Université Paris-Sud Master AIC

Extraction d'information

Compte Rendu 2

Étudiants :
Yang Mo

Superviseur :
Anne-Laure Ligozat

06 october 2017

Travaux pratique

1 Objectif :

L'objectif de ce tp est de de réaliser la reconnaissance des entités nommés d'une certains manières, notamment sur une outil dédié 'Wapiti'. Wapiti est une programme qui permet d'apprendre depuis des corpus et traîner une modèle pour marquer les étiquettes aux texts données. Wapiti travail principalement dans les deux modes : *Train mode* et *Label mode*. Dans le train mode, Wapiti a besoin des donnée d'apprentissage comme import et traîner une modèle commme export. Une pattern fiche sera demandé pour établir une tableau d'observation en définissant des features ajoutés. Dans label mode, on peut utiliser le modèle acquis pour marquer les étiquettes aux textes. Les informations de performance sont aussi disponible si les textes sont déjà étiquetés.

Le premier étape dans ce tp, on va d'abord

tester les performances de l'annotation en entités nommée avec un ficher de patron très simple qui est déjà donnée. Ensuite on va modifier le ficher de patron pour améliorer la performance de notre modèle. En plus, on va également enrichir les corpus avec des ressources extérieur, comme ça on peut avoir une base de connaissance comment ajouter les étiquettes aux corpus. Finalement on va évaluer la performance de notre modèle et faire comparaison de la même fonction avec outil nltk en python.

2 Tester les performances de l'annotation en entités nommées avec un fichier de patron de base

ficher de patron simple :

Unigram
u1 :%x[-2,0]
u2 :%x[-1,0]
u3 :%x[0,0]

u4 :%x[1,0]
u5 :%x[2,0]

Bigram
u6 :%x[-1,0]%x[0,0]
u7 :%x[1,0]%x[0,0]

En utilisant le ficher de patron donnée, on a les résultats suivant :

```
mogolola@mogolola-VirtualBox:~/Wapiti$ ./wapiti label -c -m model1 doc/eng.test
test_resultat1
* Load model
* Label sequences
      1000 sequences labeled     8.29%/46.20%
      2000 sequences labeled     6.76%/47.45%
      3000 sequences labeled     6.35%/43.63%
    Nb sequences  : 3684
    Token error   :  6.61%
    Sequence error: 42.37%
* Per label statistics
    O       Pr=0.95  Rc=0.99  F1=0.97
    I-ORG   Pr=0.82  Rc=0.62  F1=0.70
    I-MISC  Pr=0.82  Rc=0.67  F1=0.74
    I-PER   Pr=0.88  Rc=0.70  F1=0.78
    I-LOC   Pr=0.89  Rc=0.70  F1=0.79
    B-LOC   Pr=0.00  Rc=0.00  F1=-nan
    B-MISC  Pr=0.00  Rc=0.00  F1=-nan
    B-ORG   Pr=-nan  Rc=0.00  F1=-nan
* Done
```

FIGURE 1 – précision de prévu d'apprentissage :ficher de patron simple, corpus initial

3 Modification de ficher de patron

Pour avoir mieux performance de reconnaissance les entités nommées, on a mis à jours deux types des éléments dans le ficher de patron :

1. ajouter plus de l'étiquette(les colonnes, morpho-syntaxique par exemple)
2. ajouter des expression régulières pour obtenir plus de feature dans le context

