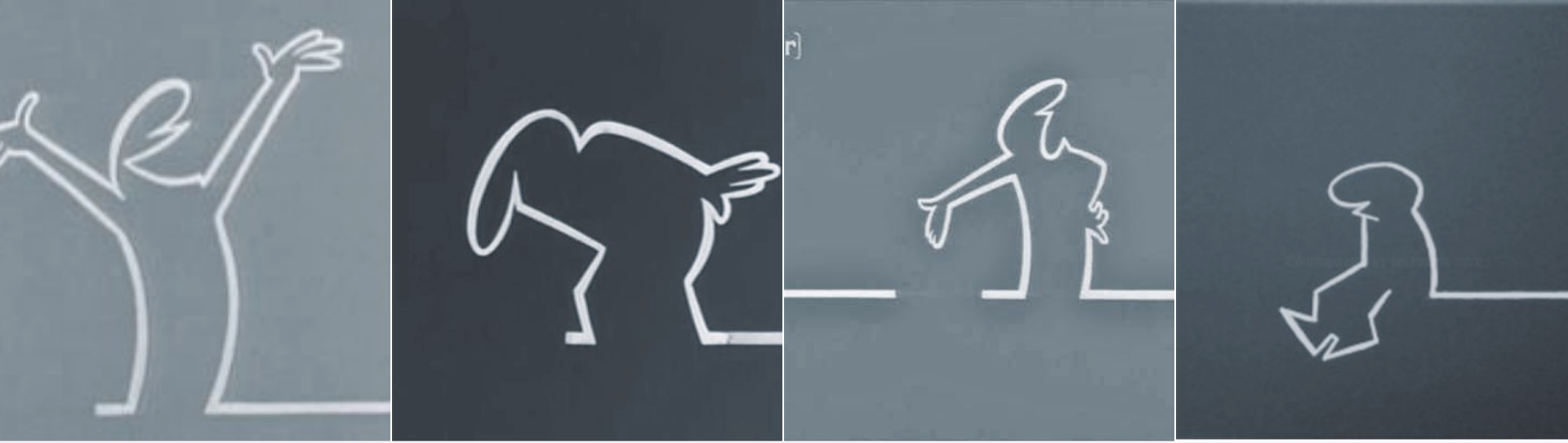


Projet n° 7 : Réalisez une preuve de concept

01

Introduction

Présentation du NLP

02

Etat de l'art

Techniques et dernières avancées

03

Application

Reconnaissance des émotions sur des posts Twitter : article de recherche de mars 2021

01

Introduction

Le NLP, qu'est ce que c'est?

- Sous domaine de l'IA dont le but est de traiter, comprendre et générer le langage humain
- Exemples du quotidien : traducteurs automatiques, les correcteurs d'orthographe et de grammaire, les moteurs de recherche, les filtres spams, les chatbots …
- 1ère expérience de traducteur automatique avec IBM en 1954 avec des règles au cas par cas. Puis utilisations de techniques statistiques et de machine learning.
- Grandes avancées à partir des années 2 000 avec le deep learning.

Les challenges du NLP

- Le langage humain peut être ambigu, c'est à dire que son intreprétation n'est pas toujours unique.

Exemples : "j'ai vu une fille avec un téléscope" ou "j'ai une souris chez moi" ou "Je suis la reine d'Angleterre"

- Le contexte des mots joue un rôle clé !

- Face à un problème discret : La modification d'une lettre dans une phrase peut en modifier son sens. (jolie voiture/ jolie toiture)

02

Etat de l’Art

Avant tout … nettoyage des données textuelles

Certaines opérations que l'on peut réaliser :

- supprimer les espaces multiples, ponctuations,
- supprimer les stopwords
- lemmatization ou stemming
- enlever la case du texte

Ces traitements varient d'un problème ou modèle à un autre

Modéliser les données textuelles: Les BOW

- Calculer le nombre d'occurrences des mots dans chaque document (phrases, posts ect.. du corpus de texte). On obtient une matrice des “term frequency”.

