Réseaux de neurones pour le traitement automatique du langage: conception et réalisatin de filtres d'informations

Mathieu Stricker

Chapitre 1 Introduction

1.1 Problématique générale

En raison de l'augmentation constante du volume d'information accessible électroniquement, la conception et la mise en œuvre d'outils efficaces, permettant notamment à l'utilisateur de n'avoir accès qu'à l'information qu'il juge pertinente, devient une nécessité absolue. Comme la plupart de ces outils sont destinés à être utilisés dans un cadre professionnel, les exigences de fiabilité et de convivialité sont très importantes ; les problèmes à résoudre pour satisfaire ces exigences sont nombreux et difficiles. L'accès à cette information pertinente peut se faire en fournissant à un utilisateur des documents pertinents ou en lui proposant des passages de documents pertinents (ou des réponses à des questions). Le premier cas relève du domaine de la *recherche de textes* et le second du domaine de l'*extraction d'informations*.

C'est dans le domaine très actif de la recherche de textes que se place le présent travail, réalisé dans le cadre d'une collaboration entre Informatique CDC, filiale de la Caisse des Dépôts et Consignations, et le Laboratoire d'Électronique de l'ESPCI.

1.2 La recherche de textes

Le domaine de la recherche de textes se divise en deux grandes disciplines :

- la recherche d'informations (également appelée recherche *ad hoc*)
- la catégorisation de textes.

La recherche d'informations consiste à trouver, dans une importante base de documents, les documents pertinents correspondant à des requêtes *ad hoc* (par mots clefs) ou posées en langage naturel. L'utilisation de moteurs de recherches sur le web et la recherche informatisée de documents dans un fonds bibliothécaire sont deux exemples d'applications de ce domaine.

La recherche d'informations est généralement effectuée en indexant préalablement tous les documents de la base selon les mots qu'ils contiennent ; la recherche consiste à trouver, le plus rapidement possible, les documents ayant des mots communs avec la requête de l'utilisateur. On montrera au chapitre 2 comment certaines méthodes modifient automatiquement la requête initiale pour améliorer le résultat de la recherche.

La catégorisation de textes, appelée également filtrage, consiste à trouver, dans un flux de documents (comme un fil de dépêches d'agence de presse), ceux qui sont relatifs à un sujet défini par avance. L'une des applications consiste à fournir à un utilisateur, en temps réel, toutes les informations importantes pour l'exercice de son métier. Dans ce cas, l'utilisateur n'exprime pas son intérêt par une requête, mais par un ensemble de documents pertinents. Cet ensemble de documents pertinents définit ce que l'on appelle, dans la suite de ce mémoire, un *thème* ou une *catégorie*. Pour un thème donné, la catégorisation consiste donc à résoudre un problème de classification supervisée à deux classes ; celui-ci peut être résolu, entre autres, par des méthodes à base d'apprentissage numérique comme les réseaux de neurones, les arbres de décision, les réseaux bayesiens, les machines à vecteurs supports, ou les modèles de Markov cachés.

La distinction entre la recherche d'informations et la catégorisation de textes peut être schématisée de la manière suivante : dans le premier cas, la base de documents est fixe et l'interrogation est variable, alors que, dans le deuxième cas, la source de documents est variable et l'interrogation est fixe.

Dans la pratique, la catégorisation de textes bénéficie de deux facilités considérables par rapport à la recherche d'information : la stabilité dans le temps de la thématique filtrée et la "faible" quantité de documents à traiter dans le temps. La stabilité de la thématique laisse le temps de construire des modèles sophistiqués permettant de rechercher la façon dont l'information est codée dans un texte. Le fait de filtrer un à un les documents au fil de leur arrivée, au lieu de s'attaquer à une base importante de documents, soumet le système à une contrainte de performance plus faible, et rend possible l'utilisation de modèles plus complexes.

1.3 L'extraction d'informations

L'extraction d'information cherche à analyser de manière précise le contenu d'un document, contrairement à la recherche de texte qui étudie sa thématique générale. Il s'agit donc d'une tâche qui ne peut être accomplie qu'après une sélection préalable des documents, et qui est considérée comme plus ardue que la catégorisation de textes. Pendant plusieurs années, les modèles d'extraction d'information ont été évalués grâce à la conférence Message

Understanding Conference (MUC) [Muc7, 1999]. Pour cette compétition, les participants doivent proposer des systèmes qui remplissent automatiquement des formulaires (ou patrons). Par exemple, pour les textes traitant des changements de poste au sein des grandes entreprises, il faut compléter automatiquement les champs suivants : nom, date d'arrivée, date de départ, changement, …. .

Beaucoup de systèmes d'extraction d'informations reposent sur l'utilisation de méthodes issues du traitement naturel du langage ; par exemple, [Vichot *et al.*, 1999] proposent une application, opérationnelle au sein de la Caisse des Dépôts, qui permet de visualiser simultanément une dépêche sur des prises de participations et le graphe de participations des sociétés qui y sont mentionnées.

Des méthodes d'apprentissage ont également été utilisées pour cette tâche, le lecteur intéressé pourra trouver des informations sur ces systèmes dans [Zaragoza, 1999] ainsi que des approches utilisant des réseaux de neurones ou des modèles de Markov cachés.

1.4 Les sources d'informations

Le filtrage de documents présuppose une source de documents, c'est-à-dire un canal par lequel sont délivrés de nouveaux documents au fil du temps.

L'avènement des documents numérisés, avec les traitements de textes puis l'Internet, a vu une croissance très rapide du nombre de sources de documents. Nous en mentionnons quatre pour leur importance :

- les agences de presse,
- les *news groups*,
- les courriers électroniques,
- le web.

Ce qui distingue particulièrement les agences de presse des autres sources, c'est la qualité, l'homogénéité et la régularité éditoriale avec lesquelles elles sont produites. Les autres sources citées ne subissent pas les contraintes éditoriales traditionnelles.

Parmi les grandes agences de presse de dimensions internationales figurent Reuters[1], Associated Press[2] et l'Agence France-Presse[3](AFP).Toutes trois fournissent depuis longtemps les médias (journaux, radios, télévisions) en dépêches, délivrées en temps réel sur les téléscripteurs de leurs clients. Aujourd'hui, ces agences touchent une gamme beaucoup plus étendue de clients allant des *traders* aux services de communications des grands groupes ; le support de transmission privilégié actuellement est le protocole TCP/IP, via l'Internet et les grands navigateurs du marché (Internet Explorer et Netscape Navigator).
Dans notre application opérationnelle, présentée au chapitre 9, la source d'information est l'AFP.

Parallèlement à ces sources alimentées par des professionnels de l'information, l'essor de l'Internet a vu se développer les *news groups,* dans lesquels l'information n'est pas travaillée, vérifiée, certifiée, authentifiée selon les règles de l'art dans une salle de rédaction de journalistes appliquant des règles éditoriales. Au contraire, l'information y est produite par ceux qui veulent faire connaître leur expérience. Ces *news groups* sont le moyen, pour des utilisateurs ayant une communauté d'intérêts, de s'informer mutuellement et de débattre de questions extrêmement spécialisées au fil de l'actualité ou des marottes de ses membres. Les faiblesses de ce média sont l'inégale qualité des contributions, la résurgence cyclique de certaines thématiques, la prise de parole excessive et, en définitive, le nombre de messages que le *news group* peut engendrer chaque jour. Compte tenu, à la fois, de la gratuité, de l'indéniable intérêt, et de l'inorganisation de cette source d'informations, il n'est pas étonnant que de nombreux travaux de filtrage de documents s'y soient attaqués [Zarogoza, 1999].
Le nombre croissant de messages électroniques, et l'envoi de courriers publicitaires, peuvent nécessiter l'utilisation d'un système de filtrage. Par exemple, [Cohen, 1996] propose de classer automatiquement les courriers reçus dans des répertoires prédéfinis et [Sahami *et al.*, 1998] proposent de filtrer automatiquement les courriers indésirables (appelés souvent *spam*).

[1] http://www.reuters.com

[2] http://www.ap.org

[3] htpp://www.afp.com

Le web est évidemment une source importante de documents dont la richesse potentielle demande qu'ils soient traités. Il s'agit d'un domaine très porteur pour le filtrage ; on peut imaginer disposer d'un outil à qui l'on spécifie des sites de référence ainsi qu'une thématique de filtrage afin de détecter les nouvelles pages. Il s'agit là d'applications en dehors de notre champ d'investigation.

1.5 La conférence TREC

La conférence annuelle Text REtrieval Conference[1] est organisée chaque année sous l'égide du National Institute of Standards and Technology (NIST) sous le patronage de la DARPA ; elle est ouverte à toutes les équipes ayant préalablement participé à la compétition.

Elle offre un forum d'évaluation et de discussions pour la communauté scientifique qui se consacre au traitement automatique des textes en général, et au filtrage en particulier.

Un ensemble de tâches différentes est proposé aux différents participants qui soumettent des résultats à autant de tâches qu'ils le souhaitent. Certaines tâches font uniquement appel à des approches issues du traitement automatique du langage naturel et d'autres, comme la tâche de filtrage, nécessitent l'utilisation de méthodes à base de statistiques. Une description générale de la huitième édition de cette conférence (TREC-8) peut-être trouvée dans [Voorhees et Harman, 2000] ; cette huitième édition de la conférence a regroupé soixante six équipes différentes venant de seize pays différents.

Parmi les multiples tâches proposées dans cette compétition, la recherche *ad hoc* (recherche d'informations) était la tâche principale jusqu'à l'édition TREC-9. Cette tâche a notamment permis d'étiqueter une très grande quantité de textes pour un grand nombre de thèmes différents. Cet ensemble constitue un corpus de référence, qui peut être utilisé par la communauté scientifique pour comparer des méthodes d'apprentissage et les faire progresser.

La tâche de filtrage proposée à TREC se décompose en trois sous-tâches :

1. Le filtrage adaptatif (*adaptive filtering*) consiste à construire un premier modèle grâce à une requête formulée en langage naturel, puis à simuler un flux de documents. Le système peut tirer parti de la pertinence ou de la non-pertinence des documents sélectionnés pour s'améliorer au fil du temps.

[1] Toutes les informations et publications relatives à cette conférence sont disponibles sur : http://trec.nist.gov

2. Le filtrage par lots (*batch filtering*) consiste à utiliser une base de documents préalablement étiquetés pour construire un modèle. Pour chaque document d'un flux, le système doit prendre une décision binaire et peut utiliser, comme précédemment, la classe des documents sélectionnés pour s'améliorer.
3. Pour le routage (routing), le système dispose également d'une base de documents étiquetés pour l'apprentissage, puis est ensuite figé dans le temps. Les documents de la base de test doivent être ensuite ordonnés, du plus pertinent au moins pertinent. Le système ne doit donc pas effectuer une décision binaire, mais il doit être capable de calculer un score de pertinence.

Il est toujours possible de passer du routage au filtrage par lots en considérant que les documents dont le score est au-dessus d'un certain seuil sont pertinents. Il est nécessaire de choisir un "bon" seuil, ce qui n'est pas trivial : un système performant pour le routage peut être médiocre pour le filtrage par lots si le seuil n'est pas correctement choisi.

1.6 Notre travail

Le but de nos travaux est de développer un modèle fondé sur l'apprentissage numérique pour la catégorisation de textes ou, plus précisément, pour ce qui correspond à la tâche de routing dans le découpage de TREC.

Notre approche propose l'utilisation d'un réseau de neurones avec une architecture qui prend en considération le contexte local des mots. Malgré un nombre de paramètres élevés et un nombre d'exemples souvent limité, l'utilisation d'une méthode de régularisation permet d'effectuer correctement les apprentissages.

Nos résultats ont été validés d'une part grâce au corpus Reuters-21578[1] qui est souvent utilisé par la communauté de la catégorisation de textes, et d'autre part, par la participation aux sous-tâches de routing de TREC-8 [Stricker *et al*., 2000b] et TREC-9 [Stricker *et al.*, 2001] qui ont permis d'effectuer des comparaisons chiffrées avec d'autres approches.

Tous ces corpus sont composés d'articles d'organes de presse les dépêches de l'AFP dont le filtrage est décrit dans le chapitre 9.

[1] Ce corpus est publiquement accessible sur le site : http://www.att.research.com/~lewis/reuters21578.html

Nos travaux ont été intégrés dans l'application ExoWeb développé à la Caisse des Dépôts [Landau *et al.*, 1993] [Wolinski et Vichot, 2001] pour y ajouter des fonctionnalités opérationnelles originales. Cette application offre, sur l'intranet du groupe, un service de catégorisation de dépêches AFP en temps réel ; cette catégorisation s'effectue grâce à des modèles à bases de règles.

La première fonctionnalité développée offre un outil pour l'administrateur du système pour surveiller automatiquement le vieillissement de filtres construits sur des modèles à base de règles [Wolinski *et al.*, 2000]. L'idée de cette application est de fabriquer une "copie" d'un filtre à base de règles avec un filtre utilisant un réseau de neurones. Comme, le réseau de neurones produit une probabilité de pertinence et non une réponse binaire, il est possible d'analyser les plus grandes divergences entre les deux filtres : les documents considérés comme pertinents par la méthode à base de règles, mais obtenant une probabilité proche de zéro avec le réseau de neurones, et les documents considérés comme non pertinents avec le premier et obtenant une probabilité de pertinence proche de un avec le second.

Nous proposons ensuite les bases d'une deuxième application pour qu'un utilisateur puisse fabriquer lui-même un filtre à sa convenance avec un travail minimum. Pour réaliser cette application, il est nécessaire que l'utilisateur fournisse une base de documents pertinents. Cela peut se faire grâce à l'utilisation d'un moteur de recherche conjointement avec un réseau de neurones [Stricker *et al.*, 2000a] ou uniquement grâce au moteur de recherche.

1.7 Plan du mémoire

Le chapitre 2 est une présentation des modèles couramment utilisés en recherche d'informations, comme le modèle vectoriel ou le modèle probabiliste. Ces modèles sont présentés, car ils sont à l'origine de beaucoup de modèles construits pour la catégorisation de textes. Des approches mettant en œuvre des méthodes d'apprentissage numérique pour la catégorisation de textes sont également présentées, en insistant sur les approches "neuronales".

Le chapitre 3 présente les corpus utilisés tout au long de cette étude : le corpus Reuters-21578, et le corpus de la tâche de routing de TREC-8. Cette présentation expose les caractéristiques de chacun de ces deux corpus afin de mettre en avant leurs spécificités. Ces

corpus sont disponibles gratuitement et ont été utilisés par d'autres auteurs, ce qui facilite les comparaisons.

Le chapitre 4 introduit les différentes mesures utilisées pour évaluer les performances des systèmes. Au-delà des définitions, ce chapitre met en évidence le bruit inhérent à ces mesures et l'impossibilité qu'il existe à faire une mesure exacte. Les performances absolues n'ont pas un grand sens, et seules les performances relatives sont importantes.

Le chapitre 5 montre comment les textes sont transformés pour pouvoir être utilisés par les méthodes d'apprentissage numérique. Ce chapitre met en évidence la nécessité et la difficulté d'effectuer une sélection de descripteurs. Nous proposons une méthode entièrement automatique en deux étapes. La première étape détermine le vocabulaire spécifique des documents pertinents par rapport à l'ensemble du corpus ; la deuxième étape est une procédure d'orthogonalisation selon la méthode de Gram-Schmidt qui présente l'avantage d'être adaptée à la classification. Les comparaisons avec d'autres méthodes comme l'information mutuelle ou la méthode du chi-2 prouvent que, malgré des approches différentes, il existe peu de différences significatives entre toutes ces méthodes pour les problèmes qui nous concernent.

Le chapitre 6 est une présentation succincte des réseaux de neurones. Ce chapitre insiste sur la notion de surapprentissage pour les problèmes de classification, et montre que les méthodes de régularisation comme le *weight decay* apporte une solution à ce problème tout en ajoutant de nouveaux paramètres appelés hyperparamètres. Ces hyperparamètres peuvent être théoriquement déterminés grâce à l'approche bayésienne qui est présentée avec les approximations nécessaires à sa mise en œuvre.

Le chapitre 7 présente les premières expériences effectuées sur le corpus Reuters ainsi que la description de notre participation à TREC-8. Ce chapitre étudie l'impact des différents paramètres intervenant dans la sélection des descripteurs sur les performances (nombre de documents non pertinents, choix de ces documents, nombre de descripteurs initiaux). L'étude

du nombre optimal de neurones cachés montre qu'il est nécessaire d'améliorer la représentation des textes avant de complexifier l'architecture des réseaux de neurones en ajoutant des neurones cachés.

Le chapitre 8 présente une méthode originale pour déterminer automatiquement le contexte caractéristique d'un mot pour effectuer une désambiguïsation partielle de ce mot. L'architecture neuronale est modifiée pour prendre en considération cette nouvelle représentation. Avec celle-ci, l'utilisation d'une méthode de régularisation est indispensable ; malheureusement les résultats de l'approche bayesienne n'ont pas permis d'obtenir des résultats satisfaisants pour la détermination des hyperparamètres.
Finalement les résultats obtenus sur le corpus Reuters et sur le corpus de TREC-8, montrent une amélioration notable des résultats par rapport au chapitre 7.
Ce chapitre se termine par la description de notre participation à TREC-9.

Enfin le chapitre 9, montre comment ces résultats sont intégrés dans une application existante de filtrage de dépêches de l'AFP en temps réel. Les méthodes d'apprentissage numérique permettent de proposer de nouvelles fonctionnalités à cette application. Une première application utilise la sortie d'un filtre construit sur des méthodes à base de règles conjointement avec la sortie d'un filtre neuronal pour surveiller le premier filtre. Une deuxième application utilise un moteur de recherche couplé aux méthodes neuronales pour permettre à un utilisateur de définir son propre filtre.

Chapitre 2 Catégorisation de textes et apprentissage numérique : état de l'art

Afin de mettre l'apport proposé dans ce mémoire dans la perspective des travaux publiés sur le même sujet, nous consacrons ce chapitre à une présentation des approches les plus apparentées à la nôtre. Le nombre très important de conférences et de publications relatives à la recherche de textes rend impossible une présentation exhaustive des méthodes. Nous insistons plus particulièrement sur les travaux qui mettent en œuvre des méthodes d'apprentissage numérique. Les comparaisons portant sur les techniques proprement dites sont faites dans les chapitres correspondants.

Comme notre travail porte sur l'utilisation des réseaux de neurones pour la catégorisation de textes, nous portons une attention particulière aux approches mettant en œuvre ces outils, en insistant sur les difficultés mises en évidence par ces travaux.

En conclusion, nous exposons les méthodologies habituellement utilisées pour comparer toutes ces approches.

2.1 La recherche d'informations

Nous présentons ici les modèles les plus couramment utilisés pour la recherche d'informations, notamment le modèle vectoriel et le modèle probabiliste. Bien que la recherche d'informations ne soit pas le thème d'étude de ce mémoire, ces modèles sont importants car ils sont à l'origine de nombreux modèles de catégorisation de textes : la distinction entre ces deux domaines n'est pas toujours très facile à établir.

[Grossman et Frieder, 1998] détaillent les différents modèles de recherches d'informations.

2.1.1 Les requêtes booléennes

L'approche booléenne consiste à trouver les documents qui ont exactement les mêmes termes qu'une requête construite par mots clefs. Les requêtes peuvent être affinées grâce aux opérateurs *OR* ou *AND* ou encore au moyen d'opérateurs comme *NEAR*. Ce type de recherche

est à la base des moteurs de recherche comme Altavista[1] ou Google[2]. Cette approche est très efficace pour des requêtes utilisant des termes très spécifiques ou portant sur des domaines techniques particuliers avec leur vocabulaire propre ; son intérêt reste néanmoins limité. De plus, les requêtes booléennes ont le désavantage de fournir une réponse binaire (les documents contiennent les termes demandés ou ne les contiennent pas).

Les deux approches présentées ci-dessous, largement utilisées en pratique, permettent de remédier à ces inconvénients.

2.1.2 Modèle vectoriel et formule de Rocchio

Le modèle vectoriel introduit par [Salton *et al.*, 1975] représente chaque document, ainsi que la requête, par un vecteur et calcule un coefficient de similarité entre chaque document et la requête (appelé *Retrieval Status Value* ou *RSV*) ; il est donc possible de classer les documents par ordre de pertinence décroissante. Ce coefficient de similarité correspond, par exemple, au cosinus des angles entre le vecteur de la requête et le vecteur d'un document, afin de trouver les documents dont le vecteur de représentation est le plus colinéaire avec le vecteur de la requête. Dans ce modèle, chaque mot du corpus représente une dimension de l'espace et le codage des vecteurs se fait soit par une fonction booléenne, soit par une fonction du nombre d'occurrences d'un mot dans le document. Les composantes des vecteurs peuvent également être des paires de mots ou des phrases ; les composantes des vecteurs sont appelées *termes* dans la terminologie de la recherche d'information.

Avec cette approche, seule la présence ou l'absence de termes est porteuse d'information. Aucune analyse linguistique n'est utilisée, ni aucune notion de distances entre les mots : les documents sont représentés en "sacs de mots".

De nombreuses solutions ont été proposées dans la littérature pour coder les composantes des vecteurs, c'est-à-dire pour attribuer un poids à chaque terme (cf. [Salton et Buckley, 1990]). Historiquement, le plus connu de ces codages s'appelle *tf.idf*, et donne parfois son nom à l'approche vectorielle ; ce codage signifie : *term frequency * inverse document frequency.* Certains auteurs proposent également d'utiliser des fonctions différentes pour coder les termes

[1] http://www.altavista.com

[2] http://www.google.com

de la requête et les termes des documents, ainsi qu'une fonction de similarité qui tienne compte des différences de longueurs des documents [Singhal, 1996].

Ces différents codages sont présentés au chapitre 5.

La formule de Rocchio

La formule de Rocchio[1] est une extension du modèle vectoriel qui transforme automatiquement une requête initiale (représentée par un vecteur noté Q_0) en une nouvelle requête (représentée par un vecteur noté Q_1) plus performante.

Grâce au modèle vectoriel, un ensemble de documents répondant à la requête initiale est proposé à un utilisateur qui les étiquette (*relevance feedback*). La nouvelle requête Q_1 est construite grâce à la formule de Rocchio [Rocchio, 1971], dont l'idée est d'ajouter à la requête initiale les termes des documents pertinents et de lui retrancher les termes des documents non pertinents :

$$Q_1 = \alpha Q_0 + \beta \frac{1}{R} \sum_{d \in P} d - \gamma \frac{1}{N - R} \sum_{d \notin P} d$$

Dans cette formule, les documents sont représentés par un vecteur d, P est l'ensemble des documents pertinents, R son cardinal et N le nombre total de documents étiquetés ; le triplet (α, β, γ) est choisi en fonction de l'importance que l'on souhaite donner à chaque terme.

S'il existe en plus de la requête initiale une base de documents étiquetés, ces documents sont utilisés comme s'il s'agissait de documents jugés par un utilisateur.

S'il n'existe pas de requête initiale, mais uniquement des documents étiquetés comme pertinents ou non pertinents (c'est-à-dire dans le cas de la catégorisation de textes), alors le premier terme est supprimé et une requête Q_1 est construite grâce à la formule. Cette formule permet donc d'effectuer également de la catégorisation de textes.

Il est également possible de simuler l'interaction d'un utilisateur en postulant que les dix premiers documents trouvés par une première recherche sont pertinents et les cent derniers sont non pertinents (*pseudo relevance feedback*).

[1] Dans la suite de ce mémoire, on utilisera indifféremment les expressions "formule de Rocchio" et "algorithme de Rocchio".

La reformulation automatique d'une requête grâce à l'utilisation de documents pertinents et non pertinents à été un succès de la recherche d'informations [Salton et Buckley, 1990]. Le modèle vectoriel est à l'origine du modèle SMART [Salton, 1989] qui a fait ses preuves à la conférence TREC ; l'algorithme de Rocchio est également très performant pour la catégorisation de textes [Schapire *et al.*, 1998].

2.1.3 Modèle probabiliste

Dans l'approche du modèle probabiliste, le coefficient de similarité entre un document et la requête est la probabilité que le document soit pertinent connaissant la requête.
[Robertson et Sparck Jones, 1976] ont proposé un calcul de cette probabilité qui s'appuie sur les calculs de la probabilité qu'un terme soit présent sachant que le document est pertinent et la probabilité qu'un terme soit présent sachant que le document est non pertinent.
Une description détaillée de cette approche peut être trouvée dans [Sparck Jones, 1999].
Ce modèle a donné lieu à beaucoup d'extensions et est à l'origine du système OKAPI qui est l'un des systèmes les plus performants de TREC (avec le modèle vectoriel) dont une description peut être trouvée dans [Robertson et Walker, 2000].

2.2 La catégorisation de documents et l'apprentissage

Les méthodes d'apprentissage se divisent en deux approches principales : l'approche numérique et l'approche symbolique. Comme notre travail s'inscrit dans le cadre des approches fondées sur l'apprentissage numérique, nous proposons dans ce paragraphe, une revue des méthodes d'apprentissage numérique pour la catégorisation de textes utilisées dans la littérature. Un bon exemple de l'utilisation de l'apprentissage symbolique pour la catégorisation de textes peut être trouvé dans [Moulinier, 1997].

La plupart de ces approches utilisent une représentation des textes en sacs de mots issus du modèle vectoriel. Etant donné le grand nombre de descripteurs potentiels, il est, en général, nécessaire d'effectuer une sélection de descripteurs avant de pouvoir utiliser un modèle d'apprentissage.
Ces approches comprennent donc deux grandes étapes : la sélection de descripteurs et le choix de la méthode d'apprentissage numérique.

2.2.1 La sélection de descripteurs

2.2.1.1 Les méthodes de sélection de descripteurs

Quel que soit le modèle d'apprentissage utilisé, la problématique de sélection de descripteurs se pose, car, avec la représentation en sac de mots, chacun des mots d'un corpus est un descripteur potentiel. Or pour un corpus de taille raisonnable, ce nombre peut être de plusieurs centaine de milliers. En général, il est admis que les mots les plus fréquents peuvent être supprimés : ils n'apportent pas d'information sur le sens d'un texte puisqu'ils sont présents sur l'ensemble des textes. Les mots très rares, qui n'apparaissent qu'une ou deux fois sur un corpus, sont également supprimés, car il n'est pas possible de construire de statistiques fiables à partir d'une ou deux occurrences.

Cependant, même après la suppression de ces deux catégories de mots, le nombre de candidats reste encore très élevé, et il est nécessaire d'utiliser une méthode statistique pour déterminer les mots utiles pour la discrimination entre documents pertinents et documents non pertinents. Parmi les méthodes les plus souvent utilisées figurent le calcul de l'information mutuelle [Lewis, 1992] [Mouliner, 1997] [Dumais *et al.*, 1998], la méthode du chi-2 [Schütze *et al.*, 1995] [Wiener *et al.*, 1995] ou des méthodes plus simples utilisant uniquement les fréquences d'apparitions [Wiener, 1993] [Yang et Pedersen, 1997] ; d'autres méthodes ont également été testées [Sahami, 1998] [Zaragoza, 1999].

Une comparaison de l'information mutuelle et de la méthode du chi-2 avec d'autres méthodes est effectuée dans [Yang et Pedersen, 1997] ; il semble en résulter que l'information mutuelle est légèrement supérieure aux autres.

Une autre approche, appelée *latent semantic indexing* (LSI) proposée par [Deerwester *et al.*, 1990], consiste à effectuer une décomposition en valeurs singulières de la matrice dont chaque colonne représente un document grâce à un vecteur des occurrences des termes qui le composent. Cette matrice est projetée dans un espace de dimension plus faible où les descripteurs considérés ne sont plus de simples termes. Avec cette méthode, les termes apparaissant ensemble sont projetés sur la même dimension. Cette représentation est censée résoudre partiellement le problème des synonymes et des termes polysémiques. Initialement, cette approche a été utilisée pour effectuer de la recherche d'informations et permet théoriquement de trouver des documents pertinents pour une requête même s'ils ne partagent

aucun mot avec cette requête. Cette méthode de réduction des dimensions a ensuite été utilisée en entrée des modèles d'apprentissage numérique.

La méthode LSI a été utilisée pour la sélection de descripteurs dans [Wiener *et al.*, 1995] pour sélectionner les entrées d'un réseau de neurones ; la comparaison avec une méthode plus simple de sélection de termes montre très peu de différences, bien que la méthode LSI proposée dans l'article soit légèrement améliorée et implique l'utilisation d'un plus grand nombre de descripteurs. [Schütze *et al.*, 1995] ont également utilisé la méthode LSI pour sélectionner les descripteurs, et ont comparé les résultats obtenus avec une sélection de descripteurs effectuée avec la méthode du chi-2 : la sélection avec la méthode LSI n'améliore pas les performances.

2.2.1.2 Le nombre de descripteurs retenus

Les méthodes de sélection de descripteurs fournissent, en général, une liste de descripteurs ordonnés du plus important au moins important (la notion d'importance dépend de la méthode de classement considérée) ; il reste ensuite à déterminer combien de descripteurs sont à conserver dans cette liste. Ce nombre dépend souvent du modèle, puisque, par exemple, les machines à vecteurs supports sont capables de manipuler des vecteurs de grandes dimensions alors que, pour les réseaux de neurones, il est préférable de limiter la dimension des vecteurs d'entrées.

Pour choisir le bon nombre de descripteurs, il faut déterminer si l'information apportée par les descripteurs en fin de liste est utile, ou si elle est redondante avec l'information apportée par les descripteurs en début de liste.

Dans son utilisation des machines à vecteurs supports, Joachims [Joachims, 1998] considère l'ensemble des termes du corpus Reuters, après suppression des mots les plus fréquents et l'utilisation de racines lexicales (les *stems*) (la définition des racines lexicales est fournie au chapitre 5). Il reste alors 9962 termes distincts qui sont utilisés pour représenter les textes en entrée de son modèle. Il considère que l'ensemble de ces termes apporte de l'information, et qu'il est indispensable de les inclure tous. Cependant [Dumais *et al.*, 1998] utilisent également les machines à vecteurs supports mais ils ne considèrent que 300 descripteurs pour représenter les textes. Ils obtiennent néanmoins de meilleurs résultats que Joachims sur le

même corpus ; ce qui laisse à penser que tous les termes utilisés par Joachims n'étaient pas nécessaires.

Dans leur étude sur la sélection de descripteurs, [Yang et Pedersen, 1997] critiquent [Koller et Sahami, 1996] qui étudient l'impact de la dimension de l'espace des descripteurs en considérant des représentations allant de 6 à 180 descripteurs. Pour Yang et Pedersen, une telle étude n'est pas pertinente, car l'espace des descripteurs doit être de plus grande dimension ("*an analysis on this scale is distant from the realities of text categorization*").

À l'opposé, d'autres auteurs considèrent qu'un très petit nombre de descripteurs pertinents suffisent pour construire un modèle performant. Par exemple, [Wiener *et al.*, 1995] ne retiennent que les vingt premiers descripteurs en entrée de leurs réseaux de neurones. Plus récemment, [Stoica et Evans, 2000] ont proposé une méthode de sélection de descripteurs pour leur système CLARIT [Evans et Lefferts, 1995] et montrent que, pour obtenir des performances optimales avec leur système, 30 termes suffisent en moyenne sur le corpus Reuters.

Entre ces deux ordres de grandeurs, d'autres auteurs choisissent de conserver une centaine de mots en entrée de leur modèle [Lewis, 1992] [Ng *et al.*, 2000].

Finalement, il n'est pas prouvé qu'un très grand nombre de descripteurs soit nécessaire pour obtenir de bonnes performances, puisque, même avec des modèles comme les machines à vecteurs supports qui sont, en principe, adaptées aux vecteurs de grandes dimensions, les résultats sont contradictoires.

2.2.2 Les méthodes d'apprentissage numérique

Parmi les méthodes d'apprentissage les plus souvent utilisées figurent la régression logistique [Hull, 1994], les réseaux de neurones [Wiener, 1993] (et [Wiener *et al.*, 1995]) [Schütze *et al.*, 1995], l'algorithme du perceptron [Ng *et al.*, 2000], les plus proches voisins [Yang et Chute, 1994], les arbres de décision [Lewis et Ringuette, 1994] [Quinlan, 1996] [Apté *et al.*, 1998], les réseaux bayesiens [Lewis, 1992] [Lewis et Ringuette, 1994] [Joachims, 1998] [McCallum et Nigam, 1998a] [Sahami, 1998], les modèles de Markov Cachés [Zaragoza, 1999], les machines à vecteurs supports [Dumais *et al.*, 1998] [Joachims, 1998] et plus récemment les méthodes basées sur la méthode dite de *boosting* [Schapire *et al.*, 1998] [Iyer *et al.*, 2000].

Comme l'approche proposée dans ce mémoire repose sur l'utilisation des réseaux de neurones, nous présentons ci-dessous deux des approches mentionnées ci-dessus.

2.2.2.1 Les approches neuronales pour la catégorisation de textes

Une approche fondée sur les réseaux de neurones a été proposée dans la thèse de [Wiener, 1993] dont les résultats ont été repris dans [Wiener *et al.*, 1995]. Deux architectures neuronales sont proposées et testées sur le corpus Reuters-22173 (qui est une ancienne version du corpus Reuters-21578 disponible aujourd'hui).

La première architecture est un perceptron multi-couche avec une couche de neurones cachés et un neurone de sortie (cette architecture est présentée au chapitre 6) ; un réseau de neurones différent est construit pour chaque catégorie. Les descripteurs sont sélectionnés soit par une méthode de sélection de termes, soit par la méthode LSI, soit par une méthode LSI améliorée (*local LSI*).

Pour la deuxième architecture, les catégories du corpus Reuters sont regroupées en cinq grands ensembles (*agriculture, energy, foreign exchange, government, metals*). Un réseau est ensuite utilisé pour déterminer à quel ensemble appartient un document, puis cinq réseaux différents sont construits pour déterminer, à l'intérieur d'un ensemble, la catégorie exacte du document. Cette architecture a l'avantage de permettre à chacun des cinq réseaux d'être "spécialisé" et d'utiliser une représentation particulièrement adaptée pour distinguer des catégories proches.

Cette deuxième architecture améliore les résultats, mais elle nécessite un découpage manuel des catégories pour déterminer les ensembles et n'est réalisable que sur un corpus pour lequel le nombre de catégories est connu à l'avance et n'évolue pas.

Dans l'ensemble de cette étude, le surajustement est limité en considérant un terme de pénalisation dans la fonction de coût conjointement avec la méthode de l'arrêt prématuré ; ces différentes notions et les techniques correspondantes sont présentées au chapitre 6.

[Schütze *et al.*, 1995] ont également effectué de la catégorisation de textes avec des réseaux de neurones comportant une couche de neurones cachés. Leur modèle est identique au premier modèle utilisé dans [Wiener, 1993] ; les entrées sont sélectionnées soit par la méthode du chi-2, soit par la méthode LSI.

Cette étude montre notamment que, même lorsque le nombre de neurones cachés est nul (le modèle est une simple régression logistique), le modèle peut être surajusté. La mise en œuvre d'une procédure d'arrêt prématuré limite ce surajustement et améliore significativement les résultats.

A partir de ces deux études, il est possible de tirer plusieurs conclusions :

- Malgré ses avantages théoriques, la méthode LSI n'apporte pas d'amélioration sur une méthode de sélection des termes.
- Dans les deux études, l'ajout de neurones cachés n'améliore pas les résultats par rapport à une régression logistique. Nous reviendrons sur ce point au chapitre 7 en confrontant ces observations avec nos propres résultats.
- Il est nécessaire de se protéger du surajustement, même pour le modèle sans neurone caché, par une méthode de régularisation qui peut prendre la forme d'un terme de pénalisation dans la fonction de coût ou d'une procédure d'arrêt prématuré. Ce point est discuté en détail dans le chapitre 6.

2.2.3 Conclusion : quelle est la meilleure méthode pour la catégorisation de textes ?

Comme beaucoup d'approches différentes ont été utilisées pour la catégorisation de textes, une des questions récurrentes est : quelle est la meilleure méthode pour la catégorisation de textes ? Il existe, en pratique, plusieurs méthodologies pour tenter de répondre à cette question ; nous les décrivons ci-dessous.

La première consiste à comparer différentes méthodes mises en œuvre par différents auteurs sur le même corpus. L'inconvénient de cette méthode est qu'il faut que tous les auteurs utilisent exactement le même découpage du corpus. Pour le corpus Reuters-21578, qui est souvent utilisé, certains auteurs considèrent 90 catégories [Joachims, 1998], [Schapire *et al.*, 1998], [Yang et Liu, 1999], d'autres en considèrent 118 [Dumais *et al.*, 1998]. De plus, la plupart des auteurs considèrent 3299 documents sur la base de test, mais [Yang et Liu, 1999] en considèrent uniquement 3019 en supprimant tous les documents de la base de test qui n'appartiennent à aucune catégorie.

Finalement, ces légères différences de découpage rendent difficiles les comparaisons à travers ces publications. De plus, tous les auteurs n'utilisent pas les mêmes mesures de performances, et peuvent calculer les moyennes de manières différentes (les différentes mesures sont présentées au chapitre 4). Enfin, même dans le cas où les auteurs utilisent les mêmes mesures, il est nécessaire d'utiliser des tests statistiques pour vérifier que les différences ne sont pas dues au hasard [Hull, 1993].

Une autre approche souvent proposée est l'utilisation de plusieurs méthodes par le même auteur ; de cette manière, le découpage et les mesures sont identiques pour toutes les méthodes.

[Yang et Liu, 1999] comparent ainsi les machines à vecteurs supports, les plus proches voisins, les réseaux de neurones, une combinaison linéaire, et des réseaux bayesiens. [Dumais *et al.*, 1998] proposent également une série de comparaisons en mettant en compétition une variante de l'algorithme de Rocchio (appelée *find similar*), des arbres de décision, des réseaux bayesiens et des machines à vecteurs supports.

Le problème vient du fait que toutes ces méthodes sont délicates à mettre en œuvre et leurs performances dépendent fortement des algorithmes utilisés.

Par exemple, l'implémentation des machines à vecteurs supports proposées par [Dumais *et al.*, 1998] obtient de nettement meilleurs résultats que celle proposée par [Joachims, 1998].

Les réseaux de neurones testés par [Yang et Liu, 1999] sont des perceptrons multi-couche avec une couche cachée comportant 64 neurones, 1000 descripteurs en entrées et 90 neurones de sorties correspondant aux 90 catégories ; ils considèrent un seul réseau pour l'ensemble des catégories comportant plus de 64000 poids (l'algorithme d'apprentissage n'est pas précisé). Il n'est pas surprenant, dans ces conditions, que les performances obtenues ne soient pas très bonnes : de telles démarches jugent plus la capacité des auteurs à mettre en œuvre des méthodes, que les capacités des méthodes elles-mêmes.

L'algorithme de Rocchio est considéré comme un algorithme ancien, mais [Schapire *et al.*, 1998] ont montré que cet algorithme obtient d'excellents résultats pour la catégorisation de textes à condition d'utiliser un codage efficace, de bien choisir les documents non pertinents, et d'effectuer une optimisation des poids ("*a state of the art version of Rocchio's algorithm is*

quite competitive with modern machine learning algorithms for text filtering"). Leurs conclusions vont à l'encontre d'autres comparaisons qui montrent que cet algorithme n'est pas performant par rapport aux méthodes fondées sur l'apprentissage numérique [Schütze *et al.*, 1995] [Lewis *et al.*, 1996] [Cohen et Singer, 1996].

Ces différentes remarques prouvent que le succès d'une méthode dépend d'un ensemble de paramètres qui vont du codage des documents au choix des algorithmes et de leur utilisation, et qu'il est, par conséquent, extrêmement difficile de tirer des conclusions définitives sur une approche.

Il nous semble que la conférence TREC est une bonne solution pour comparer différentes méthodes, car chaque participant propose des solutions qu'il connaît bien avec des algorithmes dont il a pu tester l'efficacité. Le corpus est évidemment identique pour tout le monde, ainsi que les méthodes d'évaluation et la répétition annuelle de cette conférence permet de juger les approches sur le long terme.

De plus la conférence TREC a l'avantage de proposer un état de l'art à un instant donné contrairement aux comparaisons faites à partir des publications pour lesquelles le décalage dans le temps peut rendre certaines conclusions obsolètes.

Chapitre 3 Présentation des corpus utilisés

Ce chapitre présente les corpus auxquels nous ferons référence dans la suite de ce mémoire. Deux corpus ont été principalement étudiés : le corpus Reuters-21578 et le corpus de la tâche de filtrage de TREC-8. Le but de cette présentation est de mettre en évidence les caractéristiques de chacun de ces corpus ainsi que leurs différences. Le corpus Reuters-21578 a notamment une taille beaucoup plus réduite que le corpus de TREC-8 et présente l'avantage de regrouper des thèmes avec beaucoup de documents pertinents et d'autres pour lesquels il y en a peu.

Le corpus utilisé pour la tâche de filtrage de TREC-9 est présenté à l'annexe A car il n'est que dans le chapitre 8.

3.1 Le corpus Reuters-21578

Le corpus Reuters-21578 est un ensemble de dépêches financières émises au cours de l'année 1987 par l'agence Reuters, en langue anglaise, et disponible gratuitement sur le web[1]. Ce corpus est une mise à jour du corpus Reuters-22173 étudié notamment par [Lewis, 1992], [Wiener, 1993], [Moulinier, 1997]. Cette mise à jour, effectuée en 1996, a permis de supprimer les documents présents deux fois, de corriger des erreurs typographiques, de préciser certains formats, et de mieux définir le découpage à considérer pour l'apprentissage et le test. Du fait de ces corrections, il n'est pas possible de comparer les performances obtenues sur les différentes versions du corpus. Cependant, les caractéristiques globales du corpus sont restées identiques, et les remarques sur le comportement général des systèmes étudiés sont toujours valables.

La Figure 3.1 est un exemple de texte présent dans le corpus ; on peut noter la présence des sigles *<AXP>* et *<MER>* utilisés pour signaler des noms d'entreprise. Cet exemple montre qu'il s'agit de textes de style journalistique, rédigés.

[1] http://www.research.att.com/~lewis/reuters21578.html

Dans toute la suite de ce mémoire, ce corpus Reuters-21578 sera simplement nommé *corpus Reuters.*

```
SHEARSON LEHMAN NAMES NEW MANAGING DIRECTOR
Shearson Lehman Brothers, a unit of
American Express Co &lt;AXP>, said Robert Stearns has joined the
company as managing director of its merger and acquisition
department.
    Shearson said Stearns formerly was part of Merrill Lynch
Pierce, Fenner and Smith Inc's &lt;MER> merger and acquisitions
department.
```

Figure 3.1 : *Exemple de texte du corpus Reuters-21578.*

Le corpus Reuters est souvent utilisé lors d'évaluation dans les publications, comme dans [Schapire *et al.*, 1998] pour comparer leur algorithme *AdaBoost* avec la formule de Rocchio, ou dans [Joachims, 1998] et [Dumais *et al.*, 1998] pour évaluer les performances des machines à vecteurs supports. [Yang et Liu, 1999] ont également utilisé ce corpus pour comparer différents algorithmes (machines à vecteurs supports, réseaux de neurones, arbres de décision, réseaux bayesiens).

Il est possible, sur ce corpus, de comparer les performances de l'approche proposée dans ce mémoire avec celles d'autres approches. Cependant, pour pouvoir faire ces comparaisons, il est nécessaire de remplir plusieurs conditions : d'une part, la base de test servant à l'évaluation des performances doit être identique pour toutes les méthodes, et d'autre part, les performances doivent être évaluées avec les mêmes mesures.

Afin de faciliter les comparaisons ultérieures, nous présentons, dans ce paragraphe, le découpage que nous avons considéré dans toute la suite de ce mémoire.

3.1.1 Distinction entre bases d'apprentissage et de test : le découpage Apté

Le découpage le plus souvent rencontré se nomme découpage *Apté* du nom des premiers auteurs à l'avoir proposé [Apté *et al.*, 1994]. La base d'apprentissage est constituée des documents antérieurs au 8 avril 1987, soit 9603 documents, et la base de test de tous les documents ultérieurs, soit 3299 documents.

Les catégories retenues sont celles pour lesquelles il existe au moins un document pertinent sur la base d'apprentissage et un document pertinent sur la base de test, ce qui permet de retenir 90 catégories différentes. Certains documents de la base de test peuvent appartenir à plusieurs catégories, d'autres à aucune.

C'est ce découpage que nous avons utilisé dans tout notre travail.

Comme nous l'avons déjà souligné au chapitre précédent, il existe malheureusement de légères modifications à ce découpage qui rendent certaines comparaisons difficiles. Ainsi [Yang et Liu, 1999] ont supprimé de la base de test tous les documents qui n'appartiennent à aucune catégorie : ils n'utilisent que 3019 documents sur la base de test. La suppression de ces documents ne peut qu'améliorer les résultats par rapport au découpage traditionnel, puisque les risques de mauvais classement sont réduits. [Dumais *et al.*, 1998] considèrent 118 catégories : certaines catégories n'ont donc pas de documents pertinents sur la base de test, et la façon dont ces catégories sont prises en considération dans leur évaluation n'est pas très claire.

3.1.2 Définition des catégories du corpus Reuters-21578

Les 90 catégories issues du découpage sont présentées à la Figure 3.2, ainsi que le nombre de documents pertinents disponibles sur chaque base. Elles sont classées par ordre décroissant du nombre de documents pertinents sur la base d'apprentissage. Le nombre de documents pertinents disponibles pour effectuer l'apprentissage décroît rapidement ; dès la vingt-sixième catégorie, ce nombre est inférieur à cinquante. Il faut noter que les documents pertinents sont à peu près également répartis sur les deux bases, c'est-à-dire que les catégories ayant beaucoup (respectivement peu) de documents pertinents sur la base d'apprentissage ont également beaucoup (respectivement peu) de documents pertinents sur la base de test.

La difficulté de ces catégories est variable : [Wiener *et al.*, 1995] ont montré que les distinctions entre certaines catégories reposent presque exclusivement sur la présence ou l'absence d'un mot-clef, (cette étude repose sur le corpus Reuters-22173, mais les conclusions sont vraies pour la nouvelle version du corpus).

	Catégorie	Apprentissage	Test
1	earn	2877	1087
2	acq	1650	719
3	money-fx	538	179
4	grain	433	149
5	crude	389	189
6	trade	369	118
7	interest	347	131
8	wheat	212	71
9	ship	197	89
10	corn	182	56
11	money-supply	140	34
12	dlr	131	44
13	sugar	126	36
14	oilseed	124	47
15	coffee	111	28
16	gnp	101	35
17	gold	94	30
18	veg-oil	87	37
19	soybean	78	33
20	nat-gas	75	30
21	bop	75	30
22	livestock	75	24
23	cpi	69	28
24	reserves	55	18
25	cocoa	55	18
26	carcass	50	18
27	copper	47	18
28	jobs	46	21
29	yen	45	14
30	ipi	41	12
31	iron-steel	40	14
32	cotton	39	20
33	gas	37	17
34	barley	37	14
35	rubber	37	12
36	alum	35	23
37	rice	35	24
38	palm-oil	30	10
39	meal-feed	30	19
40	sorghum	24	10
41	retail	23	2
42	zinc	21	13
43	silver	21	8
44	pet-chem	20	12
45	wpi	19	10

	Catégorie	Apprentissage	Test
46	tin	18	12
47	rapeseed	18	9
48	orange	16	11
49	housing	16	4
50	strategic-metal	16	11
51	hog	16	6
52	lead	15	14
53	soy-oil	14	11
54	heat	14	5
55	soy-meal	13	13
56	fuel	13	10
57	lei	12	3
58	sunseed	11	5
59	dmk	10	4
60	lumber	10	6
61	tea	9	4
62	income	9	7
63	oat	8	6
64	nickel	8	1
65	l-cattle	6	2
66	groundnut	5	4
67	instal-debt	5	1
68	rape-oil	5	3
69	platinum	5	7
70	sun-oil	5	2
71	jet	4	1
72	coconut	4	2
73	coconut-oil	4	3
74	potato	3	3
75	propane	3	3
76	cpu	3	1
77	copra-cake	2	1
78	palmkernel	2	1
79	naphtha	2	4
80	palladium	2	1
81	rand	2	1
82	dfl	2	1
83	nzdlr	2	2
84	rye	1	1
85	cotton-oil	1	2
86	lin-oil	1	1
87	castor-oil	1	1
88	sun-meal	1	1
89	groundnut-oil	1	1
90	nkr	1	2

Figure 3.2 : *Définition des catégories, avec le nombre de documents pertinents disponibles pour chaque partie de la base.*

3.2 Le corpus TREC-8

La conférence TREC (TREC)[1] a été présentée brièvement au premier chapitre avec les trois sous-tâches de relatives au filtrage : le filtrage adaptatif, le filtrage par lots (*batch*) et le routage (*routing*)[2].

Parmi ces trois sous-tâches, le filtrage adaptatif est considéré comme le moins bien adapté à la mise en œuvre de méthodes statistiques d'apprentissage, et le routing comme le mieux adapté à l'application de ces méthodes. Nous verrons, en effet, dans le paragraphe suivant, que le routing est la sous-tâche pour laquelle le nombre de documents pertinents est le plus élevé.

3.2.1 Description des données utilisées pour le filtrage dans TREC-8

Pour la compétition TREC-8, il fallait construire cinquante profils correspondant à cinquante requêtes différentes, identifiées par un numéro allant de 351 à 400. La Figure 3.3 présente deux requêtes proposées pour la compétition. Elles sont définies par un titre, une description qui détaille le titre et une partie narrative qui précise exactement ce que doivent être les documents pertinents, et également ce qu'ils ne doivent pas être. Ces deux exemples montrent que les requêtes sont assez précises, donc difficiles à satisfaire.

À l'époque de la compétition, les documents pertinents pour chacune des requêtes provenaient de plusieurs corpus : le *Financial Times* 1992, 1993 et 1994 (noté FT92, FT93, FT94), *Federal Registe*r 1994 (FR94), *Congressional Record* 1993 (CR), *Foreign Broadcast Information Service* (FBIS), et *LA Times*.

[1] http://trec.nist.gov

[2] Dans la suite de ce mémoire, nous utilisons indifféremment les mots *routing* et *routage.*

```
<num> Number: 351
<title> Falkland petroleum exploration

<desc> Description:
What information is available on petroleum exploration in
the South Atlantic near the Falkland Islands?

<narr> Narrative:
Any document discussing petroleum exploration in the
South Atlantic near the Falkland Islands is considered
relevant.  Documents discussing petroleum exploration in
continental South America are not relevant.


<num> Number: 352
<title> British Chunnel impact

<desc> Description:
What impact has the Chunnel had on the British economy and/or
the life style of the British?

<narr> Narrative:
Documents discussing the following issues are relevant:

 - projected and actual impact on the life styles of the British
 - Long term changes to economic policy and relations
 - major changes to other transportation systems linked with
   the Continent

Documents discussing the following issues are not relevant:

 - expense and construction schedule
 - routine marketing ploys by other channel crossers (i.e.,
   schedule changes, price drops, etc.)
```

Figure 3.3 : *Deux exemples de requêtes pour la compétition TREC-8.*

La Figure 3.4, qui est un exemple de texte extrait du corpus *Financial Times*, montre que, comme sur le corpus Reuters, les textes sont rédigés.

```
FT 21 SEP 93
Hounslow to cut 250 jobs
HOUNSLOW, the Labour-controlled London borough, is to shed 250 jobs as part
of a Pounds 6m cuts package it said would be necessary to avoid budget-
capping by central government.
It is the first council to announce its planned budget for next year, and
it said the cuts had been made necessary by the government's plan to cut
budgets by 2 per cent.
Hounslow's cut is equivalent to 4.7 per cent of its controllable budget and
the council predicted that other authorities would have to follow suit. Mr
John Chatt, the council leader, said: 'We have gone as far as we can in
recommending cuts which will minimise the effect on direct services to the
community. If we don't get a decent grant settlement then any further cuts,
and the damage they will cause, will be due to the government's failure to
listen to our pleas.'
(…)
```

Figure 3.4 : *Exemple de texte issu du Financial Times.*

La Figure 3.5 précise, pour chacune des différentes sous-tâches, les bases à utiliser ; comme indiqué plus haut, le routing bénéficie du plus grand nombre de documents pertinents pour la base d'apprentissage.

	Adaptatif	Filtrage par lots	Routage
Apprentissage	-	FT92	FT92, FR94, LA CR, FBIS
Test	FT92, FT93, FT94	FT93, FT94	FT93, FT94

Figure 3.5 : *Répartition des exemples en base d'apprentissage et base de test selon les différentes sous tâches.*

Pour le routing, la Figure 3.6 indique le nombre de documents pertinents disponibles pour chaque thème, ainsi que la répartition entre les bases ; le nombre moyen de documents pertinents pour la base d'apprentissage est de 66, et la médiane se situe à 55.

La partie *Financial Times* 92 du corpus comporte 64139 documents. La base de test (FT93, FT94) est composée de 140650 documents ; le nombre moyen de documents pertinents sur la base de test est de 46,7 documents par thème, soit une proportion moyenne de 0,03 %.

Thème	Total	FT92	FBIS	FR	LA	Test (FT93-94)
351	31	11	19	0	1	17
352	62	55	3	0	4	190
353	88	15	34	8	31	29
354	311	21	175	0	115	49
355	44	1	38	2	3	1
356	10	8	0	1	1	15
357	230	24	205	0	1	61
358	51	0	0	0	51	2
359	10	4	0	0	6	30
360	139	2	89	0	48	12
361	8	0	1	0	7	2
362	36	0	17	0	19	5
363	12	2	1	0	9	4
364	33	1	16	2	14	2
365	32	8	1	7	16	3
366	84	4	17	20	43	17
367	164	12	94	4	54	30
368	56	0	2	4	50	5
369	13	1	0	1	11	0
370	326	15	0	267	44	31
371	16	1	1	10	4	1
372	37	3	0	17	17	12
373	23	2	0	17	4	12
374	150	19	22	0	109	59
375	76	9	34	1	32	10
376	87	13	65	0	9	17
377	29	5	5	3	16	10
378	61	51	9	0	1	44
379	16	0	0	0	16	0
380	6	0	1	0	5	1
381	26	5	8	2	11	2
382	18	2	10	2	4	4
383	75	15	0	5	55	67
384	48	1	34	0	13	3
385	61	9	30	1	21	27
386	16	1	0	3	12	6
387	76	4	57	12	3	9
388	39	3	6	1	29	14
389	95	41	32	2	20	129
390	85	2	1	12	70	68
391	67	55	0	0	12	127
392	82	27	28	1	26	32
393	66	4	3	0	59	5
394	12	0	3	1	8	5
395	156	36	102	1	17	62
396	55	6	1	27	21	5
397	21	2	0	0	19	5
398	136	10	66	0	60	10
399	96	4	60	14	18	9
400	109	34	49	0	26	16

Figure 3.6 : *Nombre de textes pertinents pour chaque thème.*

3.2.2 Fabrication du fichier des pertinences avant la conférence TREC-8

La constitution d'une base de documents étiquetés de grande taille est un problème complexe. Nous exposons ici la méthode retenue par NIST pour constituer ces bases, car elle a des conséquences importantes.

L'ensemble des documents représente environ cinq Gigaoctets de données ; il n'est donc évidemment pas possible de lire chaque document pour vérifier sa pertinence par rapport à l'ensemble des thèmes.

Le fichier des pertinences pour TREC-8 a été fabriqué grâce à la tâche *ad hoc* de TREC-7. Pour cette recherche *ad hoc*, les participants devaient fournir, pour chacune des 50 requêtes 351 à 400, les 1000 documents les plus pertinents, en ne disposant que des requêtes. Ensuite, pour chacune des requêtes, les cent premiers documents fournis par chaque candidat ont été assemblés pour former un ensemble soumis à des assesseurs qui classent chacun des documents de l'ensemble : pertinent ou non pertinent pour la requête. La liste des documents soumis aux assesseurs est ordonnée par ordre chronologique : ils ne savent pas à quel rang était classé chaque document, ni par quel système il a été sélectionné.

Grâce à ce travail, les bases sont étiquetées pour les requêtes 351 à 400, qui peuvent alors être utilisées pour TREC-8. L'ensemble des documents est ainsi divisé en trois parties : les documents étiquetés par les assesseurs comme pertinents, les documents étiquetés par les assesseurs comme non pertinents, et les documents qui ne faisaient pas partie de l'ensemble à étiqueter et qui sont considérés comme non pertinents.

On peut donc considérer qu'il existe deux sous-ensembles de documents non pertinents :

- les documents qui avaient été sélectionnés par un système et dont la non-pertinence a été vérifiée par un assesseur.
- les documents qui n'ont été sélectionnés par aucun système et qui n'ont donc jamais été lus par les assesseurs ; ils sont considérés comme non pertinents.

À l'aide de ce fichier, disponible pour chaque requête, il est possible, grâce aux documents reconnus comme pertinents, d'effectuer des apprentissages pour la sous-tâche du routing ou du filtrage par lots.

3.2.3 Fabrication du fichier des pertinences après la conférence TREC-8

Pour TREC-8, il avait été décidé qu'une nouvelle étude de la pertinence des documents relativement aux requêtes 351 à 400 serait effectuée grâce aux résultats fournis par les candidats de TREC-8, selon le principe expliqué au paragraphe précédent.

Grâce à cette nouvelle évaluation, de nouveaux documents pertinents ont été trouvés pour plusieurs requêtes sur l'ensemble de la base FT92, FT93 et FT94 ; les performances officielles de la compétition ont été calculées avec ces nouveaux fichiers de documents pertinents.

3.2.4 Conséquence importante

Comme il y a eu deux fichiers successifs de documents pertinents, il faut savoir exactement quels fichiers sont utilisés. Pour comparer les résultats d'une étude aux résultats de la compétition, il faut utiliser le fichier des documents pertinents disponible *avant* la compétition pour effectuer les apprentissages : sinon, on dispose de plus de documents pertinents pour l'apprentissage qu'avant la compétition. Il faut ensuite évaluer les résultats en utilisant le fichier des documents pertinents délivré *après* la compétition pour la base de test FT93, FT94 puisque les performances des systèmes à la compétition ont été évaluées avec ce fichier.

Il faut noter qu'il existe alors un léger biais, puisque les nouveaux documents proposés par une nouvelle méthode ne sont pas évalués par des assesseurs comme lors de la compétition.

3.3 Conclusion

Le corpus Reuters présente l'avantage d'être de taille modérée, et de représenter une grande variété de situations, avec des catégories comportant beaucoup d'exemples pertinents, et d'autres qui en comportent très peu.

Une des particularités du corpus TREC-8 pour la tâche de routing est que les exemples pertinents sont issus de corpus différents ; nous n'avons pas étudié l'impact exact de cette hétérogénéité, mais il semble que, pour certaines catégories, cela soit préjudiciable.

Il faut noter que ces deux corpus sont composés de textes journalistiques issus d'organes de presse de qualité puisqu'il s'agit de l'agence Reuters et du Financial Times. Ces textes sont rédigés et structurés dans une langue correcte et sont proches des dépêches de l'AFP traitées au chapitre 9. Si l'on avait travaillé sur la classification des courriers électroniques, il aurait été préférable de choisir d'autres corpus.

Il existe cependant une différence importante entre ces corpus et les dépêches AFP : ils sont en langue anglaise alors que les dépêches AFP sont en langue française. Mais toutes les méthodes utilisées reposent sur les statistiques d'apparition des mots et sont donc à peu près indépendantes de la langue, tant que les mots sont définis comme des chaînes de caractère entourées d'espace ou de signes de ponctuations (ce qui n'est pas nécessairement le cas pour toutes les langues, notamment certaines langues asiatiques).

Néanmoins, des exemples de thèmes avec les dépêches AFP sont également utilisés dans ce mémoire, pour vérifier que le comportement des méthodes n'est pas modifié avec la langue française. Il en effet facile d'utiliser l'application ExoWeb du chapitre 9 pour disposer d'un ensemble important de documents étiquetés comme pertinents puisqu'il existe déjà un système de filtrage à la Caisse des Dépôts.

Chapitre 4 Évaluation des performances d'un filtre

Pour comparer les différents systèmes de filtrage, il faut définir une mesure pour évaluer leurs performances. Malheureusement, différentes mesures sont utilisées dans la littérature, ce qui rend les comparaisons souvent difficiles. Le but de ce chapitre n'est pas de les présenter toutes, mais d'introduire uniquement les plus importantes d'entre elles, et de mettre en évidence la difficulté de l'évaluation des performances.

4.1 Mesures de la précision et du rappel

4.1.1 Définitions

La précision et le rappel sont deux quantités qui sont définies lorsque les filtres prennent des décisions binaires : soit un document est sélectionné par le filtre, soit il ne l'est pas. Lorsque les ensembles de documents pertinents et non pertinents sont connus sur un corpus, il est alors possible d'évaluer les quantités définies à la Figure 4.1.

	Pertinents	Non pertinents
Sélectionnés	*a*	*b*
Non sélectionnés	*c*	*d*
Total	*P*	*NP*

Figure 4.1 : *Table de contingences pour un filtre binaire.*

Le rappel R et la précision P et sont définis par :

$$R = \frac{a}{a + c} \qquad P = \frac{a}{a + b}$$

Le *rappel* est le rapport du nombre de documents pertinents trouvés par le filtre au nombre de documents pertinents disponibles. Il s'agit de la proportion de documents bien classés pour la classe des documents pertinents : c'est une mesure utilisée habituellement en classification.

La *précision* est la proportion de documents pertinents parmi les documents sélectionnés. Cette quantité ne représente pas un taux d'exemples bien classés par rapport à une classe et n'est donc pas normalisée.

Ces deux notions sont souvent utilisées, car elles reflètent le point de vue de l'utilisateur : si la précision est faible, l'utilisateur sera insatisfait, car il devra perdre du temps à lire des informations qui ne l'intéressent pas. Si le rappel est faible, l'utilisateur n'aura pas accès à une information qu'il souhaitait avoir.

Un filtre parfait doit avoir une précision et un rappel de un, mais ces deux exigences sont souvent contradictoires et une très forte précision ne peut être obtenue qu'au prix d'un rappel faible et vice-versa.

En effet, dans le cas limite où aucun document n'est sélectionné, la précision vaut un et le rappel est nul. Dans le cas limite où tous les documents sont sélectionnés, le rappel vaut un et la précision est de $\frac{P}{P + NP}$; cette quantité, appelée *densité* du thème, est généralement assez faible, et représente la précision moyenne que l'on obtiendrait en sélectionnant les documents aléatoirement.

On peut également définir les notions de *bruit* (*B*) et de *silence (S)* qui sont respectivement les notions complémentaires de la précision et du rappel :

$$B = 1 - P \qquad S = 1 - R$$

4.1.2 Une estimation pas si simple

Dans la pratique, les valeurs exactes de la précision et du rappel ne peuvent pas être calculées et, par conséquent, les valeurs absolues n'ont pas de sens.

Pour mesurer les performances d'un filtre (comme celles de tout classifieur), il faut utiliser une base de test indépendante de la base d'apprentissage. Pour que les résultats obtenus sur cette base de test aient un sens, deux conditions doivent être remplies : il faut, d'une part, que cette base soit suffisamment grande et représentative, et, d'autre part, il faut pouvoir évaluer les quantités *a*, *b*, *c* et *d* sur cette base. Or la connaissance de ces quantités nécessite la catégorisation manuelle de chaque document de la base, ce qui est justement très difficile à faire si la base est grande.

Ainsi, sur la base de test utilisée pour TREC-8 qui comporte 140.000 documents, seule une partie de ces documents a été examinée par des assesseurs. Tous ceux qui n'ont pas été examinés sont considérés comme non pertinents (la manière dont les documents ont été étiquetés a été présentée au chapitre précédent et est détaillée dans [Voorhees et Harman, 2000]). Il peut donc exister des documents pertinents sur la partie de documents non examinée.

Sur le corpus Reuters, l'ensemble des documents de la base de test est supposé avoir été correctement affecté "manuellement". En fait, dans la pratique il n'en est rien : certains documents pertinents sont étiquetés comme non pertinents. Par exemple, la Figure 4.2 montre un texte étiqueté comme non pertinent alors qu'il est pertinent pour la catégorie *money, foreign-exchange*, comme le montre le passage qui figure en gras.

MIYAZAWA SEES EVENTUAL LOWER U.S. TRADE DEFICIT
Japanese Finance Minister Kiichi Miyazawa told a press conference he expects the U.S. Trade deficit to eventually start reflecting economic fundamentals, which should influence exchange rates.
The minister was not referring to the U.S. Trade data to be released in Washington later today.
Miyazawa also said he told major industrial nations when he was in Washington last week that **present exchange rates are not necessarily good. He had said earlier in Washington that current exchange rates were within levels implied in the February Paris currency accord**.
REUTER

Figure 4.2 : *Texte 16745 du corpus Reuters. Ce texte est pertinent pour la catégorie money, foreign-exchange alors qu'il a été étiqueté manuellement comme non pertinent.*

De même le texte 21477 de la base de test présenté à la Figure 4.3 n'est pas étiqueté comme pertinent pour le thème *interest* alors que le passage en gras prouve qu'il est, en fait, pertinent.

