Université de Montréal

IFT6285 (TALN) — Devoir8
Analyse syntaxique avec NLTK

Contact :
Philippe Langlais
RALI/DIRO
Université de Montréal

+1 514 343 61 11 ext: 47494
felipe@iro.umontreal.ca
http://www.iro.umontreal.ca/~felipe/

dernière compilation : 11 novembre 2020 (23:43)

Contexte

Dans le devoir 7, vous avez utilisé NLTK pour faire de l'analyse morpho-syntaxique, vous allez maintenant l'utiliser pour faire de l'analyse syntaxique. NLTK vous donne accès à de nombreux analyseurs qui sont décrits ici. Vous aurez également besoin de quelques connaissances sur la façon dont une grammaire est représentée que vous trouverez ici.

Données

Le Penn Tree Bank est une ressource payante distribuée par LDC (Linguistic Data Consortium) offrant des phrases manuellement arborées syntaxiquement et NLTK vous offre un accès gratuit à un sous-ensemble de ces phrases. Vous pouvez obtenir les arbres de ces phrases comme suit :

```
from nltk.corpus import treebank
train = treebank.fileids()[:190]
test = treebank.fileids()[190:]

for item in train:
    for tree in treebank.parsed_sents(item):
         print(tree)
```

Dans ce devoir, vous garderez ce découpage train/test : 190 fichiers pour l'entraînement, 10 pour les tests.

À faire

1. Prenez connaissance de cette documentation simple sur les facilités principales d'analyse syntaxique offertes par NLTK.
2. Utilisez les arbres des phrases d'entraînement pour entraîner une grammaire PCFG. Vous pouvez utiliser la fonction `induce_pcfg` pour cela. Un exemple de code se trouve aux cellules 20 et 21 de la section PCFG du document sus-mentionné.

   Comme vous le constaterez, le nombre de règles peut être important, aussi pouvez vous garder un nombre donné de règles (les plus fréquentes par exemple) avant d'apprendre la grammaire.
3. Utiliser cette grammaire pour analyser les phrases de test à l'aide de l'analyseur ViterbiParser qui ne requiert pas des règles de production au format CNF. Vous êtes bien sûr libre d'en tester d'autres.

   Vous remarquerez très rapidement que la présence d'un mot inconnu dans une phrase à analyser lève une exception. Vous devez régler ce problème de façon à pouvoir analyser une phrase contenant des mots inconnus. Une solution possible à ce problème consiste à ajouter des règles $A \rightarrow UNK$ pour tout non terminal A intervenant dans une règle lexicale (e.g. $A \rightarrow the$).
4. Curieusement, les mesures parseval ne sont pas (à ma connaissance) disponibles dans NLTK. Aussi, vous évaluerez votre analyseur sur la tâche d'étiquetage morpho-syntaxique (POS tagging). Vous devez donc récupérer la séquence d'étiquettes morpho-syntaxiques de l'arbre le plus probable et la comparer à l'étiquetage de l'arbre de référence.
5. Vous intéresserez au moins aux points suivants :
   — le temps d'analyse en fonction de la longueur des phrases,
   — le nombre d'analyses par phrase,
   — la performance en fonction de facteurs comme la longueur des phrases, ou le nombre de mots inconnus dans la phrase à analyser,
   — l'influence du nombre de règles dans la grammaire.

Vous devez produire un **rapport** d'au plus 3 pages (format pdf, en français ou en anglais) qui contient les analyses des approches qui vous sont demandées.

Remise

La remise est à faire sur Studium sous le libellé `devoir8`. Vous devez remettre votre code, votre rapport (format pdf, texte en anglais ou en français) dans une archive (gzip, tar, tar.gz) dont le nom est préfixé de `devoir8-name1` ou `devoir8-name1-name2` selon que vous remettez seul ou a deux, où `name1` et `name2` sont à remplacer par l'identité des personnes faisant la remise (`prénom_nom`). Assurez vous que le nom des personnes impliquées dans le devoir soit indiqué sur tous les documents remis (code et rapport). Le devoir est à remettre en groupe d'au plus deux personnes au plus tard vendredi 20 novembre à 23h59.

Note : Aucun modèle n'est demandé.