Ré-entrainement Et Évaluation De Modules Pour L'Étiquetage De Partie Du Discours

Laura Darenne
M2 TAL IM
Inalco
darenne.laura@outlook.fr

Abstract

Dans le domaine du Traitement Automatique des langues, il est important de bien choisir les modules avec lesquels travailler, car leur qualité impacte nécessairement les résultats des recherches. Un module avec de très bons résultats, mais consommant trop d'énergie et de temps, n'est pas toujours le choix optimal. L'étude se propose donc de tester plusieurs modèles d'étiquetage de partie du discours pour le chinois classique, de les comparer et d'identifier les mesures à prendre en compte dans l'évaluation d'un modèle.

Keywords: chinois classique, évaluation, étiquetage en parties du discours, module

1. Introduction

Il n'est pas rare dans le domaine du Traitement Automatique des Langues (TAL) de se retrouver face à un nombre significatif de modules pour une tâche donnée. Comment choisir le plus adéquat pour son projet ? Quel est le plus efficace ? Le plus fiable ? La décision finale peut également prendre en compte des contraintes qui dépendent de l'étude sur laquelle on travaille, les plus courantes étant les coûts en termes de temps et d'énergie. Le défi est donc de déterminer quel module répond le mieux aux exigences d'une tâche données.

Les modèles d'étiquetages de partie du discours pour le chinois classique, bien qu'ils restent encore peu nombreux, se sont multipliés depuis la campagne EvaHan 2022 pour le chinois classique[1]. Cette étude se propose donc de, modestement, comparer quelques modèles et d'identifier les mesures à prendre en compte dans l'évaluation d'un modèle d'étiquetage. Le code se trouve sur le dépôt github https://github.com/pmagistry/TALA540A/tree/LD_tpfinal.

2. Etat de l'art

Cette expérience de comparaison de modules d'étiquetage de partie du discours pour le chinois classique s'est faite sur trois modules : le module Jiayan[2] et Spacy[3].

Jiayan est un modèle d'étiquetage statistique utilisant les champs aléatoires conditionnels (conditional random fields ou CRFs en anglais). Un modèle est proposé, mais celui-ci a été entraîné sur des corpus avec des étiquettes de *Harbin (CTD)*. Cela rend la comparaison difficile, bien qu'un tableau de correspondance existe[4]. Cependant, des fonctions d'entraînement de modèles sont proposées sur le dépôt github ce qui m'a permis d'avoir des modèles avec des étiquettes *UD Chinese HK*.

Spacy est un module très connu pour beaucoup de tâches en Traitement Automatique des Langues. L'évaluation a été faite sur ses différents modèles pré-entraînés pour le chinois mandarin (https://spacy.io/models/zh). J'ai également entrainé mes propres modèles en jouant avec les paramètres proposés.

3. Données

Le corpus est le corpus UD Classical Chinese Kyoto[5]. L'étiquetage en partie du discours a été faite avec 14 étiquettes d'Universal Dependencies : ADP, ADV, AUX, CCONJ, INTJ, NOUN, NUM, PART, PRON, PROPN, PUNCT, SCONJ, SYM, VERB. Le texte de 433168 tokens n'est pas ponctué, n'a pas été tokenisé par des espaces, et ne contient que des caractères chinois traditionnels.

Il existe trois versions du corpus de ré-entraînement. Le corpus *sentence* est un collage des phrases du fichier conllu en fonction du titre de chaque partie pour en faire des paragraphes. Par exemple, les tables "KR1h0004_012_par1_1-4" à "KR1h0004_012_par1_65-69". Le corpus *table* est le corpus tel qu'il a été découpé de base et le corpus *word* est le corpus découpé caractère par caractère.

[1] https://circse.github.io/LT4HALA/2022/EvaHan.html
[2] https://github.com/jiaeyan/Jiayan/
[3] https://spacy.io/

Modèle	F-mesure sans OOV	F-mesure avec 00V
original	56.49%	42.82%
sentence	89.96%	54.62%
table	90.52%	54.36%
word	80.57%	28.21%

Table 1: résultats des évaluations des modèles jiayan et CRFs

4. Evaluation

4.1. Module Jiayan

Le tableau 1 montre que le modèle Jiayan est celui qui a obtenu le plus faible score. Cependant, il reste difficile de le comparer aux autres modèles, car il y a d'autres paramètres à prendre en compte. Ce module a été entraîné avec un corpus annoté avec des étiquettes différentes et le tableau de conversion[6] est loin d'être parfait. Il est beaucoup plus difficile de convertir des étiquettes Harbin en étiquettes UD que des étiquettes Penn Chinese.

