LETTRES SORBONNE UNIVERSITÉ

2019-2020

M1SOL020

Epistémologie de l'Informatique

TD7
Annotation et Evaluation

Gaël Lejeune, Sorbonne Université

Inspiré d'Agnès Delaborde 2015-2016

Objectifs

- Comprendre les enjeux de l'annotation d'un corpus
- Savoir évaluer l'annotation d'un corpus

Attention, le dernier exercice est à rendre sur Moodle pour le 08/12/2019

1 Annotation linguistique

1.1 Annotation "simple"

Annotez chaque mot des phrases ci-après à l'aide des étiquettes " Nom ", " Adjectif ", " Verbe ", " Pronom ", " Déterminant ", " Adverbe " ou " Autres " (notez vos annotations sur papier ou dans un tableur, nous en aurons besoin plus tard).

- La belle ferme le voile.
- La petite brise la glace.

1.2 Ambiguïté grammaticale

Annotez le paragraphe ci-dessous et comparez vos résultats avec votre voisin.

- Une fois sortie de sa maison, elle se rend compte que la petite brise la glace, alors elle enfile un épais manteau.

2 Annotation par myriadisation (crowdsourcing)

Pour résoudre (une partie) des problèmes d'annotation, on peut avoir recours à l'avis de plusieurs experts ou à l'avis de la foule (myriadisation).

2.1 Les principes du crowdsourcing

Pour en comprendre les buts et les principes, allez observer le travail d'Alice Millour, doctorante en TAL à la Maison de la Recherche:

- `https://bisame.paris-sorbonne.fr/recettes/`
- `http://krik.paris-sorbonne.fr/`

Un travail d'annotation par myriadisation peut être réalisé dans tous les domaines, sur tous les types de support, par ex: `https://www.zooniverse.org/about`.

2.2 Avantages et inconvénients du crowdsourcing

Quels peuvent-ils être ?

3 Evaluation des annotations

Si plusieurs personnes ont annoté un échantillon de façon différente, comment déterminer lequel des annotateurs a raison ?

Établissez maintenant la **matrice de confusion** présentant les résultats de vos deux annotations (tableau 1). L'annotateur 1 représente les lignes, l'annotateur 2 représente les colonnes. A l'intersection de Nom et Nom le 3 signifie que dans 3 cas, les deux annotateurs ont été d'accord sur l'étiquette nom; Sur la même ligne, le "4" signifie que dans 4 cas, annotateur 1 a étiqueté "NOM" là où l'autre annotateur a étiqueté "Pronom".

	Nom	Adjectif	Verbe	Pronom	Déterminant	Adverbe	Autres	Total
Nom	3			4				
Adj.								
Ver.								
Pro.								
Dét.								
Adv.								
Aut.								
Total								

Table 1: Matrice de confusion pour l'étiquetage

Calculez pour chaque étiquette du jeu d'annotation le pourcentage d'accord réel (somme des cases grisées*100/nombre total d'échantillons).

3.1 Accord inter-annotateur

La seconde étape est d'évaluer des annotations. Les échantillons utilisables (pour une analyse statistique, un classifieur automatique, etc.) sont ceux pour lesquels un grand nombre d'annotateurs s'accordent.

Calculez le pourcentage d'accord réel pour chaque étiquette. Reportez ces résultats dans un tableau récapitulatif.

Pour vérifier que l'accord inter-annotateur est plus fort que le hasard, on peut mesurer le Kappa de Cohen :

$K = \frac{P_o - P_e}{1 - P_e}$

Où P_o est la proportion d'accord observé (somme des effectifs diagonaux divisée par la taille de l'échantillon). Où P_e est la proportion d'accord aléatoire (somme des produits des effectifs marginaux pour une même classe, divisée par le carré de la taille de l'échantillon)

Evaluez le taux d'accord entre vous et votre voisin au moyen du code ci-dessous (Notebook sur Moodle) où pour chaque annotateur on crée une liste correspondant aux étiquettes données dans l'ordre (cf listes `annotateur1` et `annotateur2` du code ci-après).

```
from nltk import agreement
annotateur1 = ["DET", "NOM", "ADJ", "ADJ", "ADV", "DET","ADJ", "ADV"]
annotateur2 = ["DET", "NOM", "ADJ", "NOM", "ADV", "NOM","ADV", "ADV"]
annotateurs = [annotateur1, annotateur2]
donnees = []
for i in range(len(annotateurs)):
  for j in range(len(annotateur1)):
    donnees.append([str(i), str(j), annotateurs[i][j]])
ratingtask = agreement.AnnotationTask(data=donnees)
print("kappa " +str(ratingtask.kappa()))
print("fleiss " + str(ratingtask.multi_kappa()))
print("alpha " +str(ratingtask.alpha()))
print("scotts " + str(ratingtask.pi()))
```

On peut désormais représenter cela sous forme d'une matrice de confusion :

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(annotateur1, annotateur2)
classes = sorted(list(set(annotateur1+annotateur2)))

fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(cm, annot=True, xticklabels=classes, yticklabels=classes,cm
plt.ylim([len(classes), 0])

plt.show()
```

3.2 Finalisation et évaluation des scores (à finir à la maison et à rendre en Binôme)

Vous rendrez un notebook Python qui compilera vos résultats. Tout d'abord, annotez chacun de votre côté l'exemple suivant : "Je me promène dans la campagne, au milieu des champs. J'arrive vers la magnifique ferme de mon voisin, et je remonte le chemin afin d'aller admirer son potager. Cependant, arrivé à la clôture, je me rends compte que la belle ferme le voile."

Calculez les scores P_o, P_e et K puis répondez aux questions suivantes:

- Représentez vos annotations avec une matrice de confusion
- Quelle étiquette présente le meilleur taux d'accord ?
- Les données sont-elles homogènes ? Y a-t-il des classes sous-/sur-représentées ?
- De quelle façon pourrions-nous améliorer les scores d'accord?

Vous rendrez ceci sur Moodle pour le 08/12/2019

Approfondissement

Consultez l'article suivant pour confronter vos idées avec celles d'une chercheuse du domaine :
https://www.liberation.fr/societe/2015/05/07/miracles-et-mirages-du-crowdsourcing_1297262