Quel modèle pour détecter une opinion? Trois propositions pour généraliser l'extraction d'une idée dans un corpus

Eric Charton, Rodrigo Acuna-Agost[†]

Laboratoire Informatique d'Avignon, Université d'Avignon et des Pays de Vaucluse
B.P 1228 84911 Avignon Cedex 9 - France
{eric.charton, rodrigo.acuna}@univ-avignon.fr

Résumé : Nous décrivons dans cet article trois méthodes d'extraction d'une opinion dans un corpus, mises en œuvre dans le cadre de la campagne DEFT07. La première repose sur des mesures de similarité cosine et de probabilité d'appartenance d'un document à une classe en fonction des mots qu'il contient. La seconde exploite la régression logistique, méthode rarement utilisée en classification de textes. La troisième met en œuvre une technique à base de mesure de densité et de compacité inspirée des systèmes de question-réponse. Notre approche tente de tirer parti de la pluridisciplinarité de nos travaux pour obtenir une solution algorithmique de classification adaptée de manière générique à la recherche d'idées dans un texte.

Mots-clés : Ingénierie des connaissances, Optimisation, Classification de textes, Apprentissage machine, Similarités, Système de question-réponse, Recherche d'informations.

1 Introduction

La classification d'un corpus en classes pré-déterminées est une problématique importante du domaine de la fouille de textes. De manière générale, la classification revient à rechercher des dissimilarités ou des similarités entre groupes d'individus dans une population donnée. Les applications sont variées : détection de langues, filtrage de grands corpus, recherche d'information, classement thématique. DEFT07 nous propose d'explorer le domaine applicatif de la classification non plus orientée par une thématique, mais plutôt vue sous l'angle des idées. Plus précisément, ici, la tâche proposée consiste à identifier une opinion, mais une méthode appropriée pour répondre à ce besoin pourrait tout aussi bien être transposée à des segmentations de documents par les jugements, avis ou tendance qu'ils expriment. Les possibilités applicatives sont nombreuses : mesure d'une opinion retournée par la presse suite à un lancement de produit, analyse de l'engagement d'un média dans le cadre d'une étude politique, classification automatiquement des actes juridictionnels, etc...

Pour ce qui concerne DEFT07, l'étiquetage d'un corpus avec deux ou trois classes représentant n opinions "bonne", "moyenne" ou "mauvaise" (ou "favorable" et "défavorable" dans le cadre des débats parlementaires) revient à un partionnement de corpus en n classes. Nous remarquons que les corpus de textes présentés ont pour particularité de provenir de quatre sources très différentes, et par voie de conséquence, d'exprimer l'opinion sous des formes elles aussi très différentes. Ceci ne sera pas sans incidence sur nos choix de méthodes de classification. Les systèmes de classification de textes les plus efficaces sont des algorithmes d'apprentissage supervisés qui peuvent être entrainés sur un jeu de données déjà étiquetées. Ces apprentissages conduisent à la construction de modèles de classes, qui, dans notre exemple, correspondront pour un corpus de textes donné, à une opinion. Quelle forme discriminante pourrait caractériser une opinion ? Peut-elle être caractérisée par un vocabulaire et ainsi entrer dans le cadre d'un système de recherche d'information classique (mesure de distance, de similarité cosine, de probabilité d'apparition de mots, mesure statistique) ? Doit-elle être considérée comme une réponse à une question (qui aurait la forme d'une question booléenne de type "ce produit est-il bon" ?), et dans ce cas, être recherchée avec un système de question-réponse (QR) ?

[†]Supported in part by ALFA Grant II-0457-FA-FCD-FI-FC

Pour nous faire une première opinion, nous avons étudié visuellement ces textes, puis utilisé des outils de mesures statistiques et de comptage des mots contenus dans les textes (le logiciel countworld.pl[1], ainsi que l'outil LSA de mesure de co-occurences)(Favre *et al.*, 2005).

1.1 Les corpus de DEFT07

On observe en lisant le tableau TAB.1 que les corpus d'apprentissage fournis sont très disparates : le corpus relectures d'articles, par exemple contient une quantité relativement faible de documents (881) alors qu'à l'opposé, le corpus de débats parlementaire est très volumineux. On note également que ces deux corpus, à l'inverse des deux premiers, de taille plus moyenne (JeuxVidéo, et Avoir A Lire) ont pour particularité de présenter des textes dont la taille peut varier de manière conséquente.

- Nous avons pu relever d'après ces premières investigations que le corpus issu de *jeuxvidéo.com* répond à des normes journalistiques classiques en matière d'essais de produits. Ce qui revient à exprimer l'opinion de l'auteur dans le chapô [2] d'introduction et dans la conclusion, en adoptant des séquences répétitives, faisant appel à un vocabulaire essentiellement qualificatif et relativement restreint.
- Celui issu de *AvoirAlire*, qui relève lui aussi de la critique, exprime les opinions sur des produits culturels de manière bien moins tranchée et localisée, en adoptant un vocabulaire plus étendu que celui de *jeuxvidéo.com*.

Ces deux corpus, bien que répondant à des disciplines d'expression journalistique identiques - la critique - expriment les opinions de leurs auteurs par des moyens radicalement opposés : tranché et localisé dans un cas, subtil et diffus dans l'autre.

- Pour ce qui est du corpus de relectures d'articles, on observe très nettement une concentration de la qualification autour d'un ensemble de mots très réduit (il est possible de visualiser ce phénomène en utilisant LSA pour mesurer les co-occurences de mots les plus fréquentes).
- Dans le corpus *Débats*, on relève une modalité d'expression de l'opinion (qui est en réalité l'expression d'un engagement) très variée, et parfois délicate à évaluer, y compris après une lecture "humaine".

Nom	Classes	Nbr Train	Doc Taille max	Doc Taille min
Avoir A Lire	3	2074	4167	931
JeuxVideo	3	2537	13703	4600
Relectures	3	881	7584	153
Débats parlementaires	2	11533	3100	300

TAB. 1 – Caractéristiques des Corpus de DEFT07

1.2 Algorithmes

Partant de ces constats, nous avons imaginé des propositions originales de classification d'opinions. Nous allons dans cet article, décrire trois algorithmes d'apprentissage et de classification supervisés, appliqués sur les corpus de DEFT07.

- Nous présentons en premier lieu un algorithme classique de mesure de distance entre un document et une classe par mesure de similarité cosine entre deux vecteurs de poids de mots. Ces vecteurs représentant pour l'un une classe, pour l'autre un document. Dans cette méthode, la construction des classes repose sur une préparation du texte en vue d'en supprimer les éléments non discriminants (lemmatisation, filtrages), d'en extraire les éléments significatifs (localisation de l'opinion dans le texte, utilisation de n-grammes). La construction de classes est réalisée par apprentissage sur une partie des corpus d'entrainement. La qualité discriminante de la classe est ensuite évaluée par calcul de F-Score en testant les classes sur la partie du corpus d'entrainement non utilisée pendant l'apprentissage. Nous procédons à une optimisation de F-Score en mesurant les résultats obtenus avec les combinaisons d'options de filtrages et en ne conservant que le paramétrage le plus performant.
- En second lieu, nous utilisons un algorithme statistique basé sur la régression logistique. La méthode cherche par un processus de calcul de fréquences de mots observés dans les différentes classes du

[1] Outil trivial de comptages des occurences de mots dans un document, mis au point par Benoit Favre.