DEUTSCHE BANK CHIEF SAYS LOUVRE PACT STILL INTACT
Deutsche Bank AG joint chief executive Friedrich Wilhelm Christians said he believed the Louvre accord on currency stability was still intact.
Christians told a news conference he met U.S. Treasury Secretary James Baker in the last two weeks, after **short term German interest rates had risen twice**.
"I am sure that with 1.7720 marks the dollar is still within the Louvre agreement. I do not see that the accord has been terminated," Christians said. He was responding to questions about comments by Baker, who said the Louvre accord was still operative but criticised rises in West German interest rates.
REUTER

Figure 4.3 : *Texte 21477 du corpus Reuters. Ce texte est pertinent pour la catégorie money, interest alors qu'il a été étiqueté manuellement comme non pertinent.*

La pertinence ou non d'un document dépend également de la personne qui étiquette les documents : d'une personne à l'autre, le même document peut être déclaré comme pertinent ou non pertinent. Ceci est surtout vrai pour la catégorisation de documents, puisqu'il n'existe pas, en général, de définition très précise du thème. Ainsi le texte présenté en Figure 4.4 peut être considéré comme pertinent pour le thème des *participations* ou non pertinent selon la définition que chacun donne exactement à ce thème.

```
Les banques italiennes san Paolo di Torino et Istituto Mobiliare Italiano
(IMI) ont signé lundi à Turin l'acte formel de fusion de leurs deux
établissements, qui donnera naissance à la première banque italienne.
```

Figure 4.4 : *Exemple de texte difficile à classer pour le thème participation.*

Pour toutes les raisons décrites ci-dessus, les valeurs exactes de précision et de rappel ne sont pas accessibles, et, en tout état de cause, leurs valeurs peuvent varier selon les personnes qui jugent les documents. En pratique les valeurs absolues n'ont donc pas beaucoup de sens ; en revanche, les valeurs relatives ont un sens pour comparer des systèmes entre eux, puisque les évaluations sont faites avec les mêmes approximations.

Cependant, pour que les approximations n'avantagent pas artificiellement un système par rapport à l'autre, il est indispensable de moyenner les performances de chaque système sur un ensemble de thèmes différents. On verra dans le paragraphe 4.4 comment agréger les résultats pour un ensemble de thèmes.

4.2 Courbes rappel-précision

4.2.1 Courbes non interpolées

En général, les filtres statistiques fournissent une probabilité de pertinence pour chaque document ; pour en déduire une réponse binaire, il faut déterminer une valeur pour le seuil de décision utilisé. Il est donc possible de calculer la précision et le rappel correspondant à chaque valeur du seuil de décision et de tracer l'évolution de ces deux quantités.

4.2.2 Courbes interpolées

On préfère calculer la précision pour des valeurs prédéfinies du rappel, de 0 % à 100 % par pas de 10 %. En pratique ces valeurs du rappel peuvent ne pas être atteintes exactement : les valeurs de la précision doivent donc être interpolées. La règle d'interpolation est la suivante : la

valeur interpolée de la précision pour un niveau de rappel *i* est la précision maximale obtenue pour un rappel supérieur ou égal à *i.* Cette règle d'interpolation définit donc également une précision pour un rappel nul alors qu'une telle valeur n'existe pas. La Figure 4.5 montre la courbe obtenue pour le filtre précédent avec les valeurs interpolées.

Cette courbe montre qu'il est toujours possible d'obtenir une précision élevée au prix d'un rappel faible ou un rappel élevé au prix d'une précision faible. Dans la pratique, on essaye de choisir un compromis entre ces deux exigences. On verra néanmoins dans le chapitre 9 que, pour certaines applications, le filtre fonctionne avec une précision élevée au détriment du rappel.

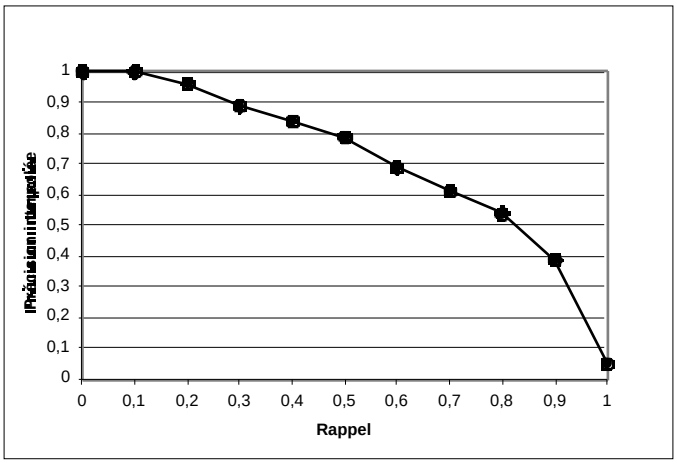


Figure 4.5 : *Courbe rappel-précision interpolée.*

L'avantage de cette interpolation est qu'elle permet de connaître la précision pour des valeurs standardisées. Lorsque plusieurs thèmes sont étudiés, on peut facilement obtenir la courbe moyenne d'un système en moyennant simplement toutes les précisions obtenues aux différents seuils du rappel pour les différents thèmes. Les performances d'un filtre sur un ensemble de thèmes peuvent donc être caractérisées par une seule courbe.

4.3 Caractérisation d'un système binaire par une seule grandeur

Le paragraphe précédent a montré qu'un système est caractérisé par une courbe ou par un couple (rappel, précision). Dans la pratique, il n'est pas facile de comparer les filtres sur la

base de ces caractéristiques ; on cherche donc à évaluer leurs performances par un seul nombre. Nous présentons trois mesures parmi les plus utilisées, en précisant leurs inconvénients éventuels. Il ne faut pas perdre de vue que la caractérisation d'une courbe par un seul nombre implique nécessairement une simplification ; lorsque cela est possible, il peut être avantageux d'utiliser tout de même les courbes rappel-précision

4.3.1 Précision moyenne sur 11 points

La précision moyenne sur 11 points consiste simplement à moyenner les 11 précisions interpolées obtenues pour les seuils de rappels fixes définis, de 0 %, à 100 % par pas de 10 %.

4.3.2 Point moyen

Le point moyen est défini comme le point pour lequel la précision est égale au rappel : c'est l'intersection de la courbe rappel-précision avec la diagonale principale.

S'il n'existe pas de seuil permettant de satisfaire cette égalité, une extrapolation linéaire est effectuée : si (R_1, P_1) et (R_2, P_2) sont les deux couples qui encadrent le point moyen, une extrapolation linéaire permet de trouver le point moyen :

$$R = P = \frac{R_2 P_1 - R_1 P_2}{R_2 - R_1 + P_1 - P_2}$$

Malheureusement, la valeur obtenue grâce à cette extrapolation n'a pas de signification réelle : elle caractérise plus une propriété d'un point de la courbe rappel-précision que la qualité qu'un système.

Nous éviterons cette mesure dans la suite de ce mémoire.

4.3.3 Mesure *F*

La mesure F_β [van Rijsbergen, 1979] prend en considération la précision et le rappel simultanément. Elle est définie par :

Pour utiliser cette mesure, il est donc nécessaire de fixer préalablement un seuil de décision pour le classement, et de calculer la valeur de F_β pour ce seuil.

Le paramètre β permet de choisir l'importance relative que l'on souhaite donner à chaque quantité. On choisit en général de donner la même importance aux deux critères : on utilise F_1 (noté F dans toute la suite de ce mémoire) qui s'écrit :

$$F = \frac{2.P.R}{P + R}$$

Une des propriétés intéressante de cette mesure est le fait que, si $P = R = X$, alors $F = X$; cette mesure a alors une interprétation simple.

Lorsque l'on utilise la mesure F, il ne faut pas perdre de vue qu'une partie de l'information est perdue, puisque cette mesure rend compte d'un seul point de la courbe dans la zone particulière où la précision et le rappel sont du même ordre de grandeur. La Figure 4.6 montre les résultats obtenus pour le thème *interest* du corpus Reuters pour trois jeux de paramètres différents. Avec la mesure F, ces trois systèmes ont des performances similaires, mais les courbes font apparaître des différences qui ne sont pas mises en évidence lorsque l'on caractérise les courbes par un seul point. D'après ces courbes, le système ayant la mesure de F la plus faible (F = 65,5), semble présenter le meilleur compromis : en effet, pour des valeurs de rappel faible, il obtient une précision élevée aussi élevée que le troisième système, et, dans la zone de rappel élevée, il a une précision comparable à celle du premier système.

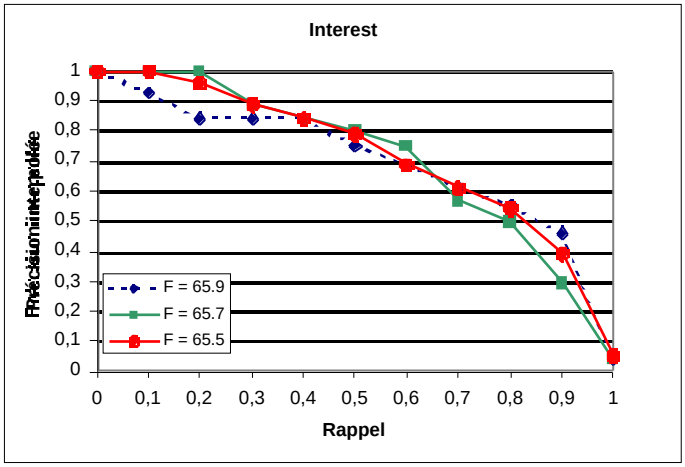


Figure 4.6 : *Courbes rappel-précision interpolée pour trois systèmes différents pour le thème interest du corpus Reuters.*

Malgré ces imprécisions, cette mesure présente un grand intérêt pratique, et elle est souvent utilisée dans les publications ; nous l'utiliserons fréquemment dans la suite de ce mémoire.

En pratique, la mesure de *F* est une fonction du seuil de décision du filtre ; par conséquent, afin de s'affranchir du choix du seuil de décision, il est possible de tester plusieurs seuils de 0 à 1 par pas de 0,1, et de conserver le seuil correspondant à la valeur de *F* la plus élevée sur la base de test. Cette mesure est appelée *F optimal* et représente la borne supérieure de la valeur de *F* que le système peut atteindre dans la pratique.

4.4 Moyenne sur un ensemble de thèmes : macro-moyenne et micro-moyenne

Comme les différences entre les performances de deux systèmes peuvent être liées aux incertitudes sur les mesures, il est indispensable d'effectuer une comparaison sur un ensemble assez vaste de thèmes, et de calculer une moyenne des performances sur cet ensemble de thèmes. Cette moyenne peut être calculée de deux manières.

- la macro-moyenne consiste à faire simplement une moyenne sur l'ensemble des thèmes des performances individuelles de chaque thème pour la mesure choisie. Cette moyenne donne un poids égal à chaque thème ;
- pour calculer la micro-moyenne, les tables de contingences de chaque thème, comme celle de la Figure 4.1, sont additionnées, puis les valeurs de précision et de rappel sont calculées sur cet ensemble, qui n'est plus considéré que comme un seul thème.

La différence de comportement des deux calculs se comprend bien sur un corpus ayant les caractéristiques du corpus Reuters.

Puisque toutes les catégories ont la même importance dans le calcul de la macro-moyenne, le résultat est surtout gouverné, sur ce corpus, par les catégories ayant peu de documents pertinents puisque celles-ci sont majoritaires.

En revanche, la micro-moyenne est gouvernée par les premières catégories (celles qui ont le plus de documents pertinents). Plus précisément, si un classifieur sélectionne tous les documents pertinents pour les vingt premières catégories et aucun document pour les catégories restantes, le rappel calculé pour la micro moyenne est de 83,9 %. Pour les trente premières catégories, le rappel est de 89,2%. Pour obtenir de bonnes performances avec la micro-moyenne, une bonne stratégie consiste donc à ne rien faire pour les soixante dernières

catégories puisque la perte de rappel est très faible et que le risque de faire chuter la précision est supprimé.

4.5 Précision moyenne non interpolée

Les mesures précédentes nécessitent l'utilisation d'un seuil de décision, alors que la précision moyenne non interpolée caractérise la qualité d'un classement. Le système calcule une probabilité de pertinence pour l'ensemble des documents qui constituent la base de test, et les classe par ordre de pertinence décroissante à la manière des moteurs de recherche sur le web.
En parcourant cette liste, la précision est calculée pour chaque document pertinent ; la précision moyenne non interpolée est obtenue en additionnant ces différentes précisions, et en divisant la somme par le nombre total de documents pertinents présents sur la base de test.

Pour fixer les idées, nous présentons un exemple. Supposons qu'une base de test comporte sept documents parmi lesquels trois sont pertinents, et qu'un filtre a permis d'obtenir le classement des documents présenté sur la Figure 4.7.

Classement	Pertinent	Précision
t1	non	-
t2	oui	1/2
t3	non	-
t4	oui	2/4
t5	non	-
t6	non	-
t7	oui	3/7

Figure 4.7 : Exemple de classement avec les précisions calculées pour chaque document pertinent.

Pour cet exemple, la **précision moyenne non interpolée** est (on utilise l'acronyme *UAP* pour désigner cette mesure, du nom anglais : *Uninterpolated Average Precision*) :

$$UAP = \frac{1}{3}(1/2 + 2/4 + 3/7) = 0,47$$

Ce nombre mesure la qualité du classement, grâce à un score évoluant entre 0 et 1, indépendamment du choix d'un seuil de décision.

4.6 Une mesure issue de TREC : l'utilité

4.6.1 Motivation et définition de l'utilité

Tous les critères d'évaluation présentés jusqu'ici présentent un inconvénient majeur : pour les calculer, il est nécessaire de connaître l'ensemble des documents pertinents. Or, dans la pratique, pour un système qui filtre les dépêches quotidiennement, il est impossible de calculer le rappel, puisque l'utilisateur ne dispose pas des dépêches pertinentes non sélectionnées.

De plus, considérons deux filtres différents pour un même thème, tels que le premier sélectionne 100 documents non pertinents et aucun document pertinent, alors que le deuxième sélectionne un document non pertinent et aucun document pertinent. Ces deux systèmes ont un rappel et une précision nuls ; néanmoins, dans la pratique, le deuxième filtre est préférable au premier puisqu'un utilisateur ne perd pas de temps à lire des documents qui ne l'intéressent pas. Dans ce cas, les mesures de précision et de rappel (et les mesures dérivées) ne permettent pas de différencier les deux systèmes.

Pour remédier à ces inconvénients, les fonctions d'utilité ont été introduites lors de la compétition TREC [Hull, 1999] dans le cadre de la tâche de filtrage.

Pour différencier les deux systèmes précédents, l'idée consiste à donner un nombre positif de points au système pour chaque document pertinent sélectionné et à retirer des points négatifs pour chaque document non pertinent sélectionné. L'utilité est donc de la forme :

$$U = a.P + b.NP$$

où P est le nombre de documents pertinents sélectionnés, et NP est le nombre de documents non pertinents sélectionnés. Les coefficients a et b varient selon l'importance relative que l'on souhaite donner à chaque terme. Les valeurs les plus couramment utilisées sont a = 3, b = -2 et a = 3, b= -1.

L'évaluation de l'utilité ne nécessite que l'observation des documents sélectionnés ; elle est donc plus facilement calculable que le rappel.

4.6.2 Inconvénients de l'utilité

Cette mesure présente quelques inconvénients, qui font qu'elle est peu utilisée en dehors de la conférence TREC.

· Elle n'est pas facilement interprétable, contrairement à la précision et au rappel. Plus précisément, le lien avec ces deux notions n'est pas immédiat. En fait si l'on considère deux systèmes X et Y, il est même possible d'obtenir les caractéristiques suivantes :

$$P(X) > P(Y)$$

$$R(X) > R(Y)$$

et néanmoins :

$$U(X) < U(Y)$$

La démonstration peut être trouvée dans [Hull et Robertson, 2000].

· La définition de l'utilité n'est pas normalisée d'un thème à l'autre : l'utilité maximale pour un thème donné dépend du nombre de documents pertinents présents dans l'ensemble du corpus P_c puisque cette utilité maximale est $a.P_c$. Il est donc impossible de moyenner les résultats obtenus à travers différents thèmes. Une normalisation a été proposée [Hull, 1999], mais son utilisation n'est pas très simple.

- Pour la tâche du filtrage adaptatif de la compétition TREC-8, tous les systèmes avaient une utilité négative. En conséquence, la meilleure des stratégies consistait à utiliser un système qui ne faisait rien puisqu'un tel système aurait eu une utilité nulle.

Ces différentes considérations font que l'utilité est rarement employée en dehors de TREC. Dans l'avenir, cette mesure devrait évoluer pour pallier ces inconvénients.

4.7 Conclusion

Nous avons montré dans ce chapitre que les mesures absolues de performances ont une portée limitée. Cette limitation est due, d'une part, à l'impossibilité de définir précisément la notion de pertinence, et d'autre part, à l'impossibilité d'obtenir des corpus de grande taille totalement et correctement étiquetés.

Il est nécessaire de mesurer les performances d'un filtre sur un ensemble de thèmes pour d'une part limiter l'impact des erreurs d'annotations et d'autre part, pour juger globalement une approche sur des thèmes de difficultés différentes.

Néanmoins toutes les mesures présentées dans ce chapitre traitent toutes les erreurs avec la même importance, alors que du point de vue de l'utilisateur, cette assertion n'est pas vraie :

- lorsqu'un document non pertinent est sélectionné, l'utilisateur sera plus indulgent si le document est "proche" du thème que si le document n'a absolument rien à voir avec celui-ci ;
- il n'est pas très grave de ne pas sélectionner certains documents pertinents si l'information qu'ils contiennent a déjà été apportée par d'autres documents ; en revanche, si l'utilisateur n'est pas du tout informé d'une nouvelle qu'il apprend par ailleurs, sa confiance dans le système diminuera fortement.

Il est cependant très difficile de prendre ces informations en considération, puisqu'il existe une grande part de subjectivité dans ces appréciations, et que finalement la seule vraie mesure est la satisfaction de l'utilisateur.

Dans la suite de ce mémoire, nous utiliserons principalement :

- les courbes rappel-précision, car elles apportent une information complète sur le comportement du filtre.
- la mesure *F*, car elle rend compte du comportement d'un filtre, une fois qu'un seuil de décision a été choisi : elle traduit la "perception" d'un utilisateur dans le cadre d'une application de filtrage.
- la précision moyenne non interpolée (notée *UAP*), car elle rend compte de la qualité du classement proposé par un filtre indépendamment du seuil de décision : un filtre peut avoir une valeur de *F* faible uniquement parce que le seuil de décision est mal choisi, et non parce que le classifieur est de mauvaise qualité.

Chapitre 5 Représentation des textes

Ce chapitre montre comment les textes sont transformés en vecteur de nombres pour être utilisés par les approches mettant en œuvre des apprentissages numériques. En général, les représentations n'utilisent pas d'information grammaticale ni d'analyse syntaxique des mots : seule la présence ou l'absence de certains mots est porteuse d'informations.

Nous présentons dans ce chapitre une méthode originale de sélection de descripteurs en deux étapes, qui présente plusieurs avantages. Elle est entièrement automatique et ne nécessite pas de ressources externes (comme une liste de mots les plus fréquents dans une langue donnée) et elle est couplée avec un critère d'arrêt pour trouver le "bon" nombre de descripteurs.

5.1 La représentation en sac de mots

La représentation des textes la plus simple a été introduite dans le cadre du modèle vectoriel présenté au chapitre 2, et porte le nom de "sac de mots". Les textes sont transformés simplement en vecteurs dont chaque composante représente un terme. Dans un premier temps, les termes sont les mots qui constituent un texte. Dans les langues comme le français ou l'anglais, les mots sont séparés par des espaces ou des signes de ponctuations ; ces derniers, tout comme les chiffres, sont supprimés de la représentation. On peut choisir de conserver les majuscules pour aider, par exemple, à la reconnaissance de noms propres, mais il faut alors résoudre le problème des débuts de phrase.

Les composantes du vecteur sont une fonction de l'occurrence des mots dans le texte.

À titre d'exemple, nous présentons, sur la Figure 5.1, une dépêche de l'Agence France Presse qui fournit des informations sur des prises de participations entre des entreprises. La transformation de ce texte en vecteur est présentée sous le texte. À partir de ces informations, un filtre doit détecter que cette dépêche est pertinente pour le thème des participations.

Cette représentation des textes exclut toute analyse grammaticale et toute notion de distance entre les mots : c'est pourquoi cette représentation est appelée "sac de mots".

```
Marionnaud: Union et Etudes Investissement franchit 5% des droits de vote
 PARIS, 31 juil (AFP) - La société Union et Etudes Investissement (caisse
Nationale de Crédit Agricole) a franchi en hausse le seuil de 5% des droits
de vote du groupement français de parfumerie Marionnaud et détient
désormais 292.157 actions, soit 8,09% du capital et 5,05% des droits de
vote, a indiqué vendredi le Conseil des Marchés Financiers.
Ce franchissement de seuil résulte de l'acquisition de 11.460 actions,
précise le CMF.
```

a	2	détient	1	le	3
acquisition	1	en	1	marchés	1
actions	2	et	4	marionnaud	2
agricole	1	études	2	nationale	1
caisse	1	financiers	1	parfumerie	1
capital	1	franchi	1	précise	1
ce	1	franchissement	1	résulte	1
cmf	1	franchit	1	seuil	2
conseil	1	français	1	société	1
crédit	1	groupement	1	soit	1
de	9	hausse	1	union	2
des	4	indiqué	1	vendredi	1
droits	3	investissement	2	vote	3
du	2	l	1		
désormais	1	la	1		

Figure 5.1 : *Exemple d'un texte et de son vecteur associé. Les composantes du vecteur sont simplement les occurrences des mots du texte.*

Représentation des textes avec des racines lexicales

Dans la description du modèle précédent, chaque flexion d'un mot est considérée comme un descripteur différent ; en particulier, les différentes formes d'un verbe sont autant de mots. Par exemple, dans le texte de la Figure 5.1, les mots *franchi* et *franchit* sont considérés comme des descripteurs différents alors qu'il s'agit de deux formes conjuguées du même verbe qui ont *a priori* le même sens.

Pour remédier à ce problème, il est possible de considérer uniquement la racine des mots plutôt que les mots entiers (on parle de *stem* en anglais). Plusieurs algorithmes ont été proposés pour substituer les mots par leur racine ; l'un des plus connus pour la langue anglaise est l'algorithme de Porter [Porter, 1980][1]. Nous n'avons pas utilisé un tel algorithme sur les corpus en langue française.

[1] Il est possible de trouver des implémentations rapides et efficaces de cet algorithme sur le web : http://www.muscat.com/~martin/stem.html

La Figure 5.2 est un exemple de texte du corpus Reuters dans sa version d'origine et la Figure 5.3 présente le même texte dont tous les mots ont été remplacés par leur racine grâce à l'algorithme de Porter.

```
TEXAS COMMERCE BANCSHARES &lt;TCB> FILES PLAN
Texas Commerce Bancshares Inc's Texas
Commerce Bank-Houston said it filed an application with the Comptroller of
the Currency in an effort to create the largest banking network in Harris
County.
    The bank said the network would link 31 banks having 13.5 billion dlrs
in assets and 7.5 billion dlrs in deposits.
```

Figure 5.2 : *Exemple de texte du corpus Reuters.*

```
TEXA COMMERC BANCSHAR &LT;TCB> FILE PLAN
Texa Commerc Bancshar Inc's Texa
Commerc Bank-Houston said it file an applic with the Comptrol of the
Currenc in an effort to creat the largest bank network in Harri Counti.
    The bank said the network would link 31 bank have 13.5 billion dlr in
asset and 7.5 billion dlr in deposit.
```

Figure 5.3 : *Texte précédent dont les mots ont été remplacés par leur racine.*

Il existe néanmoins d'autres algorithmes que celui de Porter pour déterminer les racines lexicales ; une comparaison entre différents algorithmes a été menée dans [Hull, 1996].

Représentation des textes avec des lemmes

La lemmatisation consiste à utiliser l'analyse grammaticale afin de remplacer les verbes par leur forme infinitive et les noms par leur forme au singulier. La lemmatisation est donc plus compliquée à mettre en œuvre que la recherche de racines, puisqu'elle nécessite une analyse grammaticale des textes. Un algorithme efficace, nommé *TreeTagger*[1] [Schmidt, 1994], a été développé pour les langues anglaise, française, allemande et italienne. Cet algorithme utilise des arbres de décision pour effectuer l'analyse grammaticale, puis des fichiers de paramètres spécifiques à chaque langue.

La Figure 5.4 et la Figure 5.5 montrent deux exemples, l'un en français et l'autre en anglais, de textes dont les mots d'origine ont été remplacés par leur lemme (l'algorithme remplace les nombres par le signe @card@).

[1] Les publications relatives à cet algorithme ainsi que les codes source sont disponibles sur le site : http://www.ims.uni-stuttgart.de/projekte/corplex/DecisionTreeTagger.html

```
TEXAS COMMERCE BANCSHARES LT TCB> FILE PLAN
Texas Commerce Bancshares inc's Texas
Commerce Bank Houston say it file an application with the
comptroller of the currency in an effort to create the large
banking network in Harris County
    The bank say the network would link bank have
@card@ billion dlrs in asset and @card@ billion dlrs in deposit
```

Figure 5.4 : *Texte de la Figure 5.2 dont les mots ont été remplacés par leur lemme.*

```
Marionnaud: Union et Etudes Investissement franchir @card@ de+le droit de
vote
le société union et Etudes investissement (caisse National de Crédit
Agricole avoir franchir en hausse le seuil de @card@ de+le droit de vote
de+le groupement français de parfumerie Marionnaud et détenir désormais
@card@ action être @card@ de+le capital et @card@ de+le droit de vote avoir
indiquer vendredi le conseil de+le marché financier.
Ce franchissement de seuil résulter de l'acquisition de @card@ action,
préciser le CMF.
```

Figure 5.5 : *Texte de la Figure 5.1 dont les mots ont été remplacés par leur lemme.*

La substitution des mots par leur racine ou leur lemme réduit l'espace des descripteurs et permet de représenter par un même descripteur des mots qui ont le même sens. Par exemple, le remplacement des mots *bank*, *banks*, *banking* dans le texte de la Figure 5.3 par l'unique racine *bank* semble être avantageux tout comme le remplacement des formes conjuguées *franchit* et *franchi* par le lemme *franchir* dans le texte de la Figure 5.5.

Néanmoins ces substitutions peuvent augmenter l'ambiguïté des descripteurs en représentant par un même descripteur des mots avec des sens différents (des exemples seront donnés au chapitre 7). Même le simple remplacement de la forme plurielle d'un mot par sa forme singulier peut augmenter l'ambiguïté d'un mot comme dans la Figure 5.5 où *actions* est représenté par le descripteur *action*. Dans un contexte économique, en effet, le mot *actions* se réfère le plus souvent à des actions des entreprises et n'a rien à voir avec le concept *action* employé par exemple dans la phrase : "le domaine d'*action* du gouvernement".

Il n'est pas possible de savoir *a priori* quelle représentation conduira aux meilleures performances et des expériences sont effectuées aux chapitres 7 et 8 afin de déterminer la représentation la mieux adaptée à la mise en œuvre des apprentissages numériques.

Dans toute la suite de ce chapitre, chaque mot est considéré comme un descripteur différent.

5.2 Etude de la fréquence des mots sur un corpus : la loi de Zipf

5.2.1 Enoncé de la loi de Zipf

La distribution de l'occurrence des mots dans un corpus de texte donné n'est pas uniforme : certains mots apparaissent très fréquemment, tandis que d'autres apparaissent très rarement. Les mots les plus fréquents en français sont les mots grammaticaux comme *le, la, les, et, … .* Sur le corpus Reuters, les cinq mots qui apparaissent le plus fréquemment sont : *the, of, to, in, and.*

La distribution de fréquence des mots dans un corpus a été étudiée empiriquement par Zipf [Zipf, 1949] et les résultats de cette analyse sont connus sous le nom de loi de Zipf. Pour énoncer cette loi, Zipf est parti d'un principe général qu'il a ensuite énoncé mathématiquement. Si l'on considère un corpus contenant T textes et que l'on note $TF(m, t)$ l'occurrence d'un mot m dans un texte t, on peut définir $CF(m)$, l'occurrence totale du mot m sur le corpus T :

$$CF(m) = \sum_{t \in T} TF(m,t)$$

Si l'on classe ensuite l'ensemble des mots du corpus par ordre décroissant d'occurrence totale, on obtient pour chaque mot un rang $r(m)$. La loi formulée par Zipf s'écrit alors :

$$CF(m) \cdot r(m) = K_T$$

K_T est une constante qui dépend du corpus. Cette relation peut s'écrire également :

$$Log(r(m)) = Log(K_T) - Log(CF(m))$$

Cette dernière relation, montre que si l'on trace le logarithme de l'occurrence en fonction du logarithme du rang, on doit obtenir une droite de pente –1.

La Figure 5.6 montre la vérification expérimentale de la loi de Zipf sur le corpus Reuters. On compte 51427 termes différents sur ce corpus ; à partir du rang 6243, les termes ont une fréquence totale sur l'ensemble du corpus inférieure à trente.

En fait, il est fréquent, comme le montre la Figure 5.6, que la loi ne soit pas très bien vérifiée pour les hautes fréquences et les basses fréquences, et il existe différentes méthodes pour corriger cette loi dans les domaines où elle n'est plus tout à fait vérifiée [Manning et Schütze, 1999]. Cependant, cette loi reflète bien le comportement général de la distribution des

occurrences : il existe un petit nombre de mots très fréquents, il existe un grand nombre de mots très rares n'apparaissant qu'une fois ou deux sur le corpus et il existe tout un ensemble de mots dont la fréquence d'apparition se situe entre ces deux domaines.

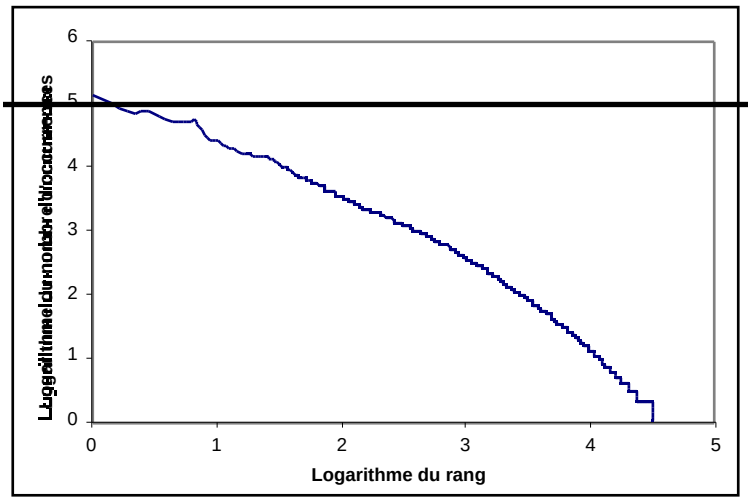


Figure 5.6 : *Vérification expérimentale de la loi de Zipf sur le corpus Reuters.*

5.2.2 Conséquences de la loi de Zipf

Comme le nombre de mots présents dans un corpus peut être très grand, les méthodes statistiques cherchent, en général, à réduire le nombre de mots utilisés pour représenter les textes. Nous verrons dans la suite comment s'effectue cette opération, mais les observations du paragraphe précédent permettent d'effectuer une première réduction de la dimension de l'espace de descripteurs.

Il s'agit de supprimer les mots dont on sait *a priori* qu'ils ne seront pas utiles pour les algorithmes d'apprentissage. Cette étape est critique, car les mots supprimés lors de cette étape le sont définitivement et il ne faut donc pas supprimer de mots importants.

La distribution de fréquences décrite par la loi précédente a deux conséquences importantes pour la représentation des textes.

5.2.2.1 Suppression des mots fréquents

Les mots qui apparaissent le plus souvent dans un corpus sont, comme on l'a vu précédemment, les mots grammaticaux ou les mots de liaisons. Ces mots doivent être supprimés de la représentation des textes pour deux raisons :

- d'un point de vue linguistique, ces mots ne comportent que très peu d'informations. La présence ou l'absence de ces mots n'aident pas à deviner le sens d'un texte. Pour cette raison, ils sont communément appelés mots vides (ou *stop words* en anglais).
- d'un point de vue statistique, ces mots se retrouvent sur l'ensemble des textes sans aucune discrimination et ne sont d'aucune aide pour la classification.

Comme le nombre de mots concernés est faible, il est possible de définir une liste de mots qui sont automatiquement supprimés de la représentation.

Par exemple [Sahami, 1998] définit une liste de 570 mots courant en anglais, plus une liste de 100 mots très fréquents sur le web, pour supprimer les mots les plus courants.

Cependant, l'établissement d'une telle liste peut poser des problèmes. D'une part, il n'est pas facile de déterminer le nombre de mots exacts qu'il faut inclure dans cette liste. D'autre part, cette liste est intimement liée à la langue utilisée et n'est donc pas transposable directement à une autre langue.

5.2.2.2 Suppression des mots rares

En général, les auteurs cherchent également à supprimer les mots rares d'un corpus afin de réduire de façon appréciable la dimension des vecteurs utilisés pour représenter les textes, puisque, d'après la loi de Zipf, ces mots rares sont très nombreux. D'un point de vue linguistique, la suppression de ces mots n'est pas nécessairement justifiée : certains mots peuvent être très rares, mais très informatifs. Néanmoins, ces mots ne peuvent pas être utilisés par des méthodes à bases d'apprentissage du fait de leur très faible occurrence.

Une des méthodes communément retenues pour supprimer ces mots consiste à ne considérer que les mots l'occurrence totale est supérieure à un seuil fixé préalablement.

Nous exposons dans le paragraphe suivant une méthode de détermination du vocabulaire spécifique d'un thème permettant de d'écarter automatiquement les mots rares et les mots fréquents sans utiliser de liste de mots prédéfinis. Cette méthode présente en outre l'avantage d'être transposable automatiquement du français à l'anglais, et d'être adaptée à la classification.

5.3 Une méthode automatique de détermination du vocabulaire spécifique

Cette méthode a été développée dans le cadre de l'application Exoweb présentée au chapitre 9.

Lorsque que l'on dispose d'un ensemble de textes pertinents pour un thème donné, on cherche à trouver le vocabulaire spécifique de ce thème, constitués par les termes que partagent ces textes parmi les mots ni trop fréquents, ni trop rares. Cette méthode se décompose en deux étapes : la première étape élimine de chaque texte les mots fréquents, la deuxième étape élimine les mots rares.

5.3.1 Elimination des mots fréquents

Si l'on note, comme précédemment, *TF(m, t)* l'occurrence d'un mot *m* dans un texte *t,* et *CF(m)* l'occurrence de ce mot sur le corpus, on calcule pour chaque mot d'un texte *t* le rapport :

$$R\,(m,t) = \frac{TF\,(m,t)}{CF\,(m)}$$

Les mots du texte sont classés par ordre décroissant de ce rapport. Plus le mot *m* est fréquent dans le corpus, plus le ratio est faible et, inversement, plus un mot est rare, plus le ratio est élevé. Dans le cas limite où un mot n'apparaît qu'une seule fois dans le corpus, ce ratio vaut 1 et le mot est classé en première place.

On supprime la deuxième moitié de la liste des mots de la représentation du texte, si bien que cette représentation ne contient plus les mots fréquents du corpus.

A la fin de cette étape, chaque texte *t* est représenté par un vecteur $\overrightarrow{v(t)}$ avec un codage booléen (1 si le mot est présent et 0 s'il est absent).

La Figure 5.7 montre l'application de cette méthode au texte de la Figure 5.1. On retrouve bien dans les premiers mots sélectionnés (le haut de la colonne de gauche) les mots les plus rares avec en tête le nom propre *marionnaud*. La fin de la liste (le bas de la colonne de droite) montre que les mots grammaticaux sont automatiquement supprimés. Les mots de la colonne de gauche sont conservés, et ceux de la colonne de droite sont supprimés.

marionnaud	désormais
études	union
parfumerie	actions
investissement	précise
franchissement	conseil
financiers	capital
franchit	ce
résulte	français
marchés	soit
groupement	société
franchi	vendredi
caisse	hausse
nationale	indiqué

agricole	la
seuil	et
cmf	des
vote	le
droits	du
détient	de
acquisition	a
crédit	en

Figure 5.7 : *Représentation du texte : la colonne de gauche montre les mots conservés pour la représentation et la colonne de droite montre les mots supprimés.*

5.3.2 Elimination des mots rares

Comme on l'a souligné précédemment, il est également nécessaire de supprimer de la représentation les mots qui apparaissent peu sur le corpus. Pour cela, on calcule la somme des vecteurs sur l'ensemble des textes dont on veut trouver le vocabulaire spécifique. On considère un sous-ensemble *S* du corpus *T* tel que tous les textes de *S* soient relatifs à un même thème. On définit le vocabulaire spécifique d'un thème en considérant la somme vectorielle suivante :

$$\vec{V}_s = \sum_{t \in S} \overrightarrow{v(t)}$$

Si l'on classe maintenant les composantes du vecteur obtenu par ordre décroissant, on obtient une liste de mots classés de telle sorte que les mots rares sont en fin de liste et les mots fréquents en haut de la liste.

5.3.3 Conclusion sur la méthode et exemples de mise en œuvre

La méthode proposée ci-dessus supprime automatiquement les mots rares et les mots fréquents d'un corpus sans utiliser une liste externe de mots liés à une langue particulière.
Cette méthode détermine le vocabulaire commun d'un sous-ensemble de textes d'un corpus ; les mots choisis ont la particularité de figurer dans le milieu de la distribution de la loi de Zipf.

La Figure 5.8 présente les dix premiers mots de quatre listes de vocabulaire spécifique obtenus sur des corpus différents ; ces listes ont été construites à partir de quatre ensemble de textes pertinents, chaque ensemble définissant un thème.
Les deux premiers ensembles ont été obtenus grâce à l'application ExoWeb présentée au chapitre 9 et sont composés de dépêches de l'AFP. Ces ensembles correspondent aux thèmes *participations* qui traite des échanges de participations entre entreprises (1400 dépêches pertinentes) et au thème *inforoute* qui traitent des informations sur les nouvelles technologies

et notamment l'internet (1000 dépêches pertinentes). Le troisième thème, *coffee*, est issu du corpus Reuters (111 documents pertinents). Le dernier thème est le thème 351 du corpus TREC-8 (31 documents pertinents) et traite de la recherche de pétrole au large des îles malouines (*Falkland petroleum exploration*).

Participation		*Inforoute*		*Coffee*		*351*	
422	détient	603	internet	98	coffee	28	islands
335	capital	229	accès	53	quotas	28	argentina
328	participation	202	site	49	ico	24	argentine
295	actionnaire	181	ligne	40	producers	21	aires
226	actionnaires	175	électronique	39	organization	21	buenos
223	actions	170	web	32	quota	19	sovereignty
192	cession	169	sites	32	producer	17	malvinas
181	parts	159	www	31	bags	17	exploration
177	vote	141	utilisateurs	30	export	16	oil
176	acquisition	135	com	27	colombia	14	islanders

Figure 5.8 : *Exemples des dix premiers mots trouvés sur trois thèmes.*

A ce stade, les résultats ne sont pas évalués quantitativement, mais dans tous les cas, les listes de mots semblent caractéristiques de la description du thème. Il est intéressant de noter que ces exemples correspondent à trois corpus différents (AFP, Reuters et Financial Times), avec des langues différentes et un nombre variable de documents pertinents, et que dans tous ces cas, la méthode semble fiable.

Une évaluation quantitative est effectuée au paragraphe 5.5.3 sur d'autres exemples.

La Figure 5.9 reprend la courbe de la de Zipf présentée précédemment et fait apparaître avec des cercles les dix premiers mots trouvés pour la catégorie *coffee* par la méthode précédente. Les mots trouvés se situent bien dans le milieu de la distribution.

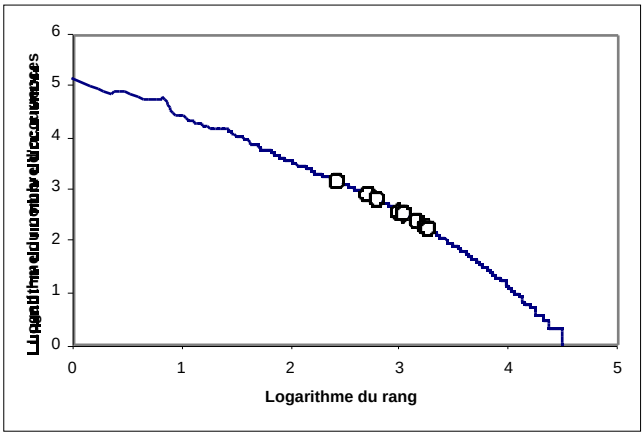


Figure 5.9 : *Répartition des mots spécifiques de la catégorie coffee dans la loi de Zipf.*

5.4 Codage des termes

Une fois les composantes des vecteurs choisies pour représenter un texte, il faut décider comment coder chaque coordonnée du vecteur. Si *TF(m, t)* est l'occurrence d'un terme *m* dans un texte *t*, la composante d'un vecteur est codée *f(TF(m, t))*, où la fonction *f* doit être déterminée.

Dans la représentation de la Figure 5.1, la fonction *f* a été choisie égale à l'identité ; ce qui représente le choix le plus simple.

Il est également possible de choisir la fonction booléenne définie par :

$$f = \begin{cases} 1 \text{ si } TF(m,t) > 0 \\ 0 \text{ si } TF(m,t) = 0 \end{cases}$$

C'est cette fonction qui a été utilisée dans le paragraphe 5.3.2 pour éliminer les mots rares lors de la détermination du vocabulaire spécifique.

Néanmoins, cette fonction est rarement utilisée pour les méthodes statistiques, car ce codage supprime de l'information qui peut être utile : l'apparition du même mot plusieurs fois dans un texte peut constituer un élément de décision important.

5.4.1 Le codage *tf.idf*

Le codage "tf.idf" a été introduit dans le cadre du modèle vectoriel présenté au chapitre 2 et donne parfois son nom à la méthode vectoriel. Le codage utilise une fonction de l'occurrence multipliée par une fonction qui fait intervenir l'inverse du nombre de documents différents dans lequel un terme apparaît. Ce sigle provient de l'anglais et signifie : *term frequency * inverse document frequency*.

Il est admis que l'occurrence d'un terme est une information importante, mais cependant, on considère généralement que la fonction *f* ne doit pas être l'identité. Si, par exemple, un mot apparaît deux fois dans un texte, son importance n'est pas nécessairement deux fois plus grande que s'il n'apparaissait qu'une seule fois. Pour cette raison, si l'occurrence n'est pas nulle, la fonction souvent utilisée est de la forme :

$$1 + Log\left(TF(m,t)\right)$$

En fait, les mots-clefs ont tendance à apparaître plusieurs fois dans un document pertinent : ainsi, deux mots peuvent avoir la même d'occurrence sur le corpus, mais l'un des deux peut apparaître dans beaucoup plus de documents que l'autre. Pour compenser cet effet, le modèle vectoriel utilise souvent le codage appelé *tf.idf* qui consiste à choisir une fonction telle que :

$$f = \begin{cases} (1 + Log(TF(m,t))).Log\dfrac{N}{DF(m)} & si\ TF(m,t) > 0 \\ 0 & si\ TF(m,t) = 0 \end{cases}$$

DF(m) est le nombre de documents différents où le terme *m* apparaît au moins une fois et *N* est le nombre de documents contenus dans le corpus. Ainsi, si un terme apparaît dans tous les documents, son poids a une valeur nulle puisqu'il n'apporte aucune information.

Des variations de ce codage ont été proposées dans [Salton et Buckley, 1990].

5.4.2 Un codage efficace : le codage Lnu

Les différents textes qui composent un corpus ont des tailles différentes dont il faut tenir compte dans le codage des termes.

Selon Singhal [Singhal, 1996], il existe deux phénomènes à considérer dans les textes longs par rapport aux textes courts :

- les mots présents tendent à avoir des fréquences plus élevées,
- les textes longs sont plus susceptibles de contenir des mots-clefs différents.

Il propose le codage Lnu, défini à la Figure 5.10, pour tenir compte de ces deux phénomènes ; ce codage tient également compte des remarques faites lors paragraphe précédent et est inspiré du codage tf.idf.

$$\text{Lnu} = L * u$$

$$L = \frac{1 + Log\left(TF(m,t)\right)}{1 + Log\left(\overline{TF}(m)\right)}$$

$$u = \frac{1}{0.8 + 0.2\,\dfrac{U(t)}{\bar{U}}}$$

Figure 5.10 : *Définition des termes du codage Lnu.*

$\overline{TF}(m)$: Fréquence moyenne dans le texte *t.*

$U(t)$: Nombre de termes uniques dans le texte *t.*

$\bar{U}$: Nombre moyen de termes sur l'ensemble des textes du corpus.

Ce codage est utilisé avec succès dans [Singhal, 1998] et [Ng *et al*., 2000].

5.5 Sélection de descripteurs pour la catégorisation de textes

5.5.1 Nécessité de la sélection de descripteurs

Les considérations précédentes ont permis de définir des critères pour représenter les textes en vecteur, mais il faut ensuite choisir plus spécifiquement les descripteurs qui vont être utilisés comme vecteurs d'entrées des modèles obtenus par apprentissage.

Comme dans tout le reste de cette étude, le filtrage est considéré comme un problème de classification à deux classes ; chaque thème nécessite une représentation différente des textes et un choix différent de descripteurs. Comme pour d'autres problèmes de classification, certains descripteurs peuvent être non discriminants pour la tâche que l'on cherche à résoudre. Par exemple, pour un problème de ciblage de clientèle pour un magasin, la couleur des yeux des clients a peu de chance d'être une caractéristique très pertinente pour construire un modèle.

Les entrées non discriminantes doivent être supprimées pour deux raisons différentes :

- Pour les modèles tels que les réseaux de neurones, le nombre de poids du réseau croît linéairement avec le nombre de descripteurs utilisés en entrée du modèle. Donc plus le nombre d'entrées est grand, plus le nombre de paramètres à déterminer est élevé : il faut alors disposer d'une base d'exemples plus grande afin d'avoir une bonne estimation de ces paramètres, même si les méthodes de régularisation présentées au chapitre 6 permettent de pallier partiellement ce problème[1].

- Comme les bases d'apprentissage ne sont pas infinies, (on verra qu'elles sont souvent de petite taille), des corrélations fortuites peuvent apparaître entre un descripteur non

[1] Il faut noter néanmoins que, pour les modèles linéaires par rapport à leurs paramètre tels que les polynômes, la nécessité de sélectionner les descripteurs est encore plus grande puisque le nombre de paramètres varie exponentiellement avec le nombre de descripteurs.

informatif et les individus d'une classe ; elles peuvent avoir une influence négative sur la qualité du modèle.

Les méthodes de sélection de descripteurs ont donc pour but de choisir parmi un ensemble de descripteurs possibles, les "bons" descripteurs, c'est-à-dire ceux qui vont permettre d'obtenir de bonnes performances sur une base différente de la base d'apprentissage.

La problématique de la sélection de descripteurs est très générale, mais le traitement du langage naturel présente des spécificités que nous allons présenter ci-dessous.

5.5.2 Spécificité de la sélection de descripteurs pour le filtrage de documents

Pour la problématique du filtrage de documents avec le modèle vectoriel, l'ensemble des descripteurs potentiels est constitué de l'ensemble des mots du corpus, ce qui peut représenter plusieurs centaines de milliers d'individus sur un corpus de taille raisonnable. Même si l'on a montré que les mots les plus fréquents et les plus rares pouvaient être éliminés facilement, soit parce qu'ils n'étaient pas discriminants, soit parce qu'ils n'étaient pas exploitables statistiquement, le nombre de candidats reste de plusieurs milliers. Parmi l'ensemble des descripteurs restants, on peut penser que tous ne sont pas nécessairement discriminants pour un thème donné, et que certains peuvent être très corrélés.

On cherche donc à supprimer ces mots de la représentation des textes, tout en sachant que chaque suppression de mot entraîne une perte d'information ; il faut trouver le bon compromis entre, d'une part, la nécessité de réduire l'espace des descripteurs et, d'autre part, le besoin de garder suffisamment d'information.

L'une des difficultés du filtrage provient du fait qu'il existe beaucoup de tournures différentes pour exprimer le même concept ou la même idée. La Figure 5.11 montre des exemples de passages pertinents pour le thème *participation*, qui traite des échanges de participations entre entreprises. Cet exemple montre la variété du vocabulaire employé, et la nécessité de conserver suffisamment de descripteurs, sachant que la présence d'une seule de ces phrases suffit à rendre le texte pertinent pour ce thème.

```
Telia est sur le point d'être privatisé, un tiers de son capital devant
être introduit en Bourse

BSCH a porté sa participation de 4,89% à 5,1% dans la banque allemande
Commerzbank
```

KPN a annoncé mercredi qu'il allait **céder les 21% qu'il détient** dans son homologue irlandais Eircom

Une compagnie nationale dont la **majorité de contrôle reste détenue** par l'état.

Tennessee **détenait 12,64% des actions** et **11,24% des droits de vote**.

L'entrée du Crédit Agricole **au capital** du Crédit Lyonnais comme **premier actionnaire** (avec 10%).

Figure 5.11 : *Exemples de phrases pertinentes pour le thème participation.*

La sélection de descripteurs est une étape primordiale de la construction d'un filtre, car, quel que soit le modèle statistique utilisé ultérieurement, si la représentation des textes n'inclut pas certains descripteurs ou si les descripteurs retenus sont trop nombreux ou mal choisis, le filtre aura des performances médiocres.

5.5.3 Les méthodes de sélection de descripteurs

5.5.3.1 Principe des méthodes

La méthode idéale consisterait à tester tous les sous-ensembles possibles de descripteurs afin de conserver l'ensemble donnant les meilleurs résultats sur une base de test. Une telle solution n'est évidemment pas possible, car si l'on considère un ensemble de p candidats, le nombre d'ensembles à tester s'élève à 2^p, donc si l'ensemble initial comporte cent descripteurs, le nombre de combinaisons s'élève à environ 10^{30} ce qui est évidemment beaucoup trop grand pour être réalisable.

Parmi les méthodes testées ici, deux approches différentes sont utilisées ; toutes deux tiennent compte de la tâche que l'on cherche à accomplir : différencier les textes pertinents des textes non pertinents.

La première approche consiste à calculer un score pour chaque descripteur, indépendamment des autres, en s'appuyant sur les statistiques d'apparition et d'absence du descripteur en fonction de la classe à laquelle appartiennent les textes. Les descripteurs sont ensuite classés selon ce score, les descripteurs en tête de liste étant les plus discriminants pour distinguer les textes pertinents des textes non pertinents. Les méthodes de l'information mutuelle et du chi-2 exposées ci-après reposent toutes deux sur ce principe.

La deuxième approche est constructive : elle construit itérativement un modèle, en partant d'un ensemble vide et en ajoutant successivement de nouveaux descripteurs en tenant compte des

descripteurs déjà sélectionnés. Cette construction est faite en utilisant l'algorithme d'orthogonalisation de Gram-Schmidt.

Pour toutes ces méthodes le résultat se présente sous la même forme : il s'agit d'une liste de mots ordonnés du plus discriminant au moins discriminant.

5.5.3.2 La méthode du chi-2

La statistique du χ^2 (chi-2) mesure l'indépendance entre un descripteur *t* et un thème *T*. Cette mesure a été utilisée pour la sélection des descripteurs dans [Schütze *et al.*, 1995] et [Wiener *et al.*, 1995]. Le calcul nécessite de construire pour chaque descripteur *t* du corpus le tableau de contingences de la Figure 5.12 :

	Descripteur *t* présent	Descripteur *t* absent	
Thème *T* présent	*a*	*c*	$T_1 = a+c$
Thème *T* absent	*b*	*d*	$T_0 = b+d$
	$D_1 = a+b$	$D_0 = c+d$	$N = a+b+c+d$

Figure 5.12 : *Tableau de contingences pour l'absence ou la présence d'un descripteur.*

On définit :

$$\chi^2(t,T) = \frac{N(ad - cb)^2}{(a + c)(b + d)(a + b)(c + d)}$$

Si un descripteur *t* et le thème *T* sont totalement indépendants, alors *t* apparaît avec la même fréquence dans le sous-ensemble des textes pertinents et dans le sous-ensemble des textes non pertinents, ce qui se traduit par (ad = bc) et la valeur de $\chi^2(t,T)$ est nulle.

A l'inverse, si le descripteur *t* apparaît systématiquement dans l'ensemble des textes pertinents et jamais dans l'ensemble des textes non pertinents, on a c = b = 0 et $\chi^2(t,T)$ vaut N, ce qui est sa valeur maximale. Cette valeur est également atteinte si un descripteur apparaît systématiquement dans l'ensemble des textes non pertinents et jamais dans l'ensemble des textes pertinents.

Entre ces deux valeurs extrêmes, plus la valeur de $\chi^2(t,T)$ est grande, plus t et T sont liés. Les descripteurs du corpus sont donc classés par ordre décroissant de $\chi^2(t,T)$, les plus discriminants figurant en tête de liste.

5.5.3.3 La méthode de l'information mutuelle

L'information mutuelle est employée pour mesurer la quantité d'information apportée par la présence ou l'absence d'un descripteur dans un document. Cette mesure a été fréquemment utilisée pour la catégorisation de textes pour effectuer la sélection de descripteurs [Lewis, 1992] [Mouliner, 1997] [Dumais *et al.*, 1998].

Dans le cas d'une classification à deux classes notées C_0 et C_1, si l'on note $P(t=1)$ la probabilité de présence d'un descripteur t dans un document et $P(t=0)$ son événement complémentaire, l'information apportée par la présence ou l'absence de t se mesure par :

1

La première somme fait intervenir les probabilités *a priori* de chaque classe ; elle est indépendante des descripteurs et n'est donc pas prise en considération. En reprenant les notations du tableau de la Figure 5.12, on calcule pour chaque descripteur t la quantité (sans tenir compte du terme constant) :

$$G'(t) = \frac{D_0}{N}\left(\frac{d}{D_0}\log\left(\frac{d}{D_0}\right) + \frac{c}{D_0}\log\left(\frac{c}{D_0}\right)\right) + \frac{D_1}{N}\left(\frac{a}{D_1}\log\left(\frac{a}{D_1}\right) + \frac{b}{D_1}\log\left(\frac{b}{D_1}\right)\right)$$

Les descripteurs sont ensuite classés par ordre décroissant de valeurs de G' ; comme les termes constants ont été supprimés, la valeur maximale de G' est 0. Cette valeur est obtenue pour un descripteur qui apparaît dans tous les textes pertinents et dans aucun texte non pertinent ou vice-versa, la formule étant symétrique.

5.5.3.4 Une méthode constructive : l'orthogonalisation de Gram-Schmidt

Cette méthode est issue des méthodes utilisées pour trouver la solution des moindres carrés d'un problème linéaire par rapport à ses paramètres. Une description détaillée de cet algorithme et son application à la sélection de modèle NARMAX peut être trouvée dans

[Chen *et al.*, 1989]. Cette procédure itérative classe les descripteurs par ordre décroissant d'importance tout en tenant compte de ceux déjà classés.

Elle a été utilisée avec succès en classification pour sélectionner les entrées de réseaux de neurones pour un problème de classification des collectivités locales en fonction d'un ensemble de ratio financiers [Stoppiglia, 1997] ou pour déterminer les caractéristiques les plus discriminantes d'un fichier client, pour une application dont le but est de mieux cibler les clients potentiels pour un nouveau produit [Stricker et Haré, 1998].

Il existe deux techniques de mise en œuvre de l'algorithme de Gram-Schmidt. La méthode dite "classique" est économe en termes d'occupation de la mémoire, mais elle est très sensible aux erreurs d'arrondi contrairement à la méthode dite "modifiée" qui est numériquement plus stable [Björck, 1967]. Précisons que ces méthodes seraient strictement équivalentes en l'absence d'erreurs d'arrondi. Puisque la taille de la mémoire des machines à notre disposition le permet, nous utilisons l'algorithme de Gram-Schmidt modifié, qui assure la stabilité numérique.

Le principe de cette méthode itérative est de choisir à chaque itération, le "meilleur" descripteur puis de supprimer l'influence de ce descripteur sur les descripteurs restants.

La mise en œuvre est décrite ci-dessous :

Soit Q le nombre de descripteurs candidats et N le nombre d'exemples d'apprentissage avec leurs Q descripteurs possibles et leur sortie associée (qui vaut +1 si l'exemple est pertinent et –1 s'il ne l'est pas). Notons $X_i = {}^T [x_1^i, x_2^i, ..., x_N^i]$ le vecteur des réalisations pour le descripteur i et $Y = {}^T [y_1, ..., y_N]$ le vecteur de dimension N contenant les sorties à modéliser. Le codage des descripteurs est une fonction de la fréquence du descripteur *i*. Soit la matrice X (N, Q)

$$X = \begin{matrix} x_1^1 & \dots & x_1^j & \dots & x_1^Q \\ \dots & & \dots & & \\ \dots & & x_j^j & \dots & = [X_1, \dots X_Q] \\ \dots & & \dots & & \\ x_N^1 & & x_N^j & \dots & x_N^Q \end{matrix}$$

Le modèle s'écrit $Y=X\theta$, la matrice X étant la matrice des entrées et le vecteur Y le vecteur de sortie.

A la première itération de l'algorithme, il faut trouver le vecteur d'entrée le plus corrélé avec la sortie. Pour cela, on calcule le carré des cosinus des angles entre le vecteur de sortie et les vecteurs d'entrée selon la formule :

$$\cos^2(X_P, Y) = \frac{(X_P^T.Y)^2}{(X_P^T.X_P).(Y^T.Y)}$$

Le vecteur sélectionné est celui pour lequel cette quantité est maximale. On élimine ensuite la contribution de l'entrée sélectionnée en projetant le vecteur de sortie ainsi que tous les vecteurs d'entrée restants sur le sous-espace orthogonal au vecteur sélectionné.

La procédure se poursuit en choisissant, une nouvelle fois, le vecteur d'entrée projeté qui explique le mieux la sortie projetée. La procédure se termine lorsque tous les vecteurs d'entrées ont été ordonnés.

La mise en œuvre de cette méthode nécessite la mise en mémoire de la matrice X ; pour que les calculs soient réalisables en un temps raisonnable, il est nécessaire de limiter la taille de cette matrice. Cette méthode doit donc être précédée d'une première sélection de descripteurs pour limiter l'espace de recherche. La sélection de descripteurs se fait donc uniquement dans l'espace des descripteurs sélectionnés lors de la détermination du vocabulaire spécifique, expliquée au paragraphe 5.3.

On cherche donc dans le vocabulaire spécifique d'un thème les descripteurs les plus discriminants.

5.5.3.5 Comparaison des approches

Avant de voir la mise en œuvre de ces différentes méthodes sur des exemples pratiques de catégorisation de textes, nous faisons quelques remarques sur leurs différences.

Utilisation des fréquences

Les méthodes du chi-2 et de l'information mutuelle reposent toutes les deux sur un décompte des apparitions des mots. Ces deux mesures se fondent uniquement sur la présence ou l'absence d'un mot dans un document sans prendre en considération l'occurrence. À l'opposé, la méthode d'orthogonalisation de Gram-Schmidt utilise explicitement l'occurrence des mots dans un texte ou une fonction de cette occurrence.

Prise en considération des corrélations

La méthode de Gram-Schmidt peut être qualifiée de constructive : le modèle est construit progressivement en partant d'un ensemble vide de descripteurs puis en les ajoutant un à un en tenant compte des descripteurs préalablement sélectionnés.

Donc, contrairement aux deux méthodes statistiques, elle tient compte des corrélations éventuelles entre les descripteurs. Dans le cas limite où deux descripteurs t_1 et t_2 sont systématiquement associés, ils sont sélectionnés tous les deux par les méthodes de l'information mutuelle et du chi-2 avec le même score, alors qu'un seul de ces deux descripteurs est sélectionné par la méthode de Gram-Schmidt. Or la présence de descripteurs redondants dans le classifieur ajoute un paramètre inutile, et risque de dégrader les performances.

Mots négatifs et positifs

Les formules de calcul de l'information mutuelle et du chi-2 sont totalement symétriques pour la classe des documents pertinents et celle des documents non pertinents. Plus précisément, cela signifie que certains descripteurs peuvent être choisis parce qu'ils sont caractéristiques des documents pertinents, et d'autres parce qu'ils sont caractéristiques des documents non pertinents.

Dans toute la suite, les descripteurs caractéristiques des textes pertinents sont appelés *mots positifs* et les descripteurs caractéristiques des textes non pertinents sont appelés *mots négatifs*. Cette dénomination vient du fait que si l'on considère un modèle linéaire du type $Y=X\theta$ (la matrice X étant la matrice des descripteurs et le vecteur Y étant le vecteur de sortie avec le codage +1 pour les textes pertinents et –1 pour les textes non pertinents), alors les composantes de θ associées aux mots dits positifs seront positives et les composantes associées aux mots dits négatifs seront négatives.

La présence de mots positifs dans un texte tend donc à indiquer que ce texte est pertinent, et la présence de mots négatifs tend à indiquer qu'il n'est pas pertinent.

En revanche, dans notre implémentation de l'algorithme de Gram-Schmidt nous ne prenons comme descripteur potentiel dans la matrice X que les mots issus du vocabulaire spécifique du thème donc, par construction, des mots positifs. Il faut noter cependant que rien n'interdit d'ajouter dans la matrice X des descripteurs négatifs afin qu'ils puissent éventuellement être

sélectionnés par la méthode de Gram-Schmidt ; l'impact de l'utilisation de ces mots est étudié au paragraphe 5.5.6.

Défauts communs

Chacune de ces méthodes sélectionne les descripteurs pour leur pouvoir discriminant, mais deux descripteurs peuvent avoir un pouvoir discriminant très faible pris séparément, alors que la présence simultanée de ces deux descripteurs peut avoir un rôle important. Pour le thème des *participations*, par exemple, les deux descripteurs *droits* et *vote* ne sont pas des descripteurs très intéressants pris séparément, mais l'interaction de ces deux mots forme le concept de *droits de vote* qui a un sens très précis. Aucune des méthodes présentées ci-dessus ne répond à ce problème.

L'interaction éventuelle entre plusieurs variables peut être prise en considération dans la méthode d'orthogonalisation de Gram-Schmidt en ajoutant des polynômes. Par exemple, au lieu de se contenter de construire la matrice X avec uniquement x_1 et x_2 il est possible d'ajouter le produit $x_1.x_2$ dans la matrice. Dans le cas de la sélection de descripteurs pour le filtrage de textes, le nombre de candidats potentiels est très grand, donc le nombre de monômes l'est également. De plus il est nécessaire de tenir compte de la distance entre les mots : *droits* et *vote* peuvent être dans le même texte sans que l'association *droits de vote* ne soit présente.

Finalement, même avec l'orthogonalisation de Gram-Schmidt, il est difficile de prendre en considération les interactions éventuelles ; néanmoins nous verrons au chapitre 8 comment ce problème est résolu au moins partiellement.

Notons enfin que ces méthodes ne tiennent pas compte des synonymes de la langue ; si un descripteur représentant un mot a été sélectionné dans la liste des descripteurs, ses synonymes n'y figurent pas.

5.5.3.6 Exemple de mise en œuvre des différentes méthodes

Les trois méthodes décrites ci-dessus ont été mises en œuvre sur trois thèmes du corpus Reuters, afin de comparer leur comportement. Les thèmes étudiés sont *interest*, *oilseed* et *nat-gas*. La Figure 5.13 montre les quinze meilleurs mots sélectionnés par chaque méthode ; les mots figurant en gras dans ces listes sont des mots négatifs.

Il n'est pas possible de juger de la qualité d'une méthode à partir de la liste des mots sélectionnés, mais il est intéressant de constater qu'il existe peu de différences entre toutes ces listes. Globalement, toutes ces méthodes semblent conduire aux même choix de descripteurs.
On peut remarquer la présence du mot *lt* dans la liste des mots sélectionnés par l'information mutuelle : il s'agit en fait d'une marque utilisée dans le corpus pour indiquer un nom d'entreprise. Pour les catégories étudiées ici, il s'agit clairement d'un mot négatif.

La Figure 5.14 montre l'évolution des performances sur la base de test pour chaque modèle en fonction du nombre de descripteurs retenus pour la représentation des textes. Le modèle utilisé est un simple neurone logistique avec un paramètre de régularisation fixé à la valeur 1.0 comme expliqué au chapitre 7. Le nombre de descripteurs varie par incrément de 10, entre 10 et 500.