voici la liste de features on a défini dans le fichier de patron :

```
# Unigram
u1:%x[-2,0]
u2:%x[-1,0]
u3:%x[ 0,0]
u4:%x[ 1,0]
u5:%x[ 2,0]

# Bigram
u6:%x[-1,0]/%x[0,0]
u7:%x[ 1,0]/%x[0,0]

#label morpho-syntaxique
u6:%x[-2,1]
u7:%x[-1,1]
u8:%x[ 0,1]
u9:%x[ 1,1]
u10:%x[ 2,1]

#Caps?
u13:%t[-1,0,"\u"]
u14:%t[0,0,"\u"]
u15:%t[1,0,"\u"]

#Begin Caps?
u16:%t[ 0,0,"^\u"]


#ALL Caps?
u17:%t[0,0,"^\u*$"]


#Prefix
u18:%m[0,0,"^.?"]
u19:%m[0,0,"^.?.?"]
u20:%m[0,0,"^.?.?.?"]
u21:%m[0,0,"^.?.?.?.?"]

#surfix
u22:%m[ 0,0,".?$"]
u23:%m[ 0,0,".?.?$"]
u24:%m[ 0,0,".?.?.?$"]
u25:%m[ 0,0,".?.?.?.?$"]

#puncation?
u26:%t[-1,0,"\p"]
u27:%t[0,0,"\p"]
u28:%t[1,0,"\p"]

#punctuation inside?
u30:%t[0,0,".\p."]

#number?
u31:%t[-1,0,"\d"]
u32:%t[0,0,"\d"]
u33:%t[1,0,"\d"]

#All number?
u34:%t[0,0,"^\d*$"]

#label word in geolist?
u35:%x[-2,2]
u36:%x[-1,2]
u37:%x[ 0,2]
u38:%x[ 1,2]
u39:%x[ 2,2]
```

FIGURE 2 – les contenu de ficher de patron modifié

après appliquer ce ficher de patron sur l'entraînement d'une modèle, on a reévaluer la performance de ce modèle. Les résultats :

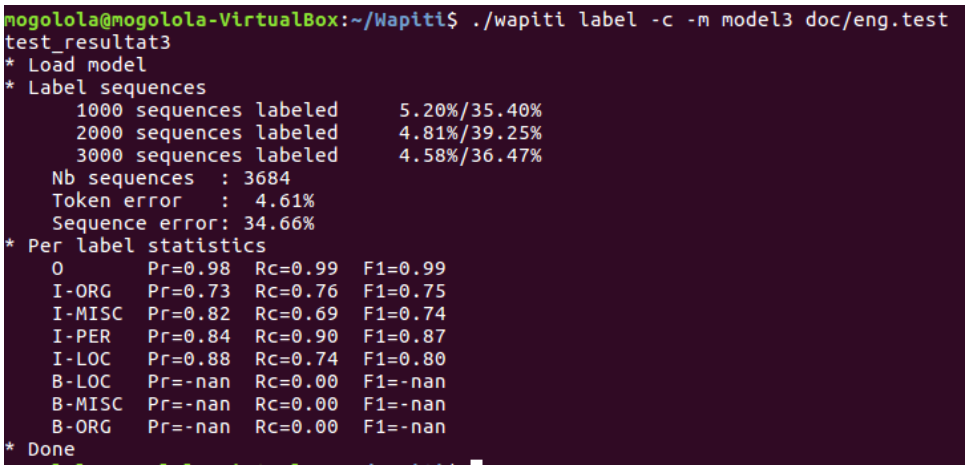

FIGURE 3 – précision de prévu d'apprentissage :ficher de patron modifié, corpus initial

On vois évidemment que le précision sont un peu élevé par rapport à l'avant.

