Bases de NLP

Sylvain Hazard

18/11/2019

Wiidii

Bordeaux INP ENSC

Bordeaux INP ENSEIRB MATMECA

Sommaire

- Introduction
- Du texte aux features
- L'état de l'art

Introduction

De quoi on parle ?

Natural Language Processing
=
Traitement Automatique du Langage (Naturel)

Neuro-Linguistic Programming
=
Programmation Neuro-Linguistique

Bref historique

Années 1950	Années 1970	Années 1980	Aujourd’hui
Alan Turing	Travaux basés sur des ontologies	Utilisation du Machine Learning	Deep Learning
Premiers travaux de traduction automatique	Ensembles de règles	Arbres de décision	

Tâches

Natural Language Generation

- Summarization
- Simplification
- Translation

Natural Language Understanding

- QA
- NER
- Classification
- Coreference Resolution

Spoken Language

- Text-to-speech
- Speech-to-text

Pourquoi c'est si difficile ?

1. Le langage est infini et évolue constamment.
2. Ce qui fonctionne dans une langue ne fonctionnera pas forcément dans une autre.
3. Les humains n'utilisent pas correctement leur propre langue.
4. Le texte ne se suffit pas toujours.
5. La quantité de donnée annotée est faible.

Passer du texte à des données utiles

Ou comment on tente de résoudre ces problèmes

Etape 1 : Tokenization

"Ceci est une phrase simple à tokenizer."

["Ceci", "est", "une", "phrase", "simple", "à", "tokenizer", "."]

"C'est moins simple de tokenizer celle-là."

Tokenization

["C'est", "moins", "simple", "de", "tokenizer", "celle-là", "."]

["C' ", "est", "moins", "simple", "de", "tokenizer", "celle", "-là", "."]

["C", " ' ", "est", "moins", "simple", "de", "tokenizer", "celle", "-", "là", "."]

Etape 2 : Passer des tokens aux nombres

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ultrices nibh fermentum orci blandit, eu mollis eros ornare. Nam id egestas sapien. Aenean fermentum, risus a bibendum dictum, lacus urna ultricies turpis, ac malesuada dui quam eu sem. Morbi nulla odio, tempor eget elit a, finibus auctor purus. Suspendisse id urna nec mauris consequat rhoncus. Cras rutrum est arcu, non volutpat odio venenatis nec. Etiam at feugiat dolor. Nulla porttitor nulla sed posuere pharetra. Etiam feugiat, odio vel laoreet imperdiet, enim lacus bibendum purus, et accumsan nisi tortor et lectus. Aliquam tincidunt dui non libero vulputate porta. Sed imperdiet, dolor ut molestie eleifend, massa augue pretium lacus, ac bibendum mauris lorem at ligula. Nullam mauris libero, suscipit eget orci sit amet, auctor vulputate quam. Mauris orci sem, condimentum sodales nulla

54 23 167 895 12 631 45 89 65 122 3549 657 12 0 33 65 89 78 451 21 321 56 9 8 45 651 20 35 468 987 65 123 456 789 956 321 45 878 12 65 34 48 1321 45 98 7 54 23 167 895 12 631 45 89 65 122 3549 657 12 0 33 65 89 78 451 21 321 56 9 8 45 651 20 35 468 987 65 123 456 789 956 321 45 878 12 65 34 48 1321 45 98 7 54 23 167 895 12 631 45 89 65 122 3549 657 12 0 33 65 89 78 451 21 321 56 9 8 45 651 20 35 468 987 65 123 456 789 956 321 45 878 12 65 34 48 1321 45 98 754 23 167 895 12 631 45 89 65 122 3549 657 12 0 33 65 89 78 451 21 321 56 9 8 45 651 20 35 468 987 65 123 456 789 956 321 45 878 12 65 34 48 1321 45 98 754 23 167 895 12 631 45 89 65 122 3549 657 12 0 33 65 89 78 451 21 321 56 9 8 45 651 20 35 468 987 65 123 456 789 956 321 45 878 12 65 34 48 1321 45 98 7

Etape 2 - Méthode 1 : Indexation

Input
Tout le monde aime les phrases de test.

Vocabulaire

tout : 1
le : 2
chat : 3
aime : 4
les : 5
test : 6
cornemuse : 7
...

Output

1
2
12
4
5
27
34
6
21

Etape 2 - Méthode 2 : Word Vectors

Input
Tout le monde aime les phrases de test.

Vocabulaire
tout : [0.23, 0.47, 0.69, 0.12]
le : [0.26, 0.65, 0.98, 0.57]
chat : [0.36, 0.79, 0.21, 0.54]
aime : [0.75, 0.97, 0.42, 0.45]
les : [0.76, 0.21, 0.40, 0.22]
test : [0.02, 0.94, 0.23, 0.47]
cornemuse : [0.56, 0.78, 0.46, 0.12]
...

Output

[0.23, 0.47, 0.69, 0.12]
[0.26, 0.65, 0.98, 0.57]
[0.69, 0.12, 0.01, 0.65]
[0.75, 0.97, 0.42, 0.45]
[0.76, 0.21, 0.40, 0.22]
[0.65, 0.23, 0.99, 0.67]
[0.34, 0.54, 1.00, 0.46]
[0.02, 0.94, 0.23, 0.47]
[0.45, 0.16, 0.94, 0.21]

Etape 2 - Méthode 3 : Sentence Vectors

Input
Tout le monde aime les phrases de test.

Vocabulaire
tout : [0.23, 0.47, 0.69, 0.12]
le : [0.26, 0.65, 0.98, 0.57]
chat : [0.36, 0.79, 0.21, 0.54]
aime : [0.75, 0.97, 0.42, 0.45]
les : [0.76, 0.21, 0.40, 0.22]
test : [0.02, 0.94, 0.23, 0.47]
cornemuse : [0.56, 0.78, 0.46, 0.12]
...

Output

[0.23, 0.47, 0.69, 0.12]
[0.26, 0.65, 0.98, 0.57]
[0.69, 0.12, 0.01, 0.65]
[0.75, 0.97, 0.42, 0.45]
[0.76, 0.21, 0.40, 0.22]
[0.65, 0.23, 0.99, 0.67]
[0.34, 0.54, 1.00, 0.46]
[0.02, 0.94, 0.23, 0.47]
[0.45, 0.16, 0.94, 0.21]

[0.46, 0.48, 0.63, 0.42]

Etape 3 : Donner du sens aux vecteurs

Etape 3 - Méthode 1 : Préprocessing

Input
Cette phrase est étrangement inutile, n'est-ce pas ?

