
classification supervisée de brevets : d'un jeu d'essai au cas réel

J.G. Beney* — **C.H.A. Koster****

* *Département Informatique, INSA de Lyon*
7 Av. Jean Capelle
F69621 Villeurbanne Cedex

** *Université de Nimègue (KUN),*
Toernooiveld 1, NL6525ED Nijmegen, Pays-Bas
jean@if.insa-lyon.fr; kees@cs.kun.nl

RÉSUMÉ. Nous présentons les différences constatées entre les résultats de deux séries d'expériences de classification de documents avec Winnow : d'abord sur un choix de documents typiques d'un petit nombre de classes, puis sur toutes les classes existantes, avec tous les documents. On constate que les valeurs optima des paramètres sur le sous-ensemble ne le sont plus pour l'ensemble complet. D'autre part, les résultats sont nettement moins bons sur l'ensemble complet principalement à cause des documents multi-classes.

ABSTRACT. We present the differences between the results of two series of experiment of supervised document classification with Winnow: first on a selection of typical documents in a small number of classes; then on all the documents in all the classes. We see that the optimal parameter values for the subset are not optimal for the whole set. And the results on the whole set are really worse mainly because of the multi-class documents.

MOTS-CLÉS : Winnow, Classification supervisée de documents.

KEYWORDS: Winnow, supervised document classification.

1. Introduction

L'office européen des brevets (EPO à Rijswijk, Pays-Bas) doit traiter chaque jour plus de 400 demandes qui sont d'abord examinées par une ou plusieurs de leurs 549 équipes de spécialistes. Dans le but d'automatiser l'aiguillage des demandes vers les bonnes équipes, l'EPO a proposé à plusieurs équipes de recherche d'appliquer leurs méthodes de classification automatique à ce problème.

Il a été réalisé 2 campagnes d'expériences : EPO1 (en 1999) a porté sur 16 catégories techniques bien séparées, donc avec des documents mono-classes (1000 documents par groupe) ; EPO2 (en 2000) sur leurs 44 directoires sans sélection des documents (2000 documents par repertoire sélectionnés uniquement sur leur date de dépôt). L'idée sous-jacente était, dans un premier temps, de minimiser les efforts des chercheurs en leur fournissant des données propres (documents bien classés) puis de permettre une expérimentation en vraie grandeur.

Outre les documents dont la classification nous était connue et qui ont été utilisés dans les expériences décrites dans cet article, un deuxième jeu de document nous avait été fourni par l'EPO pour un test à l'aveugle [KOS01]. Sur ce jeu, l'EPO a trouvé une précision¹ de 96 % pour EPO1 et 72 % pour EPO2 [KRI02] ; ce dernier résultat est nettement trop faible, comparé aux 81 % de précision fournis par les personnes actuellement chargées de l'aiguillage.

Nous examinons ci-dessous les différences entre les deux ensembles de documents qui peuvent expliquer la grande différence dans la qualité de la classification.

1.1. travaux en rapport

Une revue d'ensemble des méthodes de classification de documents peut être trouvée, entre autres, dans [SEB02].

La classification des demandes de brevets est une application intéressante des techniques d'apprentissage, à cause du grand nombre de documents disponibles, de la longueur des textes (environ 5000 mots) et de la grande richesse du vocabulaire (550 000 mots différents dans EPO2).

Ces caractéristiques en font une tâche très différente d'autres applications à des textes courts où les classes sont caractérisées par peu de termes : Reuters dataset [APT94] ou les données utilisées par TREC. Dans ces applications Rocchio [ROC71] ou kNN sont généralement préférables [AUL01], alors que pour des textes plus riches, Winnow [DAG97] ou SVM [JOA98] sont connus pour être robustes lorsque les termes sont très nombreux.

1. EPO calcule la précision ainsi : chaque document est affecté à une seule classe et on compte le nombre d'affectations correctes.

