Étiquetage morpho-syntaxique avec un classifieur

Adrien Chesneau et Julien Guyot

13 avril 2019

1 Introduction

Ce projet porte sur l'étiquetage morpho-syntaxique, dit Part of Speach, d'un corpus.

1.1 Part of speech

L'étiquetage morpho-syntaxique est l'association à chaque mot d'un texte des informations grammaticales liées à ce même mot.

Dans le cadre de ce projet, les informations grammaticales se limitent aux informations liés à la nature grammaticale des mots. Les natures possibles sont les suivantes :

ADJ Les adjectifs qui caractérisent le nom et s'accordant en genres et en nombres.

ADV Les adverbes qui précisent le sens d'un verbe et marque le degré d'intensité de la phrase.

INTJ Les interjections qui montrent l'exclammation.

NOUN Les noms qui désignent une réalité concrète ou abstraite et s'accordant en genres et en nombres.

PROPN Les noms propres qui désignent le nom d'un individu, un lieu ou un objet spécifique.

VERB Les verbes qui est les noyaux de la phrase, ils expriment un état ou une action et varient en personne.

DET Les déterminants précèdent un nom et qui marque le genre et le nombre de celui-ci.

NUM Les chiffres qui expriment un nombre et une relation avec un nombre.

PRON Les pronoms qui désignent une personne et remplacent un élément déjà nommé.

CCONJ Les conjonctions de coordinations qui permettent de relier deux mots, propositions et phrases.

AUX Les auxiliaires qui sont des verbes qui se combinent à un verbe principale afin de constituer un temps composé, expriment le temps, le mode et la voix.

ADP Les prépositions qui permettent de relier un nom, verbe ou adjectif à un nom, pronom, adverbe ou à un verbe.

SCONJ Les conjonctions de subordinations permettant d'introduire une proposition subordonnée.

PART Les particules qui permettent de donner un sens une fois associés à un mot.

PUNCT Les ponctuations qui permettent l'organisation de l'écrit en indiquant les pauses et intonations ou en précisant le sens de la phrase.

SYM Les symboles qui sont des mots spéciaux différants des mots ordinaires par leurs formes, fonctions ou les deux.

X Les autres qui sont des mots ne pouvant être assignés aux autres classes grammaticales.

1.2 Objectif

Notre objectif est d'implémenter et d'entraîner un neurone artificiel afin de prédire la nature de chaque mot, connaissant son contexte (égal à la phrase dans lequel il se trouve).

Pour cela, nous disposons de plusieurs jeux de données provenant pour la plupart de la base *Universal Dependencies* . Nous avons écarté les *dataset* en anglais pour nous concentrer sur les données écrites en Français. Ces données seront décrites plus en détail dans la deuxième partie.

1.3 Plan

Dans un premier temps, nous allons décrire les données, et montrer quelques informations sur ces dernières.
Ensuite, nous présenterons la structure et le contenu du code, ainsi que les perceptrons et la modélisation choisie
Nous montrerons et expliquerons les résultats obtenus, avant de conclure.

2 Présentation des données

Comme dit précédemment, nous avons à dispositions plusieurs sets de données en anglais et en français. Nous utiliserons uniquement les sets de données en français.

Ces données sont sous forme de fichiers `.json`, chaque fichier correspond à un *dataset* , qui est un ensemble de couples (phrase, labels), ou chaque label $\in$ labels correspond à la nature grammatical du mot correspondant.

Pour chaque set de données, nous indiquons les informations suivantes :

Présentation brève du dataset

Pourcentage des mots hors vocabulaire Pourcentage de mots apparaissant uniquement dans les données de test, par rapport à l'ensemble du vocabulaire

KL-Divergence La divergence de Kullback-Leibler entre les données d'entraînement et de test

Distribution des différents labels dans le dataset

Mots les plus utilisés dans le dataset.