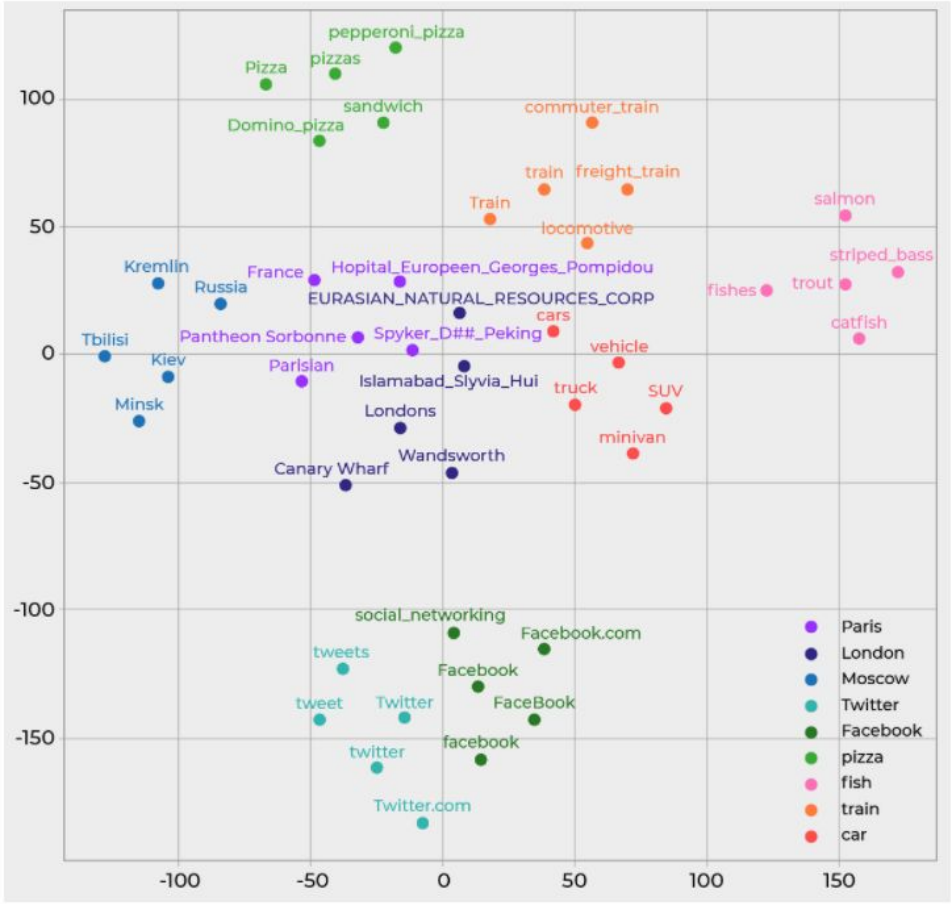
	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird	1	5	0	0	2	0	0
You heard about the bird	1	1	1	0	1	0	1
The bird is the word	0	1	0	1	2	1	0

BOW on Surfin' Bird

- Pour donner + de poids aux mots qui apparaissent dans peu de documents et moins aux mots fréquents, on peut calculer la matrice “TFIDF”

Modéliser les données textuelles: Les words embeddings

En 2013, apparaît une nouvelle méthode de vectorisation qui permet de garder le lien sémantique des mots.



Word similarity according to Word2vec

Modéliser les données textuelles: Les words embeddings

- Permettent de conserver les analogies entre les mots (“Paris” est à la France ce que “Berlin” est à l’Allemange)

$$\overrightarrow{queen} - \overrightarrow{woman} = \overrightarrow{king} - \overrightarrow{man}$$
$$\overrightarrow{France} - \overrightarrow{Paris} = \overrightarrow{Germany} - \overrightarrow{Berlin}$$

- Plusieurs méthodes existent pour créer ces words embeddings: word2vec, glove, fastText, negative sampling.

- Des modèles pré-entraînés sur des données textuelles (twitter, wikipedia, gigaword …) sont disponibles.

Les words embeddings: construction de word2vec

Développé par google en 2013

Idée : entraîner un modèle supervisé sur un corpus de texte qui, avec un petit réseau de neurones, à partir d'un mot pris aléatoirement dans une phrase (target) essaye de prédire un mot qui lui est proche dans la phrase.

Exemple : "Je veux un verre de jus d'orange pour aller avec mes céréales"

target	context word to predict
orange	jus
orange	pour
orange	céréales

Les words embeddings: Attention aux biais !

