Projet d'Approches probabilistes appliquées au TAL

Par Marie-France Jallade, Léonard Fromont - M1 LI

Afin de démarrer le programme, il faut mettre en argument le fichier AA puis le fichier subtitles. L'accès aux fichiers d'annotations sémantiques et phonétiques est codée en dur dans le programme.

I - Extraction des binômes

a) Comment les extraire ?

Faut-il prendre en compte les catégories dans l'extraction et dans la représentation des binômes? Comment visualiser nos données?

Nous utilisons deux dictionnaires, un dico_word qui prend en clé un binôme et son nombre d'occurrence et un dico_cat qui prend en clé une suite de catégorie et en valeur son nombre d'occurrences, ce dernier nous permettra de représenter la distribution des binômes. Les dictionnaires avant filtrage ressemblent à cela:

('sujets/NC', 'et/CC', 'articles/NC')	10242
('Cet/NC', 'article/NC', 'ou/CC', 'cette/DET', 'section/NC')	845
('lacs/NC', 'et/CC', 'cours/NC')	845
('e/NC', 'et/CC', 'f/NC')	584
("l'/NC", 'agriculture/NC', 'et/CC', "l'/DET", 'agronomie/NC')	460
('noir/ADJ', 'et/CC', 'blanc/ADJ')	392
('g/NC', 'et/CC', 'h/NC')	357
('secondaires/ADJ', 'ou/CC', 'tertiaires/ADJ')	326
('article/NC', 'ou/CC', 'section/NC')	324
("l'/NC", 'architecture/NC', 'ou/CC', "l'/DET", 'urbanisme/NC')	324
('sources/NC', 'ou/CC', 'liens/NC')	320
('contraire/ADJ', 'ou/CC', 'complémentaire/ADJ')	295
('nationales/ADJ', 'et/CC', 'continentales/ADJ')	286
('textuelles/ADJ', 'et/CC', 'lexicales/ADJ')	282
('internes/ADJ', 'ou/CC', 'externes/ADJ')	267
('religions/NC', 'et/CC', 'croyances/NC')	244
('h/NC', 'et/CC', 'i/NC')	238
('f/NC', 'et/CC', 'g/NC')	238
('une/NC', 'chronologie/NC', 'ou/CC', 'une/DET', 'date/NC')	233
('Une/NC', 'réorganisation/NC', 'et/CC', 'une/DET', 'clarification/NC')	217
('lieu/NC', 'et/CC', 'place/NC')	205
('A/NC', 'et/CC', 'B/NC')	198
('i/NC', 'et/CC', 'j/NC')	186
('de/P', 'commerce/NC', 'et/CC', "d'/P", 'industrie/NC')	172
('récompenses/NC', 'et/CC', 'distinctions/NC')	171
('postaux/ADJ', 'et/CC', 'téléphoniques/ADJ')	156
('écrites/VPP', 'et/CC', 'composées/VPP')	151
('humaines/ADJ', 'et/CC', 'sociales/ADJ')	140
('Évolution/NC', 'et/CC', 'structure/NC')	131
("d'/P", 'art/NC', 'et/CC', "d'/P", 'histoire/NC')	128

Dictionnaire de binômes AA

('M./NC', 'et/CC', 'Mme/NC')	6953
('Mesdames/NC', 'et/CC', 'messieurs,/NC')	4488
('de/P', 'vie/NC', 'ou/CC', 'de/P', 'mort./NC')	3961
('mère/NC', 'et/CC', 'moi,/NC')	3689
('heure/NC', 'et/CC', 'demie/NC')	3230
('seule/ADJ', 'et/CC', 'unique/ADJ')	3094
('ans/NC', 'et/CC', 'demi/NC')	3043
('la/NC', 'vie/NC', 'et/CC', 'la/DET', 'mort./NC')	3043
('le/NC', 'bien/NC', 'et/CC', 'le/DET', 'mal./NC')	2958
('hommes/NC', 'et/CC', 'femmes/NC')	2635
('père/NC', 'et/CC', 'moi,/NC')	2567
('en/P', 'chair/NC', 'et/CC', 'en/P', 'os./NC')	2482
('Mesdames/NC', 'et/CC', 'messieurs/NC')	2482
('les/NC', 'hommes/NC', 'et/CC', 'les/DET', 'femmes/NC')	2397
('mesdames/ADJ', 'et/CC', 'messieurs,/ADJ')	2380
('jour/NC', 'et/CC', 'nuit/NC')	2074
('frères/NC', 'et/CC', 'sœurs/NC')	1904
('Ma/NC', 'femme/NC', 'et/CC', 'ma/DET', 'fille/NC')	1853
('seul/ADJ', 'et/CC', 'unique/ADJ')	1836
('femme/NC', 'et/CC', 'moi,/NC')	1751
('frères/NC', 'et/CC', 'soeurs/NC')	1734
('la/NC', 'vie/NC', 'et/CC', 'la/DET', 'mort/NC')	1734
('plaise/V', 'ou/CC', 'non,/V')	1734
('les/NC', 'femmes/NC', 'et/CC', 'les/DET', 'enfants/NC')	1547
('un/NC', 'garçon/NC', 'ou/CC', 'une/DET', 'fille/NC')	1547
('un/NC', 'homme/NC', 'et/CC', 'une/DET', 'femme/NC')	1513
('faits/NC', 'et/CC', 'gestes/NC')	1479
('Mr/NC', 'et/CC', 'Mme/NC')	1479
('noir/ADJ', 'et/CC', 'blanc/ADJ')	1462
('Tôt/NC', 'ou/CC', 'tard./NC')	1445