[2] Le terme *chapô* décrit, dans le secteur de l'édition périodique, le résumé intercalé entre le titre et le corps de l'article

corpus d'apprentissage, à élaborer les caractéristiques d'un modèle. Ces caractéristiques sont les variables explicatives. Ces caractéristiques sont ensuite utilisées pour analyser les valeurs prises par une variable quantitative catégorielle, correspondant aux observations faites sur un document à classer dans l'une des catégories d'opinion. On déduit de cette analyse une probabilité d'appartenance à une classe d'opinion.

– Notre troisième algorithme est inspiré des mesures de densité et de compacité de mots. Ces méthodes sont mises en oeuvre dans les systèmes de question-réponse élaborés pour les campagnes Trec ou Technolangue-EQueR. Nous cherchons dans chaque sous corpus (par exemple Corpus :*Débat* Classe :*favorable*), à localiser un ou plusieurs mots centroïdes susceptibles de représenter le milieu d'une phrase ou d'un passage, exprimant l'opinion. Ces mots identifiés, nous construisons des classes représentant la probabilité d'apparition d'un mot ou d'un n-gramme à proximité d'un centroïde (exemple *"bon"* à côté de *"article"*), pour une opinion donnée. Nous utilisons ensuite ces probabilités pour attribuer une classe à un document.

Cet article est organisé comme suit : dans la section 2 nous exposons les modèles de nos trois algorithmes. Dans la section 2.1, nous présentons notre méthode de classification par mesure de similarité cosine. Dans la section 2.2, nous détaillons notre implémentation de la méthode d'affectation d'un document à une classe d'opinion par régression logistique. Dans la section 2.3 nous détaillons notre proposition d'implémentation d'un système de mesure de densité et de compacité. Dans la section 3, nous développons les résultats obtenus à DEFT07 avec les données d'entrainement, puis avec les données d'évaluation, et les commentons. Nous terminons cet article par un ensemble de conclusions et tentons d'élaborer, à la lumière de nos résultats, quelques pistes de recherches futures.

2 Méthodes de classification proposées

2.1 Classification par mesures de similarité

La principale application de la recherche documentaire par mesure de similarité est constituée des moteurs de traitement de requêtes. On retrouve son principe au cœur de tous les méta-chercheurs, de Google à Yahoo, MSN, Exalead... L'idée directrice de cette méthode est la possibilité de mesurer la distance qui sépare un groupe de mots contenus dans une requête de plusieurs groupes de mots représentant les documents d'un corpus. En projetant ces ensembles dans un espace vectoriel sous une forme numérique (par exemple en affectant des poids aux mots) on évalue leur degré de proximité. Les références des ensembles dont on a ainsi mesuré la distance sont ensuite retournées sous forme d'une liste triée en fonction de leur degré d'éloignement avec la requête. Intrinsèquement, cette méthode est donc d'autant plus efficace que les sous ensembles de mots comparés ont des périmètres clairement délimités et que leurs intersections sont les plus reduites possibles. En d'autres termes, la mesure de similarité est adaptée aux extractions d'informations thématiques, mais les subtiles nuances que l'on peut observer dans l'expression d'une opinion, semblent - de prime abord - plus délicates à traiter. Particulièrement dans la langue française, où un même ensemble de mots agencés différement, peut exprimer deux opinions radicalement opposées. Ainsi, par exemple, les phrases *"Il n'est pas très bon ce film .."* et *"Il est très bon ce film n'est ce pas"*, bien qu'antagonistes, sont pourtant quasiment indiscernables avec une mesure de distance, et établissent clairement les limites applicatives du modèle. Néanmoins, la méthode de recherche par similarité étant à la fois très répandue, et peu coûteuse en terme de temps de calcul, il nous a semblé important de l'exploiter ici pour en délimiter la portée, dans le cadre applicatif de DEFT07. La recherche par similarité telle qu'elle existe dans les moteurs de recherche documentaire sous la forme $Distance(\textbf{\textit{requête}}, document_k)$, peut être adaptée à un système de classification thématique $Distance(classe_k, document)$ selon le modèle suivant :

– Considérant les vecteurs $\vec{A_k}$, représentant k classes d'opinion.
– Considérant un vecteur $\vec{D}$, représentant un document à classer d'après $\vec{A_k}$.

Les vecteurs $\vec{A_k}$ et $\vec{D}$ sont composés de poids des mots contenus dans les documents qu'ils représentent. Ces poids sont calculés par la formule TF.IDF (Salton & Buckley, 1988) :

$$w_{ti} = tf_{ti}.idf_{ti} = tf_{ti}\left(log\left(\frac{N}{df_{ti}}\right) + 1\right)$$

où pour $\vec{A_k}$ et $\vec{D}$:

- w_{ti} est le poids du terme ti dans le document où dans la classe
- tf_{ti} dit Term Frequency est le nombre d'apparitions du terme ti dans le document où dans la classe
- N étant le nombre de documents composant le corpus, ce qui peut être ramené ici à k pour les vecteurs de classes $\vec{A_k}$ et 1 pour le vecteur de document à classer $\vec{D}$
- df_{ti} étant le nombre de documents qui contiennent ti dans le corpus d'apprentissage, pour la classe k

Considérant que tf_{ti} est équivalent à la probabilité de voir apparaître le mot w dans un document sachant sa classe, soit $p(w|\vec{A_k})$, on en déduira qu'en sélectionnant dans une classe k les mots ou objets textuels les plus représentatifs de l'opinion qu'elle caractérise, on maximisera le caractère discriminant de son vecteur $\vec{A_k}$. En théorie, et dans le cadre d'un système de mesure de similarité cosine appliqué à la recherche d'opinion, l'objectif sera donc de sélectionner les termes w les plus fréquents dont la probabilité d'appartenir à une classe $\vec{A_k}$ et à aucune des autres classes est la plus élevée.
Pour mesurer le degré de similarité entre $\vec{A}$ et $\vec{D}$, on reprend le principe des calculs de similarités sur des données numériques qui peuvent être ramenés au calcul du produit scalaire sur les deux vecteurs $\vec{D}$ et $\vec{A_k}$. Lorsque la norme euclidienne est choisie pour normaliser les composants des vecteurs $\vec{A}$ et $\vec{D}$, le calcul du produit scalaire se ramène à celui du cosinus. Soit :

$$cosine(\vec{A}, \vec{B}) = \frac{\vec{A}.\vec{B}}{\|\vec{A}\|.\|\vec{B}\|}$$

Et dans le cadre de DEFT07, pour chercher la classe k qui correspond au document représenté par $\vec{D}$ à maximiser le calcul du cosinus soit :

$$\vec{A}(k) = Argmax_{A_i}\left(cosine(\vec{D}, \vec{A_i})\right)$$

Si l'efficacité des familles de mesures qui découlent de ces applications est démontrée dans le cadre de moteurs de recherche documentaire, se posait pour nous la question de savoir si une opinion bonne, mauvaise ou moyenne pouvait se mesurer avec elles.
Pour maximiser le caractère descriptif des valeurs de TF.IDF des mots représentés par les vecteurs $\vec{A_k}$ on cherche généralement à réduire l'espace de représentation du vocabulaire pour ne conserver dans cet espace que des objets textuels caractéristiques. On adopte ainsi un ensemble de possibilités de filtrage du texte que l'on activera ou non, en fonction de la qualité du F-Score obtenu avec les données de vérification :