2.1 pud

Présentation : Phrases qui proviennent de sites informatifs, id est de journaux ou de Wikipedia
Pourcentage de mots hors vocabulaire : 54.04
KL-Divergence : 0.256

Données de test
Nombre de phrases : 1000

Mot	Apparitions
,	1178
de	1175
.	985
la	675
et	442
le	414
les	399
à	370
l'	328
des	327

FIGURE 1 – Mots les plus utilisés dans le set pud(test)

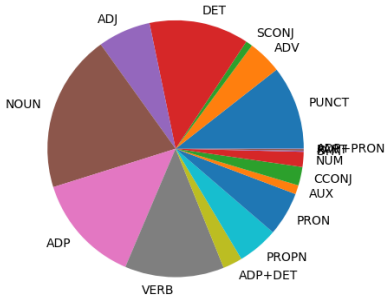

FIGURE 2 – distribution des labels

Données de train
Nombre de phrases : 803

Mot	Apparitions
de	1114
,	1016
.	750
la	683
et	544
l'	486
des	432
à	404
d'	397
le	385

FIGURE 3 – Mots les plus utilisés dans le set pud(train)

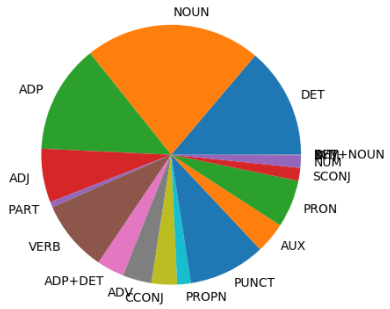

FIGURE 4 – distribution des labels

2.2 ftb

Présentation : Phrases provenant du journal «Le Monde»
Pourcentage de mots hors vocabulaire : 7.099
KL-Divergence : 0.110

Données de test
Nombre de phrases : 2541

Mot	Apparitions
,	4528
de	3593
.	2258
__DIGIT__	2200
la	2016
l'	1350
à	1333
le	1293
des	1255
les	1222

FIGURE 5 – Mots les plus utilisés dans le set ftb(test)

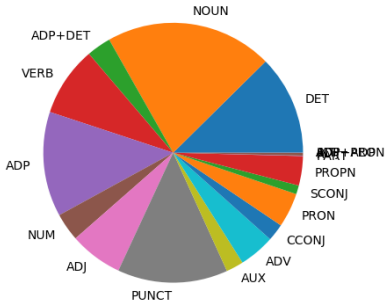

FIGURE 6 – distribution des labels

Données de train
Nombre de phrases : 14759

Mot	Apparitions
,	27318
de	22518
.	13735
__DIGIT__	11104
la	10780
l'	8129
à	7984
le	7255
les	7086
des	7071

FIGURE 7 – Mots les plus utilisés dans le set ftb(train)

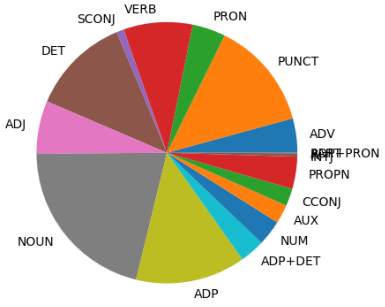

FIGURE 8 – distribution des labels

2.3 spoken

Présentation : Données sur le français oral
Pourcentage de mots hors vocabulaire : 32.14
KL-Divergence : 0.366

Données de test
Nombre de phrases : 726

Mot	Apparitions
de	380
est	262
euh	222
la	219
et	193
que	188
le	181
l'	177
à	169
je	162

FIGURE 9 – Mots les plus utilisés dans le set spoken(test)

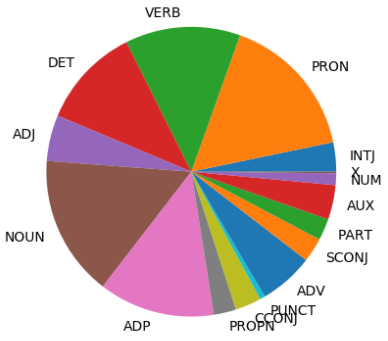

FIGURE 10 – distribution des labels

Données de train
Nombre de phrases : 1153

Mot	Apparitions
de	529
euh	489
est	362
la	339
et	328
le	294
à	282
c'	264
je	223
qui	218