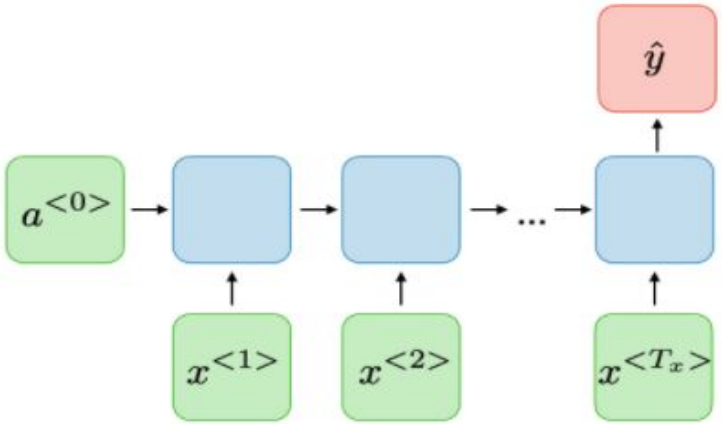
La nature des textes peut amener à certains biais de genre, d'ethnicité, d'orientation sexuelle, d'âge ect dans les word embeddings.

Des méthodes pour réduire ces biais sont exposés dans l'article Bolukbasi et al. 2016, *Man is to computer programmer as woman is to homemaker ?*

Les réseaux de neurones récurrents

Les réseaux de neurones récurrents (RNN pour recurrent neural networks) sont un type de réseau de neurones adaptés aux problèmes séquentiels. Ils permettent de traiter des entrées aux dimensions variables.

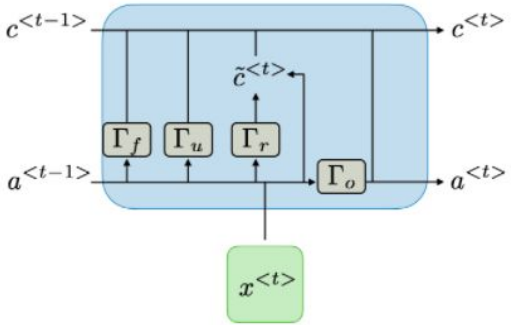
Plusieurs configuration selon le problème sur le nombre de sorties. Par exemple, pour une classification de texte on aura une configuration “many-to-one”:



Les réseaux de neurones récurrents

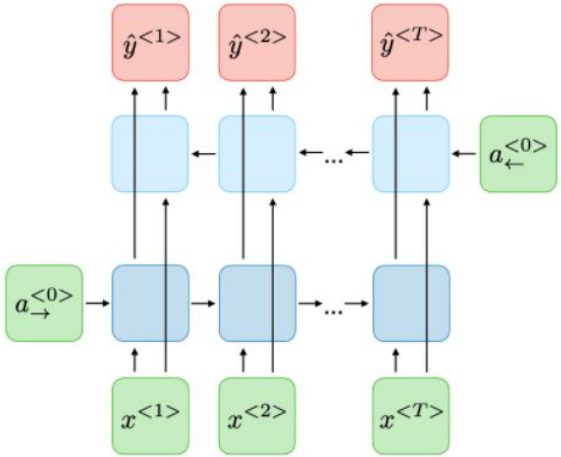
En raison du grand nombre de couches, on peut avoir un problème de “vanishing gradient” ou “d’exploding gradient”.
Cela implique des difficultés à capter les dépendances entre des termes éloignés d’une phrase.

Face au problème du “vanishing gradient”: Les réseaux de neurones LSTM. A chaque étape t, nous avons une cellule mémoire c<t> qui est actualisée qui va permettre de gérer les informations précédentes



Les réseaux de neurones récurrents bidirectionnels

Pour les termes placés en début de phrase, il peut être plus intéressant d'avoir les informations en partant de la fin de la phrase.
Une solution est de créer des couches d'activation "backward" c'est à dire allant dans l'autre sens.



Les Transformers

Les Transformer network sont un type de réseau de neurones qui ont complètement révolutionné le domaine du NLP. Ils se basent sur deux concepts clés :

- le mécanisme du self attention
- le fait de traiter les entrées non plus de manière séquentielle mais en parallèle.

Les Transformer ont été développés dans l'article "Vaswani et al; 2017 Attention is all you need".

Self - Attention

Mécanisme qui permet d'avoir une représentation de chaque mot dans son contexte, ce que ne permettent pas en général les word embeddings.

Par exemple, dans les phrases "Jane plays basketball" ou "Hamlet is a play of Shakespeare", le mot *play* n'a pas le même sens pourtant le word embedding à lui seul ne permet pas de faire une différence de contexte.

