VENCE : un modèle performant d'extraction de résumés basé sur une approche d'apprentissage automatique renforcée par de la connaissance ontologique

Thèse

Jésus Antonio Motta

Doctorat en informatique
Philosophiæ Doctor (Ph.D.)

Québec, Canada

Résumé

De nombreuses méthodes et techniques d'intelligence artificielle pour l'extraction d'information, la reconnaissance des formes et l'exploration de données sont utilisées pour extraire des résumés automatiquement. En particulier, de nouveaux modèles d'apprentissage automatique semi supervisé avec ajout de connaissance ontologique permettent de choisir des phrases d'un corpus en fonction de leur contenu d'information. Le corpus est considéré comme un ensemble de phrases sur lequel des méthodes d'optimisation sont appliquées pour identifier les attributs les plus importants. Ceux-ci formeront l'ensemble d'entrainement, à partir duquel un algorithme d'apprentissage pourra abduire une fonction de classification capable de discriminer les phrases de nouveaux corpus en fonction de leur contenu d'information. Actuellement, même si les résultats sont intéressants, l'efficacité des modèles basés sur cette approche est encore faible notamment en ce qui concerne le pouvoir discriminant des fonctions de classification. Dans cette thèse, un nouveau modèle basé sur l'apprentissage automatique est proposé et dont l'efficacité est améliorée par un ajout de connaissance ontologique à l'ensemble d'entrainement. L'originalité de ce modèle est décrite à travers trois articles de revues. Le premier article a pour but de montrer comment des techniques linéaires peuvent être appliquées de manière originale pour optimiser un espace de travail dans le contexte du résumé extractif. Le deuxième article explique comment insérer de la connaissance ontologique pour améliorer considérablement la performance des fonctions de classification. Cette insertion se fait par l'ajout, à l'ensemble d'entraînement, de chaines lexicales extraites de bases de connaissances ontologiques. Le troisième article décrit VENCE[1], le nouveau modèle d'apprentissage automatique permettant d'extraire les phrases les plus porteuses d'information en vue de produire des résumés. Une évaluation des performances de VENCE a été réalisée en comparant les résultats obtenus avec ceux produits par des logiciels actuels commerciaux et publics, ainsi que ceux publiés dans des articles scientifiques très récents. L'utilisation des métriques habituelles de rappel, précision et F_measure ainsi que l'outil ROUGE a permis de constater la supériorité de VENCE. Ce modèle pourrait être profitable pour d'autres contextes d'extraction d'information comme pour définir des modèles d'analyse de sentiments.

[1] VENCE: **V**ectorial **E**nhancing **N**oise free by means of **C**onceptual **E**xtracting

Abstract

Several methods and techniques of artificial intelligence for information extraction, pattern recognition and data mining are used for extraction of summaries. More particularly, new machine learning models with the introduction of ontological knowledge allow the extraction of the sentences containing the greatest amount of information from a corpus. This corpus is considered as a set of sentences on which different optimization methods are applied to identify the most important attributes. They will provide a training set from which a machine learning algorithm will can abduce a classification function able to discriminate the sentences of new corpus according their information content. Currently, even though the results are interesting, the effectiveness of models based on this approach is still low, especially in the discriminating power of classification functions. In this thesis, a new model based on this approach is proposed and its effectiveness is improved by inserting ontological knowledge to the training set. The originality of this model is described through three papers. The first paper aims to show how linear techniques could be applied in an original way to optimize workspace in the context of extractive summary. The second article explains how to insert ontological knowledge to significantly improve the performance of classification functions. This introduction is performed by inserting lexical chains of ontological knowledge based in the training set. The third article describes VENCE[2], the new machine learning model to extract sentences with the most information content in order to produce summaries. An assessment of the VENCE performance is achieved comparing the results with those produced by current commercial and public software as well as those published in very recent scientific articles. The use of usual metrics recall, precision and F_measure and the ROUGE toolkit showed the superiority of VENCE. This model could benefit other contexts of information extraction as for instance to define models for sentiment analysis.

