Fouille de données, fouille de textes : les titres de HAL

Table des matières

[Table des matières 1](#_Toc536393177)

[Introduction 1](#_Toc536393178)

[1. Analyse des champs non textuels 1](#_Toc536393179)

[2. Analyse du titre en fonction du domaine 3](#_Toc536393180)

[3. Classification automatique d’un titre par domaine 4](#_Toc536393181)

[Conclusion 4](#_Toc536393182)

# Introduction

Dans ce travail, nous étudions des titres de publications stockées sur l’archive ouverte HAL. 53 426 titres ont été fournis avec, pour chacun, des informations supplémentaires liées à la publication titrée : domaine (Lettres, Linguistique ou Informatique), type (article, communication ou chapitre d’ouvrage), année de publication et nombre d’auteurs. Nous parlerons de **variables** (appelées aussi traits, attributs, descripteurs ou *features*). Dans une première partie, nous nous attachons à rechercher des relations entre les variables non textuelles, les numériques et les énumérations (*Nominal* pour Weka, *categorical* pour Scikit-learn). Une relation permet, à partir d’une ou plusieurs **variables**, de trouver la valeur d’une autre, appelée **variable cible** (nous ne parlons pas du cas de plusieurs variables cibles). Dans la deuxième partie, nous prenons comme variable cible le **domaine** et essayons d’établir des **variables** **dérivées du titre** pour ensuite construire un modèle de classification automatique dans la troisième partie.

# 1. Analyse des champs non textuels

Dans un premier temps, nous utilisons simplement Python et **pandas** pour recueillir des informations générales sur les champs non textuels à partir du CSV encodé en UTF-8 :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Domaine | Nb. titres | % |  | Support | Nb. titres | % |
| Informatique | 18 694 | 35 % |  | Communication | 24 693 | 46 % |
| Linguistique | 17 582 | 33 % |  | Article | 17 273 | 32 % |
| Lettres | 17 150 | 32 % |  | Chapitre | 11 460 | 22 % |

On compte entre 1 et 38 auteurs par titres, avec une moyenne de 1,83. 62 % des titres ont été écrits par un auteur, 14 % par deux auteurs, 12 % par trois et 6 % par 4 soit 94 % des titres ont été écrit par moins de 5 auteurs. Si on écarte certaines incohérences, les années s’étendent de 1771 à 2018. Les titres de 2000 à 2018 comptent pour 93 % des titres.

Après avoir étudié dans les grandes lignes nos données non-textuelles, nous utilisons Weka pour analyser les relations entre celles-ci. Nous typons nos variables, *Nominal* pour **domaine** et **support**, *Numeric* pour **années** et **auteurs**, et supprimons ensuite la variable **titre**. Pour voir les relations qu’entretiennent les variables entre eux, nous utilisons un classifieur de type arbre de décision avec l’algorithme J48 avec ses paramètres par défauts car il gère les variables de type *Numeric* en plus du *Nominal*. Nous ne pouvons construire d’arbre qu’en ciblant les variables de type *Nominal* néanmoins, nous en construisons donc deux : l’un ciblant le **domaine**, l’autre le **support**. En observant les arbres, on peut estimer l’importance des variables pour déterminer la cible. Pour la cible **domaine**, rien n’apparaît clairement à l’étude de l’arbre de Weka. Pour la cible **support**, le premier choix de l’arbre, et donc le critère le plus déterminant, est la variable **domaine**, mais il n’est pas le seul : vient ensuite l’**année** et plus tard le **nombre d’auteurs**. En reprenant nos statistiques en Python, on dresse ces trois tableaux en considérant deux variables :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sup-port** | **Domaine** | | |  | **Sup-port** | **Domaine** | | |  | **Sup-port** | **Domaine** | | |
| Info. | Lett. | Ling. |  | Info. | Lett. | Ling. |  | Info. | Lett. | Ling. |
| Article | **3 898** | 6 617 | 6 758 |  | ART | 23 % | 38 % | 39 % |  | ART | 21 % | 39 % | 38 % |
| Comm. | **13 629** | 4 498 | 6 566 |  | COMM | **55 %** | 18 % | 27 % |  | COMM | **73 %** | 26 % | 37 % |
| Chapitre | 1 167 | 6 035 | 4 258 |  | COUV | **10 %** | 53 % | 37 % |  | COUV | **6 %** | 35 % | 24 % |