4 Enrichissez les corpus

Pour enrichir les corpus, on utilise aussi une ressource lexique pour ajouter plus de étiquettes. On a construire une liste de nom géographique ci dessous :

```
['Afghanistan', 'Albania', 'Antarctica', 'Algeria', 'American Samoa', 'Andorra', 'Angola', 'Anti
gua and Barbuda', 'Azerbaijan', 'Argentina', 'Australia', 'Austria', 'Bahamas', 'Bahrain', 'Bang
ladesh', 'Armenia', 'Barbados', 'Belgium', 'Bermuda', 'Bhutan', 'Bolivia', 'Bosnia and Herzegovi
na', 'Botswana', 'Bouvet Island', 'Brazil', 'Belize', 'British Indian Ocean Territory', 'Solomon
Islands', 'British Virgin Islands', 'Brunei Darussalam', 'Bulgaria', 'Myanmar', 'Burundi', 'Bela
rus', 'Cambodia', 'Cameroon', 'Canada', 'Cape Verde', 'Cayman Islands', 'Central African', 'Sri
Lanka', 'Chad', 'Chile', 'China', 'Taiwan', 'Christmas Island', 'Cocos (Keeling) Islands', 'Colo
mbia', 'Comoros', 'Mayotte', 'Republic of the Congo', 'The Democratic Republic Of The Congo', 'C
ook Islands', 'Costa Rica', 'Croatia', 'Cuba', 'Cyprus', 'Czech Republic', 'Benin', 'Denmark', '
Dominica', 'Dominican Republic', 'Ecuador', 'El Salvador', 'Equatorial Guinea', 'Ethiopia', 'Eri
trea', 'Estonia', 'Faroe Islands', 'Falkland Islands', 'South Georgia and the South Sandwich Isl
ands', 'Fiji', 'Finland', '\xc3\x85land Islands', 'France', 'French Guiana', 'French Polynesia',
'French Southern Territories', 'Djibouti', 'Gabon', 'Georgia', 'Gambia', 'Occupied Palestinian T
erritory', 'Germany', 'Ghana', 'Gibraltar', 'Kiribati', 'Greece', 'Greenland', 'Grenada', 'Guade
loupe', 'Guam', 'Guatemala', 'Guinea', 'Guyana', 'Haiti', 'Heard Island and McDonald Islands', '
Vatican City State', 'Honduras', 'Hong Kong', 'Hungary', 'Iceland', 'India', 'Indonesia', 'Islam
ic Republic of Iran', 'Iraq', 'Ireland', 'Israel', 'Italy', "C\xc3\xb4te d'Ivoire", 'Jamaica', '
Japan', 'Kazakhstan', 'Jordan', 'Kenya', "Democratic People's Republic of Korea", 'Republic of K
orea', 'Kuwait', 'Kyrgyzstan', "Lao People's Democratic Republic", 'Lebanon', 'Lesotho', 'Latvia
', 'Liberia', 'Libyan Arab Jamahiriya', 'Liechtenstein', 'Lithuania', 'Luxembourg', 'Macao', 'Ma
dagascar', 'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta', 'Martinique', 'Mauritania', 'Mauri
tius', 'Mexico', 'Monaco', 'Mongolia', 'Republic of Moldova', 'Montserrat', 'Morocco', 'Mozambiq
ue', 'Oman', 'Namibia', 'Nauru', 'Nepal', 'Netherlands', 'Netherlands Antilles', 'Aruba', 'New C
aledonia', 'Vanuatu', 'New Zealand', 'Nicaragua', 'Niger', 'Nigeria', 'Niue', 'Norfolk Island',
'Norway', 'Northern Mariana Islands', 'United States Minor Outlying Islands', 'Federated States
of Micronesia', 'Marshall Islands', 'Palau', 'Pakistan', 'Panama', 'Papua New Guinea', 'Paraguay
', 'Peru', 'Philippines', 'Pitcairn', 'Poland', 'Portugal', 'Guinea-Bissau', 'Timor-Leste', 'Pue
rto Rico', 'Qatar', 'R\xc3\xa9union', 'Romania', 'Russian Federation', 'Rwanda', 'Saint Helena',
'Saint Kitts and Nevis', 'Anguilla', 'Saint Lucia', 'Saint-Pierre and Miquelon', 'Saint Vincent
and the Grenadines', 'San Marino', 'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Seychell
es', 'Sierra Leone', 'Singapore', 'Slovakia', 'Vietnam', 'Slovenia', 'Somalia', 'South Africa',
'Zimbabwe', 'Spain', 'Western Sahara', 'Sudan', 'Suriname', 'Svalbard and Jan Mayen', 'Swaziland
', 'Sweden', 'Switzerland', 'Syrian Arab Republic', 'Tajikistan', 'Thailand', 'Togo', 'Tokelau',
'Tonga', 'Trinidad and Tobago', 'United Arab Emirates', 'Tunisia', 'Turkey', 'Turkmenistan', 'Tu
rks and Caicos Islands', 'Tuvalu', 'Uganda', 'Ukraine', 'The Former Yugoslav Republic of Macedon
ia', 'Egypt', 'United Kingdom', 'Isle of Man', 'United Republic Of Tanzania', 'United States', '
U.S. Virgin Islands', 'Burkina Faso', 'Uruguay', 'Uzbekistan', 'Venezuela', 'Wallis and Futuna',
'Samoa', 'Yemen', 'Serbia and Montenegro', 'Zambia']
```

