4 -> 83.026% test accuracy
  5 -> 88.404% train accuracy   5 -> 84.579% test accuracy
  6 -> 89.840% train accuracy   6 -> 85.526% test accuracy
  7 -> 90.837% train accuracy   7 -> 85.961% test accuracy
  8 -> 91.589% train accuracy   8 -> 86.447% test accuracy
  9 -> 92.329% train accuracy   9 -> 86.500% test accuracy
 10 -> 93.203% train accuracy  10 -> 86.803% test accuracy
 11 -> 93.632% train accuracy  11 -> 87.066% test accuracy
 12 -> 94.239% train accuracy  12 -> 87.289% test accuracy
 13 -> 94.432% train accuracy  13 -> 87.197% test accuracy
 14 -> 95.001% train accuracy  14 -> 87.105% test accuracy
 15 -> 95.347% train accuracy  15 -> 87.303% test accuracy
 16 -> 95.343% train accuracy  16 -> 87.053% test accuracy
 17 -> 95.964% train accuracy  17 -> 87.092% test accuracy
 18 -> 96.373% train accuracy  18 -> 87.158% test accuracy
 19 -> 95.870% train accuracy  19 -> 86.711% test accuracy
```

Zone d'incrustation vidéo

Cas pratique RNN (5/6)

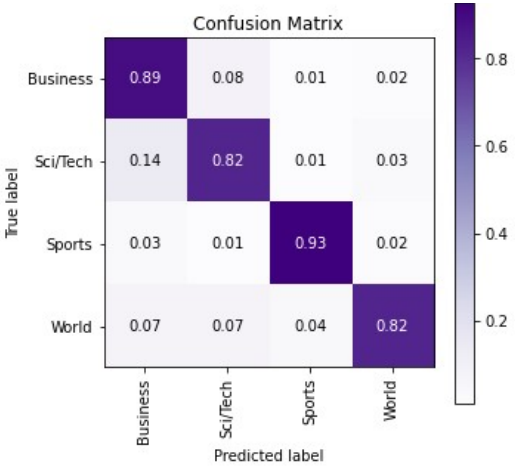
```
import torch.nn.functional as F

def make_predictions(model, test_loader):
    y_true, y_pred = [], []
    with torch.no_grad():
        for X, y in test_loader:
            preds = model(X)
            y_pred.append(preds)
            y_true.append(y)

    y_true, y_pred = torch.cat(y_true), torch.cat(y_pred)
    y_pred         = y_pred.argmax(dim = -1)
    y_true, y_pred = y_true.numpy(), y_pred.numpy()

    return y_true, y_pred

y_true, y_pred = make_predictions(model, test_loader)
```

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

print(f"accuracy on testset : {accuracy_score(y_true, y_pred)}")
print("\nclassification report : ")
print(classification_report(y_true, y_pred, target_names = target_classes))
print("\nconfusion matrix : ")
print(confusion_matrix(y_true, y_pred))
```

Zone d'incrustation vidéo

Cas pratique RNN (6/6)

```
accuracy on testset : 0.8671052631578947

classification report :
              precision    recall  f1-score   support

       World       0.92      0.82      0.87      1900
      Sports       0.95      0.93      0.94      1900
    Business       0.79      0.89      0.83      1900
    Sci/Tech       0.84      0.82      0.83      1900

    accuracy                           0.87      7600
   macro avg       0.87      0.87      0.87      7600
weighted avg       0.87      0.87      0.87      7600
```



Zone d'incrustation vidéo

Explicabilité du modèle entraîné (lime-text)

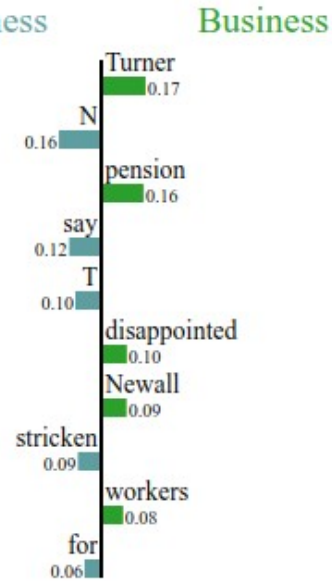
Text with highlighted words

Fears for T N pension after talks Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul.

```
prediction :  Business
truth :       Business
```

Prediction probabilities

World	0.02
Sports	0.00
Business	0.88
Sci/Tech	0.10



Zone d'incrustation vidéo

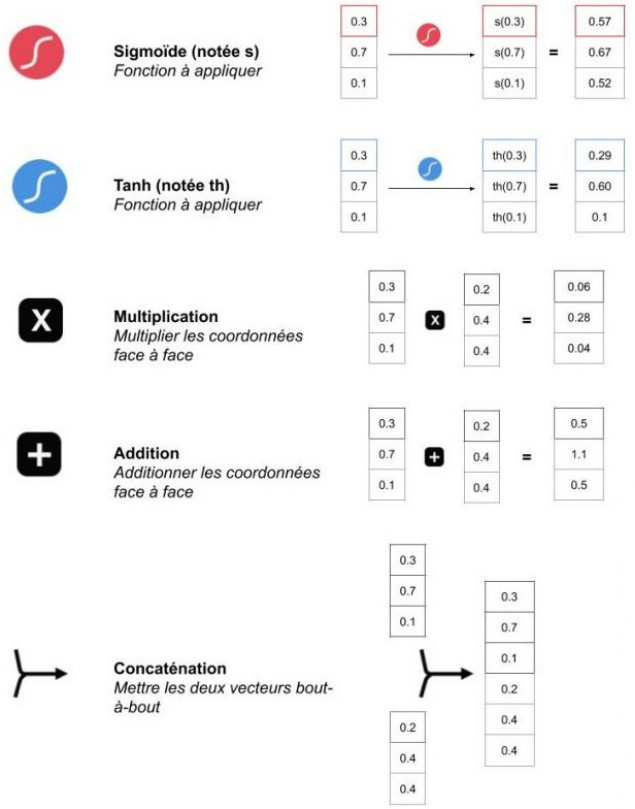
4.3 – CNN, RNN, LSTM

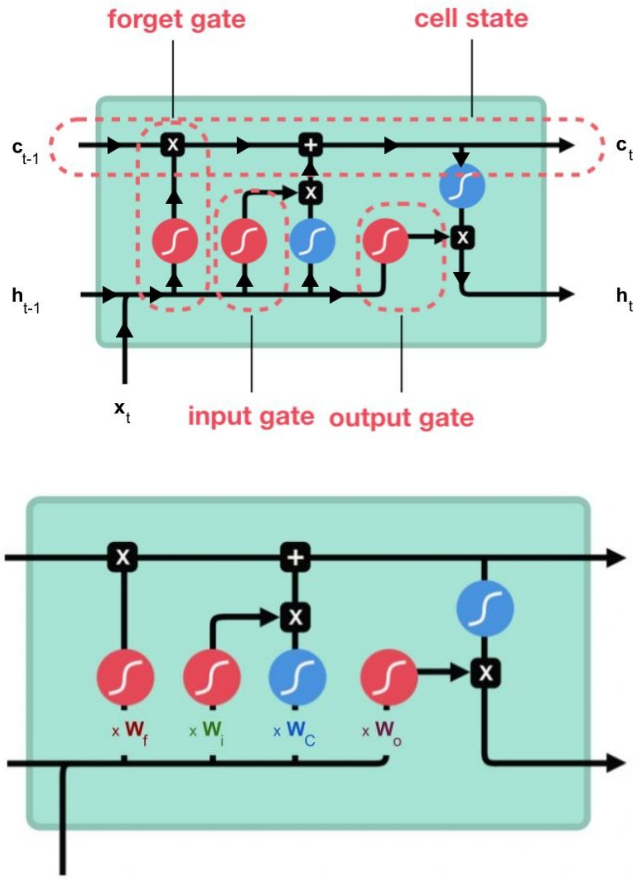
- Long Short-Term Memory (Mémoire à court-terme étendue)
  - Disparition du gradient → un élément lointain ne peut affecter l'élément actuel
  - Difficile à entraîner
  - Utilisables sur des séquences courtes
- Améliorations par rapport au RNN
  - Chaque neurone LSTM possède une mémoire entraînable
  - Cela lui permet d'être affecté par des éléments lointains
  - Avant 2018, les LSTM & LSTM bidirectionnels étaient les meilleurs models pour le NLP

Zone d'incrustation vidéo

skills4 Artificial Intelligence certified

Couche LSTM





Zone d'incrustation vidéo

LSTM implémentation (1/2)

```
class LSTM(nn.Module):
    def __init__(self, emb_size = 300, hidden_size = 64):
        super(LSTM, self).__init__()
        self.hidden_size = hidden_size

        self.lstm = nn.LSTM(
            input_size  = emb_size,
            hidden_size = hidden_size,
            batch_first = True
        )

        self.lstm_transfer   = nn.ReLU()
        self.hidden_layer    = nn.Linear(hidden_size, 64)
        self.output_layer    = nn.Linear(64, 2)
        self.hidden_transfer = nn.ReLU()
        self.output_transfer = nn.LogSoftmax(dim = 1)

    def forward(self, x):
        h_0 = Variable(torch.zeros(1, len(x), self.hidden_size))
        c_0 = Variable(torch.zeros(1, len(x), self.hidden_size))

        out, (h_n, c_n) = self.lstm(x, (h_0, c_0))
        x = self.lstm_transfer(h_n)
        x = x.reshape(-1, self.hidden_size)
        x = self.hidden_layer(x)
        x = self.hidden_transfer(x)
        x = self.output_layer(x)
        x = self.output_transfer(x)

        return x
```