À gauche, le comptage brut. Au milieu, on prend **domaine** pour cible et **support** comme variable, les pourcentages sont par rapport au nombre total de titres pour un support donné. À droite, on intervertit en prenant **support** pour cible et **domaine** comme variable, les pourcentages sont donnés par rapport au nombre total de titres pour un domaine donné. Sur nos données, un titre de chapitre a seulement 10 % de chance d’être dans le domaine informatique alors que pour une communication ce pourcentage s’élève à 55 %. Un titre dans le domaine informatique a 73 % de chance d’être celui d’une communication et seulement 6 % celui d’un chapitre. Rien d’aussi probant ne se dégage des autres statistiques qui ont une répartition plus équitable. **On peut donc affirmer que les domaines ne sont pas représentés de la même façon entre les supports et que les supports ne sont pas représentés de la même façon entre les disciplines**.

Nous utilisons à présent une autre fonction de Weka : la visualisation. Nous testons plusieurs combinaisons de variables en abscisse et en ordonnée, ainsi que d’éparpillement (*jitter*). Nous supprimons pour utiliser cette fonction les trois titres ayant comme année 195, 217 et 218 sinon les années sont irreprésentables. Le résultat est décevant : le fait que la représentation soit écrasée sur les bordures ne nous permet pas d’identifier clairement les tendances, même celle évoquée du domaine Informatique vers le support Communication, alors qu’elle est forte. Nous détectons cependant dans la configuration **support** en abscisse et **auteurs** en ordonnée des formes de points différentes. Nous construisons un script Python pour établir des statistiques précises en prenant pour cible le **nombre d’auteurs** par rapport aux **supports** et obtenons des résultats dont nous reprenons les 4 premières lignes :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nombre d’auteurs** | **ART** | | | **COMM** | | | **COUV** | | |
| Nb | % | Cumul | Nb | % | Cumul | Nb | % | Cumul |
| **1** | 12 485 | 72 % | **72 %1** | 11 213 | 45 % | **45 %1** | 9 490 | **83 %1** | 83 % |
| **2** | 2 409 | 14 % | 86 % | 4 052 | 16 % | 61 % | 1 270 | 11 % | **94 %2** |
| **3** | 1 204 | 7 % | **93 %2** | 4 804 | 20 % | 81 % | 418 | 4 % | 98 % |
| **4** | 6 25 | 4 % | 97 % | 2 733 | **11 %4** | **92 %2** | 160 | 1 % | 99 % |

Ce tableau confirme l’intuition visuelle : les différents **supports** n’ont pas le même profil concernant le **nombre d’auteurs**. (1) Si le support est un chapitre d’ouvrage, il y a 83 % de chance qu’il soit écrit par un seul auteur, cette proportion est à 72 % pour un article et elle tombe à 45 % pour une communication. (2) L’accélération du cumul n’est pas la même : si le support est un chapitre d’ouvrage, il y a 94 % de chance qu’il soit écrit par 1 ou 2 auteurs. Pour dépasser les 90 %, il faut considérer les titres écrits par de 1 à 3 auteurs pour les articles, et les titres écrits par de 1 à 4 auteurs pour les communications. **Le** **nombre d’auteurs** **varie donc bien en fonction du** **support** **de publication.** On peut également inverser la réflexion et prendre pour cible le **support** par rapport au **nombre d’auteurs**, en affichant seulement de 1 à 4 auteurs :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Support** | **1** | | **2** | | **3** | | **4** | |
| Nb | % | Nb | % | Nb | % | Nb | % |
| **ART** | 12 485 | 38 % | 2 409 | 31 % | 1 204 | 19 % | 6 25 | 18 % |
| **COMM** | 11 213 | 34 % | 4 052 | 52 % | 4 804 | 75 % | 2 733 | **78 %4** |
| **COUV** | 9 490 | 29 % | 1 270 | 16 % | 418 | 7 % | 160 | 5 % |

