APPRENTISSAGE ET DONNEES TEXTUELLES

Patrick Gallinari

Hugo Zaragoza, Massih Amini

LIP6, Université Paris 6, 4 Place Jussieu

75252 Paris cedex 05

(Patrick.Gallinari@lip6.fr)

Résumé

Le déploiement du web incite actuellement plusieurs communautés de l'informatique à travailler sur l'accès à l'information et en particulier à l'information textuelle. La communauté apprentissage s'intéresse depuis quelques années à l'analyse de l'information textuelle en vue d'automatiser les traitements pour une gamme de tâches allant de la recherche à l'extraction de l'information. Nous présentons dans ce texte d'une part les travaux réalisés en recherche et extraction d'information pour introduire de l'apprentissage dans les chaînes de traitement et d'autre part des apports récents de l'apprentissage au domaine du texte.

1. Introduction

Plusieurs communautés travaillent depuis de nombreuses années sur l'information textuelle, linguistes, statisticiens, informaticiens, avec des outils et des objectifs souvent très différents. En informatique, le principal objet d'étude est l'utilisation du langage naturel en vue de la réalisation d'interfaces homme-machine. Ce n'est que beaucoup plus récemment que la communauté apprentissage a tenté d'appliquer ses outils à l'information textuelle, l'arrivée d'internet et l'émergence du gigantesque ensemble de données qui lui est associé a constitué le coup de pouce qui a incité les chercheurs en apprentissage à investir ce domaine. Une communauté a commencé a émerger autour de ce thème et à côtoyer les autres acteurs du texte. Les chercheurs en apprentissage sont très souvent venus au texte en se penchant sur les problèmes soulevés par l'accès à l'information sur le web ou dans les entrepôts de données, par la modélisation utilisateur pour des tâches ou l'information textuelle jouait un rôle important (e.g. navigation). Ils ont eu ainsi à se familiariser en particulier avec les outils développés en recherche d'information. L'apprentissage s'intéresse bien sûr au développement de méthodes automatiques, et autant que possible, indépendantes du domaine et nécessitant un minimum de

traitements linguistiques. Sans nier l'importance des connaissances du domaine et des aspects langue naturelle, le credo de cette communauté est qu'il est possible d'effectuer de nombreuses tâches pratiques par les analyses plus frustres qui sont réalisées en apprentissage. En même temps, la nature des textes est en train de changer. Ceux qui sont présents sur le web ou qui circulent sur internet ont souvent une structure grammaticale réduite au minimum, de nouveaux standards de documents structurés sont apparus (html, xml, ...) et les travaux récents prennent en compte ces structures pour rechercher l'information. Remarquons que d'autres communautés de l'informatique se sont récemment immiscées dans le monde du texte, par exemple les bases de données, le multi-média et les interfaces homme machine (IHM).

Notre but est de présenter les liens entre apprentissage et information textuelle. Nous introduisons tout d'abord brièvement quelques unes des techniques d'apprentissage qui ont été fréquemment utilisées pour traiter du texte. Nous passons ensuite en revue des utilisations de l'apprentissage réalisées par les communautés de la recherche d'information (**RI**) et de l'extraction d'information (**EI**) puis présentons quelques travaux récents issus de l'apprentissage. En faisant ces revues, nous n'avons aucune prétention d'exhaustivité - celle ci serait d'ailleurs difficile à atteindre dans des domaines comme la recherche d'information qui utilisent depuis longtemps des méthodes issues des statistiques et de l'apprentissage- mais nous avons essayé de décrire quelques exemples représentatifs de ce qui s'est fait dans ces domaines. Pour finir, nous présentons des travaux que nous avons mené au LIP6.

Notations

Nous noterons *d* un document, *q* une requête, qui peut être constituée de mots clés ou être elle même un document, *c* la classe d'un document quand cette notion est pertinente, *w* la séquence de termes associée au document et *t* une séquence d'étiquettes associée à *w*. La notion de classe intervient dans le cas de la recherche d'information pour les applications de catégorisation. Un document *d* pourra être codé sous diverses formes, nous appellerons termes les représentations des mots issues du prétraitements ou du codage des mots.

2. Les méthodes de l'apprentissage

En intelligence artificielle, l'apprentissage est représenté par deux courants - numérique et symbolique - qui exploitent respectivement et majoritairement des formalismes statistiques et logiques. C'est principalement l'aspect numérique que nous considérerons ici. Nous faisons ci dessous une brève présentation de quelques unes des techniques d'apprentissage qui ont été employées pour le texte, elles sont souvent bien connues aujourd'hui, et nous renvoyons aux ouvrages de référence pour une présentation détaillée.

Naïve Bayes

Si on représente un objet *x* par l'ensemble de ses composantes x_i et que ceux ci sont supposés indépendants, alors, étant donnée la classe c, on peut écrire

$$p(x / c) = \prod p(x_i / c)$$

où *p(a/b)* est la densité de probabilité conditionnelle associée à *a* et *b*. L'utilisation de la règle de décision de Bayes permet ensuite de classer le document dans la classe *c* telle que $c = \arg\max_c p(c / d)$. En RI, l'espace des attributs -les termes- est discret et les probabilités conditionnelles des termes sont estimées par simple comptage sur une base d'apprentissage. Si l'ensemble des termes n'est pas couvert par la base d'apprentissage, ce qui est fréquent, certains termes auront une probabilité nulle. Pour éviter ce problème, on attribue en général une probabilité a priori non nulle pour chacun des termes dont on tiendra compte dans l'estimateur. Cette approche extrêmement simple est connue sous le nom "Naïve Bayes", même si l'hypothèse d'indépendance est irréaliste dans le cas du texte, la méthode se révèle souvent efficace et est très courante en RI. Dans [Mit97] on pourra trouver une introduction simple à ce modèle, [Lew98] fournit une discussion plus détaillée.

Arbres de décision

De très nombreux algorithmes à base d'arbre ont été proposés à la fois par la communauté statistique et celle de l'apprentissage. Les plus connus sont sûrement CART [Brei74] pour la première et ID3 [Qui86] qui est maintenant supplanté par C4.5 [Qui93] pour la seconde.

Réseaux de neurones

Différentes techniques issues de cette famille ont été employées pour des tâches d'accès à l'information textuelle. On citera en particulier :

- les Perceptrons MultiCouches (PMC). Ces modèles offrent souvent de très bonnes performances pour la discrimination. Par rapport aux modèles basés sur des fonctions à noyaux, ils permettent de travailler dans des espaces de grande dimension et dans le cas du texte, de prendre ainsi en compte des informations contextuelles.

- les réseaux récurrents sont utilisés en discrimination ou en régression. Par rapport aux PMC dont toutes les connexions vont de l'entrée vers la sortie, ces systèmes ont des connexions rebouclées qui les transforment en systèmes à états, à chaque calcul, ils prennent en compte des valeurs calculées dans le passé. Ils permettent d'apprendre ainsi une mémoire locale du passé. En pratique, seuls des réseaux faiblement récurrents sont utilisés et dans le domaine du langage naturel, on rencontre principalement des réseaux dits de Elman [Elm91]. [Bis95, Her91] sont de bonnes introductions aux réseaux de neurones.

Machines à vecteurs supports

Depuis quelques années, la communauté apprentissage développe les Machines à Vecteurs Supports (SVM) proposées par Vapnik [Cor95, Vap95]. Ces techniques sont actuellement très populaires et en plein essor. Elles sont particulièrement séduisantes car elles implémentent

naturellement plusieurs idées directement issues de la théorie de l'apprentissage proposée par Vapnik [Vap95]. En particulier, elles offrent une bonne approximation du principe de minimisation du risque structurel.

Intuitivement, les idées principales à la base des SVM sont

- les données sont plongées dans un espace de grande dimension par une transformation souvent non linéaire.

- dans cet espace transformé, les classes seront séparées par des classifieurs linéaires qui maximisent la marge (une distance entre les classes).

- les hyperplans peuvent être déterminés au moyen d'un nombre de points limité qui seront appelés les vecteurs supports (ce sont ces points qui définissent la frontière).

L'apprentissage des SVM demande de résoudre un problème de programmation quadratique sous contraintes inégalités, ce qui peut se révéler délicat et assez lourd. De nombreux travaux ont été consacrés à ce problème technique ces dernières années.

La complexité d'un classifieur SVM va dépendre non pas de la dimension de l'espace des données, mais du nombre de vecteurs supports nécessaires pour réaliser la séparation. De nombreuses expériences rapportent un bon comportement de ces méthodes pour la généralisation (faible sur-apprentissage).