[LAR99] décrit un système de recherche dans une base de brevets, mais sans donner d'estimation de la qualité des résultats. Son système combine les réseaux bayesiens (système Inquiry) pour la recherche de documents et kNN sur les résultats de Inquiry résultats pour la classification de nouveaux brevets.

L'EPO dispose d'ailleurs également déjà d'un système (EPOQUE) de recherche et de visualisation de brevets ([NUY00]); tandis que l'office japonais JPO réalise les préclassification avec le système OWAKE ([JAP00]) qui utilise une liste de mots préselectionnés (pas d'apprentissage).

2. L'expérimentation

2.1. l'outil LCS

Le programme LCS (pour Linguistic Classification System) développé à l'Université Catholique de Nimègue dans le cadre des projets européens DORO et PEKING, inclut plusieurs algorithmes de classification supervisée de documents, dont Winnow (voir [LIT88] pour la théorie et [DAG97] pour une présentation claire et pratique), associés à divers options de pondération des termes dont celles étudiées dans [SAL88] et de sélection a priori de ces termes [PET02]. LCS permet par ailleurs l'utilisation de syntagmes comme termes [KOS03].

2.2. Méthode Winnow

L'algorithme d'apprentissage *Winnow balancé* maintient deux poids $W_{t,c}^+$ et $W_{t,c}^-$ pour chaque terme, pour chaque classe. Chaque document est représenté par un vecteur donnant l'importance $S_{t,d}$ de chaque terme dans le document². L'appartenance d'un document à une classe est caractérisée par son score dans la classe :

$$Sc_{d,c} = \sum_t (W_{t,c}^+ - W_{t,c}^-) S_{t,d}$$

qui est comparé à un seuil θ généralement égal à 1 :

$$Sc_{d,c} \geq \theta : \text{accepté}, \quad Sc_{d,c} < \theta : \text{rejeté.}$$

à chaque itération (dont le nombre est fixé), l'algorithme considère successivement tous les documents et détermine leur appartenance à chaque classe. En cas d'erreur, les poids des termes appartenant au document sont modifiés en utilisant deux constantes : $\alpha > 1$ et $\beta < 1$

$$\text{document pertinent rejeté : promotion } W_{t,c}^+ = \alpha \cdot W_{t,c}^+, \quad W_{t,c}^- = \beta \cdot W_{t,c}^-$$

$$\text{document non pertinent accepté : démotivation } W_{t,c}^+ = \beta \cdot W_{t,c}^+, \quad W_{t,c}^- = \alpha \cdot W_{t,c}^-$$

2. Pour calculer l'importance d'un terme, nous utilisons généralement la formule LTC.

Notre implémentation intègre le *seuil épais* qui tente de mieux séparer les documents pertinents et non pertinents :

soit $\theta^+ > \theta^-$: un document non pertinent dont $S_{c_{d,c}} \geq \theta^-$ est accepté, et un document pertinent dont $S_{c_{d,c}} < \theta^+$ est rejeté. Ainsi les documents dont le score est entre θ^- et θ^+ continuent à entraîner des promotions et des démotions même si leur score est déjà du bon côté du seuil θ .

2.3. Mesure de la qualité

Nous utiliserons la mesure $F(1)$ (notée F1) qui donne une importance égale à la précision (P) et au rappel (R). La mesure des retombées (Fa) est également caractéristique de la qualité d'un classifieur.

Soit, pour une classe donnée, RS le nombre de documents pertinents et trouvés par le classifieur, RNS le nombre de documents pertinents qui ne sont pas trouvés, NRS le nombre de documents retenus bien non pertinents et $NRNS$ le nombre de documents non pertinents et écartés.

$$P = \frac{RS}{RS+NRNS}, \quad R = \frac{RS}{RS+RNS}, \quad F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}}, \quad Fa = \frac{NRS}{NRS+NRNS}$$

La qualité sur un ensemble de classes est obtenue par moyenne microscopique : les nombres $RS, RNS, NRS, NRNS$ sont sommés sur toutes les classes avant de calculer F1.