- Réduction du langage par suppression des mots outils, non représentatifs d'une opinion, avec un anti-dictionnaire[3].
- Réduction des mots à leurs lemmes pour regrouper le vocabulaire porteur du même sens (ex "bon", "bonne") en utilisant un lemmatiseur[4].
- Suppression des noms propres, à priori non porteurs de sens pour décrire une opinion, en utilisant les lettres majuscules en tant que repères.
- Combinaison de mots consécutifs sous forme de n-grammes pour conserver des séquences porteuses d'opinions (ex "Article de qualité").

A cette réduction de l'espace de représentation par filtrage du vocabulaire on peut aussi adjoindre des méthodes de valorisation de certains mots ou groupes de mots qu'on espère porteurs d'informations. On proposera pour cela les possibilités suivantes :

- Utilisation d'une partie seulement du texte pour construire les classifieurs, et classer les documents : on retient ainsi le postulat que, dans un article journalistique, l'opinion est généralement exprimée dans le chapô, l'introduction ou la conclusion. On tente donc de réduire la portion de texte à sa partie significative, pour supprimer autant que possible les phrases non porteuses d'opinion, et donc susceptibles de bruiter le classifieur. Cette option est donnée sous forme de pourcentages du texte à prendre en tête et en fin de document[5].

[3] On utilisera ici l'antidictionnaire de Jean Veronis, présenté sur http ://www.up.univ-mrs.fr/veronis/logiciels/index.html.

[4] Le lemmatiseur mis en oeuvre est celui élaboré par Benoit Favre au LIA.

[5] On notera que nous avons complété cette option par un seuil minimal de mots à prendre dans le document pour éviter les réductions excessives pour les cas où les textes sont de petite taille.

– Nous intégrons la possibilité de suppression des intersections entre classes, soit $\forall i \neq j,\ Ci \cap Cj = \emptyset$ où Ci, Cj sont des classes. On retire tous les objets (mots ou n-grammes) présents dans plus d'une classe pour étendre le caractère discriminant des classifieurs.

Pour obtenir le meilleur compromis, nous combinons tous ces paramètres et mesurons pour chaque classifieur le résultat obtenu par F-Score.

2.2 Classification par régression logistique

Dans son expression mathématique, l'analyse par régression examine la relation entre une variable dépendante (la variable de réponse) et des variables indépendantes particulières (les prédicteurs). Dans notre modèle, la variable de réponse est binaire. Mais cette variable est égale à 1 si le document en cours de comparaison est bien celui correspondant au modèle de la classe en cours d'examen. La variable est égale à 0 dans tous les autres cas.

Nous pouvons donc dire en résumé que le coeur de la régression logistique réside dans la définition d'un jeu de variables indépendantes. Ces variables de prédiction doivent être calculées uniquement en utilisant le contenu du document. Nous utilisons d'ailleurs dans cette version de notre algorithme la fréquence de certains mots dans les différentes classes de corpus déjà étiquetées, mise en rapport avec le nombre total de mots contenus dans le texte. Considérant ces aspects, nous utilisons bien la régression logistique en tant que méthode centrale de notre algorithme.

Cette idée est venue des travaux de l'un des auteurs, qui met en oeuvre le modèle de régression dans les applications de recherche opérationnelle. La méthode est utilisée pour réduire de manière astucieuse le nombre de variables entières d'un problème de réorganisation d'horaires de transport ferroviaire. (Mages, 2006). L'état de l'art de cette méthode laisse apparaître un vaste champ d'application.

Au cours des années précédentes, la régression logistique à été mise en oeuve dans le cadre de très nombreuses applications, répondant à des secteurs d'activités très variés : en médecine, (T. Cleophas, 2006; G. Venkataraman, 2006; G. Wu, 2006) ; dans le cadre des sciences de la vie et biomédicales (Oexle, 2006; D. Testi, 2001) ; Sciences du comportement (A. Menditto, 2006; B. Rosenfeld, 2005) ; Sciences sociales et de la législation (S. Wasserman, 1996; G. Robins, 1999; S. Stack, 1997) ; les sciences de la terre et de l'environnement (N. Sahoo, 1999; K. Chau, 2005; W. Wilson, 1996) ; le monde des affaires et de l'économie (Rodriguez, 2001; N. Dolsak, 2006) ; les sciences de l'informatique et de l'ingénieur (M. Collins, 2002; B. Jiang, 2004; J. Colwell, 2005) ; et finalement dans l'industrie et les sciences de la matière (A. Marinichev, 2005; A. Valero, 2006).

Tous ces travaux utilisent la régression logistique pour produire des équations prédictives. Dans la plupart de ces applications, la finalité est de démontrer que certains facteurs sont significatifs et d'exprimer l'importance de ces facteurs par une variable de réponse binaire. L'analyse par régression obtient de très bons résultats dans tous les domaines que nous venons de présenter. A notre connaissance, elle n'a pas encore été mise en oeuvre dans les activités de fouille de textes et plus généralement de TALN, en particulier dans les problèmes de classification, pour lesquels elle nous semble pourtant particulièrement appropriée.

2.2.1 Modèle de régression logistique pour extraire et classer une opinion

Selon le modèle de la régression logistique, nous commençons par définir dans notre méthode des ensembles et des index :

i : Index des documents.
j : Index des classes
C : Ensemble de catégories.
T : Ensemble de documents d'apprentissages

L'analyse par régression logistique permet de rechercher et d'estimer des modèles de regression multiple quand la réponse attendue est dichotomique, et peut être de type booléen. On utilise généralement la régression logistique lorsque l'on souhaite modéliser une question statistique qui peut être résumée par une réponse de type *"l'évenement a eu lieu/l'évenement n'a pas eu lieu"* et de manière plus générale, toute question à deux issues. Dans cet esprit, notre modèle aura deux possibilités de réponses : lorsqu'il cherche

à affecter un texte à une classe d'opinion, il répondra soit 1 lorsque la classe en cours de test est la bonne, soit 0 pour tous les autres cas.