Modèles de Markov Cachés

Les modèles de Markov cachés (MMC) sont actuellement les modèles les plus employés pour décrire la génération d'une séquence. Un MMC est la combinaison de deux processus stochastiques, une chaîne de Markov et une densité de probabilité associée à chaque état de la chaîne. Un MMC (figure 1) modélise la production d'une séquence de la façon suivante : en commençant en un état de départ, on passe d'état en état suivant la matrice de transition de la chaîne Markovienne en emmettant chaque fois qu'on arrive en un état une valeur jusqu'à ce que l'état final soit atteint. Si on note $S = \{s_1, \ldots, s_p\}$ l'ensemble des états et X l'espace des observations, $b_k(x) = p(x / s_k)$ sera la probabilité d'émettre x dans l'état s_k, $a_{ij} = p(s_i / s_j)$ sera la probabilité de transition de l'état s_j à l'état s_i. Pour rendre les calculs et l'estimation possibles, on fait des hypothèses, les plus courantes sont : l'hypothèse Markovienne, qui dit que a_{ij} dépend uniquement des états s_i et s_j , l'hypothèse d'indépendance des sorties qui dit que $b_k(x)$ dépend uniquement de s_k et x. Les probabilités de transition et d'émission sont apprises par un algorithme itératif du type EM. En reconnaissance, un MMC prendra en entrée une séquence et fournira en sortie la séquence d'états la plus probable ainsi que son score. L'application de ces modèles a principalement été développée en parole. Depuis une dizaine d'année, ils sont également très utilisés pour modéliser des séquences biologiques, en TALN, ce sont les premiers modèles a avoir été développés avec succès pour l'étiquetage morpho-syntaxique, ils sont également de plus en plus utilisés pour la reconnaissance de l'écriture. Récemment plusieurs études ont proposé des implémentations de ces modèles pour l'accès à l'information textuelle. Elles considèrent en général des MMC discrets - X est un ensemble fini, on parlera alors de symboles pour désigner ses éléments. En texte, ces symboles seront tout simplement les mots du texte ou les termes issus d'un prétraitement. [Rab93] et [Cha93] décrivent les MMC respectivement dans le contexte de la parole et du traitement du langage naturel.

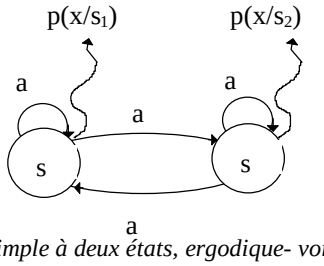

Figure 1 : Un MMC simple à deux états, ergodique- voir texte pour les notations.

Apprentissage relationnel

Tous les systèmes précédents reposent sur des techniques numériques et l'information extraite est peu explicite. Les arbres sont une exception puisqu'ils permettent de générer les règles utilisées pour la discrimination. Toutefois, ils perdent en expressivité dès que l'on utilise en un noeud des décision plus complexes qu'un simple test sur la valeur d'un attribut (e.g. des combinaisons d'attributs). De nombreux algorithmes permettant d'apprendre des règles exprimées comme des formules logiques sous forme normale disjonctive ont été développés en apprentissage symbolique [Mit97]. On distinguera les algorithmes qui apprennent un ensemble de règles propositionnelles (un attribut aura une valeur unique) et ceux qui apprennent des règles en logique du premier ordre (avec des variables).

La majorité des algorithmes apprenant des règles propositionnelles utilisent un algorithme de couverture : on construit une règle qui couvre un certain nombre d'exemples, on retire ces exemples de la base d'apprentissage et on recommence jusqu'à ce que tous les exemples soient couverts. La construction d'une règle dépend du choix des littéraux (attribut, relation, valeur) et du test d'arrêt. Pour l'ajout d'un littéral à une règle, on peut prendre le meilleur littéral au sens d'un critère (e.g. l'entropie) et ne jamais remettre ce choix en question, ou conserver à chaque étape les k meilleurs candidats pour choisir le meilleur sous ensemble parmi ces k. L'arrêt peut se faire en fonction d'un critère numérique lié au taux d'erreur. Ces algorithmes fournissent des règles similaires à celles d'ID3. Les règles du premier ordre qui contiennent des variables ont un pouvoir d'expressivité supérieur. L'apprentissage de ces règles est le domaine de l'ILP (Inductive Logic Programming). Une des approches est d'étendre l'algorithme de couverture précédant au cas des règles du premier ordre, c'est ce qui est fait dans l'algorithme FOIL (voir [Mit97]).

Ces techniques ont été utilisées par la communauté apprentissage pour des tâches d'extraction d'information.

Au delà du choix de la méthode, il existe de nombreux problèmes clés qui sont particulièrement importants pour le texte, comme :

- la sélection de caractéristiques : les espaces de représentation sont souvent de très grande taille (plusieurs milliers de termes), il est important de sélectionner les plus pertinents. De nombreuses techniques de sélection de caractéristiques ou de réduction de dimensionnalité ont été utilisées. Mentionnons la méthode Latent Semantic Indexing (LSI) [Dee90] qui utilise la décomposition en valeurs singulières d'une matrice, où chaque document est représenté par un vecteur décrivant la fréquence de ses termes, pour effectuer une projection de cette description dans un espace de taille plus faible.

- l'emploi de données non étiquetées : l'étiquetage de données est extrêmement coûteux en temps, alors que les données non étiquetées sont légion. Traiter de nouveaux problèmes ou adapter des systèmes à des domaines spécifiques demande souvent une phase d'étiquetage extrêmement longue, et même certaines fois impossible. L'emploi de faibles quantités de données étiquetées est devenu une préoccupation de premier plan pour l'analyse textuelle. Une voie prometteuse est l'utilisation simultanée d'un faible nombre de données étiquetées et d'une grande quantité de données non étiquetées. Ce problème a fait l'objet de nombreux travaux en statistiques. [Nig98] présentent une application à la catégorisation de textes.

3. La recherche d'information

3.1.1 Le domaine

La recherche d'information ou recherche documentaire traite de la recherche de documents correspondant aux besoins d'information d'un utilisateur. Ce terme recouvre un ensemble de problèmes distincts. Initialement le domaine s'est développé autour de la problématique des *requêtes libres* (ad hoc retrieval en anglais) : l'utilisateur formule des requêtes sous la forme de texte libre ou de mots clés, et le système recherche les documents pertinents dans un corpus qui change lentement. C'est le cas par exemple des recherches dans des bibliothèques électroniques.

A cette problématique initiale, s'est rapidement ajoutée celle des requêtes fermées où les requêtes en nombre fini sont connues à l'avance, le corpus lui est généralement dynamique et change au cours du temps. Le problème est alors souvent vu sous l'angle de la discrimination, un document se voyant assigner par le système une classe parmi un ensemble fini. C'est le cas par exemple de la catégorisation de documents, de la veille technologique, du filtrage de courrier ou de "news". On distingue en général le filtrage qui effectue un choix sur la pertinence d'un document par rapport à une classe et le routage qui ordonne un ensemble de documents par rapport à leur pertinence pour une classe.

Les travaux en RI abordent de nombreux autres problèmes comme la segmentation, l'interaction, les documents parlés, le suivi de sujet, etc. L'arrivée du web a mis l'accent depuis quelques années sur la recherche d'information multilingue, l'utilisation du texte dans le multi-média, la recherche dans les documents structurés (XML, hypertextes), la constitution automatique de bases de données textuelles à partir de texte libre ou semi structuré. Certains de ces axes de recherches en sont aujourd'hui véritablement à leur début et le domaine est en pleine évolution.

Nous renvoyons à [Sch97] et à [Sal93] pour une présentation du domaine, à [Leb94] pour une description de l'approche statistique. Nous introduisons simplement ci-dessous les deux grands modèles qui sont utilisés pour décrire le processus de recherche, il s'agit du modèle vectoriel et du modèle probabiliste.

3.1.2 Le modèle vectoriel

Deux ingrédients essentiels permettent de décrire un modèle vectoriel [Sal93] : la fonction de représentation des documents et la fonction de similarité entre documents. Les documents sont codés dans un espace de taille fixe sous la forme de vecteurs qui peuvent être soit booléens, un terme est présent ou absent du document, soit des vecteurs caractérisant la fréquence des termes dans le document. Dans ce dernier cas, la représentation la plus utilisée est connue sous

le nom de *tf-idf* (term frequency - inverse document frequency) : un terme sera d'autant plus caractéristique d'un document qu'il sera fréquent dans ce document et rare dans la collection de documents considérée :

$$tf\text{-}idf(w_i,d) = tf((w_i,d)idf(w_i)$$

où $tf((w_i,d)$ est le nombre d'occurrences du terme w_i, dans le document d, $idf(wi) = \log\left(\frac{1+N}{1+df(w_i)}\right)$ où $df(w_i)$ est le nombre de documents dans le corpus contenant w_i et N est la taille du corpus. Cette méthode qui a été déclinée en de très nombreuses variantes a donné très souvent de bons résultats et est utilisée comme référence.

