Classification, combinaison et regroupements pour séparer les discours de Mitterrand de ceux de Chirac

Laurent Pierron (1), Coskun Durkal (2) et Jean-Baptiste Chevalier (3)

(1) Loria – INRIA-Lorraine
Laurent.Pierron@loria.fr
(2) UHP
durka2@etumail.uhp-nancy.fr
(3) ESIAL
Jean-Baptiste.Chevalier@esial.uhp-nancy.fr

Mots-clés : classifieur Bayesien, détection du sujet du discours, structure de phrase, combinaison d'indices, détection de pourriel, fouille de textes, indicateur statistique, algorithme de recherche séquentiel.

Keywords: Bayesian classifier, speech topic detection, sentence structure, clue mixing, spam detection, text miniing, statistical indicator, sequential search algorithm.

Résumé Afin de sélectionner les phrases de Mitterrand parmi les discours de Chirac, nous avons principalement utilisé un classifieur Bayesien, qui a été complété par une détection du meilleur groupe contigu de phrases. Ensuite, nous avons tenté sans succès d'améliorer les résultats du classifieur, une basée sur le thème du discours (national / international), la seconde basée sur la détection de rupture de structure de phrases (nombre de mots).Enfin, une pondération et un seuillage des réponses du classifieur Bayesien ont finalement permis de maximiser le score final.

Abstract To find Mitterrand's sentences inside Chirac's speeches, we mainly used a Bayesian classifier, which was extended with finding the best contiguous block of sentences. After that, we tried to mix the classifier results with two other methods, without improvement, the first one based on the speech topic (national vs international), the second one based on the sentences structure change (identified by number of words). Finally, after Bayesian classifier results weighting and thresholding, we obtained our best score.

1 Introduction

Après avoir analysé rapidement les phrases, nous avons décidé d'utiliser plusieurs approches pour trouver des indices permettant de déterminer les extraits de discours de Mitterrand au sein des

discours de Chirac. Ces divers indices sont ensuite combinés et suivis d'une dernière technique permettant de sélectionner les blocs de texte contigus appartenant aux discours de Mitterrand.

Les trois approches utilisées afin de sélectionner les phrases de Mitterrand sont les suivantes :

1. Classification Bayesienne des phrases sur des groupes lexicaux extraits des phrases et sur le numéro de ligne.

2. Détection des ruptures sémantiques en début et fin des extraits de discours de Mitterrand par mesure de distance entre les phrases.

3. Recherche de l'unité de thème dans le discours, uniquement réalisée sur l'opposition national/international.

La première partie décrira la méthode utilisée pour la classification Bayesienne et les résultats obtenus, la seconde partie décrira la détection des ruptures sémantiques, la troisième l'unité de thème et la dernière partie la combinaison.

Aucun traitement spécifique par tâche n'est envisagé, la méthode choisie doit permettre de s'adapter aux différentes tâches et tenir compte automatiquement des informations supplémentaires.

2 Classification Bayesienne

2.1 Motivation

Dans cette approche, nous avons décidé de travailler sur les phrases plutôt que sur les discours, car le but du challenge est de trouver les phrases de Mitterrand, si le sujet avait été de déterminer l'auteur du discours, notre approche aurait certainement été différente.

Les phrases de Mitterrand insérées au milieu des discours de Chirac font penser aux messages non sollicités (*spam*) insérés au milieu des messages utiles dans les messageries électroniques. Celui qui a une messagerie électronique lourdement encombrée par le *spam*, peut constater que les filtres *anti-spams* à base de classifieurs Bayesiens donnent de bons résultats dans le tri du courrier.

Même si les phrases de Chirac et Mitterrand peuvent dans certains cas être assez similaires étant issues de discours politiques relativement convenus, les différences de sujets et d'époques peuvent certainement fournir des indicateurs discriminants pour les phrases.

2.2 Classifieur Bayesien

La technique utilisée pour le classifieur Bayesien est celle décrite par (Robinson, 2003).

Les phrases sont divisées en deux classes, les phrases de Chirac et les phrases de Mitterrand.

Chaque phrase est divisée en groupes lexicaux ou graphiques, que nous appellerons *mots* pour simplifier le discours. La présence d'un *mot i* dans une phrase, dont l'auteur est inconnu, permet au classifieur de déterminer une probabilité p_{ci} d'appartenir à la classe Chirac et une probabilité p_{mi} d'appartenir à la classe Mitterrand.