Vocabulaire spécifique	Information mutuelle	Chi-2	Gram-Schmidt
money	rate	rate	rate
rates	bank	money	customer
england	money	rates	money
rate	rates	bank	prime
prime	**pct**	england	band
point	**lt**	lending	rates
discount	market	prime	england
fed	england	repurchase	dollar
stg	prime	market	discount
k	lending	discount	mcentee
repurchase	repurchase	customer	cuts
funds	**cts**	**pct**	advances
lending	point	point	repurchase
bills	discount	fed	leaves
effective	**company**	band	floating

Vocabulaire spécifique	Information mutuelle	Chi-2	Gram-Schmidt
soybean	soybean	soybean	soybean
soybeans	soybeans	soybeans	soybeans
tonnes	**lt**	rapeseed	rapeseed
usda	agriculture	oilseeds	oilseed
corn	corn	oilseed	oilseeds
agriculture	usda	crushers	seaforth
grain	tonnes	corn	inspections
crop	rapeseed	agriculture	romero
rapeseed	grain	usda	sorghum
shipment	oilseeds	sunflower	peanuts
oilseeds	wheat	bushels	rouen
grains	oilseed	meal	asa
farmers	crushers	crushing	disappearance
oilseed	**u**	sorghum	agriculture
bought	crop	inspections	hrw

Vocabulaire spécifique	Information mutuelle	Chi-2	Gram-Schmidt
gas	gas	cubic	gas
natural	natural	gas	cubic
cubic	cubic	natural	butane
feet	feet	feet	exploration
barrels	oil	barrels	favored
exploration	barrels	exploration	cameron
energy	exploration	trillion	natural
reserves	production	liquids	elecetric
petroleum	energy	oil	pel>
oil	reserves	condensate	nmot
pipeline	petroleum	thousand	alaskan
drilling	trillion	discoveries	waterflooding
offshore	drilling	proven	gop>
crude	offshore	proved	stalon
barrel	liquids	offshore	dependency

Figure 5.13 : *Listes des quinze premiers mots sélectionnés par chaque méthode pour les thèmes interest, oilseed et nat-gas. Les mots négatifs sont en gras.*

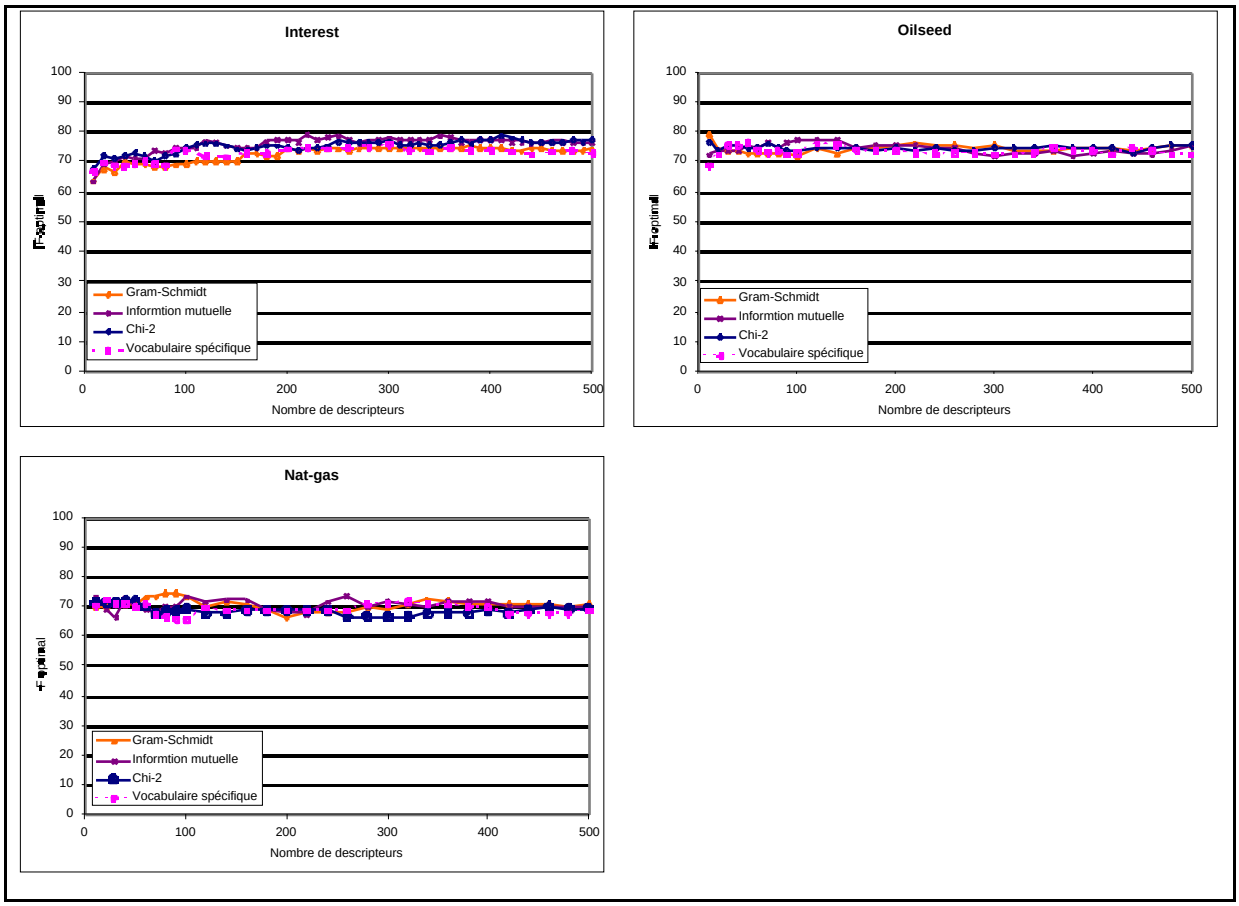


Figure 5.14 : *Évolution des performances sur la base de test en fonction du nombre de descripteurs utilisés pour la représentation des textes. La mesure de performance utilisée est la mesure de F optimal sur la base de test définie au chapitre 4.*

Ces courbes montrent qu'au-delà de dix descripteurs, les performances n'évoluent presque plus, ce qui suggère que peu, voire très peu, de descripteurs suffisent, et qu'il existe un nombre optimal de descripteurs à retenir. Avec les dix premiers descripteurs trouvés par chaque méthode, les performances sur ces trois catégories sont déjà très proches de l'optimum.

Des expériences ont également été effectuées en augmentant encore le nombre de descripteurs utilisés : soit les performances stagnent, soit elles décroissent.

Sur ces catégories, aucune méthode ne semble être largement supérieure ou inférieure aux autres, et, malgré leurs différences, ces méthodes donnent des résultats assez proches, y compris la méthode du vocabulaire spécifique qui est pourtant la plus élémentaire.

Des expériences supplémentaires sont nécessaires avant d'affirmer la supériorité d'une méthode par rapport une autre. Nous les présenterons au chapitre 7.

5.5.4 Utilisation d'un critère d'arrêt pour choisir le nombre de descripteurs

5.5.4.1 Faut-il utiliser un critère d'arrêt ?

Les courbes obtenues à la Figure 5.14 suggèrent qu'au-delà d'un certain nombre de descripteurs, il n'est plus utile d'ajouter de nouveaux descripteurs, car les performances sur la base de test n'augmentent plus. Il semble donc utile d'utiliser un critère d'arrêt afin de limiter le nombre de descripteurs retenus pour la représentation des textes : le nombre de paramètres du modèle sera moins élevé et tous les calculs d'optimisation et de traitement seront plus rapides. Cependant, lorsque l'optimum est atteint, les performances ne décroissent pratiquement pas ; par conséquent, le nombre de descripteurs à retenir ne semble pas être un paramètre très critique et n'a pas besoin d'être déterminé très précisément.

Si l'on appelle k le nombre de descripteurs retenus, beaucoup d'auteurs se contentent de choisir simplement k *a priori* [Lewis, 1992] [Mouliner, 1997] [Dumais *et al.*, 1998] en fixant par exemple $k = 50$ ou $k = 100$.

Une solution extrême consiste à n'utiliser aucun critère d'arrêt : l'ensemble des descripteurs est utilisé. Ainsi, dans ses expériences sur le corpus Reuters, [Joachims, 1998] conserve 9947 descripteurs correspondant à 9947 mots différents pour ses modèles utilisant les machines à vecteurs supports. Ce nombre correspond à l'ensemble des descripteurs disponibles sur le corpus en utilisant des racines lexicales et après suppression des mots vides, et des mots apparaissant dans moins de trois documents. Ce cas correspond à un cas critique où aucune sélection de descripteurs n'est faite. Pour prouver son assertion, il classe les 9947 mots selon leur information mutuelle moyenne pour la catégorie *acquisition (acq)* et il choisit comme entrées, soit les 200 premiers descripteurs de la liste, soit les descripteurs de rang 201 à 500, puis de 501 à 1000, puis de 1001 à 2000, puis de 2001 à 4000, et enfin de 4001 à 9947. Dans son expérience, même avec la dernière partie de la liste donc avec les descripteurs les moins discriminants, il obtient des performances qui sont nettement meilleures que le hasard. Il en déduit donc que, même vers la fin de la liste, il reste des descripteurs discriminants et que, comme il existe peu de chance que tous ces descripteurs soient redondants avec les descripteurs du début de la liste, il est nécessaire de les conserver tous.

Dans une autre étude, [Yang et Perdersen, 1997] étudient l'impact de plusieurs sélections de descripteurs sur deux modèles différents : les k plus proches voisins et une méthode de régression fondée sur les moindres carrés. Selon leurs résultats, lorsque le nombre de descripteurs utilisés devient trop élevé, les performances de leurs classifieurs diminuent ou n'augmentent plus. Leurs modèles utilisent environ 300 descripteurs.

5.5.4.2 Utilisation d'un critère d'arrêt

Dans notre approche, un critère d'arrêt est couplé à la méthode de d'orthogonalisation de Gram-Schmidt afin de déterminer automatiquement le nombre de descripteurs optimal pour chaque thème.

Cette opération est effectuée en utilisant un vecteur aléatoire qui est classé par la méthode de Gram-Schmidt exactement comme les autres descripteurs. Les descripteurs classés après ce vecteur aléatoire sont considérés comme non pertinents pour le problème posé.

Dans la pratique, le rang de ce vecteur aléatoire est en fait un nombre aléatoire. Il faut donc calculer la fonction de distribution de probabilité de l'angle entre un vecteur aléatoire et le vecteur de sortie. Le calcul de la probabilité qu'un vecteur aléatoire soit plus pertinent que l'un des n descripteurs sélectionnés après n itérations a été développé dans [Stoppiglia, 1997]. Chaque fois qu'un descripteur est sélectionné, on peut calculer la probabilité qu'un descripteur aléatoire soit plus pertinent que ce descripteur, et au-delà d'un seuil prédéfini (typiquement 1% ou 5%), tous les descripteurs sont éliminés.

Lors de l'implémentation, cette probabilité est calculée juste après la sélection d'un descripteur et, si le seuil est dépassé, la procédure s'arrête automatiquement sans chercher à classer les descripteurs restants. Donc, même si le nombre de descripteurs candidats est élevé, la procédure s'achève et aucun calcul inutile n'est effectué.

5.5.5 Comparaison des performances des méthodes de sélection de descripteurs

5.5.5.1 Description de l'expérience

Les méthodes de l'information mutuelle, de l'orthogonalisation de Gram-Schmidt et du vocabulaire spécifique sont testées sur un ensemble de thèmes pour comparer leurs performances relatives. Dans une première série d'expériences le nombre de descripteurs

retenus en entrée du classifieur est choisi arbitrairement, et, dans une deuxième série, le critère d'arrêt décrit au paragraphe 5.5.4.2 est couplé à la méthode d'orthogonalisation de Gram-Schmidt.

Pour ces expériences, le modèle neuronal est identique à celui utilisé pour les expériences du paragraphe 5.5.3.6.

Les expériences sont effectuées sur les thèmes 1 à 40 du corpus Reuters[1]. Le nombre de documents pertinents sur la base d'apprentissage varie de 2877 pour le thème 1 (thème *earn)* à 24 pour le thème 40 (thème *sorghum*). Ces quarante thèmes permettent de comparer les différentes méthodes de sélection de descripteurs sur des thèmes comportant beaucoup d'exemples pertinents comme sur des thèmes en comportant peu.

La méthode du chi-2 a été éliminée des comparaisons, car

- il a été montré que les résultats obtenus par cette méthode étaient très proches de ceux obtenus par la méthode de l'information mutuelle [Yang et Perdersen, 1997] et cette tendance a été vérifiée sur les trois thèmes de la Figure 5.14,
- selon cette même étude, la méthode de l'information mutuelle semble être légèrement plus performante.

De plus, on souhaite comparer des méthodes avec des approches différentes et les deux méthodes de l'information mutuelle et du chi-2 reposent sur les mêmes principes. Les trois différences majeures entre les trois méthodes retenues sont :

- la prise en considération de mots négatifs pour l'information mutuelle,
- la prise en considération des corrélations éventuelles avec l'algorithme de Gram-Schmidt,
- l'utilisation explicite de la fréquence d'apparition des descripteurs pour la méthode de Gram-Schmidt.

[1] Les thèmes sont ordonnés selon le nombre de documents pertinents sur la base d'apprentissage selon la présentation du corpus faite au chapitre 3.

5.5.5.2 Comparaison sans optimisation du nombre de descripteurs retenus

Pour cette première série d'expériences, le nombre de descripteurs retenus pour chaque méthode (le paramètre noté *k*) est fixé à 50, 100, ou à 200.

Pour chacun des thèmes, les performances sont mesurées avec la mesure de *F* optimale décrite au chapitre 4 sur l'ensemble de test et les macro-moyennes sont calculées sur l'ensemble des thèmes. La Figure 5.15 présentent les performances obtenues pour chacune des méthodes sur les thèmes 1 à 40 et sur les deux sous-ensembles de thèmes 1 à 20 et 21 à 40 pour observer d'éventuelles corrélations entre le comportement d'une méthode et le nombre de documents pertinents sur la base d'apprentissage.

	Information mutuelle		
	k = 50	*k* = 100	*k* = 200
Thèmes 1 à 40	81,9	81,2	80,3
Thèmes 1 à 20	83,0	83,4	82,8
Thèmes 21 à 40	80,8	79,0	77,7

	Gram-Schmidt		
	k = 50	*k* = 100	*k* = 200
Thèmes 1 à 40	81.6	81.4	81,1
Thèmes 1 à 20	82,3	82,2	82,5
Thèmes 21 à 40	80,8	80,7	79,8

	Vocabulaire spécifique		
	k = 50	*k* = 100	*k* = 200
Thèmes 1 à 40	81,3	81,2	81,1
Thèmes 1 à 20	82,2	82,8	82,5
Thèmes 21 à 40	80,4	79,6	79,8

Figure 5.15 : *Macro-moyennes sur les thèmes 1 à 40, 1 à 20, 21 à 40 du corpus Reuters pour chaque méthode de sélection de descripteurs. k est le nombre de descripteurs utilisés en entrée du classifieur.*

Ces résultats montrent que les trois méthodes de sélection de descripteurs testées conduisent à des classifieurs dont les performances sont très proches en moyenne ; ces méthodes semblent

être globalement équivalentes. La méthode du vocabulaire spécifique donne des résultats similaires aux deux autres méthodes bien qu'elle soit beaucoup plus simple.

Comme l'avaient suggéré les courbes de la Figure 5.14, le nombre de descripteurs retenus n'est pas une valeur très critique puisque, quelle que soit la méthode de sélection, les performances sont peu affectées par la valeur de *k* ; cependant, à performance égale, il est toujours préférable de choisir des valeurs de *k* faibles afin d'obtenir des modèles comportant moins de paramètres ajustables par apprentissage.

5.5.5.3 Utilisation du critère d'arrêt

Le critère d'arrêt est couplé à la méthode de Gram-Schmidt comme expliqué au paragraphe 5.5.3.4 pour déterminer le nombre optimal de descripteurs pour chaque thème, avec un seuil de 1%. Les résultats sur les mêmes thèmes que précédemment sont présentés Figure 5.16.

	Gram-Schmidt + critère d'arrêt
Thèmes 1 à 40	81,5
Thèmes 1 à 20	82,3
Thèmes 21 à 40	80,7

Figure 5.16 : *Résultats avec la méthode de Gram-Schmidt couplée avec le critère d'arrêt.*

Les résultats sont là encore très proches de ceux trouvés à la Figure 5.15 ; l'utilisation du critère d'arrêt n'améliore pas les performances, mais, sans les dégrader, elle permet de déterminer automatiquement le nombre de descripteurs retenus.

5.5.6 Impact des mots négatifs pour la méthode de Gram-Schmidt

Comme expliqué précédemment, l'utilisation de mots négatifs est l'une des différences entre l'information mutuelle et l'utilisation de la méthode de Gram-Schmidt choisie ici.

Il est également possible d'utiliser des mots négatifs par la méthode de Gram-Schmidt : il suffit pour cela de définir le vocabulaire spécifique de l'ensemble des documents non pertinents pour chaque thème. On dispose alors, pour chaque thème, de deux listes de mots : une liste du vocabulaire spécifique des documents pertinents, et une liste du vocabulaire spécifique des documents non pertinents.

On effectue deux séries d'expériences sur les thèmes précédents, qui se distinguent uniquement par la construction de la matrice *X* en entrée de l'algorithme de Gram-Schmidt :

1. La matrice *X* est construite à partir des 200 premiers mots de la liste du vocabulaire spécifique de l'ensemble des documents pertinents.
2. La matrice *X* est construite à partir des 200 premiers mots de la liste du vocabulaire spécifique de l'ensemble des documents pertinents plus les 50 premiers mots de la liste du vocabulaire spécifique de l'ensemble des documents non pertinents.

Donc, pour la première expérience, seuls des mots positifs peuvent être choisis, et, pour la seconde, l'algorithme peut choisir soit des mots positifs soit des mots négatifs.

Les résultats sur les thèmes 1 à 40 du corpus Reuters sont présentés à la Figure 5.17 (les résultats de l'expérience prenant en considération uniquement les mots positifs sont différents de ceux présentés à la Figure 5.16, car le choix des documents non pertinents utilisés pour fabriquer les bases d'apprentissage est légèrement différent).

	Gram-Schmidt mots positifs	Gram-Schmidt mots positifs et négatifs
Thèmes 1 à 40	82,1	82,2
Thèmes 1 à 20	82,7	82,8
Thèmes 21 à 40	81,6	81,7

Figure 5.17 : *Comparaison des résultats entre la méthode de sélection ne prenant en compte que les mots positifs et celle prenant en compte les mots positifs et négatifs.*

Les résultats obtenus sont identiques avec la méthode qui prend en compte les mots négatifs et les mots positifs, ce qui laisse à penser que la méthode de Gram-Schmidt ne considère pas les mots négatifs comme discriminants. La Figure 5.18 qui montre la comparaison catégories par catégories entre les deux méthodes confirme qu'il n'existe pratiquement aucune différence entre les deux expériences.

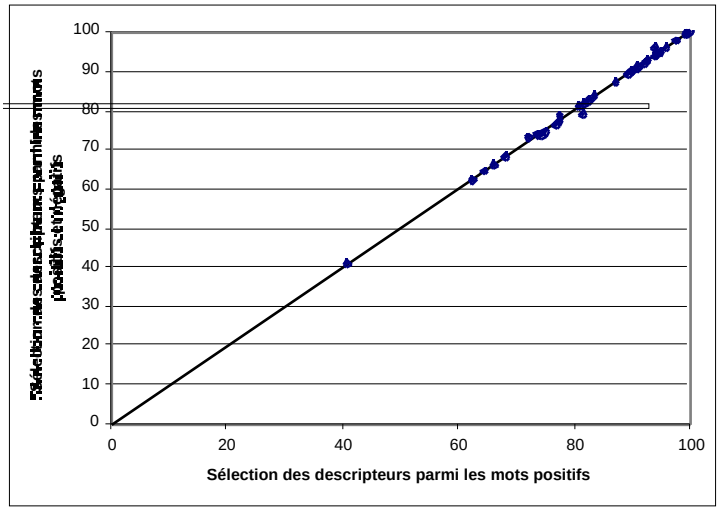


Figure 5.18 : *Comparaison catégorie par catégorie.*

Il est important de noter que sur le corpus Reuters, la catégorie *earn* représente 2877 documents sur la base d'apprentissage, et 1087 documents pertinents sur la base de test, soit à peu près 33% de la base de test. De plus, le vocabulaire utilisé par les textes de cette catégorie est assez particulier, comme le montre la liste des vingt premiers mots de la liste du vocabulaire spécifique de ce thème présentée à la Figure 5.19.

```
revs, th, shr, note, div, prior, loss, profit, dividend, shrs, avg, qtly,
qtr, includes, mths, jan, sets, vs, gain, nine
```

Figure 5.19 : *Liste des vingt premiers mots du vocabulaire spécifique de la catégorie earn.*

La Figure 5.20 montre la liste du vocabulaire négatif pour les trois thèmes *interest, oilseed* et *nat-gas* obtenue en cherchant le vocabulaire spécifique des documents non pertinents.

Interest		*Oilseed*		*Nat-gas*	
418	revs	390	revs	389	revs
373	th	374	th	370	th
299	note	285	note	287	note
292	shr	279	shr	277	shr
266	prior	251	prior	259	prior
262	div	241	div	255	div
246	dividend	235	loss	239	loss
234	profit	229	dividend	230	profit
228	shrs	212	profit	225	shrs
224	includes	211	manager	218	avg

Figure 5.20 : *Listes du vocabulaire négatif pour les trois thèmes interest, oilseed et nat-gas.*

Ces deux remarques signifient que le corpus Reuters présente la particularité de se prêter *a priori* à l'utilisation de mots négatifs puisqu'une catégorie prédomine largement, et que, de plus, elle utilise un vocabulaire spécifique comme les mots *revs*, *vs*, *shr*, *qtr* qui sont des abréviations rarement utilisées dans un contexte différent de celui de la catégorie *earn ;* le texte Figure 5.21 est un exemple de l'utilisation de ces abréviations.

```
UNIVERSAL HOLDING CORP 4TH QTR NET
Shr NA Net profit 2,000 vs profit 195,000 Revs 2,623,000 vs 2,577,000 Year
Shr NA Net loss 425,000 vs profit 278,000 Revs 15.4 mln vs 8,637,000
```

Figure 5.21 : *Exemple de texte appartenant à la catégorie earn.*

Pour le corpus utilisé dans TREC-8, le thème le plus dense contient 186 documents pertinents sur la base de test, ce qui représente une densité de l'ordre de 0,1% ; il est donc peu probable que le fait de savoir qu'un document n'appartient pas à cette catégorie constitue une information pour trouver les autres thèmes.

5.6 Conclusion

Ce chapitre a permis d'introduire la représentation la plus utilisée pour la catégorisation de textes et le niveau d'information principalement utilisé : la présence ou l'absence de certains mots pour trouver automatiquement le sens d'un texte.

Le très grand nombre de descripteurs potentiels rend nécessaire l'utilisation d'une méthode de sélection. Nous avons introduit une méthode originale par rapport à la littérature de la catégorisation de textes : la méthode d'orthogonalisation de Gram-Schmidt précédée de la détermination du vocabulaire spécifique d'un ensemble de textes.

Dans les expériences effectuées, toutes ces méthodes se sont révélées à peu près équivalentes ; néanmoins, dans la suite, nous utiliserons la méthode d'orthogonalisation de Gram-Schmidt couplée à la détermination du vocabulaire spécifique pour plusieurs raisons :

- On montrera au chapitre 7, qu'en modifiant la construction de la matrice X, il est possible d'améliorer les performances.
- Elle tient compte des corrélations éventuelles entre les mots, et, dans le cas limite où deux mots apparaissent systématiquement ensemble, la méthode de Gram-

Schmidt, contrairement aux autres méthodes, ne sélectionne qu'un seul de ces deux mots.

- Elle tient compte des mots précédemment sélectionnés.
- Elle utilise la fréquence des mots dans les textes et non pas uniquement la présence ou l'absence de mots.
- L'utilisation d'un critère d'arrêt optimise le nombre de descripteurs pour chaque thème. Il est également possible d'utiliser des critères d'arrêt avec les méthodes de l'information mutuelle et du chi-2, mais la justification théorique de ces critères semble moins bien établie.
- L'utilisation de mots dits négatifs ne semble pas apporter d'amélioration, même sur le corpus Reuters qui a pourtant une configuration particulière du fait de la répartition des catégories.

Cette méthode présente néanmoins des défauts, comme les deux autres méthodes ; les deux principaux semblent être :

- Le système ne tient pas compte des synonymes.
- Deux mots peuvent avoir un pouvoir discriminant très faible pris séparément et n'être pas sélectionnés alors que l'union de ces deux mots a un pouvoir discriminant fort. Sur le thème *participation*, par exemple, les deux mots *entrée* et *hauteur* pris séparément ne sont pas de bons descripteurs relativement à ce thème, mais l'expression *entrée à hauteur* est souvent utilisée dans des phrases dans un contexte de participation comme le montre la phrase ci-dessous :

  Toyata a annoncé son entrée à hauteur de 5,4% dans le capital de Yamaha.

Néanmoins l'ensemble des expériences a montré que la détermination du vocabulaire spécifique qui est beaucoup plus simple permet d'obtenir de bons résultats, et des comparaisons supplémentaires seront effectuées au chapitre 7.

Chapitre 6 Apprentissage des réseaux de neurones et régularisation

Après une introduction rapide aux réseaux de neurones et à la problématique de la classification, l'essentiel de ce chapitre est consacré à l'apprentissage, et notamment aux problèmes liés au surapprentissage dans les problèmes de classification. Nous montrons que dans certains cas, les "méthodes actives" comme la régularisation par le *weight decay* sont indispensables pour limiter le surapprentissage. Cette technique exige néanmoins la détermination de paramètres supplémentaires, appelés hyperparamètres. L'approche bayesienne propose une solution de principe à cette détermination, que nous présentons dans ce chapitre, et dont nous décrirons l'application dans les chapitres suivants.

6.1 Problématique de la classification supervisée

6.1.1 La catégorisation de textes est un problème de classification supervisée

Le problème du filtrage de textes pour un thème donné est abordé dans ce mémoire comme un problème de classification supervisée à deux classes : la classe des textes pertinents et la classe des textes non pertinents. Pour construire un filtre relatif à un thème donné, il faut donc disposer d'exemples de chaque classe, préalablement étiquetés comme pertinents ou non pertinents. Grâce à ces deux ensembles de textes, il est possible de construire un classifieur grâce à un algorithme d'apprentissage. Si cet apprentissage est correctement réalisé, le modèle est capable d'estimer, pour chaque nouveau texte, sa probabilité de pertinence pour le thème considéré.

6.1.2 Théorème de Bayes

Le théorème de Bayes fournit un cadre théorique pour la problématique de la classification à deux classes, et il intervient également dans l'approche bayesienne exposée au paragraphe 6.6.
Si l'on considère un problème à deux classes C_1 et C_2, le théorème de Bayes permet de calculer les probabilités *a posteriori* connaissant les distributions des observations *a priori*.

$$P(C_1|x) = \frac{p(x|C_1)\, P(C_1)}{p(x)}$$

$P(C_1|x)$ est la probabilité *a posteriori* d'appartenir à la classe C_1 connaissant le vecteur des descripteurs x, $p(x|C_1)$ est la densité de probabilité du vecteur x dans la classe C_1, $P(C_1)$ est la probabilité *a priori* de la classe C_1 et $p(x)$ est la densité de probabilité non conditionnelle définie par :

$$p(x) = p(x|C_1)\,P(C_1) + p(x|C_2)\,P(C_2)$$

Dans le cas d'un problème de classification, cette formule définit une règle de décision : la probabilité de mauvaise classification est minimisée en sélectionnant la classe qui a la plus grande probabilité *a posteriori.*

Ce théorème est au cœur de la problématique de la classification : on peut distinguer (i) les méthodes de classification qui essayent de modéliser les densités de probabilités pour calculer les probabilités *a priori*, et (ii) les méthodes qui essayent de modéliser directement les probabilités *a posteriori.* Le détail de ces différentes méthodes peut être trouvé dans [Bishop, 1995] ou [Stoppiglia, 1997] ; les réseaux de neurones utilisés dans ce mémoire appartiennent à la deuxième catégorie.

6.2 Généralités sur les réseaux de neurones

Cette partie est une présentation succincte des principales propriétés des réseaux de neurones. L'accent est surtout mis sur les algorithmes utilisés et sur le problème du surajustement. Une présentation plus générale des réseaux de neurones et de leurs applications à d'autres tâches que la classification de textes peut être trouvée dans [Dreyfus *et al.*, 1999].

6.2.1 Le neurone formel

Un neurone formel est une fonction algébrique paramétrée, à valeurs bornées, de variables réelles appelées entrées.

En règle générale, le calcul de la valeur de cette fonction peut se décomposer en deux étapes :

· une combinaison linéaire des entrées :

$$v = w_0 + \sum_{i=1}^{n} w_i.x_i$$

Les w_i sont appelés poids synaptiques ou simplement **poids**, w_0 est appelé **biais**. Le biais peut être considéré comme la pondération de l'entrée 0 fixée à 1. v est appelé **potentiel** du neurone.

· La sortie du neurone est :

$$y = f(v) = f(\sum_{i=0}^{n} w_i.x_i)$$

La fonction f est la fonction **d'activation** du neurone. Dans la suite de ce mémoire, on considérera trois types de fonctions d'activation :

La fonction **identité** : $f(v) = v$.

*La fonction **sigmoïde** :* $f(v) = \tanh(v)$. C'est une fonction bornée à valeurs réelles comprises entre -1 et +1.

La fonction **logistique** : $f(v) = 1/(1 + \exp(-v))$. C'est une fonction bornée à valeurs réelles comprises entre 0 et 1.

6.2.2 Réseaux de neurones non bouclés

Un réseau de neurones non bouclé est une composition de fonctions réalisée par des neurones formels interconnectés entre eux. Certaines applications peuvent nécessiter plusieurs sorties (dans le cas d'une classification à plusieurs classes par exemple), mais dans notre cas, tous les réseaux utilisés ont une seule sortie.

Les possibilités d'arrangements entre les neurones sont multiples. La configuration la plus classique est appelée *perceptron multicouche*. Dans cette architecture, les neurones sont organisés en couches comme le montre la Figure 6.1 : une couche intermédiaire entre les entrées et les sorties appelée couche cachée et un neurone (ou une couche de neurones) de sortie. Les connexions se font d'une couche à la suivante sans qu'il y ait de connexion entre couches non adjacentes. Cette architecture est également appelée réseau à deux couches puisqu'il y a deux couches de poids ajustables : celle qui relie les entrées aux neurones cachés et celle qui relie les neurones cachés au neurone de sortie.

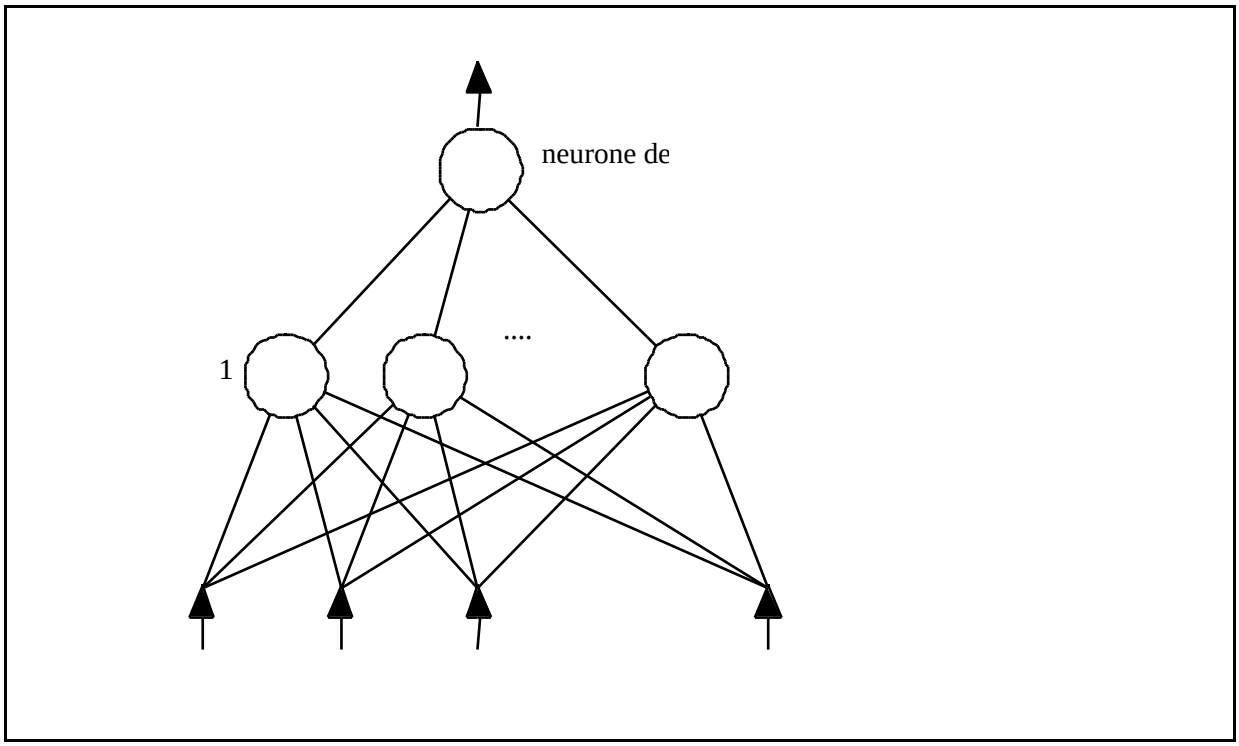


Figure 6.1 : *Réseau à couches avec N_e entrées, N_c neurones cachés et un neurone de sortie.*

Les neurones de la couche cachée sont appelés *neurones cachés*. Une fois l'architecture à deux couches choisie, il faut fixer le nombre de neurones cachés. Plus ce nombre est élevé, plus le nombre de degrés de liberté est élevé et plus la fonction modélisée par le réseau de neurone peut être complexe.

La Figure 6.2 montre deux exemples de fonctions réalisées par un réseau de neurones ; la partie gauche est obtenue avec un réseau comportant deux neurones cachés et la partie droite avec un réseau comportant dix neurones cachés. Dans le deuxième cas, la fonction obtenue comporte plus de degrés de liberté.

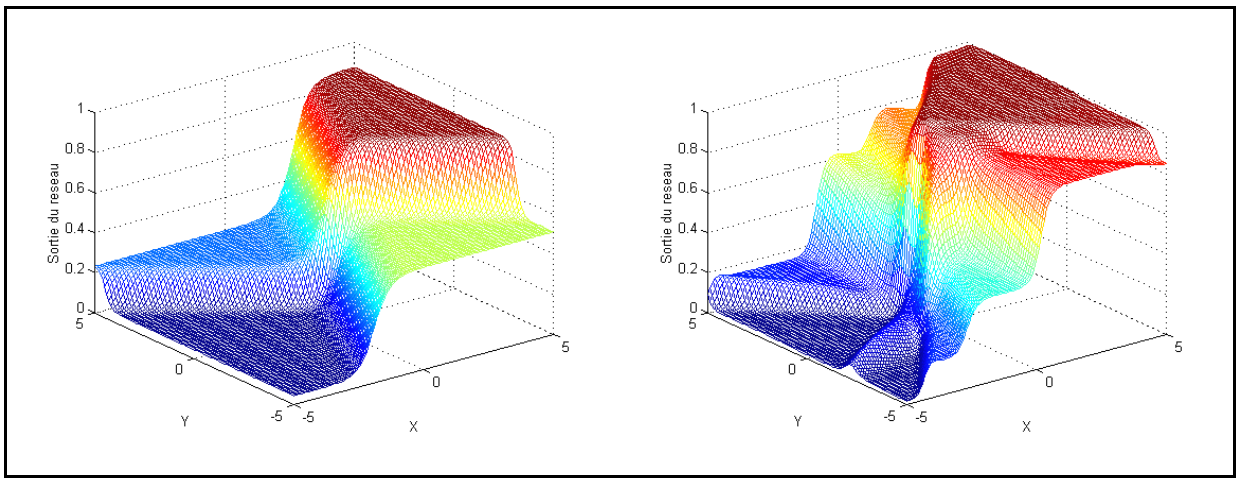


Figure 6.2 : *Réseau à deux couches avec deux entrées et un biais, les fonctions d'activations des neurones cachés sont des fonctions sigmoïdes et la sortie est une fonction logistique. Le quadrant de gauche est la sortie d'un réseau de neurones à deux neurones cachés, le quadrant de droite est la sortie d'un réseau à dix neurones cachés. Les poids de la première couche sont choisis aléatoirement dans l'intervalle [-4 ; +4], les poids de la deuxième couche sont choisis aléatoirement dans l'intervalle [-2 ; +2].*

6.2.3 Propriétés des réseaux de neurones

Les réseaux de neurones à couches, présentés au paragraphe précédent, ont la propriété générale d'être des approximateurs universels parcimonieux. Il s'agit en fait de deux propriétés distinctes détaillées ci-dessous.

6.2.3.1 La propriété d'approximation universelle

La propriété d'approximation universelle a été démontrée par [Cybenko, 1989] et [Funahashi, 1989] et peut s'énoncer de la façon suivante :

Toute fonction bornée suffisamment régulière peut être approchée uniformément, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire.

Cette propriété justifie l'utilisation de l'architecture présentée précédemment. Comme le montre ce théorème, le nombre de neurones cachés doit être choisi convenablement pour obtenir la précision voulue.

6.2.3.2 La propriété de parcimonie

Lorsque l'on cherche à modéliser un processus à partir des données, on s'efforce toujours d'obtenir les résultats les plus satisfaisants possibles avec un nombre minimum de paramètres ajustables. Dans cette optique, [Hornik *et al*., 1994] ont montré que :

Si le résultat de l'approximation (c'est-à-dire la sortie du réseau de neurones) est une fonction non linéaire des paramètres ajustables, elle est plus parcimonieuse que si elle est une fonction linéaire de ces paramètres. De plus, pour des réseaux de neurones à fonction d'activation sigmoïdale, l'erreur commise dans l'approximation varie comme l'inverse du nombre de neurones cachés, et elle est indépendante du nombre de variables de la fonction à approcher. Par conséquent, pour une précision donnée, donc pour un nombre de neurones cachés donné, le nombre de paramètres du réseau est proportionnel au nombre de variables de la fonction à approcher.

Ce résultat s'applique aux réseaux de neurones à fonction d'activation sigmoïdale puisque la sortie de ces neurones n'est pas linéaire par rapports aux poids synaptiques. Cette propriété montre l'intérêt des réseaux de neurones par rapport à d'autres approximateurs comme les polynômes dont la sortie est une fonction linéaire des paramètres ajustables : pour un même nombre d'entrées, le nombre de paramètres ajustables à déterminer est plus faible pour un réseau de neurones que pour un polynôme. Cette propriété devient d'autant plus intéressante dans le cas du filtrage de textes car le nombre d'entrées est typiquement de l'ordre de plusieurs dizaines.

6.3 Apprentissage des réseaux de neurones

Une fois l'architecture d'un réseau de neurones choisie, il est nécessaire d'effectuer un apprentissage pour déterminer les valeurs des poids permettant à la sortie du réseau de

neurones d'être aussi proche que possible de l'objectif fixé. Dans le cas d'un problème de régression, il s'agit d'approcher une fonction continue, dans le cas d'un problème de classification supervisée, il s'agit de déterminer une surface de séparation.

Cet apprentissage s'effectue grâce à la minimisation d'une fonction, appelée *fonction de coût*, calculée à partir des exemples de la base d'apprentissage et de la sortie du réseau de neurones ; cette fonction détermine l'objectif à atteindre.

Dans les travaux présentés dans ce mémoire, nous avons effectué cette minimisation en deux temps : un algorithme de descente du gradient, simple à mettre en œuvre et efficace loin du minimum, commence la minimisation, puis une méthode de quasi-Newton, très efficace proche du minimum, la termine[1].

Dans la suite de ce chapitre, nous utiliserons les notations suivantes : N est le nombre d'exemples de la base d'apprentissage ; à chaque exemple i est associée sa classe t_i (codée +1 ou 0) ; chaque exemple est représenté par un vecteur x_i de dimension n ; les poids du réseau sont représentés par un vecteur w ; la sortie du réseau de neurones associée au vecteur d'entrée x_i est notée y_i.

6.3.1 Algorithmes de minimisation

6.3.1.1 Principe des algorithmes

Soit $J(w)$ la fonction de coût (le choix de la forme de cette fonction est expliqué au paragraphe 6.3.2). Les algorithmes utilisés nécessitent que $J(w)$ soit dérivable par rapport aux poids. Le principe de ces méthodes est de se placer en un point initial, de trouver une direction de descente du coût dans l'espace des paramètres w, puis de se déplacer d'un pas dans cette direction. On atteint un nouveau point et l'on itère la procédure jusqu'à satisfaction d'un critère d'arrêt. Ainsi, à l'itération k, on calcule :

$$w_k = w_{k-1} + \alpha_{k-1} . d_{k-1}$$

α_k est le pas de la descente et d_k est la direction de descente : les différents algorithmes se distinguent par le choix de ces deux quantités.

[1] La méthode de Levenberg-Marquardt, également très efficace, ne s'applique qu'aux fonctions de coût quadratiques, ce qui n'est pas le cas dans les travaux que nous présentons.

6.3.1.2 Descente du gradient

L'algorithme le plus simple consiste à choisir comme direction de descente l'opposé du gradient de la fonction de coût ($d_k = - J(w) = -$ Grad$(J(w_k))$). Cette méthode est efficace loin du minimum et permet uniquement de s'en approcher. Pour cette raison, la détermination du pas n'est pas cruciale : loin du minimum, il faut seulement vérifier que le pas n'est ni trop petit ni trop grand. En pratique, on utilise, selon les cas, deux méthodes :

· soit un asservissement par la norme du gradient :

$$\alpha_k = \frac{\alpha_0}{1 + \left\| J(w_k) \right|}$$

où α_0 est une constante qui vaut typiquement 0,01.

· soit la méthode de Goldstein [Minoux, 1983] pour laquelle le pas est adapté afin de satisfaire deux conditions :

1. $J(w_k + \alpha_k d_k) < J(w_k) + m_1 \alpha_k \quad J^T(w_k) d_k$
2. $J(w_k + \alpha_k d_k) > J(w_k) + m_2 \alpha_k \quad J^T(w_k) d_k$

La première condition s'assure que le pas choisi n'est pas trop grand (sinon l'algorithme risque d'avoir un comportement oscillatoire), alors que la deuxième s'assure qu'il n'est pas trop petit (sinon l'algorithme a une convergence très lente). Les valeurs habituelles pour les deux paramètres m_1 et m_2 sont respectivement 0,1 et 0,7.

Ces deux méthodes de recherche du pas sont "économiques", car elles ne demandent pas de calculs inutiles de gradient (seul celui dans la direction de descente est nécessaire).

6.3.1.3 La méthode de Newton

La méthode de Newton utilise la courbure (dérivée seconde) de la fonction de coût pour atteindre le minimum. La modification des paramètres s'écrit ainsi :

$$w_k = w_{k-1} - H_{k-1}^{-1} . \quad J\left(w_{k-1} \right)$$

La direction de descente est $- H_{k-1}^{-1} . \quad J(w_{k-1})$ où H_{k-1}^{-1} est l'inverse du hessien de la fonction de coût, et le pas est constant fixé à un.

Cet algorithme converge en une seule itération pour une fonction quadratique. C'est donc un algorithme qui est inefficace loin du minimum de la fonction et très efficace près du minimum.

Dans la pratique, le calcul du hessien et surtout de son inverse est à la fois complexe et source d'instabilités numériques ; on utilise de préférence une méthode de "quasi-Newton".

6.3.1.4 La méthode de quasi-Newton

Les méthodes de quasi-Newton consistent à approcher l'inverse du hessien plutôt que de calculer sa valeur exacte.

La modification des paramètres s'écrit :

$$w_k = w_{k-1} - \alpha_{k-1} . M_{k-1} . \quad J(w_{k-1})$$

La suite M_k est construite de façon à converger vers l'inverse du hessien avec M_0 égale à la matrice identité. Cette suite est construite grâce à la méthode dite BFGS [Broyden, 1970] [Flechter, 1970] [Goldfarb, 1970] [Shanno, 1970], dont la vitesse de convergence est beaucoup plus grande que celle de la méthode du gradient. De plus, elle est relativement insensible au choix du pas, qui peut être déterminé économiquement par la méthode de Goldstein.

6.3.1.5 Problème des minima locaux

Les minima trouvés par les algorithmes précédents sont des minima locaux. Le minimum trouvé dépend du point de départ de la recherche c'est-à-dire de l'initialisation des poids. En pratique, il faut effectuer plusieurs minimisations avec des initialisations différentes, pour trouver plusieurs minima et retenir le "meilleur". Il est néanmoins impossible et généralement inutile, de s'assurer que le minimum choisi est le minimum global. Les réseaux de neurones à couches présentent des symétries, si bien que l'on peut montrer que pour une architecture avec N_c neurones cachés, il existe $2^N c\, N_c!$ minima équivalents [Bishop, 1995].

6.3.2 Choix de la fonction de coût

Le choix de la fonction de coût est conditionné par l'objectif à atteindre.

6.3.2.1 Erreur quadratique

Pour les problèmes de régression, l'ensemble d'apprentissage est constitué d'exemples pour lesquels la sortie désirée t est une variable continue. La fonction de coût la plus utilisée est l'erreur quadratique sur la base d'apprentissage : elle consiste à minimiser la somme des carrés des erreurs entre la sortie du réseau et la valeur réelle de la sortie.

$$J(w) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i(w) - t_i\right)^2$$

Cette fonction de coût est issue du principe de maximum de vraisemblance avec une hypothèse gaussienne sur la distribution des sorties. Pour les problèmes de classification à deux classes, la sortie désirée est une variable binaire codée 1 ou 0 selon que l'exemple appartient respectivement à C_1 ou C_0. L'hypothèse gaussienne sur la distribution des sorties n'est alors clairement plus vérifiée. Cependant, si l'apprentissage est effectué en minimisant l'erreur quadratique, la sortie du réseau de neurones peut être interprétée comme la probabilité *a posteriori*, au sens du théorème de Bayes, d'appartenance à la classe C_1 [Richard et Lippman, 1991].

6.3.2.2 Entropie croisée

L'entropie croisée, comme l'erreur quadratique moyenne est issue du principe du maximum de vraisemblance. Comme l'hypothèse sous-jacente pour l'utilisation de l'erreur quadratique est erronée pour les problèmes de classification, un autre modèle est construit pour tenir compte de la spécificité du codage utilisé dans ces problèmes.

Considérons un problème de classification à deux classes C_1 et C_0 où les sorties t sont codées 1 ou 0. Afin que la sortie du réseau de neurones approche la probabilité *a posteriori* d'appartenir à la classe C_1, considérons tout d'abord la probabilité d'observer l'une ou l'autre des valeurs de la sortie t en un point de l'espace x si la sortie du modèle est y :

$$p(t|x) = y^t.(1 - y)^{1-t}$$

La probabilité d'observer l'ensemble d'apprentissage en supposant que les données sont indépendantes s'écrit :

$$\prod_{i=1}^{N} (y_i)^{t_i}(1 - y_i)^{1-t_i}$$

Pour maximiser cette fonction, on préfère minimiser l'opposé de son logarithme. La fonction de coût utilisée est donc finalement :

$$J(w) = -\sum_{i=1}^{N}\left\{t_i.\ln y_i(w) + (1 - t_i).\ln(1 - y_i(w))\right\}$$

Cette fonction, appelée *entropie croisée*, atteint son minimum lorsque $t_i = y_i$ pour tout i. Par construction, la sortie du réseau est interprétée comme la probabilité *a posteriori* d'appartenir à la classe C_1.

6.3.3 Calcul du gradient de la fonction de coût

Les méthodes de minimisation exposées au paragraphe 6.3.1 nécessitent le calcul du gradient de la fonction de coût par rapport aux poids du réseau. Les fonctions de coût présentées étant additives, le gradient total est la somme de tous les gradients partiels calculés pour chacun des exemples de la base d'apprentissage :

$$J(w) = \sum_{i=1}^{N} J^i(w)$$

Pour chaque exemple, le gradient partiel $\nabla J^i(w)$ est effectué de manière économique grâce à l'algorithme de rétropropagation [Rumelhart *et al.*, 1986]. La mise en œuvre de cet algorithme, nécessite l'expression analytique de la quantité $\frac{\partial J(w)}{\partial y_i(w)}$ où $y_i(w)$ est la sortie du réseau pour l'exemple *i.*

- Si la fonction de coût est l'erreur quadratique, alors $\frac{\partial J(w)}{\partial y_i(w)}$: c'est la différence entre la sortie du réseau et la sortie désirée, c'est-à-dire l'erreur de modélisation, on parle alors de "rétropropagation de l'erreur".
- Si la fonction de coût est l'entropie croisée, alors $\frac{\partial J(w)}{\partial y_i(w)}$ _____ .

La modification des poids peut être effectuée, soit après chaque calcul de gradient partiel, soit après le calcul du gradient total. Dans toute la suite de ce mémoire, les modifications sont effectuées après le calcul du gradient total.

6.4 Le problème de surajustement

6.4.1 Définition du surajustement

Si l'on considère un ensemble d'apprentissage et une fonction de coût quadratique, en vertu de la propriété d'approximation universelle exposée au paragraphe 6.2.3.1, il est toujours possible d'obtenir une fonction de coût aussi petite que l'on veut sur l'ensemble d'apprentissage, à condition de mettre suffisamment de neurones cachés. Cependant, le but de l'apprentissage n'est pas d'apprendre exactement la base d'apprentissage, mais le modèle sous-jacent qui a servi à engendrer les données. Or, si la fonction apprise par le réseau de neurones est ajustée

trop finement aux données, elle apprend les particularités de la base d'apprentissage au détriment du modèle sous-jacent : le réseau de neurones est *surajusté*.

6.4.2 Biais et variance des modèles

Le surajustement est souvent expliqué grâce aux concepts de biais et variance introduits dans la communauté des réseaux de neurones par [Geman *et al.*, 1992].

Si l'on considère plusieurs ensembles d'apprentissage, le biais rend compte de la différence moyenne entre les modèles et l'espérance mathématique de la grandeur à modéliser. Le biais est donc lié à la valeur du bruit du processus que l'on cherche à modéliser. La variance rend compte des différences entre les modèles selon la base d'apprentissage utilisée.

On parle souvent de compromis entre le biais et la variance. Si un modèle est trop simple par rapport au processus à modéliser, alors son biais est élevé, mais sa variance est faible puisqu'il est peu influencé par les données. Si un modèle est trop complexe, son biais est faible puisqu'il est capable de s'ajuster exactement à la base d'apprentissage, mais sa variance est élevée puisqu'une nouvelle base avec une réalisation différente du bruit peut entraîner un modèle très différent : c'est le cas du surajustement.

Ainsi, la complexité du modèle doit être ajustée pour trouver un compromis entre le biais et la variance. Dans leur article [Geman *et al.*, 1992] contrôlent la complexité du modèle et donc le surajustement en limitant le nombre de neurones cachés.

Cependant [Gallinari et Cibas, 1999] ont montré que cette vision théorique avait des limites pour un réseau à couches dont l'apprentissage était effectué avec une base d'apprentissage comprenant peu d'exemples. En étudiant différentes architectures pour un problème de régression, ils ont montré que le biais et la variance n'évoluent pas nécessairement en sens contraire lorsque le nombre de neurones cachés augmente. Dans leur cas, un modèle avec quinze neurones cachés à une variance plus élevée qu'un modèle avec soixante neurones cachés.

En résumé, le surajustement ne s'explique pas seulement par le compromis biais-variance, notamment lorsque le nombre d'exemples est faible. De plus, l'interprétation du surajustement en ces termes a été développée pour les problèmes de régression et ne se transpose pas simplement aux problèmes de classification.

6.4.3 Deux exemples artificiels de surajustement

Nous présentons ci-dessous deux problèmes artificiels pour illustrer simplement comme se manifeste le phénomène de surajustement. Le premier problème est un exemple de régression : le réseau de neurones doit approcher une fonction continue ; le deuxième problème est un exemple de classification : le réseau de neurones doit définir une frontière de séparation.

Ces deux exemples artificiels de nature différente montrent que le surajustement se traduit différemment selon le problème.

6.4.3.1 Le surajustement pour les problèmes de régression

Dans le cas d'une régression, les données de la base d'apprentissage sont bruitées. Donc, si le modèle possède trop de degrés de liberté, il peut s'ajuster localement à certains points, et apprendre la réalisation particulière du bruit sur la base d'apprentissage et non pas le processus lui-même.

Supposons que l'on cherche à modéliser un processus f comme celui qui est représenté sur la Figure 6.3. On dispose d'un ensemble d'apprentissage constitué de cinquante points choisis aléatoirement auquel est ajouté un bruit gaussien ε de variance $5{,}0.10^{-3}$.

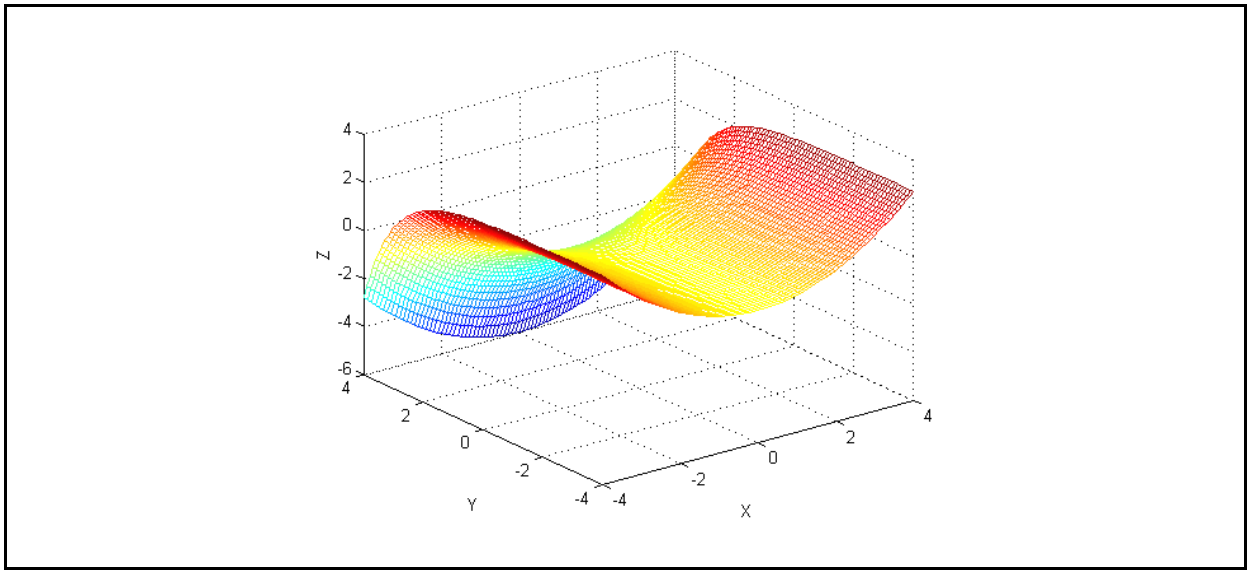


Figure 6.3 : *Représentation de la fonction* $z = f(x, y) = -0{,}5 + 0{,}2\ x^2 - 0{,}1\ \exp(y)$.

L'ensemble d'apprentissage est constitué de cinquante exemples pour lesquels la sortie t_i est calculée par :

L'apprentissage est réalisé en minimisant l'erreur quadratique moyenne sur l'ensemble d'apprentissage. Les surfaces modélisées après apprentissage par un réseau à trois neurones cachés et par un réseau à dix neurones cachés sont représentées sur la Figure 6.4. Le coût quadratique sur la base d'apprentissage est plus faible avec le réseau comprenant dix neurones cachés qu'avec le réseau en contenant deux ($6{,}0.10^{-4}$ contre $3{,}7.10^{-3}$). On voit nettement sur cette figure que le réseau avec dix neurones cachés a utilisé ses degrés de liberté pour s'ajuster localement à certains points et que le modèle trouvé est loin de la surface théorique de la Figure 6.3, contrairement à la surface modélisée par le réseau avec trois neurones cachés.

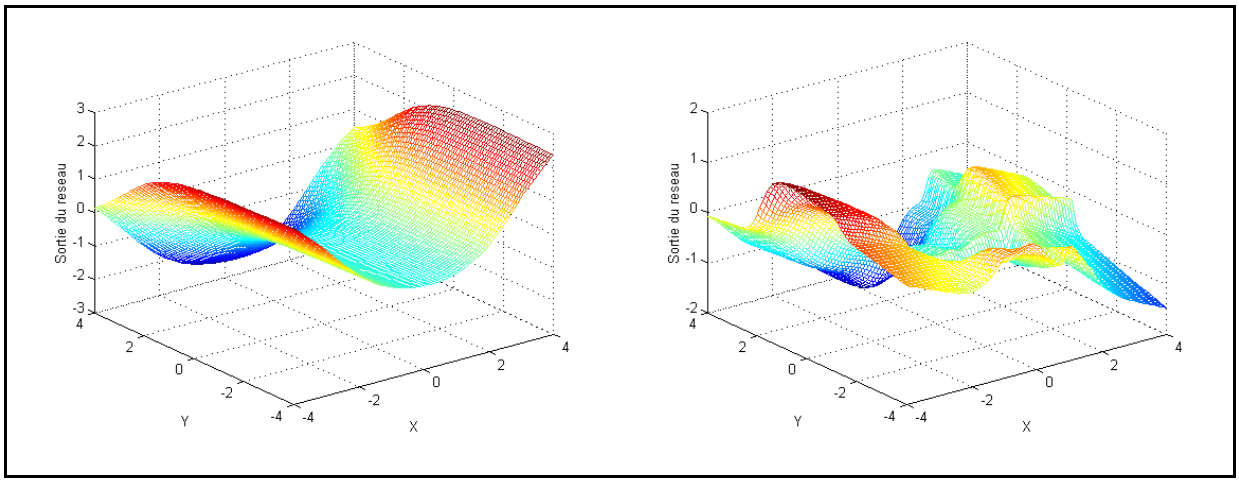


Figure 6.4 : *Surfaces modélisées par le réseau de neurones après apprentissage. Quadrant de gauche : réseau avec deux neurones cachés. Quadrant de droite : réseau avec dix neurones cachés.*

Dans cet exemple artificiel, la variance du modèle à dix neurones cachés est donc grande alors que son biais est faible ; le modèle avec trois neurones cachés semble être un bon compromis entre les deux exigences.

6.4.3.2 Le surajustement pour les problèmes de classification supervisée

Pour la classification supervisée, les données ne sont pas bruitées puisque l'on considère que le superviseur qui attribue les classes sur l'ensemble d'apprentissage ne fait pas d'erreur. Cependant, il arrive que, pour un même point de l'espace des entrées, la probabilité d'appartenance à une classe ne soit pas égale à 1 ou 0 : le problème n'est pas linéairement séparable. La sortie du réseau doit alors être la probabilité *a posteriori* au sens du théorème de

Bayes pour ce point. Si le réseau s'ajuste trop finement à la base d'apprentissage, il surestime ou sous-estime cette probabilité.

Considérons un problème de classification à deux classes avec deux entrées x et y issues de distributions gaussiennes notées $N(\mu ; \sigma)$ où μ est la moyenne de la distribution et σ son écart-type. Pour la première classe, la distribution selon x est une combinaison de deux distributions $N(-2 ; 0,5)$ et $N(0 ; 0,5)$, et celle selon y est issue de $N(0 ; 0,5)$. Pour la deuxième classe, la distribution est issue de $N(-1 ; 1)$ pour x, et $N(1 ; 0,5)$ pour y.

La Figure 6.5 montre le résultat d'un tirage aléatoire des points pour les deux classes et la probabilité *a posteriori* calculée grâce à la formule de Bayes (pour ce problème artificiel, les densités de probabilités sont connues et il est donc possible de calculer la probabilité *a posteriori* théorique). La surface de cette figure est la sortie idéale que doit avoir un réseau de neurones après l'apprentissage.

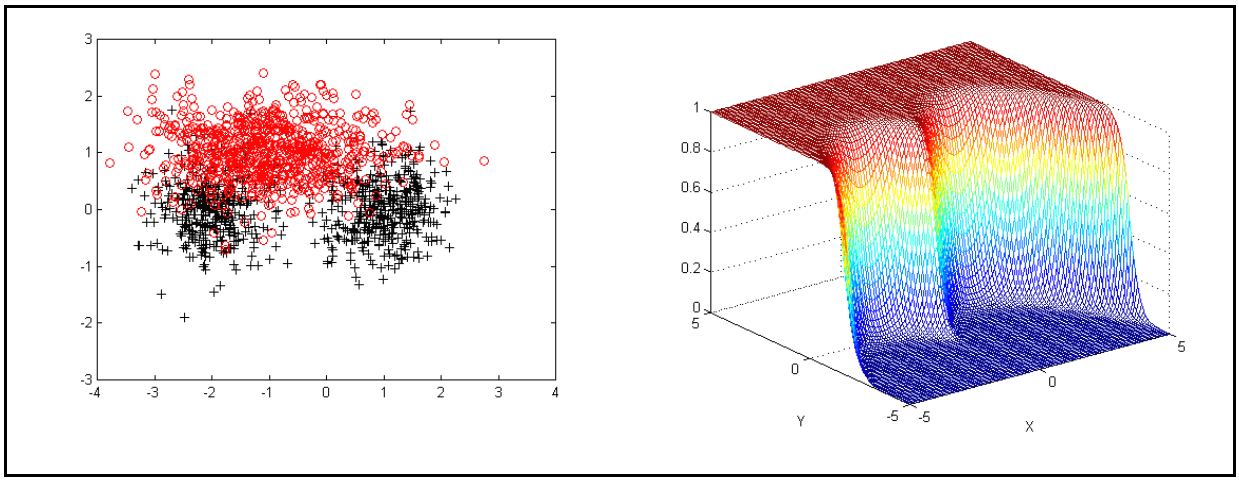


Figure 6.5 : *Répartition des points et distribution théorique de la probabilité a posteriori.*

L'un des causes du surajustement est le trop grand nombre de degrés de liberté de la fonction par rapport au modèle. Ce problème est illustré par la Figure 6.6 : la base d'apprentissage est constituée de 500 points, les réseaux de neurones sont des réseaux à couches dont on fait varier la complexité grâce au nombre de neurones cachés. Le modèle de gauche avec deux neurones cachés est bien adapté au modèle : la sortie du réseau est très proche de la sortie théorique. Le modèle de droite avec dix neurones cachés dispose clairement de trop de neurones cachés et s'ajuste pour passer exactement par certains points.

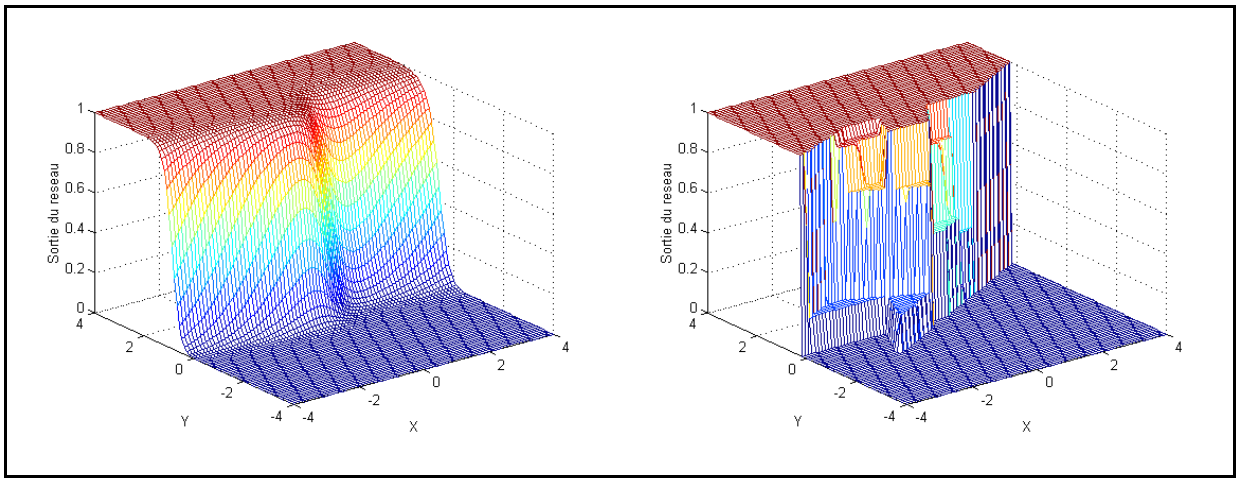


Figure 6.6 : *Sortie d'un réseau avec deux neurones cachés à gauche et dix neurones cachés à droite après apprentissage sur un ensemble de 500 points.*

Dans ce cas, si l'on sait détecter le surajustement, il suffit de réduire le nombre de neurones cachés pour trouver la bonne architecture.

Lorsque le nombre de points disponibles pour l'apprentissage diminue, le phénomène précédent s'accentue et le surajustement peut être observé même pour des architectures très simples. La Figure 6.7 montre la sortie d'un réseau à deux neurones cachés après un apprentissage avec un ensemble contenant cinquante points.

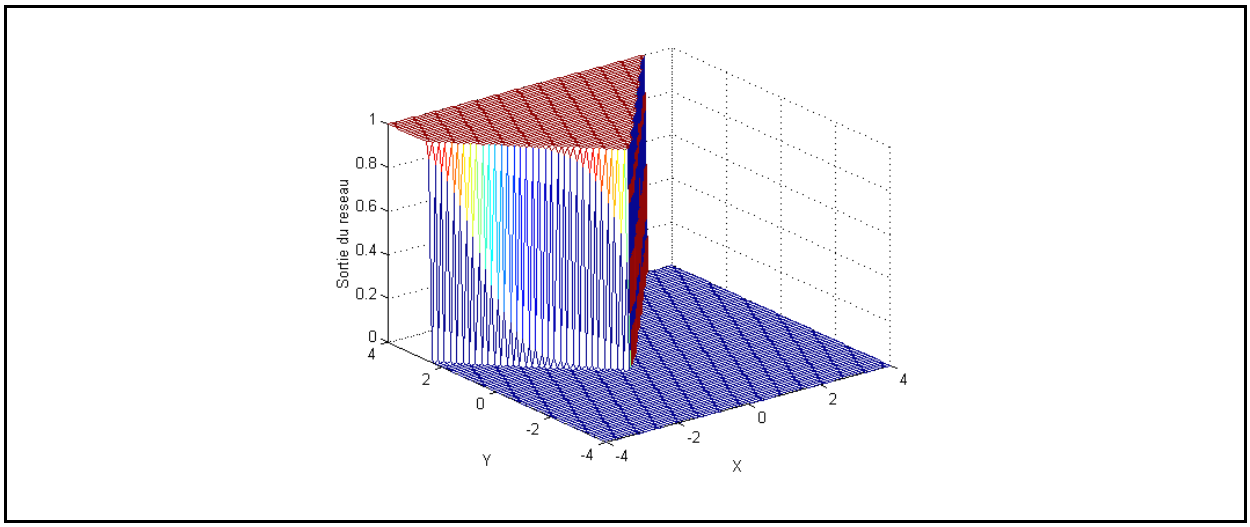


Figure 6.7 : *Sortie d'un réseau avec deux neurones cachés après apprentissage avec un ensemble de cinquante points.*

Dans ce cas, la sortie est un échelon, dont la frontière dépend grandement de la base d'apprentissage puisque cette frontière se place selon les points qui figurent dans cette zone :

le modèle trouvé a une variance élevée. De plus, pour un point situé à proximité de la frontière, le réseau produit une sortie qui vaut 0 ou 1 et qui a donc la même valeur que pour un point situé par exemple en (-2, -2) et dont la classe n'est pas du tout ambiguë. Dans ce cas, la sortie du réseau n'est plus une probabilité, mais une sortie binaire, il n'est plus possible notamment de classer les exemples par ordre de pertinence : le réseau de neurones ne fait plus de nuance. Il n'est plus possible de tracer des courbes rappel-précision, ni de changer le seuil de décision afin d'obtenir un filtre favorisant la précision ou le rappel.