Majuscules et ponctuation

cette phrase est étrangement inutile n est ce pas

Diacritiques

cette phrase est etrangement inutile n est ce pas

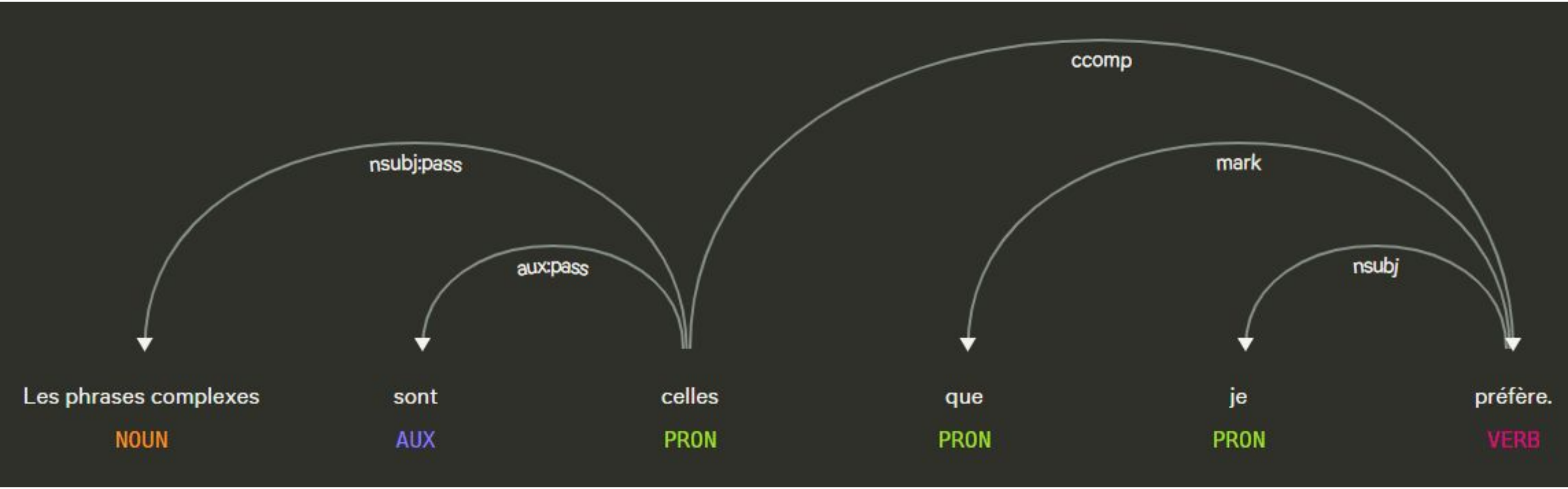
Stop words

phrase etrangement inutile

Etape 3 - Méthode 2 : Lemmatization/Stemming

Input		Output
Input Je suis surprenamment étonné.	**Stemming**	**Output** Je suis surpr eton.
Input Je suis surprenamment étonné.	**Lemmatization**	**Output** Je être surprendre étonner.

Etape 3 - Méthode 3 : POS Tagging



Source : DisplaCy

Etape 3 - Méthode 4 : Simplifier le vocabulaire

Input
Réserve une table au restaurant le Sapajou à Bordeaux jeudi 21 novembre à 20h.

Named Entity Recognition

Output
Réserve une table au restaurant le Sapajou à VILLE DATE à HEURE.

Etape 3 - Méthode 5 : TF-IDF

Input
Tout le monde aime les phrases de test.

Vocabulaire

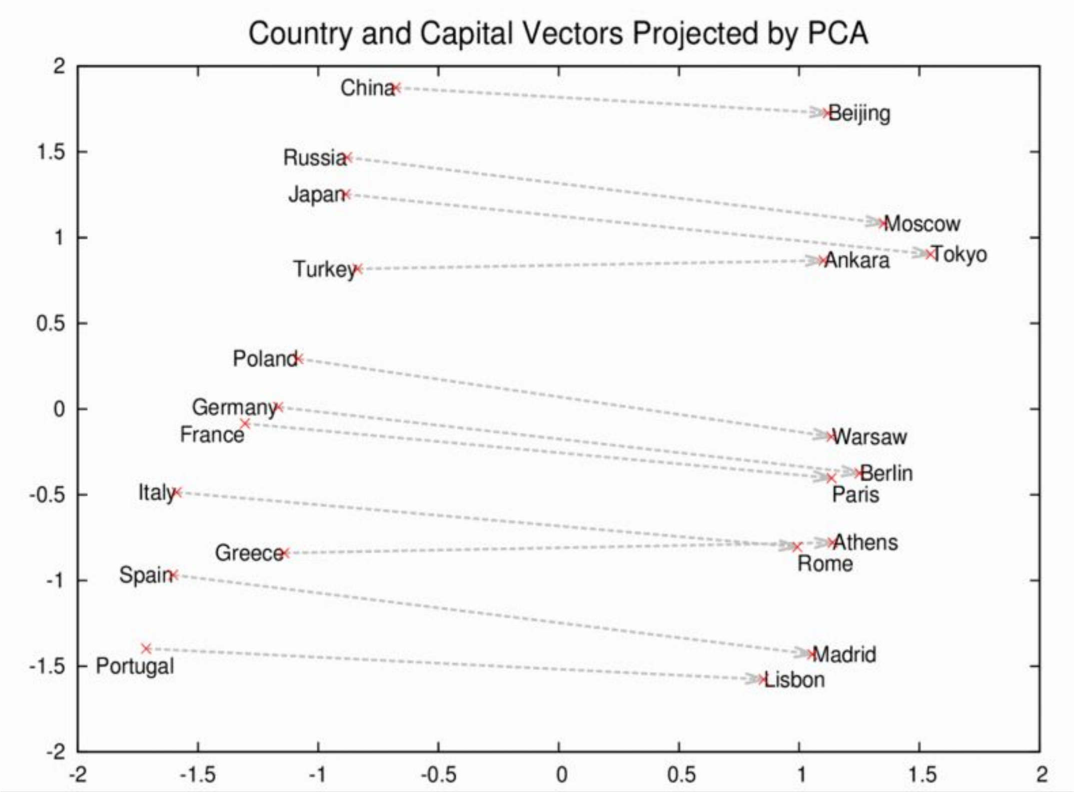
TF-IDF
tout : 10
le : 8
chat : 4
aime : 6
les : 8
test : 9
cornemuse : 27
...

Output

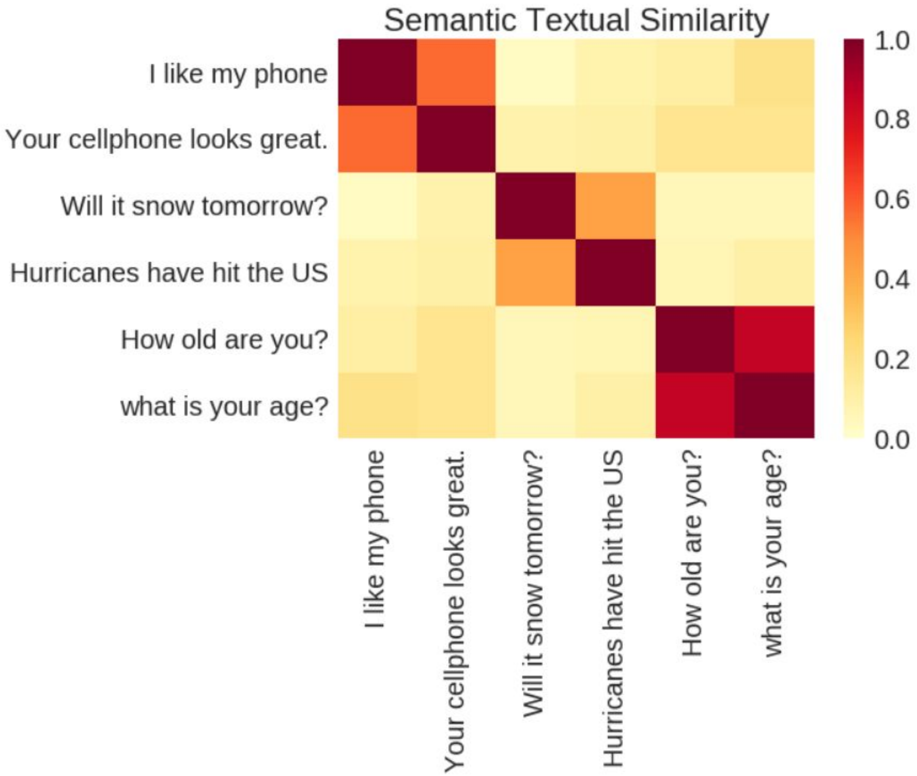
[0.23, 0.47, 0.69, 0.12] * 1/10
[0.26, 0.65, 0.98, 0.57] * 1/8
[0.69, 0.12, 0.01, 0.65] * 1/21
[0.75, 0.97, 0.42, 0.45] * 1/6
[0.76, 0.21, 0.40, 0.22] * 1/8
[0.65, 0.23, 0.99, 0.67] * 1/2
[0.34, 0.54, 1.00, 0.46] * 1/40
[0.02, 0.94, 0.23, 0.47] * 1/26
[0.45, 0.16, 0.94, 0.21] * 1/9

[0.58, 0.44, 0.80, 0.51]

Etape 3 - Méthode 6 : Pre-trained word vectors



Etape 3 - Méthode 7 : Pre-trained sentence vectors



Etape 3 : Comment on fait en vrai ?