Zone d'incrustation vidéo

LSTM Implémentation (2/2)

- Application à la détection de sentiment (CNN)

```
epochs        = 50
learning_rate = 1e-3
batch_size    = 64
model         = LSTM()
loss_func     = nn.NLLLoss()
optimizer     = optim.Adam(
  params = model.parameters(),
  lr     = learning_rate
)

train(
  model, loss_func, optimizer,
  train_dataset, test_dataset,
  epochs, batch_size
)
```

```
 0 -> 71.250% train accuracy   0 -> 68.325% test accuracy
 1 -> 51.475% train accuracy   1 -> 49.950% test accuracy
 2 -> 54.600% train accuracy   2 -> 52.875% test accuracy
 3 -> 59.775% train accuracy   3 -> 56.700% test accuracy
 4 -> 50.300% train accuracy   4 -> 50.075% test accuracy
 5 -> 71.000% train accuracy   5 -> 68.225% test accuracy
 6 -> 73.675% train accuracy   6 -> 69.850% test accuracy
 7 -> 73.775% train accuracy   7 -> 70.100% test accuracy
 8 -> 76.075% train accuracy   8 -> 72.100% test accuracy
 9 -> 76.925% train accuracy   9 -> 72.850% test accuracy
10 -> 75.775% train accuracy  10 -> 71.875% test accuracy
11 -> 75.175% train accuracy  11 -> 70.875% test accuracy
12 -> 78.100% train accuracy  12 -> 73.200% test accuracy
13 -> 79.800% train accuracy  13 -> 75.200% test accuracy
14 -> 80.425% train accuracy  14 -> 74.450% test accuracy
15 -> 82.275% train accuracy  15 -> 75.575% test accuracy
16 -> 83.475% train accuracy  16 -> 75.850% test accuracy
17 -> 84.800% train accuracy  17 -> 76.825% test accuracy
```

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

Partie 5 : État de l’art du NLP

Skills4All
digital-learning certifiant

5 – État de l'art du NLP

Sommaire

- Du mécanisme d'attention au Transformer
- Utilisation et ré-utilisation des Transformers
- Cas pratiques de classification
- Cas pratiques de génération de texte

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Partie 5.1 : Du mécanisme d’attention au Transformer

Skills4All
digital-learning certifiant

5.1 – Du mécanisme d'attention au Transformer

Point de départ

- En pratique les modèles RNN ou LSTM sont efficaces que sur des séquences courtes
- Cela a été observé sur les problèmes de traduction, résolus avant 2015 avec des systèmes LSTM encodeur-décodeur
- Plus les séquences sont longues, plus ce genre de systèmes fait des erreurs de traduction de contexte
- Trop d'information et peu de capacité de discerner ce qui est important de ce qui ne l'est pas
- Explicabilité du résultat limité

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

5.1 – Du mécanisme d'attention au Transformer

Mécanisme d'attention (1/5)

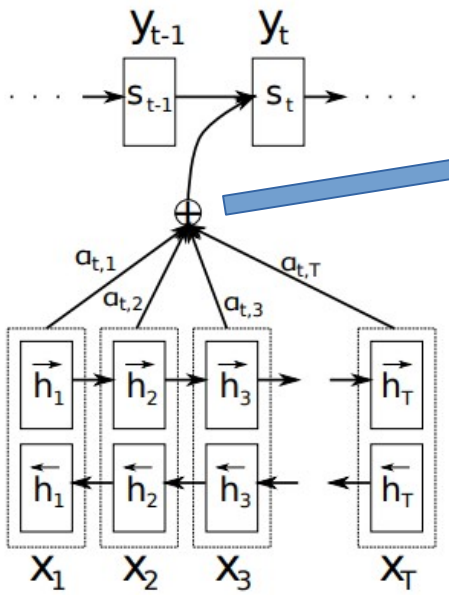
- L'attention est un mécanisme qui permet de sélectionner une poignée de état cachés récurrents dans la séquence afin de prédire le token suivant
- Grâce à cela il est possible d'utiliser des séquences de grande taille et le modèle sélectionnera ce qui est important
- Mécanisme développé sur le modèle seq2seq composé de deux RNN sur un schéma encoder-decoder

Zone d'incrustation vidéo

Skills4 digital-learning certificat

Mécanisme d'attention (2/5) : seq2seq

- Un RNN bidirectionnel encode la séquence, et un RNN simple utilise l'ensemble des états cachés pour prédire le token suivant



$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

Softmax (la somme des coefficients est 1)

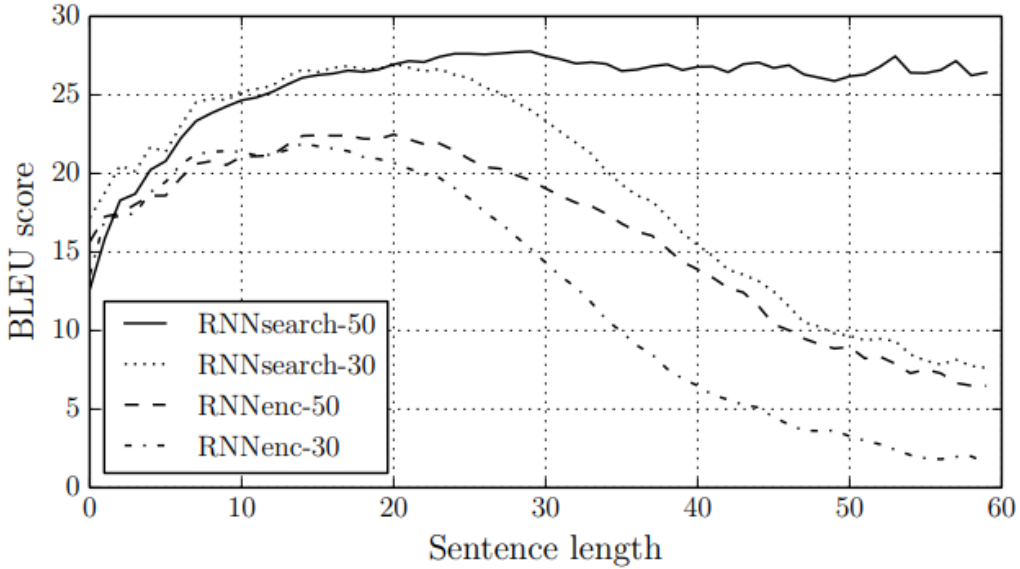
$$e_{ij} = a(s_{i-1}, h_j)$$

« a » est un modèle (Perceptron), entraîné avec les RNN qui donne un score d'alignement des tokens

Zone d'incrustation vidéo

Mécanisme d'attention (3/5)

- Les performances avec le mécanisme d'attention ne chutent pas avec l'augmentation des tailles des séquences



Zone d'incrustation vidéo

Mécanisme d'attention (4/5)

An admitting privilege is the right of a doctor to admit a patient to a hospital or a medical centre to carry out a diagnosis or a procedure, based on his status as a health care worker at a hospital.

Traduction de RNNencdec50 (sans mécanisme d'attention) :

Un privilège d'admission est le droit d'un médecin de reconnaître un patient à l'hôpital ou un centre médical d'un diagnostic ou de prendre un diagnostic en fonction de son état de santé.

Traduction de RNNsearch50 (avec mécanisme d'attention) :

Un privilège d'admission est le droit d'un médecin d'admettre un patient à un hôpital ou un centre médical pour effectuer un diagnostic ou une procédure, selon son statut de travailleur des soins de santé à l'hôpital.

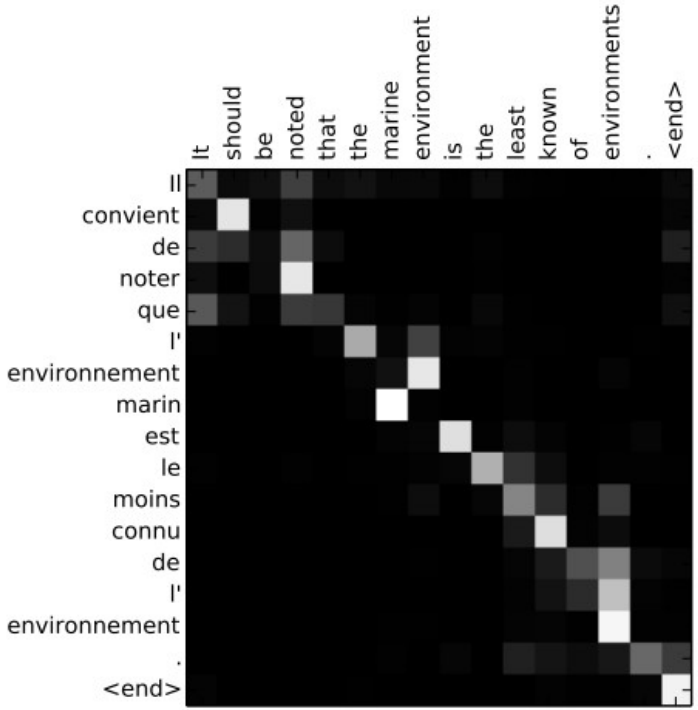
digital-learning certifiant

Zone d'incrustation vidéo

Mécanisme d’attention (5/5)

- **Les coefficients permettent d’interpréter le résultat directement**

Pléthore de mécanismes d’attention naissent à partir de ce moment !