La méthode : Les mesures ont été obtenue par cross-validation : les documents sont divisés aléatoirement en 4 groupes. Tour à tour chaque groupe sert à l'apprentissage, les 3 autres au test. Les valeurs données sont les moyennes des 4 valeurs obtenues.

2.4. Les corpus

Les demandes de brevets doivent être aiguillées vers une ou plusieurs parmi les 549 équipes relevant de 44 *directoires* divisés en 16 *groupes*. Ces documents peuvent également être classés suivant les codes ECLA (European CLAssification) : environ 1000 codes couvrant 624 sous-classes .

Le premier test a été effectué avec un jeu d'essai basé sur les groupes de directoires et composé de documents choisis (chacun n'appartient qu'à un seul groupe). Le deuxième test a porté sur tous les directoires et incluait des documents qui relèvent de plusieurs classes (jusqu'à 7). Les brevets ont deux formes : la description complète et un résumé. Nous ne parlerons ici que du classement des descriptions complètes.

Voici les principales caractéristiques de ces deux ensembles de documents.

	classes	doc./classe	tot. docs.	vocabulaire
EPO1	16	1000	16000	116103
EPO2	44	2000	68418	556995

Les demandes complètes sont assez longues : environ 5000 mots ; il y a peu de fautes d'orthographe mais une partie de l'information peut être incluse dans des tables, graphes.

2.5. Les résultats

Voici les résultats obtenus (sur les documents d'apprentissage et les documents de test) dans les 2 cas après optimisation des paramètres de Winnow pour les documents de test.

	α	β	θ^+	θ^-	nb. iters.	F1-train	F1-test
EPO1	1.5	0.59	5	0.95	5	99.63	95.75
EPO2	1.03	0.97	2	0.5	3	82.10	62.67

3. Influence du nombre de classes

Le nombre de classes n'a pas une influence directe sur la qualité de la classification. En effet, la méthode réalise son apprentissage pour chaque classe séparément en séparant les documents pertinents et les documents non pertinents. Si l'on cherche à séparer une classe du *reste du monde*, le résultat est très proche³.

Le facteur important est le ratio entre le nombre d'exemples pertinents et le nombre d'exemples non pertinents (contre-exemples).

En supprimant aléatoirement des contre-exemples d'une classe donnée, on obtient la figure 1 qui montre la distribution des scores des documents pour cette classe lorsqu'il y a autant de contre-exemples que d'exemples et lorsqu'il y a 10 fois plus de contre-exemples que d'exemples. Pour ce faire, les documents n'appartenant pas à la classe choisie (dir36) ont été mis dans une unique classe *Reste-du-monde*.

On voit que pour un même seuil, le nombre de document non pertinents acceptés (représenté par la surface sous la courbe, à droite du seuil) est bien plus important. Pourtant la *proportion* de documents non pertinents acceptés (retombées) reste la même.

Si comme dans notre cas chaque classe comporte le même nombre d'exemples, chaque classe supplémentaire augmente le nombre de contre-exemples et en conséquence diminue la précision pour des retombées identiques.

Un classifieur paraît donc moins bon lorsqu'il est en présence de plus de classes.

3. Dans ce cas, on ne peut pas utiliser le fait que la majorité des documents appartiennent à 1 ou 2 classes Cf. 5

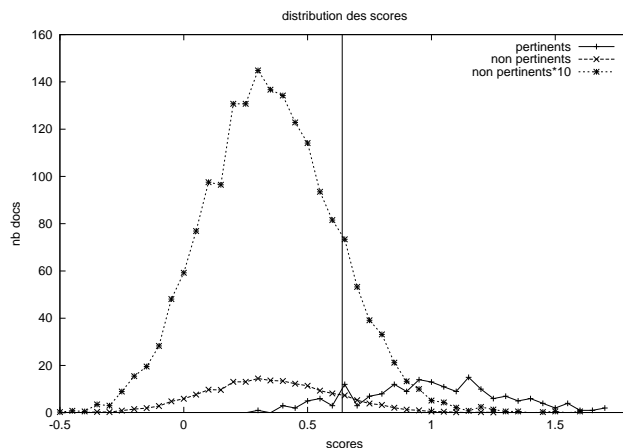


Figure 1. *distribution des scores à parité et pour un ratio 1/10*

En pratique, on augmentera le seuil pour que la précision ne soit pas catastrophiquement basse, ce qui diminuera le rappel. Dans LCS, ceci est réalisé lors de la phase de calcul du seuil optimum pour chaque classe en vue de l'optimisation de F1.

