Développement d'un classifieur statistique pour l'étiquetage morpho-syntaxique

AGOSTINHO DA SILVA Ninoh
BEJI Ahmed

1. Introduction

Le but de ce projet était de développer un classifieur qui, après une phase d'entraînement, arriverait à donner des étiquettes correctes sur le plus de mots possibles d'un corpus du français. Nous avons choisi le perceptron comme modèle pour répondre à ce problème.
Ce projet a été réalisé en étroite collaboration avec le monôme WATIEZ Arno, avec qui nous avons partagé les pistes de réflexion et les caractéristiques principales pour améliorer la précision.

2. Données "*in-domain*"

Le corpus utilisé est French GSD, de la base de données ***Universal Dependencies***.
Il y a 3 fichiers textes différents dont un pour l'entraînement de notre modèle et un autre pour l'évaluation.

2.1. Extraction des données

Bien que chaque ligne du corpus soit divisée en de nombreuses colonnes regroupant des informations différentes, nous avons choisi de restreindre notre sélection au mot et à son étiquette de référence ***GOLD_LABEL.***

3. Extraction des caractéristiques

Nous avons étudié les pistes des suffixes des mots (notamment -er et -ir en espérant isoler les verbes), ainsi que la présence du mot en début de phrase, ou bien la présence d'une majuscule au début du mot. Mais les caractéristiques qui ont le mieux fonctionné et qui nous ont permis de passer d'un taux d'erreur de plus de 60% à 26% est la prise en compte du contexte. Initialement, nous avions hésité à l'intégrer de cette façon par peur d'une trop grande complexité en espace mais les résultats ont été sans appel. En tenant en compte ce que nous avons mentionné à la partie précédente, nous ajoutons donc l'information sur le mot précédent et suivant dans les caractéristiques. Ainsi, pour

le mot “chien” dans le contexte “le chien dort”, nous extrayons une entrée de la forme suivante :

{"mot": "chien", "prev_word": "le", "next_word": "dort", "gold": "NOUN"}

4. Évaluation des performances

```
ninoh@pop-os:~/Cours/M1/AATAL/Projet/Perceptron$ python3 Perceptron.py
Taux erreur: 28.7046% (1 epochs) - Temps d'execution: 13.680006504058838
Taux erreur: 27.5102% (2 epochs) - Temps d'execution: 27.931928396224976
Taux erreur: 27.0344% (3 epochs) - Temps d'execution: 45.293028116226196
Taux erreur: 26.8013% (4 epochs) - Temps d'execution: 56.58945941925049
Taux erreur: 26.7722% (5 epochs) - Temps d'execution: 72.04812002182007
Taux erreur: 26.8596% (6 epochs) - Temps d'execution: 83.8762435913086
Taux erreur: 26.5974% (7 epochs) - Temps d'execution: 103.99395942687988
Taux erreur: 26.7916% (8 epochs) - Temps d'execution: 116.63467478752136
Taux erreur: 26.5488% (9 epochs) - Temps d'execution: 136.09972405433655
Taux erreur: 26.7042% (10 epochs) - Temps d'execution: 146.89687037467957
Taux erreur: 26.6460% (11 epochs) - Temps d'execution: 162.0421724319458
Taux erreur: 26.3935% (12 epochs) - Temps d'execution: 175.12671780586243
Taux erreur: 26.6168% (13 epochs) - Temps d'execution: 186.64290118217468
Taux erreur: 26.5974% (14 epochs) - Temps d'execution: 218.23804354667664
Taux erreur: 26.6168% (15 epochs) - Temps d'execution: 213.1383798122406
Taux erreur: 26.3838% (16 epochs) - Temps d'execution: 246.7874813079834
Taux erreur: 26.5974% (17 epochs) - Temps d'execution: 254.27593803405762
Taux erreur: 26.6557% (18 epochs) - Temps d'execution: 258.78149938583374
```