Zone d’incrustation vidéo

5.1 – Du mécanisme d'attention au Transformer

- ***Attention is all you need***
  - **Des chercheurs ont réalisé que l'on pouvait se passer entièrement du RNN et garder uniquement le mécanisme d'attention**
  - **Ainsi né le *Transformer,* qui est un système encoder-decoder d'attention (multi-head) qui se connecte directement sur les embeddings d'entrée**
  - **Le Transformer est l'unité de calcul fondamentale de tous les meilleurs modèles à ce jour de l'état de l'art**

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

5.1 – Du mécanisme d'attention au Transformer

Comprendre le Transformer : *Self-Attention*

The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .

Zone d'incrustation vidéo

5.1 – Du mécanisme d'attention au Transformer

Generalized Attention : Key, Query, Value

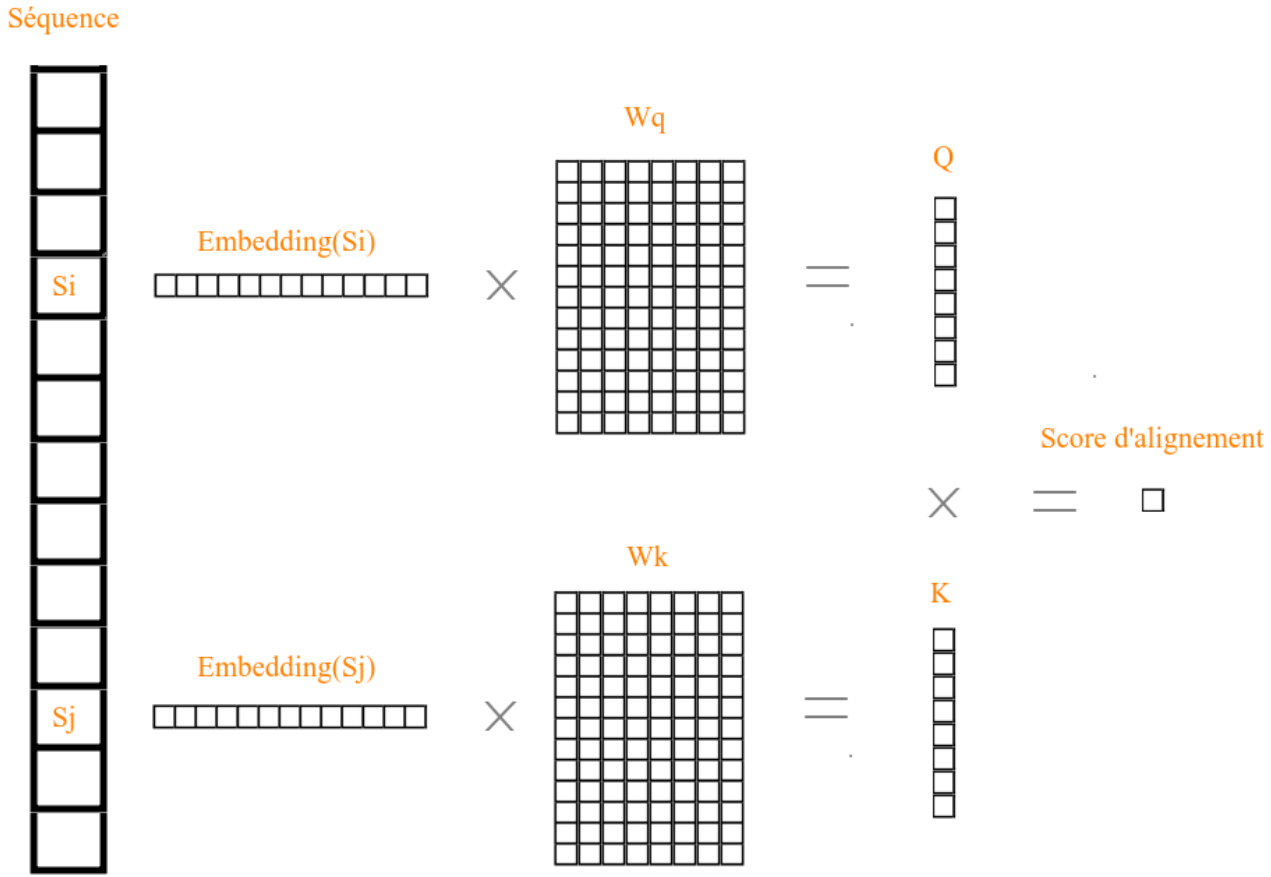
The FBI is chasing a criminal on the run .

- Chaque variable d'entrée (ici chaque embedding) a 3 vecteurs associés par multiplication : Key, Query et Value (K, Q, V)
- Query (current_token) * Key (other_token) = Score (score d'alignement)
- Sum(Score * Value (other_token)) = Context Vector (qui le vecteur encodant le context)
- Les scores d'alignement sont des softmax ~ c.a.d des probabilités dont la somme est 1

