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.