Projets - Méthodes neuronales
Master LiTL 2021-2022

chloe.braud@irit.fr

Le projet est à réaliser en groupe de **3 étudiant-e-s**.
- **14 février 2022**, chaque groupe devra envoyer par email : le code (par exemple, un fichier notebook python ou des fichiers python[1]), les données utilisées (éventuellement lien vers un drive) et un rapport (format pdf).
- **15 février 2022**, chaque groupe présentera son projet en classe (environ 15mn de présentation).

Sujets de projet

Les projets correspondent à une tâche et 2 ensembles de données (2 langues ou 2 domaines).

Sentiment analysis

- Tâche : classification binaire
- Ensemble de données de base : English movie reviews, IMDB: https://ai.stanford.edu/~amaas/data/sentiment/
- Adaptation multilingue : utiliser les données françaises des TP
- Adaptation multi-domaines : English product reviews, Multi-domain Sentiment Dataset https://www.cs.jhu.edu/~mdredze/datasets/sentiment/

Twitter data - crisis

- Tâche : classification binaire (tweet utile ou inutile) ou multiclasse
- Ensemble de données de base : choisir un ensemble de données (i.e. une crise) sur https://crisisnlp.qcri.org/lrec2016/lrec2016.html
- Adaptation multilingue : choisir soit le corpus en français soit l'un des deux en espagnol
- Adaptation multi-domaines : adapter à un type de crise différent (e.g. Earthquake à Typhoon, ou à War and conflicts)

Twitter data - Fake news detection

- Tâche : classification binaire
- Ensemble de données : commencer par développer un système sur les données anglaises https://github.com/bigheiniu/MM-COVID
- Adaptation multilingue : choisir une autre langue parmis les 5 autres disponibles

Name tagging

- Tâche : classification (either identify name mentions OR assign a type to each mention)
- Ensemble de données : commencer par développer un système sur les données anglaises

[1] Merci d'ajouter un bref Readme expliquant comment utiliser le code dans ce cas et un script d'exécution.

- https://elisa-ie.github.io/wikiann/
- Adaptation multilingue : choisir une autre langue parmis toutes celles disponibles

Discourse segmentation

- Tâche : classification binaire (un mot est une frontière de segment ou non)
- Ensemble de données : commencer par développer un système sur les données anglaises disponible, i.e. GUM dataset
- Site de la shared task : https://sites.google.com/georgetown.edu/disrpt2021/home
- Site avec les données : https://github.com/disrpt/sharedtask2021/tree/main/data
- Adaptation multilingue : choisir une autre langue parmis toutes celles disponibles

Description des systèmes attendus

Il est demandé aux étudiants :
- de lire 1 article correspondant à un système neuronal sur ces données (à décrire dans le rapport)
- de développer un système utilisant une architecture neuronale pour résoudre la tâche
- D'étendre ce système dans un cadre multilingue ou multi-domaines.

A- Données

La première étape du projet consiste à récupérer les données et à écrire un programme permettant de lire ces données pour leur utilisation en PyTorch. Éventuellement, un pré-traitement pourra être appliqué.

Vous pourrez vous limiter à un sous-ensemble des données dans le cas où le corpus est vraiment gros afin de limiter les temps de traitement. Vous pouvez essayer différentes tailles et en choisir une 'raisonnable'.

B- Système langue / domaine source

Le système doit permettre de réaliser la tâche sur les données de 'base' (anglais, domaine source). L'architecture (neuronale) est laissée au choix des étudiants. Il est cependant requis :
- de procéder à une évaluation rigoureuse, notamment en optimisant les hyper-paramètres sur un sous-ensemble de données séparé, en donnant différents scores (globaux et par classe) et en analysant les erreurs du système
- de proposer une évaluation supplémentaire fondée sur un critère au choix : par exemple, rapport entre performance et taille des embeddings en entrée / taille de la couche cachée / nombre de données d'entraînement …

Le système pourra être amélioré en implémentant un ou plusieurs des éléments suivants :
- un système de référence non neuronal (e.g. Naïve Bayes avec représentation bag of words)
- la comparaison de différents types d'embeddings, aléatoires et / ou pré-entraînés
- la comparaison de différentes architectures
- l'ajout d'autres types de traits
- des visualisations
- ...

Note : commencez par développer toute la chaîne pour un petit sous-ensemble des données, afin d'obtenir rapidement des résultats. Vous pourrez ensuite augmenter la quantité de données en entrée. De même, choisissez initialement des valeurs basses pour les dimensions des couches pour accélérer les calculs.

C- Extension multilingue ou multi-domaines

Enfin, les étudiants devront étendre leur système soit à une nouvelle langue (e.g. de l'anglais vers le français) OU à un nouveau domaine (e.g. avis de films vers avis de produits). La langue / le domaine de référence est dit "source", la nouvelle langue / le nouveau domaine est dit "cible".

Il est proposé d'étendre le système de la manière suivante :

1- **Cas multilingue** : adapter grâce à des embeddings crosslingues[2], i.e. :
- utiliser la version de la langue source de ces embeddings dans le système de base = modèle du système source de référence ; reporter les scores obtenus au test sur la langue source.
- utiliser la version de la langue cible de ces embeddings à l'évaluation sur les données de test cible = transfert cross-lingue ; reporter les scores obtenus au test sur la langue cible.
- Le transfert est-il efficace, a-t-on des performances similaires pour les deux langues ?

2- **Cas multi-domaine**s : adapter grâce au corpus d'évaluation
- évaluer sur l'ensemble de test cible le système entraîné et optimisé sur les données source
- créer un nouveau modèle entraîné sur les données source mais optimisé sur un sous-ensemble des données cibles, puis évalué sur le même ensemble de test cible
- Avez-vous amélioré les performances et donc adapté votre système au nouveau domaine ?

Note : Dans les deux cas, l'architecture ne change pas par rapport à B, seuls changent les embeddings en entrée, ou la façon de construire le modèle. Pour simplifier, vous pouvez conserver les meilleures valeurs d'hyper-paramètres obtenus à l'étape précédente.

Description du rapport attendu

Le rapport (max. 6 pages) doit décrire la tâche, les données utilisées, l'architecture du système, la procédure d'optimisation et d'évaluation, les scores obtenus, une analyse des résultats et les problèmes rencontrés. Il est aussi demandé de décrire brièvement un système neuronal existant construit pour la tâche et décrit dans un article.

[2] Attention : embeddings crosslingues et multilingues, c'est différent. Ce qu'on veut ce sont des embeddings alignés pour les différents langues, cf : https://fasttext.cc/docs/en/aligned-vectors.html