Zone d'incrustation vidéo

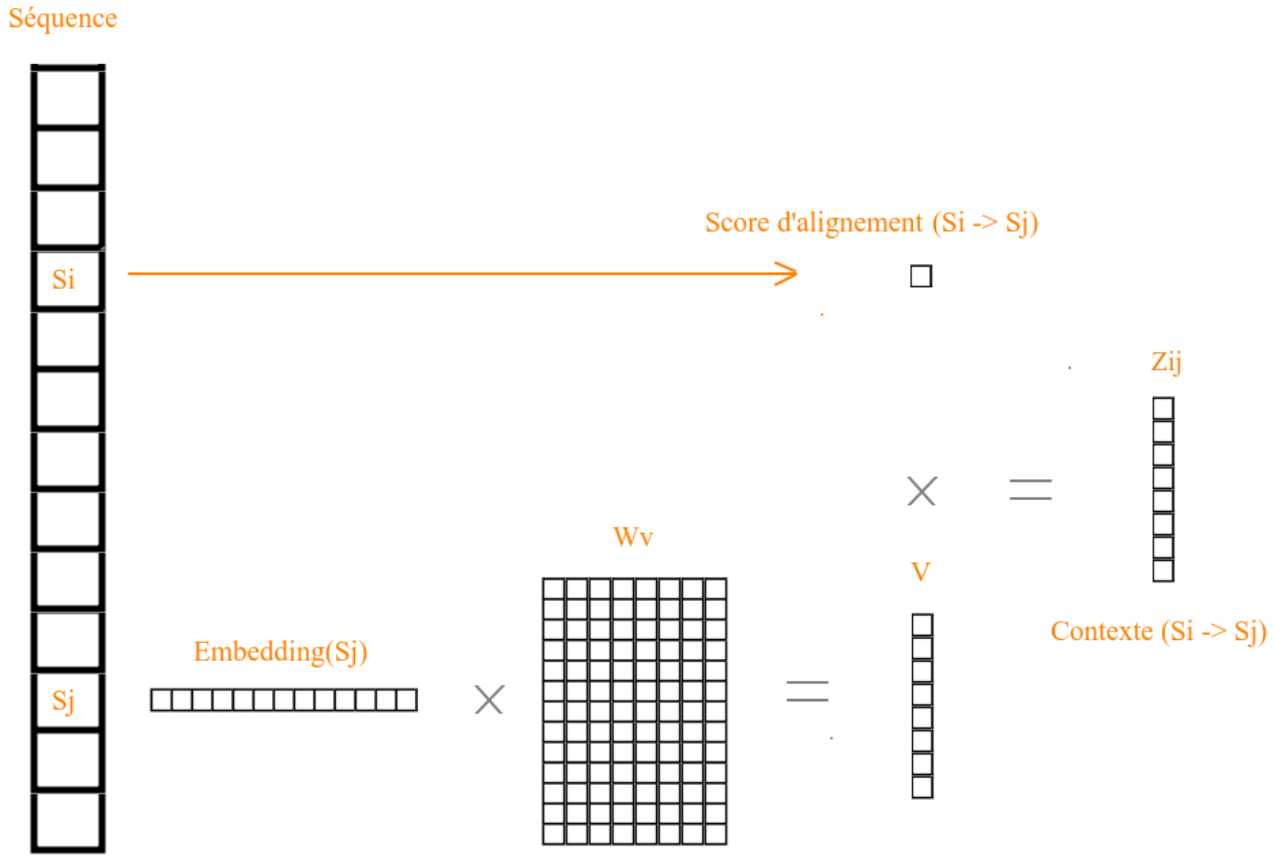
Skills4
digital-learning certifiant

GA : Calcul d'un score d'alignement



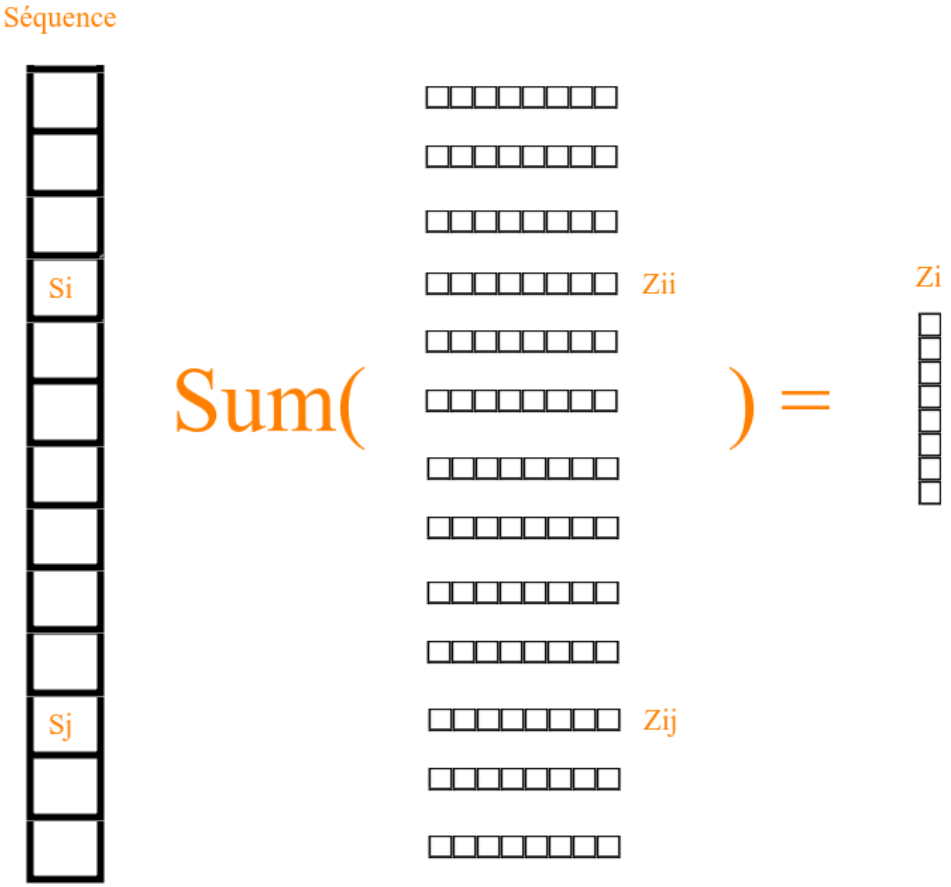
Zone d'incrustation vidéo

GA : Calcul d'un vecteur de contexte



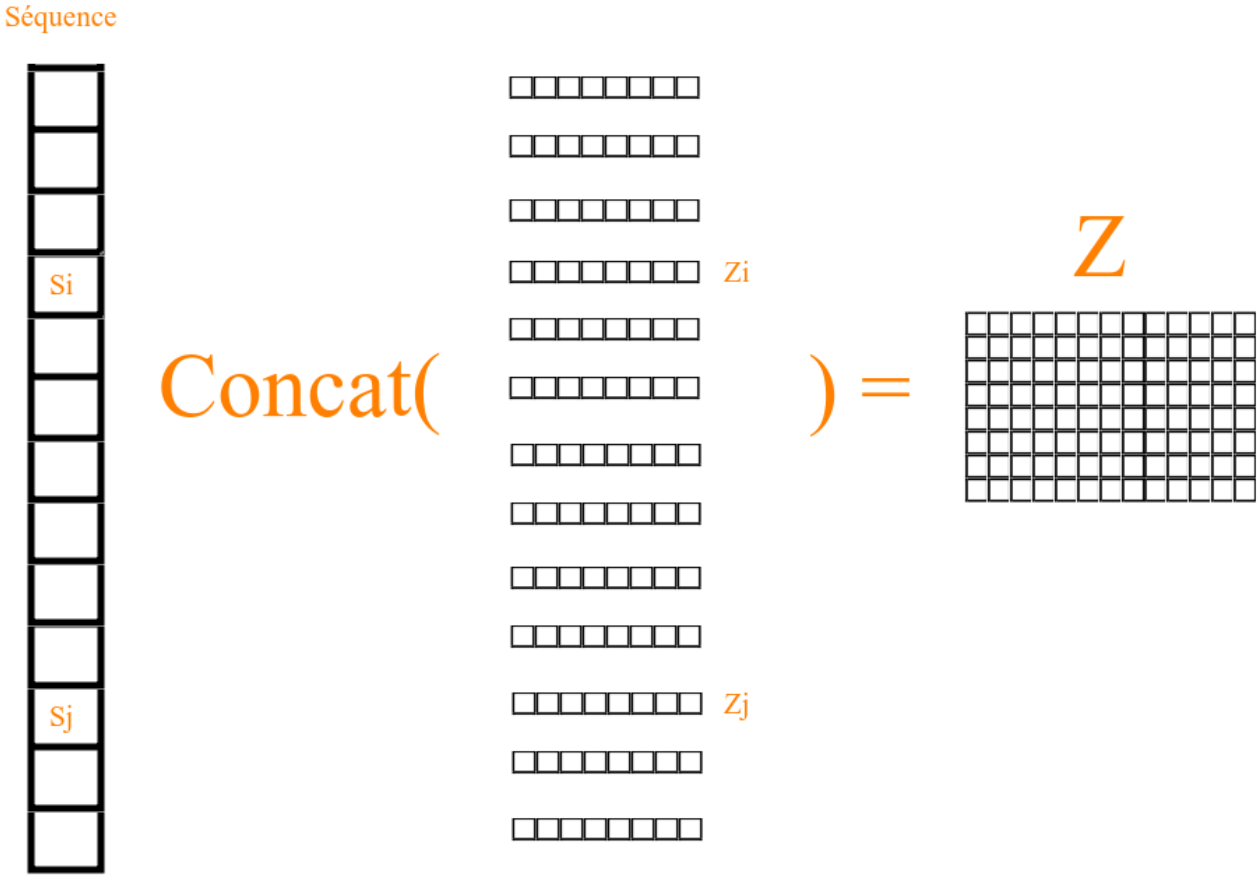
Zone d'incrustation vidéo

GA : Calcul du vecteur de contexte



Zone d'incrustation vidéo

GA : Calcul de la matrice de contexte



Zone d'incrustation vidéo

Attention généralisée

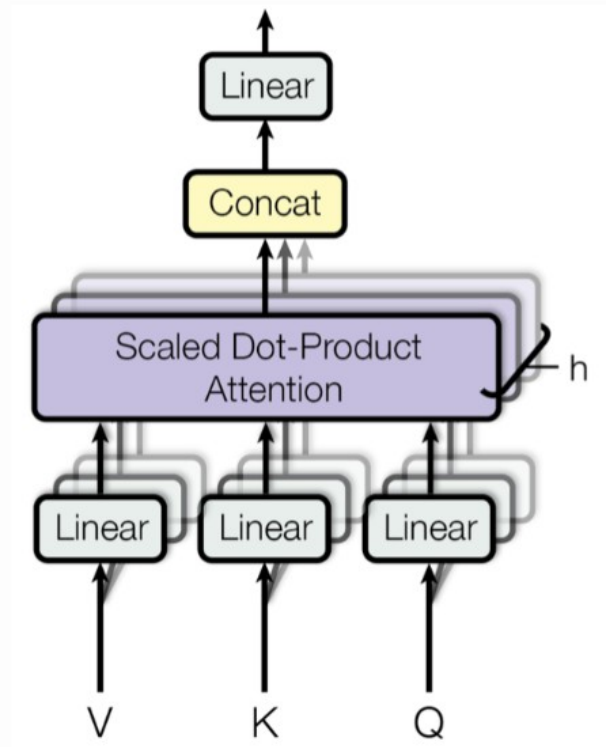
$$\text{Attention}(S) = \text{MLP}(\overset{Z}{\boxed{}})$$

Skills4
digital-learning certifiant

Zone d'incrustation vidéo

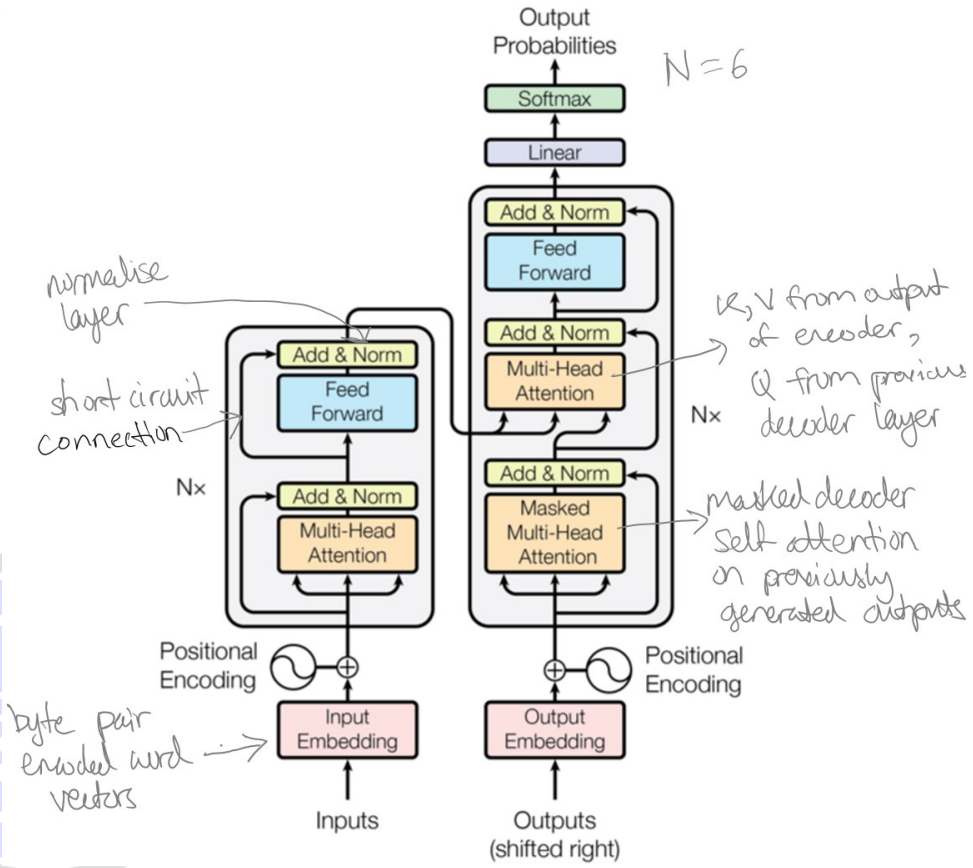
5.1 – Du mécanisme d'attention au Transformer