Dictionnaire de binômes subtitles

Par Marie-France Jallade, Léonard Fromont - M1 LI

('NC', 'CC', 'NC')	77013
('ADJ', 'CC', 'ADJ')	56025
('DET', 'NC', 'CC', 'DET', 'NC')	87852
('P', 'NC', 'CC', 'P', 'NC')	37442
('VPP', 'CC', 'VPP')	10313
('V', 'CC', 'V')	6251

Dictionnaire des catégories AA subtitles

('DET', 'NC', 'CC', 'DET', 'NC')	1752343
('NC', 'CC', 'NC')	799459
('ADJ', 'CC', 'ADJ')	432701
('V', 'CC', 'V')	216648
('P', 'NC', 'CC', 'P', 'NC')	222462
('VPP', 'CC', 'VPP')	114767

Dictionnaire des catégories

Afin de mieux cibler nos données et d'avoir un corpus relativement homogène, il fallait filtrer les données rares, les données dont le nombre d'occurrences élevé, qui sont dû au type de corpus traité et au "bruit". En effet, certains binômes sont propres au langage utilisé sur Wikipédia et donnent l'impression qu'ils sont particulièrement fréquents dans l'absolu. De plus, le "bruit" correspond aux données qui sont présentes dans l'extraction, car elles correspondaient au format à extraire, sans pour autant être des binômes (les dates, les lettres, les parenthèses et les chiffres).

Ainsi, nous pouvons remarquer dans l'image du dictionnaire de binômes AA située plus haut, que le binôme "sujets et articles" apparaît au minimum deux fois plus que les autres binômes. Alors, nous pouvons remarquer que les quatres premières clés de ce dictionnaire sont toutes propres à Wikipédia et qu'il faut donc les supprimer. En effet, leur nombre d'occurrences est dû au corpus, mais pas à une propriété d'une binomiale.

Comment nettoyer les données? Quels filtres appliquer?

Afin de ne considérer que les données extraites qui sont intrinsèquement des binômes, nous avons utilisé trois filtres. Le premier supprime les binômes qui apparaissent qu'une seule fois, le deuxième supprime les données "bruit" et le troisième supprime les binômes ayant un nombre élevé d'occurrences. Ce filtrage va réduire la taille des données.

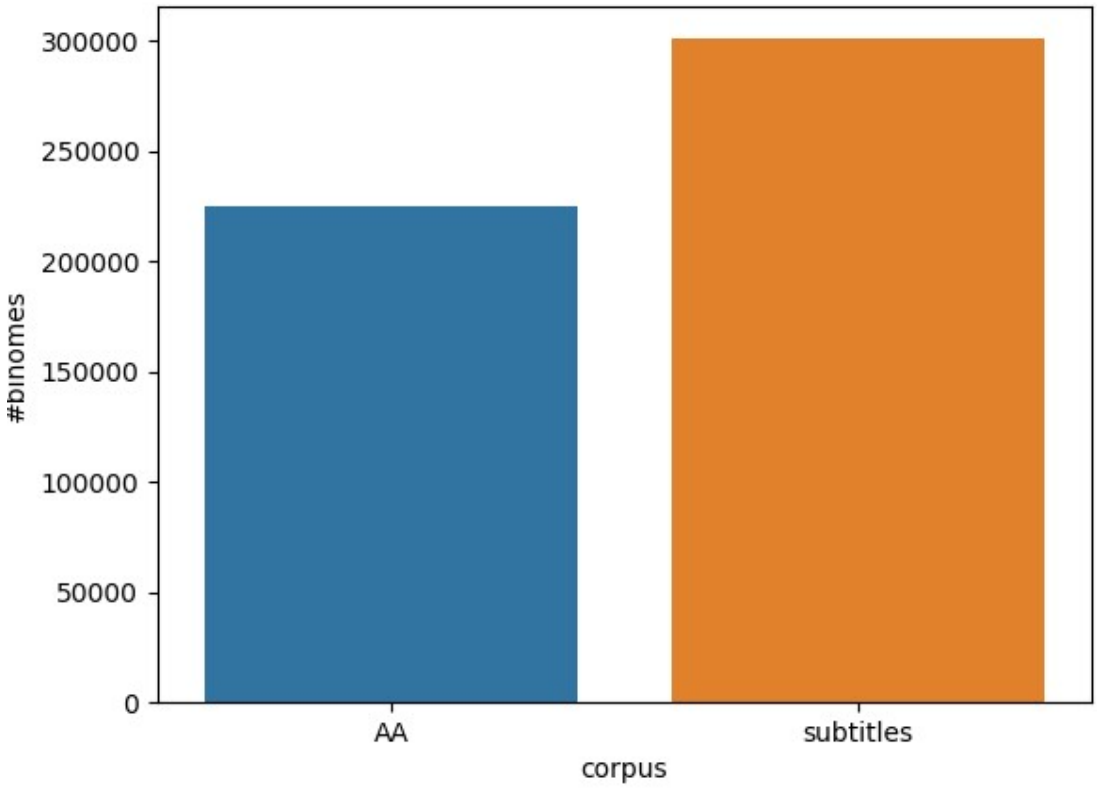
b) La comparaison des distributions des corpus