6.4.3.3 Conclusions sur l'étude de ces deux exemples

Pour la classification, comme pour la régression, et pour une architecture donnée, le surajustement est d'autant plus marqué que le nombre d'exemples est faible par rapport à la dimension du vecteur d'entrée et la complexité de la fonction à approcher.

Ces exemples montrent que dans le cas d'un problème de classification, le surajustement est beaucoup plus localisé que pour un problème de régression. Dans le premier cas, le surajustement intervient dans la zone frontière entre les deux classes, alors que dans le second, il se fait sur l'ensemble du domaine.

6.5 Les méthodes pour limiter le surajustement

On distingue deux familles de méthodes pour prévenir le surajustement : les méthodes passives et les méthodes actives. Les philosophies de ces deux familles de méthodes sont différentes.

- Les méthodes passives essayent de détecter le surajustement *a posteriori* pour supprimer les mauvais modèles. Parmi les méthodes les plus classiques figurent l'utilisation d'une base de validation pendant l'apprentissage, et les mesures de critère d'information.
- Les méthodes actives interviennent pendant la phase d'apprentissage pour empêcher le modèle de faire du surajustement. Les méthodes de régularisation comme l'arrêt prématuré ou la pénalisation entrent dans ce cadre.

6.5.1 Les méthodes passives

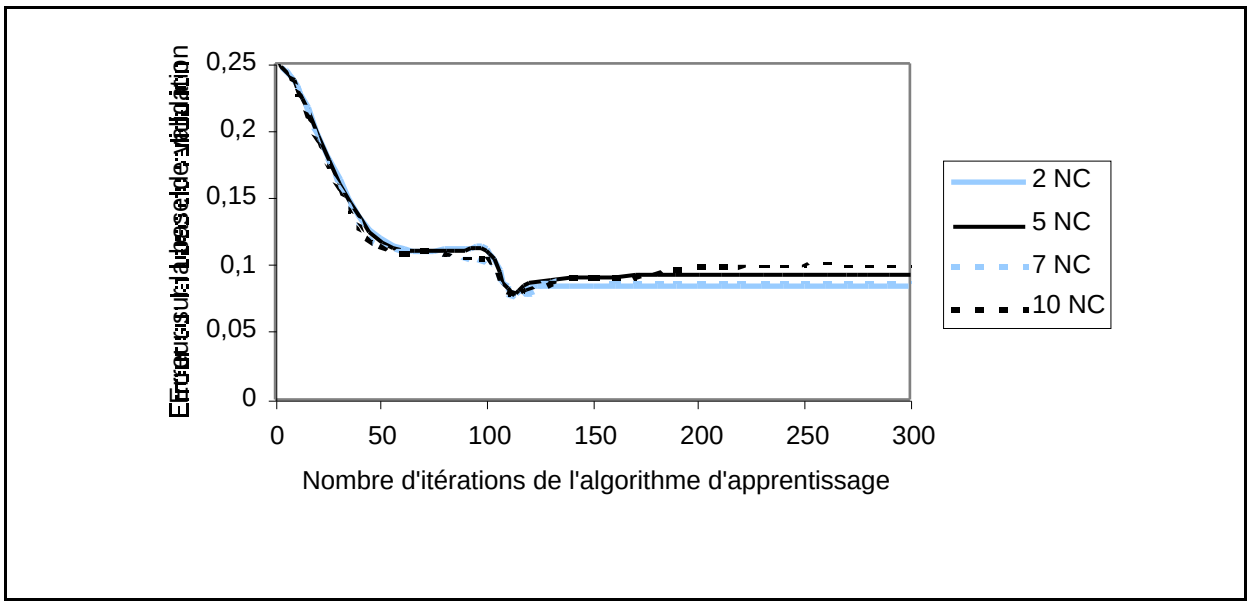
6.5.1.1 Utilisation d'une base de validation pendant l'apprentissage

Le principe consiste à mesurer les performances pendant l'apprentissage sur une base de validation qui est différente de la base d'apprentissage. Lorsque le modèle n'est pas trop ajusté

aux données de l'apprentissage, les fonctions de coût sur la base de validation et d'apprentissage diminuent ensemble. Lorsque le modèle commence à être surajusté, la fonction de coût sur la base d'apprentissage continue de diminuer, alors que la fonction de coût sur la base de validation augmente.

Cette méthode est surtout efficace pour les problèmes de régression, car comme l'a montré la Figure 6.4, le réseau tend à s'ajuster aux données sur l'ensemble de l'espace : les variations de la fonction de coût sur la base de validation sont plus facilement détectables. Dans l'exemple de la régression du paragraphe 6.4.3.1, si l'on mesure les performances sur une base de validation comprenant 500 exemples générés de la même manière que la base d'apprentissage, alors l'erreur quadratique commise par le réseau comprenant deux neurones cachés vaut $6{,}9.10^{-3}$ tandis qu'avec dix neurones cachés, cette erreur est de $2{,}4.10^{-2}$ ce qui montre que ce dernier a mal appris le processus.

Pour le problème de classification, le surajustement ne se produit pas uniformément sur l'espace, mais a tendance à apparaître dans les zones frontières entre les deux classes comme l'a montré l'exemple artificiel du paragraphe 6.4.3.2. Dans ce cas, la dégradation des performances sur une base de validation est moins évidente. Pour le problème de classification présenté au paragraphe 6.4.3.2, la Figure 6.8 montre l'évolution, pendant l'apprentissage, de l'erreur quadratique moyenne (EQMV) et du taux d'exemples mal classés sur une base de validation contenant 300 points (la base d'apprentissage est constituée de 500 exemples).



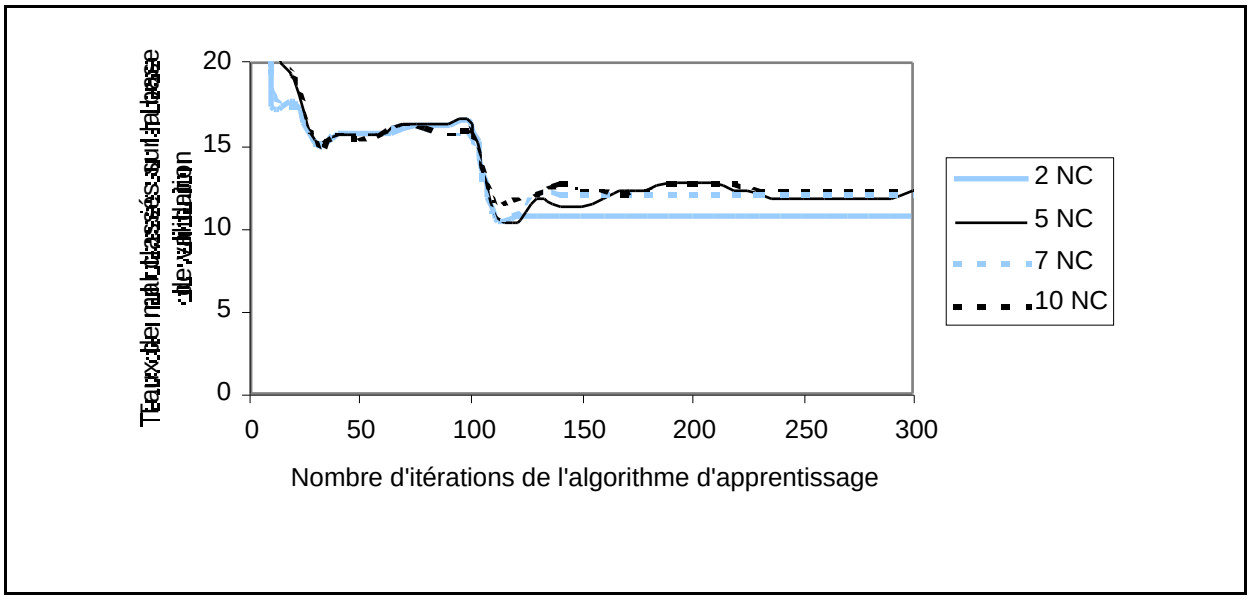


Figure 6.8 : *Évolution de la fonction de coût (EQMV) et du taux d'exemples mal classés sur la base de validation pour différentes architectures. L'axe des abscisses représente le nombre d'itérations de l'algorithme d'apprentissage.*

Ces courbes montrent que les performances sur la base validation se dégradent peu lorsque le nombre de neurones cachés augmente même pour le modèle avec dix neurones cachés, alors que la surface de la Figure 6.6 a montré clairement que le réseau de neurones était surajusté. Ceci est dû au fait que le surajustement se produit principalement dans la zone frontière et concerne peu de points de la base de validation.

En conclusion, cette méthode ou les méthodes dérivées comme le leave-one-out (cf. [Monari, 1999] pour une étude théorique et pratique du leave-one-out) ne semblent pas les plus adaptées pour éviter le surajustement dans les problèmes de classification.

6.5.1.2 Les critères d'information

Les critères d'information associent à chaque modèle une mesure qui tient compte à la fois de la qualité de l'approximation sur la base de l'apprentissage et de la complexité du modèle. On retrouve donc, en général, deux termes pour ces fonctions : le premier est d'autant plus petit que l'approximation du modèle sur la base d'apprentissage est bonne et le deuxième augmente avec la complexité du modèle. Le meilleur modèle est celui pour lequel cette mesure est la plus petite.

Parmi les mesures couramment utilisées figurent le critère d'Akaike [Akaike, 1974] ou le critère d'information développé par [Schwartz, 1978], et connu sous le nom de BIC. En pratique, ces mesures doivent être utilisées lorsque le nombre d'exemples d'apprentissage est grand devant le nombre de paramètres du modèle. Sur un exemple pour lequel le nombre d'exemples d'apprentissage est faible, ces critères conduisent à de mauvais modèles et ne sont pas utilisables [Gallinari et Cibas, 1999].

6.5.1.3 Conclusion sur les méthodes passives

Les méthodes passives, issues de la description du problème de surajustement en termes de biais-variance ne propose comme solution qu'une limitation de la complexité du modèle par l'intermédiaire d'une limitation du nombre de neurones cachés.

6.5.2 Les méthodes actives : les méthodes de régularisation

Les méthodes de régularisation, par opposition, peuvent être qualifiées d'actives, car elles ne cherchent pas à limiter la complexité du réseau, mais elles contrôlent la valeur des poids pendant l'apprentissage. Il devient possible d'utiliser des modèles avec un nombre élevé de poids et donc un modèle complexe, même si le nombre d'exemples d'apprentissage est faible.
[Bartlett, 1997] a montré que la valeur des poids était plus importante que leur nombre afin d'obtenir de modèles qui ne sont pas surajustés. Il montre, que si un grand réseau est utilisé et que l'algorithme d'apprentissage trouve une erreur quadratique moyenne faible avec des poids de valeurs absolues faibles, alors les performances en généralisation dépendent de la taille des poids plutôt que de leur nombre.
Plusieurs méthodes de régularisation existent dans la littérature, comme *l'arrêt prématuré* (*early stopping*) qui consiste à arrêter l'apprentissage avant la convergence ou les méthodes de pénalisation. Les méthodes de pénalisation ajoutent un terme supplémentaire à la fonction de coût usuelle afin de favoriser les fonctions régulières :

J est une fonction de coût comme celles présentées au paragraphe 6.3.2, et Ω est une fonction qui favorise les modèles réguliers. L'apprentissage est réalisé en minimisant la nouvelle fonction *J'*. Un modèle qui a bien appris la base d'apprentissage correspond à une valeur faible de *J*, alors qu'une fonction régulière correspond à une fonction Ω faible : l'apprentissage doit

trouver une solution qui satisfasse ces deux exigences. Parmi les différentes formes possibles pour la fonction Ω, la méthode du *weight decay* est souvent utilisée, car elle est simple à mettre en œuvre, et plusieurs études ont montré qu'elle conduisait à de bons résultats (voir par exemple [Hinton, 1987] [Krogh et Hertz, 1992] [Gallinari et Cibas, 1999]) ; de plus, elle trouve une interprétation théorique dans l'approche bayesienne développée au paragraphe 6.6.

6.5.2.1 Arrêt prématuré

Comme nous l'avons vu précédemment, l'apprentissage consiste à minimiser, grâce à un algorithme itératif, une fonction de coût calculée sur la base d'apprentissage. La méthode de l'arrêt prématuré (*early stopping*) consiste à arrêter les itérations avant la convergence de l'algorithme. Si la convergence n'est pas menée à son terme, le modèle ne s'ajuste pas trop finement aux données d'apprentissage : le surajustement est limité.

Pour mettre en œuvre cette méthode, il faut déterminer le nombre d'itérations à utiliser pendant l'apprentissage. La méthode la plus classique consiste à suivre l'évolution de la fonction de coût sur une base de validation, et à arrêter les itérations lorsque le coût calculé sur cette base commence à croître. Cependant, comme le montre la Figure 6.8, cette méthode peut être inapplicable, car il est difficile de déterminer avec précision le moment exact où il faut arrêter l'apprentissage puisque les performances sur la base de validation ne se dégradent pas nettement.

On préfère donc utiliser les méthodes de régularisation, d'autant que [Sjöberg, 1994] a montré que l'arrêt prématuré était identique à un terme de pénalisation dans la fonction de coût.

6.5.2.2 Weight Decay

Lorsque les poids du réseau sont grands en valeur absolue, les sigmoïdes des neurones cachés sont saturées, si bien que les fonctions modélisées peuvent avoir des variations brusques. Pour obtenir des fonctions régulières, il faut travailler avec la partie linéaire des sigmoïdes, ce qui implique d'avoir des poids dont la valeur absolue est faible.

Pour illustrer ce propos, on reprend le problème artificiel de classification introduit au paragraphe 6.4.3.2. La Figure 6.9 montre, pour différentes architectures, l'évolution de la somme des carrés des poids pendant l'apprentissage. Excepté pour le modèle comprenant deux neurones cachés, toutes les architectures obtiennent des poids très grands en valeur absolue.

Ces grandes valeurs des poids conduisent à des surfaces de séparation avec des variations brusques comme l'a montré la sortie du modèle comprenant dix neurones cachés (Figure 6.6).

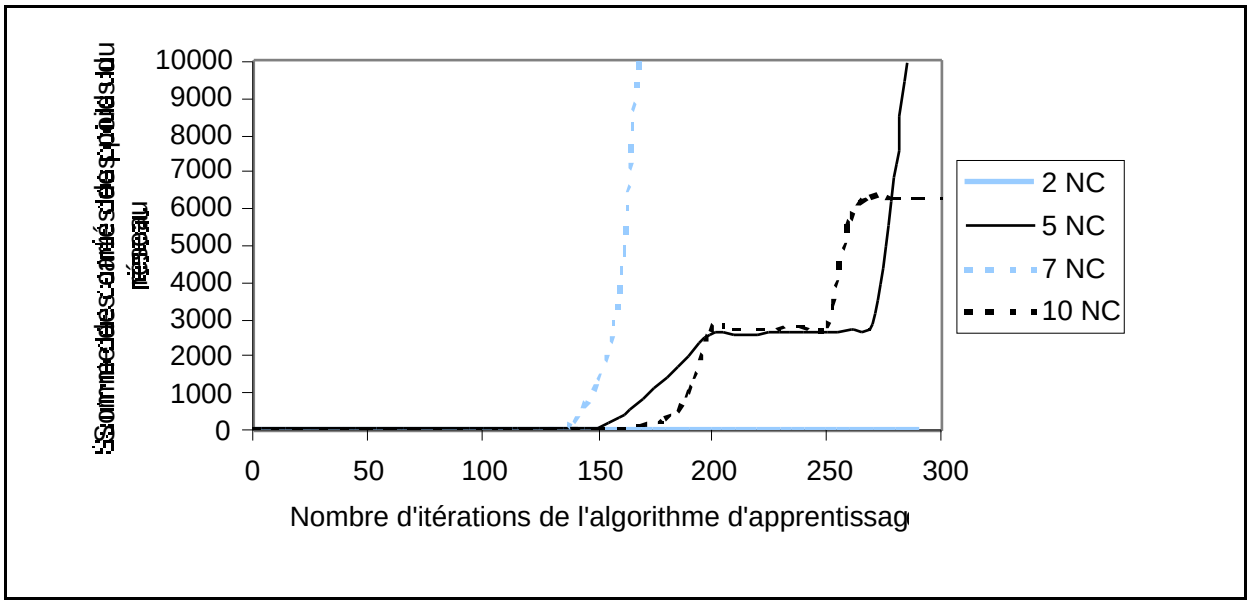


Figure 6.9 : *Évolution de la moyenne des carrés des poids — en fonction du nombre d'itérations de l'algorithme d'apprentissage. La courbe correspondant à l'architecture à deux neurones cachés est confondue avec l'axe.*

La méthode de régularisation du *weight decay* limite la valeur absolue des poids en utilisant

—.

L'apprentissage s'effectue en minimisant :

—

où *p* est le nombre de poids que comporte le réseau.

Cette méthode est appelée *ridge regression* dans le cas de modèles linéaires par rapport aux paramètres [Saporta, 1990].

α est un *hyperparamètre* qui détermine l'importance relative des deux termes dans la nouvelle fonction de coût. Si α est trop grand, les poids tendent rapidement vers zéro, le modèle ne tient plus compte des données. Si α est trop petit, le terme de régularisation perd de son importance et le réseau de neurones peut donc être surajusté. Dans le cas intermédiaire, les poids après l'apprentissage ont des valeurs modérées.

Cette méthode présente l'avantage d'être très simple à mettre en œuvre, puisque le gradient de J' se calcule très simplement à partir du gradient de J et du vecteur des poids du réseau w :

$$J' = J + \frac{\alpha_1}{2} \sum w^2 + \frac{\alpha_2}{2} \sum_{W_0} w^2 + \frac{\alpha_3}{2} \sum_{W_1} w^2$$

Il suffit d'ajouter la quantité αw au vecteur calculé par l'algorithme de rétropropagation.

En pratique, pour tenir compte du caractère différent des poids en fonction des couches, il faut considérer plusieurs hyperparamètres [MacKay, 1992b] :

W_0 représente l'ensemble des poids reliant les biais aux neurones cachés, W_1 représente l'ensemble des poids reliant les entrées aux neurones cachés et W_3 représente l'ensemble des poids reliés au neurone de sortie (y compris le biais du neurone de sortie). Le modèle comprend trois hyperparamètres α_1, α_2, α_3, qui doivent être déterminés.

L'une des solutions consiste à tester plusieurs valeurs pour ces hyperparamètres et à conserver le meilleur modèle par une méthode de validation croisée. Mais comme il y a trois hyperpraramètres à déterminer, le nombre de valeurs à tester est rédhibitoire. L'approche bayesienne expliquée au paragraphe 6.6 propose une solution théorique pour déterminer ces valeurs.

6.5.3 Exemple d'utilisation des techniques de régularisation

L'exemple présenté ci-dessous illustre les notions de surajustement et montre l'impact de l'utilisation des méthodes d'arrêt prématuré et du *weight decay.* Il s'agit d'un exemple réel de filtrage de dépêches AFP ; le filtre sélectionne les dépêches relatives au thème des *participations* que nous avons déjà présenté au chapitre 5.

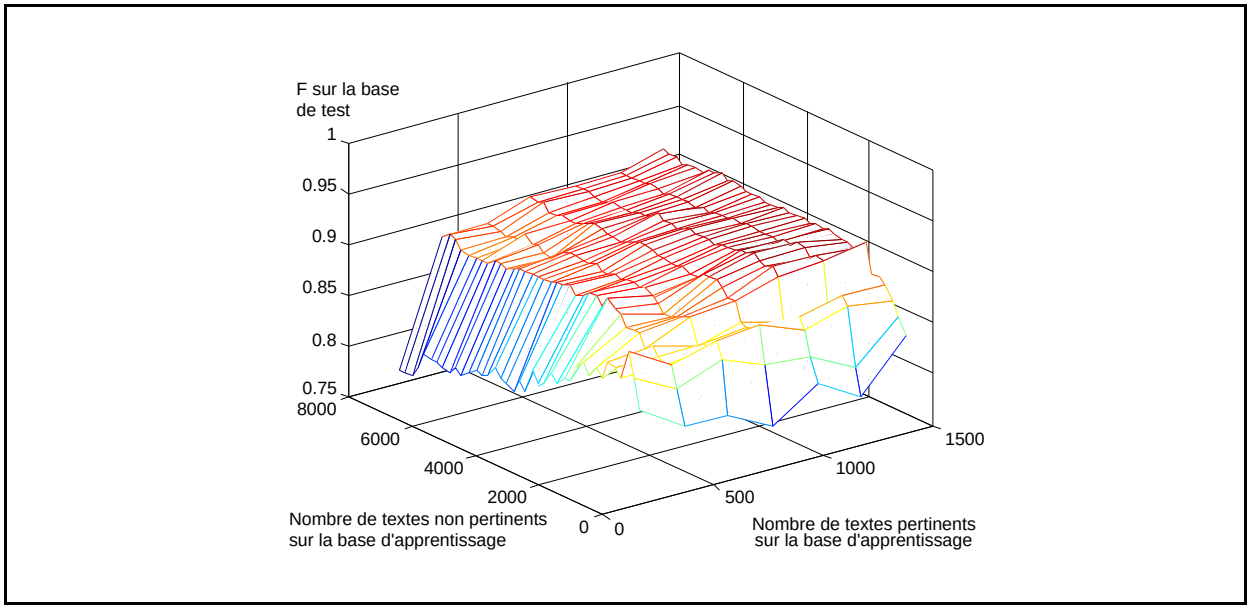
La base d'apprentissage est constituée de 1400 exemples de dépêches pertinentes au maximum et de 8000 dépêches non pertinentes. Plusieurs apprentissages sont réalisés avec un réseau contenant un unique neurone sigmoïde, et avec des tailles de la base d'apprentissage différentes ; les performances sont évaluées sur une base de test indépendante, qui comprend

200 dépêches pertinentes et 1000 dépêches non pertinentes. La mesure utilisée est la mesure *F*, le seuil de décision étant ajusté de façon à maximiser cette valeur sur la base de test.

La Figure 6.10 montre l'évolution des performances sur la base de test et l'évolution de la norme des poids, en fonction de l'évolution des proportions des exemples pertinents et non pertinents sur la base d'apprentissage. L'axe des abscisses représente le nombre de dépêches pertinentes présentes dans la base d'apprentissage et l'axe des ordonnées le nombre de dépêches non pertinentes. Pour chaque composition de la base d'apprentissage, la figure du haut rapporte les performances sur la base de test sur l'axe *z*, alors que celle du bas montre la norme euclidienne des poids du réseau après apprentissage. L'apprentissage est effectué sans aucune méthode de régularisation.

Les résultats montrent que lorsque le nombre d'exemples est faible, la norme euclidienne des poids est très grande et les performances sont faibles. Il n'est pas possible de simplifier l'architecture du réseau puisqu'il ne comporte qu'un seul neurone : l'utilisation d'une méthode de régularisation est obligatoire.

La méthode de l'arrêt prématuré a été utilisée sur le même problème de façon très simple : l'algorithme de minimisation implémente uniquement une descente de gradient simple. Plusieurs initialisations sont testées, et celle qui donne le coût le plus faible sur la base d'apprentissage est conservée. Les performances sont calculées sur la base de test comme précédemment.



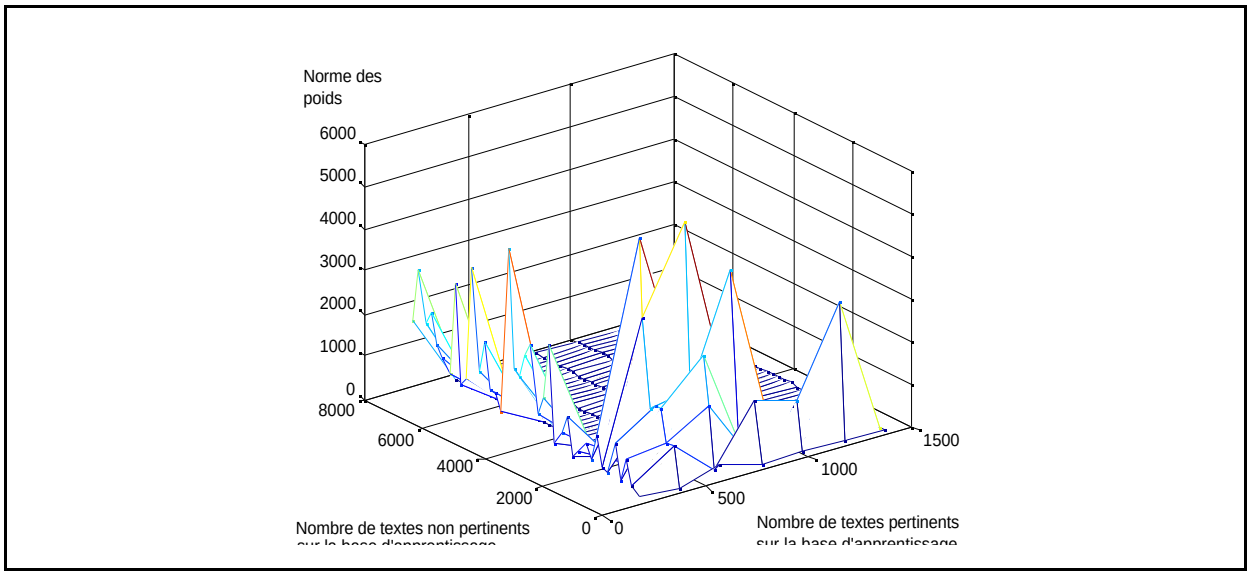


Figure 6.10 : *Évolution des performances (calcul de F sur la base de test), et de la norme euclidienne des poids, en fonction des proportions de dépêches pertinentes et non pertinentes dans l'ensemble d'apprentissage. Le nombre d'exemples pertinents varie de 200 à 1800 par pas de 200, et le nombre d'exemples non pertinents varie de 200 à 8000 par pas de 200. L'apprentissage est effectué par une descente de gradient suivi de la méthode de quasi-Newton sans aucun terme de régularisation.*

Les résultats, présentés à la Figure 6.11, montrent que, grâce à cette méthode, les performances dans la zone où le nombre d'exemples de la base d'apprentissage est faible sont nettement améliorées. En revanche, dans la zone où le nombre d'exemples est grand, les performances sont moins élevées qu'avec la méthode simple : notre implémentation de l'arrêt prématuré empêche d'exploiter toute la connaissance disponible dans la base d'apprentissage.

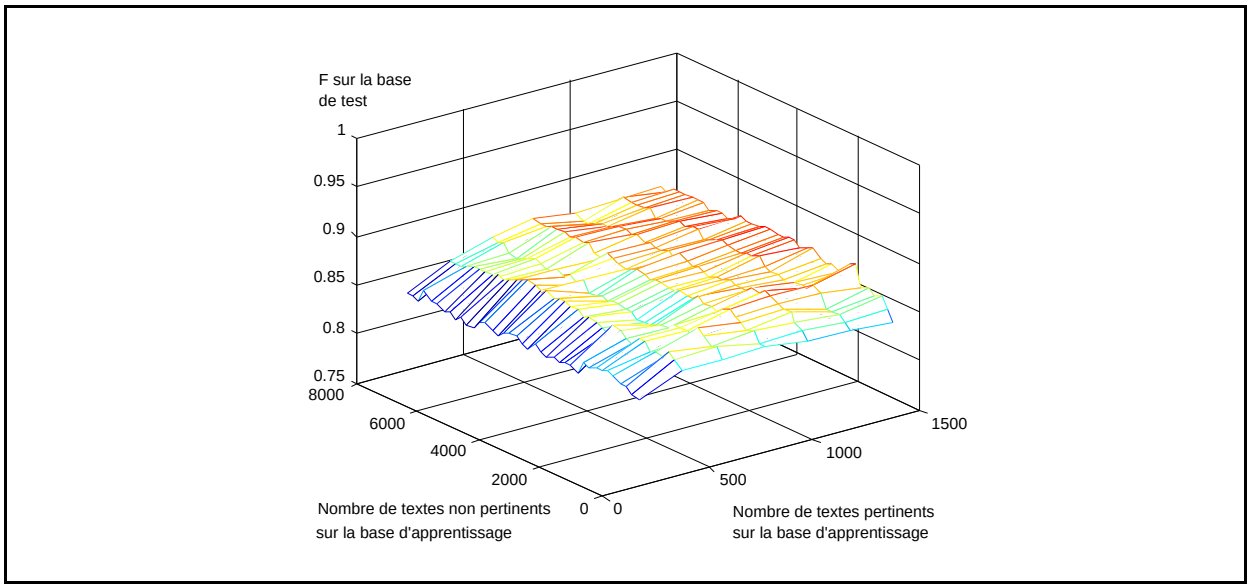


Figure 6.11 : *Arrêt prématuré : évolution des performances en fonction du nombre de dépêches pertinentes et non pertinentes sur l'ensemble d'apprentissage. Le réseau est un simple neurone sigmoïde, la minimisation de la fonction de coût s'effectue avec 800 itérations de gradient simple.*

Le graphe de la norme de poids n'est pas présenté, car dans tous les cas, les normes obtenues sont très faibles.

La méthode du *weight decay* a également été utilisée sur cet exemple, en utilisant deux hyperparamètres : un pour le biais (λ_b) et un pour les connexions entre les entrées et le neurone de sortie (λ_e). On ne s'intéresse pas, ici, à l'optimisation de ces hyperparamètres : leurs valeurs sont donc constantes durant l'apprentissage. Les résultats présentés à la Figure 6.12 montrent que, dans la zone où le nombre d'exemples est faible, les performances sont nettement améliorées par rapport à la méthode sans régularisation, et, dans la zone où le nombre d'exemples est élevé, les performances ne sont pas modifiées par rapport à l'optimum obtenu sans régularisation.

Comme précédemment, le graphe qui rend compte de l'évolution des normes n'est pas présenté, puisque les normes restent faibles quel que soit le nombre d'exemples d'apprentissage.

Cet exemple montre que le manque d'information contenu dans la base d'apprentissage peut être compensé avantageusement grâce à une méthode de régularisation comme l'arrêt prématuré ou le *weight decay*.

Ces résultats montrent la nécessité d'utiliser des méthodes de régularisation pour les problèmes de filtrage, car il est fréquent que le nombre d'exemples pertinents disponibles pour fabriquer un filtre ne dépasse pas la centaine. La méthode du *weight decay* semble préférable à la méthode de l'arrêt prématuré, car, quel que soit le nombre d'exemples disponibles, elle permet d'obtenir des résultats optimum.

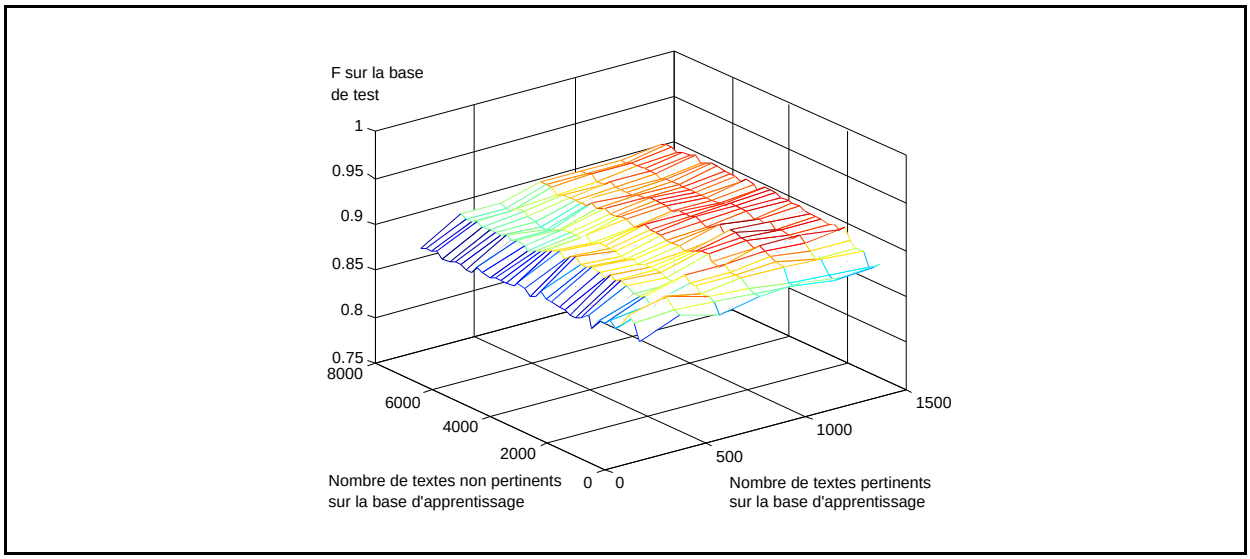


Figure 6.12 : *Weight decay : évolution des performances en fonction du nombre de dépêches pertinentes et non pertinentes sur l'ensemble d'apprentissage. Le réseau est un simple neurone sigmoïde. Les paramètres de régularisation sont* $\lambda_b = 0{,}001$ *et* $\lambda_e = 1$.

6.6 L'approche bayesienne

L'approche bayesienne a été appliquée ces dernières années aux réseaux de neurones par différents auteurs, notamment dans les travaux de [MacKay, 1992a], [MacKay, 1992b], [Neal, 1994], repris dans [Neal, 1996]) et [Buntine et Weigend, 1991] et plus récemment par [Thodberg, 1996]. Une synthèse de ces différentes approches peut être trouvée dans [Bishop, 1995].

6.6.1 Principe de l'approche bayesienne

Dans le paragraphe 6.3, l'apprentissage était effectué en trouvant une valeur du vecteur des poids qui minimise une fonction de coût issue du principe de maximum de vraisemblance. Dans l'approche bayesienne, tous les paramètres, notamment les poids du réseau, sont considérés comme des variables aléatoires issues d'une distribution de probabilité. L'apprentissage d'un réseau de neurones consiste à déterminer la distribution de probabilité des poids connaissant les données d'apprentissage : on attribue aux poids une probabilité fixée *a priori*, et, une fois que les données d'apprentissage ont été observées, cette probabilité *a priori* est transformée en probabilité *a posteriori* grâce au théorème de Bayes.

Ainsi, si D représente l'ensemble des données d'apprentissage, $p(w)$ est la densité de probabilité *a priori* des poids, $p(D|w)$ la densité de probabilité d'observer les données

connaissant les poids du réseau, et $P(w|D)$ la probabilité *a posteriori* que l'on cherche à déterminer, alors le théorème de Bayes s'écrit :

$$P(w|D) = \frac{p(D|w)\ p(w)}{p(D)}$$

Pour un problème de classification, la probabilité d'observer les données connaissant les poids a été calculée au paragraphe 6.3.2.2 :

Si l'on fait une hypothèse gaussienne pour la probabilité *a priori* des poids, elle s'écrit alors :

où $Z_w(\alpha)$ est une constante de normalisation qui ne dépend que de α :

Comme les quantités $p(D)$ et $Z_w(\alpha)$ ne dépendent pas des poids du réseau, maximiser la probabilité *a posteriori* des poids du réseau revient à minimiser la quantité :

On retrouve la fonction de coût avec un terme de *weight decay* introduite au paragraphe 6.5.2.2. Le terme de régularisation trouve, avec l'approche bayesienne, une interprétation naturelle, et la valeur de l'hyperparamètre α est liée à la variance de la probabilité *a priori* des poids.

Le formalisme décrit ici correspond à l'utilisation d'un seul hyperparamètre ; l'utilisation de plusieurs hyperparamètres correspond à des probabilités *a priori* différentes pour les différentes familles de poids. Comme ces probabilités sont indépendantes, la probabilité globale est le produit des probabilités, et, du fait des propriétés mathématiques de l'exponentielle, la nouvelle fonction de coût peut s'écrire, par exemple, pour trois hyperparamètres :

Les hyperparamètres α_1, α_2, α_3 sont liés aux variances des distributions gaussiennes.

Il est possible de choisir d'autres formes pour la probabilité *a priori* des poids. [Buntine et Weigend, 1991] choisissent, par exemple, une probabilité *a priori* fondée sur l'entropie et aboutissent à une fonction de coût de la forme :

6.6.2 Les avantages de l'approche bayesienne

D'un point de vue théorique, [Neal, 1996] a montré que, lorsque les probabilités *a priori* des poids sont convenablement choisies, il n'est pas nécessaire de limiter la taille du réseau pour éviter le surajustement, et le nombre de neurones cachés peut tendre vers l'infini. Selon cette étude, le seul facteur qui doit limiter la taille du réseau est la capacité des ordinateurs utilisés et le temps disponible pour effectuer les calculs nécessaires.

D'un point de vue pratique, la théorie de l'approche bayesienne pour l'apprentissage des réseaux de neurones apporte d'importantes améliorations :

- Le concept de régularisation peut être interprété de façon naturelle dans le contexte bayesien.
- Les hyperparamètres intervenant dans la fonction de régularisation sont calculés lors de la phase d'apprentissage sans utiliser de base de validation. Le détail des calculs est précisé au paragraphe 6.6.5.
- Le calcul de l'évidence explicité au paragraphe 6.6.6 permet de sélectionner, parmi une famille de modèles, le meilleur modèle, uniquement grâce à la base d'apprentissage.
- Comme tous les calculs se font à partir de la base d'apprentissage, il n'est plus nécessaire de disposer d'une base de validation. Il est donc possible d'utiliser toutes les données dont on dispose pour estimer les poids du réseau.
- Des barres d'erreurs peuvent être calculées pour les problèmes de régression.
- L'incertitude sur les poids peut être prise en considération pour corriger la probabilité calculée par un réseau dans un problème de classification.

· Les entrées peuvent être sélectionnées grâce à la méthode *Automatic Relevance Determination* : un hyperparamètre est associé à chaque entrée et après l'apprentissage, les hyperparamètres avec de grandes valeurs indiquent des entrées non pertinentes.

6.6.3 Les inconvénients de l'approche bayesienne

Comme les paramètres utilisés sont maintenant issus de distributions de probabilité, il est nécessaire, pour connaître un paramètre, de calculer des intégrales faisant intervenir les distributions des autres paramètres. Il est, en général, impossible de calculer ces intégrales analytiquement, et plusieurs approches ont été proposées pour effectuer ces calculs. Mais soit ces méthodes sont très lourdes à implémenter, soit elles reposent sur des approximations qui peuvent fausser les résultats.

Finalement les résultats théoriques proposés par l'approche bayesienne sont souvent inapplicables en l'état dans le cadre des réseaux de neurones.

Dans ses travaux, Neal [Neal, 1992] utilise des méthodes de Monte Carlo couplées à des modèles de Markov cachés pour calculer les différentes intégrales intervenant dans les différentes étapes. Les calculs sont très lourds à mettre en place et nécessitent beaucoup de temps de calcul. Nous n'avons pas cherché dans ce travail à utiliser cette approche.

MacKay [MacKay, 1992a] [MacKay, 1992b] [MacKay, 1992c] a proposé des approximations reposant sur des hypothèses gaussiennes des probabilités *a posteriori*. Grâce à ces hypothèses, les calculs d'intégrales se trouvent simplifiés et peuvent être effectués plus ou moins simplement. Ces approximations sont parfois discutables, surtout pour les problèmes de classification. Néanmoins, grâce à ces approximations, les calculs sont simplifiés de sorte que l'approche bayesienne devient utilisable en pratique. L'approche proposée par MacKay est connue sous le nom de *evidence framework*.

6.6.4 Principe de l'approximation gaussienne

Dans son approche, MacKay considère une approximation gaussienne de la probabilité *a posteriori* des poids. Cette approximation est obtenue en effectuant un développement au second ordre de la fonction de coût *J(w)* autour de son minimum :

$$J(w) = J\left\{w_{MP}\right\} + \frac{1}{2}\left\{w - w_{MP}\right\}^T A \left\{w - w_{MP}\right\}$$

w_{MP} est la valeur la plus probable des poids et A est le hessien de la fonction de coût $J(w)$.

Avec cette approximation de la fonction J(w), la probabilité *a posteriori* des poids s'écrit :

$$p(w|D) = \frac{1}{Z_J}\exp\left(-J(w_{MP}) - \frac{1}{2}(w - w_{MP})^T A (w - w_{MP})\right)$$

Z_j est une constante de normalisation appropriée à l'approximation gaussienne, dont la valeur est obtenue en considérant le calcul de l'intégrale d'une gaussienne :

$$Z_j(\alpha) = \exp\left(-J(w_{MP})\right)(2\pi)^{\frac{p}{2}}\left(\det(A)\right)^{-\frac{1}{2}}$$

Grâce à ces approximations, les calculs d'intégrales sont simplifiés.

6.6.5 Calcul des hyperparamètres

Le traitement correct des hyperparamètres dans l'approche bayesienne implique le calcul des intégrales sur l'ensemble de leurs valeurs possibles. Par exemple, pour connaître la probabilité *a posteriori* des poids si le modèle comporte un hyperparamètre α, il faut calculer :

$$\int$$

Deux approches différentes ont été proposées dans la littérature : la maximisation et l'intégration.

6.6.5.1 Calcul des hyperparamètres par le principe de maximisation

Le principe de la maximisation a été proposé par MacKay et repose sur les techniques développées par [Gull, 1988]. Si la densité de probabilité de l'hyperparamètre α est très étroite autour de sa valeur α_{MP} la plus probable, alors :

$$- \qquad \int$$

Cette méthode consiste à calculer la valeur la plus probable pour les hyperparamètres, et à faire la suite des calculs avec ces valeurs pour les hyperparamètres.

La valeur la plus probable de l'hyperparamètre α est déterminée grâce au théorème de Bayes :

$$\frac{\qquad}{\qquad}$$

Donc, si la probabilité de l'hyperparamètre est uniforme, le maximum de la probabilité *a posteriori* de l'hyperparamètre est obtenu lorsque $p(D|\alpha)$ est maximum :

∫

Dans le formalisme développé par MacKay, la quantité $p(D|\alpha)$ est appelée *evidence* de l'hyperparamètre.

La densité de probabilité *a priori* des poids $p(w)$ et la quantité $p(D|w)$ ont été calculées précédemment ; on a donc :

∫

soit :

En utilisant le résultat de l'approximation gaussienne de la probabilité *a posteriori,* indiqué au paragraphe 6.6.4, le logarithme de l'évidence de l'hyperparamètre s'écrit :

Pour trouver le maximum de cette expression, afin de maximiser l'évidence de l'hyperparamètre, il faut donc dériver l'expression précédente par rapport à α.

Pour calculer la dérivée du déterminant de la matrice *A*, on écrit :

I est la matrice identité et *H* est le hessien de la fonction de coût non régularisée (l'entropie croisée), c'est donc une matrice de dimension (p, p). Si l'ensemble des valeurs propres de la matrice *H* est noté $\{\lambda_i\}$, alors la matrice *A* a pour valeurs propres l'ensemble $\{\alpha+\lambda_i\}$. Par conséquent :

On obtient finalement :

$$\alpha_{MP} = \frac{\gamma}{\left\| w_{MP} \right\|^2}$$

Une interprétation de ce calcul a été donnée par [Gull, 1989] : γ est le nombre de paramètres bien déterminés, c'est-à-dire le nombre de paramètres dont la valeur est effectivement déterminée par les données d'apprentissage plutôt que par les probabilités *a priori*.

Si plusieurs hyperparamètres interviennent dans la fonction de coût, le calcul de γ précédent n'est plus valable, il est nécessaire de calculer une valeur de γ_k différente pour chaque hyperparamètre α_k :

$$\left(\text{———} \left(\quad \right) \right)$$

et

$$\alpha_k = \frac{\gamma_k}{\sum_{\omega \in W_k} w^2}$$

où W_k représente le sous-groupe de poids liés à l'hyperparamètre α_k, $\{\eta_j\}$ représente l'ensemble des valeurs propres de la matrice A, V est la matrice des vecteurs propres, et I_k est une matrice dont tous les termes sont nuls sauf les éléments diagonaux liés au groupe de poids gouvernés par l'hyperparamètre pour lesquels la valeur est 1.

Il est aisé de voir que dans le cas où il n'y a qu'un seul paramètre, on retrouve la formule précédente, puisqu'on a alors $\eta_j = \lambda_j + \alpha$ et $V^T V = I$ car, la matrice A étant symétrique, la matrice V est orthogonale.

Par conséquent, contrairement au cas précédent, il faut également calculer les matrices de vecteurs propres, ce qui demande plus de temps que pour le calcul unique des valeurs propres [Press *et al*., 1992].

La détermination des hyperparamètres par cette méthode nécessite donc plusieurs approximations, notamment l'approximation gaussienne de la probabilité *a posteriori*. De plus dans le calcul de la dérivée par rapport à α, les termes $\frac{f}{\text{——}}$ ont été négligés.

6.6.5.2 Calcul des hyperparamètres par le principe d'intégration

Une autre méthode du calcul des hyperparamètres a été appliquée par [Buntine et Weigend, 1991] et [Williams, 1995]. Elle consiste à calculer analytiquement l'intégrale qui fait intervenir les hyperparamètres. En faisant des hypothèses pour la probabilité *a priori* des hyperparamètres de la forme $p(\ln \alpha) = 1$, le calcul exact de l'intégrale devient alors possible. Les auteurs définissent des valeurs efficaces pour les hyperparamètres :

$$\alpha_{eff} = \frac{p}{\| w_{MP} \|^2}$$

La minimisation de la fonction de coût s'effectue en recalculant α à chaque itération de l'algorithme de minimisation. Cette méthode est également appelée *MAP* pour *Maximum a posteriori.*

6.6.5.3 Maximisation ou intégration

Il y a eu un grand débat dans la communauté pour savoir si l'intégration était préférable à la maximisation. *A priori* l'intégration semble être la bonne méthode puisqu'elle correspond à l'application de la théorie. Cependant, dans une publication plus récente, [MacKay, 1999] affirme la supériorité de la méthode de maximisation sur la méthode d'intégration.

Cependant, les deux expressions sont très proches et le deuxième résultat peut être considéré comme une approximation au premier ordre du premier (qualifiée de "cheap and cheerful" dans [MacKay, 1992a]) en prenant $\gamma = p$.

6.6.5.4 Implémentation et convergence

L'hyperparamètre est initialisé à une valeur aléatoire, qui ne doit pas être trop grande afin que les poids ne tendent pas vers zéro dès les premières itérations. Ensuite, après un certain nombre d'itérations pour l'algorithme de minimisation, l'hyperparamètre est estimé à nouveau régulièrement selon l'une des deux formules suivantes :

· soit α ——

où la quantité γ_n représente un nombre de termes effectifs ; elle est calculée par :

$$\gamma_n = p - \alpha_{n-1}.\mathrm{Tr}(A^{-1}) ,$$

ce qui nécessite le calcule de la matrice A^{-1} ; le calcul de la trace est alors immédiat ;

· soit ——— ,

ce qui nécessite le calcul des valeurs propres de la matrice *H*.

Dans tous les cas, il est nécessaire de calculer la matrice du hessien de la fonction de coût non régularisée (l'entropie croisée dans notre cas). Ce calcul peut être effectué par un algorithme inspiré de la rétropropagation [Bishop, 1992]. Il nécessite un nombre de calculs en $O(p^2)$ ce qui n'est pas prohibitif même pour des réseaux comprenant une centaine de poids. De plus, ces calculs n'interviennent que lors de l'apprentissage, et, comme il n'est plus utile, théoriquement, d'effectuer de validations croisées, le temps consacré au calcul des hyperparamètres est gagné par ailleurs.

Il est également possible d'utiliser des approximations de la fonction du hessien, mais dans les expériences qui suivent, le calcul exact a été utilisé. Les calculs d'inverse de matrice ou de valeurs propres ont été implémentés selon [Press *et al*., 1992].

Lorsque les hyperparamètres sont estimés, la surface de coût est modifiée et une nouvelle minimisation est effectuée, jusqu'à ce que les hyperparamètres soient à nouveau modifiés.

Les deux étapes suivantes sont répétées un certain nombre de fois jusqu'à trouver des poids et des hyperparamètres qui n'évoluent plus :

1. minimisation partielle de la fonction de coût régularisée.
2. estimation des hyperparamètres.

La convergence de cet algorithme n'a pas, à notre connaissance, été démontrée pour les modèles non linéaires comme les réseaux de neurones. De plus, différents problèmes numériques peuvent se poser avec cette méthode pendant l'apprentissage : en effet, le minimum trouvé lors de la phase d'apprentissage est un minimum pour la fonction de coût régularisée, mais pas pour la fonction de coût non régularisée. La matrice *H* n'est donc pas nécessairement définie positive : certaines valeurs propres peuvent être négatives et entraîner une valeur de γ négative. Pour remédier à ce problème, une méthode *ad hoc* peut être utilisée : ne pas tenir compte des valeurs propres négatives comme il est suggéré dans la FAQ sur l'approche bayesienne[1].

[1] http://wol.ra.phy.cam.ac.uk/mackay/Bayes_FAQ.html

6.6.6 Sélection de modèles : calcul de l'évidence d'un modèle

6.6.6.1 Principe et calcul de l'évidence d'un modèle

D'après l'ensemble de la théorie, chaque modèle H_i trouvé est lié à une probabilité, le meilleur modèle étant celui pour lequel la probabilité est la plus grande connaissant les données. Un modèle est défini par son architecture, les valeurs de ses hyperparamètres et la distribution *a posteriori* de ses poids.

Il est donc nécessaire de calculer pour chaque modèle sa probabilité *a priori* ; or, toujours selon le théorème de Bayes :

$$P(H_i|D) = \frac{p(D|H_i)\, P(H_i)}{p(D)}$$

A priori, chaque modèle étant équiprobable, la quantité $P(H_i)$ est la même pour tous les modèles. De plus, comme le dénominateur ne dépend pas du modèle, seule la quantité $p(D|H_i)$ est déterminante. Cette quantité est appelée *évidence* du modèle et doit être calculée.

$$p(D|H_i) = \int p(D|w,H_i)\, p(w|H_i) dw$$

A partir des approximations déjà effectuées pour le calcul des hyperparamètres et d'autres approximations qui ne sont pas reprises ici, MacKay propose une formule pour calculer le logarithme de l'*évidence* :

Pour un modèle donné, plus cette quantité est grande, plus le modèle a une probabilité *a posteriori* élevée ; entre plusieurs modèles, il faut donc sélectionner celui qui possède la plus grande *évidence.*

6.6.6.2 Lien entre l'évidence et les performances en généralisation

Le modèle ainsi retenu est supposé être le meilleur modèle au sens de la théorie. Or, dans la pratique, le meilleur modèle est celui qui a les meilleures performances en généralisation. Il est donc important de vérifier si la notion d'évidence est corrélée à la performance en généralisation, et plus particulièrement si les modèles avec les plus grandes évidences ont les meilleures performances.

Selon Bishop [Bishop, 1995], il existe plusieurs raisons pour lesquelles cette corrélation est souvent mauvaise :

- La base de test étant de taille finie, l'estimation des performances sur cette base n'est pas précise, et dépend évidemment du choix de cette base.
- Il peut exister plusieurs modèles différents qui font exactement les mêmes prédictions et qui ont donc la même performance en généralisation ; cependant, le calcul de l'évidence va favoriser le modèle le plus simple.
- La performance est généralement calculée avec la valeur la plus probable des poids, or l'approche bayesienne nécessiterait de tenir compte de la distribution des poids.
- Le calcul de l'évidence présenté ici résulte d'approximations qui ne sont pas nécessairement justifiées.
- Numériquement, le calcul de l'évidence peut être instable, notamment le calcul du logarithme du déterminant de la matrice *A*.

Selon MacKay, une mauvaise corrélation entre ces deux mesures révèlent un modèle mal adapté c'est-à-dire un nombre d'hyperparamètres mal choisis. Ainsi, dans [MacKay, 1992b], il utilise un modèle avec un seul hyperparamètre et trouve une mauvaise corrélation. Lorsqu'il utilise trois hyperparamètres, la corrélation devient nettement meilleure.

[Thodberg, 1996] étudie également, sur un problème de régression, la corrélation entre ces deux valeurs. Cette corrélation est loin d'être parfaite, mais cependant, l'ensemble des modèles avec l'évidence la plus élevée ont bien les meilleures performances en généralisation.

6.7 Conclusion

Ce chapitre a permis de rappeler les propriétés principales des réseaux de neurones utilisés dans la suite de ce mémoire. La définition du surajustement a été rappelée afin de fixer précisément la nature du problème, et sa spécificité dans le cas de la classification.

Nous avons montré, sur un problème artificiel et sur un problème réel de filtrage, la nécessité d'ajouter un terme de *weight decay* à la fonction de coût usuelle pour les problèmes où le nombre d'exemples de la base d'apprentissage est limité. L'étude du cas réel a montré que les mauvaises performances obtenues lorsque la base d'apprentissage possédait peu d'exemples n'étaient pas uniquement dues au manque d'information, mais également à la façon de conduire

l'apprentissage. Sur cet exemple, l'ajout du terme de *weight decay* compense en grande partie le manque d'informations.

L'introduction de l'approche bayesienne propose un cadre théorique pour le terme de *weight decay* et résout, théoriquement, le problème de la détermination des hyperparamètres.

Enfin, l'intérêt de l'ajout d'un terme de *weight decay* a été prouvé sur problème particulier de filtrage ; les chapitre 7 et 8 vont permettre de tester cette approche, ainsi que les formules de calcul des hyperparamètres, sur un ensemble de thèmes différents.

Chapitre 7 Filtrage de textes représentés par des "sacs de mots"

Ce chapitre expose les résultats obtenus sur les corpus Reuters et TREC-8 avec un réseau à une couche cachée et une représentation des textes en sac de mots. Afin de bien comprendre les résultats, l'accent est mis sur la sélection de descripteurs pour mieux mesurer l'influence des différents paramètres.

Ce chapitre se termine par la description de notre participation à la compétition TREC-8.

7.1 Présentation du modèle

Le premier modèle utilisé est un modèle simple, qui possède une architecture classique de perceptron multi-couche rappelée sur la Figure 7.1, avec des fonctions d'activation sigmoïdes pour les neurones cachés et une fonction d'activation logistique pour le neurone de sortie.

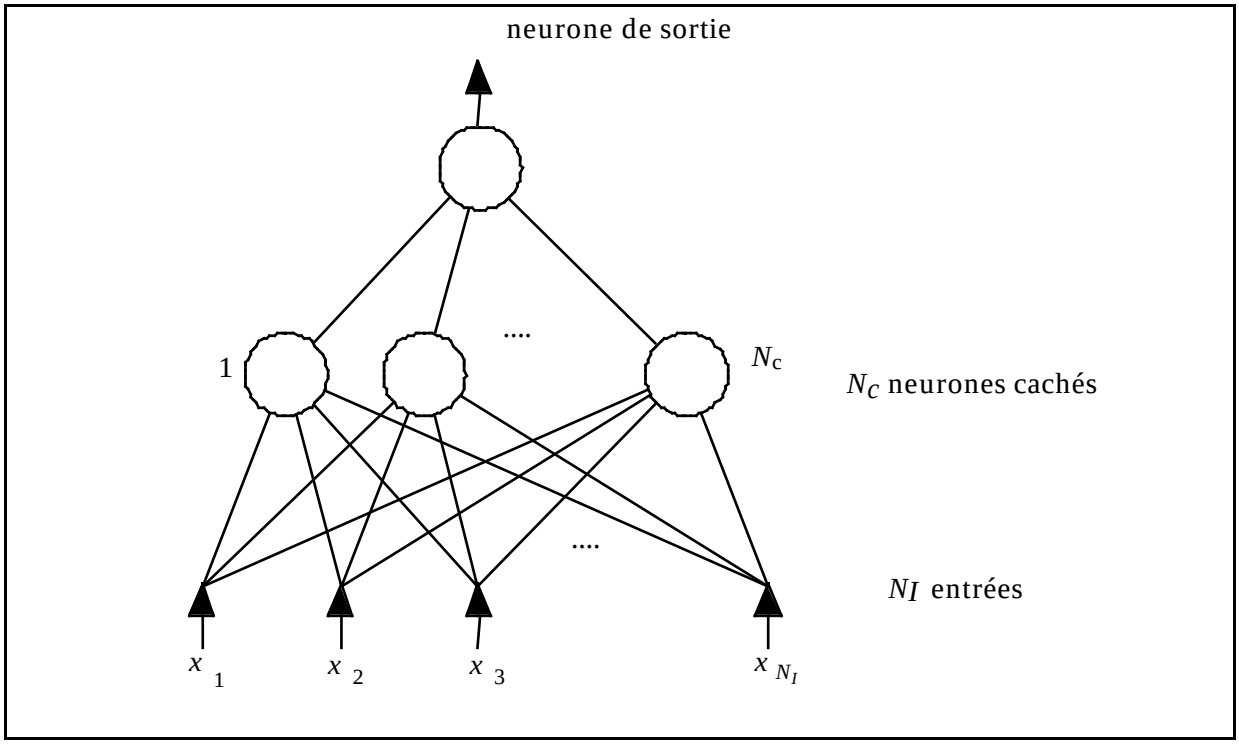


Figure 7.1 : *Architecture utilisée pour le modèle simple (les biais ne sont pas représentés pour plus de lisibilité). Les entrées x_i sont des fonctions des fréquences des mots auxquels sont associées les entrées.*

Dans ce modèle, les descripteurs utilisés pour représenter les textes sont uniquement des mots simples, sélectionnés par une méthode de sélection de descripteurs (cf. chapitre 5) ; le nombre de neurones cachés détermine la complexité de la fonction de classification.

Pour fabriquer un filtre, il est nécessaire :

- de définir les caractéristiques du vecteur des entrées : il faut déterminer sa dimension ainsi que la nature de ses composantes.
- de déterminer le nombre N_c de neurones cachés optimal afin d'obtenir les meilleures performances sur une base de test différente de la base d'apprentissage.
- de choisir une méthode de régularisation pendant la phase d'apprentissage comme indiqué dans le chapitre 6 : arrêt prématuré ou *weight decay* et, dans ce dernier cas, il faut déterminer les hyperparamètres intervenant dans la fonction de coût.

Chacun de ces choix a une grande influence sur la qualité du filtre obtenu ; nous allons décrire en détail la mise en œuvre de ces différentes étapes.

7.2 Sélection des descripteurs par la méthode de Gram-Schmidt

Le chapitre 5 a permis de montrer que, parmi les trois méthodes de sélection de descripteurs étudiées (vocabulaire spécifique, information mutuelle et orthogonalisation de Gram-Schmidt), aucune ne prévalait nettement sur les autres.

Néanmoins, la méthode d'orthogonalisation de Gram-Schmidt, couplée au critère d'arrêt décrit au chapitre 5, présente des avantages par rapport aux deux autres méthodes ; nous allons donc chercher à mieux comprendre son comportement.

Les différentes étapes de la mise en œuvre de la méthode d'orthogonalisation de Gram-Schmidt sont les suivantes :

1. À partir de l'ensemble des textes pertinents, on détermine une liste du vocabulaire spécifique du sous-ensemble des documents non pertinents.
2. À partir des textes pertinents, d'un sous-ensemble de textes non pertinents et de la liste du vocabulaire spécifique, on construit une matrice *X(N, Q)* où *N* est le nombre d'exemples total et *Q* est le nombre de descripteurs initiaux.

3. On met en œuvre l'algorithme d'orthogonalisation de Gram-Schmidt avec l'utilisation d'un vecteur aléatoire pour déterminer le critère d'arrêt.

La construction de la matrice *X* intermédiaire nécessite donc d'effectuer plusieurs choix :

1. Le codage des fréquences des descripteurs.
2. Le nombre *Q* de descripteurs initiaux.
3. Le nombre *N* d'exemples, et plus précisément le nombre d'exemples pertinents et le nombre d'exemples non pertinents.

La procédure de sélection de descripteurs détermine la qualité des descripteurs sélectionnés et leur nombre, cette qualité se mesurant grâce à la performance du filtre construit. Par conséquent, l'évaluation de la qualité de la sélection de descripteurs dépend du modèle construit à partir de cette sélection, mais elle ne peut pas être mesurée intrinsèquement. Afin de pouvoir comparer les méthodes de sélection de descripteurs, les techniques d'apprentissage (algorithmes, hyperparamètres) que nous mettrons en œuvre seront toujours identiques.

Ces choix sont tous interdépendants, et dépendent également de la nature du thème traité. Il est difficile de les étudier séparément, mais il existe des règles de conduite que nous allons mettre en évidence ci-après.

7.2.1 Codage des matrices

7.2.1.1 Centrage et normalisation des données

Le calcul intervenant dans la procédure de classement fait intervenir des produits scalaires, notamment le calcul du cosinus carré entre le vecteur de sortie et les vecteurs de descripteurs X_p :

$$\cos^2(X_p,Y) = \frac{\left[X_p^T.Y\right]^2}{\left[X_p^T.X_p\right].\left[Y^T.Y\right]}$$

Le codage de ces vecteurs influe sur ces produits scalaires, et par conséquent sur le classement trouvé.

Si l'on considère un ensemble d'exemples comportant T_1 exemples pertinents et T_0 exemples non pertinents, et que le vecteur de sortie représente par le nombre S_1 le premier ensemble et par S_0 le second, le produit scalaire entre un vecteur de descripteurs X_p et le vecteur de sortie Y s'écrit :

$$\left[\quad \right]^2 - \left(\quad \right)$$

Cette expression montre tout d'abord qu'il n'est pas possible de choisir une valeur nulle pour S_1 ou S_0 puisque cela reviendrait à ne pas tenir compte de l'une des deux classes.

Il est simple d'étudier certains cas limites. Par exemple, si un mot apparaît avec la même fréquence dans chaque texte quelle que soit sa classe, alors la valeur du cosinus doit être nulle puisque ce descripteur n'apporte aucune information. Si A est la valeur que prend ce descripteur, le produit scalaire s'écrit :

$$\left[X_p.Y\right]^2 = A^2\left(S_1.T_1 + S_0.T_0\right)^2 = 0$$

Comme cette relation doit être vraie quel que soit A, il est nécessaire d'avoir :

$$S_1.T_1 + S_0.T_0 = 0$$

Cette relation implique qu'il est nécessaire d'avoir un vecteur des sorties de moyenne nulle.

Dans la pratique, on utilise le codage suivant :

$$S_0 = \frac{-2.T_1}{T_1 + T_0} \text{ et } S_1 = \frac{2.T_0}{T_1 + T_0}$$

qui correspond à un codage initial de +1 pour les textes pertinents et –1 pour les textes non pertinents, puis à un centrage de ce vecteur.

De même, il est nécessaire de centrer la matrice X des descripteurs, car, dans le cas contraire, l'absence d'un descripteur est codée 0, ce qui est une valeur particulière dans la somme du produit scalaire.

7.2.1.2 Codage des fréquences

Le codage de la matrice X tient compte des fréquences de chaque terme dans les textes. Il provient du codage utilisé dans [Singhal, 1996] qui prend en considérations les variations de longueurs entre les textes : si $TF_j(i)$ est la fréquence du descripteur i dans le texte j, et si $\overline{TF_j}$ est la fréquence moyenne dans le texte j, on a :

$$x_j^i = \frac{1 + \log\left(TF_j(i)\right)}{1 + \log\left(\overline{TF_j}\right)}$$

7.2.2 Impact des caractéristiques de la matrice ***X***

7.2.2.1 Problématique

La méthode d'orthogonalisation de Gram-Schmidt nécessite la construction d'une matrice *X(N, Q)* constituée de *N* exemples, chacun étant décrit par *Q* descripteurs.

Pour chaque thème, l'ensemble des *N* exemples est divisé en deux sous-ensembles : N^+ est le nombre de documents pertinents et N^- le nombre de documents non pertinents.

$$N = N^+ + N^-$$

L'étude des corpus, présentée au chapitre 3, a montré que le nombre d'exemples pertinents pour un thème est en général limité : il semble raisonnable de prendre en considération l'ensemble des documents pertinents disponibles. En revanche, le nombre d'exemples non pertinents dont on dispose est, en général, très grand, puisque ce sont tous les documents du corpus non pertinents pour un thème.

Il est donc nécessaire de déterminer le nombre de textes non pertinents à prendre en considération, et de choisir ceux-ci.

- Pour le corpus Reuters, la base d'apprentissage disponible (documents datés avant le 8 avril 1987) comporte 9600 textes ; donc pour la catégorie *earn* qui comporte le plus de documents pertinents, il reste 6723 documents non pertinents possibles ; pour les catégories comportant le moins de documents pertinents, il reste 9599 documents disponibles.
- Pour le corpus de TREC-8, la base du *Financial Times 1992* qui est utilisée pour constituer les bases d'apprentissage comporte 64139 documents, il reste donc pour chaque thème environ 64000 textes utilisables pour former le sous-ensemble des documents non pertinents.
- Pour le corpus de l'AFP, le nombre de documents pertinents représente également plusieurs centaines de milliers de documents potentiels.