Multi-head Attention : *multiply the math fun*



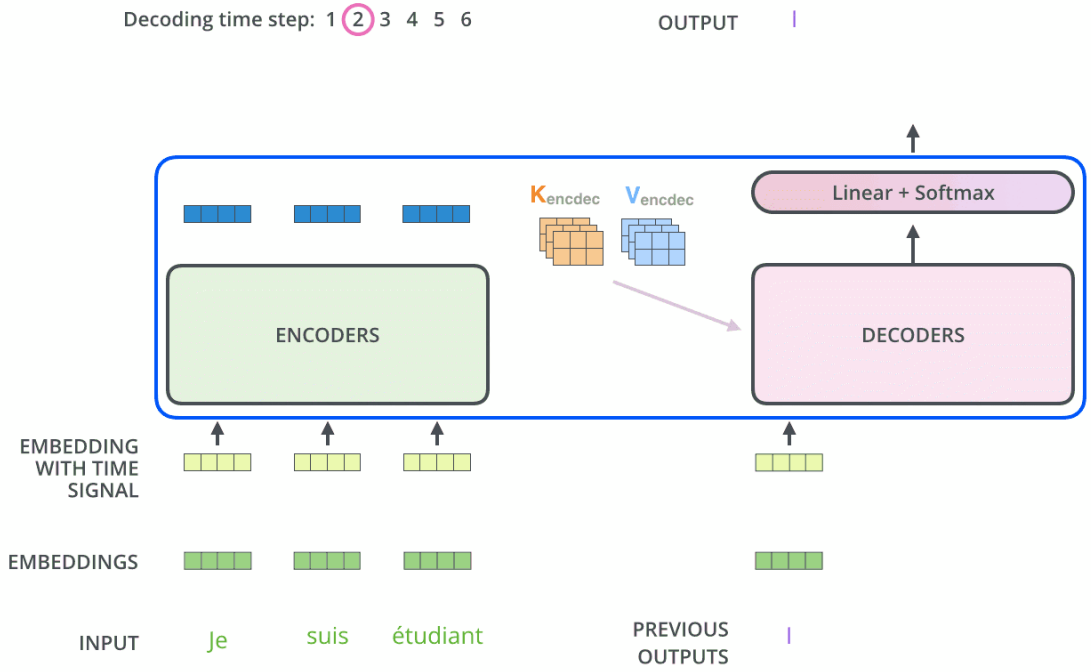
Zone d'incrustation vidéo

Architecture Transformer



Zone d'incrustation vidéo

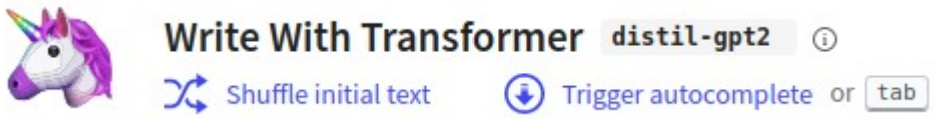
Architecture Transformer



Zone d'incrustation vidéo

Utiliser un Transformer entraîné en ligne

https://transformer.huggingface.co/



What Machine Learning and Deep Learning can enable us to do is:

1 - Machine Learning can do an interesting job with our dataset, but it is not as easy to do it with an automated process.

2 - In some cases machine learning is easier than using it as a natural intelligence algorithm, and this can be done with a human machine

3 - Machine learning can solve complex mathematical problems

digital-learning certifiant

Zone d'incrustation vidéo

Partie 5.2 : Utilisation & Réutilisation des Transformers

Skills4All
digital-learning certifiant

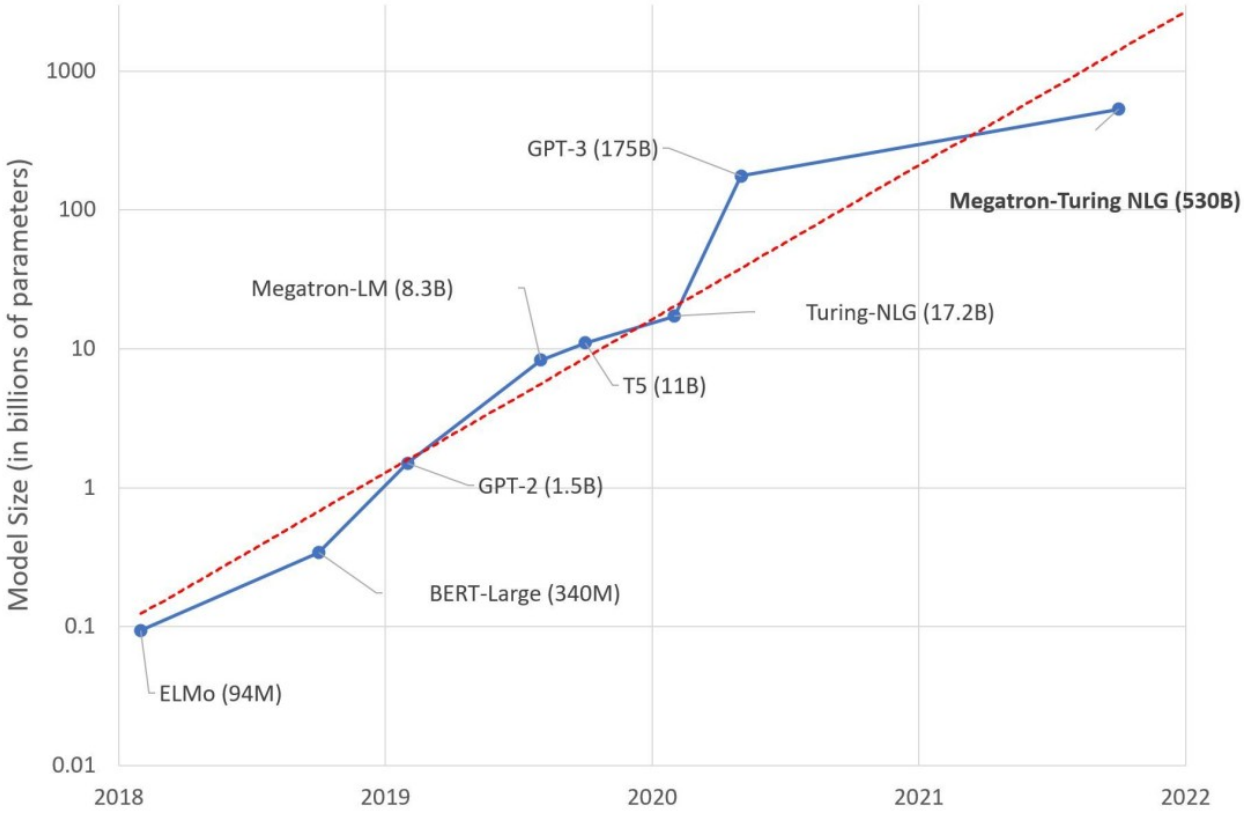
5.2 – Utilisation & ré-utilisation des Transformers

- **Réutilisation de modèles entraînés**
  - Plus les dimensions des poids du réseau sont grandes, plus le modèle est capable de généraliser
  - En pratique, à part dans certains laboratoires de recherche, on n'entraîne pas un modèle transformer de zéro
  - Il est possible d'utiliser des modèles généraux déjà entraînés tel quel ou de les ré-entrainer pour une tâche

