Rapport TAL - TME 2 à 4

AYED Hatem, GUERIN Titouan

February 2024

1 Introduction

Dans ce rapport, nous allons essayer de résumer de façon claire et concise les TME 2 à 4 que nous avons traités pendant les différentes séances de TAL. Nous nous efforcerons de fournir, pour chacun des TME, une mise en contexte ainsi qu'une analyse des différents résultats expérimentaux.

2 TME 2

2.1 TME 2a - Sequence processing with HMMs and CRFs

2.1.1 Mise en Contexte

Le TME se concentre sur l'application des Hidden Markov Models (HMMs) et des Conditional Random Fields (CRFs) pour le traitement des séquences, en particulier pour l'étiquetage des parties du discours (POS) et le chunking dans des ensembles de données de NLP. Nous commençons par une stratégie basique (associer à chaque mot un POS fixe), puis nous passons aux HMMs. Enfin, nous testons un modèle basé sur les CRFs.

2.1.2 Mise en exergue des résultats expérimentaux

Nous travaillons sur le dataset CONLL 2000. Nous obtenons une accuracy de 0.76 avec le modèle simple consistant à associer à chaque mot un POS, de 0.82 avec les HMMs et de 0.99 avec les CRFs.

2.1.3 Conclusion

La progression de la précision de 0.76 pour le modèle de base à 0.82 avec les HMMs indique que même les approches de modélisation de séquences relativement simples peuvent fournir des améliorations substantielles. Cela est dû à la nature évidemment séquentielle du langage naturel. De même pour les CRFs qui arrivent à un résultat quasi parfait en exploitant avec un modèle bien plus complexe cette séquentialité.

2.2 TME 2b - Data mining and clustering

2.2.1 Mise en Contexte

Ce TME se concentre sur la représentation de documents bruts et non étiquetés à l'aide de techniques de clustering.

2.2.2 Analyse des résultats expérimentaux

Nous chargeons d'abord un ensemble de données (newsgroups), suivi d'une conversion en représentation BoW avec TF-IDF. Après avoir manipulé les WorldClouds, nous passons à un clustering par l'algorithme des K-means. Nous faisons d'abord une analyse qualitative : on se se concentre sur l'examen des mots les plus importants pour chaque cluster. Nous passons ensuite à une analyse quantitative. Les résultats montrent une grande variabilité dans la pureté des clusters, allant d'une pureté aussi basse que 0.103 pour le cluster 1 à une pureté parfaite de 1.0 pour le cluster 18. Cette variabilité indique que certains clusters sont très homogènes avec un label dominant clairement représenté (comme le cluster 18), tandis que d'autres sont plus hétérogènes, contenant une mixité plus large de labels (comme le cluster 1).

2.2.3 Conclusion

Ce TME sur le clustering et le data mining illustre l'importance d'appliquer et de combiner des analyses qualitatives et quantitatives pour évaluer et interpréter les résultats du clustering de documents textuels non étiquetés. Plusieurs approches possibles (K-means, LSA, LDA) sont illustrées dans ce TME.

3 TME 3

3.1 TME 3a - NLP and representation learning: Neural Embeddings, Text Classification

3.1.1 Mise en Contexte

Dans ce TME, nous mettons en place des modèles d'embedding vectoriels tels que Word2Vec qui a la particularité d'avoir accès à des relations telles que France + Capitale = Paris. Nous utilisons ensuite cet embedding vectoriel de mots pour passer à un embedding vectoriel de textes, en prenant (par exemple) la somme de tous les mots du texte.

3.1.2 Analyse des résultats expérimentaux

Le but étant de trouver la thématique de chaque texte (sentiment analysis), nous pouvons dès lors que nos textes ont été "embeddés" entraîner un classifieur. Nous obtenons, (en trichant un petit peu certes puisque nous avons utilisé la librairie nltk pour tokenizer nos mots au préalable) une accuracy de 0.87 sur notre dataset de test.

3.1.3 Conclusion

Pour aller plus loin, il aurait été intéressant de déterminer par exemple quel modèle Word2Vec (Skip-Gram ou CBOW) offre les meilleures performances pour la tâche spécifique de classification.

3.2 TME 3b - Word Embedding for Sequence Processing

3.2.1 Mise en Contexte

Le but de ce TME est de réadresser le problème de prédiction de POS vu au TME 2a en appliquant cette fois-ci un embedding vectoriel sur les mots. Voici la procéduire suivie : nous avons un dataset de train constitué de textes eux-mêmes constitués de mots, et pour chacun de ces mots nous avons leur POS correspondant (label). A l'aide d'un modèle d'embedding vectoriel pré-entraîné (word2vec ici) nous "embeddons" tous les mots de nos textes, en attribuant un vecteur random aux mots non présents dans le modèle. Nous pouvons dès lors entraîner un modèle de classification.

3.2.2 Analyse des résultats expérimentaux

La performance anormalement basse (accuracy de 0.69) obtenue avec le modèle de régression logistique, suggère plusieurs points de réflexion. Premièrement, cela peut indiquer que l'embedding vectoriel utilisé n'a pas réussi à capturer suffisamment les nuances sémantiques ou contextuelles nécessaires pour la prédiction séquentielle efficace. Deuxièmement, la régression logistique, en tant que modèle linéaire, pourrait ne pas être le choix optimal pour exploiter pleinement la richesse des embeddings vectoriels dans des tâches séquentielles complexes, comparativement aux modèles spécifiquement conçus pour les séquences telles que les HMM et les CRF, où nous avons pu constater dans le TME 2a que les résultats étaient bien meilleurs.

3.2.3 Conclusion

Cette expérience réaffirme l'importance de choisir des modèles et des représentations adaptées aux spécificités des tâches de NLP séquentiel.

4 TME 4

4.1 TME 4a - RNNs

4.1.1 Mise en Contexte

Ce TME se concentre sur l'apprentissage d'un modèle RNN pour prédire séquentiellement chaque caractère dans un texte donné.

4.1.2 Conclusion

N'ayant pas su faire tourner le modèle, il nous est assez difficile d'en dire plus malheureusement.

4.2 TME 4b - Transformers

4.2.1 Mise en Contexte

Ce TME explore l'utilisation des modèles Transformers, spécifiquement BERT pour l'analyse de sentiments. L'objectif est de calculer les embeddings BERT pour chaque review en utilisant le token CLS comme représentation de l'avis, puis d'entraîner un modèle de régression logistique sur ces embeddings extraits pour effectuer la classification des sentiments.

4.2.2 Conclusion

En entraînant un modèle de régression logistique sur les embeddings BERT, ce TME démontre la performances de ces architectures avancées pour la tâche de classification des sentiments.