Plurital M1 - Semestre 2

2022/2023-LZST001-P Fouille de textes

Projet : Classification de résumés de thèses par discipline

Réalisé par :
Orane Dufour
Mikhail Biriuchinskii

Orane Dufour
Mikhail Biriuchinskii

Table des matières

Orane Dufour
Mikhail Biriuchinskii

Introduction

Choix du sujet : Nous avons choisi de classifier des résumés de thèses en fonction de leur discipline. Ce choix nous semblait intéressant car la tâche peut être plus ou moins difficile : nous supposons qu'il sera plutôt simple de différencier un résumé d'une thèse de mathématiques d'un résumé d'une thèse de littérature mais difficile en revanche de faire la différence entre philosophie et littérature par exemple.

Constitution du corpus : Tous les résumés extraits proviennent du site https://www.theses.fr/. Ce site est un moteur de recherche institutionnel dédiés aux thèses de doctorat françaises. Nous l'avons choisi car il comporte une API XML qui permet de récupérer les métadonnées sur les thèses. Pour récupérer les résumés des thèses, nous lançons d'abord une recherche sur le site : nous choisissons une discipline puis un intervalle de dates, que nous sélectionnons de sorte à avoir au moins 500 thèses. Nous récupérons les résultats de la recherche dans un fichier XML. Ce fichier contient les métadonnées sur les thèses :


```
−<arr name="ppn">
    <str>026403447</str>
    <str>035557060</str>
    <str>139542027</str>
  </arr>
  <str name="status">enCours</str>
  <date name="sujDatePremiereInscription">2002-12-01T23:59:59Z</date>
  <str name="titre">Interoperabilite et droits du marché.</str>
 </doc>
−<doc>
  <str name="accessible">non</str>
  <str name="auteur">Scheherazade Pinilla canadas</str>
  <str name="auteurPpn"/>
  <date name="dateInsert">2011-09-26T00:00:00Z</date>
  <date name="dateMaj">2012-04-02T09:56:51Z</date>
  <date name="dateSoutenance">2010-11-26T23:59:59Z</date>
 −<arr name="directeurThese">
    <str>Patrice Vermeren</str>
  </arr>
 −<arr name="directeurTheseNP">
    <str>Vermeren Patrice</str>
  </arr>
 −<arr name="directeurThesePpn">
    <str>028251873</str>
  </arr>
 −<str name="discipline">
    Philosophie (métaphysique, épistémologie, esthétique)
  </str>
  <str name="etabSoutenance">Paris 8</str>
  <str name="etabSoutenancePpn">026403552</str>
 −<arr name="etablissement">
    <str>Paris 8</str>
    <str/>
  </arr>
  <str name="id">14770</str>
  <str name="num">s11451</str>
 −<arr name="oaiSetSpec">
    <str>ddc:100</str>
  </arr>
```


Exemple de fichier xml

Avec un script python, nous récupérons le numéro de la thèse indiqué dans la balise <str name=”num>. Une fois le numéro récupéré, nous ajoutons dans un fichier l’URL de la thèse, c’est toujours le même modèle : https://www.theses.fr/NUMEROTHESE.xml.

Nous avons donc, pour chaque discipline souhaitée, un fichier texte contenant les URL des thèses dont nous voulons extraire les résumés. Ces URL mènent directement à la version XML de la page [1]. Le résumé se trouve dans la balise <dcterms:abstract xml:lang="fr">. Nous avons donc écrit un autre script python qui prend en entrée cette liste d’URL et qui pour chaque page extrait le résumé de la thèse. Chaque résumé fait l’objet d’un nouveau fichier texte qui a pour nom le numéro de la thèse en question. Nous avons dû indiquer quelques conditions pour la récupération du résumé : nous ne prenons que les résumés qui ont plus de 200 caractères (sans cette condition, nous nous retrouvions parfois avec des fichiers vides ou bien des fichiers contenant “Le résumé en français n'a pas été communiqué par l'auteur”).

Nous avons remarqué que certains résumés extraits étaient en anglais alors que la requête XPATH qui les récupère n’est censée prendre que ceux en français. Cela est dû à une erreur dans le fichier XML, les résumés français et anglais ont été inversés. Comme c’était un cas plutôt rare, nous avons corrigé manuellement l’erreur en changeant les résumés des pages concernées.

Nous avons récolté au total 2000 documents pour 10 catégories : Arts, Mathématiques, Physique, Littérature, Sciences Politiques, Sciences du Langage, Philosophie, Économie, Histoire et Biologie. Nous avons souhaité mener nos expérimentations tout d’abord sur l’ensemble du corpus et des catégories. Cependant nous avons choisi de constituer ces 10 catégories aussi pour pouvoir changer à loisir la configuration du corpus et étudier en quoi la présence ou non d’une catégorie peut influer sur la classification.

Pré-traitement sur le corpus

Nous avons commencé par lancer le script *vectorisation.py* sur le corpus afin de voir, en essayant directement des modèles sur weka, quels mots pouvaient poser problème lors de l’analyse des modèles utilisés. Nous avons lancé plusieurs classifieurs et l’un d’entre eux, *JRip*, dont les résultats sont facilement interprétables par un humain, nous a permis de relever certains problèmes avec notre corpus. Tout d’abord, dans beaucoup de résumés de thèses, le nom de la discipline est mentionné. Pour toutes nos catégories, *JRip* a donc généré une règle indiquant que, si le nom de la catégorie est présent dans le document, alors c’est qu’il appartient à la catégorie en question. Nous avons estimé que cela rendait la tâche trop facile et nous avons donc choisi d’enlever ces mots du corpus en les mettant dans une stop-list que nous appelons en lançant le script vectorisation.py. Le mot “thèse” posait problème également, présent dans la plupart des résumés, nous l’avons retrouvé dans quelques règles de JRip alors que le mot n’est pas un attribut intéressant pour identifier nos catégories, nous l’avons ajouté à la stop-list. Enfin, nous avons ajouté dans la stop-list la plupart des mots vides que nous avons rencontrés en examinant ces premiers résultats.

Nous avons également essayé de modifier le script vectorisation.py afin d’y intégrer un étiqueteur morphosyntaxique qui nous permettrait d’enlever les POS non désirés. Malheureusement, notre tentative n’a pas été fructueuse : le script générait toujours le fichier .arff cependant avec la représentation booléenne des attributs, cela créait un fichier que weka ne pouvait pas lire.