Zone d'incrustation vidéo

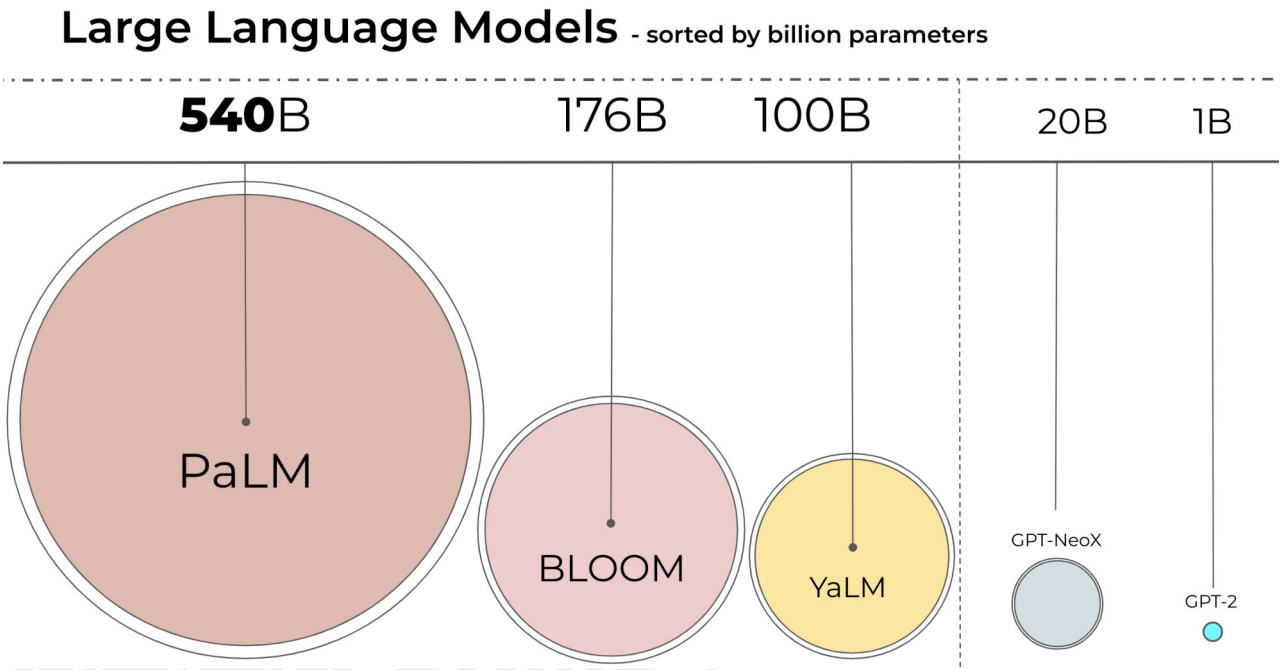
Skills4
digital-learning certifiant

5.2 – Utilisation & ré-utilisation des Transformers



Zone d'incrustation vidéo

5.2 – Utilisation & ré-utilisation des Transformers



digital-learning certifiant

Zone d'incrustation vidéo

5.2 – Utilisation & ré-utilisation des Transformers

- Quelque soit la tâche que vous résolvez
  - Il existe probablement déjà un modèle de NLP déjà entraîné pour résoudre ce problème ou un problème similaire
  - Sinon il existe probablement un modèle *zero-shot* ou *prompt-programmable* qui peut résoudre le problème
  - Sinon, il existe un modèle à partir duquel vous pouvez partir → finetuning
  - → goto *https://hugginface.co*

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

↗️ Étude de cas : *leturfu.fr* un blog 100 % automatisé avec le NLP

| Turfu |

Politique de confidentialité Contact

Article automatiquement traduit avec les meilleurs modèles de traduction disponible

Nous sommes à la veille d'un point de basculement climatique majeur, plus tôt que prévu

Image automatiquement choisie pour correspondre au contenu de l'article

Résumé automatique de l'article

La circulation méridienne de retournement de l'Atlantique a diminué de 15 % depuis le milieu du 20e siècle pour atteindre un nouveau record de faiblesse. L'alignement des régions froides…

Article catégorisé automatiquement

explications

Tags extraits automatiquement du contenu de l'article

actuel, affaiblir, amoc, arrêt, atlantique, auteur, baisse, chaleur, circulation, climat, climatique, climatiques, conséquence, conséquences, côte, côtier, débit, déplacement, désorganisation, échantillon, éléments, équateur, étude, évolution, fusion, geomar, glace, golfe, groenland, gulf, helmholtz, hémisphère, homard, impact, institute, jon, littoral, maine, médicaments, mer, mesure, météorologue, modèle, océan, océanique, océanographe, océanographique, océans,

Rechercher

Rechercher

Articles récents

Nous sommes à la veille d'un point de basculement climatique majeur, plus tôt que prévu

Pendant que nous étions occupés à débattre de la science du climat, nous avons perdu plus de la moitié de la glace permanente de l'Arctique

L'avenir de la Terre pourrait bientôt recréer un monde perdu datant de 50 millions d'années

Si les niveaux de CO2 sont suffisamment élevés, les nuages commenceront à se désagréger physiquement

Quelque chose d'inattendu est arrivé aux plantes de l'Arctique à mesure qu'elles se sont réchauffées

Zone d'incrustation vidéo

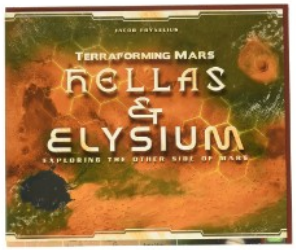
Étude de cas : *leturfu.fr* un blog 100 % automatisé avec le NLP

Traduction automatisée

Sur Mars, le deutérium est l'isotope dominant de l'hydrogène dans l'atmosphère, probablement parce que le rayonnement solaire élimine le protium, mais les rapports isotopiques dans les roches testées par les scientifiques varient de ceux de la Terre à ceux de Mars.

En savoir plus

En savoir plus



En savoir plus

Recommandation automatisé de produits

Barnes et son équipe ont donc décidé d'examiner de plus près des météorites dont ils étaient certains qu'elles provenaient de la croûte martienne.

Zone d'incrustation vidéo

Qu’est ce qu’un article ?

Farms of the future will not just need tech, but also intelligence. A new one-click rural property report pulls together data on every rural property in Australia. Backed by science.

We use digital data in our everyday lives, and now there is a way to break the ‘data drought’ for our farmers. Let’s face it, they’ve certainly had a rough trot lately.

Zone d’incrustation vidéo

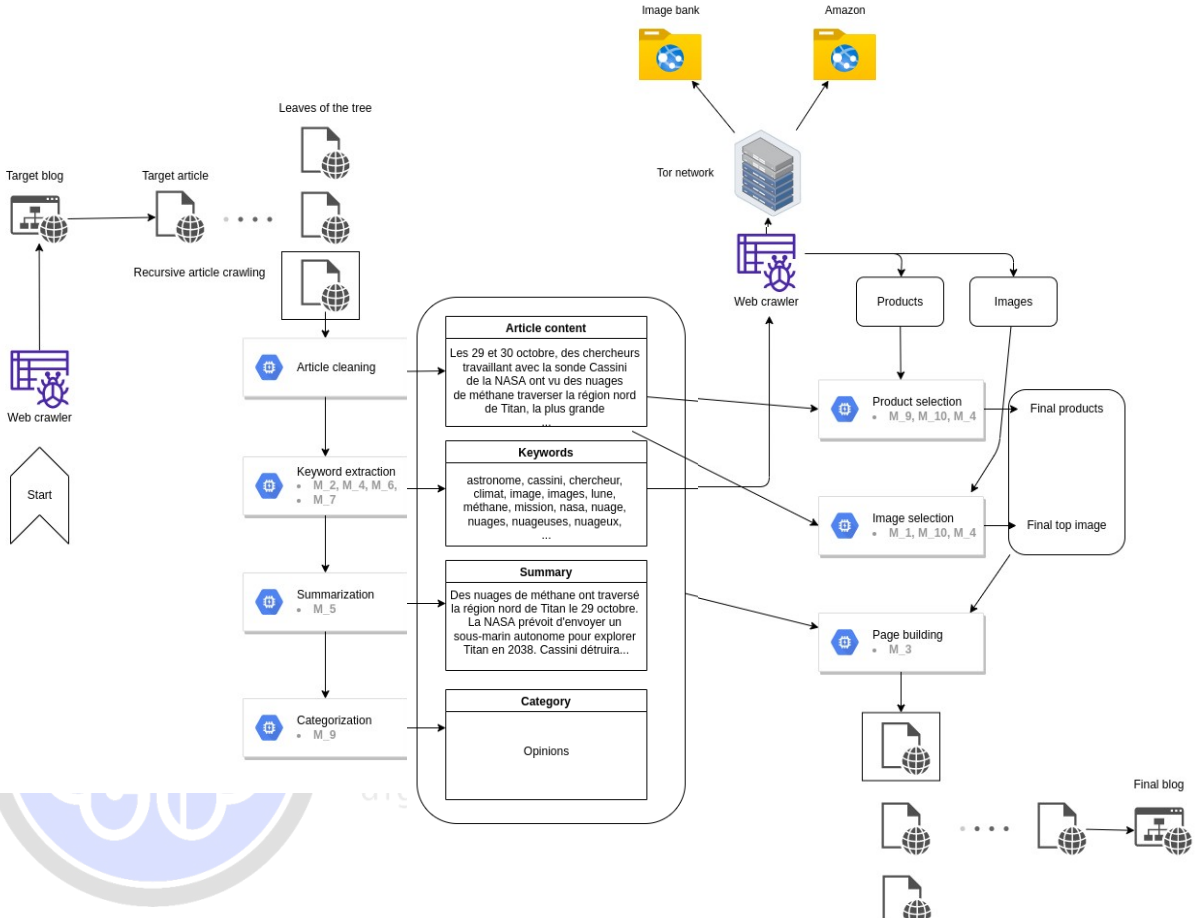
5.2 – Utilisation & ré-utilisation des Transformers

- Qu’est ce qu’un article ?
  - Un titre → peut être traduit automatiquement
  - Une image pour attirer le chaland → ne peut être réutiliser, souvent pas libre de droits
  - Un texte formaté (url, gras, parapgraphes, italique etc.) → Peut être traduit mais doit l’être en conservant le format
  - Des images explicatives → libres de droits quasi-systématiquement car informatives (doit parler du sujet précisément) donc peut être réutilisées directement