FIGURE 11 – Mots les plus utilisés dans le set spoken(train)

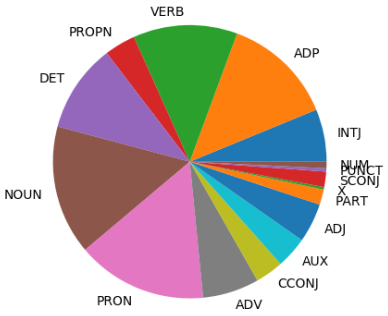

FIGURE 12 – distribution des labels

2.4 sequoia

Présentation : Ensemble de texte, à la base disponible avec des annotations «profondes», ie plus complètes. Néanmoins, ces annotations ont été normalisées par rapport aux autres données
Pourcentage de mots hors vocabulaire : 9.417
KL-Divergence : 0.113

Données de test
Nombre de phrases : 456

Mot	Apparitions
de	463
,	431
.	344
la	216
__DIGIT__	204
l'	186
à	169
et	165
des	152
d'	139

FIGURE 13 – Mots les plus utilisés dans le set sequoia(test)

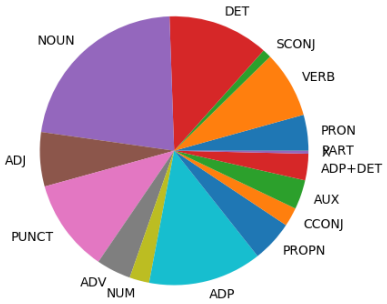

FIGURE 14 – distribution des labels

Données de train
Nombre de phrases : 2231

Mot	Apparitions
de	2435
,	2354
.	1683
la	1176
__DIGIT__	1039
l'	916
à	857
et	845
des	759
le	743

FIGURE 15 – Mots les plus utilisés dans le set sequoia(train)

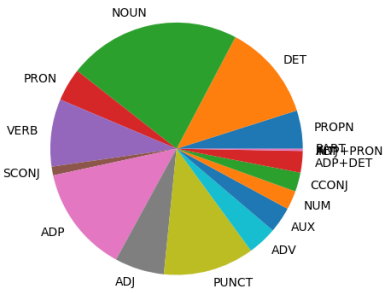

FIGURE 16 – distribution des labels

2.5 partut

Présentation : Ensemble varié de textes, incluant de extraits de conférences comme des extraits de Wikipedia ou des textes légaux
Pourcentage de mots hors vocabulaire : 5.743
KL-Divergence : 0.330

Données de test
Nombre de phrases : 110

Mot	Apparitions
de	120
.	110
,	79
la	69
à	58
l'	56
les	46
et	42
des	39
le	39

FIGURE 17 – Mots les plus utilisés dans le set partut(test)

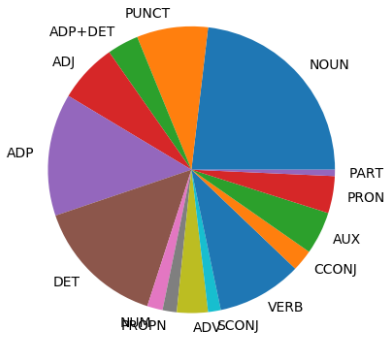

FIGURE 18 – distribution des labels

Données de train
Nombre de phrases : 803

Mot	Apparitions
de	1114
,	1016
.	750
la	683
et	544
l'	486
des	432
à	404
d'	397
le	385

FIGURE 19 – Mots les plus utilisés dans le set partut(train)

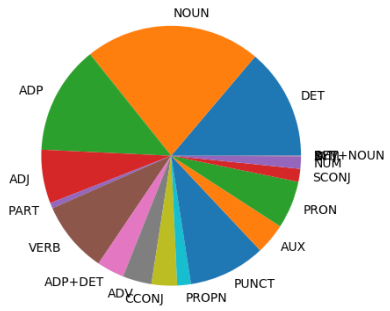

FIGURE 20 – distribution des labels

2.6 gsd

Présentation : Phrases variées
Pourcentage de mots hors vocabulaire : 1.359
KL-Divergence : 0.409