[1] Exemple : https://www.theses.fr/2021COMP2672.xml

Expérimentations - Corpus 10 catégories

1 - Arbres de décision

Les classifieurs utilisant cette technique construisent des arbres de décisions dans lesquels chaque nœud représente un test sur un attribut et chaque branche de ce nœud correspond à une des valeurs possibles pour cet attribut.

Ces classifieurs ont l'avantage d'être facilement interprétables par des humains, nous pouvons comprendre ce que le modèle fait et les solutions qu'il met en place pour identifier les classes.

```
Number of Leaves  :  287

Size of the tree :  573


Time taken to build model: 287.61 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         891               44.55   %
Incorrectly Classified Instances      1109               55.45   %
Kappa statistic                          0.3839
Mean absolute error                      0.1142
Root mean squared error                  0.2985
Relative absolute error                 63.4564 %
Root relative squared error             99.5085 %
Total Number of Instances             2000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.520    0.050    0.536      0.520    0.528      0.476    0.768     0.402     Arts
                 0.275    0.068    0.309      0.275    0.291      0.218    0.788     0.307     Biologie
                 0.325    0.056    0.392      0.325    0.355      0.292    0.717     0.285     Histoire
                 0.510    0.052    0.523      0.510    0.516      0.464    0.764     0.339     Litterature
                 0.655    0.049    0.598      0.655    0.625      0.582    0.820     0.496     Maths
                 0.425    0.053    0.470      0.425    0.446      0.389    0.749     0.334     Philosophie
                 0.565    0.048    0.565      0.565    0.565      0.517    0.785     0.422     Physique
                 0.325    0.068    0.348      0.325    0.336      0.265    0.687     0.207     Sciences_du_langage
                 0.525    0.046    0.561      0.525    0.543      0.494    0.801     0.408     Sciences_économiques
                 0.330    0.126    0.225      0.330    0.268      0.173    0.638     0.151     Sciences_politiques
Weighted Avg.    0.446    0.062    0.453      0.446    0.447      0.387    0.752     0.335

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 104   4  21  28   2   7   6   3   4  21 |   a = Arts
   7  55   6   1   4   5   6  81   8  27 |   b = Biologie
  20   8  65  15   1  20   7   5   9  50 |   c = Histoire
  30   1  12 102   3  34   2   8   2   6 |   d = Litterature
   3   5   4   1 131   0  33   1  12  10 |   e = Maths
   9   6  14  38   6  85   6   5   6  25 |   f = Philosophie
   3   9   5   1  41   2 113   1  16   9 |   g = Physique
   6  59   3   3   8   3   8  65   3  42 |   h = Sciences_du_langage
   2   6   9   1  19  10  11   0 105  37 |   i = Sciences_économiques
  10  25  27   5   4  15   8  18  22  66 |   j = Sciences_politiques
```

Classifier output

Nous avons essayé le classifieur J48 sur notre corpus, il obtient une F-mesure de 0.447. Après analyse de l'arbre de décision, nous constatons que beaucoup trop d'attributs non discriminants ont été pris en compte malgré les précautions prises lors de la phase de prétraitement du texte. De plus, étant donné la taille de notre corpus et notre grand nombre d'attributs, la modélisation de l'arbre de décision est difficilement lisible. Le plus petit rappel se trouve sur la classe Biologie avec seulement 0.275. Seuls 55 documents sur 200 ont été classés dans la bonne catégorie et 81 documents ont été classés dans la catégorie Sciences_du_langage. Nous retrouvons la difficulté avec ces deux classes rencontrée précédemment avec Naive Bayes.

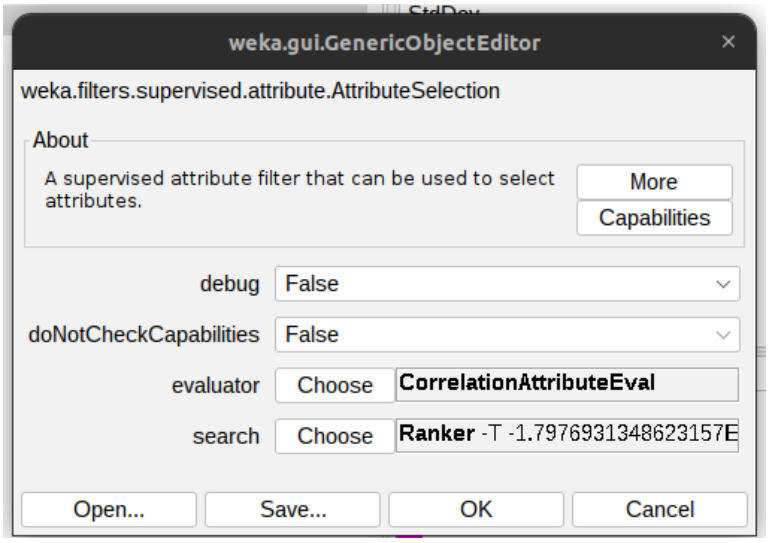
Nous avons lancé cette première analyse en utilisant la cross-validation, la génération de l'arbre a été rapide cependant les phases d'apprentissage/test ont duré une vingtaine de minutes. Nous avons donc essayé de relancer le classifieur cette fois-ci en divisant le corpus pour l'apprentissage et pour le test (60/40). Nous avons obtenu des résultats similaires à la précédente analyse avec une f-mesure à 0.445 et 45% des instances bien classées. Cependant, la phase de test n'a duré que 0.15 seconde. Le corpus n'est donc pas adapté pour cette méthode de classification, le seul moyen d'améliorer nos résultats serait d'opérer un prétraitement plus fin sur le corpus.

2 - Bayes Classifiers

a) Naive Bayes

Ce classifieur va prédire la probabilité d'une classe sachant qu'un mot figure dans le document, en émettant une hypothèse d'indépendance entre les mots. Les attributs booléens sont donc plus adaptés à ce type de classifieur.

Nous avons rencontré beaucoup de problèmes avec ce classifieur. En effet, notre trop grand nombre d'attributs (environ 35 000) rendait le temps de calcul interminable. Nous avons donc tenté de réduire ce nombre, tout d'abord en essayant d'ajouter des mots dans la liste de mots vides puis en nous intéressant à l'onglet preprocess de weka. En effet, dans cet onglet, il est possible de lancer des calculs afin de sélectionner nos attributs. Nous avons choisi un algorithme qui calcule la corrélation entre l'attribut et la classe puis qui renvoie une liste des attributs, des plus importants au moins importants.