Etape 3 : NLTK

NLTK

```
>>> import nltk
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
>>> tagged = nltk.pos_tag(tokens)
>>> tagged[0:6]
[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'),
('Thursday', 'NNP'), ('morning', 'NN')]
```

Etape 3 : SpaCy

```
import spacy

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

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

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

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

spaCy

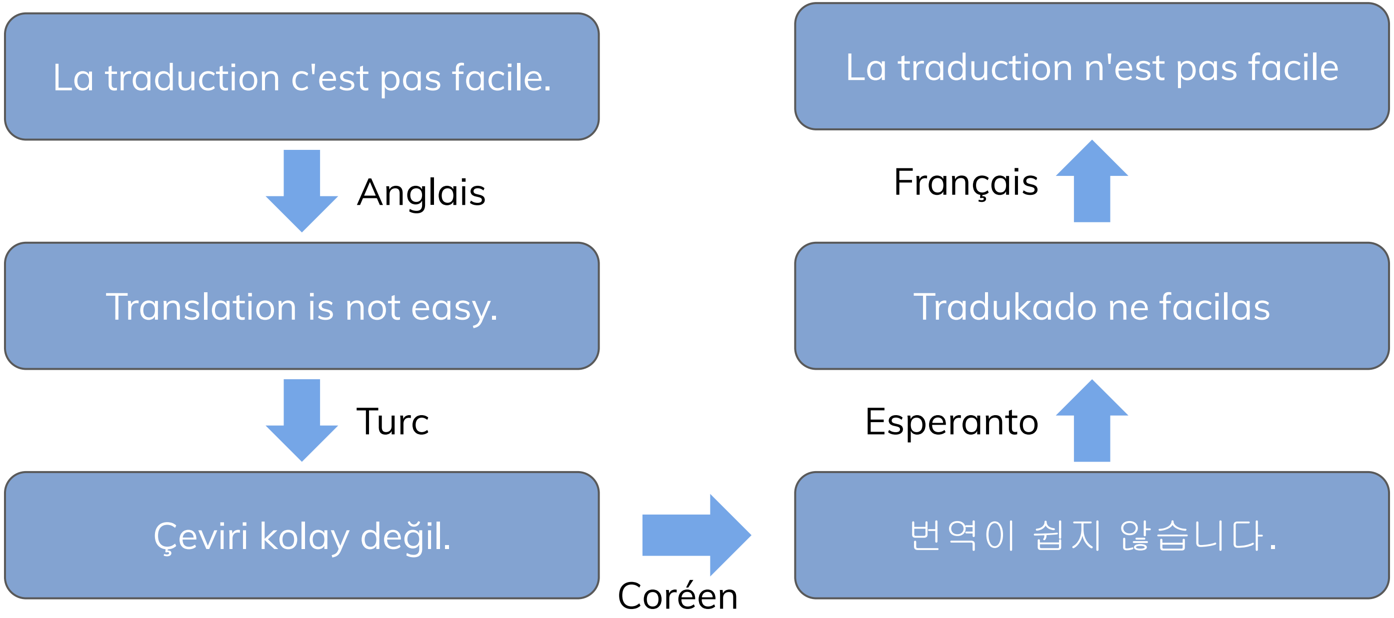
```
Noun phrases: ['Sebastian Thrun', 'self-driving cars', 'Google', 'few people', 'the company', 'h