Données de test
Nombre de phrases : 416

Mot	Apparitions
,	489
de	426
.	353
la	222
et	182
l'	171
à	168
__DIGIT__	166
le	155
les	127

FIGURE 21 – Mots les plus utilisés dans le set gsd(test)

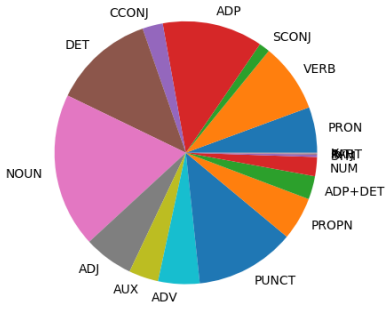

FIGURE 22 – distribution des labels

Données de train
Nombre de phrases : 14450

Mot	Apparitions
de	16964
,	16595
.	13359
la	8676
et	7148
__DIGIT__	7095
le	6389
à	6216
l'	5864
en	4793

FIGURE 23 – Mots les plus utilisés dans le set gsd(train)

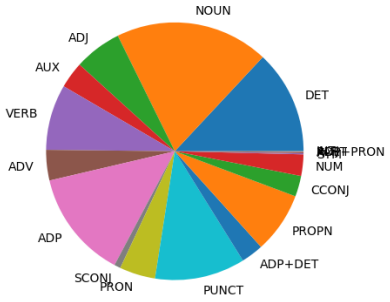

FIGURE 24 – distribution des labels

2.7 foot

Présentation : Ensemble de tweets parlant de football

Données de test
Nombre de phrases : 743

Mot	Apparitions
__MENTION__	549
la	433
__SHARP__	395
Juventus	368
:	338
de	325
!	314
le	257
,	247
__URL__	226

FIGURE 25 – Mots les plus utilisés dans le set foot(test)

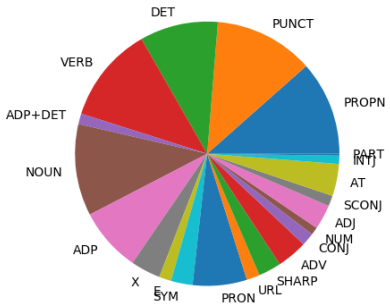

FIGURE 26 – distribution des labels

2.8 natdis

Présentation : Ensemble de tweets

Données de test
Nombre de phrases : 622

Mot	Apparitions
de	994
__URL__	469
__DIGIT__	411
terre	338
__SHARP__	300
tremblement	295
:	285
chaleur	244
au	220
vague	206

FIGURE 27 – Mots les plus utilisés dans le set natdis(test)

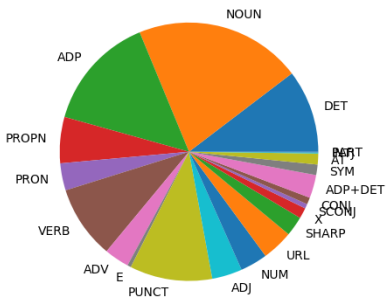

FIGURE 28 – distribution des labels

3 Présentation du code

Le code est regroupé dans les fichiers suivants :

helpers.py Regroupe des fonctions utiles pour le chargement des données ou le test des résultats.

tex_data_describe.py et tex_result_describe.py Regroupe des fonctions utiles pour (respectivement) décrire les données ou les résultats vers un ou plusieurs fichiers LaTeX

modele.py fournit une interface commune aux différents classifieurs.

multiclass_perceptron.py Le perceptron multiclasse du TP5.

modele_tp5.py Le perceptron utilisé avec les features du TP 5.

modele_pj_distrib.py Inclut le modèle de perceptron implémenteant les features proposées dans le sujet du projet, incluant les *distributional features*.

modele_pj_nodistrib.py Même chose que modele_pj_distrib, sans les *distributional features*

modeles_sklearn.py Utilise deux implémentations de classifieurs provenant de la bibliothèque SKLearn.