Zone d’incrustation vidéo

Skills-4 digital-learning certificate

Architecture globale de *leturfu.fr*



Zone d'incrustation vidéo

↗ Focus sur un des sous-problèmes : résumé de texte

```
from transformers          import pipeline
from sentence_transformers import SentenceTransformer

summarizer = pipeline('summarization', model = 'philschmid/distilbart-cnn-12-6-samsum')
```

```
text = """
Hurricane Ian continues to rain destruction onto the Florida.
Millions have been left dark in the Sunshine State and residents
along the low-lying peninsula’s many barrier islands have been
cut off from the mainland as vehicles and whole roads were
swallowed by the massive Category 4 storm.
"""

print(summarizer(
    text,
    min_length = 10,
    max_length = 25
)[0]['summary_text'])
```

```
Millions have been left dark in the Sunshine State because of Hurricane Ian.
```

3KINS4
digital-learning certifiant

Zone d’incrustation vidéo

↗️ Focus sur un des sous-problèmes : filtrer les résultats Amazon qui ne sont pas des livres

```
from operator import itemgetter

zero_shot_text_classificiation = pipeline(
    'zero-shot-classification',
    model = 'valhalla/distilbart-mnli-12-6'
)

def argmax(alist):
    return max(enumerate(alist), key = itemgetter(1))

def text_classification(text):
    res       = zero_shot_text_classificiation(text, classes)
    index, _  = argmax(res['scores'])
    label     = res['labels'][index]
    return label

classes = [
      'This product description refers to a book or an ebook',
      'This product description does not refer to a book and does not refer to an ebook'
]

text_1 = "This is the best seller book of the year, buy it now!"
text_2 = "This T-shirt will make you good looking and smarter everyday, buy it now!"

print(f'text_1: {text_classification(text_1)}')
print(f'text_2: {text_classification(text_2)}')
```

```
text_1: This product description refers to a book or an ebook
text_2: This product description does not refer to a book and does not refer to an ebook
```

Zone d'incrustation vidéo

Finetuning d'un Transformer (1/3)

```
from datasets import load_dataset

dataset = load_dataset("yelp_review_full")
dataset["train"][0]
```

```
{'label': 4,
 'text': "dr. goldberg offers everything i look for in a genera
out being patronizing; he's always on time in seeing his patien
which my parents have explained to me is very important in case
n get referrals to see specialists without having to see him fi
ere trying to think of any complaints i have about him, but i'm
```

```
from transformers import AutoModelForSequenceClassification
from transformers import AutoTokenizer
from transformers import TrainingArguments, Trainer
import numpy as np
import evaluate

tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")
model     = AutoModelForSequenceClassification.from_pretrained(
    "distilbert-base-uncased",
    num_labels = 5
)
```

Zone d'incrustation vidéo

Finetuning d'un Transformer (2/3)

```
def tokenize_function(examples):
    return tokenizer(
        examples["text"],
        padding    = "max_length",
        truncation = True
    )

tokenized_dataset   = dataset.map(tokenize_function, batched = True)
small_train_dataset = tokenized_dataset["train"].shuffle().select(range(1000))
small_eval_dataset  = tokenized_dataset["test"].shuffle().select(range(1000))

training_args     = TrainingArguments(
    output_dir          = "test_trainer",
    evaluation_strategy = "epoch"
)
metric = evaluate.load("accuracy")

def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions    = np.argmax(logits, axis = -1)
    return metric.compute(predictions = predictions, references = labels)
```

SKILLS4
digital-learning certifiant

Zone d'incrustation vidéo

Finetuning d'un Transformer (3/3)

```
trainer = Trainer(
    model           = model,
    args            = training_args,
    train_dataset   = small_train_dataset,
    eval_dataset    = small_eval_dataset,
    compute_metrics = compute_metrics
)

trainer.train()
```

[375/375 1:22:31, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	1.126187	0.503000
2	No log	1.031827	0.554000
3	No log	1.046638	0.546000

Zone d'incrustation vidéo

Partie 5.3 : Cas pratiques de classification

Skills4All
digital-learning certifiant

Cas pratique : classification de spams

- Enron Spam Dataset

SetFit/enron_spam

```
print(len(dataset['train']))
print(len(dataset['test']))
print(dataset['train'][0])
```

```
31716
2000
{
  'message_id': 33214,
  'text': 'any software just for 15 $ - 99 $ understanding oem
          software\nlead me not into temptation ; i can find
          the way myself .\n# 3533 . the law disregards trifles .',
  'label': 1,
  'label_text': 'spam',
  'subject': 'any software just for 15 $ - 99 $',
  'message': 'understanding oem software\nlead me not into temptation ;
             i can find the way myself .\n# 3533 . the law disregards
             trifles .',
  'date': datetime.datetime(2005, 6, 18, 0, 0)
}
```

Zone d'incrustation vidéo

Une solution

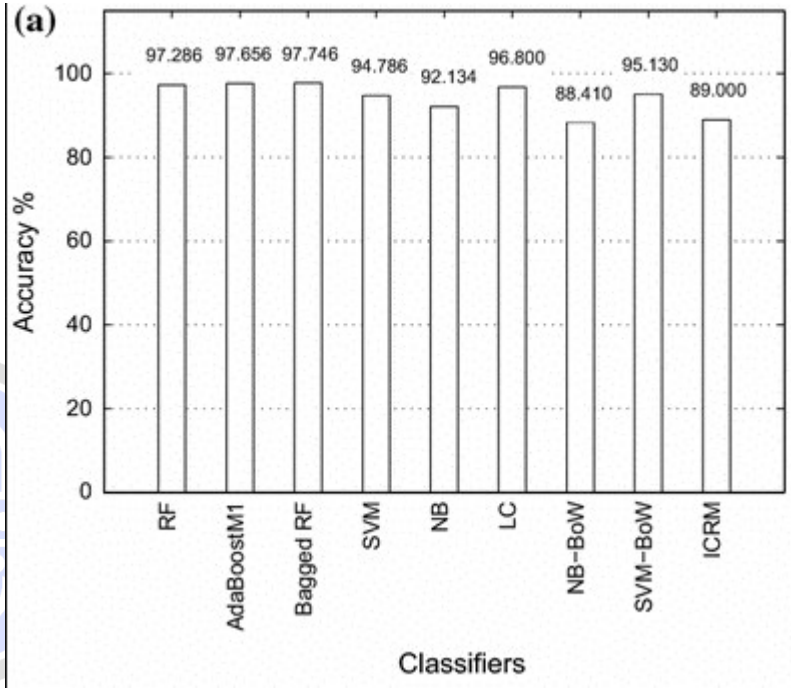
```
def tokenize_function(examples):
    return tokenizer(
        clean_text(examples["text"]),
        padding    = "max_length",
        truncation = True
    )

tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")
model     = AutoModelForSequenceClassification.from_pretrained(
    "distilbert-base-uncased",
    num_labels = 2
)

tokenizer.add_tokens(['<url>', '<email>', '<phone>', '<number>', '<digit>', '<cur>'])
model.resize_token_embeddings(len(tokenizer))

tokenized_dataset   = dataset.map(tokenize_function, batched = True)
small_train_dataset = tokenized_dataset["train"].shuffle().select(range(1000))
small_eval_dataset  = tokenized_dataset["test"].shuffle().select(range(1000))

training_args     = TrainingArguments(
    output_dir          = "test_trainer",
    evaluation_strategy = "epoch"
)
metric = evaluate.load("accuracy")

trainer = Trainer(
    model           = model,
    args            = training_args,
    train_dataset   = small_train_dataset,
    eval_dataset    = small_eval_dataset,
    compute_metrics = compute_metrics
)

trainer.train()
```

Zone d'incrustation vidéo

5.3 – Cas pratiques de classification

Résultats

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.091375	0.979000
2	No log	0.077648	0.983000
3	No log	0.079794	0.984000



Zone d'incrustation vidéo

5.3 – Cas pratiques de classification

Cas pratique : classification sentiment Twitter

- **TSATC : Twitter Sentiment Analysis Training**

  carblacac/twitter-sentiment-analysis

Dataset Summary

TSATC: Twitter Sentiment Analysis Training Corpus The original Twitter Sentiment Analysis Dataset contains 1,578,627 classified tweets, each row is marked as 1 for positive sentiment and 0 for negative sentiment. It can be downloaded from http://thinknook.com/wp-content/uploads/2012/09/Sentiment-Analysis-Dataset.zip. The dataset is based on data from the following two sources:

University of Michigan Sentiment Analysis competition on Kaggle Twitter Sentiment Corpus by Niek Sanders

```
def remap_labels(sample):
    if sample['feeling'] == 0:
        return { 'label': 0, 'label_text': 'negative' }
    else:
        return { 'label': 1, 'label_text': 'positive' }
```

```
dataset = dataset.map(remap_labels)
```

Zone d'incrustation vidéo

Une solution

```
def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions    = np.argmax(logits, axis = -1)
    return metric.compute(predictions = predictions, references = labels, average = 'macro')

def tokenize_function(examples):
    return tokenizer(
        clean_text(examples["text"]),
        padding        = "max_length",
        truncation     = True
    )
```

```
dataset = dataset.map(remap_labels)
tokenizer.add_tokens(['<url>', '<email>', '<phone>', '<number>', '<digit>', '<cur>', 'user'])
model.resize_token_embeddings(len(tokenizer))

tokenized_dataset   = dataset.map(tokenize_function, batched = True)
small_train_dataset = tokenized_dataset["train"].shuffle().select(range(1000))
small_eval_dataset  = tokenized_dataset["test"].shuffle().select(range(1000))

training_args     = TrainingArguments(
    output_dir          = "test_trainer",
    evaluation_strategy = "epoch"
)
metric = evaluate.load('recall')

trainer = Trainer(
    model           = model,
    args            = training_args,
    train_dataset   = small_train_dataset,
    eval_dataset    = small_eval_dataset,
    compute_metrics = compute_metrics
)

trainer.train()
```

