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

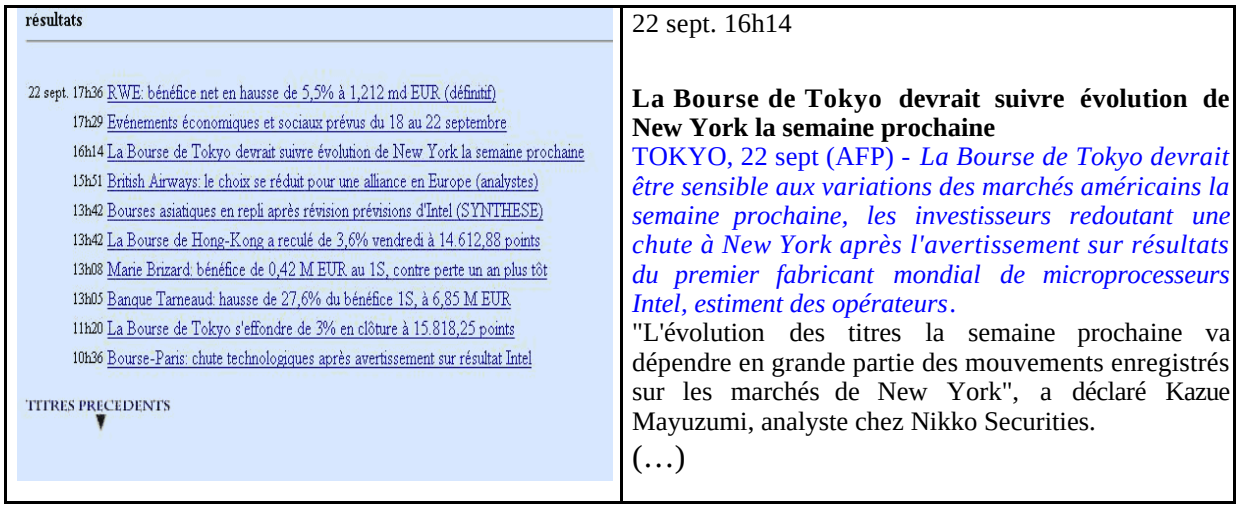
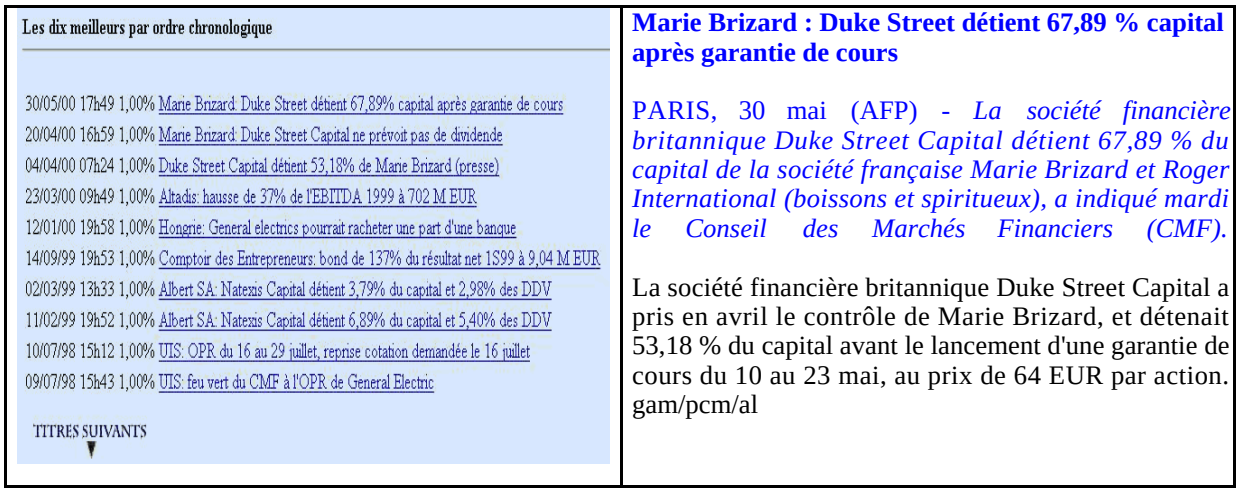


Figure 9.1 : *Dépêches sélectionnées pour le thème résultat.*

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.