4. Influence du nombre de documents

4.1. qualité de l'apprentissage

Pour un même ensemble de test, l'algorithme apprend mieux s'il y a plus de documents pour l'apprentissage comme le montre la figure 2 (obtenue sur EPO2 sans cross-validation).

4.2. optimisation des paramètres

5 paramètres ont une influence sur le fonctionnement de Winnow dans notre implémentation : $\alpha, \beta, \theta^+, \theta^-$ et le nombre d'itérations. Plusieurs expériences ayant montré que les meilleurs résultats sont obtenus pour $\alpha + \beta \simeq 2$, nous nous ramenons à faire varier 4 paramètres.

Sur des ensembles de documents plus petits, il a été possible de faire une recherche *systematique* de l'optimum avec une dizaine de valeurs pour chacun des paramètres et cross-validation. Dans le cas de EPO2, cela prendrait plus de 4 mois.

Nous avons donc diminué le nombre de valeurs, puis affiné avec quelques valeurs supplémentaires autour du premier optimum trouvé.

$$\theta^+ = 1.0, 1.5, 2.0, 3.0; \quad \theta^- = 1.0, 0.7, 0.3, -0.3;$$

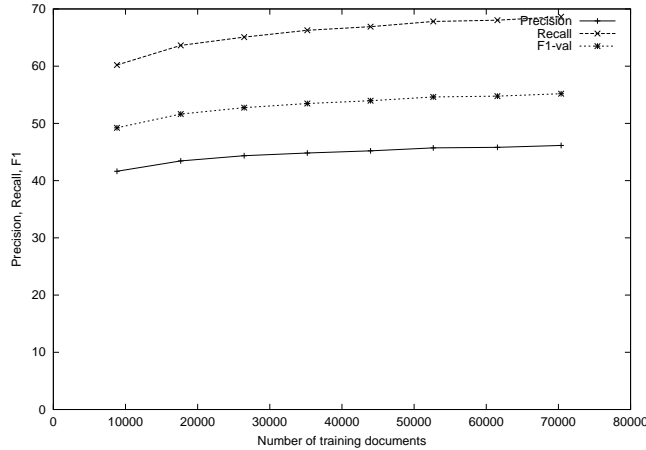


Figure 2. vitesse d'apprentissage

$\alpha = 1.1, 1.2, 1.5$; $\beta = 2 - \alpha$; 1, 3 or 5 Winnow itérations.

Et, pour aller encore plus vite, nous avons expérimenté avec seulement 1/10 des documents :

	α	β	θ^+	θ^-	nb. iters.	F1-test
valeurs standard	1.1	0.9	1.1	0.9	5	46.97
optimum grossier	1.1	0.9	1.5	0.7	3	49.71
optimum fin	1.1	0.9	1.25	0.4	3	50.22

Avec tous les documents, en partant de cet optimum, on trouve assez rapidement une meilleure solution :

	α	β	θ^+	θ^-	nb. iters.	F1-test
optimum précédent	1.1	0.9	1.25	0.4	3	61.8
optimum	1.04	0.96	1.25	0.4	3	62.7

On constate que les valeurs des facteurs de promotion ont une assez grande influence sur le résultat : avec plus de documents, il est recommandé d'apprendre *plus lentement*. Par contre, l'épaisseur du seuil n'a que peu d'influence sur la qualité de la classification, pourvu qu'il soit *assez épais*.

