IFT3335 – TP2

Utiliser la classification pour la désambiguïsation de sens de mots

Ce TP est à réaliser en groupe de 2 personnes (ou seul).
Date limite de remise :5 mai 2021, avant 23 :39.
Chaque jour de retard aura une pénalité de 10% de la note.
Ce TP correspond à 15% de la note globale.

Ce TP a pour but de pratiquer les algorithmes de classification et de les utiliser pour la désambiguïsation de sens de mots. Les algorithmes de classification sont déjà implantés dans la bibliothèque SkLearn. Votre travail dans ce TP consiste à utiliser SkLearn sur des collections de données et à examiner l'impact de différents algorithmes et les différentes caractéristiques (features).

1. Préparatifs

Pour vous familiariser avec SkLearn, commencez par vérifier que vous maitrisez les exercices pratiques avec SkLearn des démonstrations 10 et 11.

Découvrez et revoyez en particulier les fonctions suivantes :

1. Preprocess

Ceci vous permet de charger les données et faire des prétraitements sur les données, e.g. la sélection des données à traiter, transformation des données et attributs, etc.

Importez un ensemble de données existant dans le package (e.g. iris dataset, digits pour de la reconnaissance d'écriture…).

Les données sont chargées sous forme de listes.

2. Classification

Une fois les données chargées, vous pouvez maintenant choisir un algorithme et l'appliquer sur les données.

Une fois l'algorithme choisi, importez-le puis initialisez-le.

3. Sélection et pondération des features

Vous pouvez maintenant tenter de sélectionner des attributs à utiliser pour la classification.

Selon les données que vos traitez, vous pouvez avoir envie de ne pas considérer certaines dimensions ou certains features.

Il y a différentes méthodes pour sélectionner un sous ensemble d'attributs à utiliser dans la classification. Ceci est très utile quand vos données sont très bruitées, avec beaucoup d'attributs qui n'aident pas à la classification. Un nettoyage (une sélection) est très bénéfique dans ce cas. Cette sélection aide aussi à accélérer les traitements.

Jouez librement avec les datasets textuels inclus dans SkLearn. Notamment, vous devez transformer un texte en un ensemble d'attributs (chaque mot = 1 attribut). Après cette transformation, vous allez pouvoir utiliser les algorithmes de classification.

Cette transformation vous est proposée par SkLearn grâce aux classes CountVectorizer, et TfidfVectorizer.

Lisez les options proposées. Lors de votre choix, il vous est notamment possible de préciser si le résultat de cette transformation produit un ensemble d'attributs (mots) binaire (présent ou absent) ou avec un poids numérique (fréquence, tf transformé et avec idf).

Une pratique courante dans le domaine de classification de textes et de recherche d'information et de tronquer les mots pour ne garder que les racines. Par exemple, le mot « computer » sera tronqué en « comput ». Ceci a pour but de créer une représentation unique pour une famille de mots semblables (computer, computing, compute, computes, computed). Ce processus est appelé « stemming » (troncature).

Il y a des méthodes de stemming standards disponibles en python, dont celle proposée par la bibliothèque NLTK (voir ici).

2. La tâche de désambiguïsation de sens de mots

La désambiguïsation de sens de mots consiste à déterminer le sens d'un mot dont plusieurs sens sont acceptés :
Par exemple le mot « souris » qui peut faire référence à un animal ou à un périphérique informatique.
C'est une tâche de base pour la compréhension de textes. On effectue souvent cette tâche en utilisant une approche de classification : On suppose qu'on dispose d'un ensemble de textes (phrases) contenant des occurrences du mot ambigu, et que le sens de chaque occurrence du mot est annoté manuellement. En utilisant ces textes comme exemples, on entraîne un classifieur. Ce classifieur sera utilisé pour désambiguïser de nouveaux textes.

2.1. Corpus

Pour ce TP, nous allons utiliser un ensemble de phrases annotées, contenant le mot ambigu *interest*, qui peut correspondre à 6 sens différents, selon le dictionnaire Longman :

Sense 1 = 361 occurrences (15%) - readiness to give attention
Sense 2 = 11 occurrences (01%) - quality of causing attention to be given to
Sense 3 = 66 occurrences (03%) - activity, etc. that one gives attention to
Sense 4 = 178 occurrences (08%) - advantage, advancement or favor
Sense 5 = 500 occurrences (21%) - a share in a company or business
Sense 6 = 1252 occurrences (53%) - money paid for the use of money

Le texte annoté est le résultat d'une analyse de partie-de-discours + annotation de sens du mot ambigu. Voici un exemple :

```
[ yields/NNS ] on/IN [ money-market/JJ mutual/JJ funds/NNS ]
continued/VBD to/TO slide/VB ,/, amid/IN [ signs/NNS ] that/IN [
portfolio/NN managers/NNS ] expect/VBP [ further/JJ declines/NNS ]
in/IN [ interest_6/NN rates/NNS ] ./.
$$
[ longer/JJR maturities/NNS ] are/VBP thought/VBN to/TO indicate/VB
[ declining/VBG interest_6/NN rates/NNS ] because/IN [ they/PP ]
permit/VBP [ portfolio/NN managers/NNS ] to/TO retain/VB
relatively/RB [ higher/JJR rates/NNS ] for/IN [ a/DT longer/JJR
period/NN ] ./.
```

Dans cet exemple, les crochets [ ] enferment un groupe nominal. Chaque mot est suivi de sa catégorie grammaticale (e.g. /NNS), et le mot ambigu, *interest*, est annoté de son sens (_6). Les ponctuations sont elles-mêmes leur propre catégorie (comme dans ./.). Les phrases sont séparées par une ligne de $$.