FIGURE 4 – list géographique construi

On utilise ce liste pour vérifier dans le texte est-ce que le token qu'on cherche est compris dans ce liste. On ajoute une nouvelle colonne sur les corpus pour enregistrer la réponse. Si oui, on met la valeur '1', si non on met '0'. Après on fait ça, on traîne un nouveau modèle et tester encore sur ce modèle. Voici la performance :

```
mogolola@mogolola-VirtualBox:~/Wapiti$ ./wapiti label -c -m model6 doc/eng
new test_restutat6
* Load model
* Label sequences
      1000 sequences labeled      4.62%/32.60%
      2000 sequences labeled      4.28%/35.65%
      3000 sequences labeled      3.97%/32.43%
    Nb sequences  : 3684
    Token error   :  3.99%
    Sequence error: 31.19%
* Per label statistics
    O       Pr=0.99  Rc=0.99  F1=0.99
    I-ORG   Pr=0.79  Rc=0.75  F1=0.77
    I-MISC  Pr=0.77  Rc=0.77  F1=0.77
    I-PER   Pr=0.85  Rc=0.92  F1=0.89
    I-LOC   Pr=0.86  Rc=0.79  F1=0.83
    B-LOC   Pr=-nan  Rc=0.00  F1=-nan
    B-MISC  Pr=-nan  Rc=0.00  F1=-nan
    B-ORG   Pr=-nan  Rc=0.00  F1=-nan
* Done
```

FIGURE 5 – précision de prévu d'apprentissage :ficher de patron modifié, corpus agrandi

l'erreur de token et l'erreur de séquence sont brassé encore un peu, le modèle qu'on a traîné est amélioré avec réussi.

5 comparer les résultats avec ceux de nltk

Dans nltk, on peut utiliser certain taggers pour marqué les étiquettes (POS-Tagging). Le manière est pareil que ce qu'on fait avec ou-

til Wapiti. On va tester sur une corpus de nltk. D'abord, on a chosi une texte d'entraînement et une texte à tester (les textes sont déjà marqué). Et puis on va définir certains taggers(comme features dans wapiti) pour apprentissage. La différence c'est que nltk ne demande pas forcément un modèle par utilisateur. les taggers sont suffisant pour réaliser l'apprentissage. On compose une unigram tagger et une bigram tagger(le cas le plus simple), et on a finalement la performance : le taux d'erreur de tokens est 15%. C'est une résultat adaptable, mais pour mieux performance, il faut évidemment utiliser plus de taggers et peut-être plus de étiquettes

Pour la conclusion, wapiti est une outil facile à utiliser. Après bien configurer toutes les paramètre on a trouvé que le modèle a une performance très comptetant. l'erreur de tokens peut être brassé sous 4%. Mais les inconvénient c'est que le corpus n'est pas suffisamment large. Si on a besoin de encore augmenter la précision de

prédiction, il faut enrichir et agrandir les corpus certainement. Par contre, nltk a une corpus très riche. Et les taggers est aussi facile à utiliser. Il y a aussi des façon à améliorer le modèle d'apprentissage en ajoutant plus de features et étiquettes. Dans un mots, je pense que les deux outils sont également puissants Wapiti a beaucoup de 'killer features' comme la souplex de construire un modèle d'apprentissage, pour moi c'est assez intéressant pour une petit corpus. Si il y a certains manières de reformaliser les corpus déjà marqué à une forme qui peut être identifié par wapiti (en traitement de données, pas encore vérifié), l'outil wapiti va être beaucoup plus pratique.