En conséquence, choisir le nombre *N* d'exemples revient à choisir l'ensemble des documents non pertinents.

Les *Q* descripteurs sont les *Q* premiers mots de la liste du vocabulaire spécifique du sous-ensemble des documents pertinents. Les mots sélectionnés par la méthode seront sélectionnés parmi ces *Q* descripteurs qui sont appelés dans la suite *descripteurs initiaux*.

Pour effectuer la procédure de sélection des descripteurs, il faut choisir à la fois le nombre *N* d'exemples et le nombre *Q* de descripteurs initiaux. L'expérience décrite ci-dessous étudie l'influence des choix de *N* et de *Q* sur les résultats de la sélection de descripteurs.

7.2.2.2 Description de l'expérience

Pour cette expérience, le modèle retenu est une simple régression logistique comme présenté à la Figure 7.2. La détermination du vecteur de poids *w* est effectuée en minimisant une fonction de coût *J'* incluant un terme de *weight decay* par les méthodes de minimisation décrites au chapitre 6. La fonction *J'* s'écrit :

$$J'(w) = EC(\ ) \quad -\|\ \|$$

EC(w) est l'entropie croisée et l'hyperparamètre α est fixé à 1.

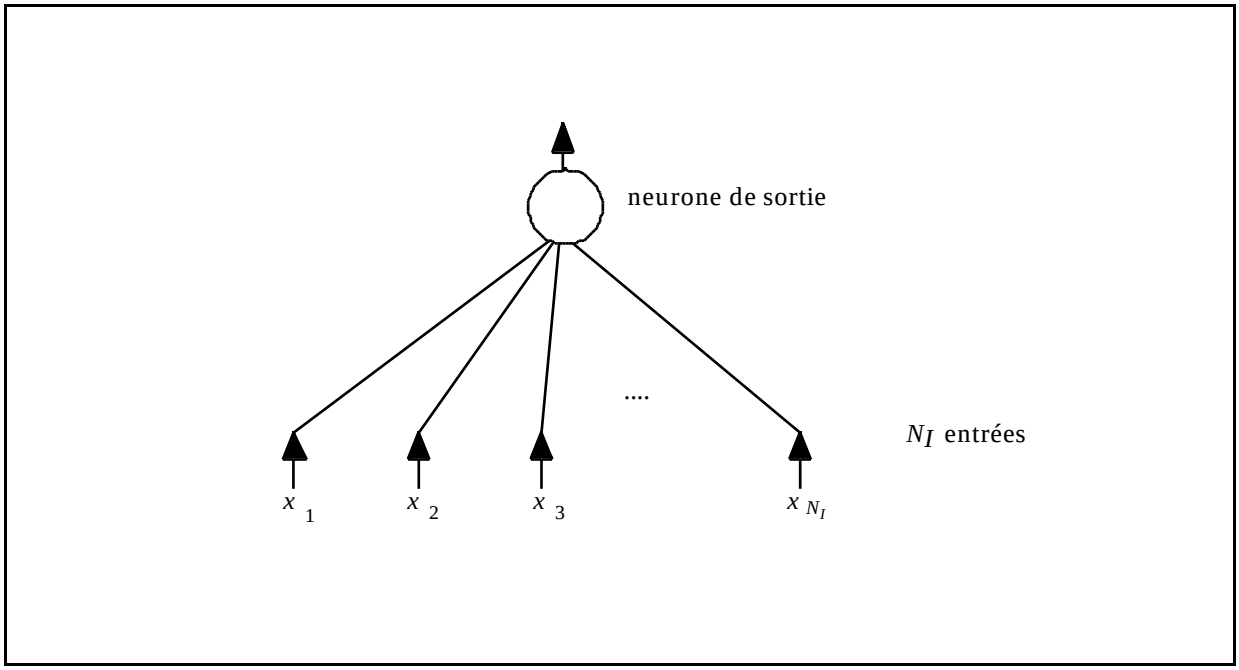


Figure 7.2 : *Régression logistique équivalente à un réseau avec 0 neurone caché (le biais n'est pas représenté).*

Les expériences sont menées sur 5 thèmes issus du corpus Reuters dont les caractéristiques sont rappelées Figure 7.3. Ces thèmes ont été choisis, d'une part, parce qu'ils ne sont pas trop faciles car ils ne dépendent pas exclusivement de la présence ou de l'absence d'un seul mot-clef. D'autre part, le nombre de documents pertinents est variable selon les thèmes, ce qui nous a permis d'étudier les corrélations éventuelles entre les valeurs des paramètres N et Q et le nombre de documents pertinents sur la base d'apprentissage.

Catégorie	Apprentissage	Test
interest	347	131
oilseed	124	47
nat-gas	75	30
sorghum	24	10
lumber	10	6

Figure 7.3 : *Liste des thèmes étudiés avec le nombre de documents pertinents disponibles sur chaque base.*

Pour chacun de ces thèmes, le nombre Q de descripteurs initiaux utilisés pour la sélection de descripteurs prend les valeurs 10, 50, 100, 200, 400.

Pour chacune des valeurs de Q, le nombre de documents non pertinents (donc le nombre N) varie ; les valeurs suivantes ont été testées : 100, 500, 1000, 2000, 3000, 4000, 5000.

7.2.2.3 Résultats des expériences

Pour chacune des expériences, les performances après l'apprentissage sont évaluées sur la base de test, par la valeur de F optimale (définie au chapitre 4) ou par les courbes rappel-précision interpolée. La valeur de F optimale est obtenue en testant plusieurs valeurs du seuil de décision pour la sortie du classifieur (de 0 à 1 par pas de 0.1) et en conservant la meilleure valeur de F sur la base de test pour s'affranchir du choix du seuil.

Les résultats obtenus pour chaque thème sont présentés sur la Figure 7.4. Pour un même thème, chaque courbe représente une valeur de Q différente, l'axe des abscisses représentant le nombre N^- de documents non pertinents utilisés pour la construction de la matrice X. Les performances sont présentées sur la colonne de gauche tandis que la colonne de droite précise le nombre de descripteurs sélectionnés par le critère d'arrêt pour chaque expérience.

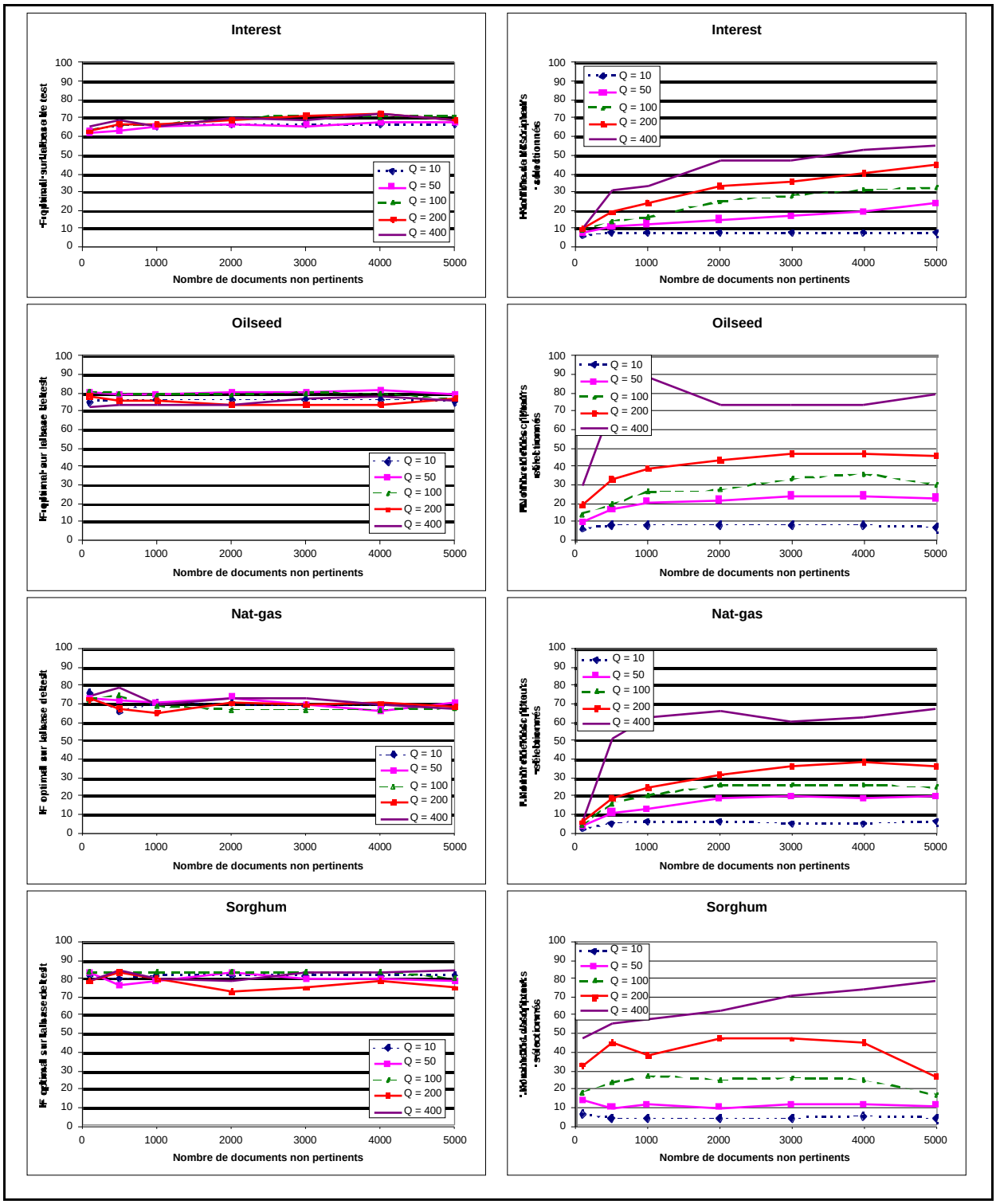


Figure 7.4 : *La colonne de gauche montre la valeur de F optimale pour chaque thème et la colonne de droite le nombre de descripteurs sélectionnés en fonction du choix de Q et du nombre de documents non pertinents.*

7.2.2.4 Interprétation des résultats

L'ensemble des résultats montre que le nombre initial de descripteurs et le nombre d'exemples non pertinents utilisés ont une influence sur les résultats de la sélection de descripteurs, et notamment sur le nombre de descripteurs sélectionnés.

Les trois premiers thèmes *interest*, *oilseed* et *nat-gas*, qui ont le plus de documents pertinents sur la base d'apprentissage, semblent avoir des comportements réguliers et de même nature tandis que les deux derniers thèmes *sorghum* et *lumber* ont un comportement différent des trois premiers.

Ces observations montrent que le choix des paramètres N et Q doit se faire en fonction du nombre de documents pertinents sur la base d'apprentissage.

Pour les trois premiers thèmes, on peut remarquer en ce qui concerne le nombre de descripteurs sélectionnés que :

1. Pour un nombre de descripteurs initial constant, le nombre de descripteurs finalement sélectionnés est d'autant plus grand que le nombre de documents non pertinents est élevé.
2. Pour un nombre de documents non pertinents fixés, le nombre de descripteurs sélectionnés est d'autant plus grand que le nombre de descripteurs initial est élevé.
3. Le facteur prédominant pour le nombre de descripteurs sélectionnés est le nombre initial de descripteurs.

Pour le dernier thème qui ne comporte que dix documents pertinents, les tendances sont différentes selon le nombre initial de descripteurs, les courbes correspondant à 200 et 400 descripteurs initiaux se détachant des autres courbes.

Les courbes de la colonne de gauche, qui indiquent les performances, montrent que les différents paramètres ont peu d'influence sur les performances du système lorsqu'elles sont mesurées par F.

Pour les trois premiers thèmes, le nombre de documents non pertinents semble avoir peu d'importance ; cependant, comme le nombre de descripteurs sélectionnés augmente avec ce paramètre, il est préférable de le limiter pour obtenir des modèles plus parcimonieux.

Pour le thème *lumber* les performances sont très variables du fait du faible nombre de documents pertinents sur la base de test : la moindre modification entraîne un grand changement dans le calcul des performances. Supposons, par exemple, que sur la base de test un filtre sélectionne six documents parmi lesquels quatre sont pertinents, la précision est alors de 66,7 % (4/6). Si l'on échange maintenant un document pertinent contre un document non pertinent, la précision devient 50 % (3/6) ce qui représente une différence de 17 points. Par conséquent les courbes présentent des fluctuations importantes, et il est difficile de tirer des conclusions claires.

La mesure du *F* caractérise un point du comportement d'un système, mais peut masquer des différences, comme nous l'avons montré dans le chapitre 4. Afin de préciser l'influence du nombre initial de descripteurs pour un nombre de documents non pertinents fixés, la Figure 7.5 présente l'évolution des courbes rappel-précision interpolée afin de préciser certains points des courbes de la Figure 7.4.
Les courbes ont été tracées avec un nombre de documents non pertinents fixé à 1000 pour le thème *interest*, à 500 pour les thèmes *oilseed*, *nat-gas*, et *sorghum* et à 100 pour le thème *lumber*. Le nombre initial de descripteurs *Q* varie d'une courbe à l'autre ; pour des raisons de lisibilité, seules les courbes correspondant à 10, 50, 200 et 400 descripteurs initiaux sont représentées.

Pour le thème *interest*, les courbes de la Figure 7.4 montrent des performances égales pour 1000 documents non pertinents en fonction du nombre initial de descripteurs. Les courbes rappel-précision précisent ce résultat et soulignent le fait que le filtre issu de la sélection des descripteurs avec 200 descripteurs initiaux est en fait supérieur aux autres, même s'ils ont tous la même valeur de *F*.
Pour le thème *oilseed* les meilleurs résultats sont obtenus avec 50 descripteurs initiaux.
Pour le thème *nat-gas*, les meilleures performances sont obtenues avec 400 descripteurs initiaux, mais les plus mauvais résultats avec 200 descripteurs initiaux. Le modèle issu de 50 descripteurs initiaux obtient des performances entre les deux.

Pour le thème *sorghum*, les meilleures performances sont obtenues avec 400 descripteurs initiaux puis avec 200. Il faut noter que, sur cette courbe, le modèle issu de 400 descripteurs initiaux apparaît comme étant supérieur notamment grâce à la précision obtenue lorsque le rappel vaut 0,8.

Les performances sur le thème *lumber* sont trop variables pour que l'on puisse en tirer des conclusions claires.

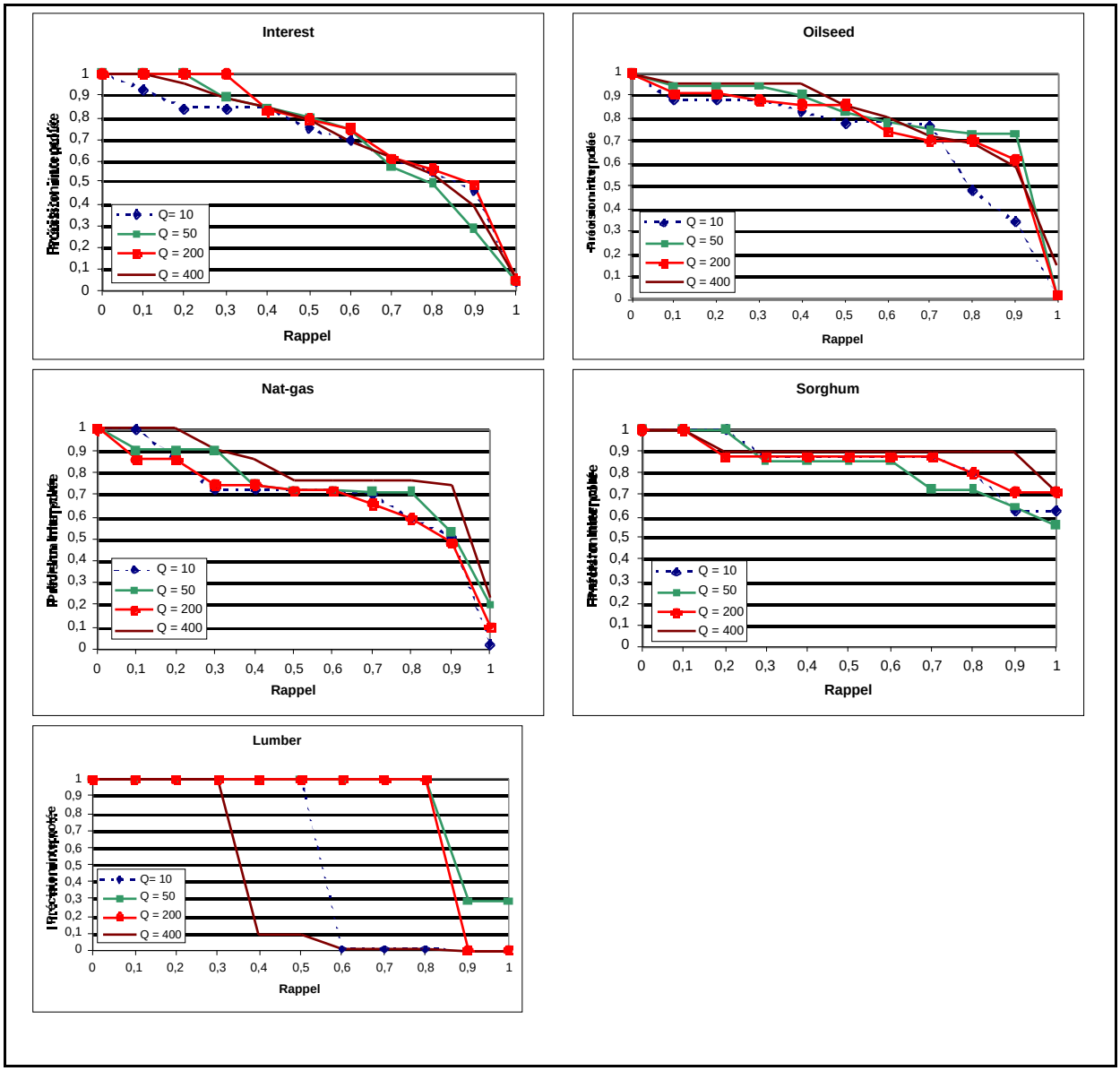


Figure 7.5 : *Courbes rappel-précision interpolée en fonction du nombre de descripteurs initiaux. Pour le thème interest le nombre de documents non pertinents est fixé à 1000, pour les thèmes oilseed, nat-gas et sorghum ce nombre est fixé à 500 et pour le thème lumber il est fixé à 100.*

L'ensemble de ces résultats montre l'existence d'une corrélation entre les deux paramètres étudiés et le nombre de descripteurs sélectionnés. En revanche, l'impact de ces paramètres sur les performances n'apparaît pas clairement et il est nécessaire d'élargir l'étude à un plus grand nombre de thèmes.

7.2.2.5 Moyenne sur les thèmes 1 à 60 du corpus Reuters

Pour mieux comprendre l'influence du nombre initial de descripteurs, nous avons réalisé une expérience sur les thèmes du corpus Reuters comprenant plus de dix documents pertinents, c'est-à-dire sur les soixante premiers thèmes.

Le modèle est toujours celui de la Figure 7.2 avec un hyperparamètre fixé à 1 pour l'apprentissage. Le nombre de documents non pertinents utilisés pour la sélection de descripteurs est de 3000 si le nombre de documents pertinents sur la base d'apprentissage est supérieur à 100, et de 500 sinon. Le nombre initial de descripteurs est de 10, 50, 100, 200 ou 400.

La Figure 7.6 présente, pour chaque valeur du nombre initial de descripteurs, le nombre moyen de descripteurs sélectionnés ainsi que les performances sur l'ensemble des thèmes. Les performances sont évaluées avec deux mesures : la moyenne de la valeur optimale de *F* obtenue sur chaque thème (macro-moyenne) et la moyenne des précisions moyennes non interpolées (UAP).

Nombre initial de descripteurs	Moyenne du nombre de descripteurs sélectionnés	UAP	*F* optimal
10	6,1	76,6	75,5
50	17,8	79,2	77,5
100	26,7	79,3	77,0
200	46,3	79,0	75,9
400	71,5	79,6	77,3

Figure 7.6 : *Macro moyenne du nombre de descripteurs sélectionnés et des performances sur les soixante premiers thèmes du corpus Reuters en fonction du nombre initial de descripteurs utilisés pour effectuer la sélection de descripteurs.*

La Figure 7.7 montre les courbes rappel-précision interpolée pour les soixante thèmes.

Ces résultats confirment que le nombre de descripteurs sélectionnés est une fonction croissante du nombre initial de descripteurs.

Avec les paramètres choisis, les moyennes varient peu en fonction du nombre Q de descripteurs initiaux ; seul le choix $Q = 10$ conduit à des résultats inférieurs sur l'ensemble des thèmes.

La solution $Q = 50$ est le meilleur choix pour l'ensemble des thèmes puisqu'elle représente le meilleur compromis entre parcimonie et performance.

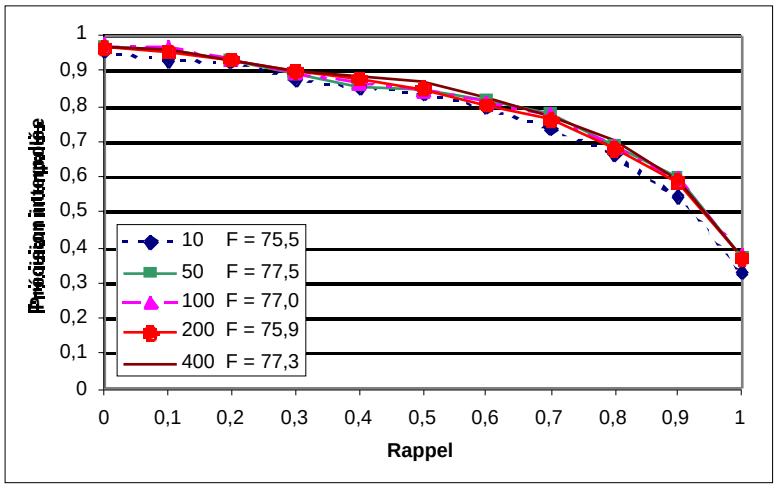


Figure 7.7 : *Courbes rappel-précision interpolée pour l'ensemble des soixante premiers thèmes du corpus Reuters en fonction du nombre initial de descripteurs utilisés pour la sélection des descripteurs. La valeur de F correspondante à chaque courbe est indiquée.*

Ces résultats sont des moyennes obtenues sur les thèmes 1 à 60, mais les performances calculées sur les sept premiers thèmes, caractérisés par un nombre de documents pertinents sur la base d'apprentissage supérieur à 300 (Figure 7.8), conduisent à des observations différentes. Dans ce cas, le meilleur compromis entre performance et parcimonie est obtenu avec 100 descripteurs initiaux. Les modèles issus de dix descripteurs initiaux sont nettement plus mauvais et contrairement à la Figure 7.6, les modèles issus de 400 descripteurs initiaux obtiennent de bonnes performances.

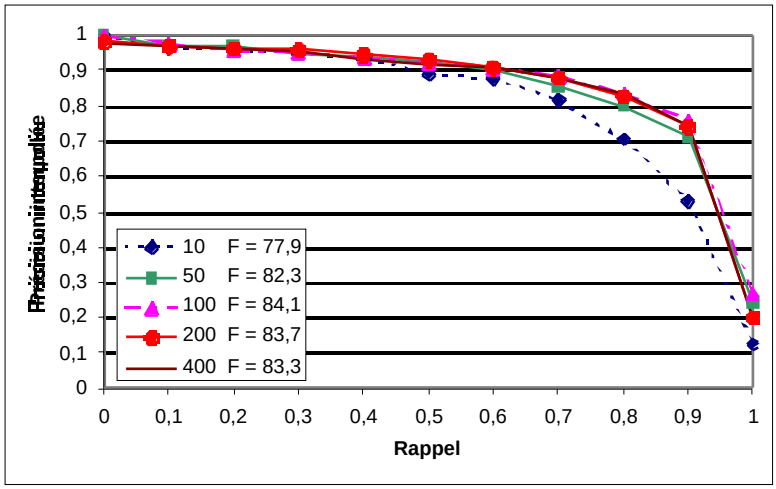


Figure 7.8 : *Courbes rappel-précision interpolée pour les sept premiers thèmes du corpus Reuters (plus de 300 documents pertinents sur la base d'apprentissage) en fonction du nombre initial de descripteurs utilisés pour la sélection des descripteurs.*

De même, les courbes de la Figure 7.9 qui présentent les résultats pour les quinze derniers thèmes (moins de vingt documents pertinents), montrent que pour obtenir les meilleures performances sur ces thèmes, il est nécessaire de limiter le nombre de descripteurs initiaux à cinquante ou dix.

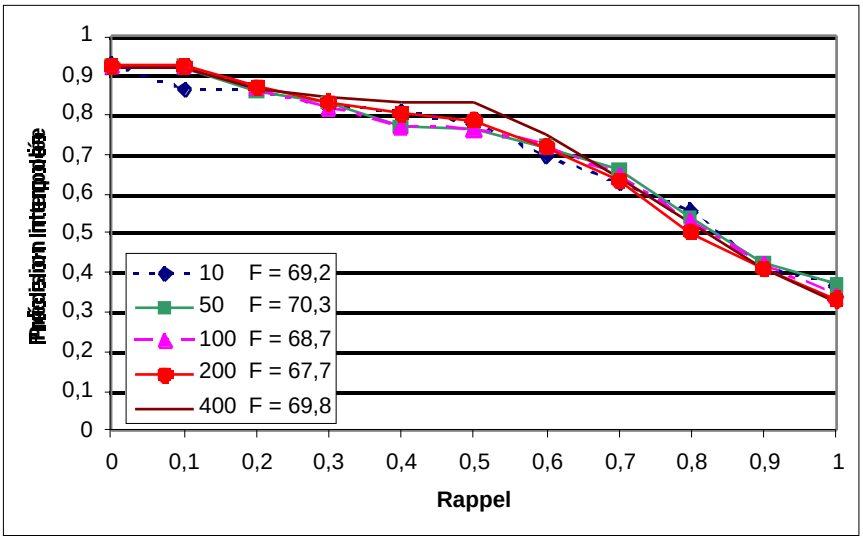


Figure 7.9 : *Courbe rappel précision interpolée pour les quinze derniers thèmes du corpus Reuters (moins de vingt documents pertinents sur la base d'apprentissage) en fonction du nombre initial de descripteurs utilisés pour effectuer la sélection de descripteurs.*

L'ensemble de ces résultats prouve que le nombre *Q* de descripteurs initiaux doit être adapté au nombre de documents pertinents. Il apparaît qu'un bon compromis consiste à choisir Q = 100

lorsque le nombre de documents pertinents est supérieur à 100, et Q = 50 lorsque le nombre de documents pertinents est inférieur à 100.

La Figure 7.10 montre les performances moyennes obtenues avec cette méthode (*Q* optimisé) par rapport à *Q* constant fixé à cinquante pour l'ensemble des thèmes.

	Moyenne du nombre de descripteurs sélectionnés	UAP	*F* optimal
Q = 50	17,8	79,2	77,5
Q optimisé	20,2	79,4	77,7

Figure 7.10 : *Comparaison entre Q = 50 et Q optimisé sur l'ensemble des soixante thèmes.*

Les différences ne sont pas très grandes entre les deux méthodes, car "l'optimisation" concerne peu de thèmes et les moyennes sur l'ensemble des thèmes masquent les différences.

La Figure 7.11 reprend la même comparaison, mais uniquement sur les dix premiers thèmes, et montre, dans ce cas, la supériorité des résultats.

	Nombre de descripteurs sélectionnés	UAP	*F* optimal
Q = 50	22,6	89,0	84,1
Q optimisé	32,2	89,7	85,3

Figure 7.11 : *Comparaison entre Q = 50 et Q optimisé sur les dix premiers thèmes.*

7.2.2.6 Influence du choix des documents non pertinents

Comme les documents non pertinents sont sélectionnés aléatoirement, il est nécessaire d'étudier la variabilité des résultats en fonction du choix de ces documents.

En conséquence, pour chacun des 5 thèmes considérés au paragraphe 7.2.2.2, 500 expériences différentes ont été effectuées, pour lesquelles le nombre de descripteur initial est systématiquement fixé à 100 et, pour chaque expérience, les documents non pertinents sont sélectionnés aléatoirement. Pour une première série d'expériences, le nombre de documents non pertinents est fixé à 1000 (Figure 7.12) et dans une deuxième série, ce nombre est fixé à 500 (Figure 7.13). Pour chacune des expériences, les valeurs moyennes du nombre de descripteurs sélectionnés, des précisions moyennes non interpolées et des mesures *F* optimales sont calculées ainsi que les écarts types.

Les résultats montrent que la dispersion des résultats augmente au fil des thèmes. L'augmentation de la dispersion s'explique car, sur le corpus Reuters, les thèmes qui ont peu de documents pertinents sur la base d'apprentissage en ont également peu sur la base de test : :pour ces thèmes, de petites modifications entraînent des variations importantes des performances.

Cependant, excepté pour le thème *lumber,* la dispersion des résultats reste faible, si bien que les résultats ne dépendent pas beaucoup du choix des documents non pertinents.

Les résultats confirment qu'il faut adapter le nombre de documents non pertinents utilisés pour la sélection de descripteurs au nombre de documents pertinents puisque, pour le thème *interest,* les meilleurs résultats sont obtenus avec 1000 documents non pertinents ; pour les

thèmes *oilseed* et *nat-gas*, les résultats sont comparables entre les deux expériences, mais les modèles sont plus parcimonieux avec 500 documents non pertinents. Pour le thème *sorghum*, les conclusions sont moins claires et les performances sont très proches avec des nombres de descripteurs sélectionnés très proches également. Enfin, pour le thème *lumber*, les écarts types sont trop élevés pour tirer des conclusions.

		Nombre de descripteurs sélectionnés	UAP	*F* optimal
Interest	Moyenne	19,4	73,0	67,5
	Ecart-type	2,0	1,2	1,3
Oilseed	Moyenne	24,5	77,6	78,4
	Ecart-type	2,6	1,3	1,5
Nat-gas	Moyenne	19,0	68,5	67,2
	Ecart-type	2,8	2,6	2,4
Sorghum	Moyenne	23,6	80,2	79,7
	Ecart-type	4,7	2,6	3,6
Lumber	Moyenne	19,0	58,0	53,2
	Ecart-type	5,8	9,8	6,5

Figure 7.12 : *Nombre de descripteurs moyens et performances moyennes pour 500 tirages différents de 1000 documents non pertinents.*

		Nombre de descripteurs sélectionnés	UAP	*F* optimal
Interest	Moyenne	15,2	72,6	66,3
	Ecart-type	1,6	1,4	1,4
Oilseed	Moyenne	20,2	77,3	78,8
	Ecart-type	2,3	1,3	1,3
Nat-gas	Moyenne	16,0	68,7	68,9
	Ecart-type	2,9	3,2	2,9
Sorghum	Moyenne	24,1	79,5	79,5
	Ecart-type	5,2	2,5	3,5
Lumber	Moyenne	22,6	55,1	52,7
	Ecart-type	7,0	13,1	6,8

Figure 7.13 : *Nombre de descripteurs moyens et performances moyennes pour 500 tirages différents de 500 documents non pertinents.*

7.2.2.7 Cas des catégories avec très peu de documents

Dans le cas du corpus Reuters, les trente derniers thèmes possèdent moins de dix documents pertinents, et les vingt-cinq derniers ont un nombre de documents pertinents inférieur ou égal à cinq. Dans ce cas, il est probable que la plupart des mots issus du vocabulaire spécifique apparaissent systématiquement ensemble dans les documents pertinents et sont donc très corrélés ; comme la méthode d'orthogonalisation de Gram-Schmidt exploite ces corrélations, elle peut être conduite à sélectionner très peu de descripteurs. De plus, les résultats du paragraphe précédent ont montré que, pour les thèmes possédant très peu de documents pertinents, les résultats dépendaient beaucoup des documents non pertinents utilisés.

Par conséquent, nous avons comparé les résultats obtenus par la méthode d'orthogonalisation de Gram-Schmidt et ceux obtenus en considérant uniquement les dix premiers mots trouvés par la méthode du vocabulaire spécifique sur les trente derniers thèmes du corpus Reuters. Pour la méthode de Gram-Schmidt, Q est fixé à dix, et 500 documents non pertinents sont utilisés.

Les résultats sont présentés à la Figure 7.14 pour les trente derniers thèmes (ensemble des thèmes avec moins de dix documents pertinents) et les vingt derniers thèmes (ensemble des thèmes avec moins de cinq documents pertinents) et sont calculés grâce à la moyenne des mesures de F pour chaque thème et par les moyennes des précisions moyennes non interpolées (UAP).

	30 derniers thèmes			20 derniers thèmes		
	Nombre de descripteurs sélectionnés	UAP	*F*	Nombre de descripteurs sélectionnés	UAP	*F*
Vocabulaire spécifique	10	59,0	43,8	10	59,0	36,7
Gram-Schmidt	14,9	50,2	44,0	12,1	38,9	33,3

Figure 7.14 : *Comparaison des méthodes de sélection de descripteurs sur les thèmes possédant peu de documents pertinents.*

La méthode du vocabulaire spécifique s'avère meilleure pour l'ensemble des thèmes possédant peu de documents pertinents, d'autant plus qu'elle implique des modèles plus parcimonieux et qu'elle nécessite moins de calculs.

Cependant, contrairement aux autres résultats obtenus, les conclusions sont légèrement différentes selon la mesure : avec la précision moyenne non interpolée, les différences entre les deux approches sont très élevées, alors que la mesure de *F* met en avant des écarts plus faibles. Cette différence s'explique par certains thèmes comme le thème *dfl* (le thème 82 de notre liste) dont les résultats sont présentés à la Figure 7.15, et qui ne possède qu'un seul document pertinent sur la base de test.

	UAP	*F*
Vocabulaire spécifique	100	0,06
Gram-Schmidt	0,06	0,06

Figure 7.15 : *Résultats pour le thème dfl.*

Pour ce thème, les mesures de *F* ont la même valeur (la valeur minimale) pour les deux approches, mais la précision moyenne non interpolée est très différente d'une méthode à l'autre, puisqu'elle est maximale avec la méthode du vocabulaire spécifique et minimale avec la méthode de Gram-Schmidt. Pour comprendre cette différence, la Figure 7.16 montre les cinq probabilités les plus élevées de la base de test lorsque le vecteur d'entrée du modèle est sélectionné par la méthode du vocabulaire spécifique : si le document pertinent est bien classé en première position, sa probabilité est trop faible pour qu'il soit sélectionné par une méthode de seuil.

Dans ce cas, la mesure de la précision moyenne non interpolée n'est plus une bonne mesure, car, malgré la valeur maximale obtenue, le système ne peut pas exploiter le bon classement.

Sortie du réseau	Pertinent
0,080	oui
0,060	non
0,059	non
0,057	non
0,052	non

Figure 7.16 : *Cinq premières probabilités pour le thème dfl.*

7.2.3 Conclusion sur la mise en œuvre de Gram-Schmidt

L'ensemble des expériences effectuées sur la sélection de descripteurs a permis de tirer plusieurs enseignements, et notamment que le nombre de documents non pertinents ainsi que le nombre initial de descripteurs doivent être choisis en fonction du nombre de documents pertinents sur la base d'apprentissage.

Plus précisément :

- Le nombre de descripteurs sélectionnés est une fonction croissante du nombre de documents non pertinents et du nombre initial de descripteurs. Le facteur prédominant est le nombre initial de descripteurs.
- Le nombre de documents non pertinents et le nombre initial de descripteurs doivent être limités en fonction du nombre de documents pertinents sur la base d'apprentissage, afin d'obtenir le meilleur compromis entre performance et parcimonie.

Après la sélection de descripteurs, il reste à déterminer les caractéristiques de la base d'apprentissage d'une part, et le nombre de neurones cachés de l'architecture neuronale d'autre part.

7.3 Choix des documents non pertinents pour la base d'apprentissage

Comme pour l'étape de sélection de descripteurs, il est nécessaire de choisir des documents pertinents et des documents non pertinents pour constituer une base d'apprentissage.
Comme précédemment l'ensemble des documents pertinents disponibles est retenu et il reste à choisir un sous-ensemble de documents non pertinents, ce sous-ensemble n'étant pas nécessairement le même que celui utilisé pour la sélection des descripteurs.
L'expérience suivante étudie l'influence de la constitution de la base d'apprentissage sur les performances et plus précisément l'impact du nombre de documents non pertinents. Les cinq thèmes du paragraphe précédent sont étudiés ; la méthode de sélection de descripteurs tient compte des résultats obtenus précédemment : elle est réalisée avec les paramètres de la Figure 7.17.

	Nombre de documents non pertinents pour la sélection de descripteurs.	Nombre de descripteurs initiaux	Nombre de descripteurs sélectionnés
interest	3000	200	36
oilseed	500	50	18
nat-gas	500	50	12
sorghum	500	50	19
lumber	100	50	17

Figure 7.17 : *Paramètres de la sélection de descripteurs.*

Comme les descripteurs sélectionnés pour représenter les textes sont en nombre réduit, la plupart des textes ne comportent aucun de ces mots. En d'autres termes, avec la représentation choisie, la majorité des textes sont tout simplement des vecteurs nuls. Par construction, les documents pertinents ne doivent pas être des vecteurs nuls puisque les descripteurs ont été choisis dans l'ensemble représentatif de ce sous-ensemble.

La Figure 7.18 donne, pour les 5 thèmes étudiés, la proportion de vecteurs non nuls parmi 5000 documents non pertinents sélectionnés aléatoirement, représentés par les descripteurs sélectionnés avec les paramètres de la Figure 7.17.

	Nombre de documents non pertinents différents du vecteur nul.	Proportion de vecteur non nuls
interest	1924	38,4%
oilseed	664	13,2%
nat-gas	581	11,6%
sorghum	783	15,6%
lumber	249	5,0%

Figure 7.18 : *Nombre et proportion de vecteurs non nuls sur 5000 documents non pertinents sélectionnés aléatoirement avec les descripteurs de la Figure 7.17.*

Les résultats montrent que, quel que soit le thème, une majorité de documents non pertinents sont en fait des vecteurs nuls, ce qui signifie, par exemple, que pour le thème *lumber*, 95 % des documents non pertinents apportent exactement la même information : un document qui ne contient aucun des descripteurs n'est pas pertinent.

Une expérience a été effectuée pour chaque thème avec le modèle de la Figure 7.2 et un hyperparamètre fixé à 1, en faisant varier le nombre de documents non pertinents dans la base d'apprentissage : 1000, 2000, ou un ensemble de documents non pertinents tel qu'aucun ne soit représenté par un vecteur nul.

Les courbes rappel-précision interpolée sont présentées à la Figure 7.19, pour un thème donné ; chaque courbe correspond à un choix différent de l'ensemble des documents non pertinents constituant la base d'apprentissage.

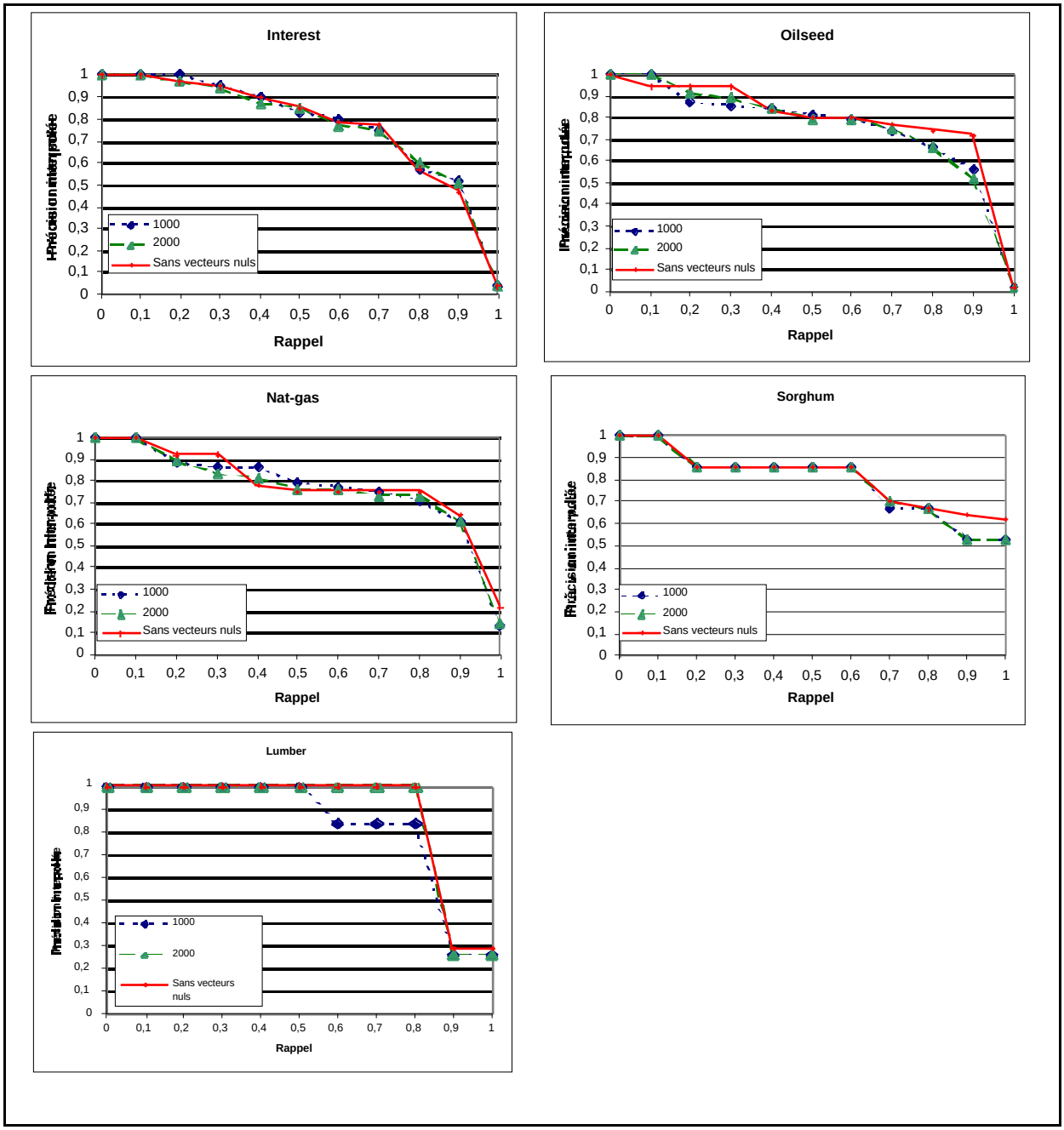


Figure 7.19 : *Courbes rappel-précision interpolée en fonction du choix des dépêches non pertinentes dans la base d'apprentissage.*

Une expérience similaire a été menée sur les cinquante thèmes du corpus TREC-8 où, pour chaque thème, deux bases d'apprentissage différentes sont testées. Dans les deux cas, tous les documents pertinents disponibles sont pris en considération, mais dans le premier cas, le sous-ensemble des documents non pertinents est constitué de 1300 documents sélectionnés aléatoirement et dans le deuxième cas, le sous-ensemble des textes non pertinents est constitué de vecteurs non nuls compte tenu des descripteurs sélectionnés.

La Figure 7.20 est la courbe rappel-précision interpolée obtenue pour les cinquante thèmes dans chacun des cas. Cette courbe est obtenue en calculant pour chaque valeur de rappel r, la moyenne des précisions interpolées obtenues pour chaque thème pour cette même valeur r.

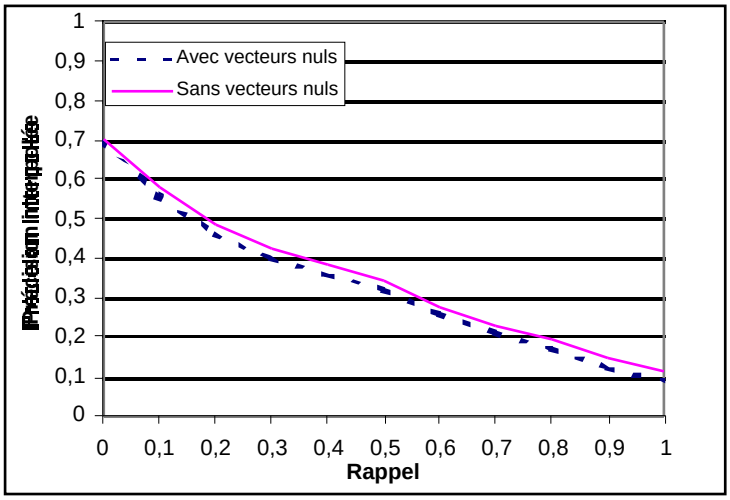


Figure 7.20 : *Courbes rappel-précision interpolée obtenues pour l'ensemble des cinquante thèmes du corpus TREC-8. Chaque point de la courbe est obtenu en effectuant la moyenne de chacun des thèmes.*

Ces expériences, menées sur des corpus différents, montrent qu'il est préférable d'utiliser des bases d'apprentissage qui ne comportent pas de vecteur nul dans le sous-ensemble des documents non pertinents.

Cette approche présente l'avantage de créer des bases d'apprentissage de taille réduite, donc de limiter les temps de calculs. De plus, elle permet de s'affranchir du choix de la taille de la base en proposant un choix systématique adapté à chaque thème.

7.4 Étude du nombre de neurones cachés

Jusqu'ici, toutes les expériences ont été effectuées avec aucun neurone caché, c'est-à-dire avec une régression logistique. Un tel modèle est appelé modèle linéaire car la surface de séparation

obtenue est un hyperplan dans l'espace des descripteurs. En ajoutant des neurones cachés dans la couche cachée, on crée des surfaces de séparation plus complexes qui peuvent améliorer les performances de classification.

7.4.1 Théorème de Cover

Avant d'essayer d'ajouter des neurones cachés à la structure neuronale, il faut s'assurer que le problème de classification traité n'est pas intrinsèquement linéairement séparable, auquel cas il est inutile d'essayer d'ajouter des non-linéarités à la surface de séparation.
Le théorème de Cover [Cover, 1965] précise les conditions dans lesquelles un problème de classification est toujours linéairement séparable.

Son énoncé est le suivant :
Soit un problème de classification à deux classes, dont les deux classes sont équiprobables, comprenant N exemples dans un espace de dimension d, alors :

- Si $N \leq d + 1$, n'importe quelle dichotomie des exemples est linéairement séparable.
- Si $N = 2(d + 1)$, n'importe quelle dichotomie est linéairement séparable avec une probabilité de 0,5.

Pour les problèmes de classification traités, le nombre d'exemples pertinents peut être inférieur au nombre de descripteurs utilisés dans la représentation des textes. Cependant, le nombre de documents non pertinents disponibles sur les corpus étudiés est très élevé, comme on l'a précisé au paragraphe 7.2.2.1. Même si une grande partie des documents non pertinents est représentée par le vecteur nul, il est toujours possible de choisir un nombre de documents non pertinents suffisamment élevé pour être dans les conditions du théorème de Cover où le problème de classification n'est pas, *a priori*, linéairement séparable.
Il n'est donc pas évident, *a priori*, que le meilleur classifieur soit un séparateur linéaire.

7.4.2 Variations des performances en fonction du nombre de neurones cachés

Pour faire des comparaisons équitables, il est important de mener correctement l'apprentissage des réseaux de neurones contenant des neurones cachés : il faut utiliser une méthode de *weight decay* et en déterminer les hyperparamètres. Or comme il a été précisé dans le chapitre 6, pour

les réseaux de neurones à couches, il est préférable d'utiliser plusieurs hyperparamètres dont le réglage peut être délicat.

Cependant, comme on le verra dans le chapitre 9, il est possible de fabriquer des bases d'apprentissage de grande taille sur le corpus AFP qui ne nécessitent donc pas nécessairement de méthodes de régularisation. Les différentes expériences qui ont pu être faites sur ces thèmes du corpus de l'AFP ont montré que l'ajout de neurones cachés n'améliorait pas les résultats [Stricker *et al*., 2000].

Sur l'ensemble des cinquante thèmes du corpus TREC-8, des architectures à 0, 1 ou 2 neurones cachés ont été testées avec une méthode d'arrêt prématuré explicitée au paragraphe 7.6.2.
Pour ces différentes architectures, les vecteurs d'entrées des réseaux de neurones sont identiques et les résultats de la Figure 7.21 sont calculés sur la base de test en moyennant les précisions moyennes non interpolées de chaque thème.

	0 neurone caché	1 neurone caché	2 neurones cachés
Moyenne des précisions moyennes non interpolées	33,18	33,09	28,92

Figure 7.21 : *Comparaison des résultats obtenus sur le corpus TREC-8 avec différentes architectures. Les thèmes sans documents pertinents sur la base de test ne sont pas pris en considération.*

Les résultats montrent que, en moyenne, le modèle linéaire donne de meilleurs résultats que le modèle non linéaire.
Il existe cependant des différences thème par thème : sur la Figure 7.22, chaque thème est représenté par un point, dont l'abscisse est sa précision moyenne non interpolée calculée avec le modèle linéaire, et dont l'ordonnée est celle issue du modèle avec deux neurones cachés. La droite $y = x$ sépare le plan en deux : si un point est dans le demi-plan inférieur, le modèle linéaire est meilleur et dans le cas contraire, le modèle avec deux neurones cachés est meilleur.

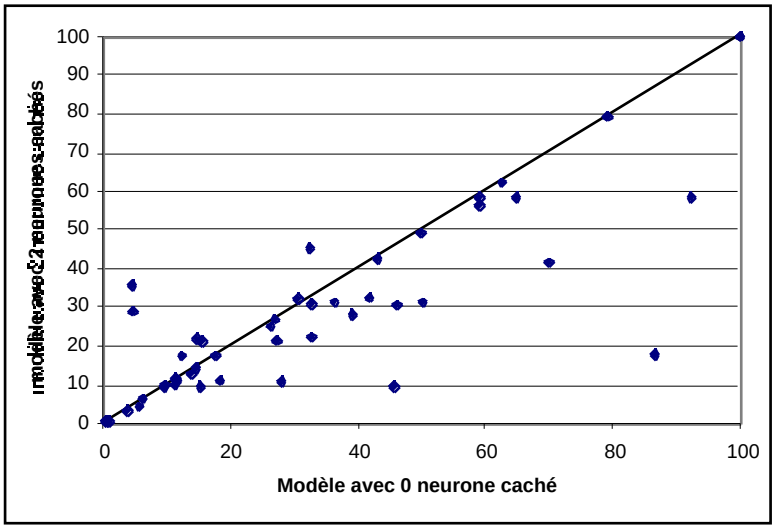


Figure 7.22 : *Résultat thème par thème pour les cinquante thèmes du corpus TREC-8. Chaque point représente un thème, son abscisse est l'interpolation moyenne non interpolée obtenue avec aucun neurone caché et son ordonnée est celle obtenue avec deux neurones cachés.*

Les résultats montrent que, pour la grande majorité des thèmes le modèle avec aucun neurone caché est meilleur que le modèle avec deux neurones cachés. Les trois thèmes pour lesquels le modèle à 2 neurones cachés apporte des améliorations significatives sont les thèmes 362, 382, et 393 qui ont respectivement 4, 5, et 5 documents pertinents sur la base de test et pour lesquels la mesure est donc très sensible.

Finalement, ces résultats montrent que le fait d'ajouter des neurones cachés n'améliore pas les performances en moyenne, et il semble même vain de chercher à optimiser ce nombre pour chaque thème puisque, dans la grande majorité des cas, le modèle le plus simple est le meilleur.

7.4.3 Conclusion sur le nombre optimal de neurones cachés

L'ensemble de ces résultats, obtenus sur des corpus différents, montre que l'ajout de neurones cachés et donc l'utilisation de surfaces de séparation non linéaires n'améliore pas les résultats, et, la plupart du temps, les diminue.

Cette observation est assez étonnante d'autant plus que le langage naturel est *a priori* complexe. Cependant, ces observations confirment les résultats obtenus par d'autres auteurs qui utilisent des réseaux de neurones pour faire de la catégorisation de textes. Ces travaux sont rapidement décrits ci-dessous.

[Wiener, 1993] utilise des réseaux de neurones avec la même architecture que celle présentée à la Figure 7.1, sur l'ancienne version du corpus Reuters (Reuters-22173). Dans cette étude, plusieurs architectures avec des nombres de neurones cachés différents sont testées, allant jusqu'à 6 neurones cachés. Pour ces différentes études, les sélections de descripteurs ont été effectuées soit par des méthodes de sélections de termes comme la méthode du chi-2, soit par des variantes de la méthode *latent semantic indexing* [Deerwester *et al.*, 1990] qui proposent une approche différente des méthodes de sélections précédentes, s'appuyant sur une décomposition en valeurs singulières. Les apprentissages sont effectués en conjuguant une méthode d'arrêt prématuré avec une méthode de pénalisation des poids

D'après les résultats obtenus dans cette étude, les réseaux de neurones avec neurones cachés n'obtiennent pas de résultats significativement meilleurs que les réseaux sans neurone caché et ce, quelle que soit la méthode de sélection de descripteurs :

"*There was surprisingly little gain in effectiveness of the non linear networks over the linear networks"*

Afin de prouver que ses algorithmes ne sont pas en cause, il invente des thèmes artificiels qui sont des compositions de thèmes existants, et qui nécessitent des modèles non linéaires. Dans ce cas, les réseaux de neurones avec des neurones cachés obtiennent bien de meilleurs résultats que les réseaux linéaires, ce qui tend à prouver qu'il ne s'agit pas d'un problème algorithmique, mais bien d'un problème structurel.

[Schütze *et al.*, 1995] ont également étudié une approche neuronale pour effectuer de la catégorisation de textes sur les corpus issus de TREC-2 et TREC-3. Dans leurs expériences, la sélection de descripteurs est effectuée comme précédemment, soit par la méthode *latent semantic indexing,* soit par la méthode du chi-2. Ils comparent également une architecture ne contenant pas de neurone caché avec une architecture contenant trois neurones cachés, l'apprentissage s'effectuant avec la méthode de l'arrêt prématuré. Pour cette expérience également, l'architecture comprenant des neurones cachés n'apporte pas d'améliorations significatives par rapport à l'architecture sans neurone caché :

"*It is safe to conclude that the non-linear components to the neural network provides absolutely no advantage*"

[Yang et Liu, 1999] utilisent des réseaux de neurones avec des neurones cachés sur le corpus Reuters-21578, et fixent le nombre de neurones cachés grâce à un ensemble de validation (dont ils ne précisent pas la composition). Ils testent différentes architectures avec 16, 64 ou 160 neurones cachés et grâce à la base de validation, ils choisissent un nombre de neurones cachés égal à 64. Pour leurs expériences, ils sélectionnent 1000 descripteurs grâce au calcul de l'information mutuelle, et par conséquent leurs réseaux de neurones comportent environ 64.000 poids à déterminer.

Ils ne font pas ici de comparaison avec un simple modèle linéaire, mais il faut noter que les résultats obtenus avec leurs réseaux sont assez faibles en comparaison des autres méthodes, si bien qu'il est impossible de dire si les neurones cachés ont permis une amélioration des résultats.

7.4.4 Vers une représentation plus élaborée

Pourquoi les réseaux linéaires obtiennent-ils de meilleurs résultats que les réseaux non linéaires ?

Une des explications provient des méthodes de sélection des descripteurs qui sélectionnent les descripteurs un par un sans tenir compte de leurs interactions éventuelles en particulier des méthodes comme l'information mutuelle ou le chi-2, cette critique étant moins vraie pour la méthode d'orthogonalisation de Gram-Schmidt et la méthode *latent semantic indexing*.

Cependant toutes les études citées, ainsi que la nôtre, conduisent aux mêmes conclusions, bien qu'elles utilisent des corpus différents qui regroupent une grande variété de situations.

Il semble que la représentation des textes par le modèle vectoriel conduise à des ensembles d'exemples qui admettent comme meilleur classifieur un séparateur linéaire. Bien entendu, cette situation est une conséquence directe de la représentation choisie, et il n'est pas certain qu'elle se retrouve pour des représentations plus élaborées telles que celles que nous décrirons dans le chapitre 8.

7.5 Mise en œuvre sur l'ensemble du corpus Reuters

Ce paragraphe présente les résultats obtenus sur l'ensemble des 90 catégories du corpus Reuters à partir de l'ensemble des remarques effectuées dans ce chapitre. Ces résultats peuvent être comparés à d'autres résultats de la littérature.

7.5.1 Choix des paramètres du modèle

Pour la sélection des descripteurs, les paramètres choisis sont les suivants :

- Si le nombre de documents pertinents est supérieur à cent, on choisit 3000 documents non pertinents et 100 descripteurs initiaux.
- Si le nombre de documents pertinents est supérieur à dix, on choisit 500 documents non pertinents et 50 descripteurs initiaux.
- Lorsque le nombre de descripteurs pertinents est inférieur à dix, les dix premiers descripteurs trouvés par la méthode du vocabulaire spécifique sont sélectionnés.

Le modèle est une régression logistique, et l'apprentissage est effectué avec la méthode du *weight decay* et un hyperparamètre fixé à 1 ; la base d'apprentissage est constituée de tous les documents pertinents, et de documents non pertinents pour lesquels la représentation est différente du vecteur nul.

7.5.2 Performances sur l'ensemble du corpus Reuters

Plusieurs mesures ont été calculées pour faciliter les comparaisons avec les autres travaux : la précision moyenne non interpolée (UAP), la moyenne sur 11 points (11-pt), la mesure de F (selon les définitions exposées au chapitre 4).

La Figure 7.23 présente les macro-moyennes pour chacune de ces mesures sur l'ensemble des thèmes. Les moyennes sont aussi calculées pour plusieurs sous-ensembles de thèmes groupés en fonction du nombre de documents pertinents sur la base d'apprentissage :

- Les dix premiers thèmes, car beaucoup d'auteurs publient des résultats sur ces thèmes qui ont le plus de documents pertinents.
- Les thèmes 11 à 40, pour les thèmes possédant plus de vingt-cinq documents pertinents.

- Les thèmes 41 à 60, pour les thèmes ayant entre dix et vingt-cinq documents pertinents.
- Les thèmes 61 à 90, pour les thèmes possédant moins de dix documents pertinents, et pour lesquels la sélection de descripteurs est effectuée différemment des autres.

	Nombre de descripteurs	UAP	11-pt	*F*
Moyenne sur l'ensemble des thèmes	16,8	72,6	72,8	66,4
Moyenne pour les thèmes 1 à 10	32,2	89,7	87,2	85,3
Moyenne pour les thèmes 11 à 40	18,3	83,2	83,6	81,4
Moyenne pour les thèmes 41 à 60	17,4	69,2	69,1	69,1
Moyenne pour les thèmes 61 à 90	10,0	59,0	59,6	43,8

Figure 7.23 : *Ensemble des résultats sur le corpus Reuters.*

Ces expériences montrent, que, en moyenne, les performances décroissent lorsque nombre de documents pertinents de la base d'apprentissage diminue.

7.5.3 Influence de la valeur de l'hyperparamètre

Pour tester la sensibilité des résultats à la valeur de l'hyperparamètre, plusieurs valeurs ont été testées. La variation de la précision moyenne non interpolée en fonction de la valeur de l'hyperparamètre α est présentée à la Figure 7.24. Ces résultats montrent que la valeur choisie a peu d'influence sur les résultats tant qu'elle n'est pas trop élevée. La valeur 5 donne clairement de moins bons résultats sur les catégories comprenant peu d'exemples d'apprentissage, car dans ce cas, le terme de pénalisation prend trop d'importance par rapport au terme d'entropie croisée, et les poids tendent vers zéro.

	$\alpha = 0,1$	$\alpha = 0,5$	$\alpha = 1,0$	$\alpha = 5,0$
Moyenne sur l'ensemble des thèmes	72,4	72,5	72,6	68,3

Moyenne pour les thèmes 1 à 10	89,2	89,5	89,7	89,6
Moyenne pour les thèmes 11 à 40	82,0	82,9	83,2	82,3
Moyenne pour les thèmes 41 à 60	70,2	69,4	69,2	65,9
Moyenne pour les thèmes 61 à 90	59,1	59,0	59,0	49,3

Figure 7.24 : *Comparaison des performances mesurées avec la précision moyenne non interpolée en fonction de la valeur de l'hyperparamètre.*

7.5.4 Hyperparamètre déterminé par la méthode d'intégration

Jusqu'ici l'hyperparamètre était fixé à une valeur constante pendant l'apprentissage, mais l'approche bayesienne présentée au chapitre 6 a montré que la valeur de l'hyperparamètre pouvait être fixée pendant l'apprentissage.

La méthode d'intégration décrite au chapitre 6 a été utilisée : si p est le nombre de poids du réseau, alors l'hyperparamètre est calculé régulièrement pendant l'apprentissage selon la formule :

$$\alpha = \frac{p}{\sum_{i=1}^{p} w_i^2}$$

La valeur initiale de l'hyperparamètre est fixée à 1.

La comparaison des performances obtenues avec cette méthode et avec un hyperparamètre fixé à 1 est présentée à la Figure 7.25.

Dans ce cas, la méthode d'intégration issue de l'approche bayesienne n'apporte pas d'amélioration : les résultats sont très proches. La méthode d'intégration nécessite cependant plus de calculs : il semble préférable de se contenter d'une valeur constante de l'hyperparamètre, d'autant plus que les résultats de la Figure 7.24 montrent que le choix de cette valeur n'est pas critique.

	α variable	α = 1
Moyenne sur l'ensemble des thèmes	71,6	72,6

Moyenne pour les thèmes 1 à 10	89,4	89,7
Moyenne pour les thèmes 11 à 40	82,2	83,2
Moyenne pour les thèmes 41 à 60	67,8	69,2
Moyenne pour les thèmes 61 à 90	58,1	59,0

Figure 7.25 : *Comparaison des performances mesurées avec la précision moyenne non interpolée en fonction de la valeur de l'hyperparamètre.*

La méthode de maximisation n'a pas été testée sur ce problème, car l'ensemble des résultats semble indiquer que la valeur de l'hyperparamètre, avec ce modèle, a peu d'influence.

7.5.5 Utilisation de racines lexicales

Jusqu'ici, les descripteurs utilisés étaient les mots tel qu'ils apparaissaient dans les textes ; chaque flexion d'un mot était considérée comme un descripteur différent.

Dans l'expérience décrite ci-dessous, tous les mots des textes sont remplacés par leur racine lexicale selon la méthode décrite au chapitre 5**,** et les fréquences d'apparition de chaque racine sur l'ensemble du corpus sont calculées**.** La détermination du vocabulaire spécifique, ainsi que la sélection de descripteurs, sont ensuite effectuées exactement comme précédemment pour chaque thème.

La Figure 7.26 montre les dix premiers descripteurs sélectionnés pour le thème *interest* lorsque les textes sont conservés tel quel ou lorsque les mots sont substitués par leur racine. La première liste contient les mots *rates* et *rate* qui deviennent la racine *rate* dans la deuxième liste. Il faut noter cependant que la liste des descripteurs sélectionnés à partir des textes utilisant les racines n'est pas identique à la liste des racines des descripteurs sélectionnés à partir des textes originaux : par exemple, le mot *opened* est sélectionné dans les descripteurs obtenus à partir des textes originaux, mais sa racine *open* ne fait pas partie des descripteurs sélectionnés à partir des textes utilisant les racines : *opened* est une forme relativement rare qui peut être discriminante, mais sa racine *open* est un mot très courant qui n'est plus discriminant.

Normal	Racine
Rate	rate
money	monei
customer	fed
prime	prime
england	england
rates	custom
band	band
bundesbank	bundesbank
discount	discount
opened	repurchas

Figure 7.26 : *Thème interest : liste des dix premiers descripteurs sélectionnés à partir des textes originaux ou à partir des textes avec les racines.*

Les performances sur l'ensemble des catégories du corpus sont présentées à la Figure 7.27 ; ces résultats sont à comparer avec les résultats obtenus à la Figure 7.23 sans l'utilisation de racines.

Quel que soit l'ensemble de catégories considérées, les résultats sont systématiquement inférieurs à ceux obtenus en conservant les mots inchangés.

	Nombre de descripteurs	UAP	11-pt	*F*
Moyenne sur l'ensemble des thèmes	14,5	71,4	71,5	66,3
Moyenne pour les thèmes 1 à 10	28,8	89,4	86,3	84,9
Moyenne pour les thèmes 11 à 40	15,1	82,5	82,7	81,0
Moyenne pour les thèmes 41 à 60	13,9	67,5	68,3	65,4
Moyenne pour les thèmes 61 à 90	10,0	57,4	58,0	46,4

Figure 7.27: *Ensemble des résultats sur le corpus Reuters avec les racines.*

Les résultats précédents sont des moyennes. La Figure 7.28 montre, pour les soixante premiers thèmes, les différences thème à thème : chaque point représente un thème, l'abscisse d'un point est la précision moyenne non interpolée obtenue sans l'utilisation de racines et son ordonnée est celle obtenue en substituant les mots par leur racine.