La règle *« Si un titre a 4 auteurs alors il a pour support une communication »* pour trouver toutes les communications a une précision de 78 %, en d’autres termes, sa valeur prédictive positive est de 78 %, mais elle a un rappel de 11 % seulement (4). Notre mesure de compromis, ou f-mesure, est de 0.19. On peut comparer cette mesure à une autre reprenant le critère le plus déterminant choisi par l’algorithme J48 pour trouver le **support**, le **domaine** : *« Si un titre est en informatique, c’est une communication »*. Cette règle a une précision de 73 % et un rappel de 55 %, soit une f-mesure de 0.63 ce qui est bien meilleur.

# 2. Analyse du titre en fonction du domaine

Nous créons à partir de nos données un nouveau fichier Weka (.arff) et nous ne gardons que la variable **titre** et **domaine** comme cible. Nous transformons la colonne titre en 1000 colonnes numériques indiquant la présence d’un seul mot avec la fonction StringToWordVector(), en appliquant le Snowball stemmer et la mise en casse minuscule (pour que représentation, Représentations et représentations ne forment qu’une colonne), avec une liste de mots vides français (pour que les mots plus fréquents les plus neutres ne soient pas pris en compte pour la création de colonnes) avec une fréquence minimum de 5 mots. On n’oublie pas d’ajouter un préfixe pour ces colonnes. On lance ensuite un classificateur arbre J48 en cross-validation avec 3 replis pour ne pas le surentraîner : nous aimerions avoir des résultats généralisables et non surentraînés sur notre jeu de données. Nous n’avons pas pu mener à bien cette approche par manque de mémoire vive. Nous décidons d’aborder la question autrement avec d’autres outils.

Nous divisons nos 53 426 titres en deux parties, 2 tiers pour l’échantillon d’apprentissage, *train*, 1 tier pour l’échantillon *test*, avec la fonction train\_test\_split de **scikit-learn** en veillant à avoir à la même proportion de domaines dans les deux grâce à l’argument *stratify*. Nous analysons les longueurs moyennes des titres en caractères et en mots, on note la position médiane de la linguistique mais malheureusement elles sont trop proches pour être véritablement distinctives :

|  |  |  |
| --- | --- | --- |
| Domaine | Nombre de caractères | Nombre de mots |
| Linguistique | 81 | 12.08 |
| Informatique | 80 | 11.70 |
| Lettres | 76 | 12.34 |

Nous décidons d’envoyer tous nos titres à Talismane pour lemmatisation. Nous construisons ensuite pour chaque titre une liste des lemmes employés, en supprimant les mots vides, les doublons et en prenant la forme en casse minuscule si le lemme est ‘\_’, que nous appelons « silhouette ». Nous regardons les mots les plus fréquents des silhouettes dans chaque domaine du corpus *train*. En linguistique, avec 1 316 occurrences, le mot « français » est le plus fréquent. Cela représente 11 % de la totalité des mots des silhouettes de ce domaine. Comme il n’a y plus de doublons, nous pouvons dire que 1 316 titres du domaine linguistique ont au moins une fois ce mot. La règle « Si le titre contient le mot "français", alors il est du domaine Linguistique » a une précision, ou valeur prédictive positive, égale à 1 316 divisé par le nombre total de titres avec ce mot, soit 1 316 en Linguistique, 103 en Informatique et 296 en Lettres : 77 %. Elle a un rappel égal à 1 316 divisé par le nombre total de titres dans le domaine Linguistique dans le corpus *train*, soit 11 722, pour un rappel de 11 % et une F-mesure de 20 %. C’est en combinant ces règles que l’on peut espérer obtenir un bon classificateur. **Si nous devions prendre 10 traits, nous choisirions les 10 mots présentant la meilleure F-mesure dans chaque domaine** :