Afin de déterminer si les binômes des deux corpus sont similaires, il nous fallait comparer leurs distributions. Nous avions une hypothèse à vérifier avec les graphiques: les binômes seraient distribués différemment étant donné la différence de langage utilisé dans les deux corpus. En effet, le corpus des sous-titres contiendrait des binômes appartenant au langage oral, à l'inverse des binômes présents dans le corpus tiré de Wikipédia puisque ce dernier correspond davantage à un langage écrit. De plus, en partant du principe que le vocabulaire de Wikipédia est plus vaste et que la construction des phrases est plus riche que celle dans le corpus de sous-titres, nous pensions aussi qu'il y aurait plus de binômes

différents dans le corpus tiré de Wikipédia que dans le corpus de sous-titres.

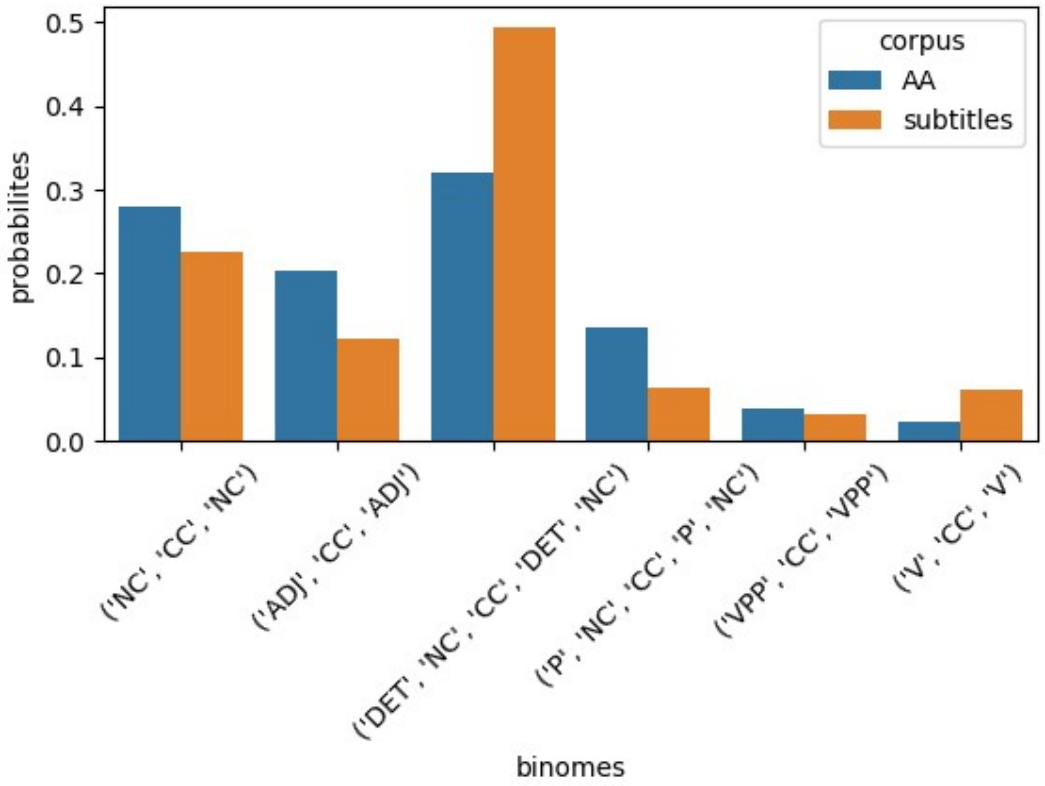
Nous nous étions aussi demandé si les caractéristiques influant sur l'ordre d'un binôme seraient les mêmes dans les deux corpus. Nous avions laissé tombé cette hypothèse puisque le type de corpus ne devraient pas influencer les caractéristiques qui prédisent l'ordre d'un binôme. D'autant plus, l'origine des données utilisées pour faire le classifieur du projet n'était pas précisée, ce qui laisse penser que cette information n'est pas cruciale.