im', 'I', 'you', 'very senior CEOs', 'major American car companies', 'my hand', 'I', 'Thrun', 'a
n interview', 'Recode']
Verbs: ['start', 'work', 'drive', 'take', 'tell', 'shake', 'turn', 'talk', 'say']
Sebastian Thrun PERSON
Google ORG
2007 DATE
American NORP
Thrun ORG
Recode PRODUCT
earlier this week DATE
```

Etape 4 : Gérer le manque de données

Etape 4 - Méthode 0 : Avoir plus de données

Le Crowdsourcing

Etape 4 - Méthode 1 : Data Augmentation



Etape 4 - Méthode 1 : Data Augmentation

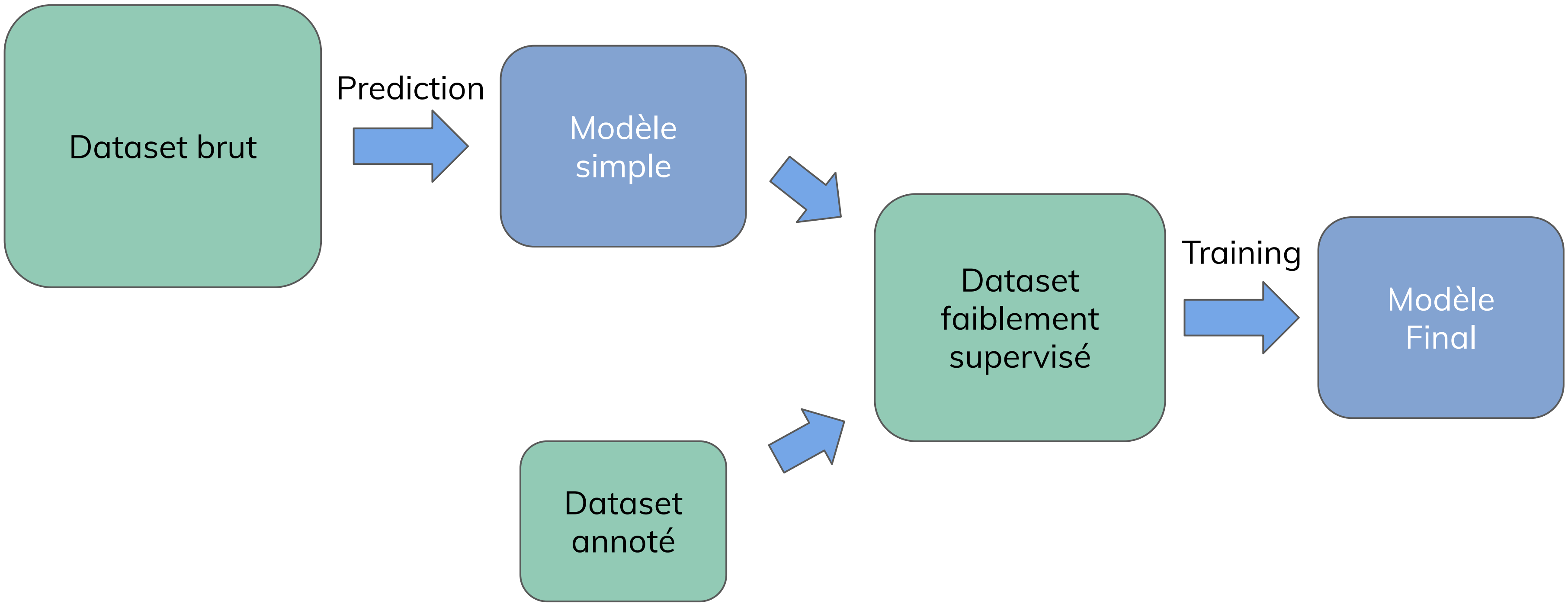
Input
Quand je suis content je vomis.

Input augmenté
Quand je suis heureux je vomis. ✓