Paramètres pour la sélection d'attributs

Nous avons effectué plusieurs essais, tout d'abord avec 10 000, puis 5 000 attributs, mais le classifieur ne fonctionnait toujours pas. Nous avons donc drastiquement diminué le nombre d'attributs pour n'en garder que 500, et nous sommes enfin parvenus à des résultats :

```
=== Summary ===

Correctly Classified Instances        1544               77.2    %
Incorrectly Classified Instances       456               22.8    %
Kappa statistic                          0.7467
Mean absolute error                      0.0466
Root mean squared error                  0.2011
Relative absolute error                 25.9119 %
Root relative squared error             67.0451 %
Total Number of Instances             2000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.760    0.020    0.809      0.760    0.784      0.761    0.973     0.863     Arts
                 0.440    0.011    0.815      0.440    0.571      0.569    0.906     0.678     Biologie
                 0.745    0.028    0.745      0.745    0.745      0.717    0.969     0.836     Histoire
                 0.820    0.033    0.732      0.820    0.774      0.748    0.978     0.793     Litterature
                 0.895    0.011    0.904      0.895    0.899      0.888    0.993     0.949     Maths
                 0.725    0.024    0.771      0.725    0.747      0.721    0.957     0.801     Philosophie
                 0.890    0.013    0.886      0.890    0.888      0.875    0.992     0.949     Physique
                 0.695    0.052    0.597      0.695    0.642      0.601    0.912     0.585     Sciences_du_langage
                 0.900    0.008    0.923      0.900    0.911      0.902    0.989     0.954     Sciences_économiques
                 0.850    0.053    0.642      0.850    0.731      0.705    0.966     0.724     Sciences_politiques
Weighted Avg.    0.772    0.025    0.782      0.772    0.769      0.749    0.964     0.813

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 152   1  19  10   0   8   0   4   1   5 |   a = Arts
   2  88   3   4   0   1   4  79   2  17 |   b = Biologie
  13   1 149   9   0   6   0   1   2  19 |   c = Histoire
  10   0   4 164   0  18   0   3   0   1 |   d = Litterature
   0   0   0   0 179   0  18   1   2   0 |   e = Maths
   5   1   5  31   1 145   0   4   0   8 |   f = Philosophie
   0   4   2   0  14   1 178   0   0   1 |   g = Physique
   3  13   4   4   0   2   1 139   1  33 |   h = Sciences_du_langage
   0   0   3   0   4   2   0   0 180  11 |   i = Sciences_économiques
   3   0  11   2   0   5   0   2   7 170 |   j = Sciences_politiques
```

Classifier output

Ce classifieur obtient donc une f-mesure moyenne de 0.769, il est parvenu à classer correctement 77.2% des mots du corpus. Les catégories Sciences du langage et Biologie sont les moins bien reconnues, on note par ailleurs qu'elles sont souvent confondues l'une avec l'autre par le classifieur.

Nous avons par la suite voulu comparer les résultats de Naives Bayes avec Multinomial Naive Bayes. Nous supposions en effet que le classifieur multinomial serait plus approprié à notre corpus étant donné qu'il permet de prendre en compte le nombre d'occurrences des mots plutôt que seulement l'apparition ou non du mot.

b) Multinomial Naive Bayes

Afin de pouvoir comparer les deux classifieurs, nous avons sélectionné les mêmes paramètres pour l'entraînement et les attributs. Ainsi, nous avons sélectionné, au moyen du même algorithme que précédemment, les 500 attributs les plus corrélés à nos catégories du corpus. Nous avons également pris garde à lancer le classifieur selon les mêmes modalités d'entraînement que le premier (training set). Enfin, le fichier arff des comptes que nous avons généré afin de lancer ce classifieur a été créé avec la même liste de mots vides que le fichier avec les attributs booléens.

```
Time taken to build model: 0.04 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.14 seconds

=== Summary ===

Correctly Classified Instances        1564               78.2    %
Incorrectly Classified Instances       436               21.8    %
Kappa statistic                          0.7578
Mean absolute error                      0.045
Root mean squared error                  0.1943
Relative absolute error                 25.002  %
Root relative squared error             64.7702 %
Total Number of Instances             2000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.775    0.020    0.812      0.775    0.793      0.771    0.976     0.878     Arts
                 0.420    0.008    0.857      0.420    0.564      0.573    0.910     0.684     Biologie
                 0.830    0.028    0.765      0.830    0.796      0.773    0.977     0.869     Histoire
                 0.800    0.026    0.777      0.800    0.788      0.764    0.982     0.822     Litterature
                 0.910    0.012    0.897      0.910    0.903      0.892    0.994     0.960     Maths
                 0.740    0.028    0.744      0.740    0.742      0.713    0.961     0.803     Philosophie
                 0.900    0.013    0.882      0.900    0.891      0.879    0.993     0.955     Physique
                 0.690    0.047    0.619      0.690    0.652      0.613    0.921     0.595     Sciences_du_langage
                 0.885    0.007    0.937      0.885    0.910      0.901    0.994     0.968     Sciences_économiques
                 0.870    0.053    0.644      0.870    0.740      0.717    0.966     0.709     Sciences_politiques
Weighted Avg.    0.782    0.024    0.793      0.782    0.778      0.760    0.967     0.824

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 155   1  21   7   0   8   1   1   0   6 |   a = Arts
   1  84   3   4   0   3   6  77   1  21 |   b = Biologie
  12   0 166   6   0   2   0   1   2  11 |   c = Histoire
   9   1   6 160   0  22   0   2   0   0 |   d = Litterature
   0   0   0   0 182   0  15   1   2   0 |   e = Maths
   8   1   5  24   2 148   1   1   2   8 |   f = Philosophie
   0   2   1   0  15   1 180   0   1   0 |   g = Physique
   3   9   4   4   0   4   1 138   0  37 |   h = Sciences_du_langage
   1   0   1   0   4   3   0   1 177  13 |   i = Sciences_économiques
   2   0  10   1   0   8   0   1   4 174 |   j = Sciences_politiques
```

Classifier output

Ce classifieur obtient une f-mesure moyenne de 0.778, très légèrement au-dessus du premier classifieur. La catégorie qui pose le plus de problèmes est là encore Biologie avec une f-mesure à 0.420, suivie de la classe Sciences du langage avec une f-mesure à 0.652. Nous pouvions nous attendre à la difficulté de classer les résumés de thèses ayant pour discipline les sciences du langage car c'est un domaine qui emprunte à beaucoup d'autres disciplines. Le détail des probabilités pour chaque attribut permet par ailleurs de rendre compte du caractère interdisciplinaire des sciences du langage : on voit en effet que pour certains attributs, la probabilité que cet attribut appartienne à un document étiqueté science du langage est la même que pour une autre étiquette.