5. Mono/multi classification

LCS accepte 2 paramètres *minranks* et *maxranks* qui indiquent à combien de classes peut appartenir un document : un document sera toujours accepté dans les *minranks* classes pour lesquelles il a le meilleur score, et peut-être dans d'autres classes (*maxranks* au maximum) s'il obtient un score supérieur au seuil de ces classes.

L'apprentissage se faisant indépendamment pour chaque classe, ces paramètres ne sont utilisés que lors de la phase de test, où ils introduisent un lien entre les classes.

Pour le jeu d'essai EPO1, nous savons que les documents appartiennent à une seule classe ; cette information ($minranks = maxranks = 1$) améliore les résultats sur l'ensemble de test comme sur l'ensemble d'apprentissage.

<i>min</i>	<i>max</i>	test	train
0	3	94.86	99.49
0	1	96.81	99.79
1	3	95.45	99.56
1	1	97.46	99.83

Tableau 1. *minranks, maxranks sur EPO1*

Pour EPO2, nous savons que les documents connus appartiennent au moins à 1 classe et au plus à 7 avec la distribution suivante. Cette information est donc difficilement exploitable comme le montre la table. L'utilisation du fait que la majorité des documents appartiennent à 1 ou 2 classes améliore très légèrement le résultat : la mesure F1 varie de 1.5 % entre les cas extrêmes, alors qu'elle variait de 2.6 % pour EPO1F.

		<i>min</i>	<i>max</i>	test P	R	F1
1 classe	51981	0	7	64.91	68.20	66.51
2 classes	13745	0	4	64.91	68.20	66.52
3	2323	0	3	65.07	68.17	66.58
4	306	0	2	66.85	67.04	66.95
5	46	0	1	78.67	56.22	65.58
6	13	1	7	63.08	70.50	66.58
7	4	1	4	63.09	70.50	66.59
		1	3	63.24	70.47	66.66
		1	2	64.86	69.33	67.02
		1	1	75.08	58.22	65.58

Tableau 2. *minranks, maxranks sur EPO2*

Ces résultats ont été obtenus avec un partage 80 % entraînement / 20 % test, ce qui explique les valeurs de F1 un peu meilleurs que précédemment.

Nous avons reconstruit un jeu d'essai mono à partir du multi en tirant 16 classes au hasard et en ne gardant que les documents qui n'appartiennent qu'à une seule classe.

classes	docs	test	train
44	tous	58.40	90.34
44	mono	64.77	96.36
16	tous	67.87	96.57
16	mono	83.28	99.04

On voit par là que 16 directoires tirés au hasard sont moins bien séparés que les 16 catégories de EPO1 et que les plus grandes améliorations sont obtenues lorsque l'on supprime les documents multi-classes. Et l'entraînement fait avec les documents mono n'aide pas vraiment à classer les documents multi :

44 classes, entraînement mono, test multi : F1 = 60.12 %

6. Conclusion

Le nombre de classes, (donc le nombre de contre-exemples à rejeter) est une donnée incontournable. Lorsque ce nombre est élevé, même si l'on rejette une très grande proportion des contre-exemples, les performances visibles d'un algorithme de classification (mesurées par F1) peuvent être médiocres. Mais notre étude montre qu'une petite amélioration (diminution des retombées) entraînera une importante augmentation de la précision.

Nous constatons que l'optimisation des paramètres doit prendre en compte tous les documents disponibles même si leur grand nombre empêche de faire une recherche exhaustive. Une approche pas à pas peut être menée en quelques jours, ce qui n'est pas rédhibitoire, vu que cette optimisation n'est faite qu'une seule fois avant l'exploitation du classifieur.

Ces éléments nous ont empêché de réutiliser pour EPO2 les résultats obtenus sur EPO1. Mais le point le plus décevant, et sur lequel doit porter nos efforts, est la qualité de la classification des documents multi-classes.

Parallèlement, nous attaquons la classification selon les équipes EPO et les codes ECLA. Ici, pour chaque classe, le nombre de contre-exemples est encore plus grand, tandis que le nombre d'exemples pour l'apprentissage est bien plus petit.

