THE AUTOCOMPLETE PROJECT

in Python

Morgane Dehareng

M2 TAL IM 2017-2018

Présentation générale du projet

Objectif

Complétion au niveau d'une séquence de mots, façon Google.

Scénario : l'utilisateur saisit deux mots, le système lui propose des candidats pour le 3ème mot.

Langue

J'ai choisi l'anglais comme langue de travail. En effet, il n'y a pas encore de ressources utilisables en TAL dans ma langue de spécialité (albanais). En outre, suite à un entretien lors du mois de décembre, j'effectuerai peut-être mon stage de fin d'études au sein de la société Softbank Robotics. Lors de cet entretien, les linguistes présents m'ont annoncé que ma langue de travail serait l'anglais. Il m'a paru ainsi intéressant de travailler précisément sur cette langue. De plus, une de mes tâches consisterait en l'écriture de règles à base d'expressions régulières pour gérer les inputs et outputs d'un robot humanoïde. Les résultats des différentes fonctions que j'implémente dans ce projet pourrait éventuellement m'aider à générer ces règles (ex : I'm (not (a | going | sure) | going to)).

Ressources

Corpus

- Newspaper2012.txt : 3.000.000 de phrases issues d'un journal britannique datant de 2012
- Newspaper2013.txt : 1.000.000 de phrases issues d'un journal britannique datant de 2013

Ces corpus proviennent tous deux de la Leipzig Corpora Collection et sont répertoriés dans le catalogue VLO.

Liens :

https://vlo.clarin.eu/record?1&docId=hdl_58_11022_47_0000-0000-223E-5&fqType=languageCode:or&fq=languageCode:code:eng&index=1&count=69244

https://vlo.clarin.eu/record?2&docId=hdl_58_11022_47_0000-0000-8F18-5&fqType=languageCode:or&fq=languageCode:code:eng&index=2&count=69244

Modules

En plus des modules déjà utilisés (marisa_trie et sys), j'ai utilisé les modules suivants :

- re : un module permettant de manipuler les expressions régulières
- nltk : un module facilitant le traitement informatique du langage naturel
- sent_tokenize : une fonction permettant la tokenisation d'une chaine donnée en phrases (entrée : une chaine, sortie : une liste des phrases)
- RegexpTokenizer : une fonction permettant une tokenisation à l'aide d'une expression régulière (entrée : une chaine, sortie : une liste de tokens)
- ngrams : une fonction permettant de générer des séquences de ngrams à partir d'une chaine de caractères (entrée : une chaine, sortie : une liste des séquences de n-grams)

Méthodologie

L'utilisation des modules cités ci-dessus m'a imposé de respecter les types des données d'entrée que nécessitaient ces modules. J'ai donc procédé comme suit :

- lecture du fichier et transformation de son contenu en une chaine de caractères
- tokenisation de cette chaine en phrase à l'aide de sent_tokenize
- tokenisation de chaque phrase en mots à l'aide de RegexpTokenizer. (l'expression régulière utilisée permet d'ignorer tous les caractères non-alphanumériques à l'exception de l'apostrophe, primordiale en anglais) et génération des n-grams à l'aide de la fonction n-grams
- création d'un dictionnaire pour compter la fréquence de chaque n-gram généré
- création d'une liste qui contient les n-grams les plus fréquents

Difficultés rencontrées

- la taille du corpus Newspaper2012.txt : il m'a été impossible d'exécuter mon script sur base de ce corpus (erreur Flask, volume des données trop important)
- la non-normalisation des données (plusieurs types d'apostrophes notamment)
- l'intégration de l'apostrophe (ex : it's et non it s)
- la gestion des paramètres d'entrée et de sortie des fonctions utilisées

Solutions proposées

- travail sur le corpus Newspaper2013.txt, plus petit
- normalisation de l'apostrophe uniquement, le reste des caractères non-normalisés n'étant pas nécessaire pour le projet
- expression régulière permettant d'intégrer l'apostrophe (équivalent de "\w+" + l'apostrophe)
- fonctions génériques permettant la réutilisation du code pour d'autres projets (ex : counter)

Remarque : le dossier "script" comporte une première version du code, completion(v1).py ainsi qu'une version plus compacte, completion.py (utilisation des listes en compréhension).