Documents et requêtes sont codés de façon similaire, la proximité d'un document et d'une requête est calculée par exemple par l'angle entre les deux vecteurs de représentation.

La recherche documentaire est souvent un processus interactif, l'utilisateur intervient dans la boucle et peut renvoyer au système son appréciation sur le résultat de la recherche. L'interactivité est un pan extrêmement important de la RI.

3.1.3 Le modèle probabiliste

Le modèle probabiliste le plus courant [Fuh92] suppose qu'un corpus est généré de la façon suivante : pour créer l'ensemble des termes d'un document, on choisit une distribution de probabilité parmi un ensemble de distributions candidates, les mots du document sont ensuite générés suivant cette distribution. Le corpus sera alors modélisé par une loi de mélange, un document étant supposé généré par une seule des composantes. Pour classer le document d ou estimer sa pertinence, il faudra estimer respectivement $p(d / c)$ pour chaque classe puis utiliser la règle de décision de Bayes, la pertinence est donnée par exemple par la valeur $p(c / d)$ ou par d'autre quantités liées à cette valeur. Là aussi de très nombreux modèles ont été proposés [Lew98].

3.1.4 L'évaluation

L'évaluation des systèmes de RI est un problème complexe, d'autant plus qu'elle doit souvent prendre en compte l'utilisateur. Nous nous contenterons ici de mentionner quelques unes des mesures de base utilisées dans le domaine.

Les deux mesures les plus employées sont la précision P et le rappel R.

$$R = \frac{\text{nombre d'exemples pertinents selectionnés}}{\text{nombre d'exemples pertinents dans le corpus}}$$

$$P = \frac{\text{nombre d'exemples pertinents selectionnés}}{\text{nombre d'exemples selectionnés}}$$

Ces deux mesures sont définies pour un seuil de pertinence donné. Le comportement d'un système est en général donné par une courbe dite de *précision-rappel* qui donne les valeurs P et R pour différents seuils. Des mesures plus synthétiques permettant de résumer une courbe P-R en une valeur réelle ont été proposées [Hul94].

L'évaluation des systèmes de RI a lieu depuis une dizaine d'années dans le cadre des conférences TREC [HarXX], qui offrent un cadre pour comparer les systèmes sur plusieurs tâches de RI et sur des bases de données de grande taille.

3.1.5 Apprentissage et recherche d'information

Les méthodes de la RI sont largement empruntées aux statistiques et à l'informatique, et depuis très longtemps la RI utilise des techniques d'apprentissage. D'autre part, le domaine de la RI est extrêmement riche et a testé de très nombreuses méthodes depuis plusieurs années. Nous nous contenterons de décrire ci dessous quelques approches récentes qui visent à introduire de nouvelles méthodes d'apprentissage dans le domaine de la RI.

- *requêtes ouvertes*

Une des innovations récentes notables dans le domaine de la RI a été l'emploi de modèles de Markov cachés pour traiter les requêtes ouvertes. Les premiers modèles génératifs entrant dans ce cadre ont été proposés par [Mit94]. Cette approche permet en particulier de sélectionner les parties d'un document pertinentes pour une requête et de calculer le degré de pertinence par rapport à ces passages. On peut ainsi répondre à des questions ouvertes et présenter à l'utilisateur les sections de texte jugées les plus pertinentes sous la forme d'un surlignage. [Mil99] proposent également une modélisation markovienne de la RI, testée sur les requêtes ouvertes de TREC6 et TREC7, elle donne d'excellentes performances qui classent leur système parmi les meilleurs. L'idée de base de ce modèle génératif est simple, la génération d'une requête est modélisée par un processus de Markov, les mots de la requête sont générés par un parcours dans un MMC discret, un passage sur un état produisant un mot suivant la distribution de probabilité liée à cet état. Un *MMC* est ainsi *construit pour chaque document*, ce qui permet de calculer la pertinence d'un document en estimant la probabilité *p(q/d est pertinent)* et de sélectionner un ensemble de documents pertinents. Un modèle simple illustrant cette approche est présenté figure 2. Pour entraîner ce modèle, il faudrait disposer de couples (requête, document pertinent pour la requête) ce qui est rare dans le cas de requêtes ouvertes. Les auteurs proposent des solutions à ce problème ainsi que de nombreuses améliorations du modèle de base.

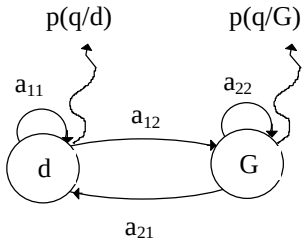

Figure 2 : l'approche de BBN pour les requêtes ouvertes, l'état d correspond au document modélisé, i.e. p(q/d) est la distribution des termes sur ce document, et l'état G à une modélisation "poubelle" de l'anglais général. Ce modèle est ergodique, l'arrivée et le départ peuvent se faire sur les deux états.

- *filtrage et routage.*

Les problèmes traités dans ce cadre se ramènent souvent à de la discrimination. Les arbres de décision sont sûrement parmi les premières méthodes à avoir été employées et sont très

répandus dans le domaine. Des perceptrons linéaires ont été utilisés dans [Sch95] pour du routage et ont montré les meilleures performances sur un corpus très large (Tipster 1M documents [Har93]) pour les tâches de routages de TREC 2 et TREC3 [Har], parmi un ensemble de 5 méthodes de discrimination. [Wie95] utilisent un système hiérarchique de PMC pour classer des documents tout d'abord en catégories grossières par rapport à un sujet spécifique dans chaque catégorie.

[Dum98] proposent une comparaison de plusieurs méthodes (Rocchio, arbres de décision, naïve Bayes, réseaux bayesiens simples, SVM linéaires) pour la catégorisation de documents avec des tests sur une collection Reuter (environ 13 K textes et 100 catégories)[1] et les SVM ont les meilleures performances. [Joa98] a été un des premiers à utiliser les SVM pour la catégorisation. Sur la même base Reuter, il compare plusieurs types de SVM avec un ensemble de méthodes (naïve bayes, Rocchio, C4.5, k-ppv) avec les mêmes conclusions. Ces méthodes présentent également l'intérêt de permettre le traitement de vecteurs de très grande taille, ce qui est particulièrement pertinent en RI. Dans ces grands espaces, les bases de données académiques utilisées sont souvent linéairement séparables. Le nombre de points de support est limité, ce qui est un cas favorable aux SVM. En terme de complexité, [Joa98] trouve ces méthodes similaires à C4.5.

De nombreuses méthodes à base de règles ont également été proposées, nous renvoyons à [Mou97] pour une bonne introduction à ces méthodes.

4. L'extraction d'information

4.1.1 Le domaine

En extraction d'information, on cherchera à analyser le contenu du texte, pour répondre à un besoin d'information spécifique. De façon schématique, on peut dire que la RI s'intéresse au document d'un point de vue global, alors que l'EI recherche des informations spécifiques au sein du document. Il s'agit d'une tâche bien plus complexe, les techniques employées sont principalement issues du traitement du langage naturel. Le développement et l'évaluation de systèmes d'EI permettant de traiter de grands corpus se sont principalement faits autour de la communauté compréhension de message qui organise depuis une dizaine d'années les conférences MUC [Muc6a, Muc6b, Muc7]. La tâche typique d'EI dans ces conférences est le remplissage de patrons ou formulaires sur des sujets pré-définis. Cette tâche réduite reste elle même extrêmement complexe. Nous nous placerons par la suite dans cette optique. Alors que dans les premières compétitions on voyait des systèmes extrêmement différents, les systèmes actuels sont assez standardisés. Ils enchaînent: une suite de transducteurs - des automates à états finis - qui réalisent des traitement locaux de leurs données d'entrée. Un système typique d'EI comporte les étapes suivantes [Cow 96, Car97], figure 3 :

- *analyse* : segmentation, analyse lexicale et morpho-syntaxique, il s'agit dans cette série d'analyses de reconnaître les constituants du texte (phrases, mots) et leurs relations, une analyse syntaxique partielle permet d'identifier des groupes de mots (verbal, nominal, ...).

- *extraction* : extraction de faits pertinents pour la tâche réalisée sur les entités déterminées lors des étapes précédentes.