Self - Attention

Dans une phrase composée de 5 mots $x^{<1>}$, $x^{<2>}$, $x^{<3>}$, $x^{<4>}$ et $x^{<5>}$, on veut une nouvelle représentation des mots donnée respectivement par $A^{<1>}$, $A^{<2>}$, $A^{<3>}$, $A^{<4>}$ et $A^{<5>}$.

A chaque mot $x^{<l>}$, on, associe des valeurs $q^{<l>}\ (query)$, $k^{<l>}\ (key)$ et $v^{<l>}(value)$ avec :

$$q^{<l>} = W^q * x^{<l>},\ k^{<l>} = W^k * x^{<l>} \text{ et } v^{<l>} = W^v * x^{<l>}$$

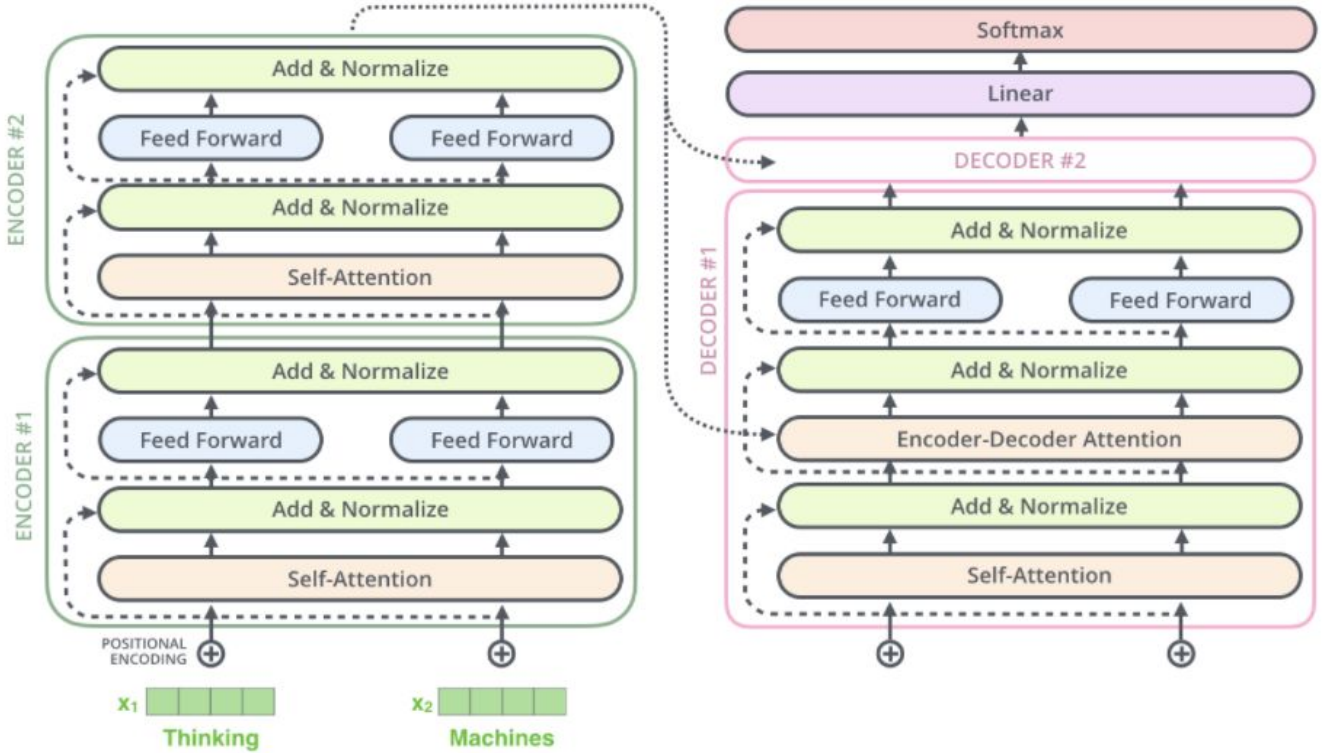
Les matrices W seront apprises par le modèle.

La valeur de la nouvelle représentation correspondant à $x^{<l>}$ est donnée par:

$$A^{<l>} = \sum_{i=1}^{5} \frac{exp(q^{<l>}.k^{<i>})}{\sum_{1 \le j \le 5} exp(q^{<l>}.k^{<j>})} v^{<i>}$$

Intuitivement, on cherche à représenter un lien question/réponse entre le mot l et les autres mots.