Zone d'incrustation vidéo

Résultats

Epoch	Training Loss	Validation Loss	Recall
1	No log	0.519062	0.759178
2	No log	0.955433	0.744959
3	No log	1.266704	0.739485

	Emoji	Emotion	Hate	Irony	Offensive	Sentiment	Stance	Topic	NER
SVM	29.3	64.7	36.7	61.7	52.3	62.9	67.3	30.5	-
fastText	25.8	65.2	50.6	63.1	73.4	62.9	65.4	24.0	-
BLSTM	24.7	66.0	52.6	62.8	71.7	58.3	59.4	27.0	-
RoB-Base	30.9	76.1	46.6	59.7	79.5	71.3	68.0	50.1	58.0
RoB-Twitter	29.3	72.0	46.9	**65.4**	77.1	69.1	66.7	-	-
TweetEval	31.4	78.5	52.3	61.7	80.5	72.6	69.3	56.8	56.8
BERTweet	33.4	79.3	**56.4**	*82.1**	79.5	73.4	71.2	52.7	58.7
TweetNLP (TimeLMs-21)	**34.0**	**80.2**	55.1	64.5	**82.2**	**73.7**	**72.9**	**58.8**	**59.7**
Evaluation metric	M-F1	M-F1	M-F1	$F^{(i)}$	M-F1	M-Rec	AVG (F)	M-F1	M-F1

digital-learning certifiant

Zone d'incrustation vidéo

Partie 5.4 : Cas pratiques de génération de texte

Skills4All
digital-learning certifiant

5.4 – Cas pratiques de génération de texte

- Objectif : automatiser le support client d'un site web
  - Un modèle de génération de texte mis à disposition du public ne peut être laissé sans contraintes
    - Il pourrait très rapidement être abusé par les clients pour dire des choses incriminantes pour la réputation de l'entreprise
    - Il pourrait répondre des choses gênantes
    - La liberté de répondre n'importe quoi rendrait le modèle inefficace pour résoudre des problèmes concrets.
    - Conséquence : les problèmes de chat-bots sont recadrés comme des problèmes de Q/A ou classification

Zone d'incrustation vidéo

Skills4 Digital Learning Partnant

5.4 – Cas pratiques de génération de texte

- **Objectif : automatiser le support client d'un site web**
  - La notice d'utilisation du site formellement écrite et le modèle répond aux questions sur le texte
  - Un nombre fini de réponses est listé et le modèle classifie les questions de l'utilisateur et réponds la solution concernant la catégorie
  - => Les modèles de génération libres de texte mis à disposition du public n'existent pas

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

↗️ Expérimentions avec sur le Q/A

```
To create a new account you need to : click on the sign up on the top
right of the website; then fill the form with your full name, email
address, shipping address, credit card and credit card information;
then verify your email address by following the instructions in the
email you've just received
```

```python
import torch
from transformers import AutoTokenizer, AutoModelForQuestionAnswering, pipeline

model_name = "mrm8488/longformer-base-4096-finetuned-squadv2"
tokenizer  = AutoTokenizer.from_pretrained(model_name)
model      = AutoModelForQuestionAnswering.from_pretrained(model_name)
qa         = pipeline("question-answering", model = model, tokenizer = tokenizer)
```

```python
textbook = "To create a new account you need to : click on the sign up on the top right
question = "I've just filled the form but my account is not created. What should I do?"

qa({"question": question, "context": textbook})
```

```
{'score': 7.429386528201576e-07,
 'start': 211,
 'end': 236,
 'answer': 'verify your email address'}
```

Zone d'incrustation vidéo

5.4 – Cas pratiques de génération de texte

Support client automatisé par classification

- **Customer Support Intent Dataset**

  bitext/customer-support-intent-dataset

Categories and Intents

The categories and intents covered by the dataset are:

- ACCOUNT: create_account, delete_account, edit_account, recover_password, registration_problems, switch_account
- CANCELLATION_FEE: check_cancellation_fee
- CONTACT: contact_customer_service, contact_human_agent
- DELIVERY: delivery_options, delivery_period
- FEEDBACK: complaint, review
- INVOICE: check_invoice, get_invoice
- NEWSLETTER: newsletter_subscription,
- ORDER: cancel_order, change_order, place_order, track_order
- PAYMENT: check_payment_methods, payment_issue
- REFUND: check_refund_policy, get_refund, track_refund
- SHIPPING_ADDRESS: change_shipping_address, set_up_shipping_address

Zone d'incrustation vidéo

↗ Custom Support Intent Dataset

Each utterance is enriched with one or more of these tags:

- Register tags: colloquial language, polite language…
  - Q - Colloquial variation
  - P - Politeness variation
- Content tags: offensive language, keyword language…
  - W - Offensive language
  - K - Keyword language
- Linguistic tags: syntactic and morphological tags (interrogative sentence, coordinated sentence…)
  - B - Basic syntactic structure
  - C - Coordinated syntactic structure
  - I - Interrogative structure
  - M - Morphological variation (plurals, tenses…)
  - L - Lexical variation (synonyms)
  - E - Expanded abbreviations (I'm -> I am, I'd -> I would…)
- Real-life errors: spelling errors, punctuation errors…
  - Z - Noise phenomena like spelling or punctuation errors

Zone d'incrustation vidéo

Formatter la base de données

```
from datasets import load_dataset

dataset = load_dataset('bitext/customer-support-intent-dataset')
dataset['train'][0]
```

```
{'utterance': 'would it be possible to cancel the order I made?',
 'intent': 'cancel_order',
 'category': 'ORDER',
 'tags': 'BIP'}
```

```
labels = set()
def map_labels(sample):
    labels.add(sample['intent'])
    labels.add(sample['category'])

    tags = sample['tags']
    for c in tags:
        labels.add(c)

dataset.map(map_labels)

index     = 0
label_map = dict()
for label in labels:
    label_map[label] = index
    index            += 1
```

```
def remap_labels(sample):
    labels = [0] * len(label_map)
    labels[label_map[sample['intent']]]   = 1
    labels[label_map[sample['category']]] = 1

    tags = sample['tags']
    for c in tags:
        labels[label_map[c]] = 1

    return { 'label': labels }

dataset = dataset.map(remap_labels)
```

Zone d'incrustation vidéo

Transformer pour classification multilabels

```
class MultiLabelDistilBERTClass(nn.Module):
    def __init__(self, num_labels):
        super(MultiLabelDistilBERTClass, self).__init__()
        self.distilbert      = DistilBertModel.from_pretrained('distilbert-base-uncased')
        self.hidden          = nn.Linear(768, 768)
        self.hidden_transfer = nn.ReLU()
        self.dropout         = nn.Dropout(0.3)
        self.output          = nn.Linear(768, num_labels)

    def forward(self, x):
        input_ids      = x[0]
        attention_mask = x[1]
        x = self.distilbert(
            input_ids      = input_ids,
            attention_mask = attention_mask
        )

        x = x[0][:, 0]
        x = self.hidden(x)
        x = self.hidden_transfer(x)
        x = self.dropout(x)
        x = self.output(x)
        return x
```

```
model     = MultiLabelDistilBERTClass(len(labels))
optimizer = optim.Adam(
    params = model.parameters(),
    lr     = learning_rate
)
loss_func = nn.BCEWithLogitsLoss()
```

Zone d'incrustation vidéo

Résultats

```
train(
  model, loss_func, optimizer, train_dataset,
  test_dataset, epochs, batch_size
)
```

```
0 -> 52.363% train accuracy   0 -> 48.289% test accuracy
1 -> 52.378% train accuracy   1 -> 48.289% test accuracy
2 -> 52.378% train accuracy   2 -> 48.289% test accuracy
```

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Partie 6 : Conclusions

Skills4All
digital-learning certifiant

6 – Conclusions

- Thématiques non abordées (1/2)
  - Surapprentissage (overfitting)
    - Technique d'arrêt prématuré (early stopping), méthodes de régularisation (dropout, batch-normalization)
  - Initialisation d'un réseau de neurone
  - Entraînements résistants aux perturbations et aux transformations de données
    - Augmentation des données d'entrée
  - Entraînements résistants aux attaques informatiques
  - Étude et correction des biais des modèles

Zone d'incrustation vidéo

Skills4

6 – Conclusions

- Thématiques non abordées (2/2)
  - Data Science : comment analyser des données
  - Mise en production & scalabilité des modèles de machine learning
  - L'utilisation des structures, Pytorch, Scikit-learn et Python
  - Et bien d'autre … Le Machine Learning est un sujet vaste !

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

6 – Conclusions

- Usage de cette formation
  - Un portail dans le monde du NLP, chaque point peut être approfondi
  - Un point de départ dans l'implémentation du Machine Learning → rien ne remplace des années de pratique
  - Un cadre théorique pour comprendre les nouvelles notions que vous rencontrerez
  - Un cadre historique des technologies (les dernières ne sont pas forcément les plus adaptées pour tous les problèmes)

Zone d'incrustation vidéo

Skills4
digital-learning certifiant

Skills4All
digital-learning certifiant

01 85 39 09 73
contact@skills4all.com
54/56, avenue Hoche
75008 PARIS