Nous avons choisi d'utiliser des corpus de taille similaire pour les sous-titres et pour Wikipédia afin que la taille ne soit pas un facteur biaisant les statistiques. Nous avons donc utilisé un corpus de 2.8 Go pour Wikipédia et un corpus de 2.3 Go pour les sous-titres. Voici donc un aperçu du nombre d'occurrences total des binômes, ainsi que leur distribution dans les corpus. Nous remarquons que c'est dans le corpus de sous-titres qu'il y a le plus de binômes.



Histogramme représentant le nombre de binômes par corpus

Afin de réaliser cet histogramme ainsi que le suivant, nous avons utilisé les librairies pandas, matplotlib et seaborn.



Histogramme représentant la distribution des binômes selon le corpus

Pour réaliser cet histogramme, nous avons eu recours aux dictionnaires des catégories. Dans ces derniers, les clés étaient initialement le compte d'occurrences de la séquence de catégories, ce qui ne nous permettait pas de créer une distribution de probabilités. Nous avons donc probabilisé les valeurs du dictionnaires en utilisant de la fréquence relative.

A l'aide de l'histogramme, nous avons pu remarquer que les deux corpus ne se comportent pas de la même manière, quand bien même des similitudes peuvent être observées. D'abord, la séquence (DET, NC, CC, DET, NC) est celle qui prédomine qu'importe le corpus pris en compte. Néanmoins, nous observons des disparités dans les sous-titres qui ne sont pas ou peu présentes dans le corpus Wikipédia: il y a une forte propension des binômes (DET, NC, CC, DET, NC) puisque cet enchaînement qui est très courant à l'oral (il peut constituer la réponse à une question par exemple) concerne presque un binôme sur deux. A l'inverse, la séquence (VPP, CC, VPP) est très peu représentée dans le corpus de sous-titres. Ce phénomène peut être expliqué par le fait que la coordination de deux participes passés est peu courante à l'oral. Ces disparités entre type de binômes est moins flagrante dans le corpus Wikipédia qui comprend presque autant de séquences (DET, NC, CC, DET, NC) que de séquences (NC, CC, NC).

Taille des données

Nous sommes d'abord passé par des corpus plus petits afin de vérifier si nos fonctions n'avaient pas de problèmes, avant de lancer le code sur des données plus importantes.

Cependant lors de notre premier essai sur les données plus lourdes, nous avons rencontré un problème dû à leur stockage dans le système. Pour cause de l'utilisation de la fonction readlines() sur nos corpus. Cette fonction n'était pas idéale car elle stockait l'intégralité d'un corpus afin de pouvoir de le lire, ce qui était trop lourd pour le programme lancé sur toutes les données. Alors, nous avons utilisé la fonction readline() qui nous permettait de lire un corpus ligne par ligne sans avoir à le stocker en intégralité, et nous avons réinitialisé la donnée de stockage.

Formatage des données

Les fichiers sous-titres n'étaient pas formatés de la même manière, il a donc fallu adapter notre code en conséquence. En effet, le fichier de sous-titres contenait une colonne en moins, nous avons modifié notre code pour qu'il puisse récupérer les mêmes informations même si le format était différent. De plus, les fichiers du Wikipédia n'était pas identiques aux données d'entraînement. En effet, chaque fichier conll du Wikipédia d'entrainement était compressé et se situait dans un unique dossier. Tandis que les données récupérées via téléchargement sur internet étaient sous forme de dossier général comprenant des sous-dossiers et chaque sous dossier comprenait un nombre de fichier conll non compressé; ajouté à cela, des fichiers d'une autre extension étaient présents. Il aurait fallu adapter à nouveau le code pour prendre en compte ces différences, c'est à dire ne pas lire les autres extensions que conll, et lire des fichiers qui n'étaient plus sous format compressé.

Nous avons donc en conséquence recréé un dossier comprenant un regroupement de fichiers conll provenant des multiples dossiers du dossier principal.

Réaliser les distributions de probabilités

Pour réaliser la distribution des binômes selon les corpus, nous avons utilisé un dictionnaire de catégories qui avait en clé la suite de catégorie contenue dans un binôme. Nous avons initialement mis en valeur le nombre d'occurrence de cette suite de catégorie. et avons probabilisé le résultat brut, qu'est le nombre d'occurrence, afin d'avoir une distribution de probabilité.