Pour chacune des deux classes Mitterrand et Chirac les probabilités individuelles des *mots* sont combinées pour obtenir une probabilité unique pour chaque phrase. La formule utilisée pour calculer la combinaison est la formule de Fisher proposée par (Robinson, 2003).

2.3 Probabilités individuelles

Les probabilités individuelles des mots sont obtenues par entraînement sur un corpus de phrases dont les auteurs sont connus.

Pour chaque mot *m* dans le corpus d'apprentissage on calcule :

- *Mit(m)* = (le nombre total de phrases de Mitterrand contenant le mot *m*) / (le nombre total de phrases de Mitterrand)
- *Chi(m)* = (le nombre total de phrases de Chirac contenant le mot *m*) / (le nombre total de phrases de Chirac)
- $p_M(m) = Mit(m)/(Mit(m)+Chi(m))$

$p_M(m)$ peut grossièrement être interprétée comme la probabilité qu'une phrase choisie au hasard contenant le mot *m* soit une phrase de Mitterand. On calcule de la même manière $p_C(m)$, le probabilité qu'une phrase choisie au hasard contenant le mot m soit une phrase de Chirac.

2.4 Gestion des mots rares

Il y a un problème avec les probabilités calculées ci-dessus, si un mot est très rare. Par exemple si un mot apparaît dans une seule phrase, qui est une phrase de Mitterrand, la probabilité $p_M(m) = 1.0$. Mais il n'est pas du tout sûr que toutes les phrases futures contenant ce mot seront des phrases de Mitterand, nous n'avons simplement pas assez de données pour décider.

Quand une et une seule phrase contient un certain mot et que cette phrase est une phrase de Mitterrand, notre croyance que la prochaine fois que nous verrons ce mot la phrase soit une phrase de Mitterrand n'est pas de 100%. Ceci parce que nous avons une connaissance a priori, qui nous guide. C'est le grand nombre de répétitions de l'evènement qui nous fera penser qu'il y a de fortes que chance que la prochaine fois que nous verrons ce mot ce soit une phrase de Mitterrand.

Pour tenir compte de cette croyance a priori, nous pouvons utiliser la formule suivante détaillée dans (Robinson, 2003) :

$$f_M(m)=\frac{(s\cdot x+n\cdot p_{(M)}(m))}{(s+n)}$$

Où :

- $p_M(m)$ est la probabilité que la phrase soit une phrase de Mitterrand si elle contient le mot *m*.
- *n* est le nombre de phrases contenant le mot *m*.

- s est la force de la croyance, elle peut prendre une valeur positive quelconque, en général entière supérieure ou égale à 1, si elle vaut 0, on retrouve la formule intiale avec $p_M(m)$. Pour le challenge, on a utilisé la valeur s à 1.

- x est la probabilité initiale que la phrase soit une phrase de Mitterrand, pour le challenge nous avons utilisé 0.5.

Il est également possible d'optimiser les valeurs de s et x par apprentissage.

Dans le programme de classification nous avons donc utilisé f_M et f_C en lieu et place de p_M et p_C.

2.5 Combinaison des probabilités

A ce point nous disposons d'une probabilité $f(m)$ qu'un mot donné soit dans une phrase de Mitterand ou de Chirac. Donc à chaque phrase est associé deux ensembles de probabilités, un ensemble pour les probabilités d'appartenir aux phrases de Chirac et un ensemble pour les probabilités d'appartenir aux phrases de Mitterrand.

En sortie du classifieur on souhaite obtenir une seule valeur de probabilité pour chacun des ensembles de phrases de Chirac et de Mitterrand, il est donc nécessaire de combiner les probabilités, pour cela une des techniques les plus éprouvées a été mise au point par R.A. Fisher. If we have a set probabilities, $p1$, $p2$, ..., pn, we can do the following. First, calculate -2ln $p1$ * $p2$ * ... * pn. Then, consider the result to have a chi-square distribution with $2n$ degrees of freedom, and use a chi-square table to compute the probability of getting a result as extreme, or more extreme, than the one calculated. This ``combined" probability meaningfully summarizes all the individual probabilities. Si nous avons un ensemble de probabilités, $p1$, $p2$, ..., pn, nous pouvons effectuer ce qui suit. Premièrement calculer 2ln $p1$ * $p2$ * ... * pn. Puis, nous considérons que le résultat suit une distribution du chi-deux avec $2n$ degrés de liberté, et on utilise une table du chi-deux pour calculer la probabilité d'obtenir un résultat aussi élevé, ou plus élevé, que celui calculé. Cette probabilité « combinée » résume significativement les probabilités individuelles.