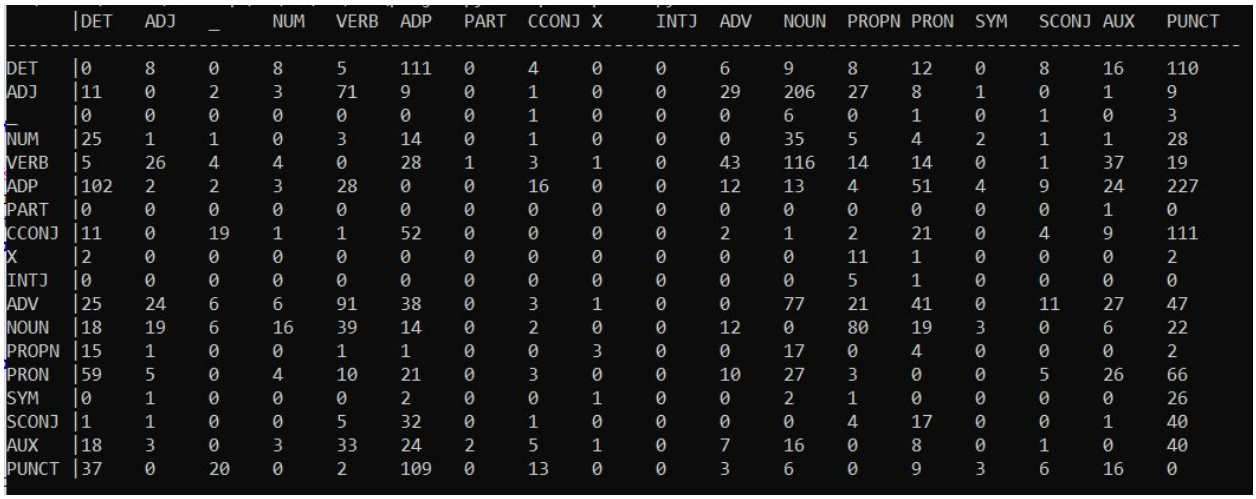
Après avoir effectué de nombreux essais sur cette boucle, nous remarquons que les taux d'erreur n'apparaissent pas toujours avec la même valeur pour l'hyper-paramètre ***MAX_EPOCH***, ce qui est dû au "shuffle" initial des données typique du perceptron. Dans le cas illustré, le taux d'erreur le plus bas est trouvé lorsque ***MAX_EPOCH*** vaut 16, mais l'amélioration est très faible par rapport à l'évaluation quand sa valeur est 12 et en comparaison avec le temps passé à effectuer des calculs.

5. Données hors domaine

Un des intérêts d'un modèle qui apprend au fur et à mesure est de pouvoir l'utiliser sur de nouvelles données, afin de vérifier son efficacité dans différents contextes. Dans notre cas, notre précision est encore insuffisante dans le domaine pour prétendre faire des évaluations hors domaine. Des essais à titre informatif hors domaine ont donné un taux d’erreur de plus de 70%.

Cependant, nous avons conçu une matrice de confusion qui nous permet déjà d'identifier les erreurs les plus fréquentes dans nos évaluations actuelles, et qui pourrait nous aider à identifier les erreurs persistantes hors domaine, afin d'éventuellement ajuster les caractéristiques.
Dans la matrice de confusion ci-dessous portant sur les évaluations ***"in-domain"***, nous remarquons que les confusions ont lieu le plus souvent sur les paires d'étiquettes suivantes :

- Adposition et ponctuation
- Nom et adjectif
- Nom et verbe
- Adposition et déterminant



	DET	ADJ	_	NUM	VERB	ADP	PART	CCONJ	X	INTJ	ADV	NOUN	PROPN	PRON	SYM	SCONJ	AUX	PUNCT
DET	0	8	0	8	5	111	0	4	0	0	6	9	8	12	0	8	16	110
ADJ	11	0	2	3	71	9	0	1	0	0	29	206	27	8	1	0	1	9
_	0	0	0	0	0	0	0	1	0	0	0	6	0	1	0	1	0	3
NUM	25	1	1	0	3	14	0	1	0	0	0	35	5	4	2	1	1	28
VERB	5	26	4	4	0	28	1	3	1	0	43	116	14	14	0	1	37	19
ADP	102	2	2	3	28	0	0	16	0	0	12	13	4	51	4	9	24	227
PART	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
CCONJ	11	0	19	1	1	52	0	0	0	0	2	1	2	21	0	4	9	111
X	2	0	0	0	0	0	0	0	0	0	0	0	11	1	0	0	0	2
INTJ	0	0	0	0	0	0	0	0	0	0	0	0	5	1	0	0	0	0
ADV	25	24	6	6	91	38	0	3	1	0	0	77	21	41	0	11	27	47
NOUN	18	19	6	16	39	14	0	2	0	0	12	0	80	19	3	0	6	22
PROPN	15	1	0	0	1	1	0	0	3	0	0	17	0	4	0	0	0	2
PRON	59	5	0	4	10	21	0	3	0	0	10	27	3	0	0	5	26	66
SYM	0	1	0	0	0	2	0	0	1	0	0	2	1	0	0	0	0	26
SCONJ	1	1	0	0	5	32	0	1	0	0	0	0	4	17	0	0	1	40
AUX	18	3	0	3	33	24	2	5	1	0	7	16	0	8	0	1	0	40
PUNCT	37	0	20	0	2	109	0	13	0	0	3	6	0	9	3	6	16	0

Avec une matrice similaire appliquée aux résultats sur les données hors-domaine, nous pourrions comparer les erreurs les plus fréquentes. Si de nouvelles paires d'étiquettes apparaissent, il faudrait alors se concentrer sur ces étiquettes.

6. Conclusion

Notre perceptron ne produit pas des évaluations satisfaisantes. Le taux d'erreur est encore trop élevé pour appliquer notre outil sur des données réelles.