[2] VENCE: **V**ectorial **E**nhancing **N**oise free by means of **C**onceptual **E**xtracting

Table de matières

Liste des tableaux

Chapitre 4

Chapitre 5

Chapitre 6

Liste des Figures

Chapitre 3

Chapitre 4

Chapitre 5

Chapitre 6

Cette thèse est dédié à mes parents Jesus Antonio y Mariela, ma fille María del Mar, mes sœurs Luz Marina, Romelia Blanca Mercedes, Mariela, Angela María, María Teresa, Gloria Esperanza, Clara Inés et Carolina et mon épouse et compagne de tous les temps Isabel Cristina.

Remerciements

Je remercie toutes les personnes du département d'informatique et de génie logiciel qui ont contribué positivement d'une manière ou d'une autre à mes études. Premièrement, je remercie mes deux directrices de thèse : Professeure Nicole Tourigny pour son dévouement, sa patience et ses sages conseils, ma directrice au début de mes études doctorales et au début de mes recherches sur le sujet de l'extraction de l'information et de résumé automatique. Ses qualités professionnelles et personnelles ont tout mon respect et ma reconnaissance. Je remercie Professeure Laurence Capus, mon actuelle directrice de thèse, qui également avec son dévouement et ses conseils intelligents a donné à mon travail, jusqu'à sa réussite, un contenu scientifique approfondi dans tous les articles produits. Elle a aussi mon admiration et ma reconnaissance pour ses grandes qualités de supervision. Je leur exprime à toutes les deux ma profonde gratitude, surtout d'avoir cru en mes capacités et mon travail.

Je remercie aussi Professeur Luc Lamontagne qui a fait partie du comité d'encadrement de thèse et tous les autres professeurs et professeures du département d'informatique et de génie logiciel qui ont contribué, avec leurs conseils et avec leurs cours, à la réussite de mes études.

Je tiens en particulier à remercier Jacques Ladouceur, professeur renommé, spécialisé en traitement du langage naturel, du Département de langues, linguistique et traduction de l'Université Laval pour ses conseils judicieux et sages, et ses commentaires élogieux sur mon travail.

Je remercie aussi les membres du jury de la thèse, des professeurs spécialistes dans le traitement automatique du langage et de l'apprentissage automatique, qui ont accepté d'être membres du jury, pour leurs importants apports et commentaires : Professeur Jacques Ladouceur, du Département de langues, linguistique et traduction de l'Université Laval, Professeur Richard Khoury, du département de génie logiciel de l'université Lakehead (Thunder Bay, Ontario), Professeur Horacio Saggion, du Département d'information et technologies de la communication de l'Université Pompeu Fabra (Barcelone), ma directrice de recherche Professeure Laurence Capus et ma codirectrice de recherche Professeure

Nicole Tourigny, professeure retraitée associée du Département d'informatique et de génie logiciel de la Faculté de sciences et de génie logiciel.

Je dois mentionner que cette thèse a été développée partiellement avec le soutien financier du Conseil de recherche en génie et sciences naturelles du Canada (CRSNG). Pour les membres de cette institution, mes sincères remerciements.

Et en terminant, je dois donner un grand merci au Tout Créateur, architecte de mes idées et réalisations!

Avant - Propos

Cette thèse avec insertion d'articles va bien au-delà de l'état de l'art des méthodes et des techniques d'extraction d'information, de reconnaissance des formes et d'extraction de données. En effet, elle a permis de créer un nouveau modèle d'apprentissage automatique avec l'intégration de connaissance ontologique pour extraire les phrases les plus porteuses d'information en vue de produire des résumés de textes, afin de rendre ce processus plus efficace.

J'ai rédigé les trois articles insérés dans cette thèse à partir de mes idées, mes recherches, mes réflexions et mes différentes expérimentations menées tout au long de mon programme. Dans les trois cas, j'en suis le principal auteur. Les deux co-auteures qui sont respectivement ma directrice de recherche et ma codirectrice de recherche m'ont encadré tout au long de cette démarche.