La dépendance dichotomique (résultat binaire d'après une expérience ou une observation), implique que la variable de dépendance peut prendre une valeur de 1 avec une probabilité de succès évaluée par θ, ou une probabilité de défaut de $1-\theta$ si la valeur de la variable est 1. Nous sommes donc ici dans le cadre applicatif d'une variable de Bernouilli. Ce qui conduit à définir pour nos besoins un estimateur θ_{ij} tel que :

$$\theta_{ij} : \text{ estime la probabilité que le document } i \text{ soit apparenté à la classe } j$$

On voit ainsi que les variables explicatives sont indépendantes et peuvent prendre n'importe quelle forme. La régression logistique ne propose aucune hypothèse quand à la distribution des variables indépendantes. Elles ne sont pas nécessairement distribuées selon une loi normale, reliées linéairement ou de variances équivalentes à l'intérieur de chaque groupe.

En conséquence, il n'existe aucune règle pour définir les facteurs, pas plus qu'il n'est possible d'affirmer que chaque document correspondant à une classe, possède une proportion identique ou proche de mots spécifiques (et donc critiques pour la segmentation). Nous prenons acte de cette particularité sous la forme d'une variable indépendante explicative.

Par ailleurs, nous incluons le nombre total de mots contenus dans le document dans une autre variable explicative, car nous postulons que la classification pourra être expliquée, au moins en partie, par la prise en compte du volume du document. Dans ce contexte, les variables explicatives sont :

$$z_{ij} : \text{Le nombre de mots critiques dans le texte } i \text{ de la catégorie } j$$
$$y_i : \text{Le nombre total de mots dans le texte } i$$

Nous savons aussi que la relation entre les variables explicatives et les variables de réponse binaires n'est pas une fonction linéaire en régression logistique. Nous utilisons donc la fonction de régression logistique qui est une transformation logit de :

$$\theta_{ij} = \frac{e^{\left(\alpha_j+\gamma_j y_i+\sum\limits_{k\in C}\beta_j^k z_{ik}\right)}}{1+e^{\left(\alpha_j+\gamma_j y_i+\sum\limits_{k\in C}\beta_j^k z_{ik}\right)}} \tag{1}$$

Où :

α_j : Constante de l'équation $\forall j \in C.$ (2)

β_j^k : Coefficient des variables de prédiction z_{ik}. Valide pour le modèle qui évalue la catégorie j $\forall j \in C.$ (3)

γ_j : Coefficient des variables de prédiction y_{ij} $\forall j \in C.$ (4)

Il en résulte que pour chaque corpus, il est nécessaire de calculer $|C|$ modèles de régression différents. Dès que tous les paramètres de chaque modèle de régression ont été estimés, il devient possible d'appliquer l'équation (1) pour évaluer la probabilité qu'un texte i appartienne à une classe j. Ainsi, pour un document donné, nous avons une probabilité calculée de son appartenance, pour toutes les classes possibles. Finalement, le critère pour assigner une classe à un document sera la plus grande probabilité obtenue d'appartenance du texte à cette classe.

$$j^* = \arg\{\max_{j\in C}\theta_{ij}\} \tag{5}$$

2.3 Classification par mesure de densité calcul de compacité

Les systèmes de question-réponse (QR) sont définis comme des systèmes de recherche orientés non plus pour fournir une réponse d'après une mesure de similarité entre un document et une requête, mais par une évaluation de l'adéquation entre le contenu sémantique d'une requête et des réponse possibles, extraites du corpus. On notera néanmoins que l'analyse du contenu sémantique est réduite à son strict minimum, c'est à dire aux structures morpho-syntaxiques et que le résultat de recherche repose quasi exclusivement sur des approches statistiques. On parle d'ailleurs de modélisation statistique du langage (Thierry Spriet, 1996).
En effet, pour répondre à une question, un système QR procède à une suite séquentielle de traitements qui sont autant d'enrichissements et de filtrages des questions et des réponses. L'enrichissement consistera dans un premier temps à étiqueter le plus finement possible la question et le corpus pour préparer la mise en relation de concepts similaires contenus à la fois dans la question et dans le corpus. Par "concepts", on entend des entités nommées, et par relation, on entend trouver dans un segment du corpus, une entité nommée cible susceptible de correspondre au concept formulé dans la requête. On peut imaginer par exemple que des questions, débutant par les termes *"Qui"*, *"A qui"* pris en tant que concepts, sont à mettre en relation avec des réponses contenant une entité nommée de type *"Personne"*. Le modèle d'algorithme utilisé dans les systèmes de question-réponse repose sur une mesure de densité pour la recherche de passages (Gillard *et al.*, 2006).
Le principe retenu est d'élaborer un score de densité qui permette d'identifier la zone d'un segment issu d'un texte qui présente le plus de similitudes avec une question. A l'intérieur de chaque document, une distance moyenne $\mu(o_i)$ est calculée entre l'occurence courante o_i et les occurences des autres objets de la requête. Le calcul de ce score est effectué pour chacun des "objets caractéristiques" o_i d'un document D. La pénalité p fixée empiriquement doit favoriser le score produit par une concentration (proximité forte) de quelques objets communs de la requête, plutôt qu'une proximité faible mais pour un grand nombre d'objets. Ainsi on a :

$$\textit{DensitéScore}(o_i, D) = \frac{log[\mu(o_i) + (card\{\bigcup_{o_i \in Q} o_i\} - card\{\bigcup_{o_i \in D} o_i\}) * p]}{card\{\bigcup_{o_i \in Q} o_i\}} \quad (6)$$

Suite à l'application de ce premier test, on obtient un ensemble d'entités réponses candidates (ERc) qui sont confrontées à des passages jugés informatifs, extraits du corpus. Ces passages sont le plus souvent obtenus en déplaçant une fenêtre de n mots sur le document dans lequel on recherche une réponse.
Il est proposé en tant que mesure de compacité pour la sélection de la réponse au sein de toutes les ERc proposées, d'associer à la mesure de densité, le critère du "Confidence Weighted Score" (Voorhees, 2006). L'idée est de considérer chaque occurence d'une ERc comme point zéro d'un repère (décrit également en tant que "centre" ou "centroïde" selon les auteurs), et la présence des mots de la question autour de ces ERc issues du corpus comme des indices de présence de réponses correctes. On cherche à retrouver ici des "sacs de mots" les plus compacts et complets provenant de la question, autour de l'ERc (Luhn, 1958). Ce calcul de compacité est défini par :

$$\textit{Compacité}(ER_{c_i}) = \frac{\sum_{X \in MQ} P_{X,ER_{C_i}}}{\mid MQ \mid} \quad (7)$$

où $p_{X,ERci}$ correspond à la précision mesurée à l'intérieur d'une fenêtre centrée sur l'ERc_i pour les mots non vides de la question, à l'intérieur d'un rayon R, fixé par l'occurence la plus proche X_p du mot X (Gillard *et al.*, 2006).
Notre idée est que les systèmes de question-réponse, en ce sens qu'ils peuvent êtres définis comme des systèmes de recherche d'information spécifiques, sont adaptable à une recherche d'opinion, si cette dernière est considéré comme une information spécifique, posée comme une question. On peut même envisager que, comme dans le cas des campagnes d'évaluation, la réponse fournie à une question posée puisse être de nature binaire (*"L'opinion de ce document est elle mauvaise"* $\Longrightarrow$ [oui/non]) ou factuelle (*"Cet article est il de qualité"* $\Longrightarrow$ [Bonne/Moyenne/Mauvaise]).
Dans le cadre applicatif qui nous intérèsse, le problème particulier posé est que la liste des questions qui permettrait de chercher une réponse, n'est pas préexistante, comme dans le cas des campagnes TREC par exemple. Nous devons donc adapter le fonctionnement du système de question-réponse pour qu'il soit en mesure de construire automatiquement et d'après le corpus, pour une classe d'opinion donnée, toutes les

questions qui pourraient être posées. Il faudra ensuite que ces questions soient formulées de telle manière que l'algorithme puisse fournir une réponse factuelle ou binaire.

