Université de Montréal

IFT6285 (TALN) — Projet 2
De l'ordre dans les mots

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 : 15 novembre 2021 (21:24)

Contexte

Dans ce projet, vous allez étudier le problème de réordonner un sac de mots. Par exemple en tant que locuteur de l'anglais, vous n'avez pas de difficulté à produire pour le sac de mots *[? everything big why to become does have such issue a]* la phrase *why everything does have to become such a big issue ?*.

Votre but est d'étudier des approches automatiques à ce problème. Aucune contrainte particulière n'est imposée ici, si ce n'est que vous ne pouvez utiliser que les données d'entraînement qui sont détaillées ci-après. Vous ne pouvez donc pas envisager d'approche consistant à interroger un moteur de recherche en ligne (ce qui ne serait d'ailleurs pas très écologique). Il est en revanche assez simple d'utiliser les modèles de langue que vous avez déployés à l'aide de `kenlm` dans le devoir 2.

Données

Pour développer votre approche, vous ne devez utiliser que les 99 tranches du 1B-word corpus (voir devoir 1) dont une copie est disponible au DIRO :

```
/u/demorali/corpora/1g-word-lm-benchmark-r13output/
    training-monolingual.tokenized.shuffled/
```

Trois corpus de développement (projet2-dev.tar.gz) vous permettent d'étudier le comportement de votre solution :

news contient 1000 phrases (`news.ref`) et les sacs de mots associés (`news.test`). Les données sont issues de la même distribution que les données d'entraînement,

hans contient 1000 phrases (`hans.ref`) et les sacs de mots associés (`hans.test`). Les données sont issues du corpus des débats parlementaires canadiens,

euro contient 1000 phrases (`euro.ref`) et les sacs de mots associés (`euro.test`). Les données sont issues du corpus Europarl.

Les données de développement ont été préparées à l'aide du programme pre-process.py de la façon suivante[1] (où <fichier> est un fichier texte contenant une phrase par ligne) :

```
cat <fichier> | python3 pre-process.py --min=5 --max=25
                                       --no='" - -- # www http'
                                       --nb=1000 --out=<fichier>.ref
                                       --lower >! <fichier>.test
```

Ce programme utilise la *tokenisation* produite par le modèle en_core_web_sm de spacy et élimine les phrases trop courtes ou trop longues ainsi que celles contenant les mots indiqués dans l'option --no.[2] Les phrases (1000 au total) ainsi découpées en mots sont converties en minuscule.

Peu avant la remise, des corpus de test seront disponibles sur lesquels vous devrez appliquer votre meilleur algorithme. La même préparation sera appliquée pour ces données.

À faire

Vous devez remettre un rapport au format pdf (anglais ou français) d'au plus 6 pages qui décrit vos expérimentations. Vous devez en particulier :

1. Décrire les métriques d'évaluation utilisées pour évaluer vos solutions,
2. Développer une solution qui répond au mieux au problème. Vous prendrez soin d'analyser les performances de votre approche en comparaison à des approches de base (*baseline*) et analyserez son comportement (temps de calculs, etc.),
3. Identifier votre meilleure approche. Une approche (non simpliste) comporte des méta-paramètres[3] et vous préciserez les méta-paramètres de l'approche que vous proposez.
4. Analyser les limites de votre meilleurs approche.
5. À l'approche de la remise, des corpus de test seront distribués. Vous appliquerez votre meilleure approche à ces fichiers. Pour chaque fichier

1. Syntaxe csh.
2. Où les mots sont séparés par un espace.
3. Seuils, choix d'architecture, sélection des données d'entraînement, algorithmes, etc.

de test fic, vous devez produire un fichier fic-noms qui contient le même nombre de lignes que le fichier et où `noms` indique le nom des personnes impliquées dans le projet. [4]

Votre rapport doit prendre la forme d'un mini article scientifique et doit témoigner de votre curiosité. Quelques lignes de python suffisent à réaliser avec des outils comme `kenlm` une approche "décente". Vous pouvez également regarder du côté de la plateforme transformer qui offre du code prêt à l'emploi pour interroger des modèles *transformers* pré-entraînés. Je vous recommande notamment la lecture de la section `Causal Language Modeling` du tour d'horizon. Le point est qu'aucune de ces solutions ne sera parfaite et votre tâche est d'améliorer une solution que vous avez mise en place (avec ou sans l'aide de tels outils).

Notation

La notation n'est pas corrélée à la performance de vos approches, mais à la **curiosité** que vous développerez et à votre esprit d'**analyse**. La clarté et l'informativité de vos rapports sont deux critères importants.

Voici quelques questions que nous poserons lors de la correction :

— les modèles *baselines* sont-ils raisonnables ?
— les efforts mis en place pour dépasser les performances du modèle de base sont-ils justifiés, bien décrits, conséquents ?
— le rapport est-il clair, les analyses sont-elles informatives ?

Si vous utilisez du code écrit par autrui, vous devez le mentionner dans votre rapport.

Remise

La remise est à faire sur Studium sous le libellé `projet2`. Vous devez remettre votre code, votre rapport (format pdf, texte en anglais ou en français), ainsi que les fichiers de test traités dans une archive (gzip, tar,

4. Les liens sur les fichiers nommés n'ont pour rôle que d'illustrer les formats, et non les attentes en terme de qualité des sorties produites : un modèle trigramme entraîné avec kenlm sur le premier million de phrases d'entraînement a produit de manière gloutonne (gauche-droite) les "phrases" montrées ici.

tar.gz) dont le nom est préfixé de `projet2-noms`, où `noms` est à remplacer par l'identité des personnes (`prénom_nom`) impliquées dans le projet. Donc si j'avais à remettre seul mon solutionnaire au projet2, je le ferais sous le nom `projet2-philippe_langlais.tar.gz`. Assurez vous que le nom des personnes impliquées dans le projet soit indiqué sur le rapport. Le projet est à remettre en groupe d'au plus trois personnes au plus tard samedi 18 décembre à 23h59.