Parmi les trois articles insérés, deux ont déjà été publiés respectivement dans les revues "Computer and Information Science" (Vol. 3, No. 3, pp. 131-138, Aug. 2011) et "Journal of Intelligent Learning Systems and Applications" (Vol. 5 No. 6, pp. 58-72, Nov. 2012). Le premier article porte sur l'étude de différents modèles pour l'optimisation des espaces de recherche afin de déterminer ceux qui donnent les meilleurs résultats dans le contexte du résumé automatique. Le deuxième montre comment faire l'abduction des fonctions d'apprentissage renforcées avec de la connaissance ontologique. Les coauteurs et moi-même avons choisi délibérément de publier ces deux articles dans deux revues à accès libre afin d'offrir un accès plus large à notre travail de recherche. De plus, le processus de révision était assez court et permettait d'avoir un rapide retour de la part des réviseurs, donc une diffusion plus rapide auprès des chercheurs du domaine. Enfin, l'accès libre permettant que ces articles soient plus facilement accessibles, j'ai pu constater qu'ils ont été largement téléchargés d'après les statistiques reçues à ce sujet. Je tiens à préciser que ces deux revues sont de renommée internationale et considérées comme réputées ; elles sont d'ailleurs répertoriées dans les plus importantes bases de données d'articles scientifiques indexées telles que : CiteSeerX (ancienne CiteSeer), créée et maintenue par l'Université de Pennsylvanie et parrainée par la NASA et la NSF (National Science Foundation), Scirus, créée par l'éditeur Elsevier, et Scholar Google. De même, ces revues répondent aux

critères de qualité fixés par des organisations comme l'OASPA (Open Access Scholarly Publishers Association), le COPE (Committee on Publication Ethics) et la STM (International Association of Scientific, Technical & Medical Publishers). Une liste de revues en accès libre contenant les titres de celles considérées comme des prédateurs potentiels, possibles ou probables a été publiée afin de mettre en garde les futurs contributeurs (List of Standalone Journals)[3]. Les deux revues dans lesquelles sont publiés deux articles de cette thèse ne font pas partie de cette liste. Le troisième article à soumettre décrit un nouveau modèle d'extraction de résumés basé sur une approche d'apprentissage automatique semi-supervisé renforcé par de la connaissance ontologique. Pour protéger son développement et sa mise en œuvre, j'ai demandé la délivrance d'un brevet, demande actuellement en cours de traitement par le Vice-rectorat à la recherche et à la création de l'Université Laval. C'est d'ailleurs pour cette raison que la soumission du troisième article a été retardée car la divulgation publique d'une invention limite fortement la possibilité d'obtenir des brevets dans de nombreux pays.

Cette thèse étant composée d'articles pour des revues scientifiques, ceci implique nécessairement quelques sections similaires au contenu des articles, notamment celles liés à l'introduction et la justification des méthodes utilisées, ainsi que celles qui se rapportent à la mesure de la performance. Ces répétitions sont souvent nécessaires pour expliciter ou justifier le contenu plus synthétique et/ou plus spécialisé des articles.

Pour ce travail de recherche, il me semble important de mentionner que j'ai utilisé, et souvent adapté au contexte, des techniques, des méthodes et des procédés mathématiques, algébriques, d'optimisation probabiliste et statistique ainsi que des techniques d'exploration de données, d'extraction d'information et de reconnaissance de formes. Une telle démarche a demandé beaucoup de travail pour réaliser une application rigoureuse des concepts mathématiques sous-jacents. Les différents programmes utilisés pour l'expérimentation et l'évaluation de la performance ont requis plusieurs logiciels spécialisés (TreeTagger, SPSS, Mathematica, R, Minitab, MatLab, ROUGE) et langages de programmation (Visual Basic, Python, Perl, C, Java), ce qui a nécessité de nombreuses périodes d'apprentissage.

[3] http://scholarlyoa.com/individual-journals/

Dans les trois articles qui composent l'épine dorsale de ma recherche, je propose différentes méthodes pour l'application de techniques afin d'obtenir des modèles d'optimisation de l'espace d'états. Ensuite, ces modèles servent à construire des fonctions d'apprentissage renforcé avec de la connaissance ontologique pour les appliquer à l'extraction de résumés automatiques.