2.3.1 Application d'un modèle de question réponse dégradé

La méthode que nous proposons est la suivante. On considère les questions comme n groupes de mots regroupés autour d'un objet M constitutif de l'expression d'une opinion O de classe k, O_k. Ces *"sacs de mots"* de chaque O_k, peuvent éventuellement être lemmatisés et traités par un anti-dictionnaire. On considérera ensuite l'ensemble des éléments contenus dans un sac O_{k_M} comme de possibles réponses caractéristiques d'une opinion donnée, lorsque l'on rencontrera l'objet M dans un segment de texte issu du corpus à classer. L'hypothèse que nous formulons est qu'un score de compacité calculé sur toutes les phrases sélectionnées, par ce que contenant M de O_k, permettra de localiser les meilleures réponses candidates à une question posée (par exemple *"Cet article est il bon ?"* pour localiser les documents de classes *"[Bon/Moyen/mauvais]"* du corpus relecture, pour l'objet M, *"article"*). On calculera le score de compacité pour chaque groupe de mot $x_i \in M$ dans chaque O_k, puis on considérera que la somme de tous les scores de compacité obtenus avec tout O_{k_M} pour chaque O_k indique à quelle classe appartient le document.

Pour classer un document Y, on recherchera dans ce documents tous les segments y d'une longueur de n mots, contenant en leur centre (soit $y_{n/2}$) une occurence de l'objet M pour une opinion O_k. Tous ces segments seront considérés comme autant d'entités réponses candidates y pour Y. Puis on calculera le score de compacité par :

$$\textit{compacité}(y \mid O_k) = \frac{\sum\limits_{X \in O_{k_M}} P_{X,y}}{\mid O_{k_M} \mid} \tag{8}$$

Ou pour tout mot X de O_{k_M} présent dans y on mesure la distance Δ qui le sépare du centre représenté par M. Dans notre application, Δ est égale au nombre de mots qui séparent X de M dans y. On notera que pour construire les listes de questions de O_{k_M}, nous avons besoin d'identifier les objets centroïdes, équivalent des entités nommées "source". Cet objet M est représenté par des mots tels que "article" ou "papier" dans le cas du corpus "relectures". Ces sources devront être accompagnées des cibles les plus porteuses de sens lorsqu'elles sont associées à l'objet M : le plus souvent des adjectifs qualificatifs dans le cas d'une opinion, par exemple "bien", "bon", "mauvais", "favorable" dans un modèle parfait. Ces cibles sont regroupées dans les sacs de mots O_{k_M}. Ce qui explique pourquoi nous avons qualifié ce modèle de "dégradé" (ou restreint) : il n'implique pas de procéder à un étiquetage du corpus pour définir les objets M qui serviront à construire les sacs de mots O_{k_M}. En effet, par défaut, deux étiquettes morpho-syntaxiques sont suffisantes dans notre modèle : *"Objet d'opinion qualifié par des adjectifs"*, et *"Adjectif qualificatif de l'objet d'opinion"*.

2.3.2 Localisation automatique des objets

Pour localiser les objets M caractérisant les sacs de mots, nous avons considéré que les mots les plus fréquents porteurs de sens sont ceux de plus forte occurence subsistant après application d'un antidictionnaire, pour chaque classe. Dans le corpus "Avoir à Lire" par exemple, les mots les plus fréquents seront notamment *"Film"* et *"Album"*. Dans le corpus *"Relectures"*, c'est le mot *"article"* qui ressort, *"loi"* pour les *"Débats parlementaires"*, etc. On notera que les mots objets peuvent être les mêmes d'un sous-corpus d'opinion à un autre.

Nous recherchons ensuite autour de ces objets M ou entités nommées "source", l'ensemble de mots non outils "cible" susceptible de qualifier la source. Cette recherche se fait automatiquement en définissant un périmètre exprimé en nombre de mots autour de l'entité nommée source (n mots précédent et n mots suivant), et en ne conservant dans ce périmètre que les mots "non outils". Cette opération est menée sur chaque sous-corpus correspondant à une classe d'opinion (exemple *"Relecture/favorable"*, *"Relecture/défavorable"*, *"Relecture/moyen"*). A la suite de cette opération nous possédons n listes de phrases "réponses possibles" associées à l'ensemble des objets M correspondants aux classes O_k de chaque corpus.

On remarque que la relation entre les objets et leurs qualificatifs n'étant pas d'ordre sémantique, mais exclusivement statistique (extraction par comptage d'occurences), il est fréquent qu'un même ensemble M_n soit présent simultanément dans deux classes (exemple "Une bonne idée mais un article de mauvaise qualité" devient l'élément M'_n "bonne idée article mauvaise qualité" dans $O_{k'_{M'}} =$ *défavorable*, et la

phrase "Une mauvaise idée pour un article de bonne qualité" est représenté sous la même forme M_n dans O_{k_M} = *favorable*, ce qui revient dans ce cas à $P(M_n \mid O = favorable) = P(M'_n \mid O =$ *défavorable*$)$. On considérera que le système peut compenser de lui-même la présence de ces résultats antagonistes si les sacs de mots construits autours des M sont les plus exhaustifs possible pour chaque O_k. Nous avons également vérifié que l'on augmentait $p(M_n \mid O_k)$ en insérant dans les sacs de mots des groupes composés de bigrammes, qui restituent dans leur classe la localisation des mots les uns par rapports aux autres (exemple *"est_bon"* ou *"pas_bon"*).

3 Résultats obtenus

3.1 Resultats du modèle par similarité

L'aprentissage a été conduit par une séparation du corpus en deux. La première moitié du corpus a été utilisée pour l'entrainement par le classifieur[6]. La seconde partie a été utilisée pour calculer les F-Score. Pour les options optimalement choisies, les scores obtenus en phase d'entrainement et en phase d'évaluation DEFT07 sont présentés dans le tableau 2.