2.6 Sélection des ensembles de travail

Le corpus d'entraînement, donné pour le challenge, est divisé en deux parties :

- une pour la création automatique du classifieur, c'est-à-dire la table des probabilités des *mots,*

- la seconde pour tester l'apprentissage en calculant les probabilités d'appartenance des phrases aux discours de Chirac et aux discours de Mitterrand.

Les deux parties ont été obtenues de deux manières différentes pour vérifier que les résultats restent stables. D'abord en coupant le corpus en deux parties égales au milieu de l'ensemble des phrases, puis en utilisant les discours de numéro pair pour l'apprentissage et les discours de numéro impair pour le test. Les résultats obtenus sur ces deux ensembles de travail ont été similaires, nous ne présenterons par la suite que les résultats obtenus sur le second ensemble de travail.

2.7 Sélection des groupes lexicaux

Nous avons utilisé trois ensembles de groupes lexicaux, afin d'essayer de trouver un ensemble optimal ou dans le but de les combiner.

4. D'abord l'ensemble le plus évident, chaque phrase est divisée en mots, un mot est une suite de lettres accentuées ou non et de chiffres. Un mot a une longueur d'une lettre ou plus. Chaque mot est un attribut caractérisant une phrase. Ce premier ensemble d'attributs sera A_{mots}.

5. Ensuite, nous avons choisi de découper le texte en tranche de n caractères non glissants, en basant l'idée sur l'article de (Brunet, 2003), qui montre qu'il n'est pas nécessaire de faire un découpage lexical parfait pour comparer des textes. Nous avons effectué plusieurs essais pour trouver un maximum pour le F-score à 7 caractères. Un des avantages de cette approche est de prendre en compte la ponctuation, qui peut varier entre deux auteurs voire entre deux types de discours. Une expérimentation pourrait être faite en faisant glisser la fenêtre de caractères.

6. Nous avons ensuite généralisé la première méthode, en l'appliquant pour des groupes lexicaux de 1 à 5 mots. Dans les groupes lexicaux de 1 mot, les mots de longueur inférieure à 3 ne sont pas pris. Nous avons effectué plusieurs essais en faisant varier le nombre maximumde mots dans un groupe et nous avons obtenu un F-score maximum pour 4 mots. C'est cette dernière méthode de lemmatisation des phrases, qui a été utilisée pour fournir les résultats du challenge, car bien qu'elle ne donnait pas un F-score très différent en sortie directe du classifieur Bayesien elle s'avérait être la plus performante après la détection des blocs contigus.

Pour calculer les F-scores en sortie du classifieur Bayesien, nous avons attribué une phrase à Mitterrand si et seulement si la probabilité que la phrase appartienne à l'ensemble des phrases de Mitterrand est supérieure à la probabilité que la phrase appartienne à l'ensemble des phrases de Chirac.

2.8 Résultats

Le programme Python Reverend de (Bakhtiar, Delord, 2003), après correction de la formule de combinaison des probabilités individuelles pour être confrome à celle exposée par (Robinson, 2003) et explicitée dans la section 2.2, a été utilisé pour créer le classifieur Bayesien et effectuer la classification.

Les résultats du point de vue du F-Score sont assez similaires de 0.35 sur les mots simples, 0.40 pour les heptagrammes et 0.42 pour les groupes lexicaux de 4 mots. Il faut remarquer quand même que sur l'ensemble d'apprentissage, les heptagrammes obtiennent un score de 0.70 et atteint 0.99 si on retire les phrases mal classées, qui représentent 10% de l'ensemble d'apprentissage, par contre ce sur-apprentissage n'améliore pas le F-score sur l'ensemble de test, il a même tendance à le faire descendre légèrement.

3 Détection et regroupement des blocs contigus

3.1 Méthode

Nous avons créé un programme qui prend en entrée un texte, qui est une suite de phrases dans l'ordre du texte, chaque phrase ayant un et un seul attribut auteur déterminé par une prise de décision après passage dans le classifieur Bayesien ou par une autre méthode.

Ce programme cherche la plus grande suite de phrases attribuées à Mitterrand. Comme il est possible d'avoir pris une mauvaise décision précédemment nous autorisons que des phrases de Chirac soient incluses dans le bloc de phrases de Mitterand. Une phrase de Chirac est autorisée si elle est précédé et suivie par une phrase de Mitterrand, nous avons également essayé avec deux phrases et trois phrases de Mitterrand, dans nos test l'optimum pour le F-score était pour deux phrases avant et après celle de Chirac, mais les résultats sont très voisins. Nous avons également tenté d'accepter deux phrases de Chirac au milieu de celles de Mitterand, mais les résultats se sont dégradés.