La distinction entre littérature et philosophie pose aussi problème mais un peu moins que pour naive bayes : seuls 24 résumés de thèses de philosophie ont été classés dans la catégorie littérature et seulement 22 pour l'inverse. Nous pensions que cette erreur aurait une bien plus grande part.

Enfin, les meilleurs résultats sont obtenus pour les catégories mathématiques, physique et sciences économiques. Cela est sûrement dû à l'utilisation de symboles qui leur sont très spécifiques dans le cadre de formules.

Testés en fonction de paramètres similaires, Multinomial Naive Bayes fait donc mieux que Naive Bayes cependant la différence est minime ce qui nous a semblé surprenant. Peut-être notre nombre d'attributs, qui reste conséquent, permet-il à Naive Bayes d'égaler le deuxième classifieur alors même que celui-ci semble plus indiqué pour notre corpus.

3 - Classifieurs Linéaires

La classification linéaire consiste à trouver, dans un hyperplan, l'espace qui sépare les attributs en plusieurs espaces (nombre de catégories) qui contiennent tous les exemples d'entraînement d'une seule classe. Cependant, avec des données réelles, il est rare qu'un tel espace existe, on utilise donc des SVM. Pour pallier ce problème, il existe deux méthodes :

- Marge douce : modifier la définition de marge pour tenir compte des exemples mal classés ;
- kernel trick : Transformer le problème en un problème équivalent mais linéairement séparable avec l'astuce du noyau.

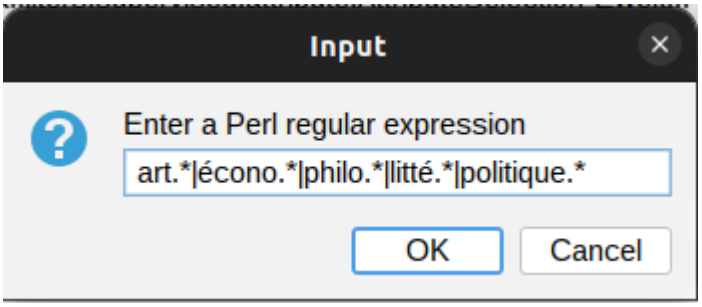
Nous avons testé sur le corpus le classifieur SMO. Au premier essai, nous avons utilisé l'intégralité du corpus, avec 200 documents par catégorie et un training set. Cependant, nous avons rencontré le même problème qu'avec Naive Bayes concernant le temps de calcul. Nous avons donc sélectionné, selon la même méthode que précédemment, 500 attributs.

```
=== Summary ===

Correctly Classified Instances        1810               90.5    %
Incorrectly Classified Instances       190                9.5    %
Kappa statistic                          0.8944
Mean absolute error                      0.1605
Root mean squared error                  0.2725
Relative absolute error                 89.1543 %
Root relative squared error             90.8175 %
Total Number of Instances             2000

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1.000    0.001    0.990      1.000    0.995      0.994    0.999     0.990     Arts
               0.435    0.002    0.956      0.435    0.598      0.623    0.952     0.663     Biologie
               0.990    0.000    1.000      0.990    0.995      0.994    1.000     0.996     Histoire
               0.975    0.014    0.886      0.975    0.929      0.922    0.992     0.875     Litterature
               1.000    0.001    0.995      1.000    0.998      0.997    1.000     0.995     Maths
               0.875    0.003    0.972      0.875    0.921      0.914    0.992     0.914     Philosophie
               0.995    0.000    1.000      0.995    0.997      0.997    1.000     0.998     Physique
               0.795    0.052    0.631      0.795    0.704      0.672    0.960     0.593     Sciences_du_langage
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Sciences_économiques
               0.985    0.033    0.767      0.985    0.862      0.853    0.982     0.762     Sciences_politiques
Weighted Avg.  0.905    0.011    0.920      0.905    0.900      0.897    0.988     0.879

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 200   0   0   0   0   0   0   0   0   0 |   a = Arts
   0  87   0   0   0   0   0  91   0  22 |   b = Biologie
   2   0 198   0   0   0   0   0   0   0 |   c = Histoire
   0   0   0 195   0   5   0   0   0   0 |   d = Litterature
   0   0   0   0 200   0   0   0   0   0 |   e = Maths
   0   0   0  25   0 175   0   0   0   0 |   f = Philosophie
   0   0   0   0   1   0 199   0   0   0 |   g = Physique
   0   3   0   0   0   0   0 159   0  38 |   h = Sciences_du_langage
   0   0   0   0   0   0   0   0 200   0 |   i = Sciences_économiques
   0   1   0   0   0   0   0   2   0 197 |   j = Sciences_politiques
```

Classifier output

Nous obtenons une f-mesure moyenne de 0.9 avec 90.5 documents classés dans la bonne catégorie. Là encore, la catégorie biologie est celle qui pose le plus de problème avec une f-mesure nettement inférieure aux autres catégories et presque la moitié de ses documents classés par erreur dans la catégorie sciences du langage. Malheureusement, les données que donnent weka sur le déroulement du processus de calcul résistent à notre analyse. Nous avons cependant remarqué que nous avions omis de retirer certains mots qui sont directement dérivés du nom de la catégorie : "artistique" pour "arts" par exemple. Cela pourrait expliquer les très bons résultats de ce classifieur.