Corpus	Lem	Antidico	$\cup = 0$	npr	pct	ngrams	FS Train	FS DEFT07
Avoir à Lire	oui	oui	non	oui	20/30	3	0.50	0.37 (0.48)
Jeux vidéos	oui	oui	oui	non	20/30	3	0.73	0.62 (0.65)
Relectures	oui	oui	non	oui	50/50	3	0.37	0.43 (0.46)
Débats	non	non	oui	non	20/30	3	0.63	0.61 (0.64)

TAB. 2 – Résultats obtenus sur chaque corpus et options utilisées pour les obtenir

Le tableau 2 présente les options retenues pour obtenir les F-Scores indiqués. On trouve respectivement : lemmatisation des mots (options oui,non), application d'un antidictionnaire (oui, non), $\cup = 0$ supression des intersections (oui, non), *npr* correspond à la supression de noms propres (oui,non), *pct* décrit les pourcentages du document pris en tête du document à classer et à la fin, *ngrams* décrit le nombre de ngrammes ajoutés au texte. Dans la dernière colonne, nous indiquons le F-Score obtenu sur le corpus d'apprentissage après séparation en deux parts égales, le F-Score final obtenu sur les données de DEFT07 et entre parenthèses le F-Score moyen des participants de DEFT07.

Ces paramètres sont obtenus par une évaluation systématique conduite sous forme d'un processus de recherche du maximum de F-Score pour tous les corpus C :

$$MaxF - Score(C) \tag{9}$$

Par itérations successives pour toutes les combinaisons des valeurs de variables de décision suivantes :

- lemmatisation=$\{0, 1\}$
- antidictionnaire=$\{0, 1\}$
- supression des intersections=$\{0, 1\}$
- supression des noms propres=$\{0, 1\}$
- supression des noms propres=$\{0, 1\}$
- pourcentage de texte pris en tête=$[0, 50]$[7]
- pourcentage de texte pris en fin=$[0, 50]$
- ngrammes=$\{1, 3\}$

On note que les résultats de DEFT07 obtenus après reconstruction des classifieurs d'après le corpus d'aprentissage complet en conservant les meilleurs paramètres, sont relativement éloignés de ceux obtenus sur les jeux d'entrainements. Ceci est très probablement dù un surapprentissage, imputable au choix de 50% du corpus d'entrainement utilisé pour valider les paramètres. Nous renouvelerons prochainement nos

[6] Le classifieur utilisé, écrit en java, est disponible sur www.echarton.com/deft, associé au programme perl de traitement préalable

[7] Les valeurs de pourcentages sont discrètes et prises par incrément de 5

expériences en améliorant notre méthode (évaluation du classifieur sur un corpus d'entrainement segmenté en 4 éléments), pour confirmer nos meilleurs scores.
Ces tests nous ont confirmé que l'expression de l'opinion ne peut être évaluée statistiquement par similarité cosine ou mesure de probabilité, qu'en supprimant le bruit et en tentant de localiser au mieux les segments d'un document où s'exprime cette opinion. Ils nous incitent aussi à penser que les performances d'un classifieur par mesure de similarité ou de distance dans ce type de tâche, sont particulièrement dépendantes de la forme d'expression (complexité, richesse de vocabulaire) utilisée.

3.2 Resultats du modèle de régression logistique

Dans cette section, nous détaillons les résultats obtenus avec notre algorithme par régression logistique appliqué aux différents corpus proposés par DEFT07 (DEFT, 2007). Les outils utilisés pour expérimenter ce modèle sont :

- Microsoft Visual Studio 2005 et Visual C# 2005 [8] principalement pour extraires les mots et les insérer dans la base de données.
- Microsoft Access 2003 [9] en tant que moteur de base de données et pour exécuter les requêtes SQL.
- Statgrapichs 4.0 [10] utilisé pour l'analyse par régression.

3.2.1 Corpus 1 : Avoir A Lire, critiques de produits culturels

En utilisant les valeurs calculées des paramètres (voir la table TAB.3) du modèle le plus vraisemblable :

$$\theta_{i0} = \frac{e^W}{1+e^W}$$

où :

$$W = -0.27299 + 0.16782 z_{i0} - 0.23223 z_{i1} - 0.16084 z_{i2} + 0.00565 y_i$$

Ces valeurs s'inscrivent dans le cadre de notre hypothèse initiale. Il y a bien une corélation entre la probabilité de classifier en catégorie 0, avec la quantité critique de mots de catégorie 0 présents dans le document évalué (i.e. le parâmètre $\beta_0^0 > 0$). On constate également que dans le même temps, la quantité de mots critiques contenus par ailleurs dans d'autres classes fait décroitre la probabilité.

Paramètre	Estimation	Standard Error	Odds Ratio estimé
α_0	-0.272992	0.282966	
β_0^0	0.167818	0.0144647	1.18272
β_0^1	-0.232226	0.0273179	0.79276
β_0^2	-0.160838	0.0126263	0.85143
γ_0	0.005654	0.0016795	1.00567

TAB. 3 – Modèle de régression estimé pour la catégorie $j = 0$ dans le corpus Avoir A Lire, DEFT07

On déduit que si la "P-value" pour le modèle, dans l'analyse de la déviance (lire table TAB.4) est inférieure à 0,01, il existe une relation statistiquement significative entre les variables, dans un intervalle de confiance à 99% . Par ailleurs, la "P-Value" pour les données résiduelles est supérieure à 0,10, indiquant que le modèle n'est pas significativement plus mauvais que le meilleur modèle possible pour ces données, dans un intervalle de confiance de 90 % (ou supérieur).

Source	Deviance	Degrés de liberté	P-Value
Modèle	774.97	4	0.0000
Résiduel	971.12	2069	1.0000
Total	1746.1	2073	

TAB. 4 – Analyse de la déviance pour la catégorie $j = 0$ dans le corpus 1, DEFT07

[8] Informations sur http ://www.microsoft.com

[9] Informations sur http ://www.microsoft.com)

[10] Edité par StatPoint, Inc. link http ://www.statgraphics.com

En cherchant à évaluer comment le modèle peut être simplifié, nous notons que la plus haute valeur pour le test de vraisemblance (likelihood ratio test) est de 0,0007 (Voir table TAB.5), lorsqu'il est associé à y_i (le nombre total de mots contenus dans le texte). Sachant que la "P-value" est inférieur à 0,01, on considère que ce terme est statistiquement significatif à un intervalle de confiance de 99% . En conséquence, nous ne retirons aucune variable du modèle.

Facteur	χ_2	Degrés de liberté	P-Value
z_{i0}	170.08	1	0.0000
z_{i1}	87.97	1	0.0000
z_{i2}	231.91	1	0.0000
y_i	11.40	1	0.0007

TAB. 5 – Test de vraisemblance (Likelihood ratio tests) pour la catégorie $j = 0$ dans le corpus Avoir A Lire, DEFT07

Ces procédures de calcul et d'analyse du modèle de régression ont été répétées pour les classes 1 et 2 (*moyen* et *bon*). Le tableau TAB.6 montre la validité du modèle de régression appliqué au corpus Avoir A Lire. Le tableau TAB.7 présente les principaux aspects de ce corpus et nos scores finaux.