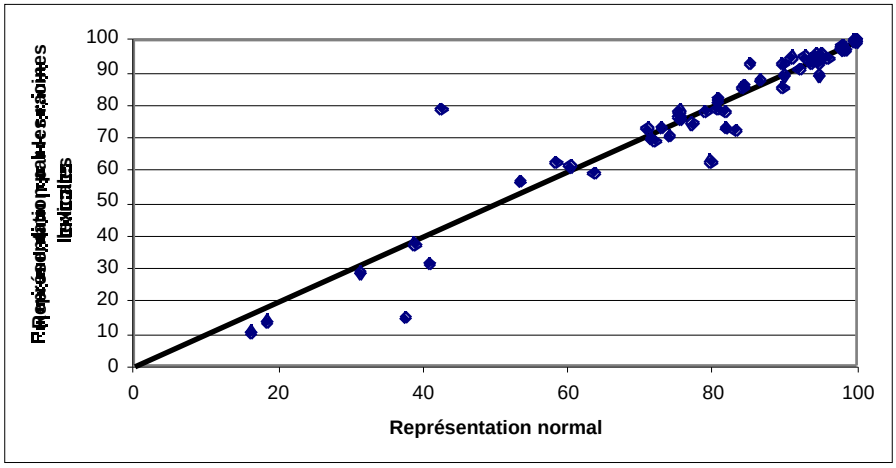


Figure 7.28 : *Comparaison des précisions moyennes non interpolées pour les soixante premiers thèmes : représentation normal ou représentation avec des racines lexicales.*

Le seul thème pour lequel il existe une grande différence est le thème *lumber* (thème 60 selon notre classement) dont la précision moyenne non interpolée passe de 0,42 à 0,78 grâce à

l'utilisation de racines. Cette différence est due à la présence du mot *wood* dans la liste des descripteurs sélectionnés lorsque l'on utilise les racines : la présence de ce mot permet de classer le document pertinent 20790 en tête de classement et, comme il n'existe que 6 documents pertinents pour ce thème, cela produit une grande différence de performance.

Pour l'ensemble des autres thèmes, les différences sont peu importantes et, selon les thèmes, certaines performances sont améliorées et d'autres dégradées. Cependant, comme l'ont montré les performances de la Figure 7.27, il semble préférable, en moyenne, de conserver les mots originaux plutôt que de les remplacer par leur racine lexicale.

L'algorithme utilisé n'est pas parfait, et certains rapprochements de mots sont injustifiés. Dans la liste des racines de la Figure 7.26 se trouve, par exemple, la racine *custom*. Or cette racine correspond aux mots *customers* (clients) et *customer* (client), mais elle correspond également au mot *customs* (douane). Donc avec l'utilisation de racines, les mots *douanes* et *clients* sont considérés comme identiques, alors que leur signification est évidemment différente. Par conséquent, selon les catégories et les descripteurs utilisés, l'utilisation de racines est susceptible soit d'améliorer, soit de dégrader, les performances de classification.

L'impact de l'utilisation des racines a beaucoup été étudié dans la communauté de la recherche d'informations ; différents algorithmes ont été utilisés, comme celui développé par [Lovins, 1968] ou d'autres fondés sur des analyses morphologiques [Hull, 1996]. Les conclusions de ces études sont parfois différentes, mais il semble que, globalement, les conclusions soient proches de nos résultats.

Dans son étude, [Harman, 1991] a testé plusieurs algorithmes, et conclut que l'utilisation de racines n'améliore pas les résultats, mais [Krovetz, 1993] a observé des améliorations significatives grâce à leur utilisation. [Hull, 1996] a également proposé une étude intensive sur l'utilisation des racines et a montré que les améliorations de performances étaient faibles en moyenne. Dans cette étude, il étudie précisément certaines requêtes et exhibe plusieurs séries de mots regroupés sous la même racine tout en ayant des sens différents comme le mot *server* de l'expression *client-server* qui devient *serve* c'est-à-dire un verbe extrêmement commun.

Dans toutes ces études, certains thèmes voient leur performance s'améliorer, d'autres voient leur performance se dégrader, si bien que, en moyenne, les améliorations éventuelles sont faibles.

Le problème de l'utilisation des racines provient essentiellement de rapprochement de mots avec des sens différents sous la même racine. Ces rapprochements fabriquent de faux synonymes et, finalement, avec notre modèle, il semble préférable de ne pas utiliser de racines.

7.5.6 Utilisation de lemmes

Le défaut principal des racines étant de regrouper trop de mots différents sous une même racine, l'utilisation de lemmes pourrait résoudre ce problème, car les rapprochements de mots sont effectués sur la base d'une analyse grammaticale grâce à l'algorithme présenté au chapitre 5.

La Figure 7.29 montre les dix premiers descripteurs sélectionnés pour le thème *interest* selon l'utilisation ou non de lemmes (la liste obtenue avec les racines est également présente pour faciliter les comparaisons). Par rapport à la liste de la Figure 7.26, *customer* n'est plus maintenant assimilé à *customs*.

Normal	**Lemme**	**Racine**
rate	rate	rate
money	customer	monei
customer	england	fed
prime	prime	prime
england	money	england
rates	band	custom
band	repurchase	band
bundesbank	discount	bundesbank
discount	feed	discount
opened	cuts	repurchas

Figure 7.29 : *Thème interest : liste des dix premiers descripteurs sélectionnés à partir des textes originaux, des lemmes et des racines lexicales.*

Les performances sur l'ensemble des catégories du corpus sont présentées à la Figure 7.30 ; ces résultats sont à comparer avec les résultats obtenus à la Figure 7.23 sans l'utilisation de lemmes, et avec les résultats de la Figure 7.27 obtenus avec les racines.

	Nombre de descripteurs	UAP	11-pt	*F*
Moyenne sur l'ensemble des thèmes	15,3	70,6	70,7	66,0
Moyenne pour les thèmes 1 à 10	30,4	89,5	86,8	84,7
Moyenne pour les thèmes 11 à 40	15,6	82,8	82,8	80,6
Moyenne pour les thèmes 41 à 60	15,5	66,4	67,1	65,8
Moyenne pour les thèmes 61 à 90	10,0	55,2	56,0	45,6

Figure 7.30 : *Ensemble des résultats sur le corpus Reuters avec les lemmes.*

Comme précédemment, l'utilisation de lemmes détériore légèrement les résultats. La Figure 7.31 qui permet de visualiser les différences pour chaque catégorie (pour les soixante premières) montre peu de différences entre les méthodes.

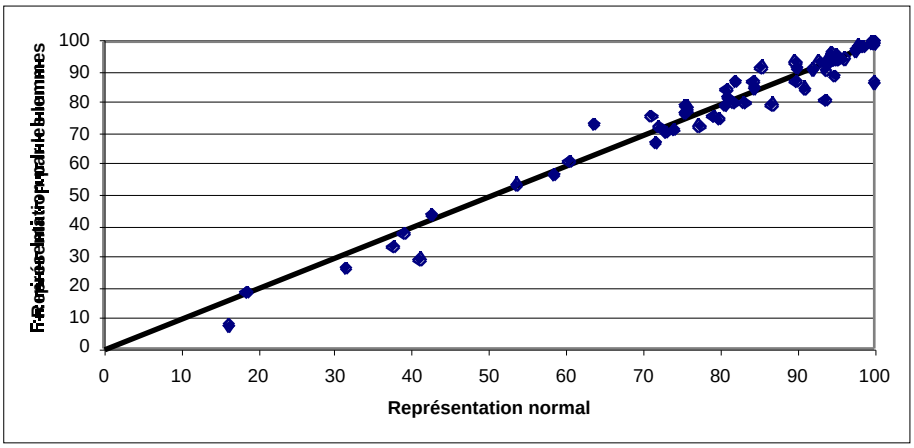


Figure 7.31 : *Comparaison des précisions moyennes non interpolées pour les soixante premiers thèmes : représentation normal ou représentation avec des lemmes.*

L'effet de la lemmatisation sur les problèmes de catégorisation de textes a été moins étudié que l'effet de substitution des mots par leur racine. [Kindermann et Leopold, 2000] ont utilisé des machines à vecteurs supports pour de la catégorisation de textes sur un corpus en langue allemande et ont comparé les performances obtenues lorsque les mots sont conservés tel quel ou lorsqu'ils sont remplacés par leur lemme. Ils observent une diminution des performances avec l'utilisation des lemmes, bien que la langue allemande soit une langue qui présente beaucoup de flexions.

Finalement l'utilisation de racines ou de lemmes n'est pas souhaitable pour nos modèles, car ces approches rendent les mots plus ambigus et finalement beaucoup d'auteurs

7.6 Mise en œuvre sur le corpus TREC-8

Ce paragraphe présente les expériences qui ont été faites pour fournir les résultats de la compétition TREC-8 [Stricker *et al.*, 2000b] au cours de l'année 1999 ; toutes les remarques faites ci-dessus ne sont pas nécessairement prises en considération, car ces résultats ont été obtenus avant la date limite du 31 août 1999. Ce paragraphe détaille le choix des paramètres qui avaient été faits pour obtenir les résultats, afin d'obtenir une référence claire pour les comparaisons. Comme on l'a précisé au chapitre 3, tous les apprentissages sont effectués avec le fichier des pertinences disponible avant la compétition et les performances sont évaluées sur la base de test en fonction du fichier des pertinences fourni après la compétition.
La synthèse et les comparaisons des résultats de la tâche de filtrage de TREC-8 sont effectuées dans [Hull et Robertson, 2000] ; une description succincte des différentes approches y est également présentée.

Nous ne reprenons pas ici la description du corpus et des données, qui a été faite au chapitre 3.

7.6.1 Les paramètres de la sélection des descripteurs

Pour chaque thème, on détermine la liste du vocabulaire spécifique grâce à la méthode exposée au chapitre 5.

La sélection de descripteurs est effectuée en construisant la matrice X nécessaire à l'orthogonalisation de Gram-Schmidt. On considère systématiquement, pour chaque thème, les 50 premiers mots de la liste du vocabulaire spécifique (cinquante descripteurs initiaux avec la terminologie utilisée jusqu'ici) et 1300 dépêches non pertinentes sélectionnées aléatoirement.
En utilisant l'algorithme d'orthogonalisation de Gram-Schmidt couplé avec le critère d'arrêt, le nombre moyen de descripteurs sélectionnés sur l'ensemble des cinquante thèmes est de vingt-cinq.

7.6.2 Apprentissage du réseau de neurone

Pour constituer la base d'apprentissage, l'ensemble des documents pertinents disponibles est pris en considération, et le nombre de documents non pertinents utilisés est choisi en fonction du nombre de documents pertinents : 2000 s'il y a plus de 30 documents pertinents et 1300 sinon.

Pour coder les composantes x_i des vecteurs d'entrées, on utilise le codage suivant :

$$x_i = \begin{cases} -1 & si\ TF_j(i) = 0 \\ \dfrac{TF_j(i)}{Log(L_j)} & si\ TF_j(i) > 0 \end{cases}$$

$TF_j(i)$ est la fréquence d'un terme i dans un texte j et L_j est la longueur de ce texte mesurée par le nombre de mots.

Le réseau de neurones est une régression logistique comme à la Figure 7.2, l'apprentissage s'effectuant en minimisant l'entropie croisée. La régularisation est faite grâce à la méthode de l'arrêt prématuré : la minimisation est faite avec une simple descente de gradient pendant quelques centaines d'itérations. Comme cet algorithme ne converge pas vers le minimum de la fonction de coût, les poids ne prennent pas de grandes valeurs et le surajsutement est évité.

7.6.3 Résultats de la compétition TREC-8

Pour la tache de routing à laquelle nous avons participé, les performances sont mesurées par la précision moyenne non interpolée : on calcule cette valeur pour chaque thème, puis la performance globale est simplement la moyenne de l'ensemble. Les thèmes ne comportant aucun document pertinent sur la base de test sont comptés avec une performance de zéro.

Sur l'ensemble des cinquante thèmes, la performance globale est de **30,7** (la performance maximale possible étant de 96,0 car deux thèmes n'ont aucun document pertinent sur le test).

Les résultats de l'ensemble des participants sont présentés à la Figure 7.32, chaque histogramme représentant un résultat. Les différents systèmes sont tous repérés par un nom qui est le nom officiel donné lors de la compétition ; notre système porte le nom **S2N2**.
Excepté pour notre système, chaque participant a présenté deux ensembles de résultats correspondant, en général, à des jeux de paramètres différents.

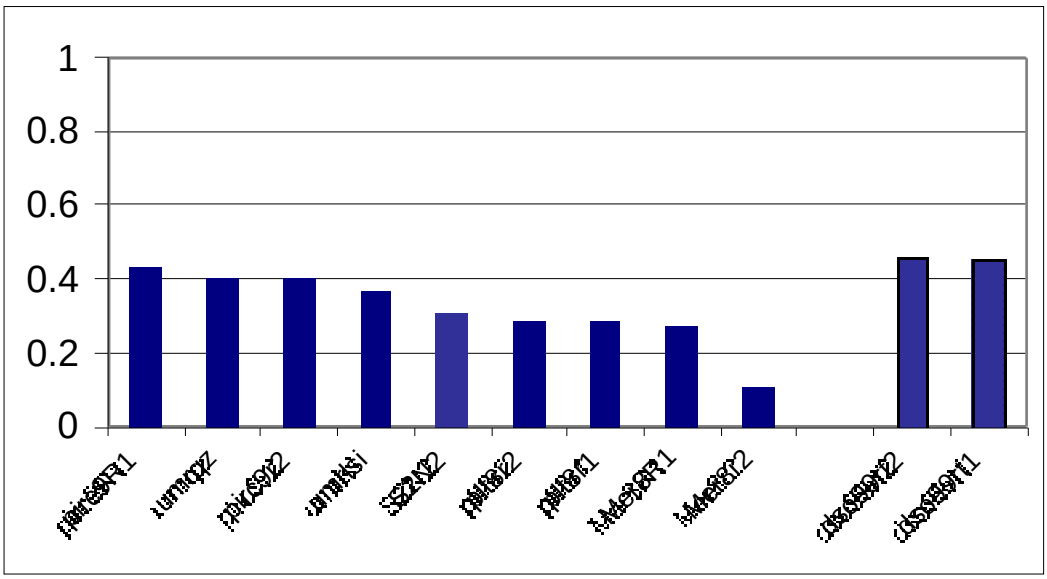


Figure 7.32 : *Moyenne des résultats obtenus sur l'ensemble des cinquante thèmes pour les participants de la tâche du routing.*

7.6.4 Revue des différents systèmes de la sous tâche de routing

Dans ce paragraphe, nous détaillons rapidement les approches utilisées par chacun des candidats, avec les références bibliographiques qui s'y rapportent.

[Ng *et al.*, 2000] ont présenté les systèmes ***dso99rt2*** et ***dso99rt1***. Leurs résultats sont présentés séparément des autres candidats, car ils ont utilisé comme descripteurs potentiels des annotations manuelles qui figuraient dans les documents du *Financial Times*. Ces annotations manuelles n'ont pas été utilisées par les autres candidats et, par conséquent, il est difficile de faire des comparaisons directes puisque leur modèle a été construit avec des informations supplémentaires par rapport aux autres.

Leurs deux systèmes reposent sur l'utilisation de l'algorithme du perceptron. La sélection des descripteurs est faite grâce à une méthode issue de l'algorithme de Rocchio amélioré [Singhal, 1998] [Schapire *et al.*, 1998], qui permet d'obtenir un classement des descripteurs par ordre de pertinence décroissante. Les cent premiers descripteurs trouvés, ainsi que les vingt meilleures paires (une paire étant définie par deux mots non vides adjacents), pour chaque thème, sont utilisés en entrée.

[Kwok *et al.*, 2000] ont présenté les systèmes ***pirc9r1*** et ***pirc9r2***. Pour le système *pirc9r1*, six profils différents sont fabriqués avec des coefficients différents. Parmi ces six profils, deux utilisent uniquement la requête, et les quatre restants utilisent la base des documents pertinents. Ces différents profils sont combinés linéairement afin d'obtenir un score pour chaque document, les coefficients affectés à chaque profil sont trouvés par un algorithme génétique pour d'optimiser la précision moyenne non interpolée.
Pour le deuxième système *pirc9r2* deux nouveaux profils sont ajoutés à la combinaison.

[Oard et Wang, 2000] ont présenté les systèmes ***umrlqz*** et ***umrlsi***. Pour leurs expériences, ils utilisent des méthodes issues de la recherche d'informations et plus particulièrement l'algorithme de Rocchio pour le système *umrlqz*. Pour le système *umrlsi*, ils essayent de faire du filtrage collaboratif grâce à la méthode *latent semantic indexing* pour trouver des structures communes à des sous-ensembles de thèmes. L'utilisation de cette deuxième méthode dégrade les résultats, peut être parce qu'il n'existe pas particulièrement de points communs entre les différents thèmes.

[MacFarlane et Robertson, 2000] ont présenté les systèmes ***plt8r1*** et ***plt8r2***. Ils ont cherché à mesurer les performances du système PLIERS qui s'appuie sur la stratégie du système Okapi de TREC 5 [Beaulieu *et al.*, 1997] : une partie de la base d'apprentissage est utilisée pour l'extraction de descripteurs et une autre partie pour la sélection. Leurs expériences ont été faites en utilisant seize ordinateurs dotés de Pentium II en parallèle qui indexent la base d'apprentissage (64.000 documents) en 5 minutes, et la base de test (140.000 documents) en 11 minutes.

[Boughanem *et al*., 2000] ont présenté les systèmes ***Mer8r1*** et ***Mer8r2*** qui utilisent un réseau de neurones pour implémenter le modèle probabiliste, couplé à des algorithmes génétiques pour trouver les paramètres d'apprentissage. Ce modèle est donc directement issu des méthodes de recherche d'informations et non pas des méthodes de catégorisation de textes.

7.6.5 Résultats obtenus après la compétition

Nous présentons dans ce paragraphe les résultats que nous avons obtenus après la compétition ; ils prennent en considération les observations faites précédemment et mettent en avant leur influence bénéfique.

Une première expérience est réalisée, pour laquelle la procédure de sélection des descripteurs n'est pas modifiée, mais les changements par rapport au système S2N2 sont les suivants :

1. Utilisation d'un terme de *weight decay* plutôt que la méthode de l'arrêt prématuré.
2. Fabrication des bases d'apprentissage en supprimant les vecteurs nuls.
3. Codage Lnu des vecteurs d'entrée (le codage Lnu a été présenté au chapitre 5).

Le modèle est toujours celui de la Figure 7.2 avec un hyperparamètre fixé à 1.

Avec ces nouveaux paramètres, la moyenne des précisions moyennes non interpolées sur les cinquante thèmes (comme précédemment, les thèmes sans document pertinent sont pris en considération dans la moyenne avec un score nul) devient **34,8** contre 30,7 précédemment.

La Figure 7.33 présente les courbes rappel-précision pour l'ensemble des thèmes, la courbe en pointillé représente le système S2N2 tandis que la courbe en trait plein est obtenue avec les nouveaux paramètres.

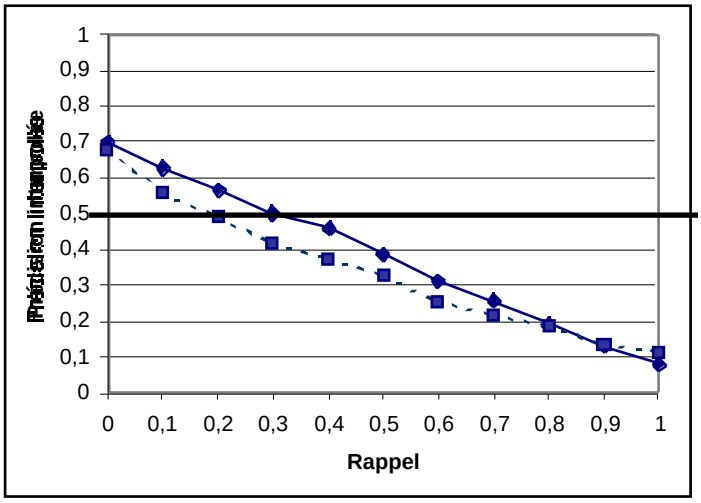


Figure 7.33 : *Courbes rappel-précision pour l'ensemble des 48 thèmes du corpus ayant des documents pertinents sur la base de test. La courbe en pointillée représente le système S2N2 présenté a TREC-8, la courbe en trait plein le nouveau système.*

L'ensemble des modifications a permis d'améliorer significativement les résultats ; cette amélioration se traduit à la fois par une amélioration de la moyenne des précisions moyennes non interpolées, et par le fait que la courbe rappel-précision de l'ensemble des thèmes est systématiquement au-dessus (sauf pour le rappel de 1).

Ces résultats confirment ce qui a été vu au paragraphe 7.3 : la suppression des vecteurs nuls améliore les performances sur la base de test. D'autre part, le codage Lnu proposé dans [Singhal, 1996] est un codage plus performant que notre codage initial. Et de même, la méthode du *weight decay* s'est avérée supérieure à la méthode de l'arrêt prématuré.

Influence de la valeur de l'hyperparamètre

Pour l'expérience précédente, l'hyperparamètre a été fixé à 1, pour l'ensemble des thèmes. En continuant d'utiliser une valeur identique pour l'ensemble des thèmes, il peut être intéressant de voir l'influence globale que peut avoir cette valeur ; plusieurs valeurs de l'hyperparamètre α ont été testées : 0, 0,5, 1, 5, 10. La valeur nulle de l'hyperparamètre correspond à une fonction de coût sans terme de régularisation.

La Figure 7.34 donne les performances obtenues sur l'ensemble des thèmes en fonction de la valeur de l'hyperparamètre et la Figure 7.35 montre l'évolution des courbes rappel-précision interpolée

	α = 0	α = 0,5	α = 1,0	α = 5,0	α = 10,0
Performances (UAP)	24,0	35,6	34,8	32,6	31,7

Figure 7.34 : *Comparaisons des performances en fonction de la valeur de l'hyperparamètre. La performance est la moyenne des précisions moyennes non interpolées sur les cinquante thèmes.*

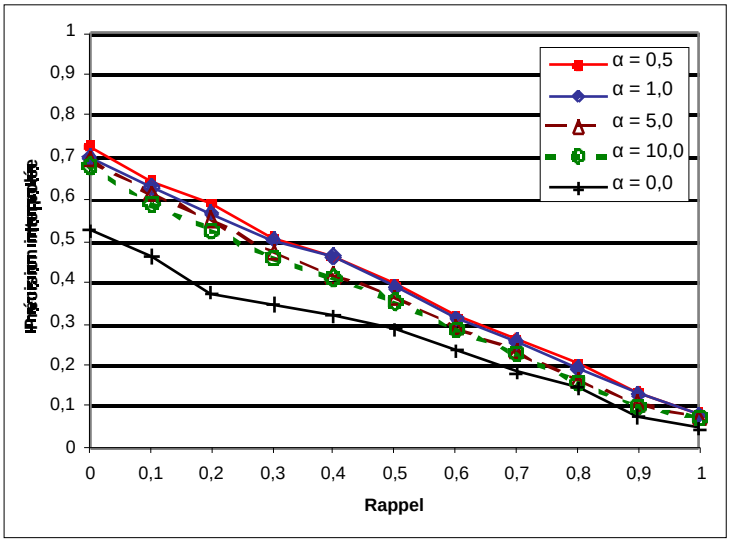


Figure 7.35 : *Courbes rappel-précision pour l'ensemble des thèmes en fonction de la valeur de l'hyperparamètre.*

À partir de l'ensemble des résultats obtenus sur TREC-8, il est possible de tirer plusieurs conclusions :

- Les résultats obtenus sans aucune régularisation sont nettement inférieurs à tous les autres, quelle que soit la valeur de l'hyperparamètre. Par conséquent, il est indispensable d'utiliser une méthode de régularisation lors de l'apprentissage.
- Les performances moyennes ne semblent pas très sensibles à la valeur de l'hyperparamètre ; les performances obtenues avec les valeurs 0,5 ou 1 sont très proches et les performances obtenues avec les valeurs 5 ou 10 sont légèrement inférieures. Il vaut donc mieux choisir des valeurs pas trop élevées pour les hyperparamètres, car les performances moyennes se dégradent légèrement lorsque cette valeur augmente.

7.7 Conclusion

Ce chapitre a permis d'introduire un modèle de filtrage, compétitif par rapport à d'autres méthodes ; dont l'efficacité a été montrée sur deux corpus différents : le corpus Reuters et le corpus de la tache de routing de TREC-8.

Les bonnes performances reposent principalement sur deux points : d'une part une sélection de descripteurs efficace et, d'autre part, l'utilisation d'un terme de *weight decay* lors de l'apprentissage. Nos modèles ont en plus la particularité d'utiliser très peu de descripteurs pour la représentation des textes : 17 en moyenne sur le corpus Reuters et 25 sur le corpus TREC-8.
Il faut noter que si l'ajout d'un terme de *weight decay* s'est révélé essentiel, les performances ne sont pas extrêmement sensibles à la valeur de l'hyperparamètre tant que celle-ci n'est pas nulle. Cependant, la méthode de détermination de l'hyperparamètre par l'approche bayesienne ne s'est pas révélée très efficace. [Wiener, 1993] qui utilise également un terme de régularisation lors des apprentissages des réseaux de neurones a choisi, lui aussi, une valeur constante plutôt que de faire varier l'hyperparamètre pendant l'apprentissage.

Finalement nos résultats vont dans le sens des résultats cités au chapitre 2 sur les approches neuronales [Wiener *et al.*, 1995] [Schütze *et al.*, 1995] : l'ajout de neurones cachés n'améliore pas les résultats et l'utilisation d'une méthode de régularisation est indispensable même avec l'architecture la plus simple.

Si les modèles obtenus sont performants, ils sont loin d'être parfaits, et il semble possible de les améliorer. Cependant ni le nombre de neurones cachés, ni les paramètres d'apprentissage (notamment la valeur de l'hyperparamètre) ne semblent être en mesure d'apporter des améliorations significatives ; de plus, l'étude du chapitre 5 sur les différentes sélections de descripteurs a montré que ces méthodes étaient équivalentes.
Pour progresser, il est nécessaire d'améliorer la qualité de représentation des textes et notamment d'inclure plus d'informations dans cette représentation, afin de définir ensuite une

architecture neuronale adéquate. C'est l'objectif de l'approche originale qui est présentée dans le chapitre suivant.

Chapitre 8 Utilisation du contexte local des mots

Les expériences du chapitre 7 ont montré que les meilleures performances étaient obtenues avec un séparateur linéaire, c'est-à-dire sans neurone caché. Les résultats d'autres auteurs confirment que ce résultat n'est pas dû spécifiquement à la méthode de sélection de descripteurs employée, mais semble être intrinsèque à la représentation des textes en sac de mots. Améliorer la représentation des textes, en enrichissant l'information qu'elle contient, peut conduire à l'utilisation de classifieurs plus complexes. La représentation en sac de mots ne tient compte ni de l'ordre, ni de la distance entre les mots. Dans ce chapitre, nous proposons une représentation des textes qui tient compte du contexte local des mots pour les désambiguïser ; l'architecture neuronale est modifiée pour tenir compte de cette nouvelle représentation.

Ce modèle a été appliqué sur les corpus Reuters et TREC-8 et a également servi pour notre participation à la sous-tâche de routing de TREC-9.

8.1 L'Ambiguïté dans la recherche de textes

8.1.1 Exemples d'ambiguïté sémantiques

La liste du vocabulaire spécifique du thème d'échanges de participations comprend le mot *participation* qui, dans un contexte économique, a un sens précis comme l'illustre la phrase suivante :

- Dexia a pris une *participation* de 35,3 % dans le capital de la banque italienne.

Or, ce mot peut être employé dans un contexte différent comme dans l'expression suivante :

- la *participation* des communistes au gouvernement.

Dans ces deux exemples, les termes proches permettent de déterminer exactement le sens du mot *participation*, car sa présence dans une phrase ne suffit pas à caractériser le concept d'échange de participations entre deux entreprises. Dans le premier exemple, la présence du

mot *capital* précise ce concept, alors que dans le deuxième exemple, la présence du mot *gouvernement* indique que le concept des entreprises est absent.

Cet exemple simple montre que la présence de mots dans le voisinage immédiat de certains mots-clefs peut renforcer, ou annihiler, le concept que le mot est susceptible de représenter.

De même, certains mots associés sous forme de paires peuvent exprimer des concepts très précis, alors que le sens de chacun des mots peut être beaucoup plus vague. Considérons, par exemple, les trois concepts suivants, construits à partir du mot *droits* :

- droits de vote
- droits de douane
- droits de l'homme

Ils définissent des concepts importants à reconnaître. Ici la proximité des mots est primordiale : ce n'est pas parce que *droits* et *vote* figurent dans le même texte que le concept de *droit de vote* est présent. La représentation en sac de mots ne reconnaît pas, de manière systématique, la présence de ces associations.

Par exemple, dans les deux extraits de phrases suivantes, provenant du corpus Reuters, deux sens différents du mot *interest* sont présentés :

- (…) to the decline in the value of the U.S dollar by raising *interest* rates (…).
- Soviet officials said foreign businessmen are expressing strong *interest* in establishing joint enterprises in the Soviet Union (…).

Dans la première phrase, le sens du mot *interest* est clairement précisé par la présence de *rates* juste à côté pour former l'expression *interest rates* (taux d'intérêt) et dans une moindre mesure par la présence du verbe *raising*. Dans la deuxième phrase, l'absence de ces termes indique que le sens n'est pas celui des taux d'intérêt.

8.1.2 Autres travaux sur la désambiguïsation et la recherche d'informations

Plusieurs auteurs ont cherché à savoir si les méthodes de désambiguïsation pouvaient avoir un impact positif sur les systèmes de recherche d'informations et, plus précisément, quelles méthodes de désambiguïsation pouvaient y parvenir. On peut distinguer deux approches générales pour effectuer cette désambiguïsation dans le cadre de la recherche d'informations.

La première approche suppose qu'il existe un nombre fini de sens pour un mot, et repose sur l'utilisation de dictionnaires. Par exemple, [Voorhees, 1993] utilise WordNet[1] [Miller *et al.*, 1990] pour la désambiguïsation de mots en langue anglaise, mais observe une décroissance des performances. Le problème de l'utilisation d'un tel dictionnaire est qu'il suppose qu'il existe un nombre fini de sens pour un mot donné, et d'autre part, il ne couvre pas nécessairement toute l'étendue de vocabulaire. De plus, certaines nuances proposées peuvent posséder un sens linguistique, mais n'être pas nécessairement utiles pour la recherche d'informations. Enfin, certaines nuances peuvent dépendre du domaine que l'on cherche à filtrer : par exemple, dans les deux phrases suivantes qui utilisent le mot *capital* :

- Il détient le *capital* de la société X.
- La société X propose du *capital*-risque.

Le mot *capital* a des sens très proches, mais pour certains filtres, il peut être utile de les différencier.

La deuxième approche repose sur l'utilisation du corpus, et plus précisément, sur une étude du contexte des mots, pour effectuer une désambiguïsation utilisable par les systèmes de recherche d'informations.

L'utilisation la plus simple du contexte consiste à utiliser des paires de mots définies comme étant deux mots non vides adjacents. [Singhal, 1998] a obtenu, par cette méthode, des améliorations de performances sur le corpus de TREC-6 et [Ng *et al.*, 2000] ont également utilisé cette définition des paires pour TREC-8. En revanche [Dumais *et al.*, 1998] observe une diminution des performances de leur système lorsqu'ils considèrent l'utilisation de paires en entrée de leurs machines à vecteurs supports. Le problème qui se pose avec les paires est évidemment celui de l'ordre des mots : par exemple si l'on cherche les documents à propos de "*car insurance rates*", et que la paire *insurance_rates* est utilisée, les documents qui parlent de "*rates for insurance car*" ne vont pas être sélectionnés.

[1] http://www.cogsci.princeton.edu/~wn

D'autres auteurs utilisent une notion de contexte plus large pour effectuer la désambiguïsation. Par exemple, [Cohen et Singer, 1996] ont testé deux notions différentes de contextes. Pour leur système RIPPER, construit à base de règles, le contexte d'un mot est défini comme une liste de mots (généralement de faible taille) qui doivent apparaître en même temps dans n'importe quel ordre et n'importe où dans le texte. Pour leur deuxième système, appelé *sleeping experts*, le contexte est constitué de plusieurs mots ordonnés et proches les uns des autres. Les deux approches, bien que différentes, donnent de bons résultats et montrent que l'utilisation du contexte peut améliorer la qualité des filtres.
Pour [Yarowski, 1995] un mot a un sens principal très corrélé aux mots figurant juste à côté. Pour mettre en œuvre cette approche, [Jing et Tzoukermann, 1999] [Schütze et Pedersen, 1995] définissent des vecteurs de contextes pour désambiguïser les mots et obtiennent une amélioration des performances.

Notre approche est proche de celle développée par [Jing et Tzoukermann, 1999], et repose sur l'utilisation du corpus pour définir le contexte usuel d'un mot. Pour chaque mot, le vecteur de contexte de chaque mot est constitué des cinq mots qui le précèdent et des cinq mots qui le suivent.

8.2 Détermination automatique de vecteurs de contexte

8.2.1 Définition du contexte local d'un mot

Dans la représentation en sac de mots, les descripteurs utilisés pour discriminer les textes pertinents des textes non pertinents sont des mots simples pris séparément les uns des autres. Or, le contexte d'un mot dans le sous-ensemble des textes pertinents est différent du contexte de ce mot dans le sous-ensemble des textes non pertinents. Cette différence peut donc être utilisée pour la discrimination.

Dans toute la suite, le contexte d'un mot est défini par une fenêtre de dix mots : les cinq mots qui le précédent et les cinq mots qui le suivent.
Par exemple, dans la phrase suivante :

La société de parfumerie Marionnaud **détient** désormais 292.157 actions, soit 8,09 % du capital.

Le contexte du mot **détient** est défini par un vecteur dont les composantes sont les occurrences des mots (*actions, capital, de, désormais, du, la, marionnaud, parfumerie, société, soit*). Les chiffres ne sont pas pris en considération, et l'ordre des mots à l'intérieur de la fenêtre n'a pas d'importance.

Si l'on considère un ensemble de textes, il est possible d'additionner tous les vecteurs de contextes trouvés pour un mot donné, et de classer ensuite tous ces contextes par ordre de d'occurrence décroissante. Mais, dans ce cas, les mots qui apparaissent avec la plus grande occurrence sont les mots les plus fréquents du corpus qui n'apportent pas d'information.

Pour définir un contexte utile, on s'appuie sur la méthode de détermination du vocabulaire spécifique présentée au chapitre 5, qui permet d'éliminer automatiquement les mots fréquents et les mots rares.

Si l'on note *TF(m, t)*, la fréquence d'un mot *m* dans un texte *t*, et *CF(m)* la fréquence de ce mot sur l'ensemble du corpus, on calcule, pour chaque mot d'un texte *t*, le rapport :

$$R\,(m,t) = \frac{TF\,(m,t)}{CF\,(m)}$$

Les mots du texte sont classés par ordre décroissant de la valeur de ce rapport, et la deuxième moitié de la liste est supprimée. On obtient, pour chaque texte, une liste *L(t)* de mots parmi lesquels les mots fréquents ont été éliminés.

Pour un mot donné, on ne tient compte d'un contexte que s'il figure dans la liste *L(t)*. Grâce à cette méthode, les mots fréquents ne sont pas pris en considération. En additionnant tous les vecteurs trouvés, et en classant les mots trouvés par ordre décroissant, les mots rares se retrouvent en fin de liste comme lors de la détermination du vocabulaire spécifique.

8.2.2 Exemples de vecteurs de contexte local

Les exemples ci-dessous montrent des contextes trouvés par cette méthode, pour des thèmes issus de corpus différents. Pour chacun de ces thèmes, le contexte de certains mots issus du vocabulaire spécifique est présenté. Le contexte de ces mots est déterminé sur un ensemble de documents pertinents pour le thème étudié, et sur un sous-ensemble de documents non

pertinents, pour mettre en évidence les différences. Par analogie avec des définitions déjà utilisées, le contexte trouvé à partir de l'ensemble des documents pertinents de la base d'apprentissage est appelé **contexte positif,** celui trouvé à partir des documents non pertinents est appelé **contexte négatif**.

La Figure 8.1 montre le contexte obtenu pour certains mots du vocabulaire spécifique du thème *participation*. Les mots dont on étudie le contexte figurent en gras, et sont suivis de leur contexte (s'il existe, c'est-à-dire s'il apparaît dans deux documents différents au moins). La colonne de gauche est le contexte positif, celle de droite présente le contexte négatif.

En comparant chacune des colonnes, on s'aperçoit aisément que le contexte dans lequel apparaissent les mots est souvent très différent selon le sous-ensemble de textes considérés (pertinents ou non pertinents). Dans le premier exemple, la présence du mot *capital* dans l'environnement immédiat du verbe *détient* précise le sens de ce verbe et d'une certaine manière le désambiguïse. On peut noter également la présence d'adverbes comme *désormais* ou *actuellement*, car, pour rendre compte des échanges de participations, ils apparaissent très fréquemment dans des tournures de phrases telles que :

- Après cette opération, la société X *détient désormais* 5 % du capital.
- La société X *détient actuellement* 5% du capital.

Par conséquent, il semble que, sur le corpus de l'AFP que nous utilisons, les adverbes *désormais* et *actuellement* précise le contexte dans lequel est utilisé le verbe détenir.

Contexte positif	**Contexte négatif**
détient	**détient**
capital	titres
actions	portefeuille
désormais	
droits	
participation	
société	
holding	
actuellement	
vote	
parts	
participation	**participation**
prise	sommet
capital	doutes
majoritaire	élève
prendre	éventuelle
minoritaire	président
détient	sérieux
céder	importante
pris	résultat

	sa		faveur
	porter		étrangère
capital		**capital**	
	détient		augmentation
	droits		risque
	augmentation		fonds
	participation		étranger
	actions		propres
	vote		investi
	acquérir		banques
	détenu		action
	représentant		ordinaire
	société		compte

Figure 8.1 : *Contexte spécifique des mots pour le thème participation.*

La Figure 8.2 est obtenue de la même manière pour le thème *partenariat* qui traite des accords de coopération et des partenariats entre entreprise.

Dans ce cas également, les mots issus du vocabulaire spécifique apparaissent dans des contextes différents, et le contexte permet de distinguer un contexte économique d'un contexte de politique de coopération internationale.

Contexte positif		**Contexte négatif**	
partenariat		**partenariat**	
	stratégique		relations
	commercial		signé
	conclu		contrat
	signature		paix
	signé		transatlantique
	accords		
	industriel		
	accord		
	global		
	renforcer		
coopération		**coopération**	
	accord		internationale
	accords		accords
	signé		développement
	domaine		franco
	conclu		étroite
	industrielle		économiques
	commerciale		renforcer
	renforcer		bilatérale
	domaines		matière
	franco		domaines
alliance		**alliance**	
	stratégique		agricole
	commerciale		atlantique
	annoncée		propos
	groupes		and
	compagnies		militaire
	possibilité		éditrice
	capitalistique		fusion
	accord		opposition
	vue		tourisme
	conclure		recours

Figure 8.2 : *Contextes spécifiques des mots pour le thème partenariat.*

La Figure 8.3 présente les résultats obtenus avec le thème *interest* du corpus Reuters. Le contexte du mot *rate* permet, par exemple, de faire la différence lorsque ce mot apparaît pour parler des taux de changes avec les expressions *floating rate* ou *exchange rate,* contrairement au cas où il apparaît dans un contexte de taux d'intérêt avec l'expression *prime rate.*

Enfin, la Figure 8.4 présente le contexte spécifique obtenu pour le thème 351 de TREC-8 (*Falkland petroleum exploration*) pour *islands* et *oil.* L'étude de ce contexte montre bien que *oil* ne doit être un indicateur que s'il est suivi de notions indiquant que l'on parle des îles malouines (*islands, malvinas*, ou *argentina*).

	Contexte positif		Contexte négatif
money		**money**	
	stg		supply
	rates		m
	k		growth
	market		stg
	call		k
	given		england
	further		liquidity
	supply		broad
	assistance		assistance
	rate		given
rates		**rates**	
	base		interest
	interest		levels
	lending		freight
	cut		current
	money		interbank
	point		stability
	k		exchange
	rates		points
	short		lower
	term		short
rate		**rate**	
	prime		floating
	cut		fixed
	base		dollar
	effective		rate
	lending		exchange
	discount		yen
	maturity		growth
	funds		discount
	point		variable
	interest		inflation

Figure 8.3 : *Contextes spécifiques des mots pour le thème interest du corpus Reuters.*

	Contexte positif		Contexte négatif
islands		**islands**	
	malvinas		highlands
	falkland		enterprise
	oil		channel
	sovereignty		china
	argentina		spratly
	argentine		cayman4
	aires		madeira
	buenos		shetland
	islands		claim
	tella		development
oil		**oil**	
	islands		gas
	malvinas		crude
	argentina		production
	oil		barrels
	exploration		exploration
	exploitation		bn
	existence		fields
	tella		africa
	ypf		saudi
	argentine		m

Figure 8.4 : *Contextes spécifiques des mots pour le thème 351 du corpus TREC-8 (Falkland petroleum exploration).*

8.3 Modèle neuronal avec contexte

8.3.1 Architecture prenant en considération le contexte

Nous venons de voir que le contexte devait, en fait, servir à renforcer ou à diminuer l'influence d'un mot. Les entrées de la régression logistique ne sont plus de simples descripteurs comme dans le cas de la représentation en sacs de mots, mais les sorties de neurones dont les entrées sont représentées sur la Figure 8.5. Le *mot principal* est issu de la méthode de sélection de descripteurs exposée au chapitre 7, et le contexte est déterminé par la méthode exposée dans le paragraphe précédent.

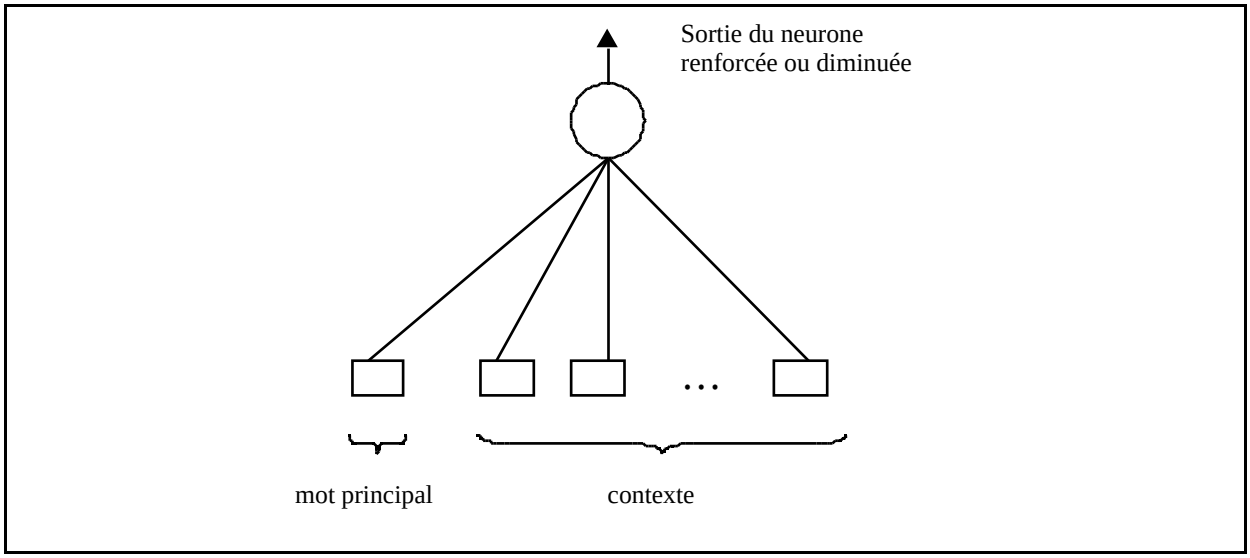


Figure 8.5 : *Unité de base de la nouvelle architecture neuronale (le biais n'est pas dessiné).*

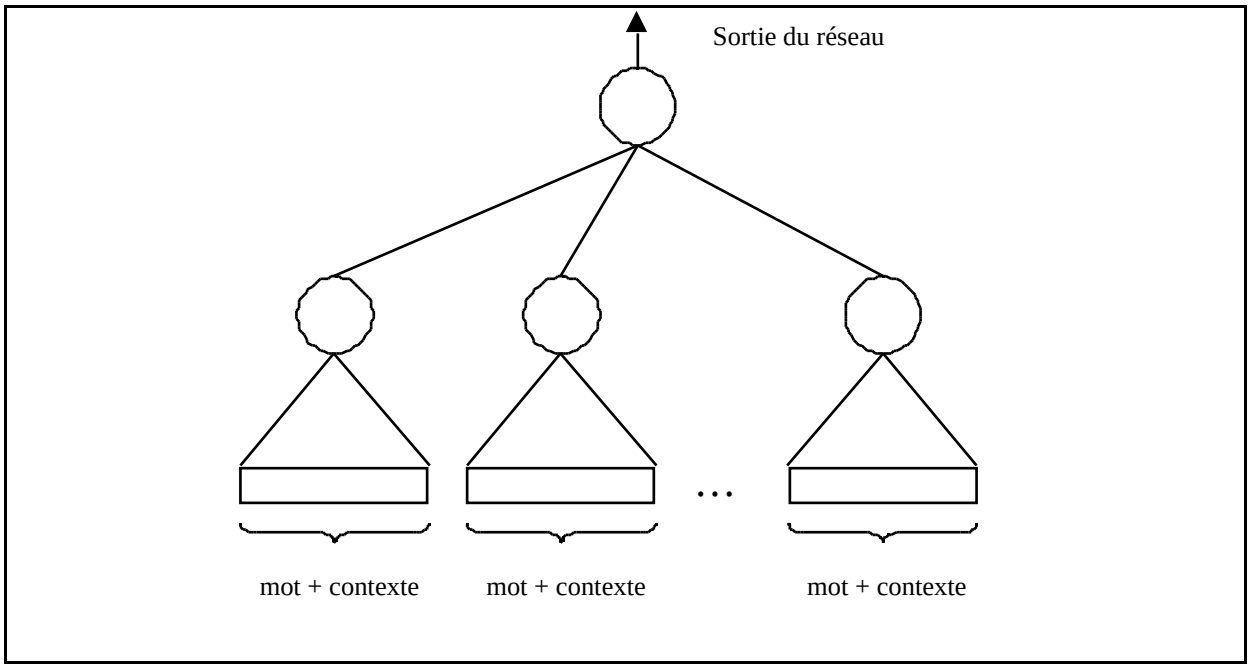


Figure 8.6 : *Architecture neuronale pour tenir compte du contexte.*

Le réseau de neurones complet est décrit par la Figure 8.6 : il prend en considération le contexte de chaque mot, et ce qui était considéré précédemment comme un descripteur d'entrée est un neurone caché dans cette nouvelle architecture.

Cette architecture est proche de l'architecture utilisée précédemment, la première couche de neurones permettant de préciser l'influence d'un descripteur particulier en fonction de son contexte.

8.3.2 Combien de poids dans la nouvelle architecture ?

Cette architecture fait intervenir plus de paramètres ajustables que la régression logistique du chapitre 7. Si, par exemple, le nombre de descripteurs sélectionnés était de 30 alors le modèle comprenait 31 paramètres (avec le biais du neurone de sortie). Si, chaque descripteur précédent est précisé par cinq contextes possibles, alors le réseau de neurones décrit à la Figure 8.5 contient 30 neurones cachés, chacun étant lié à sept entrées (le mot principal, les cinq contextes et le biais) ; le nombre de poids à déterminer lors de l'apprentissage est :

$$7*30+30+1 = 241$$

Le nombre de poids augmente considérablement par rapport à l'architecture précédente, mais la taille des bases d'apprentissage n'a évidemment pas augmenté : il faut vérifier qu'il est effectivement possible de mener l'apprentissage correctement.

8.3.3 La régularisation mise en œuvre par la méthode du weight decay

Les expériences des chapitres précédents ont montré que l'utilisation d'un terme de *weight decay* était indispensable pour obtenir de bonnes performances. Cette remarque est encore plus vraie avec cette nouvelle architecture, en raison de l'augmentation du nombre des poids.

L'architecture de la Figure 8.6 est une architecture de perceptron multi-couche partiellement connectée : il est donc nécessaire d'utiliser trois hyperparamètres α_1, α_2 et α_3 pour le terme de *weight decay* et de minimiser la fonction de coût suivante :

$$EC(w) + \frac{\alpha_0}{2} \sum w_i^2 + \frac{\alpha_1}{2} \sum w_i^2 + \frac{\alpha_2}{2} \sum w_i^2 + \frac{\alpha_3}{2} \sum w_i^2$$

où $EC(w)$ est l'entropie croisée, W_0 représente l'ensemble des poids des connexions reliant les biais aux neurones cachés, W_1 représente l'ensemble des poids des connexions reliant les entrées aux neurones cachés et W_3 représente l'ensemble des poids liés des connexions du neurone de sortie y compris le biais du neurone de sortie.

8.4 Détermination des valeurs des hyperparamètres

Afin de tester l'influence des valeurs des hyperparamètres sur l'apprentissage de l'architecture neuronale présentée sur la Figure 8.6, trois approches différentes sont mises en œuvre :

- la première consiste à fixer les valeurs *a priori* et à conserver ces valeurs fixes pendant l'apprentissage,
- la deuxième consiste à faire varier les hyperparamètres pendant l'apprentissage par la méthode de maximisation issue du formalisme de l'approche bayesienne décrit au chapitre 6**,**
- la dernière consiste à faire varier les hyperparamètres selon la méthode d'intégration issue du formalise de l'approche bayesienne décrit au chapitre 6.

Ces expériences sont effectuées sur les 90 catégories du corpus Reuters ; pour chaque catégorie, le mot principal de l'architecture de la Figure 8.5 correspond aux résultats de la sélection de descripteurs du chapitre 7, et pour chacun de ces mots, les cinq premiers contextes positifs sont ajoutés.

Trois hyperparamètres sont utilisés selon la répartition décrite au paragraphe 8.3.3.

8.4.1 Hyperparamètres constants

Pour cette expérience, les trois hyperparamètres sont constants durant tout l'apprentissage ; on cherche à étudier l'impact du choix des valeurs sur les résultats. Plusieurs expériences ont montré que la valeur de l'hyperparamètre α_0 concernant le biais des neurones cachés avait peu d'importance : cet hyperparamètre est fixé à 0,001 pour toutes les expériences suivantes.

Les deux hyperparamètres α_1 et α_2 varient dans l'intervalle [0 ; 6] par pas de 0,25. Pour chaque valeur du couple (α_1, α_2), l'apprentissage est effectué, et la précision moyenne non interpolée (UAP) ainsi que la valeur de *F* optimale sont calculées sur la base de test. Le couple (0, 0) correspond à un apprentissage effectué sans régularisation (excepté pour le biais des neurones cachés).

Les macro-moyennes sont calculées par sous-groupe de catégories afin de relier les variations aux nombres d'exemples pertinents présents dans la base d'apprentissage.

Les résultats sont présentés à la Figure 8.7 pour chaque sous-groupe. La colonne de gauche montre l'évolution de la précision moyenne non interpolée et la colonne de droite l'évolution de la valeur de *F*. L'axe des X est la valeur de l'hyperparamètre α_1 (connexions reliant les entrées aux neurones cachés) et l'axe des Y la valeur de l'hyperparamètre α_2 (connexions du neurone de sortie). L'échelle selon l'axe Z varie selon le groupe de catégories considéré pour mettre en évidence les variations.

Quel que soit le sous-groupe de catégories considéré, l'ensemble des résultats prouve qu'il est indispensable d'utiliser une méthode de régularisation lors de l'apprentissage, car, lorsque l'un des hyperparamètres a une valeur nulle, les performances sont nettement détériorées. Pour la précision moyenne non interpolée, les performances obtenues avec le couple (0, 0) sont systématiquement les plus mauvaises.

Pour l'ensemble des catégories, la précision moyenne non interpolée est peu affectée par la valeur des hyperparamètres à partir du moment où ils ne sont pas nuls. Pour les soixante premières catégories, elle diminue légèrement lorsque les deux hyperparamètres prennent des valeurs supérieures à deux. Pour les trente dernières catégories (catégories 61 à 90), le comportement est légèrement différent de celui observé pour les autres sous-groupes, car les performances ne décroissent pas lorsque les hyperparamètres ont des valeurs élevées.

L'évolution de la valeur de *F* avec les hyperparamètres montre que les performances ont tendance à décroître lorsque les valeurs des deux hyperparamètres augmentent ; le phénomène est d'autant plus prononcé que le nombre de documents pertinents diminue sur la base d'apprentissage.

Pour les catégories ayant peu de documents pertinents, la valeur de *F* diminue significativement lorsque les valeurs des deux hyperparamètres augmentent, alors que la précision moyenne non interpolée reste à un niveau comparable. Ce comportement différent des deux mesures a déjà été rencontré au chapitre 6 : la sortie du réseau de neurones prend des valeurs très faibles, et, si le classement est toujours possible, l'utilisation d'un seuil ne permet plus de séparer les documents.

Pour des valeurs égales des hyperparamètres, plus la taille de la base d'apprentissage est faible, moins le terme d'entropie croisée a d'importance dans la fonction de coût total, et plus les termes de *weight decay* sont importants. Par conséquent, les poids tendent rapidement vers zéro lors de l'apprentissage. Néanmoins, l'apprentissage s'effectue toujours, et les documents pertinents ont globalement une probabilité de pertinence plus élevée que les documents non pertinents, mais du fait de la faible valeur des poids, ces probabilités sont proches de zéro.

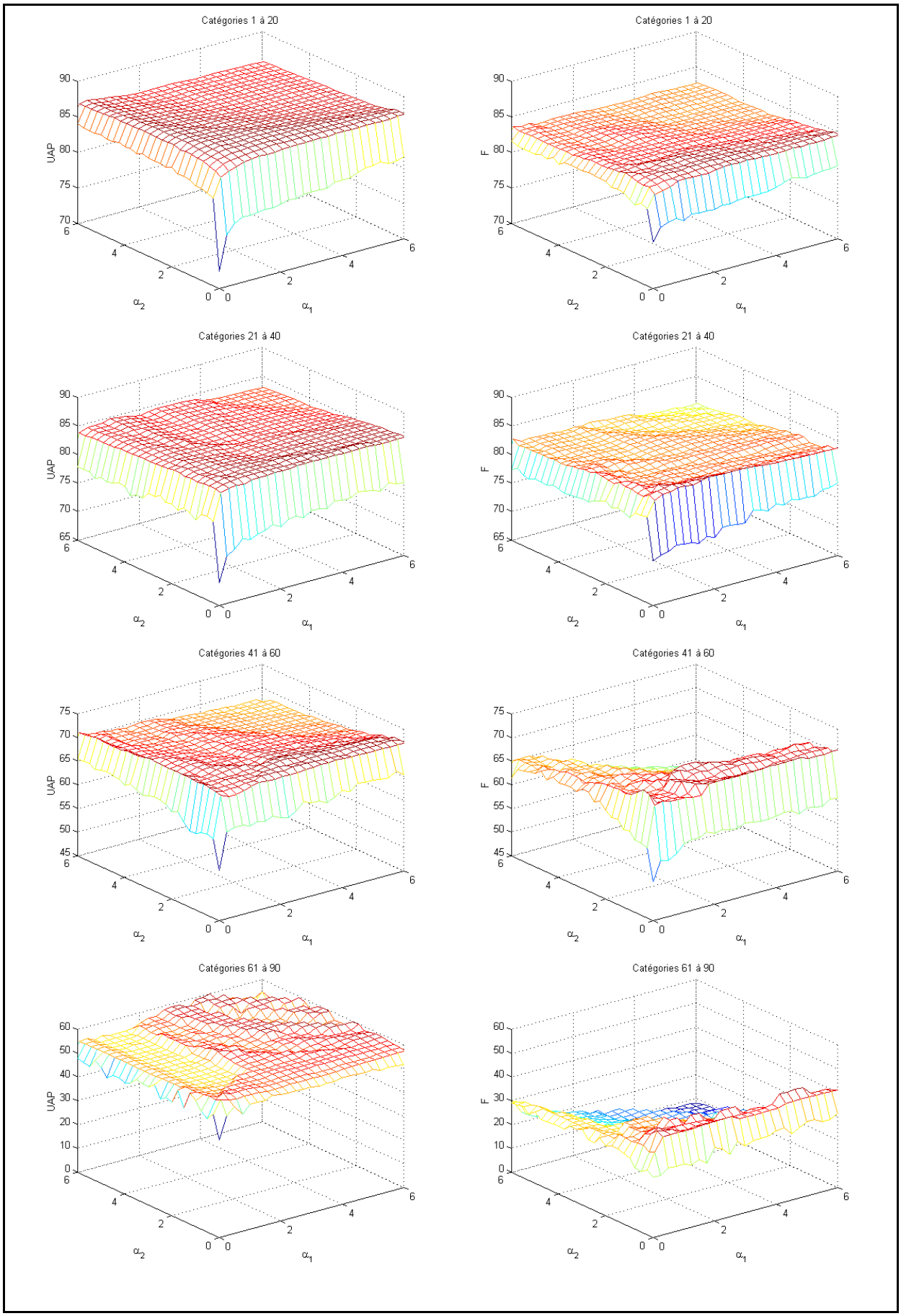

Catégories 1 à 20
UAP
Catégories 1 à 20
F
Catégories 21 à 40
UAP
Catégories 21 à 40
F
Catégories 41 à 60
UAP
Catégories 41 à 60
F
Catégories 61 à 90
UAP
Catégories 61 à 90
F
α1
α2

Figure 8.7 : *Évolution des performances (F et UAP)selon la valeur des hyperparamètres.*

En conclusion, les expériences montrent qu'il est possible de choisir des constantes pour les valeurs des hyperparamètres, car les performances ne sont pas extrêmement sensibles à ces valeurs à partir du moment où elles ne sont pas trop élevées. Pour l'hyperparamètre lié aux connexions du neurone de sortie, une valeur faible semble préférable alors que, pour l'hyperparamètre lié aux connexions entre les entrées et les neurones cachés, il est possible de choisir des valeurs légèrement supérieures.

Pour les comparaisons ultérieures, la Figure 8.8 récapitule les résultats obtenus sur l'ensemble des thèmes avec $\alpha_0 = 0{,}001$, $\alpha_1 = 1{,}0$ et $\alpha_2 = 0{,}5$.

	UAP	*F*
Moyenne sur l'ensemble des thèmes	72,28	65,6
Moyenne pour les thèmes 1 à 10	90,30	86,27
Moyenne pour les thèmes 11 à 40	84,72	83,32
Moyenne pour les thèmes 41 à 60	70,45	67,99
Moyenne pour les thèmes 61 à 90	55,52	39,76

Figure 8.8 : *Résultats obtenus avec $\alpha_0 = 0{,}001$, $\alpha_1 = 1{,}0$ et $\alpha_2 = 0{,}5$.*

8.4.2 Hyperparamètres optimisés : méthode d'intégration

Dans le paragraphe précédent, les hyperparamètres étaient fixes durant l'apprentissage. Pour l'expérience décrite ici, les hyperparamètres évoluent durant l'apprentissage, et sont déterminés par la méthode d'intégration issue de l'approche bayesienne expliquée au chapitre 6.

Dans cette approche, chaque hyperparamètre est estimé régulièrement pendant l'apprentissage selon la formule :

$$\alpha_k = \frac{\gamma}{\cdot}$$

p_k est le nombre de poids concernés par l'hyperparamètre α_k.

Pour cette expérience α_0 est fixé à 0,001 comme précédemment et n'évolue pas pendant l'apprentissage. La valeur initiale de α_1 est 0,5 et la valeur initiale de α_2 est 1,0 ; ces deux hyperparamètres sont calculés régulièrement pendant l'apprentissage.

Les résultats obtenus sont présentés à la Figure 8.9 et doivent être comparés à ceux obtenus à la Figure 8.8.

	UAP	*F*
Moyenne sur l'ensemble des thèmes	42,85	31,94
Moyenne pour les thèmes 1 à 10	89,15	84,13
Moyenne pour les thèmes 11 à 40	75,08	66,66
Moyenne pour les thèmes 41 à 60	20,35	2,78
Moyenne pour les thèmes 61 à 90	10,97	0,14

Figure 8.9 : *Résultats obtenus avec les hyperparamètres déterminés par la méthode d'intégration.*

Les résultats sont nettement inférieurs à ceux obtenus précédemment, et l'écart est d'autant plus grand que le nombre de documents pertinents sur la base d'apprentissage est faible.

En fait, avec cette approche, l'hyperparamètre α_1 tend systématiquement vers des valeurs très élevées (de l'ordre de la centaine), et par conséquent les poids liés aux entrées tendent très rapidement vers des valeurs faibles.

Les résultats dépendent de l'initialisation des hyperparamètres, mais, quelles que soient les valeurs essayées, le comportement reste identique et les performances décevantes.

8.4.3 Hyperparamètres optimisés : méthode de maximisation

La méthode de maximisation des hyperparamètres issue de l'approche bayesienne a également été testée sur ce problème. La théorie de cette approche a été détaillée au chapitre 6.

Rappelons les résultats :

$$\alpha_k = \frac{\quad}{\quad}$$

avec

$$\left(\eta_j - \left(\quad \right) \right)$$

où $\{\eta_j\}$ représente l'ensemble des valeurs propres du hessien A de la fonction de coût régularisée, V est la matrice des vecteurs propres, et I_k est la matrice ne contenant que des valeurs nulles sauf sur les éléments de la diagonale liés au groupe de poids gouvernés par l'hyperparamètre α_k où la valeur est 1.

Le paramètre γ_k peut également être calculé par la formule suivante, équivalente à la précédente :

$$\gamma_k = k - \alpha_k \mathrm{Tr}_k\left[A^{-1}\right]$$

où $\mathrm{Tr}_k\left[A^{-1}\right]$ est la trace ne portant que sur les éléments gouvernés par l'hyperparamètre α_k.

Par rapport à la méthode de maximisation, le paramètre p_k est remplacé par le paramètre γ_k dont la valeur est inférieure ou égale à p_k : on peut donc espérer corriger le défaut de la méthode précédente, qui conduisait à une valeur de l'hyperparamètre trop élevée et faisait tendre les poids vers zéro.

En pratique, comme la fonction de coût régularisée s'écrit :

$$I(w) = E(w) + \alpha \cdot E^0(w) \qquad (\quad) \qquad (\quad)$$

$E_c(w)$ est le terme d'entropie croisée et E^k = .

Le hessien A de la fonction de coût régularisée se calcule grâce à la formule :

$$A = \quad I(w) =$$

H est la matrice du hessien de la fonction d'entropie croisée dont le calcul exact est effectué grâce à l'algorithme développé par [Bishop, 1992]. Si p est le nombre de paramètres du réseau, le calcul de la matrice H de dimension (p, p) nécessite un nombre d'étapes qui varie comme p^2.

Le calcul du paramètre γ_k peut se faire, soit en inversant la matrice A par une méthode proposée dans [Press *et al.*, 1992], soit en la diagonalisant par une méthode qui permet de calculer les valeurs propres et les vecteurs propres.

La mise en œuvre de cette méthode nécessite de fixer des valeurs initiales pour les hyperparamètres, puis de commencer la minimisation de la fonction de coût. Après un certain nombre d'itérations, la fonction de coût est proche d'un minimum et les hyperparamètres peuvent être calculés à nouveau.

Cependant le minimum trouvé est un minimum de la fonction de coût régularisée et n'est donc pas nécessairement un minimum de la matrice H. La matrice H n'est donc pas nécessairement définie positive au point où l'on calcule son hessien et ses valeurs propres peuvent être négatives. Or, si les valeurs propres sont négatives, il est possible d'obtenir une valeur γ_k négative et donc une valeur négative pour l'hyperparamètre !

Pour remédier à ce problème, il est recommandé dans la FAQ sur les réseaux bayesiens[2] de ne pas prendre, dans la détermination de γ_k, la contribution de ces valeurs.

Lorsqu'on utilise la formule faisant intervenir l'inverse du hessien, il est nécessaire de vérifier les inégalités :

$$0 < \gamma_k \leq p_k$$

Après avoir choisi des valeurs initiales, il faut minimiser partiellement la fonction de coût régularisée, puis estimer les valeurs des hyperparamètres grâce à l'une des formules. L'estimation des hyperparamètres ne doit pas se faire avant une minimisation conséquente de la fonction de coût régularisée, car les approximations n'ont pas de sens loin du minimum. Après avoir modifié les valeurs des hyperparamètres, la surface de la fonction de coût a été modifiée et il est nécessaire de recommencer une nouvelle minimisation partielle.

L'apprentissage se termine après convergence de cet algorithme. En théorie, la convergence de cet algorithme n'a pas été montrée : il est nécessaire de la vérifier expérimentalement.

[2] http://wol.ra.phy.cam.ac.uk/mackay/Bayes_FAQ.html

Dans notre cas, il est nécessaire de ne pas faire varier l'hyperparamètre lié aux biais des neurones cachés, car il diverge systématiquement. Dans ce cas, les biais associés tendent vers zéro très rapidement, mais le produit $\alpha_0 E_w^0[w]$ diverge.

La Figure 8.10 montre l'évolution des deux hyperparamètres α_1 et α_2 pendant l'apprentissage pour les trois thèmes *dlr*, *nat-gas* et *ipi*, chaque point d'une courbe correspond à un nouveau calcul des hyperparamètres. Pour ces trois thèmes, l'hyperparamètre α_2 converge rapidement vers une valeur proche de 0, mais l'hyperparamètre α_1 a tendance à diverger.

Avant chaque calcul, on calcule le conditionnement de la matrice hessienne *A* grâce à une décomposition en valeurs singulières [Press *et al.*, 1992]. L'évolution du logarithme de cette valeur au fil de l'apprentissage est tracée à la Figure 8.11 pour chacun des trois thèmes. Ces courbes montrent que la matrice hessienne est de plus en plus mal conditionnée au fur et à mesure que les hyperparamètres sont calculés, et que, par conséquent, le calcul de l'inverse de cette matrice est de plus en plus instable numériquement : les calculs ne peuvent plus être menés à bien.