Architecture du transformer network



BERT

BERT (Bidirectional Encoder Representations from Transformers) est un modèle transformer pré-entraîné développé par google en 2018.

L'architecture de BERT est constituée de plusieurs couches de transformer encoder afin de réussir à contextualiser les entrées. Pour entraîner ce modèle, on s'est donné un problème de ML supervisé qui constituait à prédire chaque mot caché dans une phrase.

D'autres modèles transformers pré-entrainés existent comme ALBERT, XLnet ou GPT-2.

03 Application

Présentation

Nous nous penchons sur une étude de mars 2021 dans cet article de recherche initulé *Emotion and sentiment analysis of tweets using BERT*

Les auteurs exposent leur méthode avec BERT pour détecter des sentiments (positifs/négatif/neutre) et émotions (colère/peur/joie/tristesse) sur des posts twitter. Dans le cadre de notre étude, on se limitera à la détection des émotions.

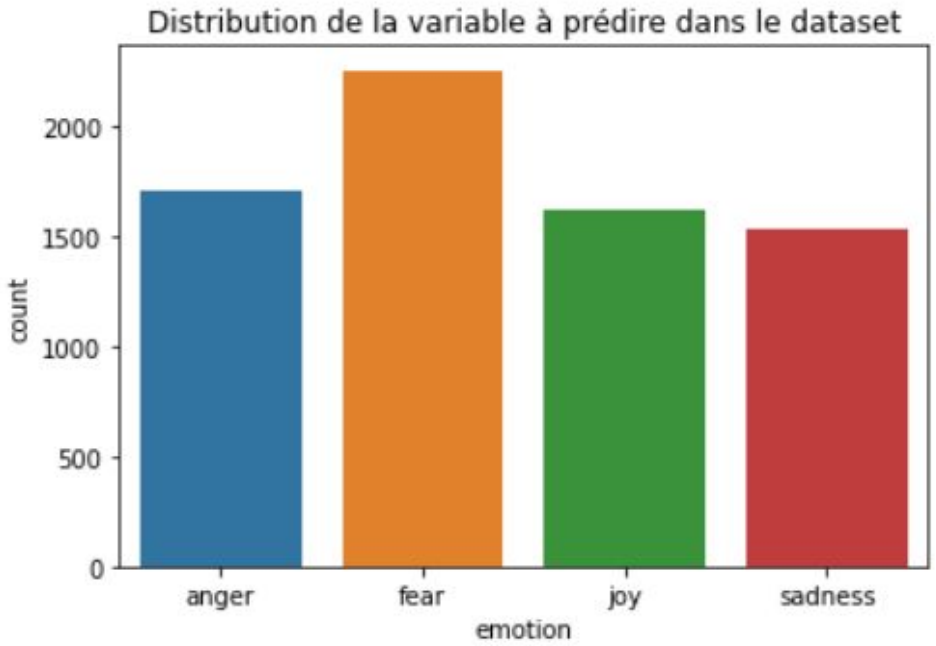
On récupère un dataset de 7102 observations

https://saifmohammad.com/WebPages/TweetEmotionIntensity-dataviz.html

Nettoyage “de base”