3.1 Description des classifieurs

Les classifieurs possèdent tous la même interface, qui est indiquée dans `modele.py`
L'algorithme de classification utilisé pour tous les modèles (hors ceux provenant du module `modeles_sklearn`) est le perceptron implémenté dans `multiclass_perceptron` Ainsi, seules les features sont différentes. Chaque feature est décrite en fonction de la phrase *sentence* , et du mot w dont il faut prédire le label, qui est situé à la position *pos* dans la phrase.

`Modele_tp5`

Ce modèle implémente les features proposées lors du tp 5, qui sont les suivantes :

- Le mot w
- Les trois derniers caractères de w. Le suffixe d'un mot est important pour déterminer le label, car certains de ces suffixes peuvent être porteurs de sens (par exemple, "ant" indiquerait une forte probabilité que le mot soit un participe présent)
- Le premier caractère de w
- Le dernier caractère de w
- un booléen indiquant si le premier caractère est une lettre majuscule (peut être utile pour déterminer les noms propres)
- un booléen indiquant si tout le mot est écrit en majuscule.
- Le mot précédent dans la phrase. Utile notamment pour différencier des mots ambigus. Par exemple, si l'on recontre le mot "basse" apres un déterminant, alors ce dernier sera très probablement un nom, tandis qu'il pourrait être un adjectif dans un autre contexte
- Le mot situé à $pos - 2$ dans la phrase. Même principe que ci-dessus
- Le mot suivant dans la phrase.
- Le mot situé à $pos + 2$ dans la phrase.

Ces features sont implémentées dans la fonction `build_sparse2`, et le fichier contenant le Modele est `modele_tp5.py`.

`Modele_projet` et `Modele_nodistrib`

Implémentent les features proposées dans le sujet du projet :

- w
- Les deux mots précédant est suivant w
- Le nombre de mots avant w dans la phrase
- Le nombre de mots suivant w dans la phrase
- Un booléen indiquant si le mot se finit par un s
- Un ensemble de booléens, indiquants respectivement si :
  - le mot contient un chiffre
  - le mot contient un tiret
  - le mot est écrit en majuscules
  - le mot se finit en `-é`. (si oui, il y a des chances pour que ce soit un participe passé)
  - le mot se finit en `-er`
  - le mot se finit en `-ant`
- les *distributional features* . Elles sont est composées de deux features, qu'on appellera respectivement "distribution à droite" et "distribution à gauche".
  Pour chaque mot w, on regarde le mot juste à droite c (resp.juste à gauche), et on calcule le nombre de fois où le nombre de couples wc (resp cw) apparaît dans le corpus de test.
  Les deux features sont égales à ce nombre. On se limite à calculer le nombre d'apparitions de ces couples pour c faisant partie des 10 mots les plus utilisés.
  Cela semble utile pour pouvoir inférer, par exemple des expressions, ou être plus utile dans le cas de

La classe `Modele_nodistrib` n'implémente pas les *distributional features*.

Les deux classes se trouvent respectivement dans `modele_pj_distrib` et `modele_ph_nodistrib`

`Modele_sklearn_Perceptron` et `Modele_SKLearn_SVM`

Les features sont construites à partir des informations suivantes :

- w
- un booléen indiquant si w est écrit en lettres majuscules
- un booléen indiquant si w commence par une lettre majuscule
- les trois dernières lettres du mot
- les deux dernières lettres du mot
- les trois premières lettres du mot
- Le mot précédent
- Le mot suivant
- un booléen indiquant si le mot est un nombre
- un booléen indiquant si le mot contient des lettres majuscules

Le dictionnaire représentant les features est donné à une instance de la classe `DictVectorizer`, qui "transformera" ces features en une forme acceptable pour le classifieur. Ensuite, la classification sera

réalisée par une instance de `Perceptron`}ou de `LinearSVC`.

4 Résultats

Dans un premier temps, voici les résultats des modèles sur le dataset `ftb`

.

Classifieur	Modele tp5	Perceptron nodistrib	Perceptron distrib	SKLearn Perceptron	SKLearn SVM
Pourcentage de réussite	94.70	93.97	93.47	94.13	96.71

FIGURE 29 – Taux de réussite sur les données pour divers classifieurs

Nous remarquons que :

— Les résultats sont bons sur cet ensemble de données (avec plus de 90 % à chaque fois. Cela reste à relativiser, étant donné que ces données d'entraînement sont de loin les plus conséquentes, avec plus de 14000 phrases.