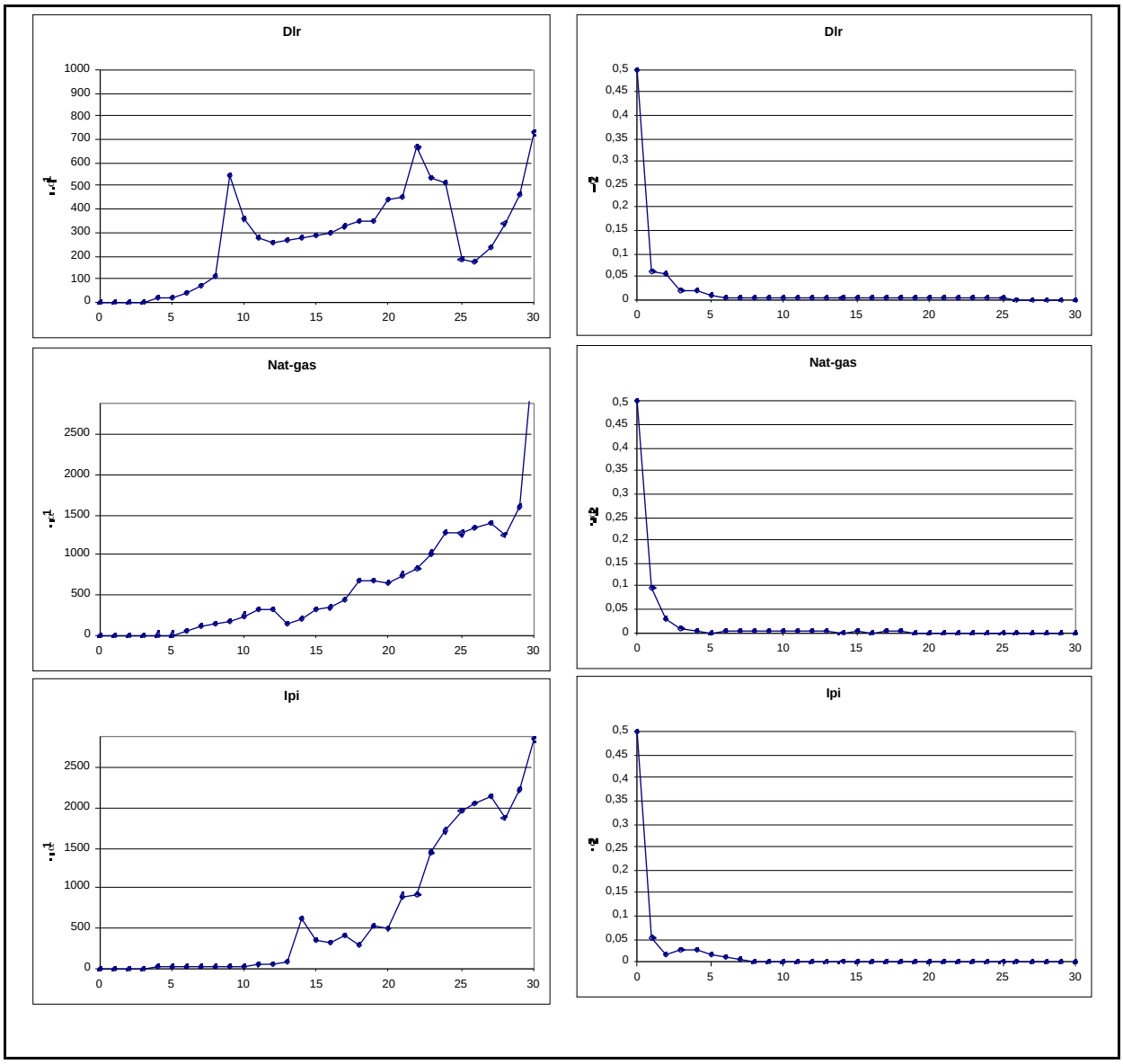


Figure 8.10 : *Évolution des hyperparamètres pendant l'apprentissage.*

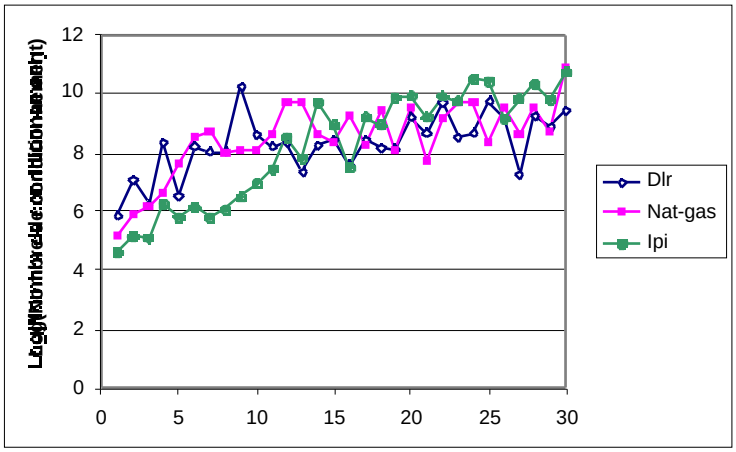


Figure 8.11 : *Évolution du logarithme du nombre de conditionnement de la matrice hessienne A au fil de l'apprentissage.*

Finalement, la méthode de maximisation n'a pas pu être mise en œuvre avec nos modèles, car l'un des hyperparamètres diverge, et rend la matrice du hessien mal conditionnée.
Les formules de calculs des hyperparamètres résultent de plusieurs approximations dont la principale est l'approximation gaussienne de la probabilité *a posteriori* des poids. On peut donc supposer, dans ce cas, que cette approximation n'est pas justifiée.

8.4.4 Conclusion sur les hyperparamètres

Les différentes expériences menées sur le corpus Reuters montrent que les méthodes d'intégration et de maximisation issues de l'approche bayesienne ne conduisent pas à des valeurs correctes pour les hyperparamètres. Il est probable que les approximations nécessaires aux calculs ne soit pas justifiées.

Cependant les résultats du paragraphe 8.4.1 montrent qu'il est possible de choisir *a priori* des valeurs constantes, puisque, d'une part, les résultats ne sont pas, en moyenne, très sensibles à ces valeurs, et, d'autre part, il est possible de retenir des valeurs correctes pour l'ensemble des thèmes. Dans la suite, les paramètres suivants sont utilisés :

$$\alpha_0 = 0{,}001$$

$$\alpha_1 = 1{,}0$$

$$\alpha_2 = 0{,}5$$

Figure 8.12 : *Valeurs des hyperparamètres retenues.*

Il est toujours possible, lorsque les données sont suffisamment nombreuses, de faire de la validation croisée en faisant varier, par exemple, le couple (α_1, α_2). Néanmoins, ces méthodes sont longues à mettre en œuvre, car il est nécessaire d'effectuer un grand nombre d'apprentissages.

8.5 Résultats sur le corpus Reuters

8.5.1 Présentation des expériences réalisées

Plusieurs expériences sont effectuées sur l'ensemble du corpus Reuters afin de mettre en œuvre le modèle proposé, et mesurer l'amélioration apportée par rapport au séparateur linéaire du chapitre 7.

Pour toutes ces expériences, les mots principaux définis à la Figure 8.5 sont les descripteurs sélectionnés lors de la définition du séparateur linéaire du chapitre 7.

Une première série d'expériences est effectuée où, pour chacun de ces descripteurs, les cinq premiers contextes positifs déterminés par la méthode exposée au paragraphe 8.2.1 sont pris en considération.

Pour mesurer l'apport éventuel de l'ajout du contexte négatif, une deuxième série d'expériences est effectuée, où le contexte est défini par les cinq premiers contextes positifs et les cinq premiers contextes négatifs, à condition qu'ils apparaissent au moins dix fois sur la base d'apprentissage.

Par exemple, pour le thème *interest,* le descripteur *rate* avait été sélectionné par la méthode de sélection de descripteurs. Lorsque les cinq premiers contextes positifs sont pris en considération, il peut être désambiguïsé par l'ensemble de descripteurs (*prime*, *cut, base, effective, lending*). Lorsque les contextes négatifs sont également pris en considération, il peut être désambiguïsé par l'ensemble de descripteurs (*prime, cut, base*, *effective, lending, floating, fixed, dollar, rate, exchange*). L'architecture de base de la Figure 8.5 comprend, avec le biais, sept paramètres dans le premier cas et douze dans le second.

Les expériences du chapitre 7 ont montré que la substitution des mots par leur racine lexicale ou par leur lemme augmentait l'ambiguïté des descripteurs et finalement avait un impact négatif sur le modèle linéaire. De nouvelles expériences sont conduites afin de voir si l'introduction du contexte permet de pallier cet inconvénient.

Tous les apprentissages sont effectués avec les valeurs des hyperparamètres indiquées au paragraphe 8.4.4.

8.5.2 Performances du modèle avec contexte

Les performances obtenues sur les 90 thèmes du corpus Reuters sont présentées à la Figure 8.13 ; la première ligne correspond à l'utilisation du contexte positif et la deuxième ligne à l'utilisation des contextes positif et négatif. Les performances sont mesurées grâce à la précision moyenne non interpolée (UAP), la précision moyenne sur 11 points et la mesure de *F* optimisée.

Les résultats sont présentés par sous-ensemble de thèmes pour faire apparaître des corrélations éventuelles entre le nombre de documents pertinents de la base d'apprentissage ; les résultats sont des macro-moyennes sur ces sous-ensembles.

	Nombre de poids du réseau	UAP	*F*	11-pt
Moyenne sur l'ensemble des thèmes	124,76 136,69	72,28 72,89	65,61 66,07	72,44 73,12
Moyenne pour les thèmes 1 à 10	202,60 223,70	90,30 90,29	86,27 86,48	87,16 87,12
Moyenne pour les thèmes 11 à 40	136,16 151,81	84,73 85,01	83,32 83,23	85,18 85,39
Moyenne pour les thèmes 41 à 60	139,95 150,05	70,45 72,26	68,00 70,07	71,48 73,46
Moyenne pour les thèmes 61 à 90	77,93 84,53	55,52 55,85	39,76 39,76	55,87 56,39

Figure 8.13 : *Ensemble des résultats sur le corpus Reuters. La première ligne correspond à l'utilisation du contexte positif et la deuxième ligne à l'utilisation des deux contextes.*

La prise en compte du contexte négatif n'améliore pas les performances alors qu'elle nécessite plus de paramètres.

La seule différence significative dans les moyennes se trouve pour les thèmes 41 à 60 et s'explique en grande partie par le thème *retail* (thème 41) pour lequel la précision moyenne non interpolée évolue de 35,0 à 62,5 grâce à l'ajout du contexte négatif. Cependant, ce thème ne contient que deux documents pertinents sur la base de test : les mesures sont donc très sensibles au moindre changement. La Figure 8.14 montre le classement des dix premiers textes de la base de test obtenu avec chacune des méthodes ; les deux textes pertinents apparaissent

en gras. Le classement des deux méthodes est quasiment identique et il n'est pas possible de tirer des conclusions à partir de cet exemple.

15546	0.777016	**15742**	**0.656229**
15742	**0.636757**	15546	0.589069
17723	0.361686	17723	0.330273
15619	0.298192	15619	0.323393
16853	0.0962382	16853	0.0880126
19625	0.0550316	19625	0.0616612
15033	0.0374317	16783	0.0400094
19483	0.0303551	**16118**	**0.0383093**
16843	0.0292854	15033	0.0380672
16118	**0.0286407**	16843	0.02986

Figure 8.14 : *Liste des dix premiers textes classés pour le thème retail. Colonne de gauche avec le contexte positif, colonne de droite avec les deux contextes.*

La Figure 8.15 présente, pour les soixante premiers thèmes, les comparaisons thème à thème : chaque point représente un thème dont l'abscisse est la précision moyenne non interpolée obtenue avec l'utilisation du contexte positif et l'ordonnée est celle obtenue avec l'utilisation des contextes positif et négatif.

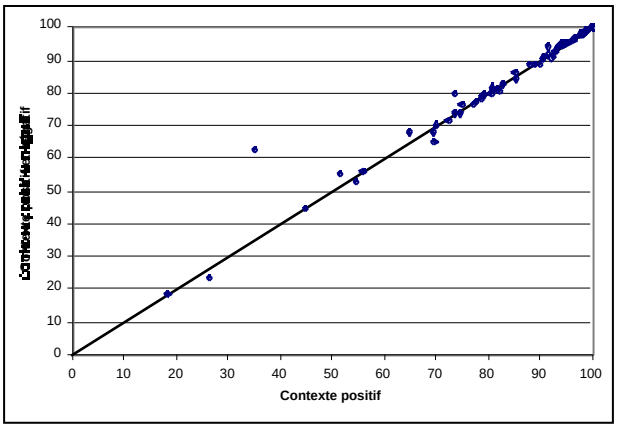


Figure 8.15 : *Comparaison thème à thème pour les soixante premiers de la précision moyenne non interpolée : contexte positif ou contexte positif et négatif.*

Le seul point éloigné de la diagonale correspond au thème *retail* étudié plus haut ; pour tous les autres thèmes, les points sont très groupés autour de la diagonale : pour la majorité des thèmes, les performances sont quasiment inchangées.

En conclusion, l'ajout du contexte négatif augmente le nombre de poids du réseau de neurones, mais l'amélioration des résultats n'est pas significative.

8.5.3 Comparaison avec le séparateur linéaire

Les résultats obtenus avec l'utilisation du contexte positif sont comparés avec les résultats obtenus par le séparateur linéaire présenté chapitre 7. La Figure 8.16 reprend les résultats déjà obtenus par chacune des approches : la première ligne correspond au modèle avec cinq contextes positifs, et la deuxième ligne correspond au séparateur linéaire du chapitre 7.

	UAP	*F*	11-pt
Moyenne sur l'ensemble des thèmes	72,28 72,59	65,61 66,45	72,44 72,83
Moyenne pour les thèmes 1 à 10	90,30 89,68	86,27 85,35	87,16 87,21
Moyenne pour les thèmes 11 à 40	84,73 83,19	83,32 81,37	85,18 83,58
Moyenne pour les thèmes 41 à 60	70,45 69,20	68,00 69,15	71,48 69,99
Moyenne pour les thèmes 61 à 90	55,52 59,03	39,76 43,79	55,87 59,64

Figure 8.16 : *Résultats sur l'ensemble du corpus Reuters. La première ligne correspond au modèle avec contexte, la deuxième ligne au séparateur linéaire.*

Sur l'ensemble des thèmes, les résultats sont très proches, mais globalement, sur les thèmes comprenant plus de dix documents pertinents sur la base d'apprentissage (les soixante premiers), le modèle avec contexte conduit à des résultats supérieurs, tandis que sur les trente derniers, le modèle linéaire est plus performant.

Cependant, les résultats précédents sont des moyennes et peuvent cacher des différences ; pour préciser les résultats, la Figure 8.17 présente, pour les soixante premiers thèmes, la précision moyenne non interpolée obtenue avec chaque méthode.

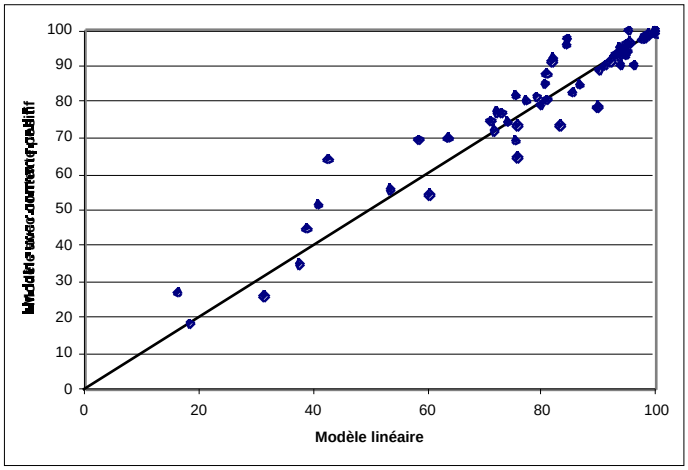


Figure 8.17 : *Comparaison de la précision moyenne non interpolée pour les thèmes 1 à 60 : modèle linéaire ou utilisation du contexte.*

Il est intéressant de noter que, en général, le modèle avec contexte améliore le score des thèmes qui ont une performance plus faible que la moyenne avec le modèle linéaire. Pour les thèmes qui atteignent une précision moyenne non interpolée supérieure à 90 avec le modèle linéaire, l'utilisation du contexte n'apporte pas d'amélioration.

La Figure 8.18 présente, pour chaque modèle, les courbes rappel-précision interpolée pour les soixante premiers thèmes et pour les trente derniers.

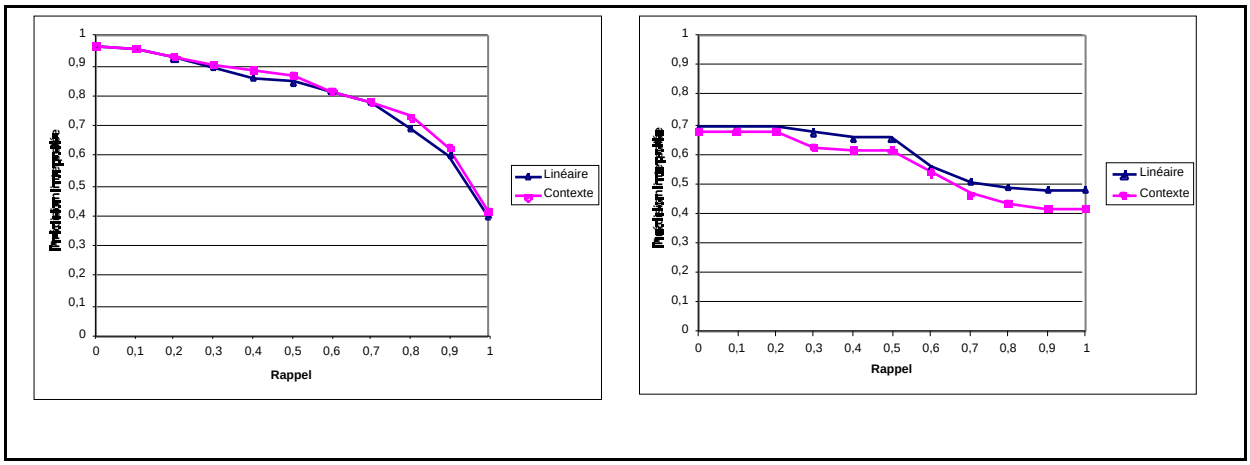


Figure 8.18 : *Courbes rappel-précision interpolée, pour les soixante premiers thèmes (à gauche), et pour les trente derniers (à droite).*

La Figure 8.19 présente les courbes rappel-précision interpolée, pour les trente-six thèmes parmi les soixante premiers pour lesquels la précision moyenne non interpolée est inférieure à

90. Pour l'ensemble de ces thèmes, l'amélioration apportée par l'utilisation du contexte apparaît plus nettement.

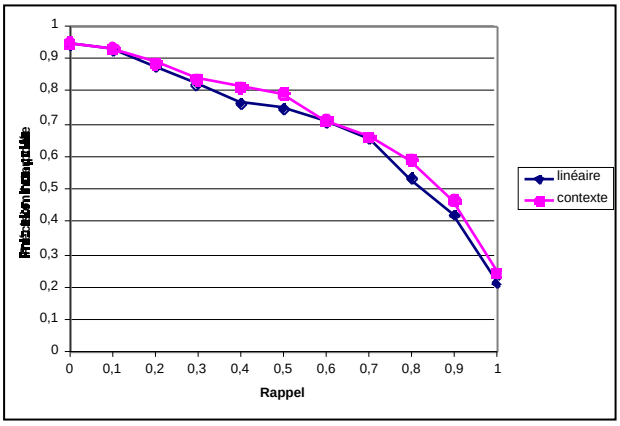


Figure 8.19 : *Courbes rappel-précision interpolée pour les trente-six thèmes dont la précision moyenne non interpolée est inférieure à 90 avec le modèle linéaire.*

8.5.3.1 Exemples de thèmes dont les performances augmentent

Ce paragraphe a pour but d'illustrer les différences de comportements entre les deux méthodes grâce à l'étude de quelques exemples. Dans la discussion, la valeur de la sortie d'un classifieur pour un texte donné est appelée *score*.

Pour le thème *interest* qui contient 347 documents pertinents sur la base d'apprentissage et 131 sur la base de test, les performances obtenues par chacune des méthodes sont présentées à la Figure 8.20 :

	UAP	*F*	11-pt
Modèle linéaire	75,33	71,49	73,49
Modèle avec contexte	81,98	76,68	79,74

Figure 8.20 : *Performance pour le thème interest en fonction du modèle.*

L'observation des sorties de chaque modèle montre qu'avec l'utilisation du contexte, les scores des textes pertinents ont tendance à être plus élevés grâce à l'utilisation d'expression comme *interest rates* qui sont reconnus par le modèle avec contexte. Cette tendance se retrouve sur les courbes rappel-précision de la Figure 8.21, qui montrent que la précision est plus élevée avec le modèle utilisant le contexte.

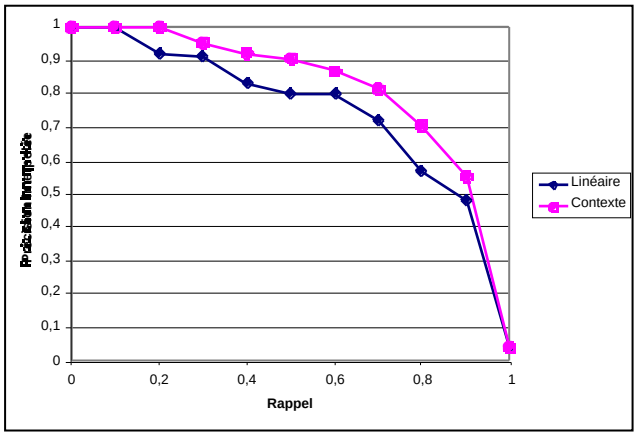


Figure 8.21 : *Courbes rappel-précision interpolée du thème interest pour chacun des modèles.*

Par exemple, le score du texte de la Figure 8.22 passe de 0,40 avec le modèle linéaire à 0,60 avec le modèle utilisant le contexte ; dans ce deuxième cas, aucun document non pertinent n'est classé avant ce texte.

```
AQUINO SAYS MANILA WATCHING INTEREST RATES CLOSELY
President Corazon Aquino said the
Philippines was closely monitoring interest rates in the wake
of Monday's record drop on Wall Street and steep declines in
Manila and other Asian stock markets.
(…)
REUTER
```

Figure 8.22 : *Extrait du texte 20021 pertinent pour le thème interest.*

Pour le thème *cpi* (thème 23, *Consumer Price Index*) qui contient 69 documents pertinents sur la base d'apprentissage et 28 sur la base de test, les performances avec chacune des méthodes sont présentées à la Figure 8.23 et montrent la supériorité du modèle avec contexte :

	UAP	*F*	11-pt
Modèle linéaire	58,38	53,73	58,75
Modèle avec contexte	69,55	61,54	68,17

Figure 8.23 : *Performance pour le thème cpi en fonction du modèle.*

Le texte 16740 (Figure 8.24), qui est pertinent pour le thème est à la 16ème place avec le modèle linéaire (son score est 0,48) et se trouve au 6ème rang avec le modèle qui utilise le contexte (son score est 0,87). Pour le modèle avec contexte, le mot-clef *consumer* est précisé par *index* lui-même précisé par le mot *price* et *inflation* est précisé par *rate* et *pct.*

NEW ZEALAND CPI RISES 2.3 PCT IN MARCH QUARTER
New Zealand's **consumer** price **index**,
CPI, which measures the rate of **inflation**, rose 2.3 pct in the
quarter ended March 31 against an 8.9 pct rise in the December
1986 quarter and a 2.3 pct rise in the March 1986 quarter, the
Statistics Department said.
(…)
Nearly half the increase in the latest quarterly **index** was
contributed by the housing group, the department said.

Figure 8.24 : *Extrait du texte 16740 pertinent pour le thème cpi. Les mots en gras sont les mots sélectionnés par la méthode de sélection de descripteurs.*

Le texte 19081 (Figure 8.25) est un nouvel exemple de texte dont le classement est amélioré par le modèle avec contexte, car *consumer* est précisé, ici, par la présence de (*index, pct*), *index* est précisé par (*price, rose*).

```
CANADA CONSUMER PRICE INDEX UP 0.6 PCT IN MAY
Canada's consumer price index rose 0.6
pct in May to 137.8, base 1981, following a 0.4 pct rise in
April and a 0.5 pct rise in May 1986, Statistics Canada said.
    The May year-on-year rise was 4.6 pct, compared with a 4.5
pct rise in April.
 Reuter
```

Figure 8.25 : *Texte 19081 pertinent pour le thème cpi.*

En revanche, le texte 18001 (Figure 8.26) est un exemple de texte dont le classement diminue puisque son score passe de 0,86 avec le modèle linéaire à 0,52 avec le modèle utilisant le contexte. Pour ce texte, le seul mot-clef présent est *inflation*, mais il n'est accompagné d'aucun contexte pertinent à chaque fois sauf dans le dernier cas avec la présence de *pct.*

```
BRAZIL'S SARNEY RENEWS CALL FOR WAR ON INFLATION
President Jose Sarney today declared "a war without quarter" on inflation
and said the government would watch every cent of public expenditure.
    Sarney, addressing his cabinet live on television, also reiterated that
he intended to remain in power for five years, until 1990. There has been a
long-running political debate about how long his mandate should be.
    Brazil is currently suffering from the worst inflation of its history.
In April monthly inflation reached 21 pct.
 Reuter
```

Figure 8.26 : *Texte 18001 pertinent pour le thème cpi.*

Ce dernier exemple montre que si la performance du thème *cpi* a augmenté grâce à l'utilisation du contexte, certains textes pertinents sont tout de même moins bien classé par l'utilisation du contexte que par le modèle linéaire.

8.5.3.2 Exemples de thèmes dont les performances se dégradent

Pour le thème *hog* (thème 51) qui contient seize documents pertinents sur la base d'apprentissage et six sur la base de test, les performances avec chacune des méthodes sont présentées à la Figure 8.27 :

	UAP	*F*	11-pt
Modèle linéaire	89,72	80,00	82,73
Modèle avec contexte	78,57	71,43	80,09

Figure 8.27 : *Performance pour le thème hog en fonction du modèle.*

Les scores des documents obtenus avec chaque modèle sont présentés à la Figure 8.28. Pour les trois premiers textes pertinents, le modèle avec contexte renforce leur score grâce à l'utilisation de la présence du mot *slaughter* dans le contexte de *hog* comme le montre l'exemple de texte à la Figure 8.29. Cependant, le score des trois derniers textes diminue, car si le mot *hog* est présent dans ces textes, il n'est jamais entouré d'un contexte pouvant renforcer ce mot : le modèle n'est plus assez sensible.

Modèle linéaire		Modèle avec contexte	
16255	0.886748	16255	0.917718
15532	0.886748	15532	0.917718
17823	0.643343	17823	0.850644
17827	0.363257	17827	0.134609
21367	0.332782	21367	0.131715
19555	0.272811	19555	0.0691451

Figure 8.28 : *Ensemble des scores des textes pertinents pour le thème hog.*

```
HOG AND CATTLE SLAUGHTER GUESSTIMATES
Chicago Mercantile Exchange floor
traders and commission house representatives are guesstimating today's hog
slaughter at about 280,000 to 300,000 head versus
294,000 week ago and 303,000 a year ago.
    Cattle slaughter is guesstimated at about 120,000 to 126,000 head
versus 120,000 week ago and 124,000 a year ago.
 Reuter
```

Figure 8.29 : *Texte 16255 pertinent pour le thème hog.*

Cet exemple illustre bien l'un des problèmes qui se pose quand le contexte est pris en considération : certains mots voient leur influence diminuer s'ils ne sont pas entourés d'un contexte pour les désambiguïser. Si ce cas se présente trop souvent, les performances du modèle avec contexte deviennent inférieures à celles du séparateur linéaire.

8.5.4 Représentation des textes avec les racines lexicales

La représentation des textes avec les racines lexicales avait conduit à une dégradation des résultats avec le séparateur linéaire (cf. chapitre 7), notamment à cause de l'ambiguïté ajoutée par l'utilisation des racines. Or, comme le modèle avec contexte effectue une désambiguïsation des mots, on peut espérer tirer profit de l'utilisation des racines sans en subir les conséquences négatives.

Par exemple, la racine *custom*, qui est un descripteur utilisé dans le thème *interest* (cf. chapitre 6), est maintenant précisé par le contexte (*repurchas, reserv, feder, fed, via*) qui sont les racines respectives de *repurchase*, *reserves*, *federal, fed* et *via* ; dans beaucoup de cas, il ne risque plus d'être confondu avec d'autres significations de *custom*.

8.5.4.1 Résultats sur l'ensemble des thèmes

La Figure 8.30 présente les résultats obtenus sur l'ensemble du corpus Reuters avec l'utilisation de racines. Comme précédemment, la première ligne correspond à l'utilisation du contexte positif et la deuxième ligne à l'utilisation des contextes positif et négatif.

	Nombre de poids du réseau	UAP	*F*	11-pt
Moyenne sur l'ensemble des thèmes	108,38 125,92	74,71 73,80	66,64 65,81	74,93 74,20
Moyenne pour les thèmes 1 à 10	183,40 214,6	89,91 89,76	86,20 86,56	86,76 86,64
Moyenne pour les thèmes 11 à 40	112,03 132,90	83,95 83,88	83,07 82,84	84,30 84,51
Moyenne pour les thèmes 41 à 60	111,55 126,20	70,45 70,40	68,34 69,20	71,76 71,79
Moyenne pour les thèmes 61 à 90	78,70 90,43	63,69 61,09	42,87 39,98	64,16 61,77

Figure 8.30 : *Ensemble des résultats avec les racines lexicales sur le corpus Reuters. La première ligne correspond à l'utilisation du contexte positif et la deuxième ligne à l'utilisation du contexte négatif.*

Comme précédemment, l'utilisation du contexte négatif fait légèrement diminuer les résultats tout en augmentant le nombre de paramètres du modèle.

La comparaison de ces résultats avec les résultats de la Figure 8.13 montre que la performance globale est améliorée avec l'utilisation de racines. Cependant l'analyse par sous-ensemble de thèmes montre que les résultats ne sont améliorés que sur le sous-ensemble des thèmes 61 à 90, c'est-à-dire sur l'ensemble des thèmes comportant moins de dix documents pertinents sur la base d'apprentissage.

La Figure 8.31 représente, pour les soixante premiers thèmes, la précision moyenne non interpolée obtenue en utilisant uniquement le contexte positif sans l'utilisation de racines (représentation "brute") et avec l'utilisation de racines.

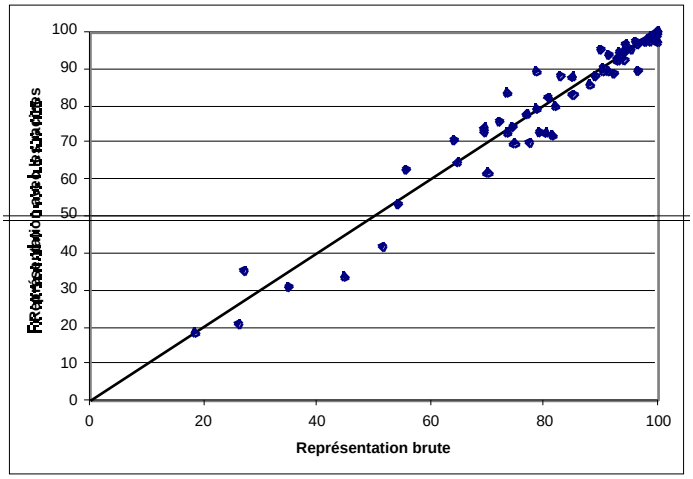


Figure 8.31 : *Comparaison thème à thème de la précision moyenne non interpolée pour les soixante premiers thèmes : représentation brute et avec utilisation de racines.*

Les points sont répartis de chaque côté de la diagonale, mais en moyenne, les performances sont supérieures avec la représentation "brute".

L'utilisation du contexte a permis une désambiguïsation partielle et une amélioration des performances par rapport au séparateur linéaire utilisant les racines, comme le prouvent les comparaisons des performances notamment sur le sous-ensemble des thèmes 41 à 60 (Figure 8.32).

	UAP	*F*	11-pt
Modèle linéaire utilisant les racines	67,5	65,4	68,3
Modèle avec contexte utilisant les racines	70,45	68,34	71,76

Figure 8.32 : *Comparaison des performances sur le sous-ensemble de thèmes 41 à 60 entre le modèle linéaire utilisant les racines et le modèle avec contextes utilisant les racines.*

Cependant, cette désambiguïsation n'est pas suffisante puisque sur les soixante premiers thèmes, les performances sont meilleures lorsque les mots ne sont pas substitués par leur racine.

8.5.4.2 Analyse des thèmes 61 à 90

La variance des résultats est très grande sur le sous-ensemble des thèmes 61 à 90 ; la Figure 8.33 reprend les résultats obtenus sur ce sous-ensemble avec le séparateur linéaire, avec le modèle utilisant les contextes positifs, et avec le modèle utilisant les contextes positifs et les racines. La Figure 8.34 présente les courbes rappel-précison pour les trois modèles.

	UAP	*F*	11-pt
Modèle linéaire	59,03	43,79	59,64
Modèle avec contexte	55,52	39,76	55,88
Modèle avec contexte et racines	63,69	42,87	64,16

Figure 8.33 : *Comparaison des performances sur le sous-ensemble des thèmes 61 à 90.*

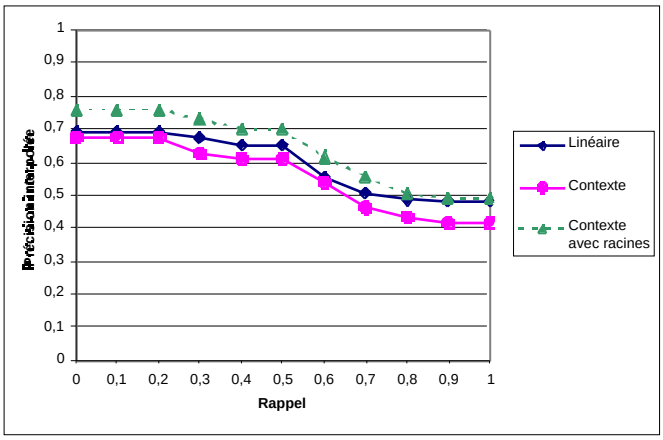


Figure 8.34 : *Courbes rappel-précision interpolée pour le sous-ensemble des thèmes 61 à 90.*

Selon le critère de la précision moyenne non interpolée, l'utilisation du contexte avec les stems est le meilleur des modèles pour ce sous-ensemble de textes. Cependant, selon la mesure *F*, les différences sont moins grandes, car comme on l'a signalé au chapitre 7, le classement des textes peut être parfait, sans qu'il soit possible d'en tirer profit.

Par exemple, la Figure 8.35 présente les résultats obtenus avec chacune des méthodes sur le thème *palmkernel* (thème 78) qui contient deux documents pertinents sur la base d'apprentissage et un document pertinent sur la base de test. Les différences mesurées avec la précision moyenne non interpolée sont très élevées, mais la mesure de *F* montre qu'il est, en fait, impossible de séparer les documents pertinents des autres documents : aucun des modèles n'est satisfaisant.

	UAP	*F*	11-pt
Modèle linéaire	0,03	0,06	0,03
Modèle avec contexte	11,11	0,06	11,11
Modèle avec contexte et racines	100	0,06	100

Figure 8.35 : *Résultats pour le thème palmkernel.*

Finalement, il n'est pas possible de tirer de conclusions sur le bien-fondé de l'utilisation de racines à partir de l'analyse de ce sous-ensemble à cause de la variance des résultats due au faible nombre de documents pertinents sur la base de test. De plus, pour ce sous-ensemble, les performances sont également plus sensibles aux variations de la valeur des hyperparamètres pour les modèles avec contexte : les différences peuvent être dues au choix de ces valeurs plutôt qu'au choix des descripteurs d'entrées.

8.5.5 Représentation des textes avec les lemmes

Le modèle utilisant le contexte a été appliqué comme précédemment, mais en utilisant les lemmes et non plus les racines lexicales comme au paragraphe précédent.

La Figure 8.36 présente les résultats : comme précédemment la première ligne correspond à l'utilisation du contexte positif et la deuxième à l'utilisation des contextes positif et négatif.

	Nombre de poids du réseau	UAP	*F*	11-pt
Moyenne sur l'ensemble des thèmes	112,66 127,91	74,58 74,79	67,08 67,87	74,51 74,89
Moyenne pour les thèmes 1 à 10	187,70 214,3	89,82 90,21	85,52 86,33	86,82 87,43
Moyenne pour les thèmes 11 à 40	115,06 133,77	84,02 84,05	81,73 82,75	84,07 84,25
Moyenne pour les thèmes 41 à 60	124,25 136,55	71,44 73,05	68,86 70,46	72,07 73,99
Moyenne pour les thèmes 61 à 90	78,50 88,66	62,65 62,01	45,50 45,49	62,93 62,41

Figure 8.36 : *Ensemble des résultats avec les lemmes sur le corpus Reuters. La première ligne correspond à l'utilisation du contexte positif et la deuxième ligne à l'utilisation des deux contextes.*

La comparaison des résultats avec les résultats de la Figure 8.30 obtenus avec les racines montre que, si l'on se réfère à la précision moyenne non interpolée, l'utilisation de lemmes est préférable à l'utilisation des racines, mais si l'on se réfère à la mesure de *F*, l'utilisation des racines est préférable.

La comparaison thème à thème des précisions moyennes non interpolées, pour les soixante premiers thèmes, est présentée à la Figure 8.37. Les points sont répartis de chaque côté de la diagonale, et seul deux thèmes présentent des différences de performances majeures.

Pour le thème *retail* (thème 41), la précision moyenne non interpolée passe de 35,0 à 0,06, mais comme on l'a précisé précédemment, ce thème ne comporte que deux documents pertinents sur la base de test : les variations de performances ne sont pas significatives.
Pour le thème *soy-oil* (thème 53) qui comporte quatorze documents pertinents sur la base d'apprentissage et onze sur la base de test, la performance passe de 26,1 sans l'utilisation de lemmes à 68,2 avec l'utilisation de lemmes. L'amélioration, pour ce thème, est réelle, mais c'est le seul thème avec une amélioration si nette.

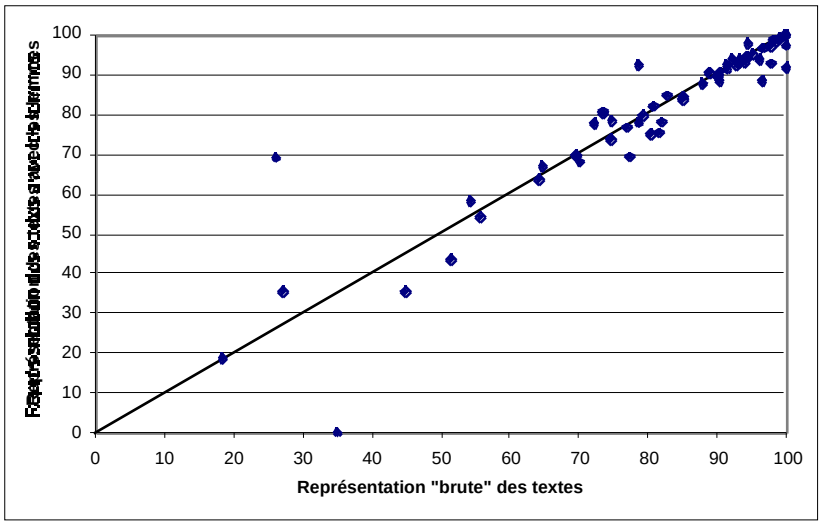


Figure 8.37 : *Comparaison de la précision moyenne non interpolée thème à thème pour les 60 premiers : sans utilisation de lemmes et avec utilisation de lemmes.*

8.5.6 Comparaison des performances obtenues sur Reuters avec d'autres études

Le corpus Reuters a souvent été utilisé comme corpus dans les publications afin de tester différentes approches. Il est possible de comparer les performances obtenues par notre méthode avec les performances obtenues par d'autres approches sur ce même corpus. Ces comparaisons sont intéressantes, car on peut supposer que chaque auteur maîtrise bien la technique qu'il met en œuvre, et qu'il utilise des algorithmes adéquats, ce qui n'est pas toujours le cas lorsqu'une même personne teste des méthodes dont elle n'est pas spécialiste.

Cependant, on ne dispose, pour faire cette comparaison, que de certaines mesures, alors que d'autres critères doivent être pris en considération, comme la marge de progression de chaque méthode, la simplicité de mise en œuvre, ou les temps de calculs.

Il faut noter que la comparaison n'est pas tout à fait objective puisque nous avons largement utilisé la base de test tout au long de ce mémoire pour mieux comprendre le réglage de certains paramètres ; la compétition TREC se prête mieux aux comparaisons objectives.

[Schapire *et al.*, 1998] ont utilisé le corpus Reuters pour tester deux algorithmes : une formule de Rocchio améliorée, et une méthode nommée *AdaBoost* qui est reconnue comme une application efficace de la méthode de *Boosting* [Freund et Schapire, 1997]. Ces deux méthodes sont comparées à l'algorithme *Sleeping Expert* proposé par [Cohen et Singer, 1996]. Les deux méthodes testées dans cet article obtiennent de meilleurs résultats que la méthode *Sleeping Expert*, et sur le corpus Reuters, la formule de Rocchio est meilleure que l'algorithme *Adaboost*, ce qui tend à prouver que son implémentation est très efficace.

Nous avons comparé nos résultats avec ceux obtenus par la méthode de Rocchio. L'implémentation proposée repose sur un codage efficace des fréquences (le codage Lnu) une sélection des documents non pertinents, et une optimisation des poids trouvés. Les comparaisons sont facilitées, car, d'une part, les auteurs considèrent le même découpage du corpus que celui que nous avons utilisé, et, d'autre part, les résultats thème par thème sont disponibles sur le web[3].

[3] http://www.research.att.com/~singhal/sigir98-rocboost.html

La Figure 8.38 présente les résultats de la méthode de Rocchio.

	UAP	*F*
Moyenne sur l'ensemble des thèmes	70,8	56,0
Moyenne pour les thèmes 1 à 10	86,3	80,4
Moyenne pour les thèmes 11 à 40	81,3	73,7
Moyenne pour les thèmes 41 à 60	70,7	59,2
Moyenne pour les thèmes 61 à 90	54,1	27,1

Figure 8.38 : *Résultats de la méthode Rocchio proposé par [Schapire et al., 1998].*

La Figure 8.39 présente les comparaisons thème à thème entre les résultats de la méthode de Rocchio et notre approche (appelée réseaux de neurones) avec l'utilisation du contexte positif : chaque point représente un thème, son abscisse est la précision moyenne non interpolée obtenue par la méthode de Rocchio et son ordonnée est celle obtenue par notre méthode.

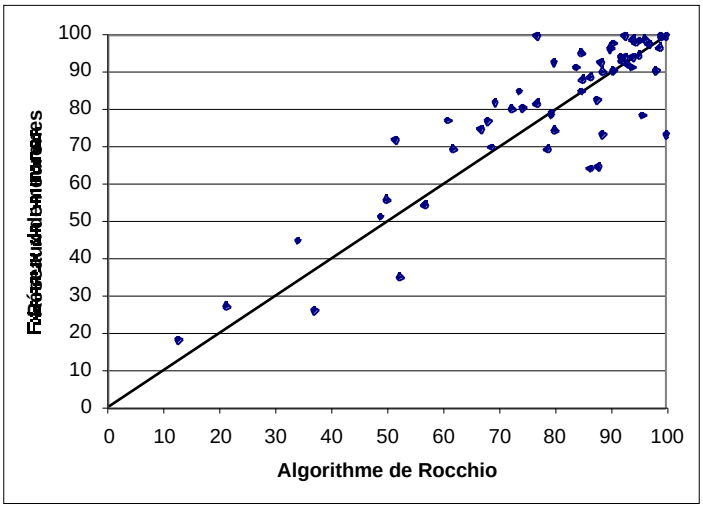


Figure 8.39 : *Comparaison pour les soixante premiers thèmes du corpus Reuters entre la méthode de Rocchio et notre méthode.*

Cette comparaison montre que notre approche est supérieure pour une majorité de thèmes, même si, pour certains thèmes, l'algorithme de Rocchio obtient de meilleurs résultats.

8.5.7 Conclusions des expériences sur le corpus Reuters

Les expériences menées sur le corpus Reuters ont montré que l'utilisation du contexte local améliorait les performances par rapport au modèle linéaire. Il est intéressant de noter que cette amélioration bénéficie surtout aux thèmes dont les performances étaient les plus basses.

L'utilisation du contexte négatif ne diminue pas les résultats, mais elle augmente le nombre de paramètres sans apporter d'amélioration significative.

Malgré la désambiguïsation partielle effectuée grâce à l'utilisation du contexte, la substitution des mots par leur racine ou par leur lemme n'a pas apporté d'amélioration des performances. De plus, bien que ces deux approches réduisent le nombre de paramètres du modèle, elles ajoutent une étape supplémentaire dans la préparation des textes.

8.6 Mise en œuvre sur le corpus TREC-8

Pour tester cette méthode sur TREC-8, on reprend les expériences sur le corpus TREC-8 avec la sélection de descripteurs qui avait abouti au système S2N2.

Pour chacun des descripteurs sélectionnés, on ajoute, dans l'architecture neuronale de la Figure 8.6, les cinq premiers contextes positifs trouvés pour chacun de ces mots.

Les valeurs des hyperparamètres sont choisies *a priori* pour chacun des thèmes :

$$\alpha_0 = 0{,}001$$

$$\alpha_1 = 1{,}0$$

$$\alpha_2 = 0{,}5$$

Les performances sont comparées aux résultats officiels présentés à la compétition (S2N2) et aux améliorations du chapitre 7 (améliorations effectuées après la compétition) apportées à ce modèle appelé *modèle linéaire*.

La Figure 8.40 montre la moyenne des précisions moyennes non interpolées pour chacune de ces méthodes.

	S2N2	Modèle linéaire	Modèle avec contexte
Performance (UAP)	30,7	34,8	38,2

Figure 8.40 : *Précision moyenne non interpolée (UAP) avec ou sans contexte pour l'ensemble des cinquante thèmes du corpus TREC-8.*

Le modèle utilisant le contexte améliore significativement les résultats, puisque l'introduction du contexte a permis une amélioration de 10 % des performances par rapport au modèle linéaire. Par rapport aux résultats de la compétition (S2N2), l'amélioration des résultats est de 24 %.

Les courbes rappel-précision de la Figure 8.41 confirment l'amélioration observée et mettent en évidence la supériorité du modèle utilisant le contexte.

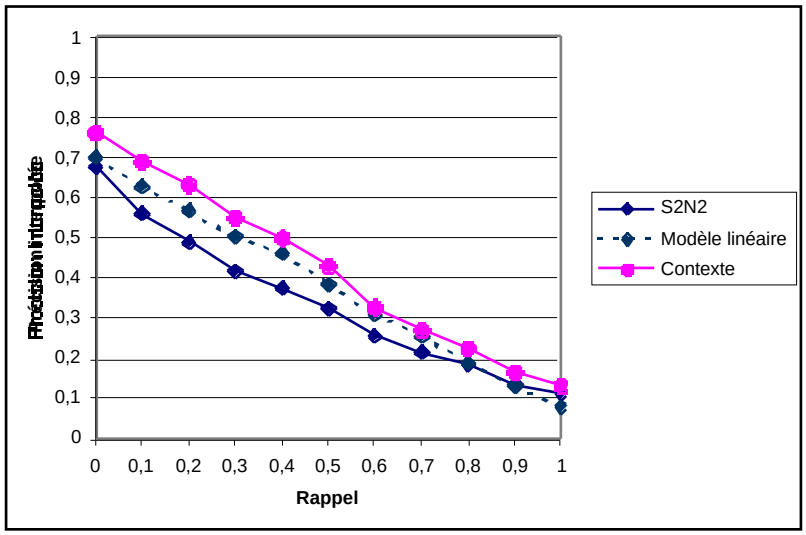


Figure 8.41 : *Comparaison des courbes rappel-précision interpolée pour l'ensemble des thèmes du corpus TREC-8 pour les trois méthodes.*

Enfin, la Figure 8.42 reprend l'ensemble des résultats obtenus par chaque participant pour la compétition ; les nouvelles performances apparaissent sous la dénomination *contexte*. Cette comparaison montre que notre système est maintenant comparable avec les meilleurs systèmes.

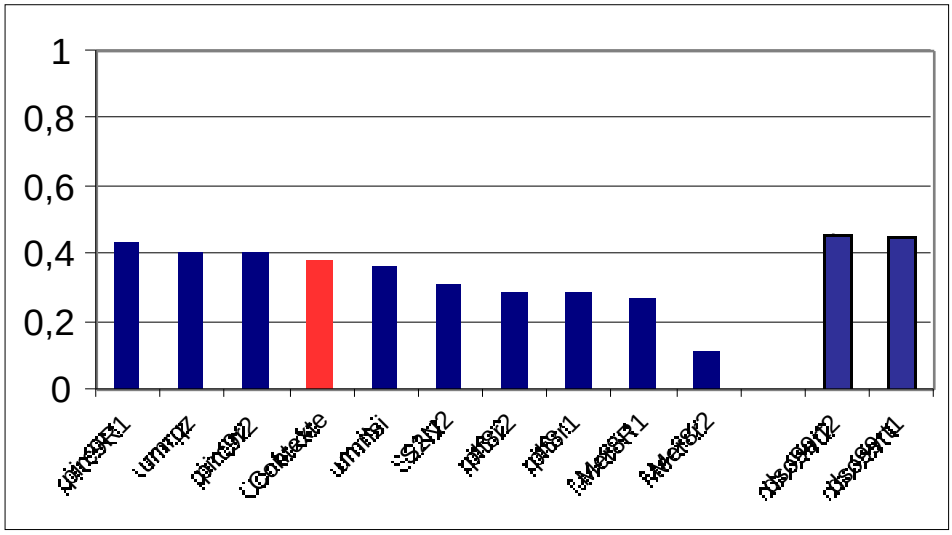


Figure 8.42 : *Résultats de l'ensemble des participants à la tâche de Routing de TREC-8. Les nouvelles performances apparaissent sous la dénomination contexte.*

L'ensemble des résultats confirme les résultats obtenus sur le corpus Reuters : malgré l'augmentation du nombre de paramètres, l'utilisation du contexte améliore les performances.

8.7 Mise en œuvre sur le corpus TREC-9

8.7.1 Détail des paramètres utilisés pour l'apprentissage

Ce paragraphe décrit notre participation à la sous tâche de routing de TREC-9 qui est détaillée dans [Stricker *et al.*, 2001]. La présentation du corpus ainsi que la définition des thèmes est précisée dans l'annexe A.

Pour cette compétition, nous avons mis en œuvre les techniques décrites dans ce chapitre.

Comme le montre la description du corpus à l'annexe A, le nombre de documents pertinents disponibles pour les bases d'apprentissage est beaucoup plus faible sur ce corpus que sur les autres corpus étudiés. Par conséquent, la sélection des descripteurs a été effectuée uniquement par la détermination du vocabulaire spécifique et la méthode d'orthogonalisation de Gram-Schmidt n'a pas été utilisée.

Les vingt-cinq premiers descripteurs trouvés par la méthode du vocabulaire spécifique sont sélectionnés et leurs contextes positif et négatif sont déterminés par la méthode du paragraphe 8.2.1.

Pour chaque thème l'architecture neuronale décrite à la Figure 8.6 contient donc 25 neurones cachés. Le nombre d'entrées liées à chaque neurone caché est défini comme suit : les cinq

premiers contextes positifs sont pris en considération s'ils apparaissent dans plus de deux documents pertinents et les cinq premiers contextes négatifs s'ils apparaissent dans plus de dix documents non pertinents.

L'apprentissage est effectué avec la méthode du *weight decay* ; les valeurs des hyperparamètres sont fixées comme précédemment.

8.7.2 Détail des fichiers envoyés pour la compétition

Nous avons présenté trois fichiers de résultats pour la tâche de routing dont les caractéristiques sont reprises à la Figure 8.43 (les définitions de *OHSUMED queries* et *MeSH sample* sont précisées à l'annexe A ainsi que les précisions sur les annotations manuelles)

	Ensemble des requêtes	Utilisation des annotations manuelles
S2RNr1	*OHSUMED queries* (63 thèmes)	non
S2RNr2	*OHSUMED queries* (63 thèmes)	oui
S2RNsamp	*MeSH Sample* (500 thèmes)	non

Figure 8.43 : *Détail des fichiers résultats soumis pour TREC-9.*

Les résultats obtenus par chaque système ont été présentés lors de la conférence TREC-9 qui s'est déroulée du 13 au 16 novembre 2000.

8.7.3 Résultats obtenus sur les 63 thèmes OHSUMED

Comme lors de la conférence TREC-8, les performances sont évaluées par la précision moyenne non interpolée.

La Figure 8.44 expose les résultats obtenus sur les 63 thèmes OHSUMED et la Figure 8.45 compare ces résultats à ceux des autres candidats.

Parmi tous les résultats proposés, seul notre fichier S2RNr2 a utilisé les annotations manuelles ; la performance obtenue avec ce fichier doit donc être différenciée des autres.

Il est intéressant de noter que la meilleure performance est obtenue par la méthode utilisant les annotations manuelles. Ce résultat prouve que les méthodes à base d'apprentissage numérique peuvent très facilement tirer parti d'informations autres que le texte lui-même sans qu'il soit nécessaire de changer quoique ce soit à la méthode.

Notre fichier S2RNr1 qui n'utilise pas les annotations manuelles est directement comparable à tous les autres participants ; les résultats montrent que notre approche conduit aux meilleures performances.

	S2RNr1	S2RNr2
Moyenne des précisions moyennes non interpolées	0,343	0,385
Nombre de thèmes où notre score est le meilleur	9/63 (14%)	29/63 (46%)
Nombre de thèmes où notre score est supérieur à la médiane	61/63 (97%)	61/63 (97%)

Figure 8.44 : *Résultats pour la tâche de routing de TREC-9 sur les 63 thèmes OHSUMED.*

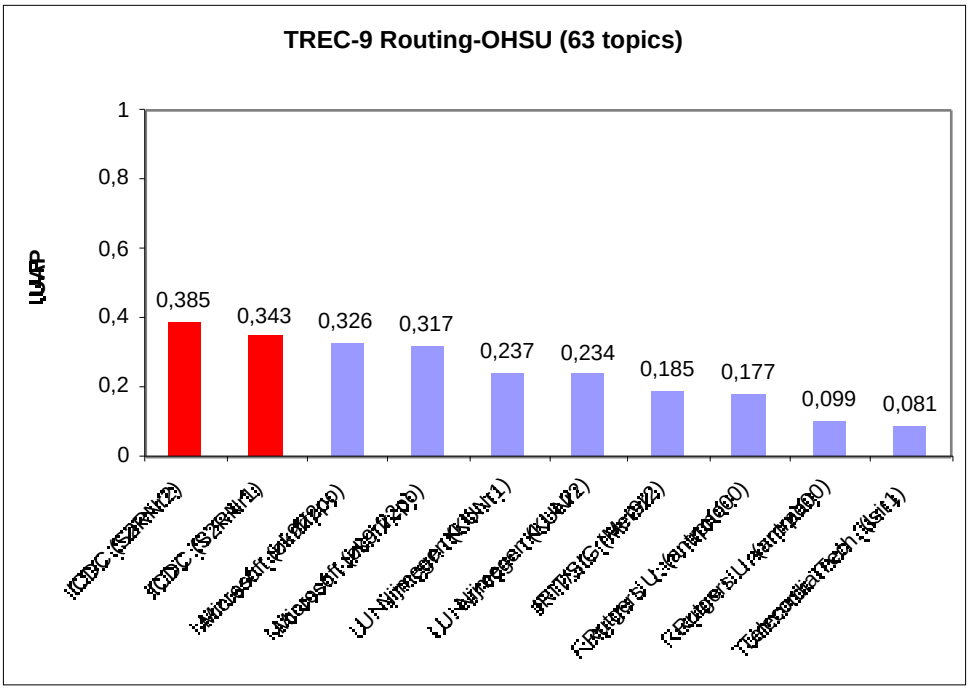


Figure 8.45 : *Moyennes des précisions moyennes non interpolées de l'ensemble des participants à la tâche de routing de TREC-9 pour les 63 thèmes OHSUMED.*

8.7.4 Résultats obtenus sur les 500 thèmes MeSH.

Comme précédemment, les performances sont évaluées par la moyenne des précisions moyennes non interpolées calculées sur chaque thème.

La Figure 8.44 expose les résultats obtenus sur les 500 thèmes MeSH et la Figure 8.45 compare ces résultats à ceux des autres candidats.

	S2RNsamp
Moyenne des précisions moyennes non interpolées	0,335
Nombre de thèmes où notre score est le meilleur	364/500 (73%)
Nombre de thèmes où notre score est supérieur à la médiane	494/500 (99%)

Figure 8.46 : *Résultats pour la tâche de routing de TREC-9 sur les 500 thèmes MeSH sample.*

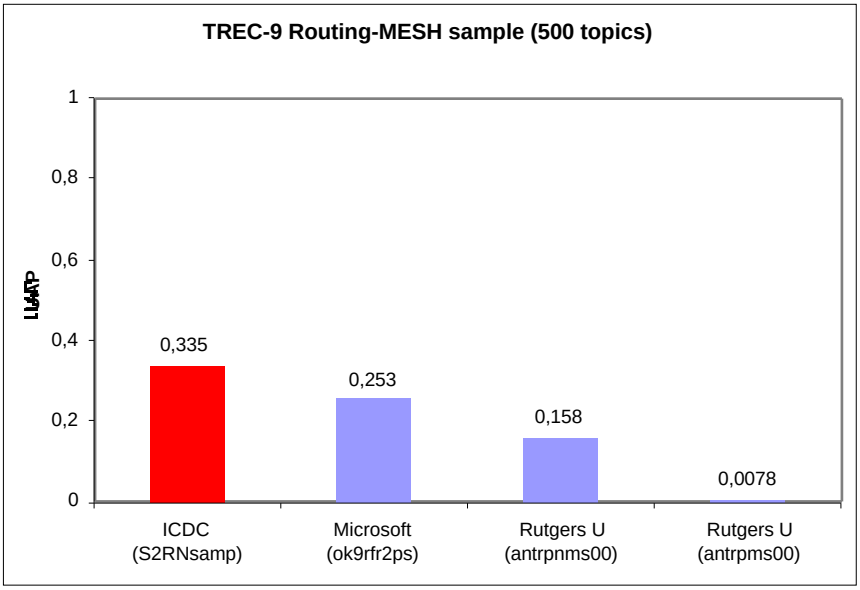


Figure 8.47 : *Moyenne des précisions moyennes non interpolées de l'ensemble des participants à la tâche de routing de TREC-9 pour les 500 thèmes MeSH sample.*

Sur cet ensemble de thèmes, notre approche obtient également les meilleurs résultats. Les écarts sont plus importants que ceux observés précédemment puisque notre système obtient une amélioration de 32 % par rapport au système suivant.

8.7.5 Revue des différents systèmes

Il n'est pas encore possible faire une présentation précise des différentes approches présentées[4] ; nous en présentons juste le principe de base.

Les fichiers **ok9rf2po**, **ok9rfrpo**, et **ok9rfr2ps** ont été présentés par Microsoft et reposent tous sur le modèle OKAPI [Robertson et Sparck Jones, 1976] [Robertson et Walker, 2000] qui est une implémentation du modèle probabiliste présenté au chapitre 2.

Les fichiers **KUNr1** et **KUNr2** ont été présentés par l'université de Nijmegen. Leur approche utilise l'algorithme appelé *Winnow*.

Le fichier **Mer9r2** a été proposé par l'université de Toulouse III. Il s'agit d'une variante du modèle appelé Mercure [Boughanem *et al.*, 2000] qui utilise un réseau de neurones pour implémenter le modèle probabiliste.

Les fichiers **antrpno00**, **antrpo00**, et **antrpnms00** ont été proposés par l'université Rutgers qui a développé un algorithme d'apprentissage appelé *Logical Analysis of Data* (LAD).

Nous n'avons pu trouver aucune information sur le système utilisé pour produire le fichier **lsir1**.

8.7.6 Conclusion

Les résultats obtenus sur la tâche de routing de TREC-9 montrent que notre approche est performante puisqu'elle a été classée première aussi bien sur l'ensemble des 63 thèmes que sur l'ensemble des 500 thèmes.
Sur l'ensemble des 63 thèmes le nombre moyen de documents pertinents disponibles pour la base d'apprentissage est de 10,6 ; donc même avec un nombre aussi faible il est possible de mettre en œuvre notre approche.

[4] Les articles complets seront disponibles sur le site internet de TREC (http://trec.nist.gov) à partir du mois de février 2001.

Les écarts entre notre système et le suivant sont plus élevés sur l'ensemble MeSH (500 thèmes) que OHSUMED (63 thèmes), alors que les deux fichiers ok9rfr2po (0,317 sur OHSUMED) et ok9rfr2ps (0,253 sur MeSH) correspondent au même système avec les mêmes paramètres. Comme sur l'ensemble MeSH le nombre moyen de documents pertinents pour la base d'apprentissage est de 46, cela prouve que notre système a su tirer avantage de ce plus grand nombre de documents pertinents.

Enfin, notre méthode est suffisamment rapide pour pouvoir être utilisée sur un ensemble de 500 thèmes en un temps limité. En effet, la liste des 500 thèmes a été délivrée 14 jours avant la date limite d'envoi des résultats et peu de participants ont fourni des résultats sur cet ensemble.

8.8 Conclusion

Nous avons proposé une extension du séparateur linéaire du chapitre 7, qui consiste à utiliser le contexte local des descripteurs sélectionnés pour effectuer une désambiguïsation. Cette méthode de désambiguïsation repose sur l'étude du corpus étudié pour définir le contexte pertinent d'un mot.

Comme le nombre de paramètres du modèle augmente considérablement par rapport au modèle linéaire, l'utilisation d'une méthode de régularisation est indispensable pendant l'apprentissage. Grâce à l'utilisation du *weight decay*, l'apprentissage est effectué efficacement même pour les thèmes comportant peu d'exemples pertinents.

Les méthodes issues de l'approche bayesienne n'ont pas permis, sur notre problème, de trouver des valeurs adéquates pour les hyperparamètres. Cependant, la méthode la plus simple, qui consiste à fixer les valeurs des hyperparamètres *a priori* donne de bons résultats. Cette méthode qui n'est pas optimale présente le grand avantage de ne pas ajouter de calculs ni de nécessiter de méthodes de validations croisées et donc n'allonge pas la durée de l'apprentissage. Ce paramètre peut être crucial quand il existe beaucoup de thèmes à traiter comme pour TREC-9, le nombre de profils à construire s'élève à cinq cents.

Finalement, les mises en œuvre sur les corpus Reuters, TREC-8 et TREC-9 ont montré que cette approche améliore les performances par rapport au modèle linéaire utilisé précédemment, et que, sur ces trois corpus, les résultats sont comparables aux meilleurs résultats publiés.

Chapitre 9 Applications opérationnelles

Ce chapitre expose comment les techniques décrites dans ce mémoire ont été intégrées dans une application opérationnelle de filtrage des dépêches de l'AFP.

Dans une première application, ces modèles sont utilisés pour contrôler des filtres construits avec des systèmes à base de règles. Cette application limite le travail d'un administrateur tout en assurant la qualité du service rendu aux utilisateurs.

La deuxième application s'adresse aux utilisateurs eux-mêmes en leur permettant de développer leur propre filtre. Les recherches sur cette application doivent être poursuivies, mais les premiers résultats semblent prometteurs.

9.1 Présentation d'un système de filtrage : l'application ExoWeb

Exosème est un procédé de filtrage d'informations développé à la Direction des Techniques Avancées de la Caisse des Dépôts à partir des années 1990. Ce procédé a donné lieu à une application, ExoWeb, qui filtre les dépêches de l'Agence France-Presse en temps réel selon des thèmes prédéfinis. Ce système fonctionne actuellement sur l'intranet de la Caisse des Dépôts et compte plus d'une centaine d'utilisateurs qui le consultent via un navigateur web.

Ces travaux ont notamment donné lieu en 1997 à la naissance d'une filiale appelée CDC-Mercure[1] qui commercialise un quotidien Internet des collectivités locales. Ce serveur fournit, entre autres, aux responsables des collectivités locales une classification en temps réel des quelque 1500 dépêches quotidiennes de l'AFP ainsi que des résumés de dépêches.

9.1.1 L'application ExoWeb

La source de données utilisées pour le filtrage de textes est le fil des dépêches de l'Agence France-Presse[1]. Devant la masse de dépêches produites chaque jour, l'AFP a été conduite à diviser sa source de dépêches en fils spécialisés dont les deux principaux sont le fils général (environ 1200 dépêches par jour) et le fil économique (environ 1000 dépêches par jour).

L'application ExoWeb propose trois services principaux :

[1] http://www.cdc-mercure.fr

1. un filtrage en temps réel des dépêches du fil économique de l'AFP.
2. une recherche d'information effectuée grâce à l'utilisation d'un moteur de recherche indexé sur deux années d'archives de dépêches.
3. une application de groupement de dépêches en fonction des différents sujets d'actualités.