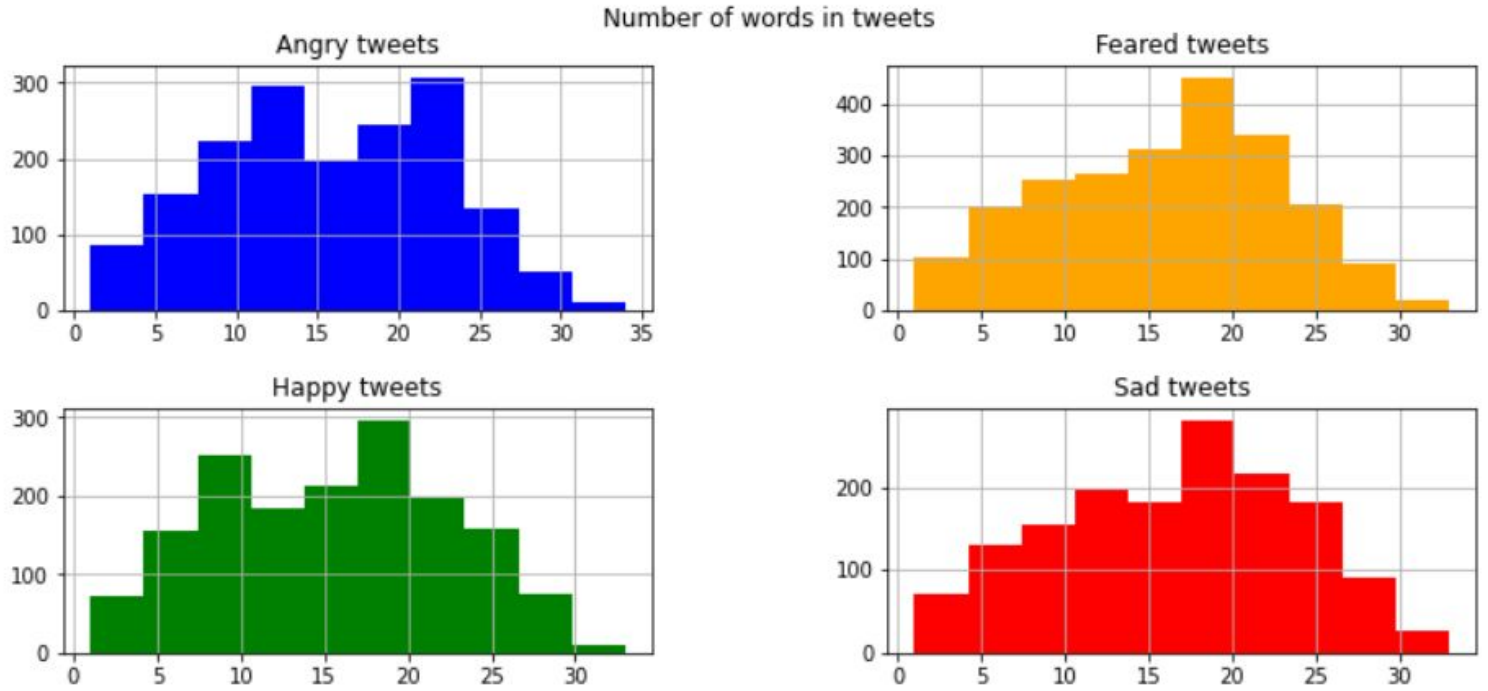
On supprime :
- les tweets qui contiennent uniquement des caractères “non ASCII”
- les tweets très courts (avec un nombre de caractères <=2)
- les mentions
- les URLS
- les retweets
- on lemmatize

Analyses des données



On réalise un undersampling

Analyses des données



Analyses des données

Angry Tweets

Feared Tweets

Happy Tweets

Sad Tweets

Modèle Baseline

TFIDF vectorizer en pre-processing + classifier

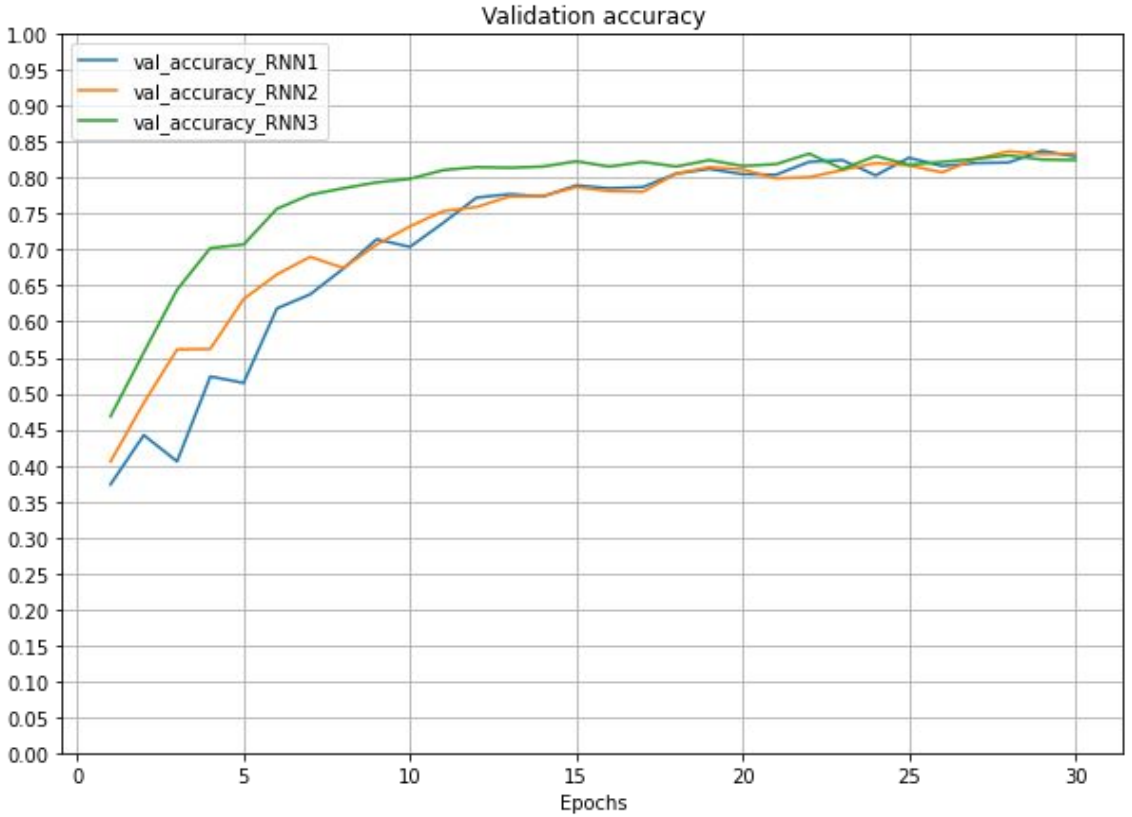
	accuracy	f1-score
KNN	0.5677	0.5688
LinearSVC	0.7153	0.7155
LogisticRegression	0.7096	0.71