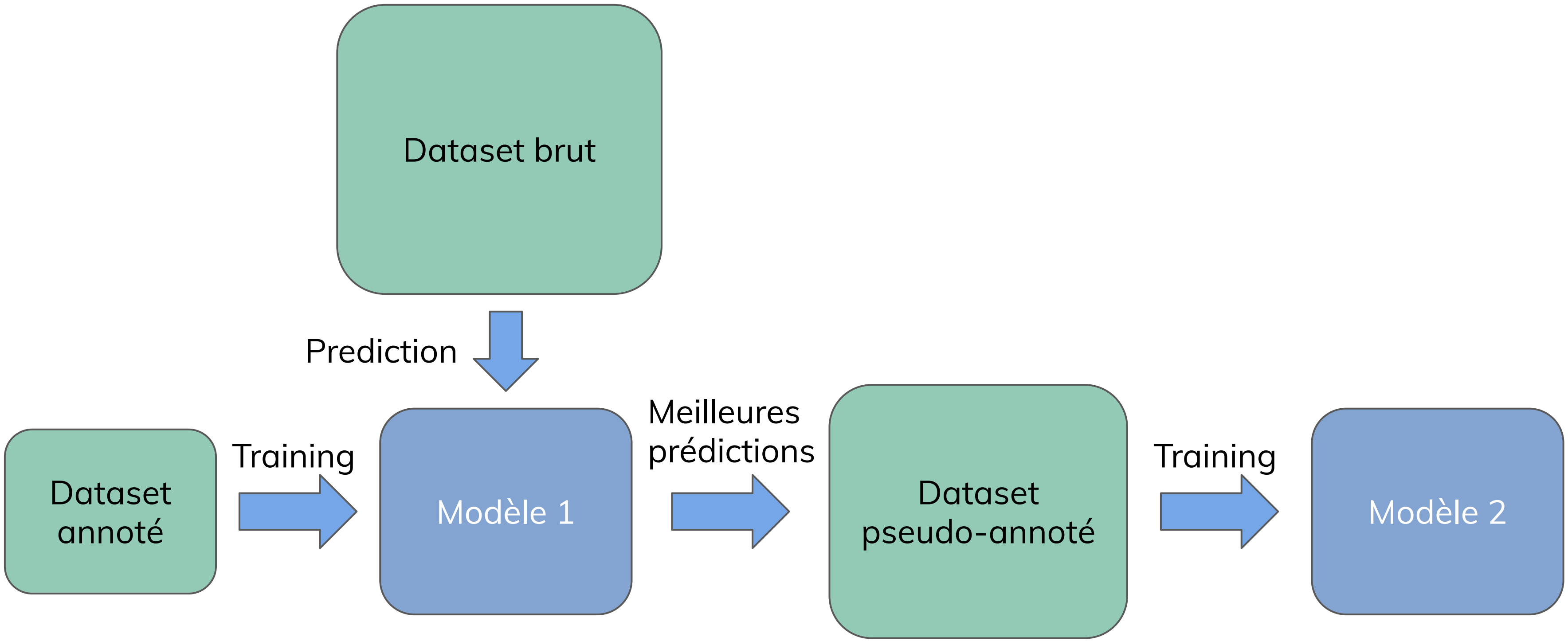
Input augmenté
Quand je suis heureux je dégobille. ✓

Input augmenté
Quand je suis paradisiaque je dégobille. ✗

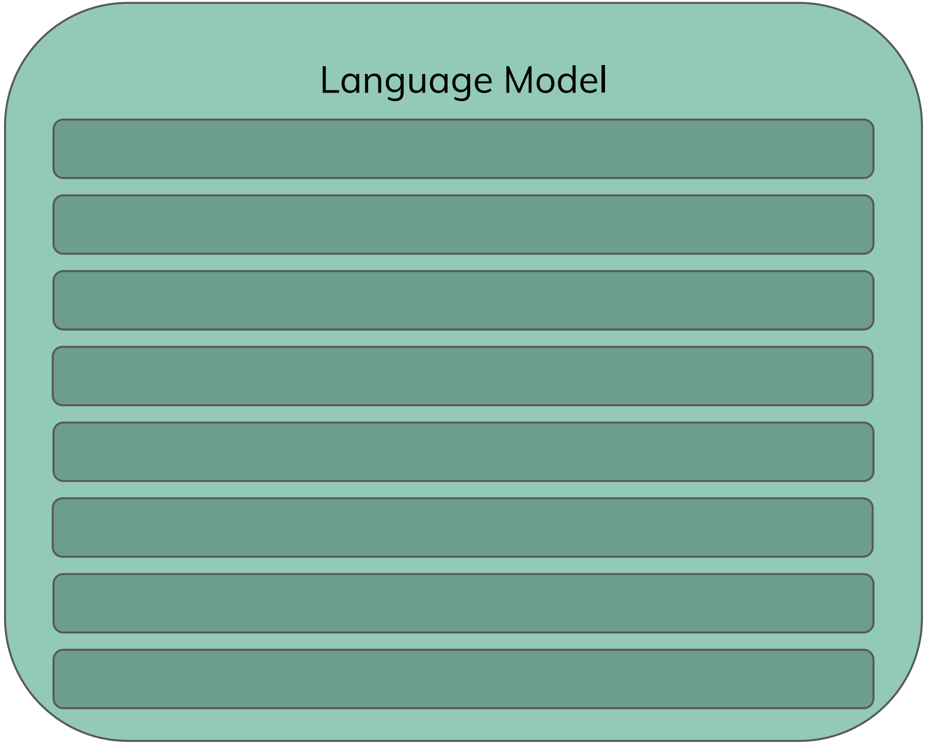
Etape 4 - Méthode 2 : Weak supervision



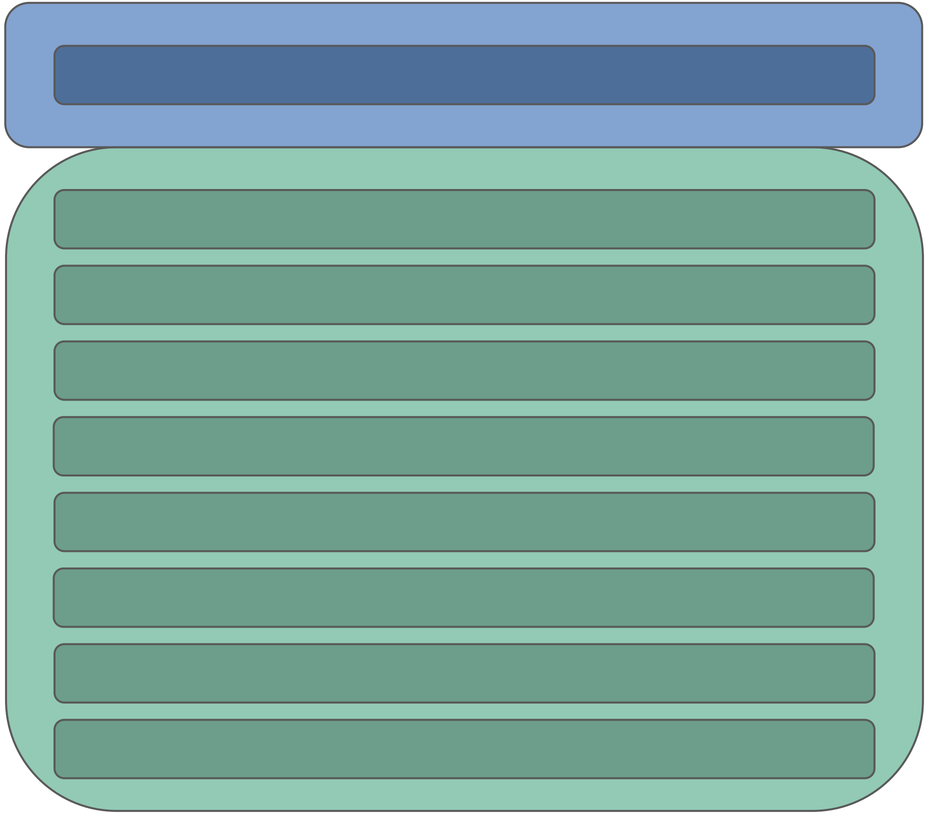
Etape 4 - Méthode 2 : Pseudo-labelling



Etape 4 - Méthode 3 : Transfer Learning



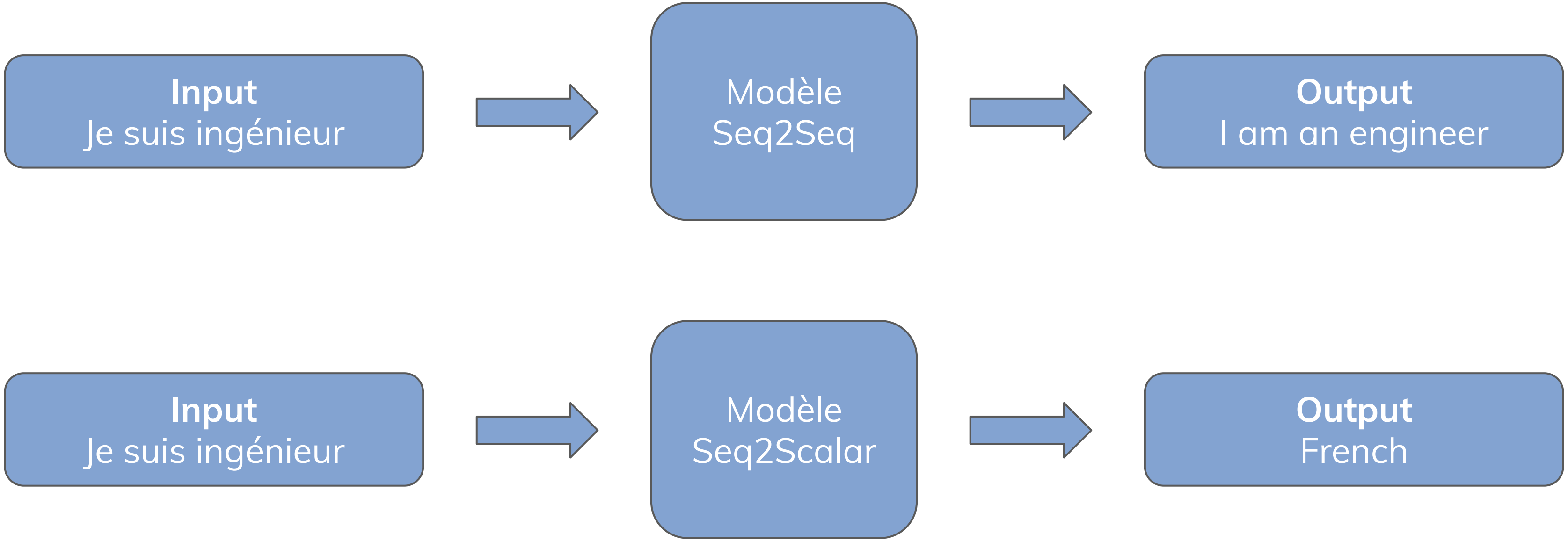
Etape 4 - Méthode 3 : Transfer Learning



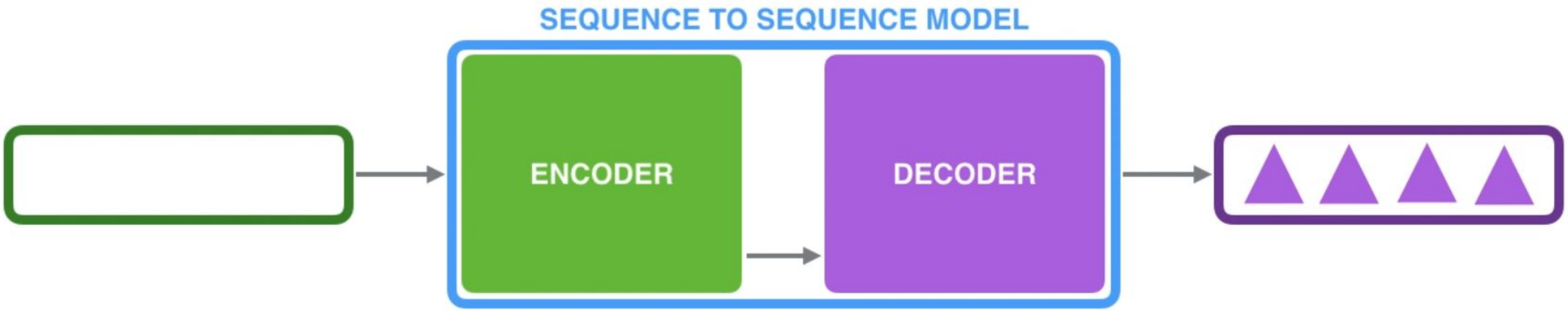
L’Etat de l’art

Qu’est-ce qu’on en fait de tous ces vecteurs ?

Sequence to Sequence ou Sequence to Scalar ?

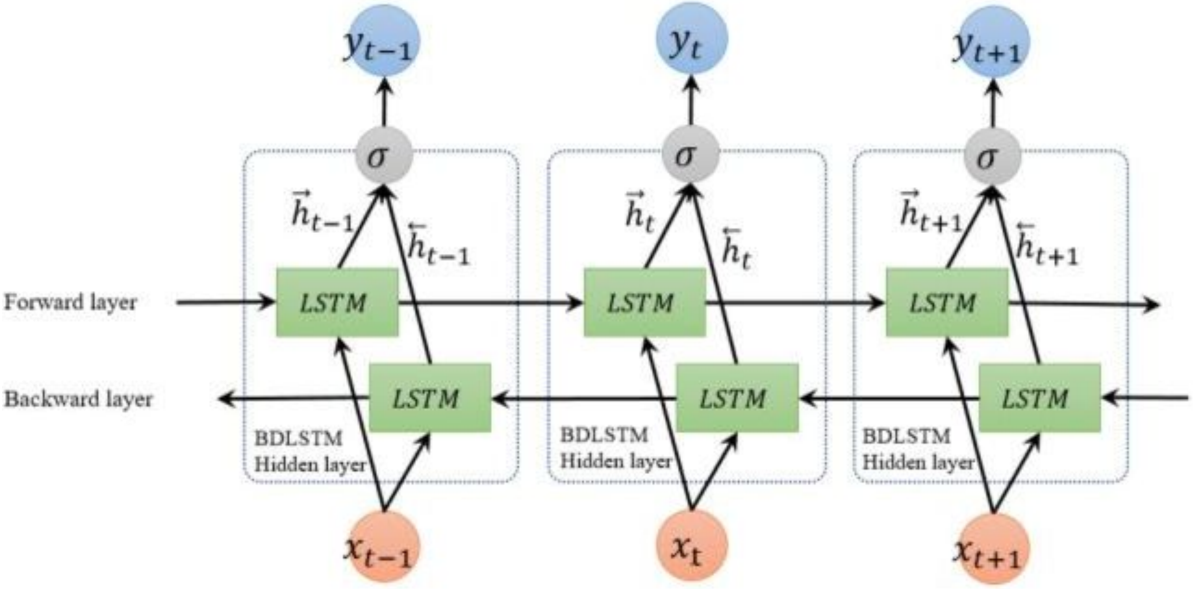