Le tableau montre aussi que l'entraînement du modèle mot à mot donne d'assez faible résultat lorsqu'il rencontre des caractères pour la première fois. C'est très probablement parce qu'il ne peut pas prendre en compte dans son calcul la notion de séquence de mot et de partie du discours dans son calcul.

Les résultats des modèles *sentence* et *table* sont quasiment identiques. Le corpus d'entraînement n'est pas très grand, mais il semble que la longueur de la séquence influe assez peu sur nos résultats lorsqu'on ne se soucie pas du vocabulaire. Cependant, il reste un problème de généralisation du modèle face à des données qu'il ne connait pas.

4.2. Module Spacy

4.2.1. modèles Spacy

Modèle	F-mesure sans OOV	F-mesure avec 00V
zh_core_web_sm	46.59%	32.82%
zh_core_web_md	51.03%	33.59%
zh_core_web_lg	50.72%	37.95%
zh_core_web_trf	65.94%	41.03%

Table 2: résultats des évaluations des modèles spacy

Le tableau 2 montre les résultats très bas des modèles Spacy. Ce n'est pas une surprise, car les modèles ont été entraînés sur un corpus ponctué en chinois mandarin. Les sorties de ses modèles ne sont pas du tout exploitables pour des études sur le chinois classique. Cependant, Spacy propose des scripts pour entraîner soi-même son propre modèle avec ses propres modèles. C'est ce que nous avons essayé.

4.2.2. modèles ré-entraînées

Modèle	F-mesure sans OOV	F-mesure avec 00V
1: vs=100, w=3	89.5%	46.92%
2: vs=50, w=3	89.27%	48.21%
3: vs=200, w=3	89.2%	42.82%
4: vs=100, w=5	89.37%	46.67%
5: vs=100, w=10	89.43%	44.1%

Table 3: résultats des évaluations des modèles ré-entraînés avec le corpus table

Modèle	F-mesure sans OOV	F-mesure avec 00V
6: vs=100, w=3	89.23%	46.41%
7: vs=50, w=3	89.23%	44.87%
8: vs=200, w=3	89.18%	45.64%
9: vs=100, w=5	89.14%	44.62%
10: vs=100, w=10	89.1%	47.18%

Table 4: résultats des évaluations des modèles ré-entraînés avec le corpus sentence

Nous avons ré-entraîné les modèles avec les corpus table et sentence en jouant sur les paramètres *vector_size* (vs, dimension des vecteurs) et *window* (w, taille de la fenêtre). Les résultats entre les différents modèles sont quasiment identiques malgré la modification des paramètres des vecteurs. Il peut y avoir plusieurs explications.

Une première explication est que notre corpus n'est pas très grand rendant les modifications moins conséquentes. Il est aussi possible que nous n'ayons pas choisis des paramètres *vector_size* et *window* assez différents pour nos expériences. Il se peut aussi que le modèle d'entraînement Spacy est fait de telle manière que les modifications de paramètres des vecteurs aient peu d'impact sur l'entraînement. Enfin, une troisième explication serait le choix des hyperparamètres des modèles d'apprentissage. Les seuls que l'on puisse vraiment modifier sont la taille des batchs et le nombre d'epochs, et les entraînements se sont tous arrêtés au bout de 8 ou 12 epochs. Les vecteurs influent plus sur le nombre d'epochs permettant d'atteindre la fin de l'entraînement plutôt que sur les résultats.

[4] R. Poiret et al, 2023.

[5] https://universaldependencies.org/treebanks/lzh_kyoto/

[6] Voir Annexe 6.1.

Il faut également noter que la langue choisie pour obtenir le fichier de configuration est le chinois (*chinese*), et non le chinois classique. Sans rentrer dans les détails, le chinois mandarin et le chinois classique sont assez différentes en termes de syntaxe et de grammaire. Il est tout à fait possible que ce détail est sont importance pendant l'entraînement des modèles.