Nous avons donc décidé de recommencer l'expérience, cette fois-ci en enlevant ces mots directement grâce à l'onglet preprocess de weka et une expression régulière :



expression régulière pour supprimer certains attributs

Sur la même méthode d'entraînement et avec un total de 487 attributs, nous obtenons une f-mesure à peine inférieure de 0.899, pour 90.documents bien classés. Surpris par ces scores si élevés, nous avons tenté de réduire le nombre d'attributs à 100 seulement.

```
Time taken to build model: 0.69 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances        1393               69.65   %
Incorrectly Classified Instances       607               30.35   %
Kappa statistic                          0.6628
Mean absolute error                      0.1625
Root mean squared error                  0.2761
Relative absolute error                 90.2556 %
Root relative squared error             92.0435 %
Total Number of Instances             2000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.590    0.039    0.628      0.590    0.608      0.567    0.903     0.531     Arts
                 0.440    0.013    0.786      0.440    0.564      0.557    0.916     0.589     Biologie
                 0.740    0.058    0.585      0.740    0.653      0.615    0.942     0.548     Histoire
                 0.650    0.030    0.707      0.650    0.677      0.644    0.937     0.591     Litterature
                 0.900    0.018    0.849      0.900    0.874      0.860    0.988     0.839     Maths
                 0.620    0.049    0.585      0.620    0.602      0.557    0.918     0.493     Philosophie
                 0.825    0.012    0.882      0.825    0.853      0.838    0.982     0.816     Physique
                 0.660    0.051    0.589      0.660    0.623      0.579    0.922     0.520     Sciences_du_langage
                 0.850    0.013    0.881      0.850    0.865      0.851    0.972     0.822     Sciences_économiques
                 0.690    0.054    0.587      0.690    0.634      0.593    0.927     0.508     Sciences_politiques
Weighted Avg.    0.697    0.034    0.708      0.697    0.695      0.666    0.941     0.626

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 118   1  29  17   0  21   0   5   2   7 |   a = Arts
   3  88   4   2   0   6   3  75   2  17 |   b = Biologie
  13   2 148   5   1   5   0   0   3  23 |   c = Histoire
  20   1  14 130   1  26   0   6   0   2 |   d = Litterature
   0   0   0   0 180   1  18   0   0   1 |   e = Maths
  19   3  13  28   2 124   0   1   0  10 |   f = Philosophie
   3   2   3   0  20   2 165   0   5   0 |   g = Physique
   5  13   8   2   1   6   1 132   2  30 |   h = Sciences_du_langage
   1   1   9   0   6   5   0   1 170   7 |   i = Sciences_économiques
   6   1  25   0   1  16   0   4   9 138 |   j = Sciences_politiques
```

Classifier output

La f-mesure pour 100 attributs descend donc à 0,685, pour 69.65%. C'est un score qui avoisine ceux obtenus avec les classifieurs bayésiens avec pourtant 5 fois moins d'attributs. Ainsi, la méthode de classification linéaire se révèle être incontestablement la meilleure pour notre corpus.

Conclusion pour le corpus 10 catégories

Nous notons que la taille du corpus a de prime abord rendu les opérations difficiles : sans une réduction considérable du nombre d'attributs, beaucoup de classifieurs étaient impossibles à utiliser. L'obligation de réduire le nombre d'attributs nous a cependant permis de nous familiariser avec l'onglet preprocess de weka qui permet de les sélectionner grâce à divers algorithmes : nous n'avions donc pas besoin de le faire au moment du prétraitement du texte. De plus, ces algorithmes de sélection sont eux aussi entraînés sur le corpus afin de sélectionner les attributs les plus significatifs ce qui permet d'obtenir une autre perspective que lorsque nous les choisissons nous-mêmes. Malheureusement, là encore la taille du corpus ne nous a permis d'en essayer seulement quelques-uns.

Enfin, quant aux résultats, nous sommes agréablement surpris d'obtenir un score aussi élevé avec notre meilleur essai. Nous pensions qu'avec 10 catégories, dont certaines très proches les unes des autres, les classifieurs ne parviendraient pas à distinguer correctement les documents.

Expérimentations - Corpus 4 catégories

Pendant les étapes précédentes de notre recherche, nous avons relevé les 4 catégories les plus difficiles à prédire : **Biologie**, **Histoire**, **Sciences du langage** et **Sciences politiques**, d'après les scores d'évaluations obtenus pendant l'expérimentation avec Multinomial Naive Bayes et Arbres de décision.

Nous avons donc décidé de nous intéresser à ces catégories. En excluant les 6 catégories qui ont donné les meilleurs résultats, nous pouvons nous concentrer davantage sur les 4 catégories moins performantes et explorer différentes techniques de classification pour les améliorer. Cela peut conduire à une performance globale meilleure que celle obtenue en incluant toutes les catégories. Cette expérience était également l'occasion pour nous de vérifier une des hypothèses que nous avions : plus il y a de catégories, moins nos résultats seront bons.

1. **ZeroR**

Le classifieur ZeroR est un modèle de classification très simple, qui est utilisé comme modèle de base pour la comparaison avec des modèles plus complexes. L'essence de ZeroR est qu'il n'utilise qu'un seul attribut (class majoritaire) pour prédire l'étiquette de la classe pour toutes les instances de données :

```
=== Summary ===

Correctly Classified Instances          73               22.8125 %
Incorrectly Classified Instances       247               77.1875 %
Kappa statistic                          0
Mean absolute error                      0.3754
Root mean squared error                  0.4337
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances              320

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.238     Biologie
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.275     Histoire
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.259     Sciences_du_langage
                 1.000    1.000    0.228      1.000    0.372      ?        0.500     0.228     Sciences_politiques
Weighted Avg.    0.228    0.228    ?          0.228    ?          ?        0.500     0.251

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
  0  0  0 76 |  a = Biologie
  0  0  0 88 |  b = Histoire
  0  0  0 83 |  c = Sciences_du_langage
  0  0  0 73 |  d = Sciences_politiques
```

Classifier output

Lorsque le classifieur ZeroR a été appliqué à notre ensemble de données de test, il n'a classé correctement que 73 instances sur un total de 320 (22,8 %). Il a également mal classé 247 instances (77,2 %). Cela montre que le classifieur ZeroR n'est pas efficace pour résoudre notre problème de classification des données. En effet, notre corpus étant équilibré (200 documents par classe), le classifieur ne parvient peut pas trouver de classe majoritaire.

2. Classification k-plus proches voisins

Pour le classifieur IBK lazy, il s'agit d'un algorithme d'apprentissage automatique lazy, ce qui signifie qu'il ne construit pas un modèle nécessitant une formation sur toutes les données disponibles, mais qu'il retarde le traitement des données jusqu'à ce qu'il reçoive une demande de classification d'une nouvelle observation.