Categorie	P-Value du modèle	Pourcentage de déviance expliquée par le modèle
0	0.0000	44.38
1	0.0000	20.88
2	0.0000	30.80

TAB. 6 – Validité du modèle de régression pour le corpus Avoir A lire, DEFT07

Item	Valeur
$\|T\|$	2074
n	2074
$\|C\|$	3
$F - Score$ Corpus de test (cette méthode)	0.50
$F - Score$ Corpus de test (moyenne DEFT07)	0.48

TAB. 7 – Résultats finaux pour le corpus Avoir A Lire, DEFT07

3.2.2 Corpus 2 : JeuxVideo.com, Tests de jeux vidéo

Les résultats obtenus sur le corpus 2 sont présentés dans les tableaux qui suivent. Le tableau TAB.8 présente la validité du modèle de régression appliqué.

Category	P-Value du modèle	Pourcentage de déviance expliquée par le modèle
0	0.0000	88.02
1	0.0000	73.73
2	0.0000	90.30

TAB. 8 – Validité du modèle de régression pour le corpus JeuxVidéo.com, DEFT07

Le tableau TAB.9 présente les principales caractéristiques de notre modèle appliqué au corpus et le score final.

Item	Valeur
$\|T\|$	2537
n	60
$\|C\|$	3
$F - Score$ Corpus de test (cette methode)	0.46
$F - Score$ Corpus de test (moyenne DEFT07)	0.65

TAB. 9 – Résultats finaux pour le corpus JeuxVideo.com, DEFT07

3.2.3 Corpus 3 : Relectures d'articles de conférences

Les résultats pour le corpus "Relectures sont présentés dans les tables qui suivent. La table 10 présente la validité du modèle de régression appliqué.

Categorie	P-Value du modèle	Pourcentage de déviance expliquée par le modèle
0	0.0000	29.58
1	0.0000	24.11
2	0.0000	21.12

TAB. 10 – Validité du modèle de régression pour le corpus Relectures, DEFT07

Le tableau TAB.11 présente les principales caractéristiques de notre modèle appliqué au corpus et le score final.

Item	Value
$\|T\|$	881
n	881
$\|C\|$	3
$F - Score$ Corpus de test (cette méthode)	0.47
$F - Score$ Corpus de test (Moyenne DEFT07)	0.47

TAB. 11 – Résultats finaux pour le corpus Relectures, DEFT07

3.2.4 Corpus 4 : Débats parlementaires

Les résultats pour le corpus 4 sont présentés dans les tableaux qui suivent. Le tableau TAB.12 présente la validité du modèle de régression appliqué.

Categorie	P-Value du modèle	Pourcentage de déviance expliquée par le modèle
0	0.0000	24.20
1	0.0000	24.20

TAB. 12 – Validité du modèle de régression pour le corpus Débats, DEFT07

Le tableau 13 présente les principales caractéristiques de notre modèle appliqué au corpus, et le score final.

Item	Valeur
$\|T\|$	17299
n	1000
$\|C\|$	2
$F - Score$ Corpus de test (cette méthode)	0.55
$F - Score$ Corpus de test (moyenne DEFT07)	0.64

TAB. 13 – Résultats finaux pour le corpus Débats, DEFT07

3.3 Résultats du modèle par densité et compacité

Notre algorithme de recherche d'opinion par calcul de compacité repose sur la définition de sous-classes de mots centroïdes (ou "attracteurs"), c'est à dire susceptibles d'être entourés par un ensemble de mots porteurs d'une opinion. Ces centroïdes sont accompagnés par des sacs de mots que nous avons remplis de tous les mots qui les entourent (portée de $+/-$ 5 mots avant et après) dans le corpus d'aprentissage. Il existe un centroïde pour chaque sous corpus d'opinion (exemple *centroïde "article"* classe *bon*, *centroïde "article"* classe *mauvais*, etc).

Nous indiquerons ici les scores de précision et rappel obtenus pour chaque classe car ils soulignent un déséquilibre entre les performances des classifieurs pour chaque opinion : certaines classes sont très bien détectées et d'autres ont des scores extrèmement faibles. Il semble après quelques expériences complémentaires que ce déséquilibre soit lié au choix de sacs des mots non individualisés par classes. Ceci nous conduira à envisager l'hypothèse que les sacs de mots devraient être construits d'après les n mots centroïdes les mieux placés pour chaque classe (et non plus choisis sur tout le corpus, de manière inter-classes).
Les résultats F-Score obtenus ici sont ceux obtenus d'après le corpus d'entrainement, divisé en 50% pour l'apprentissage, et 50% pour le test. Nous n'avons pas soumis ces résultats lors de l'évaluation DEFT07. [11].

3.3.1 Résultats sur le corpus Débats

Les mots centroïdes retenus pour la constitution de sacs de mots, après suppression des mots outils par application d'antidictionnaire, et comptage des occurences de mots restants sont : $\{Loi, projet\}$.

Classe	0 (Défavorable)	1 (Favorable)	F-Score
Précision	0.55	0.48	0.51 (0.64)
Rappel	0.89	0.11	

TAB. 14 – Scores obtenus sur le corpus Débats (entre parenthèses, le score moyen obtenu par les participants de DEFT07

3.3.2 Résultats sur le corpus Relectures

Les mots centroïdes retenus pour la constitution de sacs de mots, après suppression des mots outils par application d'antidictionnaire, et comptage des occurences de mots restants sont : $\{Article, Papier\}$.

Classe	0 (Défavorable)	1 (Moyen)	2 (Favorable)	F-Score
Précision	0.54	0.09	0.66	0.42 (0.46)
Rappel	0.19	0.46	0.57	

TAB. 15 – Scores obtenus sur le corpus Relectures (entre parenthèses, le score moyen obtenu par les participants de DEFT07

3.3.3 Résultats sur le corpus Avoir à Lire

Les mots centroïdes retenus pour la constitution de sacs de mots, après suppression des mots outils par application d'antidictionnaire, et comptage des occurences de mots restants sont : $\{Film, Roman\}$.

Classe	0 (Défavorable)	1 (Moyen)	2 (Favorable)	F-Score
Précision	0.43	0.35	0.49	0.40 (0.48)
Rappel	0.14	0.10	0.86	

TAB. 16 – Scores obtenus sur le corpus Avoir à Lire (entre parenthèses, le score moyen obtenu par les participants de DEFT07

[11] Les logiciels utilisés pour réaliser ces expériences sont des prototypes écrits en perl, spécifiquement pour ce défi. Ils sont disponibles sur http :/www.echarton.com/deft)