Architecture encoder-decoder

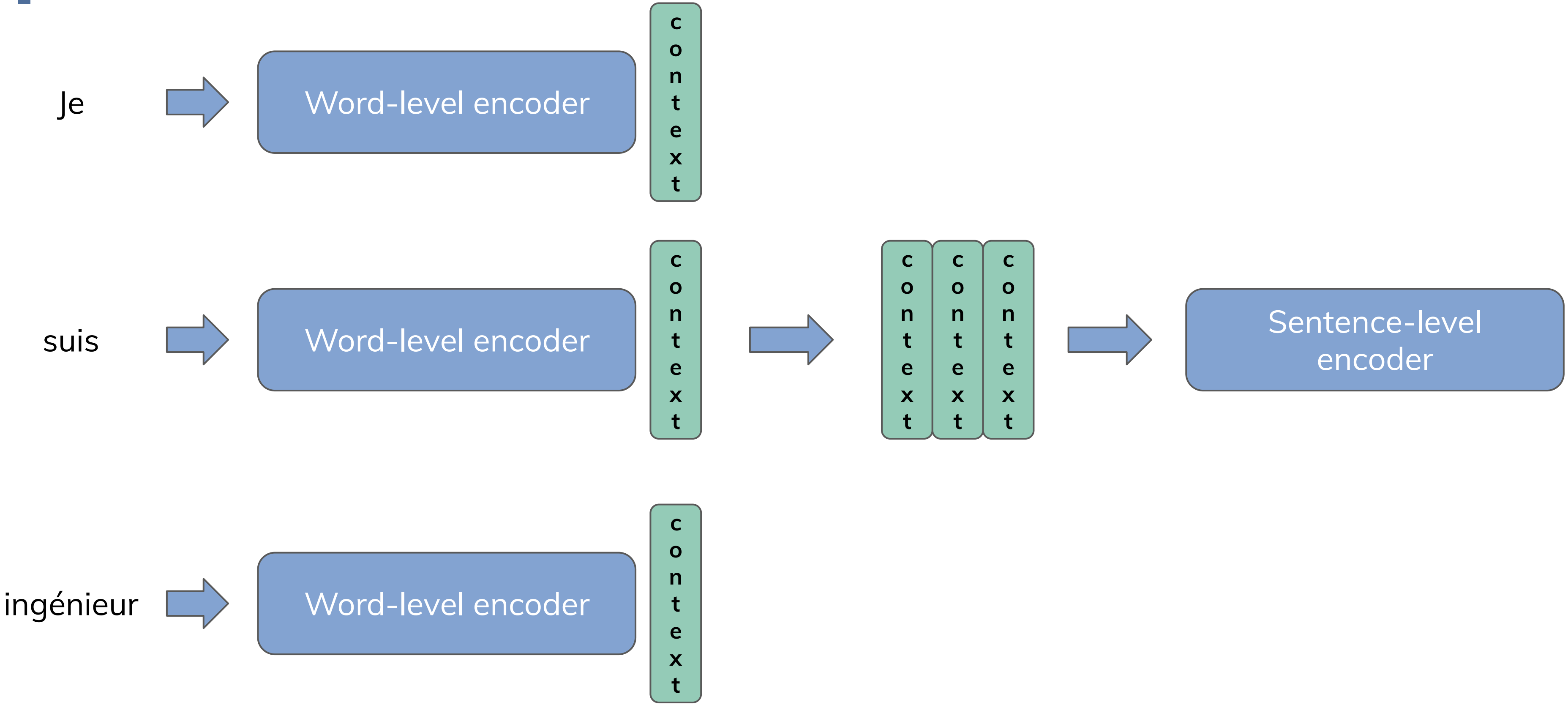


Source : Jay Alammar

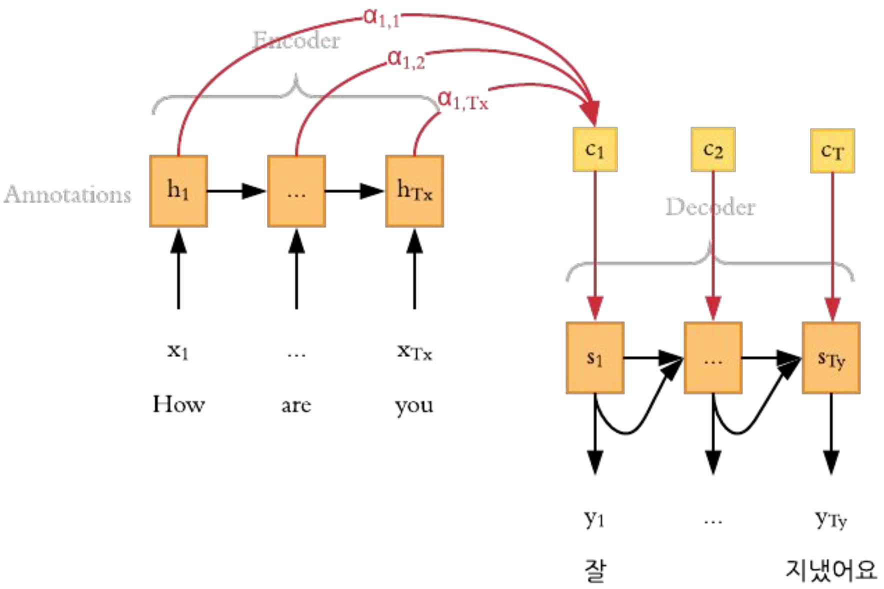
Réseaux de neurones récurrents (LSTM/GRU)



Réseaux de neurones hiérarchiques

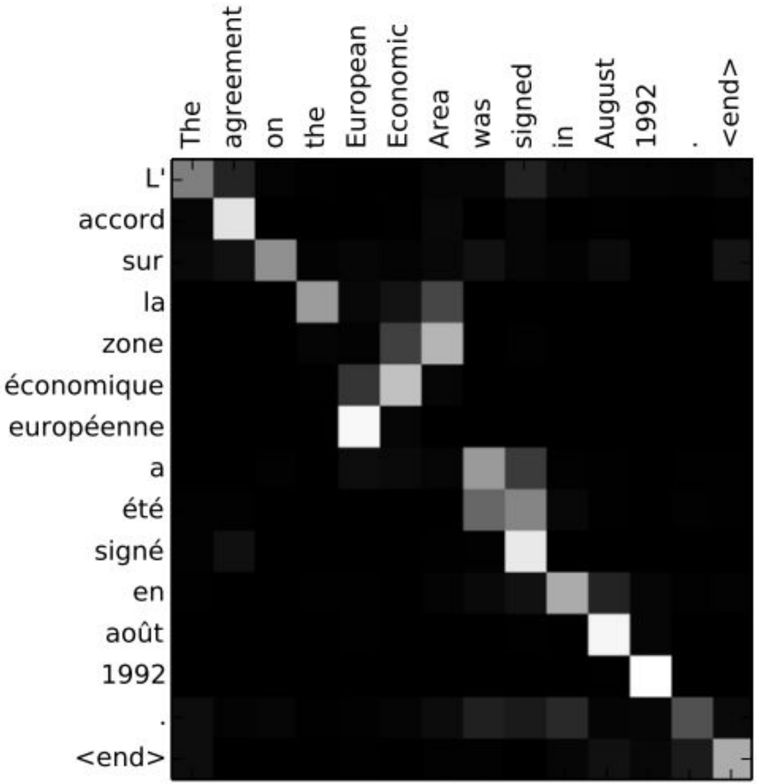


Attention



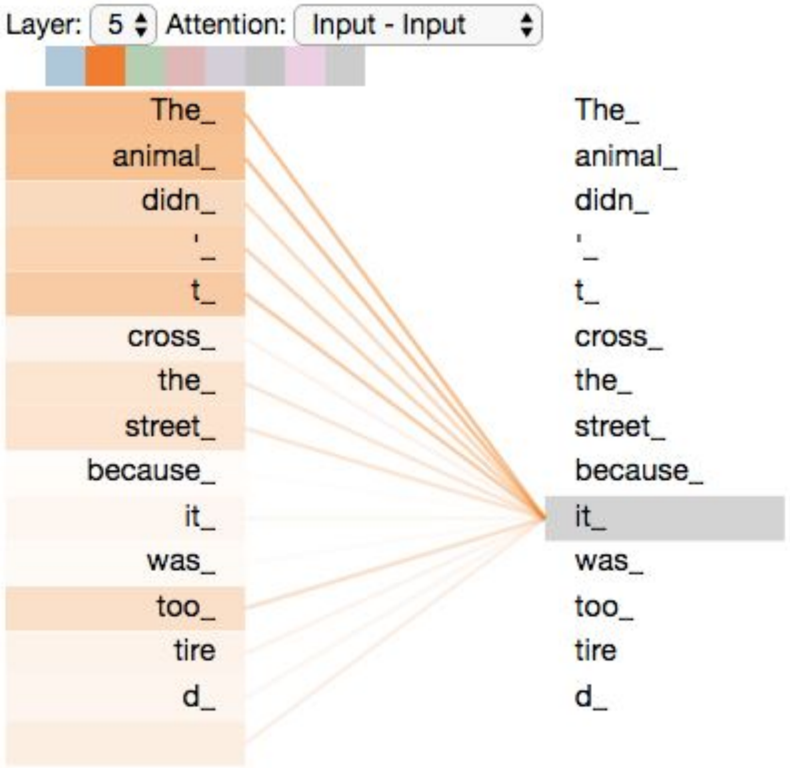
Source : Medium (article par Kate Loginova)

Attention (2)



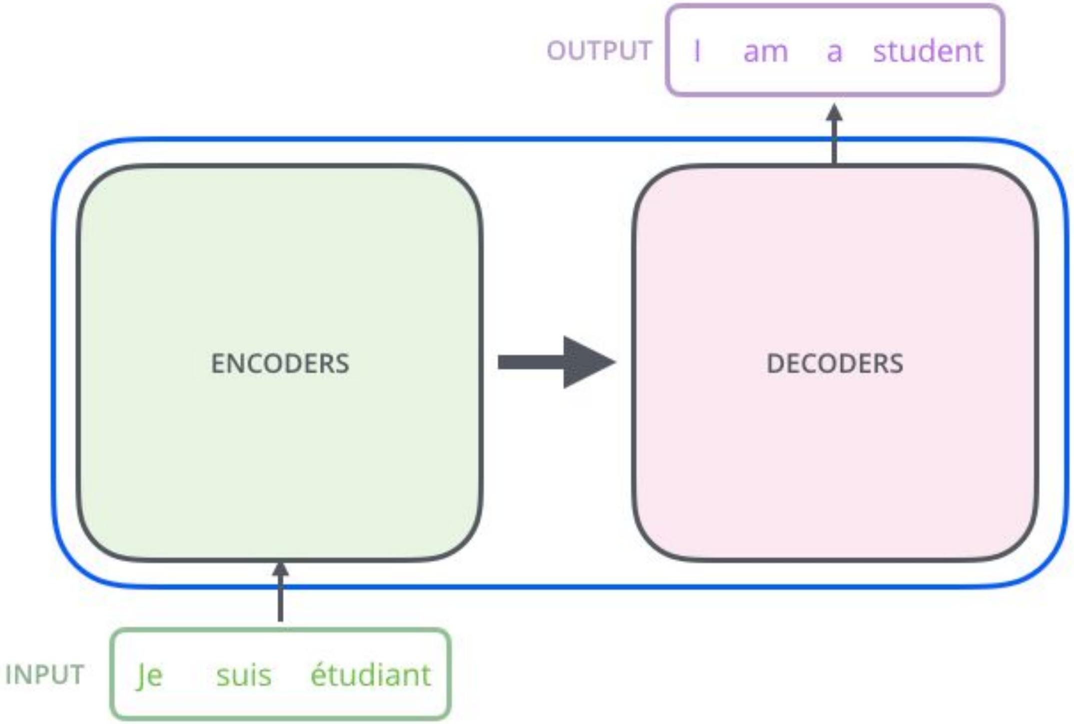
Source : Badhanau et al., 2016

Self-Attention



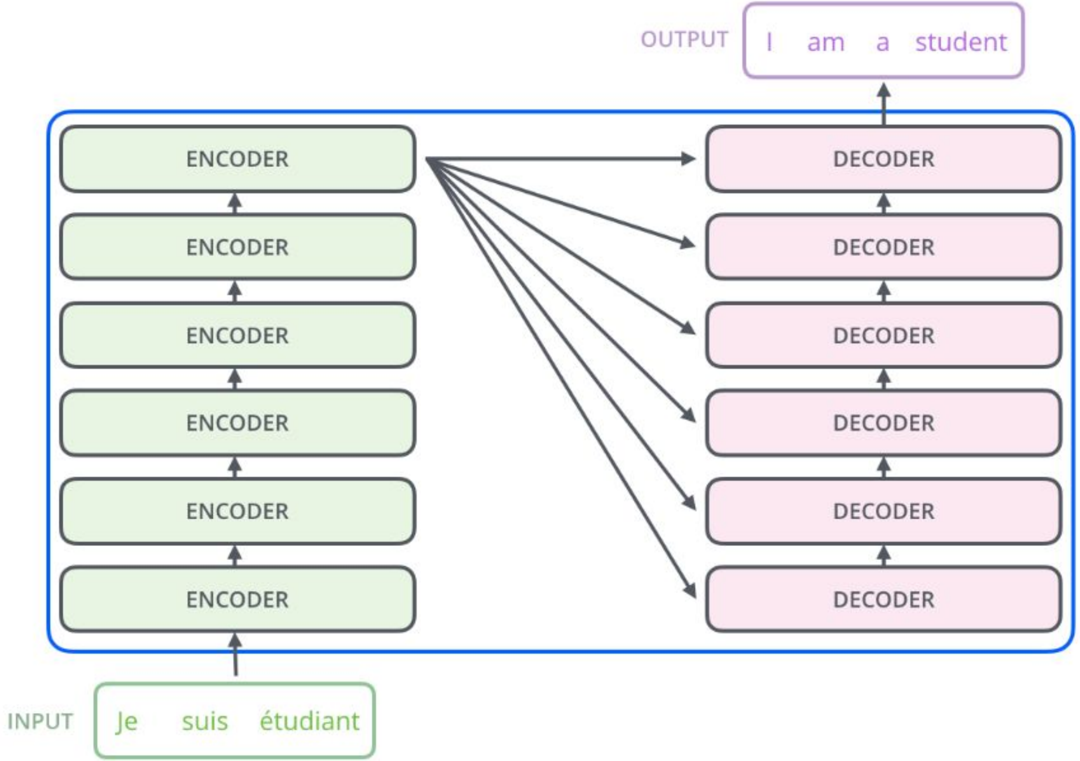
Source : Jay Alammar

Transformers



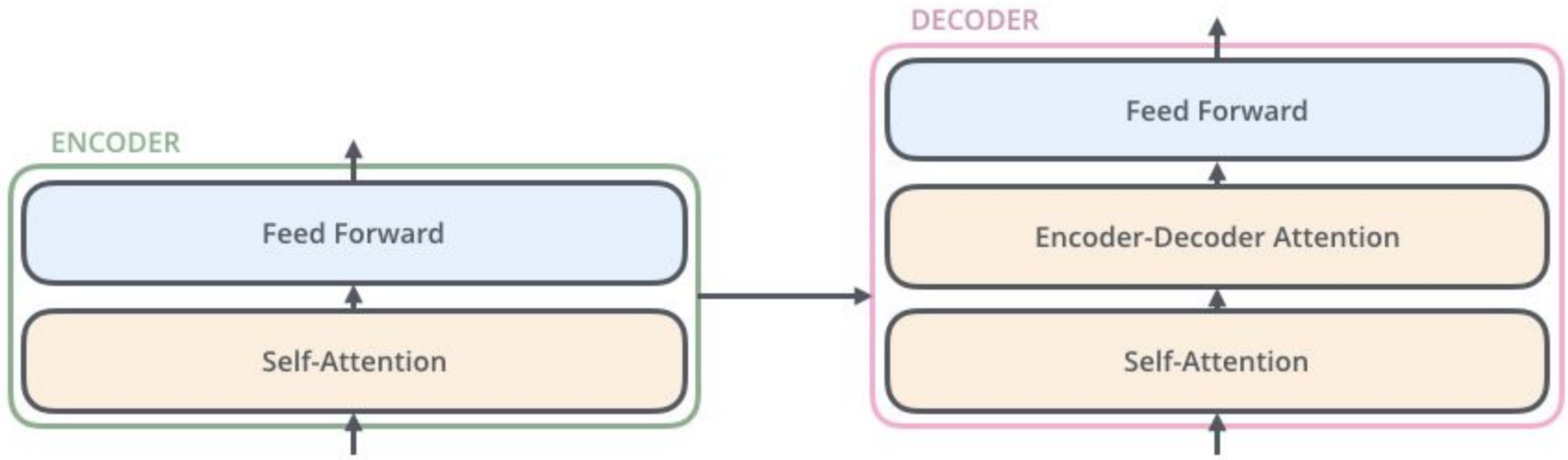
Source : Jay Alammar

Transformers