La distance euclidienne est utilisée comme métrique pour la distance entre les observations, de sorte que l'algorithme calcule d'abord la distance entre la nouvelle observation et toutes les observations de l'ensemble de données d'apprentissage. Il choisit ensuite les k voisins les plus proches et classe la nouvelle observation en fonction de la classe qui apparaît le plus souvent parmi ses k voisins.

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.995    0.232    0.589      0.995    0.740      0.669    0.941     0.793     Biologie
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Histoire
                 0.425    0.030    0.825      0.425    0.561      0.511    0.924     0.741     Sciences_du_langage
                 0.795    0.000    1.000      0.795    0.886      0.863    0.991     0.963     Sciences_politiques
Weighted Avg.    0.804    0.065    0.854      0.804    0.797      0.761    0.964     0.874

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 199   0   1   0 |   a = Biologie
   0 200   0   0 |   b = Histoire
 115   0  85   0 |   c = Sciences_du_langage
  24   0  17 159 |   d = Sciences_politiques
```

Classifier output

Nous avons utilisé ce classifieur avec test option “use training set”. Dans l'ensemble, les résultats montrent que le modèle IBK a classé correctement 80,4 % des échantillons, ce qui est un bon résultat pour de nombreuses tâches de classification. Cependant, 19,6 % des échantillons ont également été mal classés. Par exemple, on peut observer que 115 échantillons de la classe Sciences_du_langage ont été incorrectement classés dans la classe Biologie et que 24 échantillons de la classe Sciences_du_langage ont été incorrectement classés dans la classe Biologie.

3. Bayes Classifiers

a) Naive Bayes

Nous voulions à nouveau comparer Naïve Bayes et Naïve Bayes Multinomial pour ce corpus. Nous avons rencontré les mêmes problèmes qu’avec le corpus 10 catégories, trop d’attributs rendait le calcul impossible. Nous avons à nouveau sélectionné 500 attributs, toujours avec l’onglet préprocess de weka et la sélection par corrélation d’attributs. Nous avons décidé de ne garder que les 500 premiers attributs.

```
Time taken to build model: 0.07 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.21 seconds

=== Summary ===

Correctly Classified Instances         610               76.25   %
Incorrectly Classified Instances       190               23.75   %
Kappa statistic                          0.6833
Mean absolute error                      0.1188
Root mean squared error                  0.3315
Relative absolute error                 31.6927 %
Root relative squared error             76.5473 %
Total Number of Instances              800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.445    0.015    0.908      0.445    0.597      0.568    0.872     0.764     Biologie
                 0.925    0.033    0.902      0.925    0.914      0.884    0.990     0.974     Histoire
                 0.750    0.148    0.628      0.750    0.683      0.569    0.863     0.651     Sciences_du_langage
                 0.930    0.120    0.721      0.930    0.812      0.750    0.950     0.807     Sciences_politiques
Weighted Avg.    0.763    0.079    0.790      0.763    0.752      0.693    0.919     0.799

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
  89   4  86  21 |   a = Biologie
   1 185   1  13 |   b = Histoire
   7   5 150  38 |   c = Sciences_du_langage
   1  11   2 186 |   d = Sciences_politiques
```

Classifier output

Le classifieur obtient un score similaire au score obtenu avec les 10 catégories, à nouveau la plus grande confusion se fait entre biologie et science du langage. Les scores obtenus sont légèrement moins bons que sur le corpus 10 catégories.

b) Multinomial Naïve Bayes

Nous avons cette fois-ci utilisé Naïve Bayes Multinomial sans réduire le nombre d'attributs car ce corpus nous le permettait.

```
Time taken to build model: 0.27 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 3.57 seconds

=== Summary ===

Correctly Classified Instances         640               80      %
Incorrectly Classified Instances       160               20      %
Kappa statistic                          0.7333
Mean absolute error                      0.1
Root mean squared error                  0.3156
Relative absolute error                 26.661  %
Root relative squared error             72.8805 %
Total Number of Instances              800

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.420    0.000    1.000      0.420    0.592      0.593    0.913     0.813     Biologie
               0.985    0.000    1.000      0.985    0.992      0.990    1.000     0.999     Histoire
               0.795    0.153    0.633      0.795    0.705      0.599    0.908     0.734     Sciences_du_langage
               1.000    0.113    0.746      1.000    0.855      0.813    0.974     0.881     Sciences_politiques
Weighted Avg.  0.800    0.067    0.845      0.800    0.786      0.749    0.949     0.857

