Travaux Pratiques –n°6

Modèles de langage statistiques (boite à outils SLM du CMU)

Objectif du TP (sous Fedora) :

Comprendre comment construire des modèles de langage statistiques (N-grams) et quels rôles jouent les principaux paramètres qui influent sur cette construction. Evaluer des modèles de langages et savoir interpréter les résultats obtenus.

Nous utiliserons la boite à outils SLM du CMU (Carnegie Mellon University) a priori déjà installée sous **`/tools/CMU-Cam_Toolkit_v2`**. Les commandes de cette boîte à outils se trouvent sous **`/tools/CMU-Cam_Toolkit_v2/bin`**

1) SLM (Statistical Language Modeling) : Documentation et syntaxe

Les principaux éléments de syntaxe des commandes nécessaires au TP sont résumés dans ce document. Cependant, vous trouverez tout le détail des commandes sur la page web du CMU dédiée[1] dont une copie pdf est fournie dans le dossier **TP_MLS** que vous récupèrerez sur moodle.

RAPPEL (Informatique 1A) :

Les commandes du toolkit sont des commandes unix qui utilisent les **mécanismes de redirection** :
Redirection entrée : **`commande < fichier_entree`**
Redirection sortie : **`commande > fichier_sortie`**
Redirection sortie et concaténation : **`commande >> fichier_sortie`**
Redirection sortie erreur : **`commande >& fichier_erreur`**

Les parties entre [ ] correspondent aux options de la commande.
Lorsqu'on utilise une option, on ne met pas les []. Si on ne l'utilise pas explicitement, c'est la (première) valeur par défaut mentionnée dans la documentation qui est utilisée.

Respecter les extensions de fichiers mentionnées : `.text, .wfreq, .vocab,…`

Respecter les conventions de nommage en mettant votre nom dans tous les noms de fichiers générés ainsi que les caractéristiques associées (N2 ou N3, methode_prélèvement utilisée, …).
Exemple : `Dupond.vocab Dupond_N2.wngram Dupond_N3_absolute.arpa`

[1] http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html

2) Construction et Evaluation d'un modèle de langage N-gram

La Figure 1, fournie par le CMU, résume le processus général de création et d'évaluation d'un modèle de langage statistique N-grams. L'objectif du TP est de mettre en œuvre ce processus.

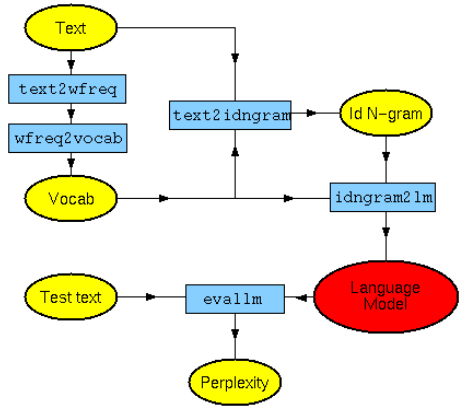

Figure 1 : Principe de fonctionnement de la construction et de l'évaluation d'un modèle de langage tel que décrit dans la documentation du CMU extrait de http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html

3) Corpus de textes

Il s'agit d'utiliser les fichiers correspondant à la transcription manuelle (après normalisation) de fichiers audio, enregistrés sur la chaîne radio FRANCE INTER. Les fichiers audio sont généralement utilisés pour l'apprentissage des modèles acoustico-phonétiques nécessaires pour la mise en œuvre d'applications de transcription automatique d'émissions de radio. Les transcriptions manuelles sont de leur côté utilisées pour **l'apprentissage des modèles de langage statistiques** et pour **l'évaluation des modèles construits**. Vous trouverez un corpus d'apprentissage et un corpus de test.

4) TP à réaliser

CONSIGNE :
Pour chacune des étapes du TP, récupérer les traces des exécutions des commandes dans un fichier texte nommé comme suit : **`VOTRE_NOM_TP_MLS.txt`**
Répondre également de manière claire et lisible aux questions posées dans le sujet.
Ce fichier sera à déposer sur moodle en fin de séance.

4.1) Apprentissage des modèles de langage

Génération de la liste de mots et calcul du nombre d'occurrence

QUESTION 1 : En utilisant l'ensemble des fichiers dédiés à l'apprentissage (**CORPUS_APP**), utiliser l'outil **text2wfreq** pour générer la liste des mots présents dans ces fichiers ainsi que leur nombre d'occurrences (fichier **.wfreq**).

```
text2wfreq     [ -hash 1000000 ]
               [ -verbosity 2 ]
               < .text     > .wfreq
```

QUESTION 2 : A l'aide de la commande unix **sort**, trier la liste de façon à la classer les mots par nombre d'occurrences décroissant. Quels sont les mots les plus fréquents ? A quoi correspondent-ils et pourquoi ?