Source : Jay Alammar

Transformers



Source : Jay Alammar

GPT-2

NLP is a pretty interesting field of research , **but we're getting to the point where the more I study it the more impressed I am with the possibilities it holds. There are so many fascinating things happening in the deep learning research community, and we need to make sure that we keep up with these developments.**

Written by Transformer · transformer.huggingface.co

Côté NLU

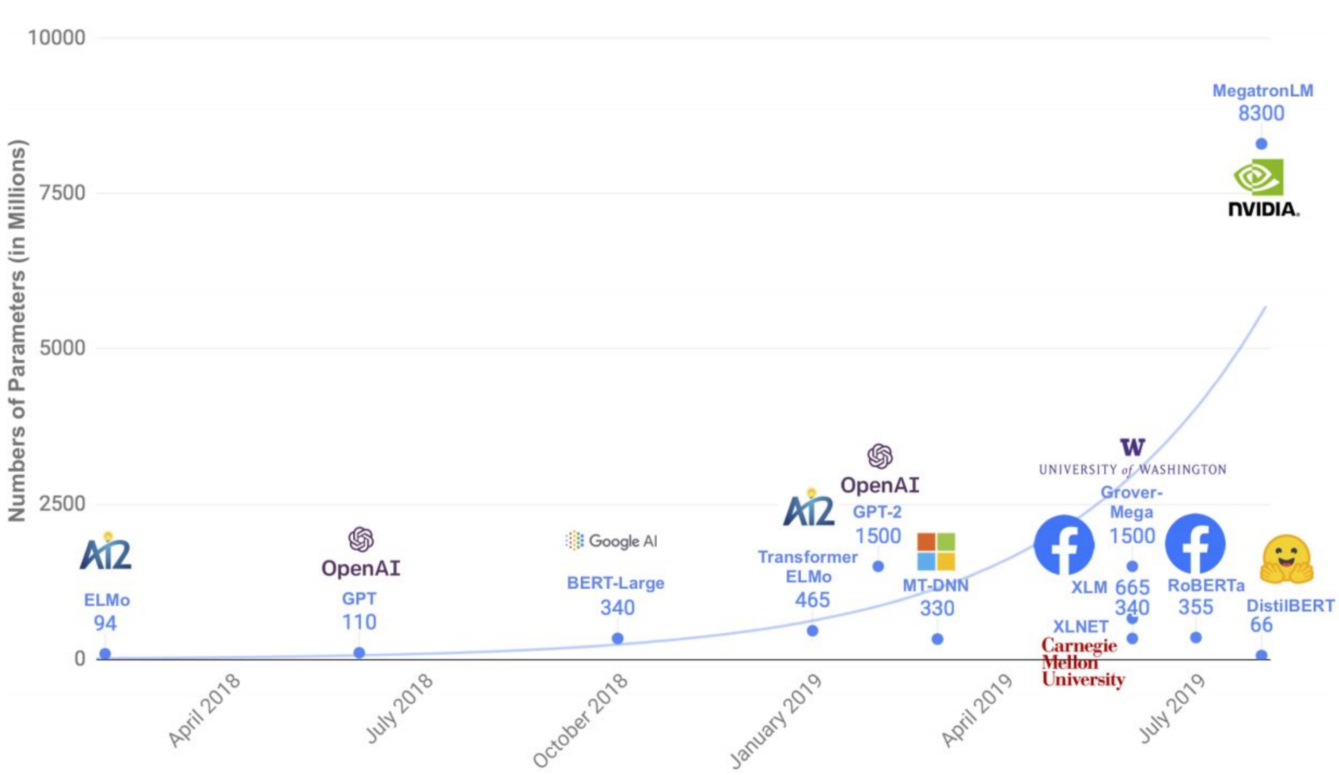
Attention	BERT	MT-DNN	XLNet	RoBERTa	T5
2016	Oct. 2018	Mars 2019	Juin 2019	Juillet 2019	Nov. 2019

Où trouver ces modèles ?

HuggingFace