Afin de limiter l'apparition de petits blocs de textes, nous n'acceptons finalement un bloc de texte, comme appartenant à Mitterand que si ce dernier a une taille minimale, nous avons fait varier la taille du bloc de 4 à 12 pour trouver un F-score maximal (0,72) pour des blocs de taille 8 sur le jeu d'essai.

Ce programme force également l'attribution des deux premières phrases du texte et des deux dernières à Chirac.

3.2 Seuillage des phrases de Mitterrand

Pour améliorer le score on crée un nouvel indice et un nouveau seuil pour décider qu'une phrase appartienne à l'ensemble des phrases de Mitterrand. Ces calculs sont placés entre la sortie du filtre Bayesien et lla détection des blocs contigus.

Le nouvel indice est égal à 2 fois la probabilité qu'une phrase soit de Mitterrand ajouté à un moins la probabilité que la phrase appartienne aux phrases de Chirac.

Le seuil est appris pour obtenir le meilleur F-score sur l'ensemble de tests à la sortie de la détection des blocs contigus. Un seuil de 1,7 permet d'obtenir le meilleur F-score qui est de 0,80, donc augmentant de'environ 10% le score obtenu sur les blocs contigus détectés directement en sortie du filtre Bayesien.

3.3 Renforcement avec international

Nous avons tenté d'améliorer la détection des blocs de phrases de Mitterrand en travaillant sur un thème : l'international.

Le thème international est défini par une liste de groupes lexicaux relatifs à des discours internationaux, obtenue de manière semi-automatique à partir de noms de pays, de noms d'habitants et de formule de politesse (cher Président, votre Altesse, etc.).

Pour chaque texte, si les deux premières lignes et les deux dernières lignes contiennent des groupes lexicaux parlant de l'international, on suppose que le discours de Chirac est international.

Alors on a des phrases de Mitterrand, quand on trouve des mots se rapportant à des thèmes nationaux.

En phase de chaque phrase, on met donc une probabilité 1 ou 0 d'être une phrase de Mitterrand. On combine le résultat obtenu pour étendre les blocs contigus détectés précédemment.

Cette technique n'a pas permis d'améliorer le F-score, mais ne l'a pas fait baisser.

4 Détection de blocs par nombre de mots

Après avoir fait un calcul sur la moyenne de nombre de mots par phrase de l'un et de l'autre des présidents de la République étudiés, sur le corpus entier, on s'aperçoit que Chirac avait une moyenne de mots par phrase beaucoup plus faible que Mitterrand. Environ 23 pour Chirac alors qu'elle est de près de 29 pour Mitterrand. Nous avons donc eu l'idée de tester un indicateur qui serait le nombre de mots par phrase.

Cependant, considérant qu'il pouvait toujours y avoir une phrase de peu de mots, mais que cette statistique était plus correcte pour un certain nombre de phrases, et sachant que les phrases de Mitterrand étaient regroupés en un bloc à l'intérieur des discours de Chirac, il est apparu nécessaire de travailler sur des fenêtres glissantes.

Nous avons donc utilisé des fenêtres de 7 phrases en glissant par pas de 3 phrases.

Nous effectuons un parcours des fenêtres comme suit : la 1ère fenêtre est celle commençant à la ligne 1, la 2ème est celle commençant à la ligne (dernière ligne – taille fenêtre), la 3ème est celle commençant à la ligne 4 (1 + pas), la 4ème commence à la ligne (dernière ligne – taille fenêtre – pas), et ainsi de suite. On associe la valeur 0 aux 2 premières fenêtres, donc en fait la 1ère et la dernière dans l'ordre du discours.

Ensuite, pour la ième fenêtre calculée, on lui attribue pour valeur :

La somme des différences entre le nombre des mots moyen des phrases de la fenêtre courante, et le nombre de mots moyens des fenêtres calculées précédemment, mais uniquement celles qui lui sont proches, donc on va de 2 fenêtres en 2, et on diminue cette somme par rapport à la distance des 2 fenêtres (plus une fenêtre est loin, moins elle doit avoir d'influence sur le calcul de la fenêtre courante).