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Info. | P | R | F | Lettres | P | R | F | Linguistique | P | R | F |
| système | 87 | 8 | 14 | siècle | 75 | 6 | 11 | français | 77 | 11 | 20 |
| application | 94 | 6 | 11 | littérature | 90 | 5 | 10 | langue | 85 | 11 | 20 |
| modèle | 80 | 6 | 11 | roman | 82 | 4 | 8 | linguistique | 91 | 7 | 12 |
| réseau | 93 | 6 | 11 | entre | 41 | 4 | 7 | discours | 79 | 5 | 10 |
| donnée | 83 | 5 | 9 | histoire | 70 | 4 | 7 | analyse | 43 | 4 | 7 |
| analyse | 54 | 5 | 9 | chez | 56 | 4 | 7 | entre | 41 | 4 | 7 |
| approche | 62 | 5 | 8 | littéraire | 82 | 3 | 6 | étude | 50 | 4 | 7 |
| image | 69 | 4 | 8 | écriture | 62 | 3 | 5 | cas | 57 | 4 | 7 |
| modélisation | 90 | 4 | 7 | français | 17 | 3 | 5 | corpus | 71 | 3 | 6 |
| multi | 97 | 4 | 7 | théâtre | 88 | 2 | 4 | construction | 55 | 3 | 5 |

Néanmoins, la couverture de ces 10 traits textuels est trop faible : si on regarde le pourcentage de titres dans un domaine donné ayant au moins de ces 10 premiers mots, on est seulement à 40 % en Informatique, 30 % en Lettres et 42 % Linguistique. Nous voulons savoir combien de traits faut-il prendre pour chaque domaine pour atteindre une couverture donnée:

|  |  |  |  |
| --- | --- | --- | --- |
| Quelle couverture pour : | Informatique | Lettres | Linguistique |
| 50 traits | 75 % | 58 % | 70 % |
| 150 traits | 90 % | 78 % | 86 % |
| 1 000 traits | 99 % | 95 % | 97 % |
| 10 000 traits | 100 % | 99 % | 100 % |

Si on mesure la dérivée de la fonction reliant le nombre de traits à la couverture donnée, on constate que l’Informatique s’identifie bien plus vite et plus facilement car elle nécessite un faible nombre de traits pour atteindre les 90 % de couverture. De plus, il reste à résoudre la problématique des mots présents dans plusieurs domaines : comment catégoriser le titre dans ce cas, avec quelles pondérations ? Ces calculs sont aux centres des modèles que **scikit-learn** fournit.

# 3. Classification automatique d’un titre par domaine

Nous avons deux corpus, le *train* et le *test*. Nous dédoublons ces deux corpus : deux avec seulement les **mots des silhouettes**, *trainMS* et *testMS*, deux avec les données non textuelles en plus **nombre d’auteurs**, **année** et **support**, *trainMSDT* et *testMSDT*. Notre variable cible est le **domaine**. Nous appliquons d’abord notre propre modèle à *trainMS* et *testsMS*. Celui-ci donne un score par domaine à chaque silhouette. Ce score est la somme des F-mesures pour un domaine donné des mots de la silhouette. L’algorithme considère le titre étant du domaine ayant le plus haut score. Pour calibrer le temps d’exécution et la consommation mémoire, nous nous limitons à reconnaître les X premiers mots de chaque domaine, les autres sont ignorés. Nous avons fait les calculs pour 100, 1 000 et 10 000 traits. Du fait des imperfections de notre algorithme, il était intéressant de le prétester sur le corpus ayant permis d’établir le modèle, pour mesurer la « perte » subit par la limitation.

avons essayé d’abord d’appliquer un modèle de notre invention au problème : en g

Nous allons construire avec **scikit-learn** un arbre de décision puis un modèle SVM. Le premier présente l’avantage d’être lisible par l’humain mais il est moins efficace que le second. Nous construisons nos données dans un DataFrame de **panda** à partir du corpus train. Nous constituons deux jeux : Nous construisons ensuite

Nous cherchons à présent à dégager certains traits textuels de la variable **titre** avec pour variable cible le **domaine**. Pour cela,

Avec la méthode describe() nous obtenons encore moyenne, min, max et les 3 autres quartiles pour l’années et le nombre d’auteurs. La fonction crosstab() nous permet d’avoir, en divisant son résultant par le total, les fréquences des domaines dans les deux sous-ensembles et nous constatons qu’elles sont très proches. Value\_count

# Conclusion

DDD