Mes objectifs ont été largement atteints et j'ai contribué à l'avancement du domaine de l'extraction d'information et en particulier du processus de construction automatique de résumés par extraction, comme le montrent les résultats de l'application du modèle présenté dans le troisième article. Ces résultats dépassent largement ceux obtenus en appliquant des logiciels actuels du domaine commercial et du domaine public, de même que ceux obtenus dans des travaux de recherche très récents.

« *Pluralitas non est ponenda sine necessitate* »

Guillaume d'Ockham

CHAPITRE 1. INTRODUCTION GÉNÉRALE

Notre thèse s'inscrit dans le domaine de l'intelligence artificielle, et plus particulièrement dans le domaine du résumé automatique. Notre objectif était de construire un modèle performant pour identifier les phrases porteuses d'information et celles non porteuses d'information dans un ensemble de documents. Pour atteindre notre objectif, nous nous sommes intéressé à l'approche par apprentissage automatique semi-supervisé pour apprendre à discriminer les phrases selon un ensemble binaire contenant des phrases « importantes » et des phrases « non-importantes ». On entend par phrases « importantes » ou significatives celles qui sont les plus porteuses de sens ou d'information et par phrases « non importantes » ou non significatives, toutes les autres. Pour améliorer cette tâche et ainsi augmenter la performance de notre modèle, nous avons optimisé les espaces de recherche et ajouté de la connaissance ontologique.

Dans ce chapitre, nous présentons un aperçu général de notre travail. Nous commençons par définir le contexte de la recherche et ses motivations. Nous présentons la problématique qui nous a amené à spécifier nos objectifs de recherche et la méthodologie suivie. Nous donnons ensuite les résultats obtenus, soit notre modèle et les contributions de recherche. Nous terminons le chapitre avec le plan de la thèse.

1.1 Contexte et motivations

L'intérêt de la recherche pour les méthodes et les techniques d'automatisation pour obtenir des résumés de textes a commencé dans les années 60 avec une remarquable croissance au milieu des années 80 (Jones, 2007). Cet intérêt se poursuit encore ces dernières années avec de nouveaux travaux (Nenkova et McKeown, 2011). La croissance exponentielle des sources d'information électroniques et la nécessité pour un utilisateur de récupérer des informations parmi ces nombreuses sources ont créé un urgent impératif de développer des systèmes automatisés efficaces pour la récupération de l'information essentielle (Sharan et Imran, 2009).

Aujourd'hui, n'importe quel utilisateur du réseau Internet est confronté au problème à savoir : la consultation d'un grand nombre de documents disponibles (souvent des milliers) qui sont proposés suite à une recherche. Cet utilisateur devra d'abord décider quels sont les documents à examiner, puis quelles seront les informations pertinentes dans chacun des

documents relatifs aux sujets d'intérêt. Ces actions sont souvent difficilement concevables pour un être humain dans un temps raisonnable. Il devient donc nécessaire de développer des systèmes capables d'extraire les informations les plus pertinentes, à partir d'un document ou d'un ensemble de documents, et de produire un extrait résumé qui reflète le plus fidèlement possible son essence, c'est-à-dire en conservant le plus d'information possible. Ce problème de résumé automatique par extraction d'information fait partie des défis de l'intelligence artificielle et plus précisément du traitement automatique du langage naturel. Certaines méthodes utilisées relèvent du domaine du raisonnement automatique avec l'apprentissage automatique basé sur des symboles.

1.2 Problématique

Le résumé automatique revient à condenser un document source, ou plusieurs documents, dans une version plus courte en préservant son contenu d'information (Jones, 1999; Sharan et Imran, 2009). Un résumé peut être produit de manière générique; il donne alors une idée générale du contenu du document, ou selon des mots-clés définis par l'utilisateur, en présentant l'information la plus pertinente en accord avec ces mots-clés (Goldstein *et al.*, 1999). Les méthodes de résumés actuelles peuvent être divisées en deux grandes catégories, extractives et non extractives (ou abstractives), qui correspondent également aux méthodes de surface et de profondeur dans la terminologie linguistique (Mani, 2001). Un résumé généré de manière extractive est composé d'un ensemble de phrases du document original sélectionnées à l'aide de méthodes statistiques et/ou heuristiques, basées sur l'entropie d'information des phrases (non importantes, redondantes). Un résumé d'abstraction est construit grâce à une analyse sémantique permettant d'interpréter le texte et de trouver des nouveaux concepts pour le décrire dans un texte de synthèse. Cette méthode implique donc un traitement linguistique d'un certain niveau d'élaboration. La génération automatique d'un résumé abstractif se compose de trois étapes distinctes: l'interprétation du document source pour obtenir une représentation de celui-ci, la transformation de la représentation de la source et la production du texte de synthèse (García-Hernandez *et al.*, 2008; Jones, 2007; Mani, 2001). Il est clair que pour répondre aux besoins de l'utilisateur confronté à une recherche dans une multitude de documents, le résumé extractif est privilégié. En effet, il permet de donner un aperçu plus rapide de l'information contenue dans ces documents,