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
  84   0  92  24 |   a = Biologie
   0 197   0   3 |   b = Histoire
   0   0 159  41 |   c = Sciences_du_langage
   0   0   0 200 |   d = Sciences_politiques
```

Classifier output

On peut remarquer que la classificateur Multinomial a pu augmenter les scores de F-mesure pour la catégorie Sciences_du_langage (f-mesure à 0.705), la catégorie étant le moins correctement évalué (f-mesure à 0.432) dans l'expérience avec 10 catégories, ce qui est le cas pour toutes les catégories concernées. Lorsque seules 4 catégories ont été choisies, la tâche de classification est devenue plus facile, permettant aux modèles de faire une distinction plus précise entre les catégories et d'augmenter le score f pour chacune de ces catégories. Dans cette expérimentation, comme précédemment, la classificateur Multinomial a obtenu de meilleurs résultats que Naïve Bayes standard.

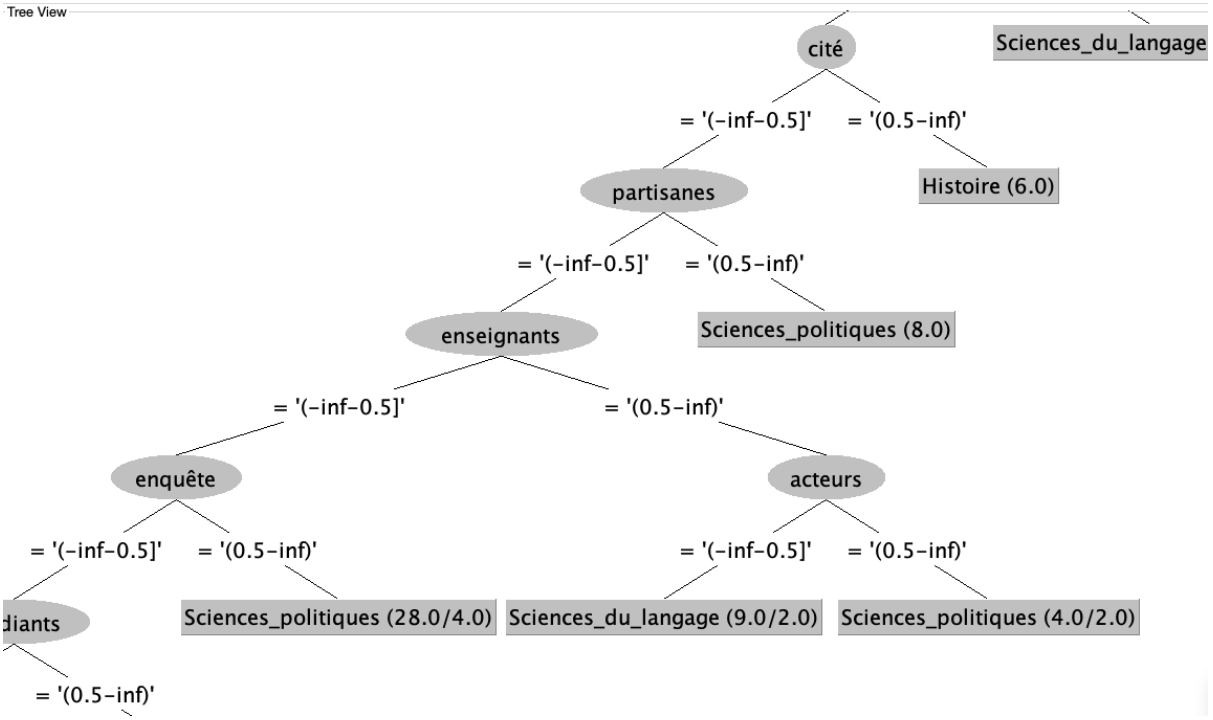
4. J48 Arbres de décision

Le modèle que nous avons choisi avec le paramètre "Use training model" avait 121 nœuds et des scores faibles, ce qui pouvait indiquer un sur-entraînement. Pour éviter cela, il a été décidé d'utiliser le mode de validation croisée. Le temps de construction du modèle a été de 39,47 secondes et la phase de validation croisée a duré environ 7 minutes.

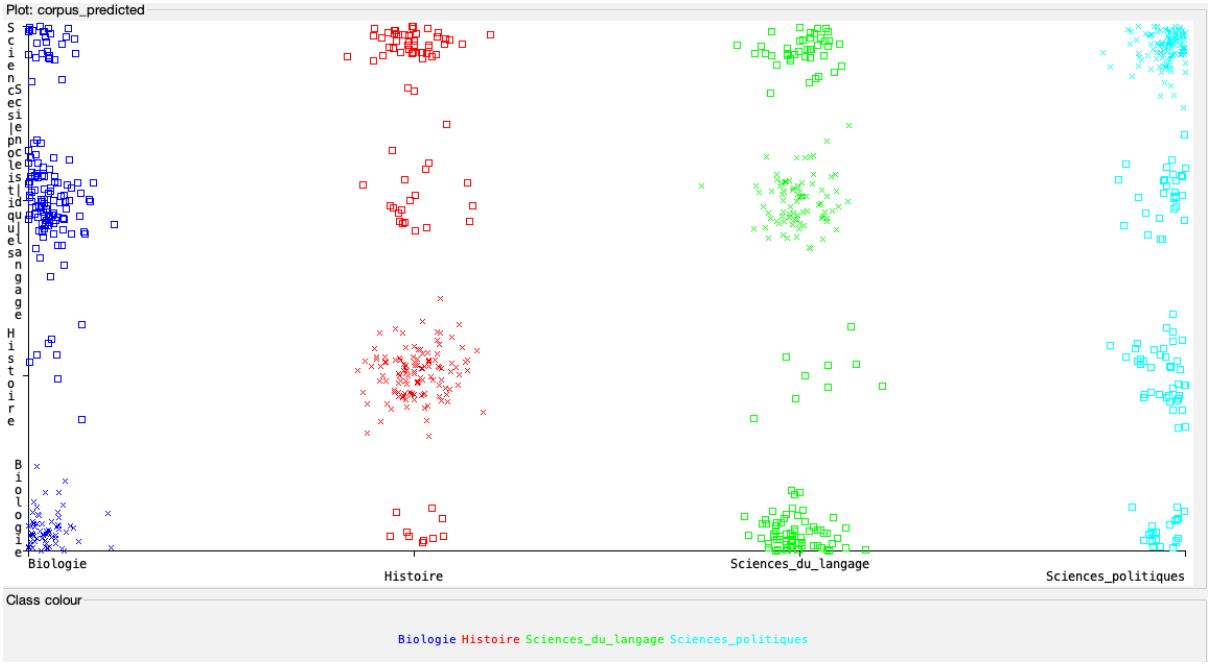
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         379               47.375  %
Incorrectly Classified Instances       421               52.625  %
Kappa statistic                          0.2983
Mean absolute error                      0.27
Root mean squared error                  0.4393
Relative absolute error                 72.0025 %
Root relative squared error            101.4413 %
Total Number of Instances              800

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.330    0.165    0.400      0.330    0.362      0.177    0.697     0.425     Biologie
               0.605    0.095    0.680      0.605    0.640      0.531    0.817     0.589     Histoire
               0.415    0.247    0.359      0.415    0.385      0.161    0.599     0.303     Sciences_du_langage
               0.545    0.195    0.482      0.545    0.512      0.337    0.701     0.458     Sciences_politiques
Weighted Avg.  0.474    0.175    0.480      0.474    0.475      0.301    0.703     0.444

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
  66   8  99  27 |   a = Biologie
  10 121  20  49 |   b = Histoire
  67   9  83  41 |   c = Sciences_du_langage
  22  40  29 109 |   d = Sciences_politiques
```

Classifier output

Malheureusement, la réduction des catégories à 4 n'a pas amélioré les performances. Comme dans l'expérience précédente, le modèle semble avoir pris en compte trop d'attributs non discriminants :



l'exemple de l'arbre de décision



Visualisation des erreurs de classification (erreurs = carrés)

Le Filtered Classifier [2] de Weka permet d'appliquer un pré-filtre à un ensemble de données avant d'appliquer le classificateur. Les options du filtre spécifient que tous les attributs de la première à la dernière colonne doivent être discrétisés avec une précision de 6 chiffres après la virgule. L'option "-S 1" spécifie la méthode de sélection des sous-ensembles pour le classificateur filtré. Dans ce cas, il s'agit d'une sélection aléatoire.