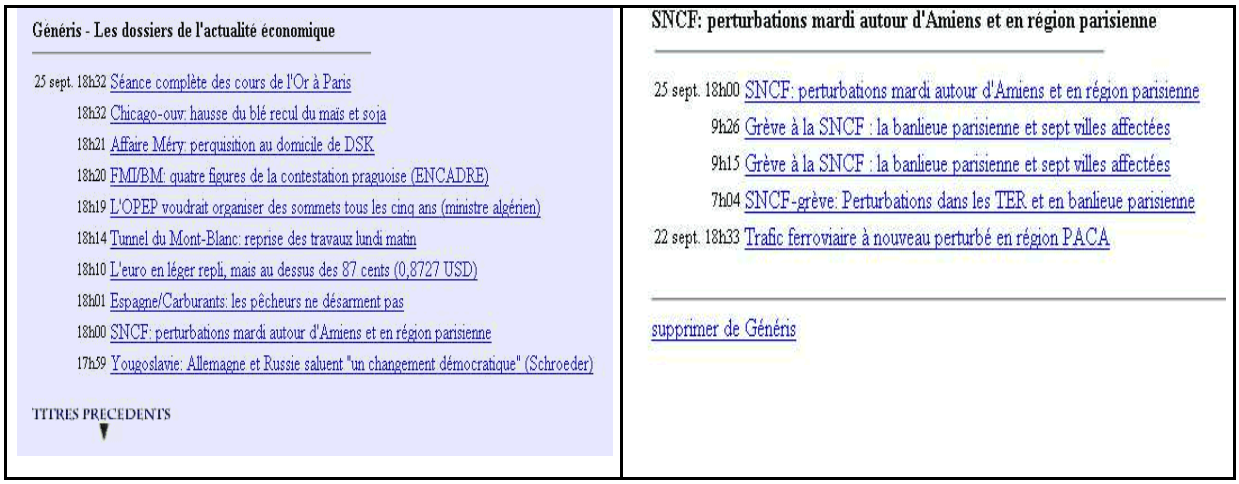


Figure 9.3 : *Exemple de dépêches de l'actualité.*

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 :

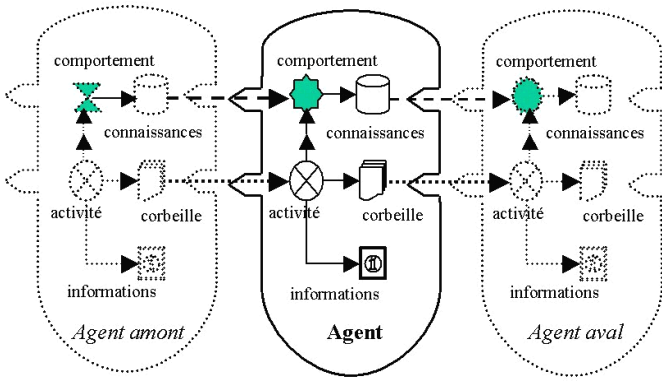


Figure 9.4 : *Modèle de l'agent dans TalLab.*

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.