4 Conclusions et perspectives

Nous avons considéré dès le début de nos expériences ce défi sous l'angle de son hétérogénéité. Tant dans la forme des corpus (quatre familles de textes issus de sources très variées), que dans la méthode d'évaluation des résultats (trois soumissions prises sous une forme atomique), ce défi invite à rechercher une solution neuve d'extraction d'une opinion, susceptible de généraliser ce type de tâche.
La confrontation des résultats obtenus avec une première méthode classique de la recherche documentaire - similarité cosine et probabilité de présence des mots - sur les corpus d'apprentissage, avec ceux obtenus par les deux autres méthodes - régression logistique et calcul de compacité - nous a semblé très encourageante, notamment si l'on considère que les résultats sont globalement proches de ceux obtenus en moyenne par l'ensemble des participants. Par ailleurs, les méthodes de régression logistique et de mesure de compacité n'ont encore - à notre connaissance - jamais été déployées pour classer des corpus d'après des "concepts" ou des "idées" (activité dont nous semble se rapprocher la thématique de l'opinion). De nombreuses possibilités d'améliorations restent à explorer.
Il reste par exemple possible d'améliorer considérablement le modèle de régression, en ajoutant d'autres variables explicatives, en passant d'une exploration des mots critiques à une étude des phrases critiques, ou encore en introduisant une utilisation des modèles n-grammes.
Il en va de même avec le modèle de mesure de densité et de calcul de compacité. Nous pourions enrichir le corpus avec un étiquetage morphosyntaxique orienté vers l'expression d'une idée, ou encore, affiner la recherche de mots centroïdes pour la construction des sacs de mots.
D'une manière générale, nous pensons que l'introduction d'un nouveau modèle statistique (la régression logistique), et l'adaptation d'un dispositif prévu à l'origine pour les systèmes de question-réponse à la classification par opinion, ouvre des pistes intéressantes dans le domaine de la classification par les idées.

5 Remerciements

Nous souhaitons adresser nos chaleureux remerciements à nos encadrants respectifs, Messieurs Philippe Michelon, et Jean-François Bonastre pour leurs encouragements et leur bienveillance dans le cadre de cette participation à DEFT07 qui s'éloigne singulièrement des recherches que nous sommes censés réaliser !
Nous remercions également Messieurs Frédéric Béchet et Juan Manuel Torres-Moreno, pour le temps qu'ils ont bien voulu nous consacrer.

Références

A. MARINICHEV, S. VYAZ'MIN I. D. (2005). A spectrophotometric study of solid. *Russian Journal of Applied Chemistry*, **vol 78 issue 10 p1662-1667**.

A. MENDITTO, D. LINHORST J. C.-N. B. (2006). The use of logistic regression to enhance risk assessment and decision making by mental health administrators. *The Journal of Behavioral Health Services and Research*, **vol 33 issue 2 p213-224**.

A. VALERO, E. CARRASCO E. A. (2006). Growth/no growth model of listeria monocytogenes as a function of temperature, ph, citric acid and ascorbic acid. *European Food Research and Technology*, **vol 224 issue 1 p91-100**.

B. JIANG, C. WANG P. C. (2004). Logistic regression tree applied to classify pcb golden finger defects. *The International Journal of Advanced Manufacturing Technology*, **vol 24 issue 7 p496-502**.

B. ROSENFELD C. L. (2005). Assessing violence risk in stalking cases : A regression tree approach. *Law and Human Behavior*, **vol 29 issue 3 p342-357**.

D. TESTI, A. CAPPELLO L. C. M. V. S. G. (2001). Comparison of logistic and bayesian classifiers for evaluating the risk of femoral neck fracture in osteoporotic patients. *Medical and Biological Engineering and Computing*, **vol 39 issue 6 p633-637**.

DEFT (2007). 3ème dÉfi fouille de textes, http ://deft07.limsi.fr/index.html.

FAVRE B., BECHET F. & NOCÉRA P. (2005). Robust named entity extraction from large spoken archives. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, p. 491–498, Vancouver, British Columbia, Canada : Association for Computational Linguistics.

G. ROBINS, P. PATTISON S. W. (1999). Logit models and logistic regressions for social networks : Iii. valued relations. *Psychometrika*, **vol 64 issue 3 p371-394**.

G. VENKATARAMAN, V. ANANTHANARAYANAN G. P. E. A. (2006). Morphometric sum optical density as a surrogate marker for ploidy status in prostate cancer : an analysis in 180 biopsies using logistic regression and binary recursive partitioning. *Virchows Archiv*, **vol 449**.

G. WU S. Y. (2006). Prediction of possible mutations in h5n1 hemagglutitins of influenza a virus by means of logistic regression. *Comparative Clinical Pathology*, **Vol 15**(issue 4), p255–261.

GILLARD L., BELLOT P. & EL-BÉZE M. (2006). Influence de mesures de densité pour la recherche de passages et l'extraction de réponses dans un système de questions-réponses. In *Actes de Coria 2006*, p. 193–204, Lyon, France.

J. COLWELL A. R. (2005). Hot surface ignition of automotive and aviation fluids. *Fire Technology*, **vol 41 issue 2 p105-123**.

K. CHAU J. C. (2005). Regional bias of landslide data in generating susceptibility maps using logistic regression : Case of hong kong island. *Landslides*, **vol 2 issue 4 p280-290**.

LUHN H. (1958). The automatic creation of literature abstracts. In *IBM Journal of research and Development*.

M. COLLINS, R. SCHAPIRE Y. S. (2002). Logistic regression, adaboost and bregman distances. *Machine Learning*, **vol 48 issue 1 p253-285**.

MAGES (2006). Mages (modules d'aide à la gestion des sillons), http ://awal.univ-lehavre.fr/ lmah/mages/.

N. DOLSAK M. D. (2006). Investments in global warming mitigation : The case of activities implemented jointly. *Policy Sciences*, **vol 39 issue 3 p233-248**.

N. SAHOO H. P. (1999). Integration of sparse geologic information in gold targeting using logistic regression analysis in the hutti maski schist belt, raichur, karnataka, india. a case study. *Natural Resources Research*, **vol 8 issue 3 p233-250**.

OEXLE K. (2006). Biochemical data in ornithine transcarbamylase deficiency (otcd) carrier risk estimation : logistic discrimination and combination with genetic nformation. *Journal of Human Genetics*, **vol 51 issue 3 p204-208**.

RODRIGUEZ A. A. (2001). Logistic regression and world income distribution. *International Advances in Economic Research*, **vol 7 issue 2 p231-242**.

S. STACK O. T. (1997). Suicide risk among correctional officers : A logistic regression analysis. *Archives of Suicide Research*, **vol3 issue 3 p183-186**.

S. WASSERMAN P. P. (1996). Logit models and logistic regressions for social networks : I. an introduction to markov graphs andp. *Psychometrika*, **vol 61 issue 3 p401-425**.

SALTON G. & BUCKLEY C. (1988). Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*.

T. CLEOPHAS A. (2006). Post-hoc analyses in clinical trials, a case for logistic regression analysis. *Statistics Applied to Clinical Trials*, **p 187-191**.

THIERRY SPRIET, FRÉDÉRIC BÉCHET, MARC .EL-BÈZE. C. D. L. L. K. (1996). Traitement automatique des mots inconnus. Avignon, France.

VOORHEES E. (2006). Overview of the trec 2002 question answering track. In *Actes de "the 11th Text REtrieval Conference"*, Gaithersburg, Maryland, USA : TREC.

W. WILSON, K. DAY E. H. (1996). Predicting the extent of damage to conifer seedlings by the pine weevil (hylobius abietis l.) : a preliminary risk model by multiple logistic regression. *New Forests*, **vol 12 issue 3 p203-222**.