Nous continuons également à explorer des techniques linguistiques qui, pour l'instant n'ont pas donné de résultats spectaculaires : extraction des syntagmes [KOS03] ou utilisation d'EuroWordNet pour les synonymes et les familles de mots.

Par ailleurs, l'utilisation de la chronologie des documents [ARA00] peut apporter une amélioration dans la mesure où le vocabulaire des inventions évolue nécessairement.

7. Bibliographie

- [APT94] APTÉ, C. AND DAMERAU, F., Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems* 12(3), 1994, pp233-251, 1994.
- [ARA00] A. ARAMPATZIS, J. BENEY, C.H.A. KOSTER, TH.P. VAN DER WEIDE, KUN on the TREC-9 Filtering Track : Incrementality, Decay, and Threshold Optimization for Adaptive Filtering Systems. *The Ninth Text REtrieval Conference (TREC-9)*, Gaithersburg, Maryland, November 13-16, 2000.

- [AUL01] THOMAS AULT AND YIMING YANG, kNN, Rocchio and Metrics for Information Filtering at TREC-10, *TREC-10 Notes*, Nov. 2001.
- [DAG97] I. DAGAN, Y. KAROV, D. ROTH, Mistake-Driven Learning in Text Categorization. *Proceedings of the Second Conference on Empirical Methods in NLP*, 1997, pp. 55-63.
- [JAP00] DELEGATION OF JAPAN, OWAKE system - primary automatic classification. *Section 59 of WIPO Report IPC/CE/29/11, on the 29th session of the Committee of experts of the IPC union, 13-27 March, 2000*.
- [JOA98] T. JOACHIMS, Text categorization with support vector machines : Learning with many relevant features. *European Conference on Machine Learning (ECML-98)*, Springer Verlag, 1998, pp. 137-142.
- [KOS01] C.H.A. KOSTER, M. SEUTTER AND J. BENEY, Classifying Patent Applications with Winnow, *Proceedings Benelearn 2001*, Antwerpen, 8pp. <http://cnts.uia.ac.be/benelearn2001/>
- [KOS03] C.H.A. KOSTER AND M. SEUTTER), Taming Wild Phrases, To Appear in *Proceedings ECIR 2003*.
- [KRI02] M. KRIER, F. ZACCÀ, Automatic categorisation applications at the European patent office. *World Patent Information*, 24, 2002, pp. 187-196.
- [LAR99] LEAH S. LARKEY, A patent search and classification system, *Proceedings of DL-99, 4th ACM Conference on Digital Libraries*, ACM Press, New York, 1999, pp. 179-187.
- [LIT88] N. LITTLESTONE, Learning quickly when irrelevant attributes abound : A new linear-threshold algorithm. *Machine Learning*, 2, 1988, pp. 285-318.
- [NUY00] A. NUYTS, Search and viewer tools at the European Patent Office. *Proceedings of the 2000 International Chemical Information Conference, Annecy, 22-25 October, 2000*, pp. 47-56.
- [PET02] C. PETERS AND C.H.A. KOSTER, Uncertainty-based Noise Reduction and Term Selection in Text Categorization, *Proceedings 24th BCS-IRSG European Colloquium on IR Research*, Springer LNCS 2291, 2002, pp 248-267.
- [ROC71] J.J. ROCCHIO, Relevance feedback in Information Retrieval, In : Salton, G. (ed.), *The Smart Retrieval system - experiments in automatic document processing*, Prentice - Hall, Englewood Cliffs, NJ, 1971, pp 313-323.
- [SAL88] G. SALTON AND C. BUCKLEY, Relevance feedback in Information Retrieval, *Information Processing and Management*, Vol.24 N°5, 1988, pp. 513-523.
- [SEB02] F. SEBASTIANI, Machine learning in automated text categorization, *ACM Computing Surveys*, Vol 34 no 1, 2002, pp. 1-47.