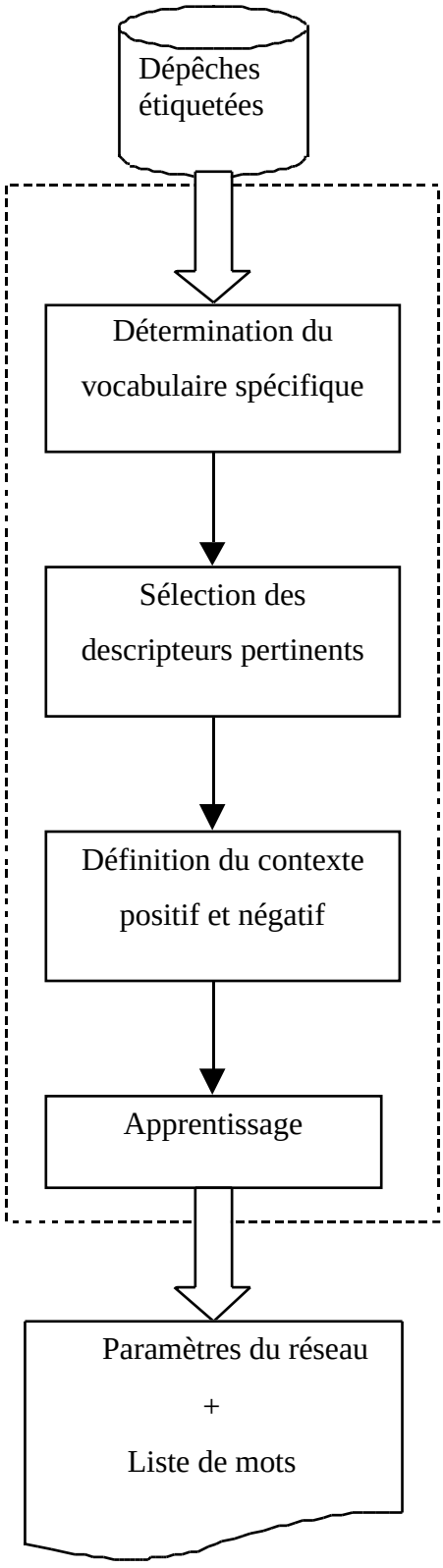


Figure 9.5 : *Différentes étapes de l'agent d'apprentissage.*

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.

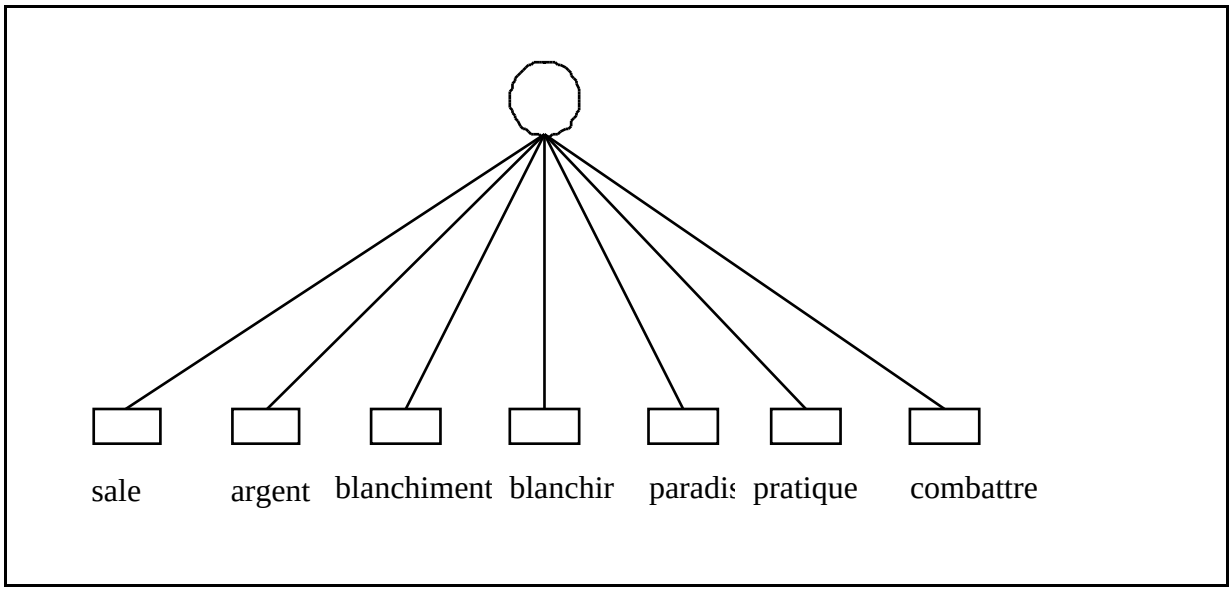


Figure 9.9 : *Connexions reliant un neurone caché pour le mot principal sale (le biais n'est pas représenté).*

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.