Pour cela nous avons voulu diviser le nombre d'occurrences d'une suite de catégories par le nombre total de binômes catégoriels. Mais nous nous sommes rendus compte que le dictionnaire pour les catégories n'était pas filtré et prenait en compte la suite de catégories de tous les binômes, même ceux qui avaient été filtrés dans le dictionnaire des binômes. Cela était problématique pour calculer la fréquence relative d'une suite de catégories, parce que la fréquence relative se faisait sur des données qui ne correspondaient pas. Pour remédier à cela, il aurait fallu créer le dictionnaire de catégories à partir du dictionnaire de binômes filtré, et ne pas créer les deux dictionnaires en même temps: nous aurions alors eu la distribution des catégories des binômes filtrés. Ainsi, le graphique représentant la distribution des binômes selon le corpus concerne l'ensemble des binômes, elle est donc biaisée puisqu'elle n'a subit aucun filtrage.

II - Ordre préférentiel des binômes

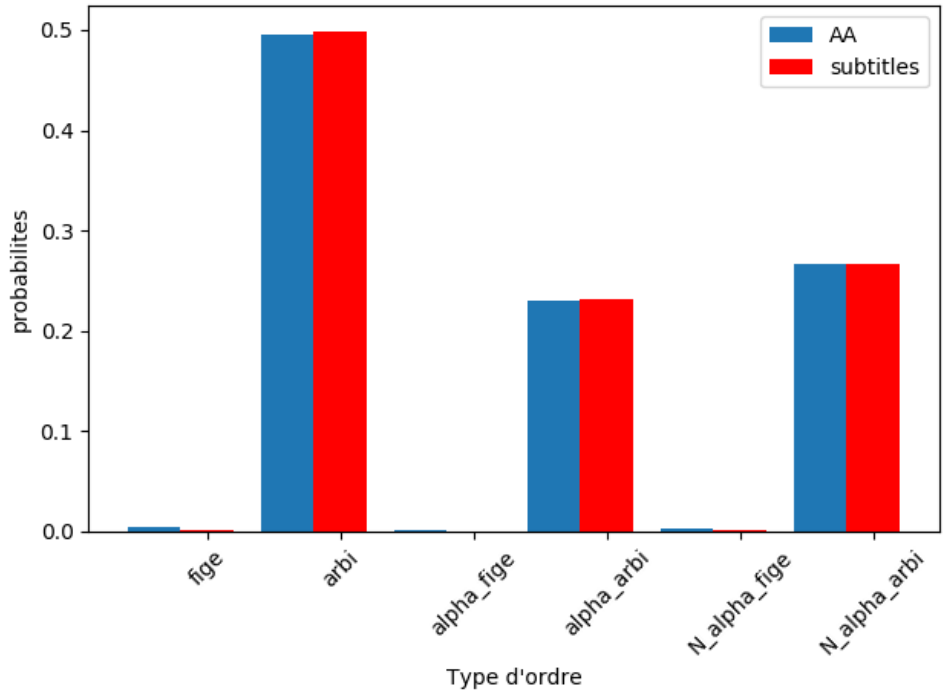
Nous cherchons à déterminer à quel point l'ordre d'un binôme est figé. Pour cela, il nous fallait choisir une frontière à partir de laquelle un binôme était considéré comme étant figé ou non.

Comment déterminer s'il existe un ordre préférentiel?

Au début, nous pensions trancher que l'ordre d'un binôme était arbitraire à partir du moment où il existait au moins une occurrence des deux ordres possible. Mais ce choix aurait été trop radical et ne laissait pas de place à une excentricité du corpus. Par excentricité, nous voulons dire une occurrence étrange comme "soeurs et frères" alors que nous savons pour sûr que "frères et soeurs" est un binôme figé. Ainsi, nous avons choisi un seuil à partir duquel un binôme était considéré comme étant figé même s'il existait une occurrence de son inverse. Pour cela, nous calculons pour l'ordre alphabétique et l'ordre non alphabétique:

$$\frac{nbr\ de\ fois\ que\ l'ordre\ apparait}{nbr\ d'occurrence\ du\ binome\ ordres\ confondus}$$

Si cette équation est supérieure à 0.7 pour un des deux ordres, alors l'ordre en question domine et le binôme est considéré comme étant figé. Dans le cas contraire, c'est à dire quand aucun des ordres ne dépasse ce seuil, nous considérons qu'il n'y pas d'ordre qui domine (et donc que l'ordre du binôme est arbitraire). Nous avons déterminé 0.7 comme seuil car pour un seuil à 0.5 nous aurions partialité entre les deux ordres, ce qui nous ne permettait pas de dire si un ordre dominait l'autre. Alors, il a fallu choisir un seuil qui était supérieur à 0.5 et qui était dans l'intervalle 0.5 - 1 sans être trop proche des extrêmes; c'est pourquoi 0.7 paraissait être un bon choix puisqu'il se situe à peu près au milieu de cet intervalle.