Enfin, les options "-C 0.25" et "-M 2" spécifient les paramètres de l'arbre de décision, où "-C" contrôle la complexité de l'arbre et "-M" spécifie le nombre minimal d'instances requises pour chaque feuille de l'arbre. À notre grande surprise, l'utilisation de Filtered Classifier n'a pu améliorer le taux d'exemples correctement définis que de 10 %.

5. Classifieur Linéaire

Comme dans l'expérience précédente, nous avons décidé de limiter le nombre d'attributs à 500 pour l'utilisation du classifieur SMO :

[2] weka.classifiers.meta.FilteredClassifier F "weka.filters.supervised.attribute.Discretize -R first-last -precision 6" -S 1 -W weka.classifiers.trees.J48 - -C 0.25 -M 2

```
Time taken to build model: 0.22 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances         643               80.375  %
Incorrectly Classified Instances       157               19.625  %
Kappa statistic                          0.7383
Mean absolute error                      0.269
Root mean squared error                  0.3405
Relative absolute error                 71.7222 %
Root relative squared error             78.6283 %
Total Number of Instances              800

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.430    0.005    0.966      0.430    0.595      0.585    0.891     0.705     Biologie
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     0.999     Histoire
                 0.800    0.157    0.630      0.800    0.705      0.598    0.862     0.576     Sciences_du_langage
                 0.985    0.100    0.767      0.985    0.862      0.821    0.947     0.762     Sciences_politiques
Weighted Avg.    0.804    0.065    0.841      0.804    0.791      0.751    0.925     0.760

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
  86   0  92  22 |   a = Biologie
   0 200   0   0 |   b = Histoire
   2   0 160  38 |   c = Sciences_du_langage
   1   0   2 197 |   d = Sciences_politiques
```

Classifier output

Les résultats montrent que le modèle a réussi à classer correctement 643 instances sur les 800. Cela représente un taux de classification correct de 80,375%. La statistique kappa, qui mesure l'accord entre les prédictions du modèle et les classes réelles, est de 0,7383, ce qui indique une bonne performance du modèle.

Cependant, si l'on commence à comparer les résultats, ce qui est le but de cette partie de l'étude, on constate que la réduction du nombre de catégories n'a pas augmenté l'efficacité du modèle. En outre, le classifieur SMO affiche une performance similaire à celle du Multinomial, soit environ 80 %, tandis que le SMO pour 10 catégories affiche une performance de 90 %. Parallèlement, on observe des tendances similaires : les catégories Biologie et Sciences_du_langage restent parmi les moins productives, affichant une f-mesure inférieure à toutes les autres classes.

Conclusion pour le corpus 4 catégories

En conclusion, la taille de données réduite n'a pas augmenté les performances des classifieurs. Comme nous l'avons vu dans la comparaison des corpus de 10 et 4 catégories, les mêmes catégories qui posent problème aux classificateurs sont retrouvées : biologie et Sciences_du_langage, alors qu'il y a une augmentation des scores de F-mesure pour la catégorie Sciences_du_langage lors de l'utilisation de Naive Bayes Multinomial pour 4 catégories. En ce qui concerne le SMO, nous remarquons que la performance globale est presque 10 % meilleure avec 10 catégories, ce qui peut s'expliquer par le fait que dans le cas de 10 catégories, la condition de la limite de 500 attributs fonctionne mieux parce que les 500 attributs importants dans 10 catégories sont plus discriminants que dans 4 catégories.

TAL M1 Orane Dufour
Mikhail Biriuchinskii

Conclusion projet

Les deux expérimentations menées sur le corpus de résumés de thèses nous ont permis de classifier les textes en fonction de leur discipline. Tout d'abord, nous avons constitué un corpus de 10 catégories comprenant 2000 résumés de thèses en utilisant une méthode de collecte de données basée sur une API XML fournie par le site web Theses.fr. Ensuite, nous avons appliqué plusieurs algorithmes de classification, tels que le K-NN, le J48 et le Multinomial Naïve Bayes, pour évaluer leur performance dans la classification des textes.

Dans l'ensemble, nous avons obtenu de relativement bons résultats avec le classifieur SMO qui a été capable de classer les résumés de thèses avec une précision moyenne de 92 % pour les 10 catégories. Cependant, la taille du corpus a rendu les opérations difficiles, en particulier le temps de calcul et la gestion de la mémoire pour les classifieurs et les outils de sélection d'attributs de Weka.

Nous avons également mené une deuxième expérimentation avec un corpus réduit à 4 catégories, afin de vérifier si la réduction du nombre de catégories affectait la performance de classification. Tout d'abord, nous avons commencé avec des classificateurs simples, qui sont souvent utilisés sur des cas plus petits, le modèle IBK prédisant 80,4 % des échantillons. Ensuite, nous avons comparé les méthodes précédentes avec un corpus réduit, obtenant en général des résultats de performance identiques ou inférieurs, à une exception près - une augmentation des scores de F-mesure pour la catégorie Sciences_du_langage.

En conclusion, nous avons montré que la classification des résumés de thèses en fonction de leur discipline est une tâche réalisable, et ce malgré des catégories qui semblent proches. Les résultats obtenus ont montré que la réduction du nombre de catégories n’influe pas sur les résultats. Le classifieurs SMO et le classifieur Multinomial Naive Bayes obtiennent les meilleurs résultats pour ce corpus, le premier atteint même un score très satisfaisant en classant correctement 90% des documents du corpus.

Weka est un outil très complet qui permet d’effectuer de nombreuses opérations de prétraitement sur le texte qui peuvent complémenter des opérations plus traditionnelles tels que l’emploi de listes de mots vides ou d’expressions régulières pour supprimer les attributs inutiles.
Cependant une connaissance fine des données du corpus reste requise afin de pouvoir interpréter au mieux les résultats obtenus par les classifieurs et prévenir des biais éventuels.