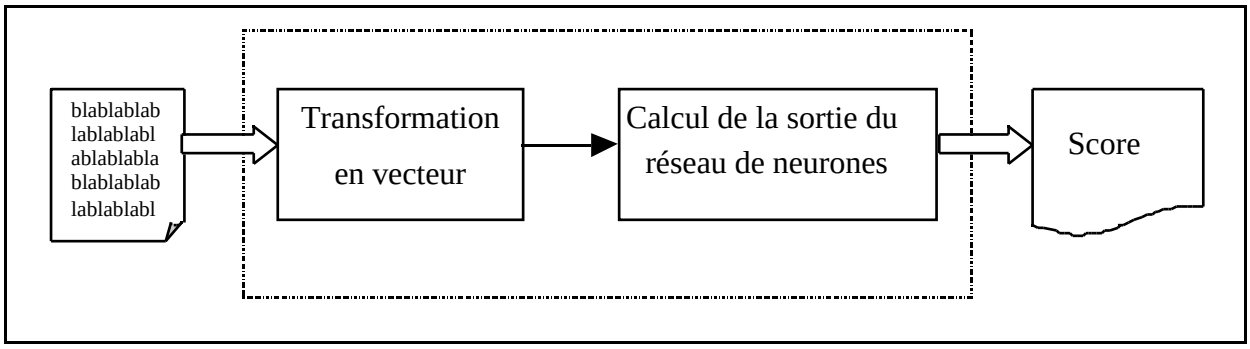


Figure 9.10 : *Étapes de l'attribution d'un score pour chaque dépêche.*

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.

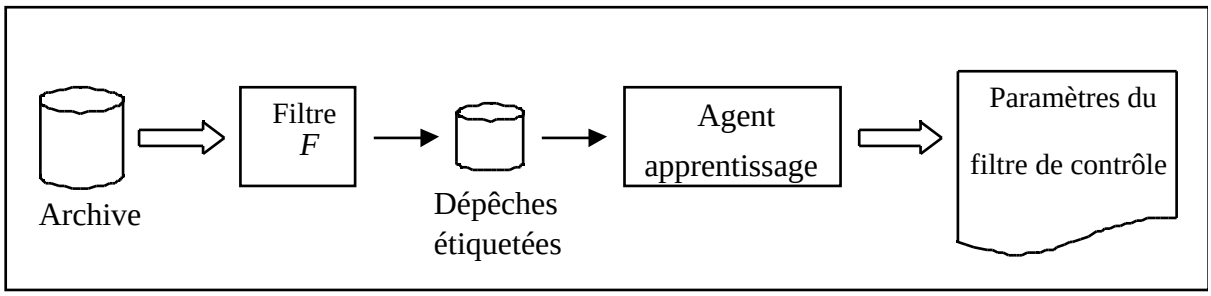


Figure 9.11 : *Apprentissage du filtre de contrôle.*

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*.