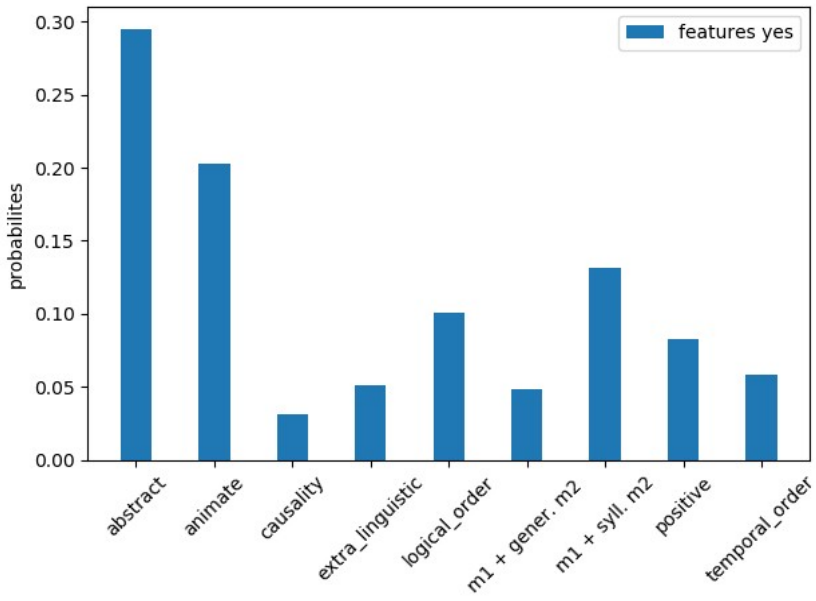
Histogramme représentant la probabilité de binômes figés selon leur ordre

D'après l'histogramme ci-dessous, il y aurait autant de binômes arbitraires que de binômes figés. En effet, il y a environ 50% des binômes qui sont dans un ordre arbitraire. Nous pouvons en conclure que le fait qu'un binôme soit figé ou non ne dépend pas du type de corpus. En effet, les binômes sont caractérisés de la même manière qu'importe le corpus étudié; les différences entre les corpus concernant les binômes se basent sur les catégories morpho-syntaxiques intervenant dans les binômes mais pas sur l'ordre interne au binôme. Nous notons néanmoins que les barres qui devraient être complémentaires de "arbi", "alpha_arbi", et "N_alpha_arbi" ne le sont pas.

III - Facteurs influant l'ordre des binômes

a) Déterminer une hiérarchie entre les facteurs

Afin de conclure quelles caractéristiques déterminent le plus l'ordre d'un binôme, il fallait rendre compte des caractéristiques qui étaient les plus présentes dans les descriptions des binômes. Pour cela, nous avons utilisé une fonction repr_graph_liste_ordre qui crée un graphique avec en ordonnée les caractéristiques prises en compte, et en abscisse la probabilité d'apparition de la caractéristique pour les mots du binômes.



Histogramme représentant les proportions des caractéristiques présentes dans un binôme

Cet histogramme nous montre qu'il y'a une majorité (30%) de mots qui détiennent la caractéristique "abstract", elle est vérifiée tandis que la caractéristique "causality" elle, ne l'est pas.

Cette démarche nous permet de commencer à voir une ébauche hiérarchique entre caractéristiques (nous pourrions dire que les caractéristiques qui apparaissent peu de fois ne sont pas significatifs), mais elle ne nous permet pas à elle seule de classifier un binôme. En effet, étant donné que toutes les caractéristiques ne rentrent pas en jeu pour déterminer l'ordre d'un binôme, il faudrait prédire l'ordre du binôme en testant toutes les caractéristiques, jusqu'à trouver la ou les caractéristique(s) qui permettent effectivement de prédire le binôme.

C'est alors en prenant appui à la fois sur l'ébauche de hiérarchie observée auparavant et sur l'étude du vecteur de paramètre d'un classifieur que nous pourrons vérifier des hypothèses concernant la hiérarchie entre caractéristiques. En effet, en regardant le poids attribué à chaque caractéristiques dans le vecteur de paramètre du classifieur, nous pouvons déterminer laquelle des caractéristiques présente dans le binôme a le plus joué sur l'ordre du binôme.

b) Conceptualiser le perceptron

Quelles caractéristiques choisir pour décrire un binôme?