4.3. Résultats: le temps de traitement

Le tableau 5 contient les temps d'exécutions du chargement de chacun des modèles avec les données test du corpus. On peut voir une assez grosse différence de temps d'exécution entre les modèles statistiques CRFs et les modèles de réseaux neuronaux Spacy. Les modèles CRFs ont un temps d'entraînement et un temps de chargement très court pour des résultats F-mesure quasi-similaire aux modèles neuronaux.

Cependant, il ne faut pas oublier que le corpus de données d'entraînement que nous utilisons reste assez petit. Pour un plus grand jeu de données avec des textes de différentes époques, de différents styles d'écriture et de thématiques très différentes, il est tout à fait possible que les modèles Spacy, beaucoup plus grands que ce que nous avons, soient plus performants que les modèles statistiques CRFs malgré le temps de traitement plus grand.

Modèle	real time	user time
jiayan table	0m21,434s	0m20,257s
jiayan sentence	0m22,956s	0m20,090s
spacy 1	3m28,886s	3m26,472s
spacy 2	3m30,167s	3m27,785s
spacy 3	3m31,256s	3m28,211s
spacy 4	3m27,991s	3m25,856s
spacy 5	3m29,602s	3m27,617s

Table 5: résultats des évaluations des modèles avec la commande 'time'

4.4. Résultats: l'empreinte énergétique

Le tableau 6 montre les résultats des mesures[7] effectuées par le module pyJoules pour connaître l'empreinte énergétique des différents modèles. Sans surprise, les modèles de réseaux neuronaux sont beaucoup plus énergivores pour la même tâche effectuée. Cela peut paraître anecdotique pour l'étiquetage d'un corpus test, mais le chiffre devient tout de suite conséquent si le modèle est utilisé pour étiqueter un corpus très important.

[7]Voir ici https://github.com/powerapi-ng/pyJoules #rapl-domain-description pour l'explication des différents titres de colonnes

Modèle	package_0
jiayan table	143 062 989
jiayan sentence	136 670 732
spacy 1	1 788 428 649
spacy 2	1 462 889 871
spacy 3	1 640 567 089
spacy 4	1 700 593 120
spacy 5	1 519 091 936

Table 6: résultats des évaluations des modèles avec le module 'pyJoules'

4.5. Résultats: les matrice de confusion

La dernière question que nous nous sommes posées, c'est : Quelles sont les erreurs les plus fréquentes des modèles d'étiquetage ? Les couples d'étiquettes problématiques sont actuellement assez prédictibles. On retrouve "VERB-AUX", "VERB-ADV", "VERB-NOUN", "VERB-PROPN", "NOUN-PROPN".

Il y a plusieurs explications à ces erreurs. Elles sont identiques à tous les modèles et peuvent donc être liées aux données d'entraînement, pas assez nombreuses, pas assez diversifiées, ect. Également, le chinois classique est une langue assez complexe à étudier pour un module qui ne peut pas se reposer sur la ponctuation du texte, inexistante à la base.

Enfin, une autre raison peut être l'ambiguïté des caractères chinois. Les verbes ne se conjuguent pas. Les noms ne peuvent se distinguer des verbes avec un affixe particulier marquant le singulier/pluriel, le masculin/féminin. Cela rend l'analyse difficile lorsqu'un même caractère peut avoir plusieurs étiquettes grammaticales. Dans ces cas-là, il faut un modèle plus performant, avec certainement plus de données et plus de temps d'entraînement.

5. Bibliographie

R. Poiret, T. S. Wong, J. Leec, K. Gerdesd, and H. Leung. 2023. Universal dependencies for mandarin chinese. *Language Resources and Evaluation*, 57(2):673–710.

6. Annexe

6.1. tableau de conversion des parties du discours

UD Chinese HK	**Harbin (CDT)**
ADJ	a, m
DET	r, b
CCONJ, SCONJ	c
ADV	d, h
INTJ	e
NUM	m
NOUN	q, n, ni, nl, nt, h
PROPN	nh, ns, nz, z
ADP	nd, p, k
PRON	r
AUX	u, v
PART	u, k
VERB	v
PUNCT, SYM	wp
automatiquement correct	i, j
absent dans corpus test	o, g, x, ws

Table 7: tableau utilisé pour l'évaluation du module Jiayan (codé à partir du tableau de l'article R. Poiret et al, 2023.)