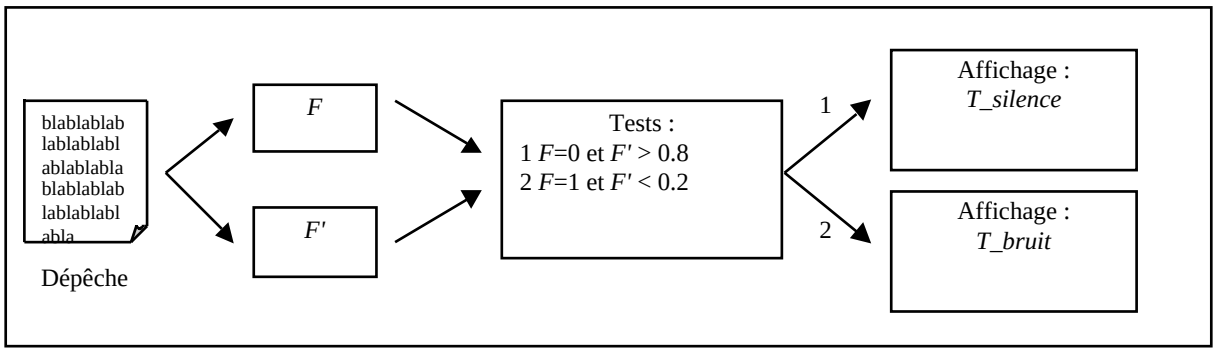


Figure 9.12 : *Procédure d'affichage pour le filtre de contrôle.*

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.

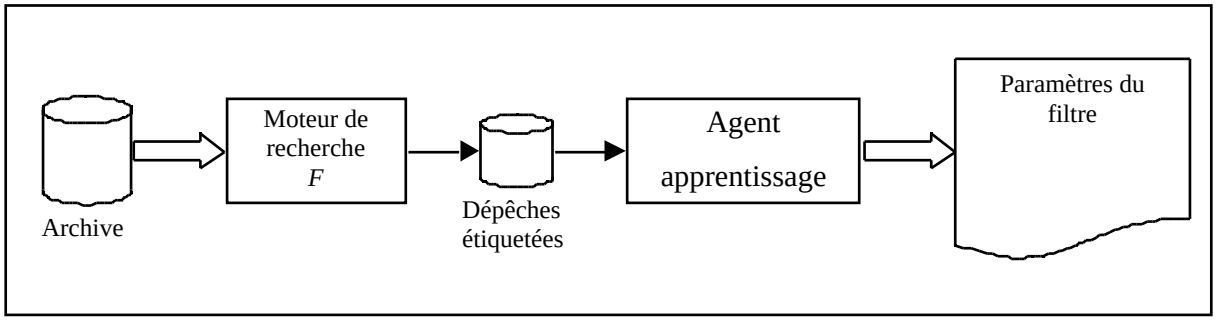


Figure 9.16 : *Création d'un filtre par un utilisateur.*

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.

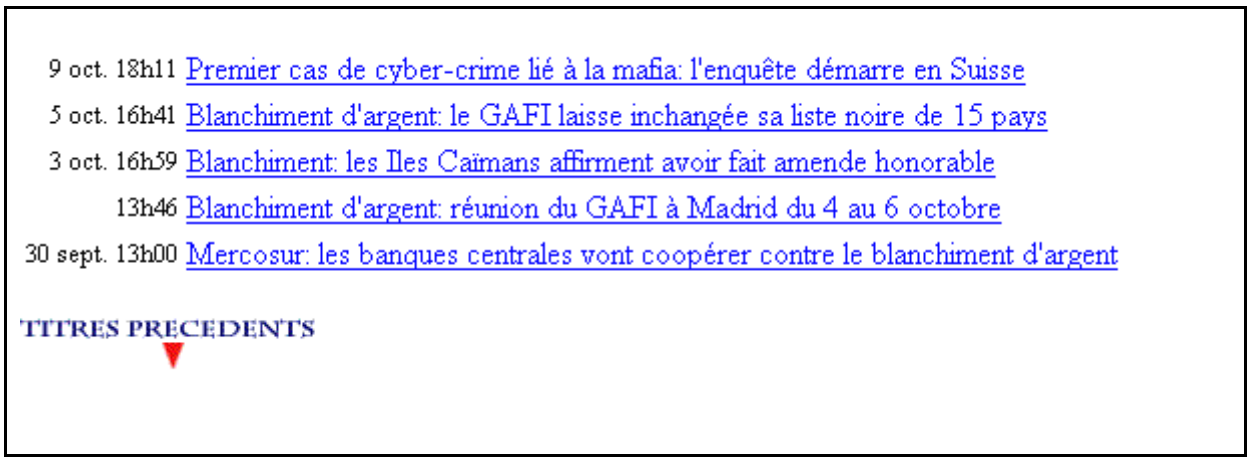


Figure 9.18 : *Dépêches sélectionnées pour le thème argent_sale.*

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.