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

- *génération* : résolution des coréférences et remplissage du patron, le système détermine si différentes entités trouvées lors de l'extraction se rapportent au même objet, puis remplit un patron pour chaque événement pertinent pour la tâche trouvé dans le texte.

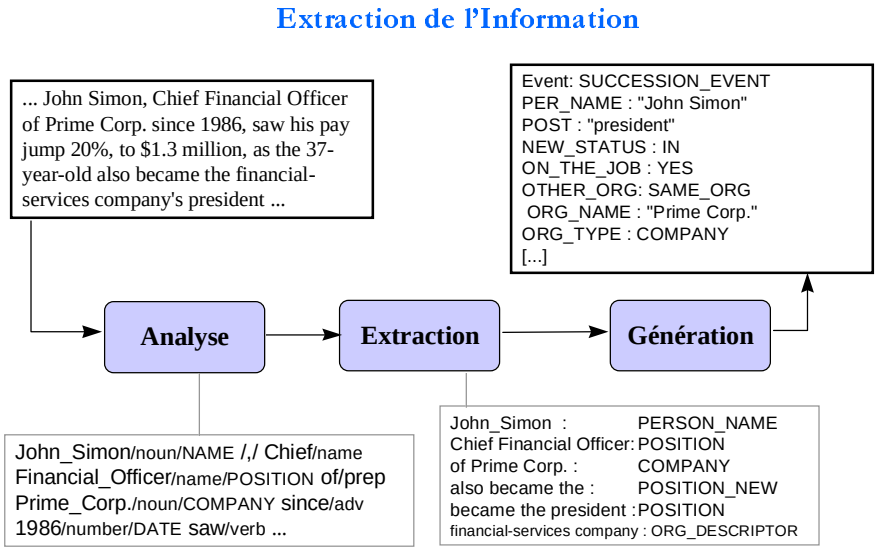

Figure 3 : extrait de [Zar99], illustration des différentes phases de l'extraction d'information. A partir du texte libre, on segmente et on réalise des analyses morpho-syntaxique, puis on extrait parmi les entités précédemment isolées des informations pertinentes pour la tâche avant d'examiner les liens entre ces différentes informations et de remplir un patron par événement pertinent pour la tâche (dans l'exemple, les mouvements de personnel). L'exemple est tiré de MUC6.

Parmi ces étapes, celles d'analyse sont supposées indépendantes du domaine, l'extraction est bien sûr dépendante du domaine, de même que la génération de patron à un moindre degré.

Evaluation

L'évaluation de ces systèmes est un problème complexe qui reste encore largement ouvert. Dans le cadre restreint du remplissage de patron, les systèmes d'EI sont évalués par des mesures copiées de celles utilisées en RI, qui sont le rappel et la précision. Le rappel mesure la proportion d'information pertinente que le système a découverte et la précision la fiabilité de ces informations.

P = nombre de champs correctement remplis / nombre de champs remplis

R = nombre de champs correctement remplis / nombre de champs à remplir

Ces mesures peuvent être définies sur les patrons ou sur un nombre limité de champs [Gri96]. A titre indicatif, pour la tâche MUC 7, la précision des systèmes est inférieure à 40% pour un rappel de 70%.

4.1.2 Apprentissage et extraction d'information

Pendant plusieurs années, les systèmes d'EI n'utilisaient quasiment pas d'apprentissage et leurs différentes composantes étaient construites en grande partie "à la main". Ces systèmes construits spécifiquement pour une tâche ou un domaine sont difficilement transposable à un autre domaine et demandent des mois de travail à des spécialistes de la tâche et des linguistes. Les compétitions MUC qui imposent d'une part de s'adapter à des tâches de plus en plus complexes et lourdes en traitant de grands corpus, et d'autre part de changer de domaine d'une

compétition à l'autre, ont montré l'importance de développer des systèmes rapidement prototypables et adaptables avec le minimum d'effort. La communauté EI a développé des efforts importants pour automatiser les différents traitements réalisés par ces systèmes, en introduisant de l'apprentissage, mais en conservant la même architecture de modèle générique. De même que pour la RI, nous passons en revue par la suite des exemples de travaux représentatifs.

En ce qui concerne la partie analyse, les premiers analyseurs morpho-syntaxique performants ont été développés à partir de MMC, e.g. [Kup92], ils restent encore parmi les plus employés et les plus performants. Des arbres de décision ont été employés par exemple pour l'analyse syntaxique.

Les systèmes d'EI utilisent généralement un étiquetage sémantique pour identifier des noms de compagnies, de lieux, d'individus, etc. C'est la tâche "named entities" de MUC avec par exemple dans MUC6, 7 types d'entités. A partir d'un corpus d'apprentissage étiqueté manuellement, qui est de l'ordre de 100 kmots, [Bik99] proposent un système MMC qui permet de détecter des noms et qui a été testé sur des bases en anglais (MUC6) et en espagnol (MET1). Même si étiqueter manuellement 100 kmots reste dans le domaine du raisonnable pour développer une nouvelle application, on voit clairement l'intérêt de concevoir des systèmes capables de généraliser à partir de peu de données. Les résultats reportés pour cette tâche sont excellents et situent leurs système au même niveau que les meilleurs systèmes à base de règles qui ont été développés manuellement.

La phase d'extraction est sûrement celle pour laquelle les tentatives pour introduire l'apprentissage automatique ont été les plus nombreuses. En particulier, de nombreux travaux ont été initiés par le groupe de W. Lehnert à l'université du Massachussets. Un corpus typique pour des tâches à la MUC contient les textes et les patrons associés à chaque texte. Pour chaque nouvelle tâche d'extraction, il faut créer un corpus. Malgré cela, l'information supervisée reste encore pauvre pour les différentes étapes qui conduisent à l'extraction finale puisqu'on ne dispose que des informations du patron et non pas d'un étiquetage précis du texte. Nous donnons ci dessous les idées principales de deux modèles qui ont été parmi les premiers à introduire l'apprentissage dans des systèmes d'EI.

Un des premiers systèmes à utiliser de l'apprentissage lors de la phase d'extraction a été autoslog [Ril93], ce travail a influencé nombre de systèmes qui ont suivi. Autoslog apprend des schémas d'extraction appelés "concept nodes". Ils vont servir à extraire les informations pertinentes du texte, comme par exemple le sujet ou l'objet d'une action que l'on veut caractériser. Un concept node sera composé d'un mot déclencheur (verbe ou nom), d'une forme linguistique et de contraintes qui garantissent l'application du concept node et caractérisent son contexte linguistique local. Par exemple si l'on s'intéresse à savoir quelles sont les personnes récemment embauchées par la compagnie Xsoft dans des phrases du type : "Zman has been hired by Xsoft", le déclencheur serait *hired*, la forme linguistique <cible>= <sujet> <verbe passif>, ce qui signifie que le nom propre à extraire apparaît comme le sujet d'un verbe passif. Autoslog applique ses "concept nodes" sur les phrases du texte après une analyse syntaxique partielle qui identifie les composants de la phrase. Le premier "concept node" qui s'applique sur la phrase (dans l'exemple simplifié ci dessus, il est déclenché par *hired* et vérifie la forme linguistique), permet d'extraire l'information recherchée, ici "Zman". Les concepts extraits correspondent en général à un des éléments à remplir dans le patron, ils sont utilisés dans les phases suivantes pour produire ce patron.

Pour l'apprentissage, ce système nécessite un corpus de textes et les mots à extraire. Il utilise un dictionnaire sémantique spécifique du domaine et un ensemble des formes linguistiques "génerales". Autoslog utilise 13 formes linguistiques et peut extraire un sujet, un complément d'objet direct ou une phrase nominale. A partir d'une cible à extraire, il détermine la phrase correspondante, réalise l'analyse syntaxique, applique les formes linguistiques générales en séquence et quand l'une d'entre elles s'applique, génère une définition de "concept node". Celle ci est ensuite vérifiée par un opérateur.

Même s'ils différent de Autoslog par les algorithmes employés, le type de forme considéré et les connaissances introduites dans le système, la plupart des systèmes qui utilisent une automatisation partielle de l'extraction obéissent à un schéma similaire.

Par exemple CRYSTAL [Sod95] est un successeur de Autoslog qui extrait des patrons sémantiques plus riches. Il nécessite également pour l'apprentissage un texte étiqueté, un dictionnaire sémantique, un analyseur syntaxique et opère sur des phrases. Par contre alors que la partie apprentissage de l'algorithme précédant était triviale, CRYSTAL est le premier système qui utilise un véritable algorithme d'apprentissage qui permet d'apprendre automatiquement un ensemble de règles logiques pour extraire les formes pertinentes. L'algorithme relationnel utilisé génère des règles propositionnelles. Le système commence par créer un concept pour chaque exemple positif de la base. Ces définitions de concepts sont ensuite généralisées en unifiant les définitions similaires. Pour unifier deux définitions, on cherche les contraintes les plus restrictives qui les couvrent toutes deux, sans produire d'erreur sur le corpus d'apprentissage. [Sod97] présente une adaptation de CRYSTAL, "Webfoot", qui permet de remplacer les phrases sur lesquelles opère CRYSTAL par des fragments de texte qui résultent d'une segmentation de marqueurs HTML. Ce système permet ainsi de traiter des pages du web.