sans nécessiter de nombreux processus linguistiques coûteux en temps et en ressources. Le résumé extractif est donc une alternative intéressante, robuste et indépendante de la langue face au résumé d'abstraction (García-Hernandez *et al.*, 2008).

Un problème crucial qui se pose dans la génération automatique d'un résumé par extraction est la détection de l'information la plus importante dans le document source (García-Hernandez *et al.*, 2008). Différentes méthodes ont été utilisées jusqu'à présent, plus ou moins performantes selon des mesures effectuées par des systèmes évaluateurs spécialisés. Ces mesures d'évaluation sont basées sur la précision (precision), soit le nombre de phrases correctes vs le nombre total de phrases choisies, et le rappel (recall), soit le nombre de phrases correctes sélectionnées vs le nombre total de phrases correctes (Korfhage, 1997). Certaines méthodes d'extraction utilisent les ontologies ou des connaissances ontologiques en général, pour analyser des termes et leurs relations (Yu *et al.*, 2006). Plus récemment, d'autres méthodes ont été rapportées. Elles utilisent des algorithmes d'apprentissage pour induire les descriptions de concepts basées sur la synthèse du document à résumer et construire un ensemble de cas d'entraînement, dont ses éléments sont classifiés en deux sous-ensembles : importants ou non importants (Larocca *et al.*, 2002; Sharan et Imran, 2009; Shen et Li, 2011). Ceci revient donc à traiter un problème de classification. En effet, dans l'approche par apprentissage automatique, on prépare un espace de travail à partir d'un corpus de documents, soit un ensemble de phrases. En réduisant cet ensemble de phrases, on obtient un ensemble d'entraînement utilisé en entrée par un algorithme d'apprentissage automatique. En sortie, une fonction de classification est apprise et peut être appliquée à de nouveaux corpus pour extraire les phrases importantes.

Notre analyse des modèles basés sur l'apprentissage automatique nous a permis de constater qu'il était nécessaire d'introduire des moyens pour améliorer la performance des fonctions de classification utilisées. Plus précisément, les méthodes statistiques qui sont à la base de beaucoup de logiciels actuels présentent une performance relativement basse et même à plusieurs reprises manquent de précision et de rappel de manière notable, comme nous avons pu le constater dans le logiciel FreeSummarizer par exemple. Nous avons également noté une réduction insuffisante de la redondance ce qui entraîne une baisse de la qualité du choix des phrases extraites. Malgré l'utilisation de la connaissance ontologique

et des transformations linéaires et par conséquence son amélioration évidente, les méthodes utilisées donnent des résultats inférieurs à ceux obtenus avec un logiciel commercial populaire. Il y a aussi une importante disparité entre les résumés obtenus par extraction et ceux produits manuellement, problème qui serait lié au traitement de l'espace de travail (Ikonomakis *et al.*, 2005). Enfin, quelques résultats importants avec l'introduction d'algorithmes d'apprentissage automatique montrent la nécessité d'approfondir encore plus l'étude des moyens pour optimiser et choisir les ensembles d'entraînement (Shen et Li, 2011).