— La construction du dictionnaire nécessaire aux *distributional features* prend beaucoup de temps.

— Les *distributional features* que l'on a construit ne servent à rien ; au contraire elles baissent le score de réussite sur le *dataset* `ftb`. Cela est probablement dû à une mauvais compréhension de l'article, ayant mené à une mauvaise implémentation.

Maintenant, nous allons nous intéresser à des tableaux de résultats plus détaillés. Pour chacun de ces tableaux, chaque colonne représente un ensemble d'entraînement, tandis que chaque ligne représente un ensemble de test.

Pour chaque ensemble d'entraînement de de test, nous avons trois mesures, toutes exprimées en pourcentage :

OOV La réussite sur les mots hors vocabulaire, ie les mots apparaissant dans l'ensemble de test mais pas dans l'ensemble d'entrainement.

AMB La réussite sur les mots ambigus, qui apparaissent dans les données de train ou de test, et qui peuvent avoir deux natures différentes

GEN Le taux de réussite général.

	ftb			gsd			partut			pud			sequoia			spoken		
	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN
foot	25.46	82.28	65.95	22.34	77.01	67.94	42.11	73.30	61.04	42.11	73.30	61.04	33.37	78.45	64.89	31.63	69.44	50.65
ftb	76.31	93.01	94.70	70.97	90.93	91.01	69.94	86.36	85.29	69.94	86.36	85.29	76.85	91.23	90.36	44.26	71.82	62.65
gsd	67.61	89.03	89.99	78.81	93.51	94.62	69.04	88.87	85.36	69.04	88.87	85.36	76.96	91.59	90.02	45.01	75.84	64.46
natdis	24.92	87.98	78.52	46.93	86.72	79.10	51.74	78.38	71.35	51.74	78.38	71.35	51.72	84.71	76.22	31.10	69.26	55.00
partut	74.03	87.81	90.81	71.73	92.37	91.84	69.28	90.81	91.13	69.28	90.81	91.13	79.17	92.70	90.13	50.87	75.71	69.22
pud	61.45	81.50	84.84	67.53	84.02	87.96	67.81	79.07	80.64	67.81	79.07	80.64	72.38	80.88	83.76	47.75	75.58	64.63
sequoia	70.82	90.18	91.06	73.17	93.20	92.73	70.61	90.15	85.69	70.61	90.15	85.69	81.31	94.37	94.64	46.44	75.60	65.06
spoken	49.55	71.92	78.64	51.09	77.00	82.50	60.32	75.70	76.85	60.32	75.70	76.85	63.71	73.80	78.02	78.57	86.65	90.34

FIGURE 30 – Taux de réussite détaillés pour le perceptron Modele tp5

De manière générale, le score de 90% sur la réussite des données était flatteur, la plupart des résultats tournant autour de 80%

Nous remarquons que les scores les plus bas sur les tests proviennent des ensembles de test natdis et foot (resp. autour de 80 et 70 % sur la plupart des corpus).

Deux explications pourraient exister : La première étant que, de manière générale, les tweets sont rarement écrits dans un français correct, la seconde serait que ces derniers contiennent un grand nombre de labels spécifiques (par exemple les mentions) à twitter.

Par conséquent, des scores particulièrement bas sur ces *dataset* se retrouveront avec les autres classifieurs.