On obtient donc des lignes comme celle-ci :

<100:1> 0 C C C C C C C
<100:33> 0 C C C C C C C
<100:4> 2.28571428571 C C C C C C C
<100:30> 1.0 C C C C C C C
<100:7> 2.42857142857 C C C C C M M
<100:27> 11.1428571429 M M C C C C C
<100:10> 2.71428571429 C C M M M M M
<100:24> 47.5714285714 M M M M M C C
<100:13> 14.2857142857 M M M M M M M
<100:21> 58.7142857143 M M M M M M M
<100:16> 18.0 M M M M M M M
<100:18> 13.4285714286 M M M M M M M

<100:19> 24.1428571429 M M M M M M M
<100:15> 32.2857142857 M M M M M M M

Dans le discours 100, les fenêtres commençant à la ligne n°1, 33, 4, 30, dans l'ordre de calcul. Les 7 caractères disent si c'est Chirac ou Mitterrand qui parle à chaque ligne.

Ici, plus la valeur est grande, plus il y a eu rupture de nombre de mots par phrase, donc on est censé détecter des coupures de blocs avec cette méthode, et aussi comme l'heuristique était que Mitterrand avait plus de mots par phrase que Chirac, plus la valeur est grande plus il devrait y avoir de chances que Mitterrand parle.

Ensuite, on on construit un nouveau fichier :

Pour chaque ligne de chaque discours, on fait la somme des valeurs données par le fichier précédent, par exemple pour la ligne 12, on a fait la somme des fenêtres commençant aux lignes 7 et 10 car la ligne 12 est comprise dans les fenêtres [7,14] et [10,17].

Cela donne des valeurs pour chaque ligne de chaque discours.

Exemple :

<3:73:M> 476.285714286
<3:70:M> 405.428571428
<3:76:M> 392.428571429
<3:67:M> 373.714285714
<3:90:C> 344.285714286
<3:91:C> 344.285714286
<3:79:M> 333.857142858
<3:88:M> 325.142857143
<3:74:M> 305.142857143
<3:75:M> 305.142857143
<3:68:M> 296.714285714
<3:69:M> 296.714285714
<3:93:C> 292.714285714
<3:77:M> 283.714285715
<3:78:M> 283.714285715
<3:71:M> 279.857142857
…

On trie par valeur décroissante, ce qui signifie d'après l'heuristique que les lignes parlées par Mitterrand devraient se retrouver en haut de la liste.

Par la suite, on obtient avec un passage sur ce fichier, un nouveau fichier, qui se présente comme suit :

3 [54:86] 27176.4162603
3 [87:101] 7403.68292573
3 [117:134] 3677.4696891
3 [135:152] 2338.71978797
3 [111:116] 1626.0858952
3 [10:27] 1498.41437507
3 [37:45] 1496.41955674

3 [153:174] 1409.26083185
3 [104:110] 1328.16439518
3 [46:51] 846.077319479
3 [28:36] 649.31079412
3 [102:103] 317.477678571
3 [52:53] 205.320408163
…

Cela représente donc des blocs. En effet, on a essayé de reconstruire des blocs à partir des valeurs précédentes. On met la première ligne dans un bloc, puis si la deuxième ligne est assez proche, on la rajoute au bloc, sinon on crée un nouveau bloc, et ainsi de suite. Donc on obtient des petits blocs de phrases avec la somme des différentes lignes dedans. Ainsi, plus un bloc a une grande valeur, plus les éléments dedans avaient déjà une grande valeur, et plus ils étaient regroupés dans le précédent fichier.

Voici les scores que nous avons obtenu en prenant des blocs sur ce fichier :

Les 3 premiers blocs de chaque discours pour fenêtre : Fscore(4270, 24143, 7523) = 0.270

Les 2 premiers blocs de chaque discours pour fenêtre : Fscore(3571, 17471, 7523) = 0.286

Le premier bloc de chaque discours pour fenêtre : Fscore(2458, 9754, 7523) = 0.285

Cette méthode a été combinée à la détection de blocs contigus, mais une fois encore le F-score n'a pas été amélioré.

Remerciements

Langage de programmation Python et son auteur Guido van Rossum, sans lequel nous n'aurions pas pu développer les programmes de ce challenge aussi rapidement.

Martine Cadot pour ses précieux conseils sur l'analyse et la fouille de données et nous avoir fait connaître ce challenge.

Références

ROBINSON G. (2003), A Statistical Approach to the Spam Problem, *Linux Journal*, http://www.linuxjournal.com/article/6467.

BAKHTIAR A., DELORD C. (2003), Divmod Reverend, http://www.divmod.org/Home/Projects/Reverend.

BRUNET E., (2003), PEUT-ON MESURER LA DISTANCE ENTRE DEUX TEXTES ?, CORPUS, NUMERO 2 LA DISTANCE INTERTEXTUELLE -décembre 2003, http://revel.unice.fr/corpus/document.html?id=30.