1.3 Objectifs et méthodologie

L'objectif général de cette thèse était de proposer un nouveau modèle d'apprentissage automatique performant pour un contexte de résumé automatique. Pour répondre à cet objectif de performance, nous avons misé sur l'optimisation de l'espace de travail. Dans le cas du résumé automatique, cet espace de travail correspond à une représentation d'un corpus de documents, un ensemble de phrases, qui sert de référence pour apprendre la fonction de classification qui sera appliquée pour résumer de nouveaux documents. De manière générale, l'espace de travail est très grand et très entropique, c'est-à-dire, avec beaucoup de bruit et une abondance de concepts non importants, créant ainsi le problème bien connu du fléau de la dimension (Smale, 1997). En effet, des données trop dispersées ne facilitent pas de bonnes estimations et par conséquent de bons modèles pour la classification. On peut résoudre ce problème en utilisant des heuristiques, comme la réduction de l'espace de travail au moyen d'approximations linéaires ou non linéaires, qui peuvent produire un sous-ensemble de l'ensemble original ou une nouvelle série de concepts à partir de l'ensemble original. Nous avons donc proposé d'optimiser l'espace de travail par l'introduction de chaînes lexicales de bases de connaissances ontologiques pour former un espace conceptuel utilisable par des algorithmes d'apprentissage linéaires et non linéaires. Notre hypothèse est qu'il est possible d'obtenir un ensemble d'apprentissage renforcé grâce à la connaissance ontologique pour sélectionner ou transformer les caractéristiques de cet ensemble, qui constitue la base pour la construction du modèle d'apprentissage automatique. Nous nous sommes efforcé de réduire l'espace de travail et de définir une heuristique efficace. Pour cela, nous avons testé différents moyens, soit

l'identification des valeurs singulières (Golub et Kahan, 1965; Golub et Van Loan, 1996) et des composantes principales (Jolliffe, 2002), l'analyse de facteurs (Golub et Van Loan, 1996; Kim et Mueller, 1978), les k-moyennes (Hastie *et al.*, 2009) et les cartes auto-adaptatives (Kohonen, 1990) de l'espace conceptuel créé.

Après avoir sélectionné les moyens les plus performants pour l'optimisation d'espaces de travail et avoir testé différents modèles pour abduire des fonctions d'entraînement renforcées avec la connaissance ontologique, nous avons mis au point un nouvel algorithme d'introduction de connaissance ontologique basé sur différents niveaux de synonymie et de mesures de la similarité. Nous avons ensuite vérifié l'impact de cette introduction en mesurant la quantité de variance ajoutée et la quantité d'information gagnée, en calculant différentes métriques telles que la précision, le rappel et la F_measure (Manning *et al.*, 2008) ainsi qu'en évaluant la sensibilité des modèles par l'intermédiaire des courbes ROC (Lasko *et al.*, 2005).

Pour finir, nous avons conçu notre modèle de classification global pour construire des résumés à partir de nouveaux documents et mesuré sa performance. Pour cela, nous avons introduit et testé plusieurs algorithmes connus dans les domaines de l'apprentissage automatique, l'exploration de données et la reconnaissance de formes: la machine à vecteurs de support (Cortes et Vapnik, 1995; Vapnik, 1999), le classeur bayésien naïf (Hastie *et al.*, 2009), le perceptron multicouche (Rosenblatt, 1957; Rumelhart *et al.*, 1986), les réseaux neuronaux de fonction de base radiale (Buhmann, 2003), les arbres aléatoires (Breiman, 2001) et la régression logistique (Agresti, 2007). En testant notre modèle avec plusieurs algorithmes pour l'apprentissage automatique, nous nous sommes assuré de ne pas créer de biais dans les résultats et de déterminer le ou les algorithmes qui devraient être privilégiés pour finaliser notre modèle.

1.4 Résultats et contribution

Nous avons obtenu un nouveau modèle basé sur l'apprentissage automatique semi supervisé, que nous avons appelé VENCE (***V**ectorial **E**nhancing **N**oise free by means of **C**onceptual **E**xtracting*), dont sa performance a été mesurée au moyen du logiciel ROUGE (Lin, 2004). Ce modèle original offre les moyens de préparer un espace de travail plus

efficace, qui correspond à une matrice renforcée obtenue à partir de l'ensemble des phrases d'un corpus. Il permet également d'élaborer un ensemble d'entraînement raffiné et optimisé à partir de cette matrice. Ensuite, cet ensemble d'entraînement peut être utilisé par un algorithme d'apprentissage pour abduire une fonction de classification. Ainsi obtenue, cette dernière permet d'extraire les phrases les plus importantes de nouveaux corpus en vue de les résumer, soit celles les plus porteuses d'information ou plus significatives.