De nombreux autres systèmes basés sur l'apprentissage de règles ont été proposés [Kim95, Huf95, Cal97, Sod99, Fre99]. Certains génèrent des règles en logique du premier ordre [Cal97, Fre99]. [Fre99] combine plusieurs types d'apprentissage, en particulier un classifieur naïve Bayes et un algorithme qui apprend des règles du premier ordre. [Sod99] et [Fre99] permettent de traiter du texte structuré, ces travaux sont motivés par les documents électroniques formatés qui constituent une part croissante de l'information textuelle.

L'utilisation de l'apprentissage pour les co-références a suscité quelques travaux. Le problème est en général traité comme une discrimination binaire : à partir d'un corpus annoté qui donne des couples de portions de texte référençant le même objet (ou ne référençant pas le même objet), on apprend à classifier ces couples avec une technique classique de discrimination. Ce classifieur est ensuite utilisé chaque fois qu'une décision sur une co-référence doit être prise. Les données sur lesquelles opère le classifieur utilisent le résultat des étapes précédentes dans le traitement. Par exemple [Aon95] utilise C4.5 sur des vecteurs caractérisant les portions de texte qui codent des informations lexicales, syntaxiques et sémantiques.

5. Accès à l'information

Les travaux de la communauté apprentissage se sont en grande partie développés en marge de ces deux communautés bien établies et se situent souvent à la frontière entre les deux. La dénomination "accès à l'information" englobe entre autres RI et EI. Nous allons présenter deux

directions de recherche récemment développées en apprentissage et qui connaissent un certain succès. La première concerne la visualisation d'information pour l'aide à la navigation dans les bases de texte, la seconde concerne l'emploi de MMC pour des tâches d'extraction d'information.

Visualisation

La visualisation de grands ensembles de données textuelles a fait l'objet de nombreux travaux tant en statistiques qu'en IHM. Dans la communauté apprentissage, le système le plus connu est sûrement le système Websom développé dans l'équipe de T. Kohonen [Koh97, WebSo]. Il utilise des cartes auto-organisatrices [Koh97] pour réaliser des projections non-linéaires de données d'un espace de grande dimension dans un espace de dimension 2. Cette projection préserve en partie la similarité des données - les documents - au sens de la métrique utilisée par l'algorithme et du mode de représentation de ces documents. L'espace de projection est discret et représenté par une grille 2-D sur laquelle est définie une topologie. Les documents sont projetés sur les points de cette grille (plusieurs documents similaires seront projetés sur un point) et des documents proches seront projetés sur des points voisins. L'apprentissage est non supervisé. Le système peut être utilisé pour *représenter* de grandes collections de documents - des essais ont été réalisés avec plusieurs millions de documents issus de différents corpus - sur la grille de dimension 2 et pour ensuite *naviguer* dans cette collection. Chaque projection a un lien avec le document original. Différentes représentations des documents ont été utilisées [Koh98], une des plus efficaces qui a de plus le mérite d'être extrêmement simple consiste à coder le document sous forme vectorielle par une représentation fréquentielle du type *tf-idf* puis à projeter ce vecteur de grande dimension (souvent plusieurs milliers) dans un espace de dimension réduite (quelques centaines) par une projection aléatoire. La matrice de projection a des composantes tirées aléatoirement suivant une loi normale. Cette méthode permet de réduire d'un ordre de grandeur la taille des documents sans perte notable de performance pour la discrimination [Kas98]. Le vecteur résultant sert d'entrée à la carte topologique. En sus de la navigation, ce système peut être utilisé pour effectuer de la recherche, la requête (dans le cas où il s'agit d'un document) est simplement codée de façon similaire aux documents et projetée sur la carte, ce qui permet de retrouver les documents proches. Les requêtes par mot clé sont traitées différemment. Des démonstrations sont accessibles à [WebSo].

Extraction et modèles de séquence

Ces deux dernières années, on a commencé à voir apparaître l'utilisation de séquence pour extraire par apprentissage, des informations au sein des documents. Il semble aujourd'hui qu'il existe un certain nombre de tâches d'extraction simple qu'il est possible de résoudre de façon quasi automatique. Les travaux que nous décrivons montrent différents exemples pour lesquels ces modèles donnent de bons résultats.

Extraire la structure d'un document à partir de son contenu textuel est pertinent pour de nombreuses tâches. Plusieurs travaux ont été consacrés à la segmentation de texte en portions qui reflètent par exemple une structure a priori, une cohérence dans le contenu, une pertinence par rapport à des requêtes. Les MMC ont été utilisés par [Sey99] pour extraire automatiquement les champs caractérisant l'en tête d'un papier de recherche dans le domaine de l'informatique, 15 types de champs sont considérés (titre, auteur, affiliation, ...). L'application visée est le remplissage automatique d'une base de données bibliographique. La méthode est implémentée dans le moteur de recherche "cora" [Cor]. Un simple modèle ergodique où un état est associé à chaque type de champs donne des performances de l'ordre de 92% de

passages correctement identifiés sur une base de 1000 textes, 500 pour l'apprentissage, 500 pour les tests.

Un autre exemple de l'utilisation de MMC est fourni par [Lee97] qui les utilise pour extraire à partir de textes d'articles en biologie des relations du type *nom de gène - localisation du gène*, ces deux entités peuvent correspondre à des mots ou des groupes de mots. Les essais réalisés sur une base de grande taille - des résumés d'articles représentant environ 20 MB de texte - donnent des performances intéressantes, e.g. une précision de l'ordre de 80% pour un rappel de 30 %. L'application envisagée est le remplissage de bases de données sur les gènes. Il s'agit là aussi d'une tâche d'extraction simple, dans un domaine spécifique, où la syntaxe employée est relativement limitée et qui utilise des connaissances très ciblées. Contrairement aux applications que nous avons déjà décrites, le MMC a une structure très adaptée à la tâche à réaliser. En particulier cette structure reflète la structure syntaxique des phrases recherchées. L'utilisation d'états "muets" qui ne participent pas au calcul du score permet de traiter des mots inconnus et de réunir dans un même fait extrait des fragments de phrases discontinus.

Dans le domaine medical, [Cra99] traite également de l'extraction de relations binaire - *identité de la protéine - localisation cellulaire de la protéine*, sur la base Medline. Les approches présentées sont d'une part une discrimination binaire par naïve Bayes - une phrase contient ou ne contient pas cette relation- et un apprentissage relationnel par FOIL.

[Fre99] considère également le problème de l'extraction dans des textes libres par MMC mais dans un cadre moins spécifique que [Lee97]. Ils considèrent des requêtes fermées, i.e. ils vont chercher à extraire de documents les informations pertinentes pour des classes prédéfinies. Ils construisent un MMC par champ à extraire. Le modèle de base est le même que dans le système de BBN (cf figure 2) : c'est un modèle ergodique dont un état modélise l'information pertinente pour le champ et l'autre l'anglais général. Toutefois, le point de vue est différent, alors que le système de BBN développe un point de vue requête - il considère un MMC par document et modélise la génération de la requête par l'ensemble des modèles -, ici, on a un point de vue document - un MMC modélisera un document du point de vue d'une des classes pertinentes (les champs à remplir). Chaque modèle segmente donc un document en deux : passages pertinents et passages non pertinents. Les états "pertinent" utilisent une modélisation du contexte et une méthode d'interpolation pour estimer les probabilités d'émission. Le système est testé sur 2 corpus :

- des annonces de séminaires composées d'environ 500 messages d'annonce d'où il faut extraire 4 champs (conférencier, lieu, heures de début et de fin).

- la partie "acquisition" du corpus Reuters [Lew92] qui comprend environ 2250 articles traitant de l'achat d'une entité par une autre entité ou par une personne. Cette collection a été largement exploitée dans le cadre de la RI. La tâche d'extraction qui est définie ici consiste à extraire 10 champs comme l'acquéreur, le vendeur, l'activité, le prix, l'état des négociations, etc.

Les performances moyennes en terme de rappel-précision de ce système simple sont par exemple meilleures que celles de l'algorithme relationnel SRV [Fre98].