Ce corpus contient 2369 instances de mot *interest*. Une description de ce corpus peut être trouvée ici : http://www.d.umn.edu/~tpederse/Data/README.int.txt. Le corpus est pris du site http://www.d.umn.edu/~tpederse/data.html.

2.2. Le processus de désambiguïsation

Pour déterminer le sens du mot, on utilise les informations sur son contexte. Le concept de contexte à la base peut signifier :

- L'ensemble des mots avant et les mots après (dans un sac de mots, sans ordre). Dans le premier exemple proposés ci-dessus, tient les 2 mots avant et 2 mots après (le mot *interest*) sont {*declines, in*, *rate*, .}.
- Les catégories des mots autour. Pour les mêmes 4 mots du premier exemple, nous allons avoir : « NNS », « IN », « NNS », « . » (la ponctuation). Ces catégories sont généralement prises en compte en ordre (C_{-2}=*NNS,* C_{-1}=*IN,* C_1=*NNS,* et C_2=*.*) afin de tenir compte de la structure syntaxique.

Ces deux groupes de caractéristiques sont ceux que vous devez utiliser au minimum. Mais il y a certaines variations possibles (que vous pouvez tester) :

- Lors de sélection des mots autour, négliger les mots outils (stopword en anglais) très fréquents et peu informatifs, tels que *in* ou les ponctuations. (*in* et *further* sont des mots outils – voir la liste de stopwords en anglais). Cette option est incluse dans les algorithmes SkLearn.
- Troncature des mots, en utilisant un algorithme de stemming.

Dans la littérature, d'autres types de caractéristiques ont été proposées et utilisées. On vous conseil de consulter la page suivante pour une présentation sommaire :
https://en.wikipedia.org/wiki/Word-sense_disambiguation
La présentation faite dans le cours vous donne aussi quelques autres types de caractéristique utiles. Vous êtes encouragés à les explorer. L'utilisation des caractéristiques supplémentaires sera prise en compte. Si ces caractéristiques

supplémentaires sont d'un nombre assez important, des points de bonus peuvent être accordés dans la correction.

2.3. Les tâches à réaliser

1. Vous devez faire un programme capable d'extraire les caractéristiques à partir des textes annotés.
   Réfléchissez en particulier au découpage du corpus, au choix des éléments utilisés pour la classification, aux mots nécessitant une transformation numérique…
2. Vous devez tester la performance de différents algorithmes de classification. Pour ce TP, on vous demande de tester au moins une version des algorithmes suivants disponibles sur SkLearn: Naive Bayes, arbre de décision, et MultiLayerPerceptron (en essayant différents nombres de neurones cachés). Utilisez d'abord les 2 types de caractéristique décrites en haut. Utilisez différentes tailles de fenêtre de contexte (1, 2, 3, …, même phrase), et observez les variations de performance de désambiguïsation en fonction de cette taille.
3. Réfléchir à des caractéristiques qu'il pourrait être intéressant d'explorer, et justifier pourquoi. Imaginer et implémenter ensuite une méthode intéressante pour les tester.
4. Commenter les résultats obtenus : sont-ils satisfaisants? Attendus? Qu'aurait-on pu proposer comme amélioration…

3. À rendre

Vous devez rendre tous les programmes utilisés pour l'extraction des caractéristiques. Faites une petite description de ces programmes dans votre rapport, et détaillez leur utilisation.

En plus des programmes, vous devez aussi rendre un rapport, de longueur comprise entre 5-10 pages. Dans votre rapport, vous devez décrire votre protocole de preprocessing, les expériences que vous avez réalisées, les résultats obtenus, et enfin des comparaisons et des analyses sur les résultats. Vos analyses doivent porter sur la performance des différents algorithmes, l'impact de différentes options - stemming, le nombre de neurones cachés, la taille de fenêtre pour la désambiguïsation, etc. Décrivez librement ce que vous observez d'intéressant dans ces expériences.

Enfin, une conclusion est attendue.

N'hésitez pas à ajouter toute remarque qui vous paraitrait judicieuse : quel point du protocole pourrait être modifié, les résultats sont-ils satisfaisants, …

Barèmes d'évaluation

- Programme d'extraction de caractéristiques : 3 points
- Test avec Naive Bayes : 3 points
- Test avec arbre de décision : 3 points
- Test avec MultiLayerPerceptron : 3 points
- Rapport : 3 points (y compris la description, les analyses et comparaisons entre différentes options, les analyses de résultats et les conclusions, ainsi que la clarté du rapport et l'orthographe)