Diagnostic automatique de la langue (majoritaire) d'un document

Introduction

On a réalisé en python3 un ensemble de scripts permettant de déterminer automatiquement la langue majoritaire d'un document. Pour y arriver, on exploite la propriété linguistique générale qui est que les mots courts et fréquents sont caractéristiques d'une langue. Il nous faut donc commencer par construire automatiquement une base de données à partir de documents dont on connaît la langue.

Vue d'ensemble des scripts

guess_language.py est le point d'entrée, il gère les arguments de la ligne de commande et supporte les formats texte « pur » et HTML.

htmlutils.py sert à lire un fichier HTML et à en récupérer le contenu textuel.

linguistic.py est utilisé pour identifier les mots courts et fréquents dans un texte.

language_guesser.py permet de lire, écrire et utiliser une base de données de mots caractéristiques.

argparse.py est la routinethèque utilisée pour parser les arguments de la ligne de commande, je l'inclue dans l'archive car elle n'est pas intégrée à python3.1.

Corpus

Les articles sont tirés de Wikipédia, ils ont été choisis en parcourant la liste des pages longues du wiki Grec avec un script pour déterminer son existence dans les 6 ou 8 autres langues désirées (find_long_multilingual_wiki_articles.py).

Ils sont séparés dans deux dossiers *a* et *b*, *a* sert à nourrir la base de données et *b* contient les documents de test. *a* contient 8 dossiers contenant chacun 6 fichiers HTML, un pour chacune des langues choisies : el, fr, en, de, nl, es. *b* est constitué de 15 dossiers contenant chacun 8 fichiers HTML, les deux langues supplémentaires sont : sv (suédois) et pt (portugais).

Algorithmes

Mots caractéristiques

On constitue deux ensembles : celui des *limit* mots les plus fréquents et celui de tous les mots de longueur inférieure ou égale à *max_word_size* (5 par défaut), puis on en récupère l'intersection. Si le résultat comporte moins de 10 mots on lève une exception.

On définit *limit* à 15 pour la base de données (valeur « optimale » déterminée expérimentalement) et à 50 pour l'analyse d'un document.

On retourne un dictionnaire avec les mots comme clés et la fréquence relative à l'ensemble des mots caractéristiques comme valeurs, c'est à dire le nombre d'occurrences du mot sur le nombre total d'occurrences des mots caractéristiques, elle est plus lisible que la fréquence relative au nombre de mots du document car elle est plus élevée.

Constitution de la base de données

L'enjeu de la constitution de la base de données est de ne conserver que les mots caractéristiques de la langue en évitant les mots courts caractéristiques des sujets des documents utilisés pour sa création.

Pour remplir cet objectif on divise par 2 fois sa longueur la fréquence de tout nouveau mot ainsi que tout mot étant déjà dans la base de données mais pas présent dans le document en cours d'analyse. De cette façon les termes spécifiques à un domaine descendent dans la liste et sont remplacés au fur et à mesure par d'autres mots dont la fréquence est plus élevée.

Comparaison de mots caractéristiques

On calcule simplement le pourcentage de mots caractéristiques inclus dans la base de données qui sont également présents dans ceux du documents. On rappelle que la base de données en contient 15 au maximum et ceux du document 50 au maximum, donc on a de bonnes chances qu'une majorité des 15 se trouve dans les 50. En pratique, avec le corpus choisi, l'algorithme identifie la bonne langue dans 100% des cas où la langue existe dans la base de données et ne produit aucun faux-positif dans le cas contraire.

Déterminer si deux probabilités sont suffisamment éloignées

Une fois que l'on a calculé la probabilité pour que le texte traité soit de telle langue pour chaque langue de la base de données, il faut estimer si deux ou plus de ces probabilités sont trop proches pour que l'on puisse déterminer avec certitude la langue du document.

Dans ce but on calcule un coefficient à partir des deux probabilités en question. Ce coefficient est proportionnel à la distance entre les deux probabilités et inversement proportionnel à leur moyenne+0,5. On vérifie enfin que ce coefficient est supérieur à 0.2.

Quelques exemples parleront mieux :

p1	p2	distance	average	coeff	>0.2
55	35	20	45	0,21	VRAI
55	50	5	52,5	0,05	FAUX
100	80	20	90	0,14	FAUX
100	75	25	87,5	0,18	FAUX
100	70	30	85	0,22	VRAI
90	70	20	80	0,15	FAUX
90	60	30	75	0,24	VRAI

Tests

Création de la base de données :

```
 $ ./guess_language.py -rl corpus/a/*/*.html
```

Tests sur les documents dont la langue est dans la base données :

```
 $ ./guess_language.py corpus/b/*/{el,fr,en,de,nl,es}.html -tq
0.0% wrong guesses, 0.0% unknown, 100.0% good guesses, 0.0% uncertain
```

Tests sur les articles dont la langue n'est pas dans dans la BDD :

```
 $ ./guess_language.py corpus/b/*/{sv,pt}.html -tq
0.0% wrong guesses, 100.0% unknown, 0.0% good guesses, 0.0% uncertain
unmatched were:  pt:50.0%  sv:50.0%
```

Conclusion

Les résultats sont parfaits, il faudrait essayer sur un corpus plus fourni avec plus de langues différentes et provenant de sources différentes pour voir si la qualité se maintient.

Annexe

Détails pour le suédois et le portugais

```
 $ ./guess_language.py corpus/b/*/sv.html
no language in the database matches, closest is es with a score of 26.7%
no language in the database matches, closest is nl with a score of 20.0%
no language in the database matches, closest is es with a score of 33.3%
no language in the database matches, closest is en with a score of 20.0%
no language in the database matches, closest is nl with a score of 13.3%
no language in the database matches, closest is es with a score of 13.3%
no language in the database matches, closest is nl with a score of 20.0%
no language in the database matches, closest is en with a score of 20.0%
no language in the database matches, closest is en with a score of 26.7%
no language in the database matches, closest is es with a score of 20.0%
no language in the database matches, closest is nl with a score of 20.0%
no language in the database matches, closest is fr with a score of 13.3%
no language in the database matches, closest is nl with a score of 20.0%
no language in the database matches, closest is nl with a score of 20.0%
no language in the database matches, closest is nl with a score of 13.3%

 $ ./guess_language.py corpus/b/*/pt.html
no language in the database matches, closest is es with a score of 40.0%
no language in the database matches, closest is en with a score of 20.0%
no language in the database matches, closest is es with a score of 33.3%
no language in the database matches, closest is es with a score of 20.0%
no language in the database matches, closest is es with a score of 26.7%
no language in the database matches, closest is es with a score of 26.7%
no language in the database matches, closest is en with a score of 26.7%
no language in the database matches, closest is es with a score of 26.7%
no language in the database matches, closest is en with a score of 40.0%
no language in the database matches, closest is es with a score of 26.7%
no language in the database matches, closest is fr with a score of 6.7%
no language in the database matches, closest is es with a score of 26.7%
no language in the database matches, closest is es with a score of 26.7%
no language in the database matches, closest is es with a score of 26.7%
no language in the database matches, closest is es with a score of 26.7%
```

On observe que le script identifie l'espagnol comme étant le plus proche du portugais dans 73% des cas mais le néerlandais comme étant le plus proche du suédois dans seulement 47% des cas.