A titre d'exemple, nous allons décrire de façon un peu plus étendue des travaux menés à Paris 6, dans notre groupe, qui exploitent également des modèles de séquence pour classifier des textes ou extraire des informations au sein de ces textes.

6. Un exemple

Nous avons développé depuis quelques années des modèles de traitement de séquence qui peuvent être utilisés pour un ensemble de tâches dans le cadre de l'analyse de textes [Zar98a,b]. Nous nous plaçons dans le cas de requêtes fermées, où les classes sont connues à l'avance.

Un document sera représenté par une séquence de termes $w=\{w_1,...,w_n\}$. A chaque document est associée une classe c, et à la séquence de termes une séquence d'étiquettes $t=\{t_1,...,t_n\}$. L'ensemble des étiquettes est noté $\{\tau_1,...,\tau_K\}$. Quand on voudra apprendre à extraire différents types d'information dans un texte codé par la séquence w, les éléments de la séquence t indiqueront le type (la classe) de l'information associée aux éléments correspondants de w. Ces différents niveaux de description des documents sont représentés sur la figure 4. En fonction de la tâche à réaliser, certaines informations de ce modèle de document seront disponibles et d'autres absentes. Nous nous plaçons dans un cadre probabiliste, i.e. les différentes quantités manipulées d, w, t, c sont la réalisation de variables aléatoires.

Figure 4. le modèle de document

Nous allons instancier notre modèle sur 4 tâches :

Catégorisation: on veut classer des documents, pour l'apprentissage, on dispose d'un ensemble de documents d et de leur classe c.

Tri: on veut ordonner un ensemble de documents par rapport à leur pertinence pour une classe, pour l'apprentissage, on dispose d'un ensemble de documents d et de leur classe c.

Surlignage: on veut extraire ou surligner les portions de document qui sont pertinentes pour différentes classes ou requêtes. Pour l'apprentissage, on dispose d'un ensemble de documents d et de leur classe c, par contre on n'a pas d'étiquette sur les portions de document.

Extraction : comme dans les applications décrites en section XX, on veut extraire des portions de texte pertinentes pour un ensemble de requêtes fixe et connu à l'avance pour l'apprentissage, on dispose de documents d étiquetés au niveau mot par la séquence t.

Sur la figure 5, le schéma de représentation de la figure 4 est repris pour indiquer les caractéristiques des différentes tâches.

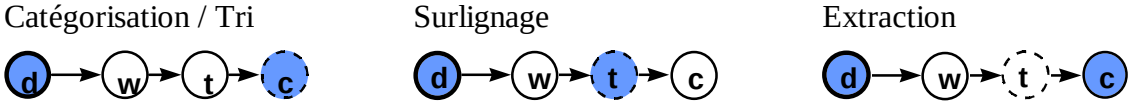

Figure 5: Les différentes tâches et leurs caractéristiques pour l'apprentissage et la reconnaissance. Les traits gras représentent l'information connue, les traits pointillés l'information que l'on désire obtenir, le gris indique l'information disponible lors de l'apprentissage.

Considérons la probabilité jointe des trois variables *w, t et c*, $P(w,t,c) = P(w)\,P(t|w)\,P(c|t,w)$. En faisant une hypothèse d'indépendance conditionnelle raisonnable, nous pouvons l'écrire sous la forme $P(w,t,c) = P(w)\,P(t|w)\,P(c|t)$ qui est plus aisément implémentable. Pour chacune des tâche décrites au dessus, nous nous intéresserons alors à déterminer les quantités suivantes :

Catégorisation / Tri : $c^* = argmax_c\, P(c|w) = argmax_c \sum_t P(c|t)P(t|w)$

Surlignage : $(c^*,t^*) = argmax_{c,t}\, P(c,t|w) = argmax_{c,t}\, P(c|t)P(t|w)$

Extraction : $t^* = argmax_t\, P(t|w)$

Nous avons donc deux types de termes à estimer qui sont $P(c|t)$ et $P(t|w)$.

$P(t|w)$ se factorise sous la forme $P(t|w) = \prod_i P(t_i \mid t_{1,i-1}, w_{1,n})$. Son utilisation pratique demande des hypothèses simplificatrices, différentes hypothèses conduisant à différents modèles. On peut par exemple dériver les modèles suivants [Ami99b] :

Si on suppose l'indépendance des mots et celle des étiquettes, on obtient :

$$P(t|w) = \prod_i P(t_i \mid w_i)$$

qui correspond au modèle naïve Bayes.

Des hypothèses un peu moins restrictives permettent de prendre en compte l'information de séquence au niveau des mots et des étiquettes. Sans détailler ces hypothèses (voir [Ami99a,b]), on peut facilement dériver les deux modèles :

$$P(t \mid w) = P(w)^{-1} \prod_i P(t_i \mid t_{i-1})\, P(w_i \mid t_i) \qquad (1)$$

qui est le modèle classiquement utilisé pour les MMC du premier ordre.

$$P(t \mid w) = \prod_i P(t_i \mid w_{i-K,\, i+K})\, P(t_i \mid t_{i-1}) \qquad (2)$$

contrairement au MMC, ce modèle prédit les étiquettes de mots en utilisant un contexte local sur la séquence de termes.

Le terme $P(c/t)$ servira en général à introduire des information a priori sur ce que l'on recherche dans le document. Une façon naturelle d'introduire cette connaissance est par le biais d'une grammaire stochastique. $P(t/c)$ sera alors la probabilité que la séquence *t* ait été générée par la grammaire associée à la classe *c* [Zar99]. Par exemple, si pour une tâche de résumé on cherche le passage le plus pertinent d'un document pour une requête, on peut considérer que la séquence d'étiquettes doit obéir à une grammaire du type *<NP><P><NP>*, où *P* dénote l'étiquette "pertinent", *NP* "non pertinent" et *<x>* une ou plusieurs répétition de *x*.

Codage

Contrairement aux modèles décrits en section 5 qui n'utilisent pas de représentation particulière des termes mais estiment directement des probabilités discrètes sur chaque terme, nous avons introduit une représentation très compacte des termes qui prend en compte la nature fermée des requêtes. Pour une classe donnée - un type d'informations à extraire - notons *n* et *n'* le nombre de séquences pertinentes et non pertinentes dans lesquelles ce terme apparaît. Notons *m* et *m'* le nombre de séquences pertinentes et non pertinentes dans lesquelles il n'apparaît pas. Le terme w_i est codé par une valeur réelle qui est la statistique *U* [And92] définie comme suit :

$$u_i = U(w_i) = \sqrt{N}.\frac{nm' - n'm}{\sqrt{(n+n')(n+m)(n'+m')(m+m')}}$$

Cette mesure qui est proche du X^2 représente de façon discriminante l'information apportée par un terme pour une classe donnée et dans ce sens code la "sémantique" des termes par rapport à la tâche. Des représentations plus complexes prenant en compte des informations morpho-syntaxiques ont également été proposées [Ami92a].

Ce modèle a été testé sur les différents types de tâche décrits plus haut. Nous ne présentons pas de résultats détaillés et renvoyons à [Zar99] pour cela, nous donnons simplement quelques illustrations du comportement et des possibilités de la méthode.

Filtrage et tri

Le modèle (2) a été utilisé. Les essais ont été réalisés à partir du corpus 20-newsgroup développé à CMU, qui consiste en 20 000 messages électroniques de 20 newsgroup (1000 par groupe). La tâche consiste à catégoriser ces e-mails. Pour l'évaluation 500 exemples par classe ont été sélectionnés aléatoirement. En apprentissage, nous avons utilisé des ensembles de différentes tailles pris dans chacune des classes, complétés par un ensemble de 5K e-mails pris sur l'ensemble des autres classes. Dans cette application, nous n'avons utilisé aucun prétraitement (élimination, stemming, ...) avant la projection par la statistique U. L'estimation de $P(t_i \mid w_{i-K,\, i+K})$ a été réalisée par des PMC. La figure 6 illustre les possibilités d'utilisation du système. Elle donne des courbes de précision rappel sur un des newsgroup (alt.computer.graphics) pour 300 exemples positifs utilisés en apprentissage. Ces courbes correspondent à trois grammaires, *G1* n'utilise pas de grammaire spécifique, mais simplement un seuil sur la probabilité conditionnelle calculée par le PMC pour selectionner les mots les plus pertinents, *G2* est la grammaire $<NP>[P]_5<NP>$, qui recherche dans un texte la séquence de 5 mots consécutifs la plus pertinente, *G3* est la grammaire *<NP>[P]<NP>*, qui recherche le passage le plus pertinent dans le texte, sans contrainte de longueur sur ce passage. Cette figure montre clairement que le choix de la grammaire influe sur les performances du système et que c'est un atout supplémentaire par rapport à un système sans contrainte. L'évaluation subjective illustrée par la partie droite de la figure 6 qui montre en surligné les mots ou séquences sélectionnés par les grammaires, est plus complexe. *G3* fournit un bien meilleur résumé du texte que G1, même si les textes courts utilisés ne sont pas très favorables à ce type d'évaluation. Remarquons que les *p(t/w)* sont estimés indépendamment de la grammaire choisie pour calculer les *p(c/t)* et qu'on peut donc utiliser deux systèmes différents pour trier et surligner.