RNN

On utilise le word embedding GloVe pré-entraîné

Ensuite on a testé trois RNN :
RNN 1 : LSTM avec un “nettoyage de base” des données (cf III.b)
RNN 2 : LSTM avec en plus du nettoyage de base suppression de la ponctuation et des emojis
RNN 3 : LSTM bidirectionnel avec un “nettoyage de base” des données (cf III.b)
La fonction d’activation de sortie est la fonction softmax et on 4 unités de sortie (une pour chaque émotion).

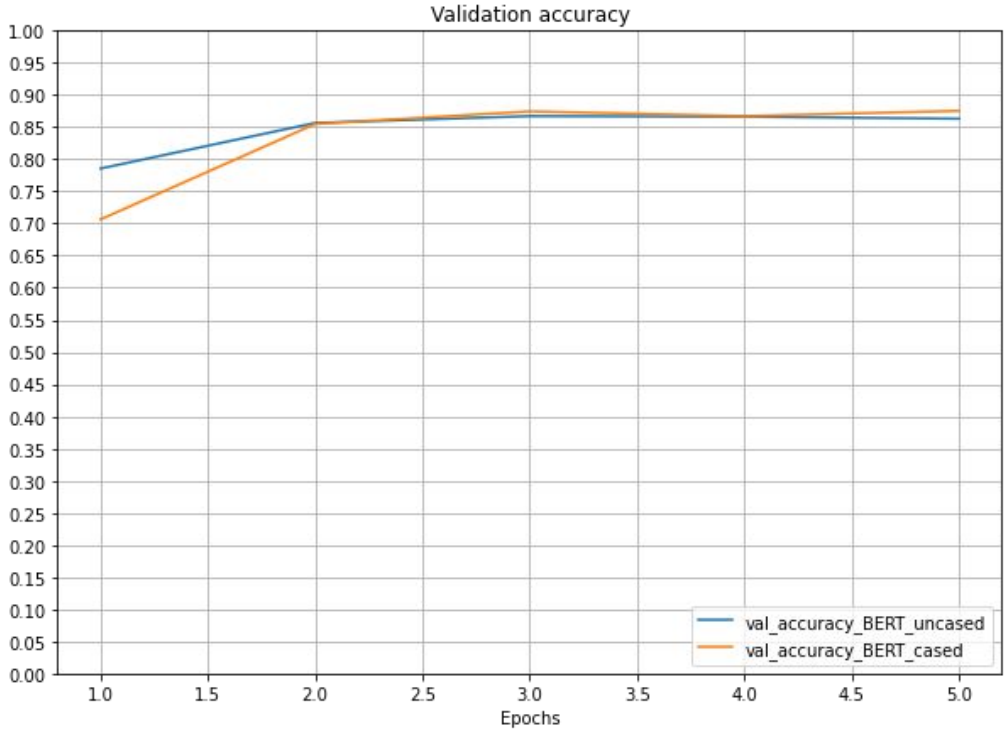
Performances des RNN



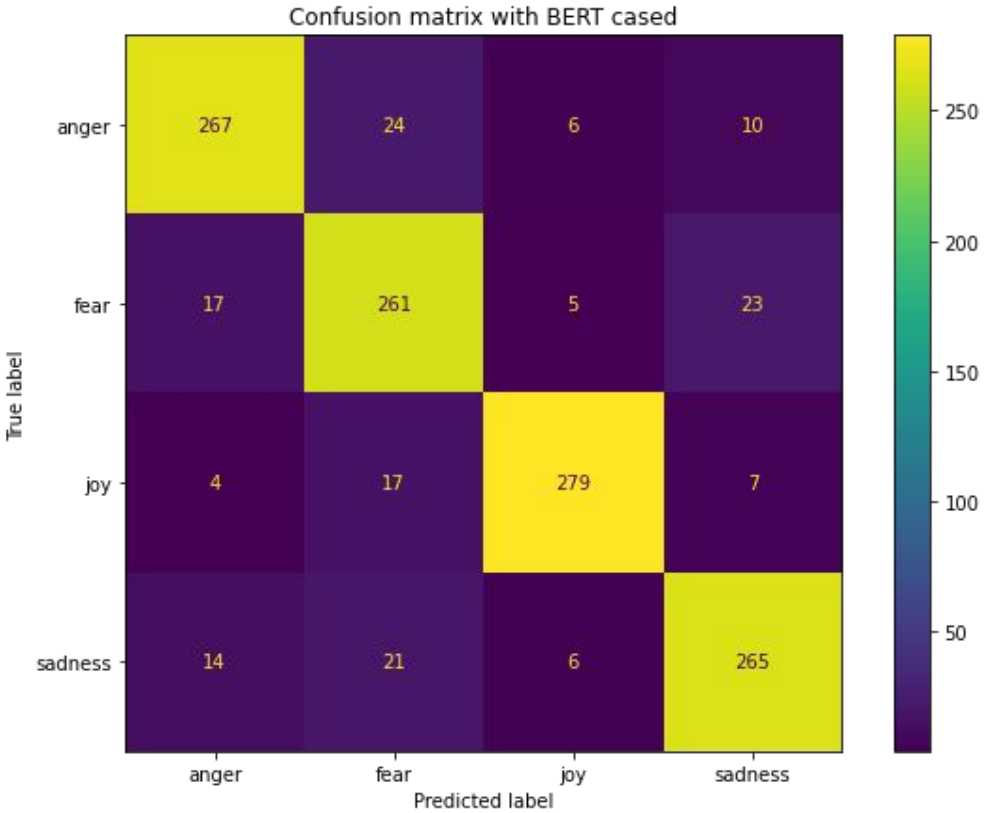
BERT

On applique le nettoyage de base et on essaye les modèles BERT cased et BERT uncased. On exporte le modèle pré-entraîné à l'aide de la librairie Transformers.
La fonction d'activation de sortie est la fonction softmax et on 4 unités de sortie (une pour chaque émotion).

Performances BERT



Performances BERT



Performances BERT : quelques tests

tweet	Prédiction BERT cased
'I feel happy for the gift! Thank you'	joy
'This play was wonderful!amazing! I wanna watch it again !’	joy
I passed my driving test !! 😀😀😀 #positive	fear
I passed my driving test !! 😀😀😀 #smile	joy
'The tornado caused a lot of material and human damage. My thoughts are with the victims'	fear
'The tornado caused a lot of material and human damage. I\'ve got so much pain for the victims.'	fear
'The tornado caused a lot of material and human damage. I feel so sad for the victims.'	sadness

MERCI pour votre écoute

sources :

Emotion and sentiment analysis of tweets using BERT
http://ceur-ws.org/Vol-2841/DARLI-AP_17.pdf (mars 2021)

Guide to using pretrained word embeddings
https://blog.paperspace.com/pre-trained-word-embeddings-natural-language-processing/#final-thoughts

Cheatsheet NLP
https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks#word-representation

Multiclass classification using transformers for beginners
https://www.analyticsvidhya.com/blog/2021/12/multiclass-classification-using-transformers/

Cours : Openclassroom “introduction to NLP”; Andrew NG “sequence models”