A partir d'un ensemble de caractéristiques observées sur les binômes, nous tirons des features que nous estimons possiblement significatives pour l'ensemble des binômes.
Nous avons utilisé deux informations pour décrire un binôme. D'abord le binôme en toute lettres qui sera utilisé pour avoir accès au nombre d'occurrences du binôme dans l'ordre alphabétique et dans l'ordre non-alphabétique; suivi une liste de caractéristiques. Ces

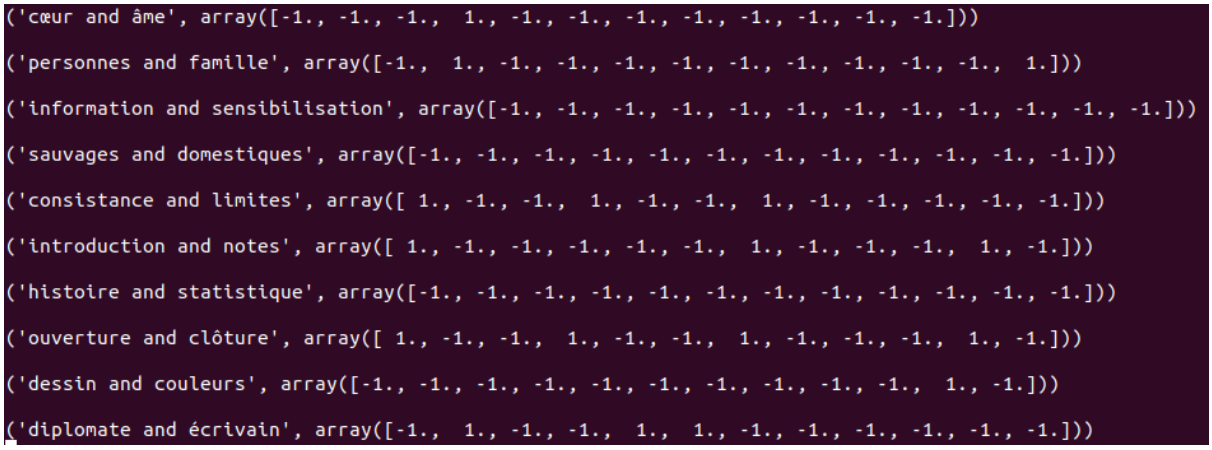
dernières sont composées des caractéristiques sémantiques (“semantic_features.pkl") fournies, ainsi qu’une caractéristique permettant de rendre compte de la longueur du premier mot. Cette dernière a été ajouté suite à la lecture de l’article de Roger Lévy donné en référence qui faisait un aperçu des différentes idées déjà acquises sur les facteurs influant sur l’ordre d’un binôme. Le fait que le premier mot du binôme soit plus court que le deuxième mot du binôme était une caractéristique.

La liste des caractéristiques correspond à cet ordre:

[m1 est abstrait, m1 est animé, m1 est positif, m2 est abstrait, m2 est animé, m2 est positif, m1 a plus de syllabes que m2, extra_linguistic relation, causality relation, temporal_order relation, logical_order relation, m1 est plus général que m2].

Afin de déterminer quel mot du binôme était le plus long, nous avons utilisé les représentations phonétique présentes dans le fichier json fourni. C’est en faisant un split sur les points qui sépare les syllabes que nous avons pu les compter et comparer leur nombre entre les deux éléments d’un même binôme.

La caractéristique ne concerne pas le binôme si elle vaut -1, ce n’est pas le cas si elle vaut 1. Par exemple, dans l’image ci-dessous, nous pouvons lire que la caractéristique “m1 est animé” n’est pas présente pour le binôme “coeur et âme”, mais qu’elle est présente pour le binôme “personnes et famille”. Nous pouvons ainsi déduire le poids de la caractéristique “m1 est abstrait” pour ces deux binômes.



E*xemples de vecteurs de binômes*

Quelles données utiliser?

Nous avions pensé utiliser les binômes extraits pour la première partie du projet comme données sur lesquelles effectuer les prédictions; mais les données qui pouvaient effectivement être utilisées (parce qu’elles étaient celles qui étaient les plus décrites) étaient les données annotées sémantiquement. Les binômes extraits n’étaient pas annotés et ne pouvaient pas donc être prédits selon les mêmes critères que les données annotées; c’est pourquoi nous avons choisi d’utiliser exclusivement les données annotées sémantiquement.

Comment déterminer la fonction de score?

Les binômes n'étaient pas étiquetés comme étant figé ou non, ce qui nous aurait permis de coder une fonction de score comparant l'étiquette et la prédiction. Nous avons donc déterminé un seuil, mentionné plus haut, à partir duquel un binôme était considéré comme étant figé, c'est son étiquette. L'étiquette d'un binôme se fait alors par estimation étant donné que l'apprentissage n'est pas supervisé. Cela peut être problématique parce que ces étiquettes ne sont pas vérifiées/ forcément correctes à 100% puisque le perceptron apprend à l'aide d'étiquettes qui ne sont pas nécessairement fiables.