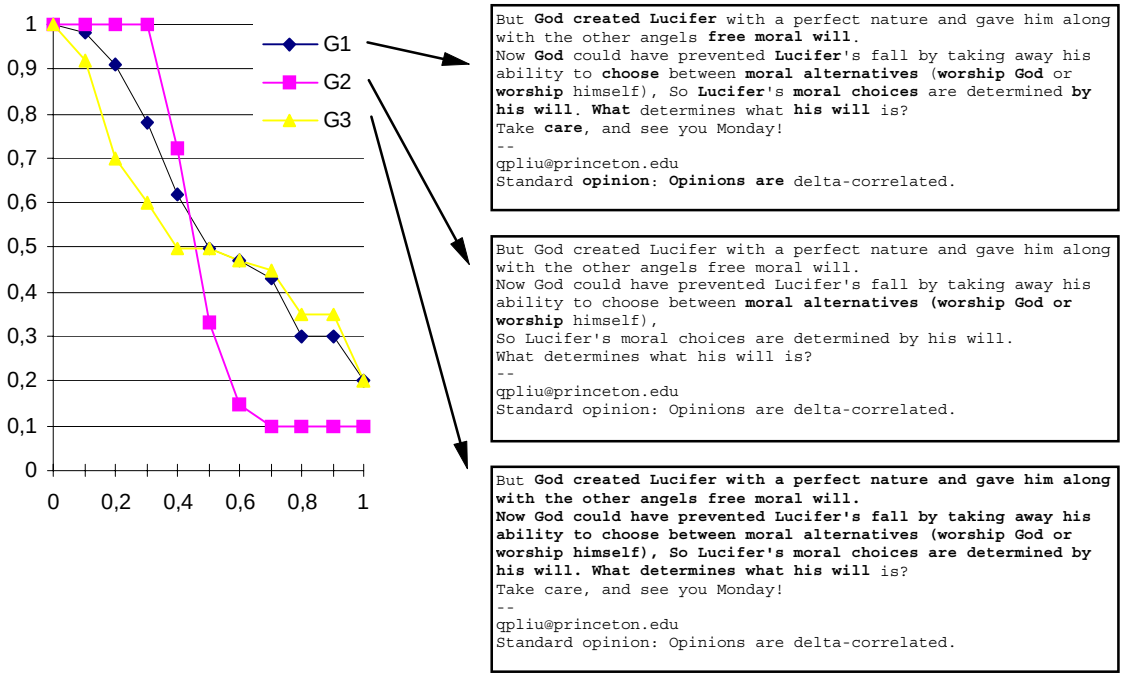

Figure 6 : Trois grammaires différentes sont appliquées à la catégorisation de e-mail. Les courbes de précision-rappel sont données à gauche et des exemples du surlignage produit lors de la classification sont donnés à droite.

6.1 Extraction d' Information

Pour l'extraction d'information, nous avons utilisé le corpus MUC6 [Gri96]. Il contient 200 articles du Wall Street Journal. La tâche est la suivante : on s'intéresse aux changements de personnel cadre de haut rang dans certaines entreprises, et pour chaque changement (arrivée, départ, changement au sein de l'entreprise) décrit dans un article, le corpus contient un patron de scénario dont les champs pré-définis caractérisent l'événement (figure 3). Nous nous sommes intéressés au remplissage de deux de ces champs qui sont les noms de personne et d'emploi pertinents pour ces changements. On peut remarquer que cette tâche est différente de celle des entités nommées car seules les informations pertinentes aux changements nous intéressent, un nom intervenant en dehors de ce contexte par exemple n'est pas pertinent. Pour l'apprentissage, nous avons étiqueté les bases au niveau des mots. Nous considérons deux tâches : le surlignage où il s'agit de déterminer les séquences pertinentes dans le texte pour les deux classes étudiées et l'extraction où ces deux classes sont distinguées.

	Surlignage		**Extraction**		
	Pertinent	*Moyenne*	*Emploi*	*Personne*	*Moyenne*
modèle de base	82.62	77.97	56.88	53.17	64.43
G1	79.71	81.10	54.75	51.27	67.54
G2	83.36	78.6	56.77	**58.9**	68.37

Table 1 : Performances du modèle PMC pour les tâches de surlignage (gauche) et d'extraction (droite). G1 et G2 sont les grammaires <I><R><I> et <I | $[R]_3$> pour le surlignage et <I><Per><Pos><I> et <I|$[Per]_3$|$[Pos]_3$> pour l'extraction.

La table 1 donne à titre d'illustration quelques performances pour ces deux tâches, l'implémentation du modèle (2) a été ici aussi réalisée avec des PMC (d'autres implémentations -e.g. MMC ou réseaux récurrents - donnent des résultats analogues). La grammaire G1 a la

forme *<I><R><I>* et *<I><Personne><Position><I>* pour le surlignage et l'extraction respectivement et *G2* $<I\ [R]_3>$ et $<I\ [Per]_3\ [Pos]_3>$. Ces performances montrent que les modèles sont effectifs sur les tâches abordées. La figure 7 illustre les deux opérations de surlignage et d'extraction sur une portion de texte issue de la base.

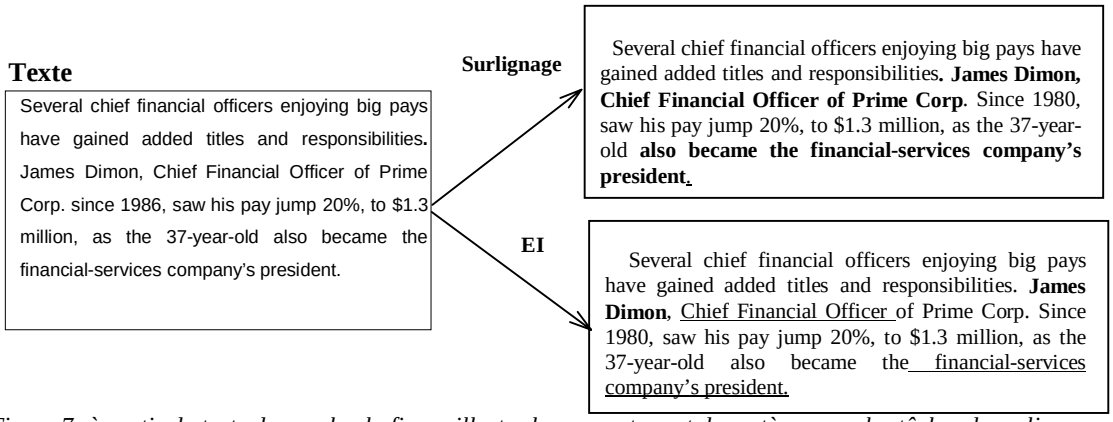

Figure 7: à partir du texte de gauche, la figure illustre le comportement du système pour les tâches de surlignage et d'extraction. Dans le cas du surlignage (en haut à droite), nous avons indiqué en gras l'information pertinente, dans le cas de l'extraction (en bas), nous avons indiqué en gras les noms de personne extraits et en souligné les emplois.

7. Conclusion

Nous avons décrit quels étaient les liens entre analyse d'information textuelle et apprentissage en nous focalisant sur les problèmes de la recherche et de l'extraction d'information. Dans les deux cas, nous avons vu que ces communautés utilisent les techniques de l'apprentissage, depuis longtemps dans le cas de la RI, plus récemment pour l'EI. Les problèmes d'analyse sur le texte changent aujourd'hui tant par la nature des textes que par les besoins de l'utilisateur. C'est ce qui a motivé une partie de la communauté apprentissage à s'intéresser à ces problèmes, c'est également ce qui motive les communautés RI et EI à s'intéresser à de nouveaux problèmes. Cette expérience est récente, mais on voit déjà apparaître des travaux originaux qui se distinguent de ceux conduits auparavant. Nous avons décrit un ensemble que nous pensons représentatif des travaux issus de l'apprentissage qui montrent certaines de ces nouvelles directions.

