Université de Paris

Master Informatique : Apprentissage Machine pour la Science des données

Data Science II

NLP, Réduction de la dimension et Clustering (R et Python)

M. Nadif

Introduction

Les modèles d'intégration contextuelle de mots basés sur des transformateurs ont révolutionné d'une certaine manière l'état de l'art du langage naturel (NLP). Plusieurs travaux se sont intéressés aux représentations contextuelles fournies par ces modèles qui se sont avérées très efficaces. Cependant, jusqu'à présent, ce sont principalement les tâches supervisées (classification supervisé de textes, reconnaissance d'entités nommées, systèmes de réponse aux questions, etc.) qui ont grandement bénéficié du gain de performance apporté par de telles représentations. Ici nous allons nous focaliser sur la classification non supervisée et particulièrement à la réduction de la dimension et le clustering (classification non supervisée). Notre objectif est de proposer des classes de mots sémantiquement proches à partir de leurs représentations obtenues par deux modèles très populaires à 12 couches Bert [1] de Google et RoBERTa [2] de Facebook.

Deux Datasets

Les données à utiliser sont à importer sous R. Chaque Dataset est constitué de classes de mots à découvrir. Cependant et dans ce cas précis, nous connaissons les vraies classes et dans ce cas l'objectif de ce projet est aussi de valider une démarche non supervisée qu'on pourrait adopter pour n'importe quelle collection de mots dont les classes sont inconnues.

Apprentissage non supervisé

1. Expliquez le fonctionnement des deux modèles Bert et Roberta.

Pour chaque dataset, visualiser les 12 différentes couches grâce à une ACP appliquée à chaque couche. Que peut-on dire ?

2. Sachant que chaque couche peut être vue comme un groupe de variables qui sont les dimensions, vous pouvez viser à obtenir une visualisation simultanée en utilisant une Analyse Factorielle Multiple. Que peut-on dire
3. Visualiser le premier plan factoriel, commenter vos résultats.

Nous choisissons garder dans la suite les 20 premières composantes principales.

4. Réaliser un clustering via un algorithme k-means, puis une CAH avec différents critères d'agrégation. Comme le nombre de classes étant inconnu vous pourrez utiliser {Nbclust} vous permettant de tester une trentaine de critères sur la base de 2 à 6 classes. Que peut-on dire ?
5. Interpréter les classes obtenues.

Evaluation de la démarche

Nous connaissons les vraies classes et nous aimerions évaluer notre démarche pour classifier un ensemble de mots. Pour cela plusieurs critères sont utilisés dans la littérature pour évaluer la performance d'un algorithme de clustering (ou pour comparer deux partitionnements données) sur un jeu de données dont on connaît les classes. Sur la base de la matrice de confusion, voici les critères les plus populaires : l'accuracy, purity, NMI et ARI.

1. Etudiez la signification de chaque critère et montrer les avantages/inconvénients de chacun.
2. Dans la suite nous allons admettre que le nombre de classes de Data1 est 3 et celui de Data2 est 4. Comparer les résultats de k-means et ceux des différents algorithmes hiérarchiques selon le critère d'agrégation qui sont disponibles dans Nbclust. Pour ce faire construire les matrices de confusions et utiliser les différents critères citées ci-dessus. Que peut-on dire ?
3. A l'aide du premier plan factoriel, projeter les classes de chaque dataset. Que peut-on dire ?
4. A l'aide de l'analyse factorielles discriminante visualiser les classes. Que peut-on dire ?

Université de Paris

Pour aller plus loin

1. Vous avez utilisé l'ACP pour visualiser les classes. Le package {dimRed} offre d'autres méthodes de visualisation à utiliser. Commentez brièvement les méthodes choisies, les motivations de l'utilisation de telles méthodes et les résultats obtenues.
2. Pour le clustering, vous vous êtes principalement appuyés sur k-means ou CAH disponibles dans {Nbclust}, d'autres algorithmes peuvent être testés : {dbscan}, {hdbscan}, {deep-k-means} ? commenter brièvement ces méthodes dont le code est disponible. Que peut-on dire ?

Consignes

1. Ce travail peut être fait par binôme ou seul.
2. Toute ressemblance entre deux projets sera lourdement sanctionnée
3. La section « pour aller plus loin » est facultative mais un bonus est prévu.
4. Toutes les méthodes sont déjà implémentées, tous les packages utilisés doivent être référencés.
5. Me faire parvenir le script et le rapport séparé en format pdf.

Références

1. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert:Pre-training of deep bidirectional transformers for language understanding.arXivpreprint arXiv:1810.04805(2018)
2. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, OmerLevy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019).