9.1.1.1 Le filtrage dans ExoWeb

L'application propose le filtrage en temps réel des dépêches en fonction de thèmes prédéfinis. L'utilisateur s'abonne aux thèmes qui l'intéressent en choisissant ses thèmes de prédilection dans un catalogue. Le nombre de thèmes disponibles est d'environ 180 à choisir parmi différents domaines : chiffres boursiers, chiffres économiques, vie des entreprises, monnaie, technologie, secteurs d'activité, Caisse des Dépôts, … .

Par exemple, pour le domaine "vie des entreprises", cinq thèmes différents sont proposés : *introduction en bourse*, *rating*, *participations*, *résultats* et *privatisation.*

Parmi ces thèmes, certains reposent fortement sur des modules de reconnaissance des noms propres (par exemple le thème sur la région *paca*), mais d'autres regroupent des concepts plus larges comme les échanges de participations entre entreprises, les perturbations dans les transports, ou encore les annonces de résultats des entreprises.

Un utilisateur s'abonne à tous les thèmes qu'il souhaite, et la liste de ses thèmes apparaît sur le côté de son navigateur. En sélectionnant l'un des thèmes, les titres des dix dernières dépêches sélectionnées apparaissent en lien hypertexte qui donne accès au texte de la dépêche.

La Figure 9.1 est un exemple des dépêches sélectionnées pour le thème *résultat* ainsi qu'un extrait de l'une de ces dépêches ; le passage pertinent apparaît en couleur dans le navigateur (en italique sur la figure). Le système de filtrage effectue donc également de l'extraction d'informations.

[1] http://www.afp.com

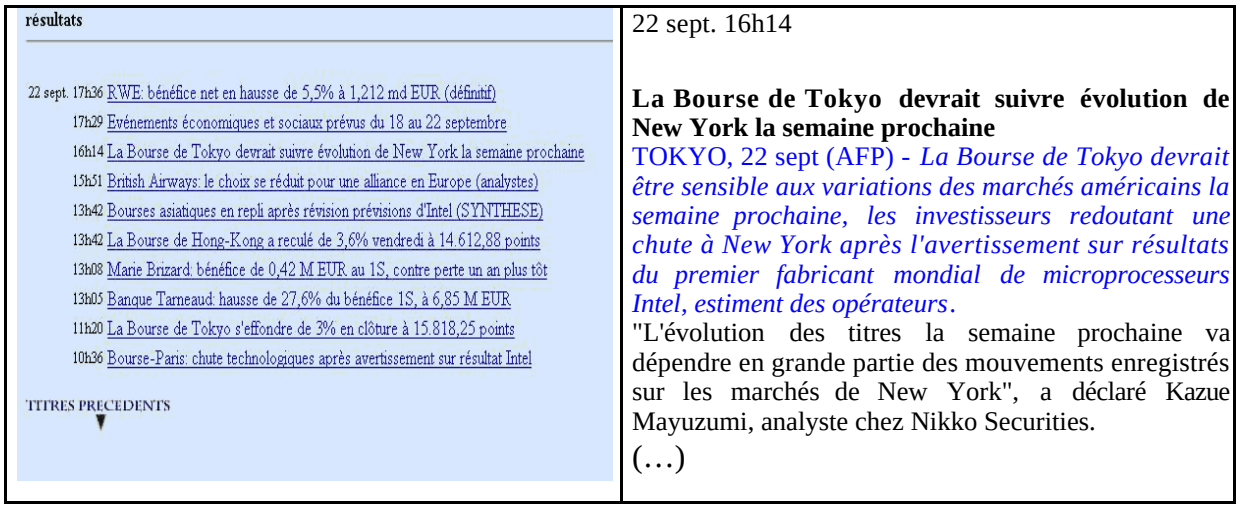
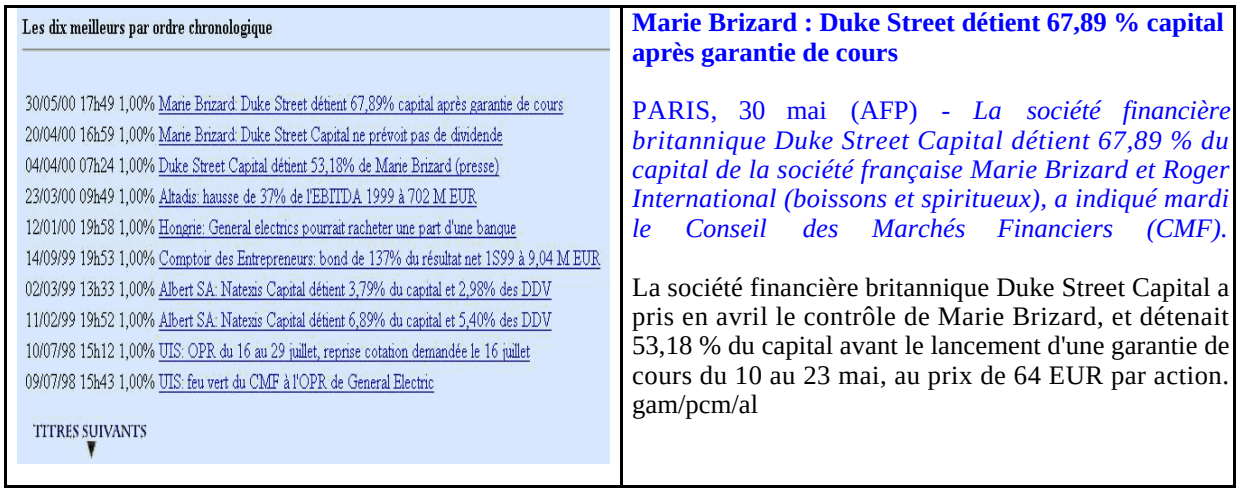


Figure 9.1 : *Dépêches sélectionnées pour le thème résultat.*

9.1.1.2 La recherche d'informations

Les dépêches du fil économique de l'Agence France-Presse sont archivées depuis deux ans, et indexées pour pouvoir être consultées grâce à un moteur de recherche de type booléen. Le moteur de recherche utilisé est le moteur commercialisé par la société Verity[1]. Ce moteur permet d'effectuer des requêtes booléennes variées en utilisant les opérateurs *or*, *and*, *not*, *near*. La Figure 9.2 est un exemple de recherche effectuée grâce à ce moteur de recherche avec la requête : "détient <near> capital". Sur la partie gauche de cette figure apparaît le titre des dépêches qui répondent le mieux à la requête, classées par ordre chronologique. La partie droite est un exemple de texte sélectionné avec le passage pertinent qui apparaît en couleur dans le navigateur (en italique sur la figure).



[1] http://www.verity.com

Figure 9.2 : *Résultat de la recherche : "détient <near> capital".*

L'indexation garde également pour chaque dépêche les thèmes assignés par ExoWeb afin de retrouver par une simple requête toutes les dépêches sélectionnées pour un thème donné. Par exemple, pour retrouver toutes les dépêches qui ont été classées dans le thème *perturbation* par ExoWeb, il suffit d'utiliser la requête suivante dans le moteur de recherche : _perturbation (le nom du thème précédé du signe _).

9.1.1.3 Le groupement statistique

Enfin, une dernière application propose de fournir des groupes de dépêches sur les thèmes d'actualité. Ces groupes de dépêches sont fabriqués pour éviter de surcharger l'utilisateur avec des dépêches redondantes. La redondance des dépêches est due au processus de rédaction, car les journalistes rédigent en premier lieu une dépêche réduite à un titre, puis ajoutent un premier paragraphe, ensuite vient la dépêche complète éventuellement suivie d'additifs ou de rectificatifs.

Ce regroupement des dépêches est effectué grâce à un calcul de similitude fondé sur l'approche vectorielle.

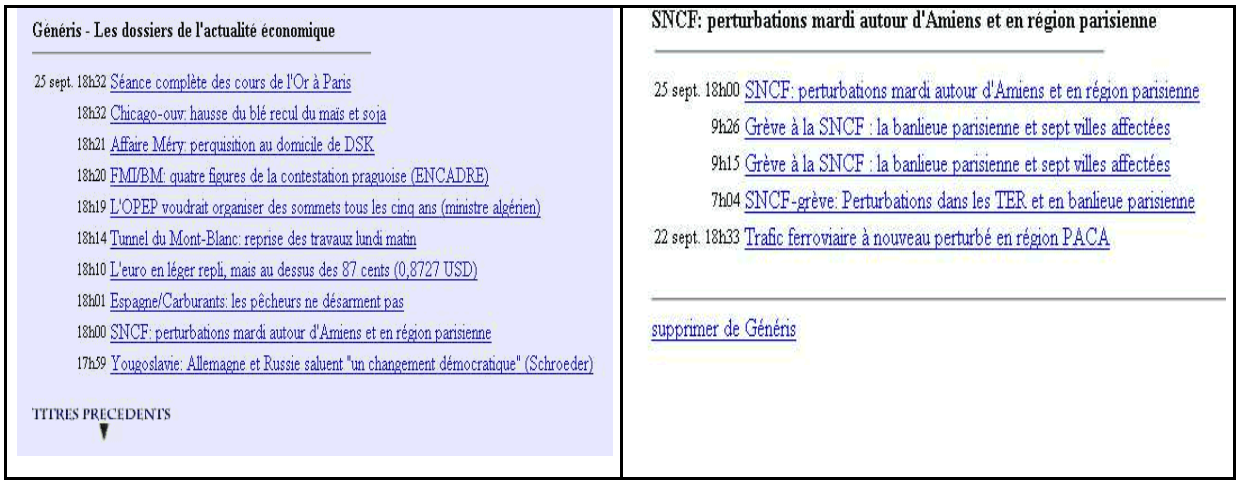


Figure 9.3 : *Exemple de dépêches de l'actualité.*

9.1.2 La technologie des filtres d'ExoWeb

Les filtres utilisés dans ExoWeb sont construits grâce à un ensemble de règles complexes développées par un administrateur. Une description détaillée de la mise en œuvre de ces règles

peut être trouvée dans [Landau *et al.*, 1993] et [Vichot *et al.*, 1997]. Un module de reconnaissance des noms propres utilisant à la fois le contexte global et local a également été développé [Wolinski *et al.*, 1995].

Cet ensemble de règles définit ce qu'est et ce que n'est pas un document pertinent. Chaque nouveau thème nécessite un ensemble de règles distinctes qui doivent être ajustées en fonction du corpus utilisé.

9.1.3 Implémentation : le système TalLab

L'ensemble des applications est inséré dans une architecture appelée TalLab qui est une architecture multi-agent pour le développement d'applications de contenu en ligne. Une description complète de cette architecture et de ses différents composants peut être trouvée dans [Wolinski *et al.*, 1998] et [Wolinski et Vichot, 2001].

Le schéma général d'un agent est repris à la Figure 9.4 :

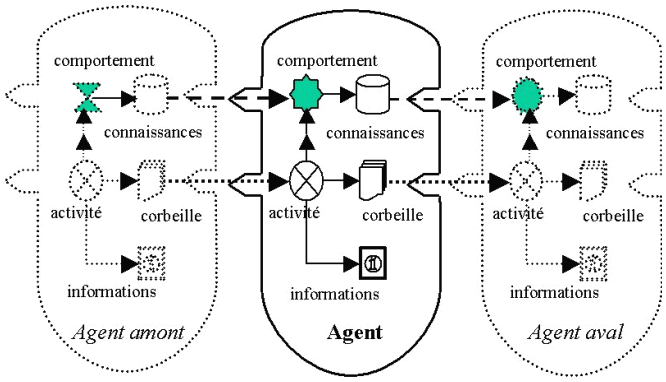


Figure 9.4 : *Modèle de l'agent dans TalLab.*

Le *comportement* d'un agent représente son savoir-faire. Pour chaque document, le comportement élabore des connaissances à partir des informations déjà produites par les agents situés en amont. Par exemple, à partir d'un dictionnaire fréquentiel alimenté par un agent avec les termes utilisés dans chaque document, un second agent peut effectuer la sélection de descripteurs décrite au chapitre 5.

La *corbeille* d'un agent reçoit les identifiants des documents. La présence d'un identifiant dans une corbeille signifie que l'agent a terminé de traiter ce document et que le document peut être traité par un agent situé en aval.

L'*activité* d'un agent scrute en permanence la corbeille d'un agent situé en amont. Pour chaque identifiant, l'activité ordonne au comportement de traiter le document. Une fois le document

traité, l'activité place son identifiant dans la corbeille de l'agent et le supprime de la corbeille de l'agent en amont.

La *persistance* d'un agent lui permet de stocker toutes les informations relatives à son exécution (par exemple nom de la machine sur laquelle il est implanté, le numéro de processus, l'identifiant du document en cours de traitement, le nombre de tentatives de traitement du document en cours, les connaissances qu'il produit, le contenu de sa corbeille). La persistance des connaissances qu'il produit peut prendre la forme de fichiers ASCII, de bases de données standards ou de bases de connaissances spécialisées.

Cette architecture permet d'intégrer aisément des modules externes comme une fonction d'apprentissage pour les réseaux de neurones.

9.2 Les agents neuronaux

9.2.1 Agent apprentissage

À partir des différentes expériences décrites dans ce mémoire, une chaîne d'apprentissage est intégrée dans l'application afin de constituer un agent d'apprentissage. Les différentes étapes de cette chaîne sont représentées à la Figure 9.5.

Le démarrage de cette chaîne suppose l'existence d'une base de dépêches étiquetées pour servir de base d'apprentissage. On verra dans la suite de ce chapitre que la façon dont est fabriquée cette base conditionne l'utilisation de cette chaîne.

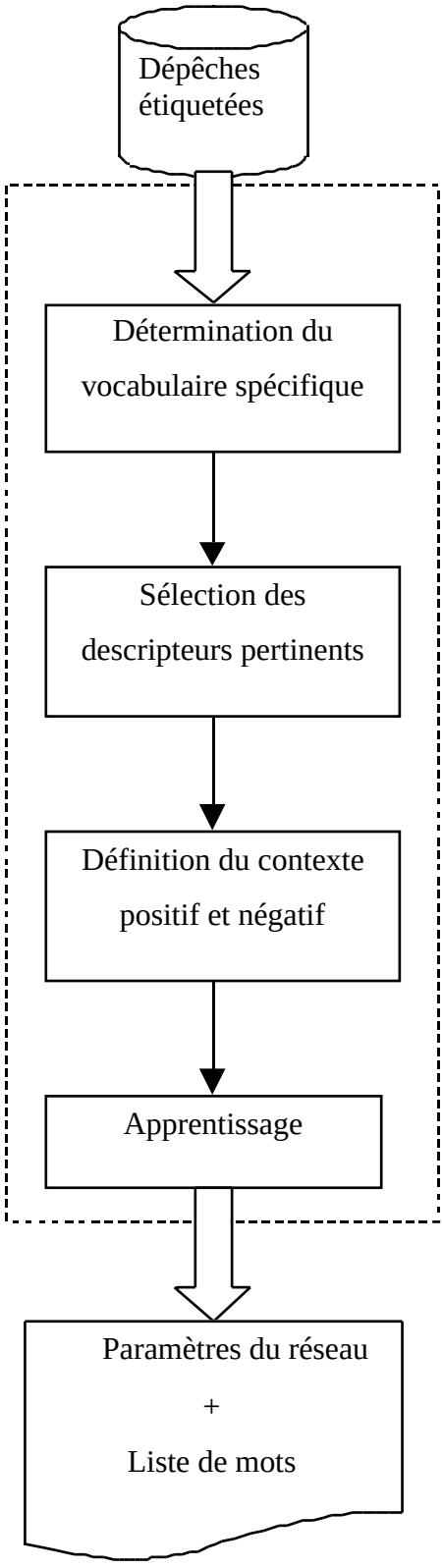


Figure 9.5 : *Différentes étapes de l'agent d'apprentissage.*

La première étape consiste à déterminer le vocabulaire spécifique des dépêches pertinentes grâce à la méthode exposée au chapitre 5. La sélection de descripteurs est ensuite effectuée grâce à l'algorithme d'orthogonalisation de Gram-Schmidt avec l'utilisation d'un vecteur aléatoire comme critère d'arrêt.

Ensuite, grâce à la méthode exposée au chapitre 8, les contextes positif et négatif des mots sélectionnés sont déterminés. À partir du nombre de contextes de chaque mot, l'architecture du réseau de neurones est définie, et l'apprentissage est effectué avec une méthode de régularisation.

Une fois toutes ces étapes terminées, une liste de mots avec leur contexte ainsi que les poids du réseau sont disponibles pour être utilisé par un autre agent.

9.2.1.1 Exemple de thème : argent sale

Un exemple est mis en œuvre afin d'illustrer les différentes étapes décrites ci-dessus.

Pour cet exemple, on dispose de 100 dépêches traitant de la problématique "argent sale" et de 500 dépêches non pertinentes pour ce thème.

Détermination du vocabulaire spécifique

Grâce au calcul de la fréquence d'apparition de chaque mot sur l'ensemble des archives des dépêches de l'AFP, le vocabulaire spécifique des dépêches pertinentes est déterminé. La Figure 9.6 montre les dix premiers mots trouvés (le mot gafi vient de GAFI qui est l'acronyme pour le Groupe d'Action Financière Internationale sur le blanchiment de capitaux).

```
sale
blanchiment
argent
lutte
paradis
capitaux
liechtenstein
liste
gafi
fiscaux
```

Figure 9.6 : *Dix premiers mots trouvés par la méthode du vocabulaire spécifique.*

Sélection des descripteurs

Les cent premiers mots de cette liste sont ensuite utilisés pour construire la matrice d'entrée de l'algorithme d'orthogonalisation de Gram-Schmidt. Cette étape retourne une liste de mots discriminants, le critère d'arrêt détermine automatiquement le nombre de mots retenus. Les dix premiers descripteurs sélectionnés sont représentés à la Figure 9.7.

```
sale
fiscalité
blanchiment
territoires
régulation
principauté
transactions
gafi
paradis
fiscal
```

Figure 9.7 : *Mots sélectionnés par la méthode de Gram-Schmidt.*

Ajout du contexte positif et négatif

Pour chacun des descripteurs sélectionnés précédemment, le contexte positif et le contexte négatif sont déterminés. Le contexte positif est pris en considération s'il apparaît dans au moins cinq documents différents et le contexte négatif dans dix documents différents ; certains mots peuvent donc n'avoir aucun contexte associé. La Figure 9.8 montre la liste des contextes trouvés pour chaque descripteur de la Figure 9.7.

```
sale
      argent
      blanchiment
      blanchir
      paradis
      pratiques
      combattre
fiscalité
      stock
      options
blanchiment
      argent
      sale
      lutte
      capitaux
      propice
      lutter
      paradis
      fiscaux
      anti
      lieu
territoires
      transactions
régulation
principauté
      territoire
transactions
      territoires
      financières
gafi
      capitaux
      blanchiment
      liste
      regroupe
      pays
paradis
      fiscaux
      fiscal
      blanchiment
      dangereux
      argent
      liste
      sale
fiscal
      paradis
```

Figure 9.8 : *Liste des mots avec leur contexte.*

Apprentissage

La liste de mots et leurs contextes définissent l'architecture du réseau de neurones utilisé, puisque le nombre de mots sélectionnés par la procédure de Gram-Schmidt détermine le nombre de neurones cachés de l'architecture ; le nombre de mots de contexte associés à chaque mot détermine le nombre de connexions d'un neurone caché. La Figure 9.9 est un exemple des connexions liant l'un des neurones cachés de l'architecture.

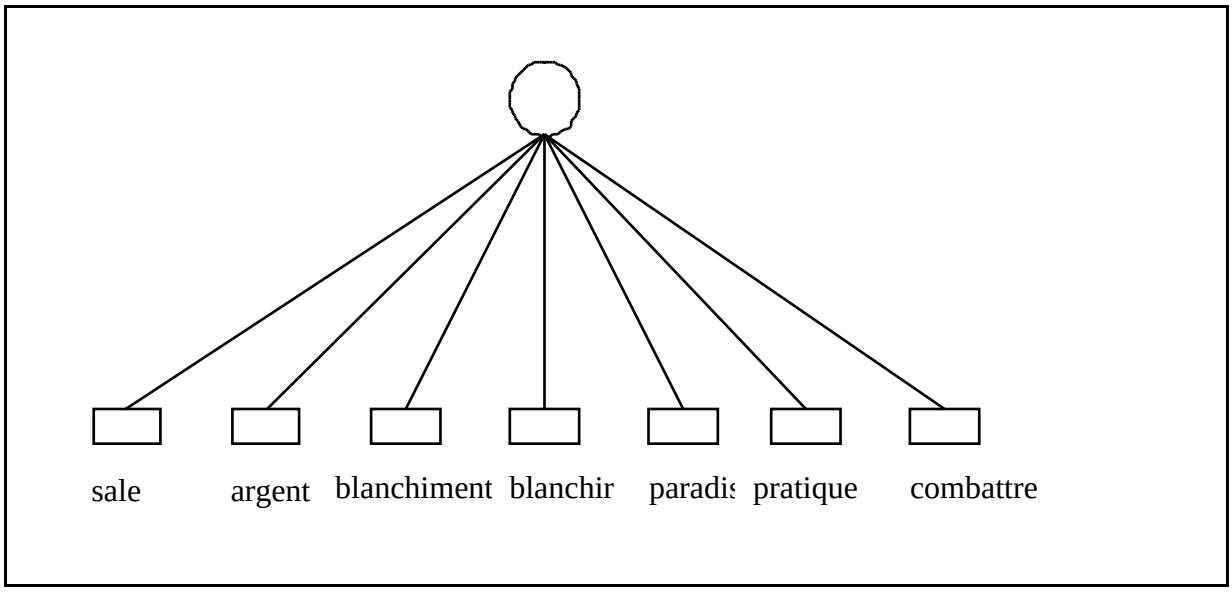


Figure 9.9 : *Connexions reliant un neurone caché pour le mot principal sale (le biais n'est pas représenté).*

L'apprentissage est effectué en utilisant la base étiquetée initiale, les hyperparamètres sont fixés *a priori* selon les valeurs du chapitre 8.

L'agent d'apprentissage définit, pour les agents suivants, une liste de mots avec leur contexte qui va être utilisée pour transformer les textes en vecteur et qui définit également l'arhitecture du réseau de neurones et l'ensemble des valeurs des poids du réseau.

9.2.2 Agent de calcul de probabilité de pertinence

Les résultats de l'étape précédente sont ensuite utilisés par un agent qui calcule un score relativement à un thème donné (assimilé à une probabilité de pertinence) pour chaque dépêche du flux. Les différentes étapes de cet agent sont détaillées à la Figure 9.10.

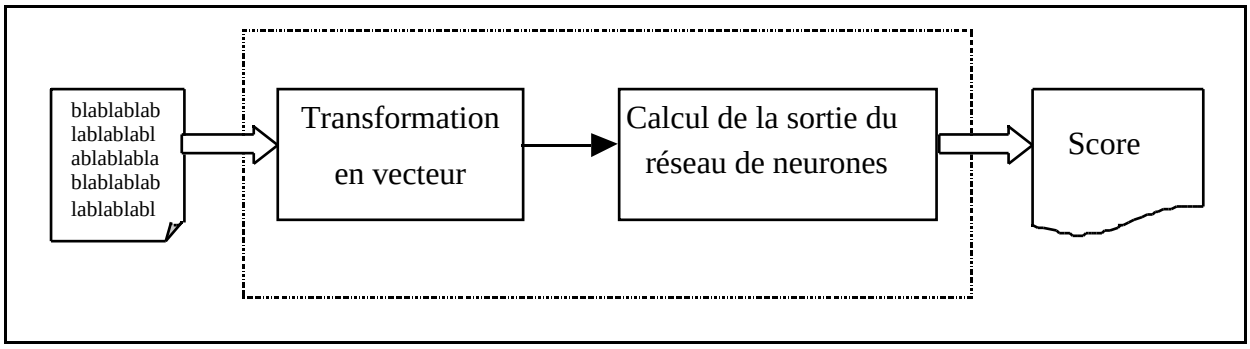


Figure 9.10 : *Étapes de l'attribution d'un score pour chaque dépêche.*

Chaque dépêche est transformée en vecteur selon la liste de descripteurs fournie par l'agent d'apprentissage ; le codage utilisé est le codage Lnu présenté au chapitre 5. Ce vecteur est présenté en entrée du réseau de neurones dont les poids ont également été calculés par l'agent d'apprentissage. Cet agent délivre donc, pour chaque dépêche du flux et pour chaque thème traité, un score assimilé à une probabilité de pertinence.

9.3 Filtres de contrôle

La première application décrite ci-dessous consiste à contrôler les filtres à base de règles grâce à l'utilisation de filtres construits avec les réseaux de neurones. Pour effectuer ce contrôle, une "copie" du filtre à base de règles est faite avec l'approche neuronale, et les différences de comportement entre les deux filtres sont exploitées.

Cette application a été présentée dans [Wolinski *et al.*, 2000].

9.3.1 Problématique : obsolescence des filtres

Lorsque l'on observe le comportement de certains filtres sur une période de temps suffisamment longue, il arrive que l'on constate une dégradation des performances avec le temps. Cette dégradation est perçue, plus qu'elle n'est mesurée, et l'administrateur a besoin d'outils pour percevoir cette dégradation et maintenir la qualité des filtres.

Cette usure des filtres dans le temps survient du fait de l'évolution naturelle de l'environnement et de la langue. Dans le domaine économique, cette évolution peut être très rapide, car de nouveaux domaines naissent régulièrement et les noms propres évoluent sans cesse au gré de la vie des sociétés.

On peut schématiser cette évolution du vocabulaire en considérant que certains termes deviennent polysémiques et d'autres deviennent polymorphes ; ces deux notions sont expliquées dans les deux paragraphes suivants.

9.3.1.1 Polysémie

La polysémie est la faculté d'un terme à représenter des concepts différents. Certains termes deviennent polysémiques avec le temps. Les noms propres sont évidemment sujets à la polysémie ; par exemple *Saint-Louis* est le nom d'une entreprise française, le nom d'une ville américaine, un roi de France, un nom de rue. Les acronymes se prêtent également beaucoup à la polysémie ; par exemple *CDC* signifie Caisse des Dépôts et Consignations, Center of Disease Control, China Development Corp.
Ces différents sens sont distingués par des méthodes de désambiguïsation ; si la phrase "le maire de Saint-Louis" est utilisée, il est probable que *Saint-Louis* se réfère à la ville.

Cependant, si certains sens n'existent pas au moment de la création des règles de filtrage, les méthodes de désambiguïsation sont inefficaces et le concept est mal compris par le système qui risque de sélectionner à tort une dépêche : la nouvelle polysémie va être la cause d'une perte de précision
Par exemple, l'un des filtres proposés par ExoWeb sélectionne les dépêches relatives aux collaborateurs importants de la Caisse des Dépôts dont l'un des membres, l'économiste renommé Patrick Artus, apparaît régulièrement dans la presse. N'ayant pas d'homonyme connu, la simple présence de la chaîne de caractère *Artus* suffit à considérer une dépêche pertinente pour ce thème, jusqu'au jour où une association pour la défense de l'ours des Pyrénées s'est elle-même baptisée du nom d'Artus. La chaîne de caractère *Artus* est alors devenue polysémique et le filtre est devenu moins précis.

9.3.1.2 Polymorphisme

Le polymorphisme est le fait qu'un concept puisse être désigné par des termes différents. L'évolution du langage dans le temps peut entraîner une variation dans les termes désignant un concept.

Les noms propres sont également sujets au polymorphisme, mais les causes en sont différentes. Cela peut être à cause d'erreurs orthographiques (*Eltsine, Elstine*), ou de l'utilisation d'abréviations (*Société Générale, SocGen*), ou de traductions différentes (*Pékin, Beijiing*), de changement de noms (*Compagnie Générale des Eaux* qui devient *Vivendi*) ou de métaphore (*IBM* ou *le groupe d'Armonk*).

L'évolution des termes désignant un concept diminue l'efficacité d'un filtre puisqu'il n'est plus à même de le détecter.

9.3.1.3 Conclusion

Ces exemples montrent que, quel que soit le soin apporté à la conception des règles de filtrage, l'évolution de la langue entraîne une diminution des performances. La polysémie implique une perte de la précision tandis que le polymorphisme implique une diminution du rappel ; ces deux phénomènes peuvent bien sûr se conjuguer et rendre l'analyse plus complexe.

Lorsque cette dégradation est observée et identifiée, l'administrateur peut modifier ou ajouter une règle pour prendre en considération cette évolution. Cependant deux problèmes se posent : d'une part cette dégradation est très lente et ne concerne qu'un faible nombre de dépêches, et, d'autre part, la grande quantité de filtres et de dépêches rend impossible une surveillance exhaustive des filtres.

Il est néanmoins nécessaire de détecter ces baisses de performances pour que le service proposé aux utilisateurs ne se dégrade pas.

9.3.2 Création d'un filtre de contrôle avec les réseaux de neurones

Les stratégies d'apprentissage présentées aux chapitres précédents reposent sur la présence simultanée de plusieurs mots-clefs et de leurs contextes : l'absence d'un mot particulier peut être compensée par la présence d'autres mots. Par rapport aux systèmes à base de règles, ces systèmes apprennent la terminologie d'un concept plutôt que le concept lui-même.

Nous montrons ci-dessous comment créer un "filtre de contrôle" qui détecte la baisse de performances des systèmes à base de règles.

Si l'on considère un thème *T* de l'application ExoWeb et son filtre à base de règles *F*, il est possible d'obtenir facilement un ensemble de dépêches indexées comme pertinentes pour ce thème grâce à une simple requête (cf. le paragraphe 9.1.1.2). Un ensemble de documents étiquetés comme non pertinents est constitué en sélectionnant aléatoirement dans l'archive des dépêches AFP des documents non sélectionnés par le filtre *F*.

Ces deux ensembles forment une base de dépêches étiquetées qui peut être utilisée par l'agent d'apprentissage décrit à la Figure 9.5. Grâce aux résultats de cet agent, l'agent de la Figure 9.10 peut calculer un score de pertinence pour chaque dépêche du flux et pour chaque thème *T*.

Un nouveau filtre *F'* a donc été créé, qui est une copie du filtre *F* puisqu'il a appris à reconnaître les dépêches pertinentes pour *F*.

Le filtre *F'* est appelé **filtre de contrôle**.

9.3.3 Utilisation du filtre de contrôle

Le score calculé par le filtre de contrôle est traité conjointement avec la sortie du filtre *F* pour exploiter les divergences entre ces deux filtres.

Dans la suite de ce paragraphe, la même notation est utilisée pour désigner à la fois un filtre et sa sortie : *F* désigne le filtre à base de règles et sa sortie binaire, *F'* désigne le filtre de contrôle et sa sortie, qui est un score entre 0 et 1.

On définit deux seuils, appelés S^+ et S^-.

Condition 1 : Silence de *F*

$F = 0$ et $F' > S^+$ (typiquement $S^+ = 0.8$)

Ce cas peut correspondre à une baisse du rappel pour le filtre *F*, donc à une détection de polymorphisme. En effet, l'apparition d'un nouveau terme peut avoir induit en erreur le filtre *F* ; en revanche, si ce terme est utilisé dans un contexte global connu, le filtre *F'* est moins susceptible de commettre l'erreur.

Condition 2 : Bruit de *F*

$F = 1$ et $F' < S^-$ (typiquement $S^- = 0.2$)

Ce cas peut correspondre à une baisse de la précision du filtre *F* et donc à une détection de polysémie. Dans ce cas, un terme censé représenter un concept a été utilisé avec un contexte

différent de son contexte habituel. Cette différence de contexte peut se faire au niveau local ou global, mais il est probable qu'elle entraîne un score faible pour le filtre de contrôle.

9.3.4 Implémentation

À partir des différents éléments décrits précédemment, il est possible de créer un filtre de contrôle pour chaque filtre existant.

L'apprentissage du filtre de contrôle est effectué selon les étapes de la Figure 9.11, où l'agent d'apprentissage a été décrit à la Figure 9.5.

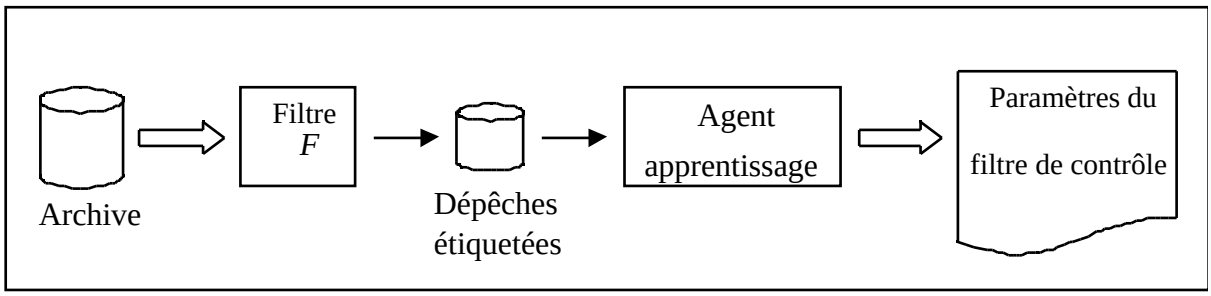


Figure 9.11 : *Apprentissage du filtre de contrôle.*

Après cet apprentissage, les paramètres du filtre de contrôle peuvent être utilisés selon le principe de la Figure 9.12. Par conséquent, pour un thème donné *T*, deux nouveaux thèmes sont créés, appelés *T_silence* et *T_bruit*, dans lesquels apparaissent respectivement les silences et les bruits supposés du filtre initial *F*.

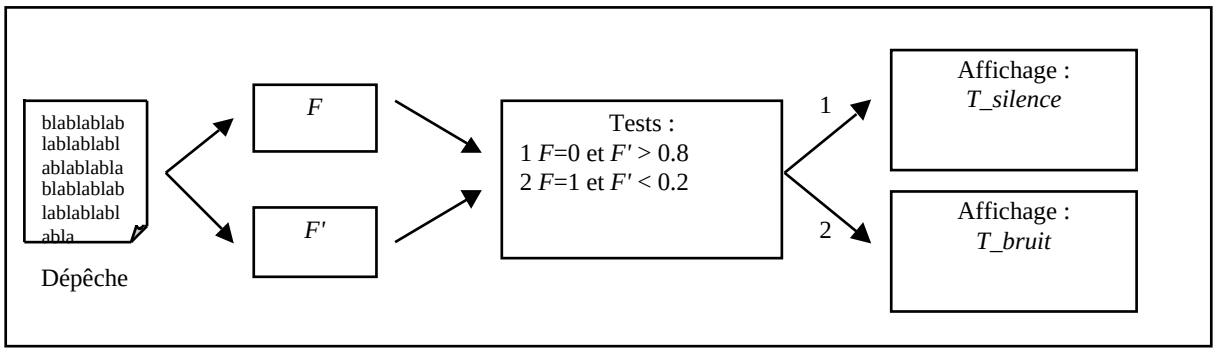


Figure 9.12 : *Procédure d'affichage pour le filtre de contrôle.*

9.3.5 Exemples

Cette application a été mise en œuvre pour différents thèmes déjà existants. On distingue deux types de thèmes : les thèmes à évolution rapide (par exemple le thème *inforoute* présenté au

paragraphe suivant) et les thèmes à évolution lente (par exemple le thème *caisse des dépôts*). Les premiers donnent des résultats très rapidement, alors que pour les seconds, les filtres de contrôle sont peu actifs.

Par exemple, le thème *inforoute* qui s'intéresse à l'Internet et aux nouvelles technologies est un thème à évolution rapide pour lequel de nouveaux mots apparaissent régulièrement ; le filtre correspondant doit être mis à jour afin que le rappel ne diminue pas significativement.
La Figure 9.13 montre une série de titres de dépêches considérées comme des silences du thème *inforoute* par le filtre de contrôle (avec les notations précédentes, $F = 0$ et $F' > 0.8$).

- **Cybercriminalité** : l'industrie fixe des limites aux pouvoirs publics.
- Vêtements "communicants" : le **I-Wear** débarque dans le monde de la mode.
- Tout n'est pas rose pour les "**dotcoms**" du commerce en ligne.
- 58 % des Français "pas choqués" par fortunes rapides de la "**net-économie**".
- L'Allemagne, numéro un de la **nétéconomie** en Europe, selon une étude.
- Croissance spectaculaire de l'"**e-publicité**" en France en 1999.
- Après l'euphorie, le doute s'installe pour **les sites de commerce en ligne**.
- Accord de partenariat entre **Réservoir Net** et Microsoft.

Figure 9.13 : *Titres de dépêches sélectionnées par le filtre de contrôle comme silence du thème inforoute.*

Ces exemples montrent l'émergence de nouveaux mots qui n'existaient pas lors de la création des règles initiales, et que l'on a fait apparaître en gras. Si le cas du mot *I-Wear* semble anecdotique, les autres mots (ou expressions) comme *cybercriminalité, dotcoms, net-économie, nétéconomie, e-publicité, les sites de commerce en ligne* doivent être intégrés dans de nouvelles règles.
Il faut noter dans cet exemple les deux orthographes différentes *net-économie* et *nétéconomie* qui doivent toutes deux être prises en considération, et qui prouvent que de nouveaux concepts peuvent avoir des orthographes qui ne sont pas encore très bien définies : les filtres de contrôle permettent de détecter ces variations.

Le dernier exemple, utilisant *Réservoir Net* est un exemple de nouvelle société liée à la nouvelle économie, qui doit être ajoutée à la liste des sociétés reconnues par le système.

Ces filtres de contrôle peuvent également être utilisés pour mettre en évidence la polysémie de certains termes. La Figure 9.14 est un exemple de bruit détecté par le filtre de contrôle pour le thème *participation*. Dans cette dépêche à propos de la suppression de la vignette automobile, les termes *Mercedes* et *Renault* se réfèrent bien évidemment aux modèles de voitures et non aux sociétés elles-mêmes ; il s'agit de deux termes polysémiques.
Or dans le système à base de règles, le mot *Twingo* est reconnu et désambiguïse *Renault*, mais il n'est pas prévu de règle pour désambiguïser le terme *Mercedes* avec l'expression *Classe S 500 ;* le terme *Mercedes* est confondu avec la société. Le système reconnaît dans une phrase le concept "propriétaire d'une société" et considère cette dépêche comme pertinente. En sélectionnant cette dépêche comme bruit potentiel du système à base de règles, le filtre de contrôle permet à l'administrateur de corriger le filtre pour introduire une désambiguïsation du terme *Mercedes.*
De même, si un nouveau modèle de voiture apparaît chez Renault, la désambiguïsation ne fonctionnera plus tant qu'une nouvelle règle ne sera pas ajoutée ; le filtre de contrôle mettra en évidence cette évolution et le besoin d'une nouvelle règle.

PARIS, 28 août (AFP) - Le gouvernement s'interroge actuellement sur l'opportunité d'une suppression de la vignette automobile, une hypothèse évoquée dans le cadre de la préparation du plan de baisses d'impôts qui doit être annoncé jeudi, a-t-on appris lundi de source proche du dossier.
(…)
Une suppression de la vignette avantagerait toutefois les automobilistes les plus fortunés. À Paris, *le propriétaire d'une Mercedes Classe S 500 neuve économiserait 12.648 FF par an, tandis que celui d'une Renault Twingo d'un peu plus de cinq ans d'âge n'aurait droit qu'à un allègement de 133 FF*. Cet argument ne manquera pas d'être soulevé par les adversaires d'une telle mesure.
(…)

Figure 9.14 : *Exemple de bruit pour le thème participation détecté par le filtre de contrôle.*

Le filtre de contrôle commet évidemment des erreurs ; la Figure 9.15 est un exemple de dépêche considérée comme un bruit pour le thème *participations* par le filtre de contrôle alors

que le passage en italique montre que cette dépêche est effectivement pertinente. Dans ce cas, l'administrateur ignore simplement ce type de dépêches.

(…)
Microsoft a déjà investi 135 millions de dollars US dans la compagnie basée à Ottawa en *rachetant plusieurs millions d'actions*, précise un communiqué. Corel, qui voit ses ventes décliner, avait annoncé dernièrement une perte de près de 11 millions au troisième trimestre.
(…)

Figure 9.15 : *Exemple de faux bruit sélectionné par le filtre de contrôle.*

9.3.6 Conclusions et remarques

Cette application propose un couplage original des méthodes issues du traitement naturel du langage et des méthodes d'apprentissage statistique. Grâce à cette utilisation conjointe, il est possible de disposer d'une base d'apprentissage de grande taille et très représentative du problème que l'on cherche à apprendre. Les méthodes d'apprentissage opèrent, dans ce cas, dans un contexte très favorable puisqu'il est rare, pour les problèmes de catégorisation de textes, de disposer de beaucoup d'exemples préalablement étiquetés.

Cette application tire entièrement parti des méthodes numériques d'apprentissage. Tout d'abord, toutes les étapes de la création d'un filtre de contrôle sont entièrement automatiques ; l'administrateur "appuie sur un bouton" et dispose de son filtre de contrôle environ un quart d'heure plus tard. L'application tire également parti du fait que la sortie du filtre neuronal n'est pas une réponse binaire, mais un nombre continu entre 0 et 1 qui permet de considérer différents niveaux de certitudes.

Les résultats ne sont pas validés par une évaluation quantitative, car l'utilisateur final de cette application est l'administrateur du système. Or ce dernier ne cherche pas à être averti de tous les soupçons de polymorphismes ou de polysémies, mais plutôt des problèmes significatifs ou récurrents qui peuvent l'aider à améliorer ses propres filtres. Le filtre de contrôle peut également faire des erreurs ; l'administrateur se contente de les ignorer. En pratique, le nombre de dépêches sélectionnées par les filtres de contrôle se révèle être assez faible ; une étude plus longue dans le temps permettra de dire si ce volume augmente et s'il existe une dégradation importante des filtres.

Finalement, grâce à ces filtres de contrôle, le travail de l'administrateur est plus efficace, et il n'est plus averti d'erreurs grossières par les utilisateurs, mais par le système.
De plus, les filtres de contrôle peuvent également être utilisés lors de la conception d'un nouveau filtre. Après avoir conçu un nouveau filtre, il est possible de vérifier sa pertinence avec un filtre de contrôle avant de le rendre disponible pour l'ensemble des utilisateurs.

9.4 Futures recherches : développement de filtres sur mesure

Ce paragraphe présente un axe de recherche qui n'a pas été totalement exploré pendant ces années, et pour lequel différents problèmes restent à résoudre. Néanmoins des débuts de solutions sont proposés et montrent que cette voie est prometteuse.

9.4.1 Création d'un filtre par un utilisateur

Par rapport à l'application ExoWeb existante, une application souhaitable serait d'autoriser la création de nouveaux filtres directement par les utilisateurs. Le système actuel n'autorise la création de nouveaux filtres que par l'administrateur ; or il s'agit d'un travail long et minutieux qui doit être recommencé pour chaque nouveau filtre.
À l'opposé, les filtres fondés sur les méthodes d'apprentissage, présentés tout au long de ce mémoire, présentent le grand avantage d'être entièrement automatiques à partir du moment où l'on dispose d'une base de documents étiquetés grâce à l'agent d'apprentissage de la Figure 9.5.
Actuellement, la seule étape qui n'est pas entièrement automatisée est la création d'une base de documents étiquetés comme pertinents ou non pertinents. Par conséquent, pour qu'un utilisateur puisse créer lui-même un filtre, il faut qu'il puisse constituer facilement une base d'apprentissage.

La création de cette base d'apprentissage doit satisfaire deux contraintes majeures :

1. Le travail de l'utilisateur doit être extrêmement simple et limité : les utilisateurs de la Caisse des Dépôts ne sont ni des informaticiens, ni des spécialistes du langage naturel ; et ils ont, généralement, très peu de temps à consacrer à d'autres tâches que celles de leur métier.
2. Pour que les filtres soient utilisés comme outils de travail, ils doivent satisfaire à deux exigences : d'une part, les utilisateurs doivent avoir une extrême confiance dans

l'information apportée par ces filtres, et, d'autre part, ils ne doivent pas perdre de temps à lire des informations qui ne les intéressent pas. Le rappel et la précision doivent donc être élevés.

Ces deux contraintes ne sont pas simples à satisfaire, car, pour qu'un filtre soit de bonne qualité, il est préférable de disposer d'une base d'apprentissage comportant des exemples en nombre important, et suffisamment représentatifs. Or cette contrainte n'est pas facilement compatible avec l'exigence d'un travail minimum de la part de l'utilisateur ; il n'est pas possible de demander à un utilisateur de catégoriser manuellement des dépêches pendant quelques jours pour obtenir une base d'apprentissage.

Nous proposons deux approches différentes qui permettent à un utilisateur de créer une base de documents étiquetés. Ces deux approches sont illustrées par des exemples concrets de mise en œuvre.

9.4.2 Utilisation d'un moteur de recherche

La première approche consiste à utiliser la base des deux années d'archive de dépêches de l'AFP et un moteur de recherche ; autrement dit, il s'agit d'utiliser l'application d'ExoWeb décrite au paragraphe 9.1.1.2.

Le principe est simple : l'utilisateur effectue une requête booléenne, et reçoit en retour un ensemble de documents pertinents pour cette requête. Cette requête peut être très simple ; si l'utilisateur maîtrise bien l'outil, il lui est possible d'effectuer des requêtes plus compliquées.

Comme l'archive regroupe un nombre important de documents (environ 230000 dépêches à la fin de septembre 2000), il est possible d'obtenir facilement une centaine de documents pertinents (sauf lorsque la requête est vraiment exotique). Un ensemble de documents non pertinents est constitué en sélectionnant aléatoirement des dépêches dans l'archive (tout en excluant celles déjà sélectionnées). Cet ensemble peut évidemment contenir des documents pertinents, mais ils sont en nombre négligeable car l'ensemble des thèmes abordés par les dépêches est extrêmement vaste.

Cette opération permet d'obtenir, avec très peu d'effort de la part de l'utilisateur, une base de dépêches étiquetées de taille suffisante pour l'apprentissage.

9.4.2.1 Réalisation

La réalisation est simple, puisqu'elle ne met en jeu que des agents déjà existants : les différentes étapes sont présentées à la Figure 9.16.

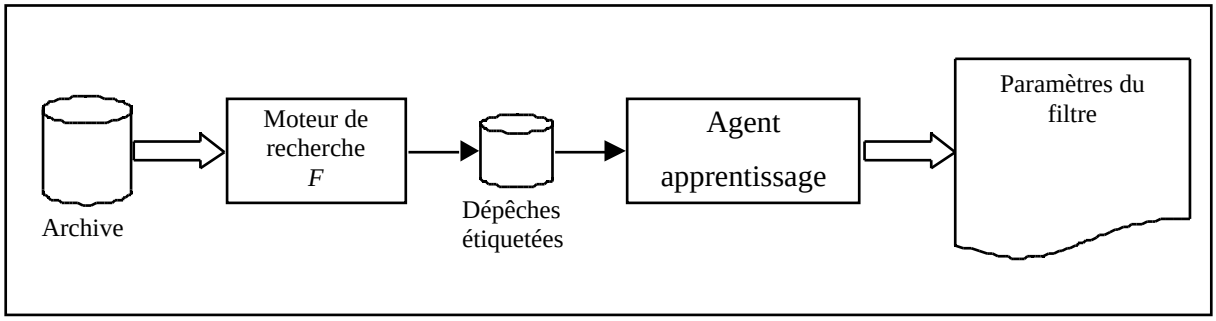


Figure 9.16 : *Création d'un filtre par un utilisateur.*

Lorsque l'utilisateur valide sa requête, il choisit également un nom pour le thème qu'il constitue ; c'est ce nom qui apparaît ensuite dans le catalogue des thèmes, une fois que l'agent d'apprentissage signifie qu'il a terminé son travail.

Pour chaque dépêche du flux, un score est calculé pour ce nouveau thème selon la Figure 9.10 : si ce score est supérieur à un seuil prédéfini, la dépêche est considérée comme pertinente et son titre apparaît sous forme de lien hypertexte.

La détermination du seuil a une influence sur les valeurs de précision et de rappel, et son choix n'est pas trivial ; nous revenons sur ce point au paragraphe 9.4.2.3.

9.4.2.2 Exemples de fabrication automatique de filtres

Trois exemples de filtres construits à partir de requêtes sont présentés dans ce paragraphe, pour montrer l'intérêt et les difficultés liées à cette approche. Les requêtes utilisées sont volontairement très simples afin de se placer dans des cas d'utilisation probables.

Le premier exemple est la fabrication d'un filtre destiné à sélectionner les dépêches qui traitent de l'argent sale, du blanchiment de capitaux, etc. La requête formulée pour fabriquer ce filtre appelé *argent_sale* est simplement :

argent <NEAR> sale

Les résultats de la sélection de descripteurs et de la détermination du contexte ont été présentés au paragraphe 9.2.1.1 et sont repris à la Figure 9.17 (qui est identique à la Figure 9.8).

sale	régulation
argent	principauté
blanchiment	territoire
blanchir	transactions
paradis	territoires
pratiques	financières
combattre	gafi
fiscalité	capitaux
stock	blanchiment
options	liste
blanchiment	regroupe
argent	pays
sale	paradis
lutte	fiscaux
capitaux	fiscal
propice	blanchiment
lutter	dangereux
paradis	argent
fiscaux	liste
anti	sale
lieu	fiscal
territoires	paradis
transactions	

Figure 9.17: *Liste des mots avec leur contexte pour argent_sale.*

Il est intéressant de noter que si l'expression "argent sale" figure dans les descripteurs, de nouvelles expressions qui ont étendu la requête sont apparues, et peuvent rendre un texte pertinent comme l'expression "blanchiment de capitaux dans un paradis fiscal".

La Figure 9.18 montre des exemples de titres de dépêches sélectionnées pour ce thème.

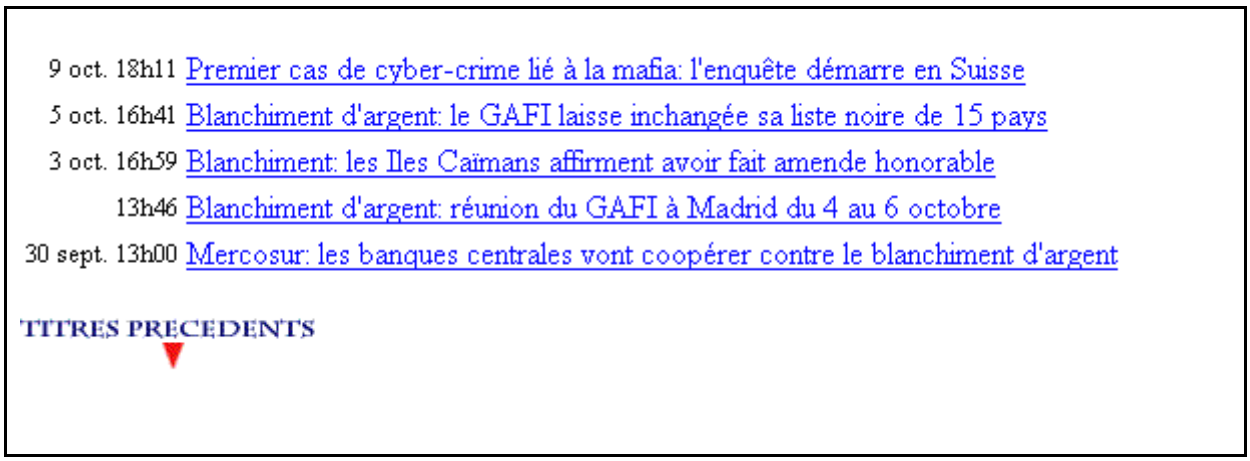


Figure 9.18 : *Dépêches sélectionnées pour le thème argent_sale.*

Un autre de filtre appelé *euro* a été fabriqué de la même manière grâce à la requête :

change <NEAR> euro

Les titres de dépêches sélectionnées à la Figure 9.19, montrent que la requête initiale a été étendue, puisque l'acronyme BCE (Banque Centrale Européenne) fait partie des descripteurs pertinents.

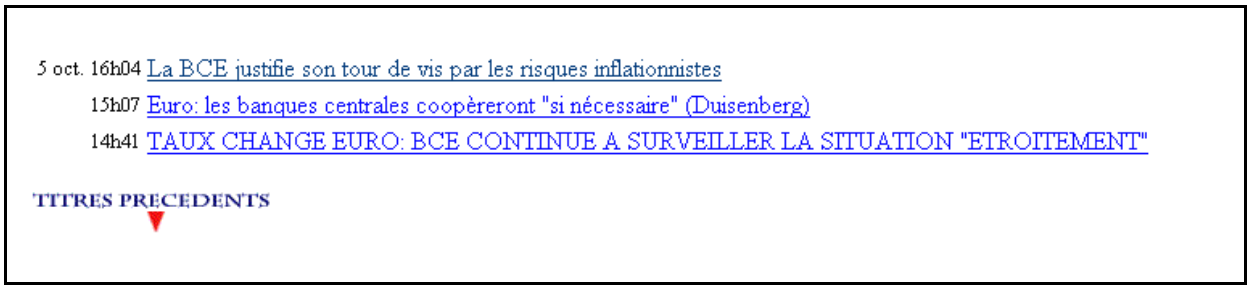


Figure 9.19 : *Dépêches sélectionnées pour le thème euro.*

Enfin, un filtre intitulé *introduction_en_bourse* a été fabriqué grâce à la requête :

introduction <NEAR> bourse

La Figure 9.20, montre là aussi, une extension de la requête avec l'apparition de l'acronyme IPO qui signifie *Initial Public Offering*, souvent utilisé comme une expression synonyme de "introduction en bourse".

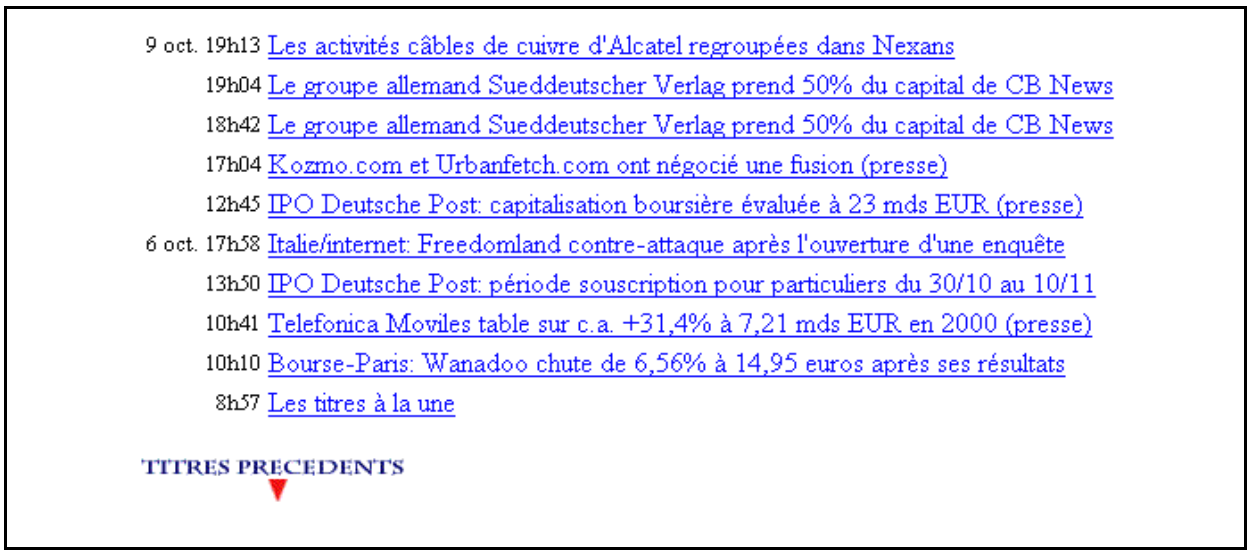


Figure 9.20 : *Dépêches sélectionnées pour le thème introduction_en_bourse.*

9.4.2.3 Détermination du seuil

La constitution de la base d'apprentissage grâce au moteur de recherche selon le modèle des exemples précédents entraîne un biais de la base d'apprentissage, imputable à trois causes :

1. Les exemples pertinents comportent systématiquement les mots-clefs.
2. Le moteur de recherche sélectionne en priorité les dépêches pour lesquelles l'occurrence des mots-clefs est élevée.
3. Le ratio entre le nombre de dépêches pertinentes et le nombre de dépêches non pertinentes dans la base d'apprentissage n'est pas identique à celui du flux réel de dépêches. Autrement dit, les probabilités *a priori* de chaque classe dans la base d'apprentissage sont différentes des probabilités *a priori* dans le flux.

Du fait de ce biais, il est difficile de déterminer une valeur optimale pour le seuil de décision.
Les dépêches du flux qui sont pertinentes ont, en général, des occurrences plus faibles pour les mots clefs que celles observées sur la base d'apprentissage. Le score de ces dépêches est donc relativement bas : il faut choisir un seuil de décision bas si l'on ne veut pas obtenir un rappel très faible.

La différence de probabilité *a priori* des classes entre la base d'apprentissage et le flux réel implique qu'il est nécessaire d'adapter le seuil optimal avec la formule de Bayes [Stricker *et al.*, 2000a]. Cette adaptation ne peut se faire que si la véritable probabilité *a priori* des dépêches pertinentes est connue, ce qui n'est pas nécessairement évident.
Il n'est pas possible d'utiliser une méthode de validation croisée ou de *leave-one-out* pour déterminer un seuil optimal, puisque toute base construite à partir des documents étiquetés comporte le même biais que la base d'apprentissage.

En définitive, nous avons fixé un seuil arbitraire pour les exemples du paragraphe précédent, mais il serait souhaitable de trouver une solution plus satisfaisante à ce problème.

9.4.2.4 Conclusion

Les exemples ont montré que, même avec des requêtes extrêmement simples, il était possible de construire des filtres qui ne se contentent pas de chercher les mots-clefs, et que la méthode

de sélection de descripteurs élargit la requête en considérant de nouveaux concepts. Cependant, cette base d'apprentissage est fortement biaisée car si la requête est simple de type "mot1 ET mot2", tous les documents contiennent les mots de la requête.

Il est nécessaire de résoudre le problème de détermination du seuil de décision pour obtenir une véritable application opérationnelle. Il faut noter qu'il est possible de diminuer le biais des bases d'apprentissage, en considérant des requêtes plus compliquées faisant intervenir un plus grand nombre de mots-clefs.

Nous nous sommes volontairement limités à des expressions très simples, car il nous semble que la plupart des utilisateurs se contenteront de ces requêtes.

Nous allons montrer qu'il est possible de résoudre le problème du biais de la base d'apprentissage en demandant un effort supplémentaire à l'utilisateur.

9.4.3 Utilisation du moteur de recherche et du réseau de neurones

La méthode précédente exigeait un travail minimum de la part de l'utilisateur, mais créait une base d'apprentissage fortement biaisée. Pour limiter ce biais, le nombre de requêtes utilisées est augmenté, tout en permettant l'utilisation de racines lexicales par le moteur de recherche. Grâce à ces deux modifications, l'ensemble des documents pertinents est beaucoup plus hétérogène.

Cependant, le moteur de recherche est plus susceptible de commettre des erreurs car l'une des requêtes peut être moins bien formulée, et les racines induisent également des erreurs.

Par exemple, si l'on cherche à obtenir des documents traitant des accords de coopérations entre entreprise, la requête :

coopération <NEAR> entreprise

sélectionne aussi les documents qui traitent "d'entreprises coopératives".

Pour corriger les erreurs commises par le moteur de recherche, nous avons proposé un processus itératif, à la manière des filtres de contrôle du paragraphe 9.3, qui exploite les différences entre un filtre neuronal et le moteur de recherche pour améliorer l'étiquetage de la base d'apprentissage [Stricker *et al.*, 2000a].

9.4.3.1 Principe de coopération entre un réseau de neurones et un moteur de recherche

Un ensemble de requêtes construit une base d'apprentissage étiquetée, et un filtre est fabriqué grâce à l'agent d'apprentissage. Le score de chaque dépêche de la base d'apprentissage est alors calculé avec les paramètres trouvés.

Des dépêches sont présentées à l'utilisateur pour qu'il confirme ou infirme l'étiquette de ces dépêches dans deux cas :

1. les dépêches étiquetées pertinentes, mais avec un score proche de zéro
2. les dépêches étiquetées non pertinentes avec un score proche de un.

Ces corrections éventuelles modifient donc l'étiquette de certaines dépêches, et un nouvel apprentissage avec une nouvelle base est effectué ; de nouvelles dépêches sont présentées à l'utilisateur selon le même critère. Ce processus est répété, ; comme le système garde la trace des dépêches vues par l'utilisateur, le nombre de nouvelles dépêches à vérifier devient nul après un certain nombre d'itérations. Un dernier apprentissage est effectué pour lequel les paramètres sont conservés, ce qui définit un nouveau filtre.

9.4.3.2 Conclusion

Cette deuxième approche réduit considérablement le biais de la base d'apprentissage par rapport à la première méthode, au prix d'une augmentation, qui peut être jugée excessive, du travail de l'utilisateur. Néanmoins cette approche est intéressante pour fabriquer des bases étiquetées de bonne qualité dans un but de recherche, car elle nettement plus économique qu'un classement entièrement manuel.

Il reste néanmoins à étudier l'impact du changement d'étiquette sur la valeur des paramètres pour limiter à la fois le nombre d'itérations et le nombre de dépêches que l'utilisateur doit étiqueter manuellement.

9.4.4 Autres travaux

La construction d'une base d'apprentissage étiquetée pour l'utilisation d'un algorithme d'apprentissage a été abordée par d'autres auteurs. De plus, la conférence TREC utilise de très grandes bases de documents étiquetés pour un grand nombre de thèmes ; il est donc intéressant de se pencher sur la construction de ces bases.

La conférence TREC aborde cette problématique à travers deux aspects. Tout d'abord pendant les huit premières éditions, les résultats de la tâche *ad hoc* ont fabriqué des bases de documents étiquetés pour les éditions suivantes ; ces bases sont utilisées pour mettre en œuvre des algorithmes d'apprentissage numérique pour la tâche de filtrage. Si ce travail a permis d'obtenir de très grandes bases de documents étiquetées, il n'est tout de même pas utilisable en pratique puisqu'il demande un travail très important de la part des assesseurs qui doivent vérifier manuellement les résultats des différents systèmes participants (pour plus de précision sur la fabrication de ces bases, se référer à [Voorhees et Harman, 2000]).
La sous-tâche de filtrage adaptatif de TREC présentée au chapitre 3 simule l'interaction d'un utilisateur qui n'étiquetterait que certains documents. Cette simulation permet donc de construire une base d'apprentissage au fil du temps et de l'utiliser pour améliorer le filtre. Néanmoins, l'utilisateur est supposé préciser la classe de chaque document sélectionné et ce, pendant une période assez longue dans le temps, ce qui en pratique est inconcevable. De plus, dans la compétition TREC, si un système fonctionne mal pendant un temps et sélectionne un grand nombre de documents, il dispose alors d'un maximum d'exemples d'apprentissage et peut devenir très performant alors qu'en pratique un utilisateur aurait arrêté de l'utiliser.
Dans ces deux exemples de la conférence TREC, la construction des profils initiaux est réalisée grâce à une requête en langue naturelle rédigée avec un titre et une partie narrative (un exemple de requête a été présenté au chapitre 3). Cependant, dans notre cas, on ne dispose pas d'une telle information même pour construire ce profil initial.
Les solutions proposées dans TREC pour résoudre ce problème d'étiquetage ne sont donc pas transposables à notre problématique et à nos contraintes.

D'autres approches, connues sous le nom *active learning*, détectent les exemples dont le classifieur à besoin pour s'améliorer [Cohn *et al.*, 1996]. Dans le cadre du filtrage de documents, cette idée a été appliquée par [McCallum et Nigam, 1998b] afin de demander à un utilisateur de n'étiqueter que les documents dont l'information est utile pour construire le classifieur. Leur approche réduit significativement le nombre de documents étiquetés nécessaires pour obtenir de bonnes performances.

Il nous semble que l'utilisation d'une méthode d'*active learning* couplée à l'utilisation d'un moteur de recherche est une voie intéressante à explorer dans de futurs travaux.

9.5 Conclusion

Nous avons montré dans ce chapitre comment nos travaux ont été intégrés dans un système opérationnel en temps réel, pour développer de nouvelles applications qui n'étaient pas envisageables avec les technologies existantes au sein de la société.

La première application propose une association originale des systèmes à base de règles et des systèmes à base d'apprentissage. Elle permet aux administrateurs d'assurer une qualité de service constante tout en limitant leur travail. L'utilisation d'un filtre à base de règles permet de surmonter l'un des problèmes majeurs auxquels est confrontée l'utilisation des systèmes d'apprentissage numérique pour le filtrage de textes : la constitution automatique d'une base de textes étiquetés pour l'apprentissage.

La deuxième application proposée propose aux utilisateurs un nouveau service : la possibilité de créer eux-mêmes leurs filtres avec un travail limité et dans un temps minimum. Les expériences préliminaires présentées dans cette dernière partie montrent que cette approche est très encourageante et de premiers filtres ont ainsi pu êtres fabriqués de manière entièrement automatique.

Il reste cependant des problèmes à résoudre, liés au biais de la base d'apprentissage, qui doit être limité ; il faut adapter le seuil de décision pour tenir compte de ce biais. Il faut noter qu'il est possible, après un certain temps (par exemple une semaine), d'utiliser, comme base d'apprentissage, les documents sélectionnés par le nouveau filtre et non plus les documents étiquetés automatiquement par le moteur de recherche. Un apprentissage régulier sur des exemples de plus en plus hétérogènes doit permettre d'améliorer les performances du filtre.