La (non)confiance en l'étiquette estimée ne nous permet pas de dire que le seuil choisi n'est pas un élément biaisant l'apprentissage. Pour déterminer notre score final il nous a fallu établir des données "gold". Etant donné que nous n'avions pas eu accès à des données gold au départ, nous avions donc stipulé que nos données de comparaison dit "gold" seraient nos binômes annotés pour lesquels nous avons au préalable séparé en deux "plages de données" différentes.

Utiliser un fichier stockant les vecteurs

Nous avons eu recours à cette méthode afin de ne pas avoir à stocker les vecteurs dans le code. Néanmoins cette méthode s'est révélée être piégeuse puisqu'elle demandait de prendre des précautions supplémentaires quant au type de formatage des données. En effet, il a fallu recaster en int du vecteur contenant les features, les int étant typés comme des strings; et séparer la représentation en toute lettre du binôme de la liste contenant les int.

Quel critère d'arrêt pour l'apprentissage / quels corpus utiliser?

Nous utilisons un train et un test correspondant comme dit précédemment à 2 plages de données provenant des données annotées en entier. Parmi les trois critères d'arrêt rencontrés pendant le semestre, nous avons choisi d'utiliser le critère selon lequel le classifieur a le meilleur vecteur de paramètre possible si le taux de bonnes réponses stagne (le vecteur de paramètre serait alors optimal). Nous avons choisi ce critère parce que nous ne savions pas quel nombre fixe d'itérations choisir et parce que un taux de bonnes réponses à 100% sur le corpus d'apprentissage est assez rare.

Alors, nous avons séparé les données annotées en 2 parties en faisant pour chaque itération sur le train une shuffle qui permet à chaque fois d'obtenir un train différent que celui précédent ce qui nous permet de ne pas surentrainer le perceptron.

c) Résultats obtenus: le vecteur de paramètre

```
90.0  tour: 5  v_param: [1, -1, -3, -3, -3, -1, -1, -1, -3, -3, 1]
```

(Rappel de l'ordre de nos features: [m1 est abstrait, m1 est animé, m1 est positif, m2 est abstrait, m2 est animé, m2 est positif, m1 a plus de syllabes que m2, extra_linguistic relation, causality relation, temporal_order relation, logical_order relation, m1 est plus général que m2])

Suite à l'exécution de notre perceptron, nous obtenons après cinq époques le vecteur de paramètres ci-dessous, ainsi que le pourcentage de bonnes réponses du perceptron. Nous constatons un résultat de 90% pour un total d'environ 300 mots. Les résultats restent tout de même très fluctuant selon les exécutions (variant entre 75-90%), tandis que le nombre d'epoch varie de 1 à 30.

Ainsi, au vu de notre vecteur de paramètres final (et comme il avait été prédit dans l'histogramme représentant les proportions des caractéristiques présentes dans un binôme) nous pouvons observer que le poids de la feature "m2 est animé". Néanmoins, nous observons quelque étrangetés: les features "m1 est positif", "temporal_order relation" et "logical_order relation" ont un poids faible (-3) alors qu'elles correspondent à des caractéristiques attestés par la littérature sur l'ordre des binômes. En ce qui concerne les autres features, nous ne pouvons pas vraiment nous avancer sur l'interprétation des poids obtenus puisque les valeurs pour lesquelles nous avons établis notre perceptron sont assez petite.

En regardant les caractéristiques généralement admises comme étant celles qui influent l'ordre d'un binôme et celles qui ont été calculées comme telles dans notre vecteur de paramètres, nous remarquons des incohérences. En effet, toutes les features déterminées comme n'influant pas l'ordre d'un binôme sont précisément celles qui influent binôme dans les faits. En fait, pour pouvoir interpréter notre vecteur de paramètre, il faudrait renverser les signes pour obtenir des résultats qui feraient sens à la fois avec les conclusions admises par la littérature et avec les résultats mentionné ci-dessous. Alors, notre vecteur serait: [-1, 1, 3, 3, 3, 1, 1, 1, 3, 3, -1], mais nous ne savons pas comment expliquer ce phénomène.

IV - Conclusion

A la suite de ce projet, nous avons un perceptron qui détermine les facteurs qui influencent le plus l'ordre d'un binôme. Cependant, ces données restent peu précises vu le peu de données mis à notre disposition. Dans le but d'obtenir un vecteur de paramètre plus significatif, nous aurions pu utiliser les données extraites lors de la première partie et les annoter sémantiquement comme les données utilisées dans le perceptron.