Génération du vocabulaire

QUESTION 3 : Utiliser l'outil **wfreq2vocab** pour obtenir la liste des mots distincts correspondant au vocabulaire extrait de ce corpus d'apprentissage (fichier **.vocab**). Examinez cette liste.

```
wfreq2vocab  [ -top 20000 | -gt 10]
             [ -records 1000000 ]
             [ -verbosity 2]
             < .wfreq     > .vocab
```

Liste des bigrammes et trigrammes et nombres d'occurrences associés

QUESTION 4 : Utiliser l'outil **text2wngram** pour générer l'ensemble des paires de mots présentes dans le corpus d'apprentissage (fichier **.wngram**) et compter leur nombre d'occurrences.

```
      text2wngram [ -n 3 ] [ -temp /usr/tmp/ ][ -chars n ]
           [ -words m ][ -gzip | -compress ]
           [ -verbosity 2 ]
           < .text     > .wngram
```

Quels sont les bigrammes les plus fréquents ? A quoi correspondent-ils ?

QUESTION 5 : Utiliser l'outil **text2idngram** pour générer une autre version de cette liste (fichier **.idngram**) et comparer le résultat avec le précédent. Ne pas oublier l'option **(-write_ascii**) pour générer le résultat sous forme ascii et non binaire plus difficile à lire mais plus efficace en terme de traitement …


```
      text2idngram -vocab .vocab
           [ -buffer 100 ] [ -temp /usr/tmp/ ] [ -files 20 ]
           [ -gzip | -compress ] [ -n 3 ] [ -write_ascii ]
           [ -fof_size 10 ] [ -verbosity 2 ]
           < .text > .idngram
```

QUESTION 6 : Examinez la trace affichée pendant l'exécution de la commande et commentez les éléments qui vous semblent significatifs.

Relancer sans l'option **-write_ascii** pour obtenir une version binaire à utiliser par la suite (question 8).

QUESTION 7 : Reprendre les mêmes étapes (Q4, Q5 et Q6) mais générer les fichiers correspondants aux triplets de mots.

Génération de différents modèles en fonction de la méthode de prélèvement choisie

QUESTION 8 : A partir du fichier **.idngram** binaire généré précédemment, construire un **modèle de langage bigramme** pour chaque méthode de prélèvement proposée. Utiliser l'outil **idngram2lm** et l'option **–arpa** pour générer un fichier au format ARPA **(.arpa).**

```
idngram2lm -idngram .idngram
           -vocab .vocab
           -arpa .arpa | -binary .binlm
         [ -context .ccs ]
         [ -calc_mem | -buffer 100 | -spec_num y ... z ]
         [ -vocab_type 1 ][ -oov_fraction 0.5 ]
         [ -linear | -absolute | -good_turing | -witten_bell ]
         [ -disc_ranges 1 7 7 ] [ -cutoffs 0 ... 0 ]
         [ -min_unicount 0 ][ -zeroton_fraction 1.0 ]
         [ -ascii_input | -bin_input ][ -n 3 ]
         [ -verbosity 2 ] [ -four_byte_counts ]
         [ -two_byte_bo_weights
         [ -min_bo_weight -3.2 ] [ -max_bo_weight 2.5 ]
            [ -out_of_range_bo_weights 10000 ] ]
```

Remarque : vous pouvez consulter la documentation section « Discounting strategies » pour voir les détails concernant les méthodes de prélèvement et calcul du discount ratio (δ ou d(r)).

Quelles informations sont données dans la trace. Quelles sont celles qui vous paraissent significatives. Comparer les valeurs obtenues lors de la génération des différents modèles.

QUESTION 9 : Reprendre l'étape précédente (Q8) pour générer un **modèle de langage trigramme** avec chacune des méthodes de prélèvement disponibles.

Vous disposez normalement de 4 modèles de langage bigrammes et 4 modèles de langage trigrammes appris sur le même corpus. Vous allez les tester sur le même fichier de test.

4.2 Evaluation des modèles de langage

Calcul de la perplexité et du taux de mots hors vocabulaire

Utiliser le fichier qui se trouve dans **`CORPUS_TEST1`** pour effectuer cette évaluation.

QUESTION 10 : Evaluer chacun des 8 modèles générés précédemment en utilisant la combinaison de commandes suivante :

```
echo "perplexity –text corpus_test -probs fichier.probs
      -oovs fichier.oovs -annotate fichier.annote " |
      evallm –arpa modele_a_tester
avec
      evallm [ -binary .binlm | -arpa .arpa [ -context .ccs ]
```

3.3. Résultats des évaluations

QUESTION 11 : Présenter sous forme de tableau ou équivalent (cf. exemple) les différents résultats obtenus (**`OOV, Entropie, Perplexité`**) et déterminer quel est le modèle a priori le plus performant.

Méthode de prélèvement	Evaluation Modèle Bigramme	Evaluation Modèle Trigramme
Linear		
Absolute		
Good Turing		
Witten Bell		

QUESTION 12 : Consulter les différents fichiers générés (**`.probs, .oovs, .annote`**). Copier-coller les 20 premières lignes (commande **`head`**) de chaque dans le fichier txt et commenter.

QUESTION 13 : Utiliser votre meilleur modèle et évaluer sur le fichier contenu dans **`CORPUS_TEST2`**. Copier-coller les résultats et commenter.

EVALUATION du TP : Aller sur moodle et déposer le fichier `VOTRE_NOM_TP_MLS.txt` rempli au fur et à mesure de la séance