	ftb			gsd			partut			pud			sequoia			spoken		
	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN
foot	12.98	79.61	59.18	18.69	76.62	63.53	18.90	76.33	46.63	18.90	76.33	46.63	21.33	78.00	57.85	20.39	72.79	43.13
ftb	61.13	92.70	93.47	62.01	90.52	89.05	51.37	84.32	75.70	51.37	84.32	75.70	58.72	90.26	83.50	33.64	73.98	55.87
gsd	47.98	89.35	87.30	65.97	92.43	91.93	46.12	85.90	75.09	46.12	85.90	75.09	56.81	89.96	82.50	33.08	76.76	57.20
natdis	16.68	86.56	75.15	36.73	85.83	75.97	28.52	78.70	57.04	28.52	78.70	57.04	34.05	86.82	66.93	23.25	74.80	50.05
partut	56.73	89.89	89.78	58.69	89.98	88.07	49.82	86.08	81.55	49.82	86.08	81.55	57.72	90.70	80.79	38.49	75.71	61.15
pud	41.10	80.53	81.12	55.42	83.10	85.25	45.70	79.59	70.91	45.70	79.59	70.91	51.88	80.47	76.15	34.36	78.40	57.09
sequoia	54.20	91.51	88.81	59.69	92.32	89.77	45.03	86.12	74.57	45.03	86.12	74.57	60.51	92.48	88.03	34.75	76.94	57.51
spoken	36.09	71.55	76.28	33.71	76.72	79.42	35.75	73.76	65.82	35.75	73.76	65.82	44.16	72.65	70.95	57.03	85.13	83.21

FIGURE 31 – Taux de réussite détaillés pour le perceptron Perceptron distrib

	ftb			gsd			partut			pud			sequoia			spoken		
	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN
foot	15.14	81.72	60.80	18.90	78.05	65.22	20.48	66.79	47.04	20.48	66.79	47.04	17.24	70.18	52.36	16.49	71.76	41.19
ftb	61.40	93.28	93.97	62.36	90.45	89.44	56.19	83.41	79.83	56.19	83.41	79.83	59.87	89.68	84.71	37.53	73.12	58.34
gsd	52.01	89.44	88.20	67.18	92.20	92.34	52.81	84.55	79.00	52.81	84.55	79.00	54.46	90.32	82.81	34.41	77.48	58.83
natdis	15.48	85.76	75.01	33.95	86.35	75.96	35.83	77.01	61.19	35.83	77.01	61.19	42.64	83.86	68.51	23.64	74.96	51.35
partut	50.96	89.73	89.86	55.43	90.60	88.50	58.02	87.56	87.51	58.02	87.56	87.51	60.88	90.58	84.73	46.42	75.90	65.16
pud	44.77	80.88	82.04	55.36	83.80	85.91	50.21	73.78	73.49	50.21	73.78	73.49	48.99	79.72	76.50	36.96	77.25	59.09
sequoia	53.51	91.04	88.61	59.81	91.63	89.90	50.71	87.13	78.54	50.71	87.13	78.54	61.84	92.70	89.49	38.14	75.87	59.79
spoken	35.87	70.77	76.30	33.33	75.47	78.53	36.93	69.03	65.81	36.93	69.03	65.81	47.67	71.04	69.67	61.74	85.79	84.63

FIGURE 32 – Taux de réussite détaillés pour le perceptron Perceptron nodistrib

Nous remarquons que la plus grosse différence entre les deux consiste en le score de réussite dans la classification des mots hors vocabulaire, mais aussi et surtout que le score général des mots hors-vocabulaire est très bas par rapport aux autres classifieurs, dont celui du TP5

Etant donné que toutes les features du TP5 sont présentes dans ce perceptron, l'explication la plus logique est que les features ajoutées ne sont pas utiles, et que par conséquent en ajoutant ces features on complexifie les calculs, ce qui cause cette baisse de la précision , spécifiquement sur les calculs des mots hors vocabulaire.

Cela est montré par le fait que le perception possédant les *distributional features* est légèrement moins précis, notamment sur les mots hors vocabulaire.