```
import tensorflow as tf
import tensorflow_datasets
from transformers import *

# Load dataset, tokenizer, model from pretrained model/vocabulary
tokenizer = BertTokenizer.from_pretrained('bert-base-cased')
model = TFBertForSequenceClassification.from_pretrained('bert-base-cased')
data = tensorflow_datasets.load('glue/mrpc')

# Prepare dataset for GLUE as a tf.data.Dataset instance
train_dataset = glue_convert_examples_to_features(data['train'], tokenizer, max_length=128, task='mrpc')
valid_dataset = glue_convert_examples_to_features(data['validation'], tokenizer, max_length=128, task='mrpc')
train_dataset = train_dataset.shuffle(100).batch(32).repeat(2)
valid_dataset = valid_dataset.batch(64)

# Prepare training: Compile tf.keras model with optimizer, loss and learning rate schedule
optimizer = tf.keras.optimizers.Adam(learning_rate=3e-5, epsilon=1e-08, clipnorm=1.0)
loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
metric = tf.keras.metrics.SparseCategoricalAccuracy('accuracy')
model.compile(optimizer=optimizer, loss=loss, metrics=[metric])

# Train and evaluate using tf.keras.Model.fit()
history = model.fit(train_dataset, epochs=2, steps_per_epoch=115,
                    validation_data=valid_dataset, validation_steps=7)

# Load the TensorFlow model in PyTorch for inspection
model.save_pretrained('./save/')
pytorch_model = BertForSequenceClassification.from_pretrained('./save/', from_tf=True)

# Quickly test a few predictions - MRPC is a paraphrasing task, let's see if our model learned the task
sentence_0 = "This research was consistent with his findings."
sentence_1 = "His findings were compatible with this research."
sentence_2 = "His findings were not compatible with this research."
inputs_1 = tokenizer.encode_plus(sentence_0, sentence_1, add_special_tokens=True, return_tensors='pt')
inputs_2 = tokenizer.encode_plus(sentence_0, sentence_2, add_special_tokens=True, return_tensors='pt')

pred_1 = pytorch_model(inputs_1['input_ids'], token_type_ids=inputs_1['token_type_ids'])[0].argmax().item()
pred_2 = pytorch_model(inputs_2['input_ids'], token_type_ids=inputs_2['token_type_ids'])[0].argmax().item()

print("sentence_1 is", "a paraphrase" if pred_1 else "not a paraphrase", "of sentence_0")
print("sentence_2 is", "a paraphrase" if pred_2 else "not a paraphrase", "of sentence_0")
```

Quelle limite à la taille des modèles ?



Source : Sanh et al., 2019

Et du coup, le NLP c'est résolu ?

- Étendre les performances actuelles sur toutes les langues
- Mieux gérer le contexte
- Mieux gérer les connaissances
- Intégrer plusieurs composantes dans un même système end-to-end

Qu'est-ce qu'on fait en tant qu'industriel ?

- Veille
- *Proofs of concept*
- Gestion de l'industrialisation
- Interaction avec les autres équipes
- Maintenance

Des questions ?