BIBLIOGRAPHIE

[Ami99a] Amini M.R., Zaragoza H., Gallinari P. (1999) " Sequence Models for Automatic Highlighting and Surface Information Extraction" 21th Annual Colloquium on IR Research, British Computer Society's IR Specialist Group.

[Ami99b] Amini M.R., Zaragoza H., Gallinari P. (1999) Stochastic Models for Surface Information Extraction in Texts ICANN99.

[And92] Anderson E., (1992), *The statistical analysis of categorical data*, Springer.

[Aon95] Aone C., Bennet W., (1995), "Evaluating automated and manual acquisition of anaphora resolution strategies", Proceedings of the 33rd meeting of the Association fo Computational linguistics, 122-129.

[Bik99] Bikel D., Schwartz R., Weischedel R.M., (1999) "An algorithm that learns what's in a name", Machine learning 34, 211-231.

[Bis95] Bishop C.M., (1995), Neural networks for pattern recognition, Clarendon press - Oxford.

[Bre84] Breiman, L. Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984) *Classification and Regression Trees*,Wadsworth.

[Bre91] Brent, R.P. (1991) "Fast training algorithms for multilayer neural nets", *IEEE Trans. on Neural Networks*, 2, pp 346-354. (Référence d'article)

[Cal97] Calif M., Mooney R. J., (1997), "Relational learning of pattern match rules for information extraction", proceedings ACL workshop on natural language learning, 9-15.

[Car97] Cardie C., (1997), "Empirical methods in information extraction", AI magazine 39 (1), 65-79.

[Cha93] Charniak E., (1993), *Statistical language learning*, MIT Press.

[Cor] www.cora.justresearch.com

[Cor95] Cortes C., Vapnik V., (1995), "Support vector networks", Machine Learning, 20, 273-297.

[Cow96] Cowie J., Lehnert W., (1996), "Information extraction", CACM 39 (1), 80-91.

[Cra99] Craven M., (1999), "Learning to extract relations from Medline", AAAI-99 workshop on machine learning for information extraction.

[Dee90] Deerwester S., Dumais S., Furnas G., Landauer K., (1990) "Indexing by latent semantic analysis", J Am. Soc Inf. Sci, 41, 391-407.

[Dum98] Dumais, S. Platt, J. Heckerman, D. Sahami, M. (1998), "Inductive algorithms and representation for text categorization", ACM CIKM98.

[Elm91] Elman J.L., (1991), "Distributed representation, simple recurrent networks and grammatical structure", Machine learning, vol 7, 195-225.

[Fre98] Freitag D., "Machine learning for information extraction in informal domain" PhD thesis, CMU, Report CMU-CS-99-104.

[Fre99] Freitag D., McCallum A., (1999), "Information extraction with HMMs and shrinkage", AAAI-99 workshop on machine learning for information extraction.

[Fuh92] Fuhr N., (1992), "probabilistic models in information retrieval", The computer journal, 35 (3), 243-255.

[Gri96] Grisham R., Sundheim B., (1996), "Message understanding conference - 6: a breif history, COLING 96 vol 1, 466-471.

[Har93] Harman D., (1993) "Overview of the first trac conference", SIGIR'93.

[Har], Harman D.K.,(Ed)., Proceedings of the TREC conferences, (TREC1 (1993) to TREC8 (1999)), special publications, NIST.

[Her91] Hertz J., Krogh A., Palmer R.G., (1991) Introduction to the theory of neural computation, Addison Wesley.

[Huf96] Huffman S., (1996), "Learning information extraction patterns from examples", in symbolic, connectionist and statistical approaches to learning for natural language processing, LNCS, Eds, Wermter S., Riloff E., Scheler G, 246-260.

[Hul94] Hull D.A., (1994) Information retrieval using statistical classification, PhD thesis, Dpt statistics, Stanford Univ.

[Joa98] Joachims T., (1998), "Text Categorization with Support Vector Machines: Learning with Many Relevant Features". European Conference on Machine Learning (ECML'98).

[Joa98] Joachims, T. (1998) "Text categorization with support vector machines: learning with many relevant features", ECML98.

[Joa99] Joachims T., (1999), Transductive Inference for Text Classification using Support Vector Machines. International Conference on Machine Learning (ICML'99).

[Kas98] Kaski S., (1998), "Dimensionality reduction by random sampling", Proc IJCNN'98, 413-418.

[Kim95] Kim J., Moldovan D., (1995), "Acquisition of linguistic patterns for knowledge-based information extraction, IEEE trans. on Knowledge and data engineering. 7(5), 713-724.

[Koh97] Kohonen T., (1997) Self-Organizing Maps, Springer Series in Information Sciences, Vol. 30, Second, edition..

[Koh98] Kohonen T. (1998), "Self organization of very large document collections: state of the art, ICANN'98, 65-74.

[Kup92] Kupiec J., (1992), "Robust part of speech tagging using a hidden Markov model", Computer Speech and Language, 6, 225-242.

[Leb94] Lebart L., Salem A., (1994), Statistique textuelle, Dunod.

[Lee97] Leek T.R., (1997), "Information extraction using hidden markov models", Master thesis, Univ. Ca. San Diego

[Lew98] Lewis, D.D. (1998) "Naïve Bayes at forty: the independence assumption in information retrieval", ECML98.

[Mil99] Miller, D.R.H. Leek, T. Schwartz R.M. (1999) "BBN at TREC7: using hidden Markov Models for Information retrieval", Proceedings of TREC7, D.K. Harman, editor.

[Mit94] Mittendorf, E. Schauble, P. (1994) "Document passage retrieval based on hidden markov models", SIGIR94, 318-327.

[Mit97] Mitchell T. M., (1997), Machine learning, McGraw-Hill.

[Mou97] Moulinier, I. (1997) Une approche de la catégorisation de textes par apprentissage symbolique, Thèse, Université Paris 6.

[Muc6a] Grishman R., Sundheim B., (1996), Message Understanding Conf. - 6: A Brief History Proc. 16th Int. Conf. on Computational Linguistics (COLING 96), vol. 1, 466-471,

[Muc6b] (1995) Proceedings of the Sixth Message Understanding Conference Morgan Kaufmann.

[Muc7] (1999) Proceedings of the Seventh Message Understanding Conference, Morgan Kaufmann.

[Nig98] Nigam K., McCallum A., Thrun S., Mitchell T., (1998), " Learning to classify text from labelled and unlabelled documents", AAAI'98.

[Qui86] Quinlan J.R. (1986), "*Induction of decision trees*", Machine learning, 1(1), 81-106.

[Qui93] Quinlan J.R., (1993), C4.5 : programs for machine learning, Morgan Kaufman.

[Rab93] Rabiner N., Juang L.Y., (1993), fundamentals of speech recognition, Prentice Hall.

[Ril93], Rilof E., (1993), "Automatically constructing a disctionary for information-extraction tasks, AAAI'93, 811-816

[Sal93] Salton G., (1993), Automatic text processing: the transformation, analysis and retrieval of information by computer, Addison Wesley

[Sch95] Schütze, Hull, Pedersen, (1995) "A comparison of classifiers and document representations for the routing problem, SIGIR'95, 229-237.

[Sch97] Schäuble P., (1997) Multimedia information retrieval, Kluwer.

[Sey99] Seymore K., McCallum A., Rosenfeld R., (1999) "Learning hidden markov model structure for ,information extraction, AAAI-99 workshop on machine learning for information extraction.

[Sod95] Soderland S., Fisher D., Aseltine J., Lehnert W., (1995) "CRYSTAL: inducing a conceptual dictionary.

[Sod97] Soderland S., (1997), "Learning to extract text based information from the world wide web", KDD'97.

[Sod99] Soderland S., (1999), "Learning information extraction rules for semi-structured and free text", Machine learning XX.

[Vap95] Vapnik V., (1995), *The nature of statistical learning theory*, Springer.

[WebSo] http://websom.hut.fi/websom/.

[Wie95] Wiener, Pedersen, Weigend, (1995), "A neural network approach to topic spotting", SDAIR'95, 317-332.

[Zar98a] Zaragoza H., Gallinari P, (1998),. "A Hierarchical Approach for the Combination of Information Retrieval and Extraction", 10th European Conf. on Machine Learning, (ECML'98) - Text Mining Workshop , 1998 ,

[Zar98b] Zaragoza H., Gallinari P, (1998), "Coupled Hierarchical IR and Stochastic Models for Surface Information Extraction" The 20th Annual Colloquium on IR Research, British Computer Society's Information Retrieval Specialis

[Zar99] Zaragoza H, (1999), "Modèles dynamiques d'apprentissage numérique pour l'accès à l'information textuelle", PhD thesis, Université Paris 6.