	ftb			gsd			partut			pud			sequoia			spoken		
	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN
foot	33.39	78.41	67.50	22.68	76.55	68.47	35.72	74.06	56.63	35.72	74.06	56.63	30.69	80.58	65.61	43.07	71.03	57.42
ftb	74.17	91.38	94.13	76.99	88.72	90.89	66.69	87.56	85.34	66.69	87.56	85.34	73.49	89.68	90.02	51.32	73.40	66.45
gsd	67.06	86.07	88.95	81.07	91.52	94.23	68.01	89.22	85.64	68.01	89.22	85.64	72.79	90.68	89.69	52.39	76.40	67.92
natdis	23.71	84.23	77.32	29.57	84.79	75.75	46.81	82.26	68.55	46.81	82.26	68.55	47.40	83.04	74.63	39.65	76.16	60.66
partut	72.11	86.06	90.09	73.91	90.60	90.69	64.84	91.75	91.96	64.84	91.75	91.96	74.44	90.70	89.38	51.45	74.38	69.34
pud	57.61	79.01	83.57	69.22	82.05	87.52	63.33	78.32	79.83	63.33	78.32	79.83	67.68	79.92	83.21	50.98	76.92	66.44
sequoia	71.21	88.30	90.63	75.84	90.99	92.49	69.14	90.62	86.17	69.14	90.62	86.17	77.53	93.17	94.68	51.87	76.88	67.82
spoken	51.10	68.74	77.67	48.26	75.71	82.27	60.32	74.82	77.24	60.32	74.82	77.24	59.58	72.82	78.74	73.97	83.31	89.31

FIGURE 33 – Taux de réussite détaillés pour le perceptron SKLearn Perceptron

Ici, les résultats sont légèrement meilleurs que le perceptron du TP5. Cela est plausiblement dû soit à une meilleure implémentation de l'algorithme (ou une meilleure extraction des features avec `DictVectorizer`). ou à un nombre d'epoch plus élevé.

L'amélioration la plus sensible par rapport au TP5 est le meilleur score sur les mots hors-vocabulaire.

	ftb			gsd			partut			pud			sequoia			spoken		
	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN	OOV	AMB	GEN
foot	36.30	83.83	70.94	23.19	79.04	70.02	38.49	75.80	61.56	38.49	75.80	61.56	33.59	82.43	67.77	30.08	74.43	51.53
ftb	83.78	95.06	96.71	80.84	91.87	92.91	76.79	89.63	89.38	76.79	89.63	89.38	79.38	92.08	92.52	44.66	76.24	64.92
gsd	73.93	90.46	92.16	85.06	94.59	96.16	77.35	90.92	89.55	77.35	90.92	89.55	78.46	92.75	92.25	47.49	79.27	67.43
natdis	25.36	88.78	80.16	43.58	87.86	79.65	51.51	83.22	73.80	51.51	83.22	73.80	50.88	85.10	77.10	30.18	81.49	57.16
partut	85.57	91.15	93.71	77.17	93.26	92.92	75.76	94.72	95.06	75.76	94.72	95.06	82.64	92.47	91.92	51.83	76.85	71.05
pud	63.01	82.10	86.08	69.06	84.31	88.90	71.48	79.61	83.47	71.48	79.61	83.47	72.09	81.61	85.46	48.04	79.47	66.65
sequoia	78.73	92.61	93.90	78.04	94.75	94.66	75.93	92.71	89.46	75.93	92.71	89.46	82.75	95.22	96.39	48.85	78.55	67.62
spoken	54.74	74.08	81.35	51.22	79.55	84.67	63.80	78.14	80.50	63.80	78.14	80.50	66.89	78.20	83.32	81.54	89.09	92.88

FIGURE 34 – Taux de réussite détaillés pour le perceptron SKLearn SVM

Nous notons des performances légèrement meilleures partout pour le SVM que pour le Perceptron (l'implémentation de SKLearn), malgré le même set de features.

Cela est logique, puisque le premier est un algorithme plus évolué que le second, qui par conséquent permet une meilleure précision.

5 conclusion

Pour conclure, nous pouvons dire que nos perceptrons, même si possèdant de bons résultats, sont décevants, en effet même si coder dans le cas spécifique du projet, il reste moins bon que les perceptrons génériques proposés par SKLearn comme dit précédemment surement dû à une meilleur implémentation ou à un nombre d'époch bien plus élévé que les notres. Sinon dans l'ensemble, le projet s'est bien passé hormis quelques problèmes sur la formule de KL-Divergence mais il nous a permis de bien comprendre le fonctionnement et le but d'un perceptron ainsi que ces limites, ici montrées grâce vocabulaire "spécifique" de Twitter.