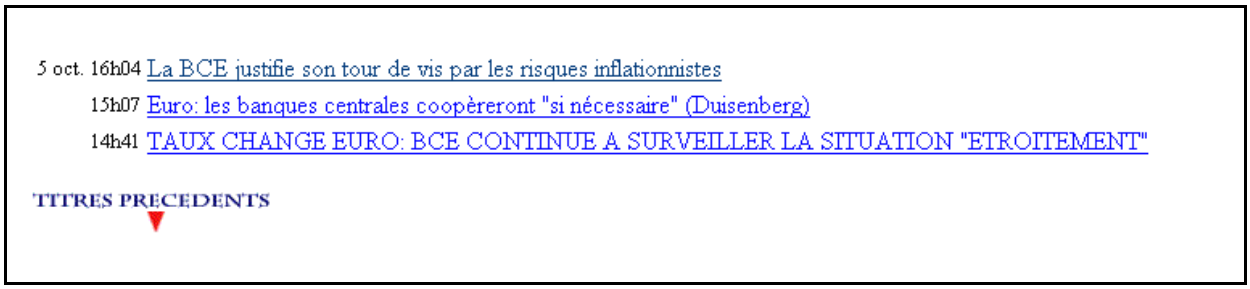


Figure 9.19 : *Dépêches sélectionnées pour le thème euro.*

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".

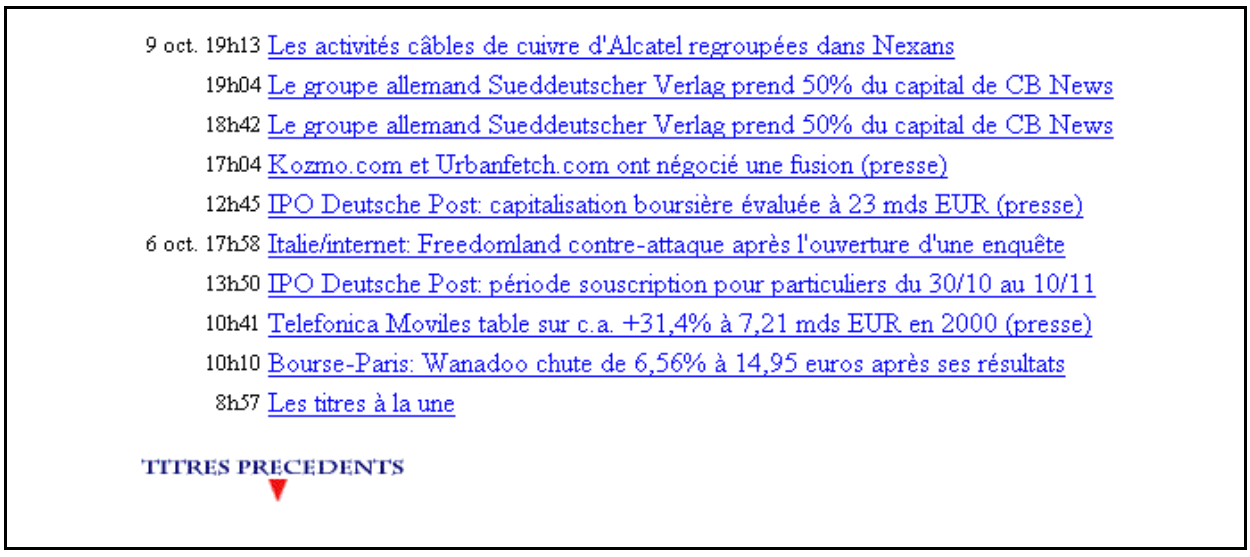


Figure 9.20 : *Dépêches sélectionnées pour le thème introduction_en_bourse.*

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.