Les différents moyens mis en œuvre dans notre modèle s'avèrent plus efficaces que ceux des modèles actuels que nous avons pu utiliser pour la comparaison. Les résultats obtenus dépassent largement ceux produits en appliquant des logiciels commerciaux et publics sur les mêmes corpus. De même, les résultats sont supérieurs à ceux de travaux similaires publiés récemment. Plus précisément, les résultats montrent que les valeurs de rappel et de précision obtenues avec VENCE sont proches de 84% et 59% respectivement. De plus, les valeurs obtenues avec VENCE comparées à celles d'autres travaux de recherche similaires sont au moins deux fois supérieures et davantage dans certains cas. Enfin, la performance de VENCE comparée à celles de logiciels existants est au moins 33 % supérieure allant jusqu'à 250 % supérieure.

La contribution majeure de cette thèse est certes le modèle original d'extraction de phrases pour produire des résumés. Toutefois, des contributions annexes peuvent être mises en évidence et pourraient être utilisées dans d'autres modèles ou d'autres contextes. En effet, nous avons appliqué de manière originale des techniques linéaires pour optimiser l'espace de travail et nous avons montré leur efficacité (Motta *et al.*, 2012). Ces techniques n'avaient pas encore été utilisées pour cet usage et nous avons dû les adapter au contexte de l'approche par apprentissage automatique du résumé extractif. De même, nous avons proposé une manière originale d'obtenir un ensemble d'entraînement à partir d'un espace de travail optimisé. L'algorithme mis en œuvre pour extraire de la connaissance ontologique et l'ajouter à l'ensemble d'entraînement est totalement nouveau (Motta *et al.*, 2011), ainsi que la manière dont nous avons sélectionné les phrases les plus porteuses d'information. Des techniques d'exploration de données et de reconnaissance de formes ont été appliquées pour le processus de classification avec succès. Or, une telle application n'existait pas auparavant. Nous considérons avoir apporté une nouvelle méthode

d'apprentissage automatique semi-supervisé, une forme d'apprentissage automatique encore jeune. Pour finir, nous pouvons conclure que le modèle proposé offre une nouvelle représentation d'un espace de travail pour le contexte du traitement automatique des langues en général.

1.5 Plan de la thèse

Cette thèse est organisée comme suit. Le chapitre 2 présente une analyse des méthodes et des techniques extractives utilisées pour obtenir des résumés ainsi que la problématique dégagée de cette analyse. Le chapitre 3 expose nos objectifs de recherche et la démarche suivie pour réaliser ces objectifs. Dans le chapitre 4, nous faisons une étude des différentes techniques d'optimisation des espaces de recherche pour l'apprentissage automatique dans un contexte de résumé automatique, soit le premier article (Motta *et al.*, 2012). Le chapitre 5 présente la façon dont nous avons introduit la connaissance ontologique pour obtenir des ensembles d'entraînement renforcés, soit le deuxième article (Motta *et al.*, 2011). Le chapitre 6 décrit notre modèle, VENCE, pour extraire des phrases importantes en vue de produire des résumés, ainsi que son évaluation, soit le troisième article. Le chapitre 7 est dédié à la conclusion de notre travail de recherche et à la proposition de quelques perspectives d'avenir.

CHAPITRE 2. État de l'art et problématique

Les méthodes de résumé extractif

L'objectif de cette section est de faire une synthèse des principales méthodes et techniques qui reflètent l'état de la recherche connue jusqu'à présent dans le domaine ciblé. Nous nous sommes en effet concentré sur les travaux réalisés pour l'obtention automatique de résumés extractifs, qui consistent d'abord en extraire un ensemble de phrases du document original, sélectionnées à l'aide de méthodes statistiques, heuristiques ou basées sur des mesures de la quantité d'information contenues dans les phrases.

Ces méthodes peuvent être regroupées en deux grands courants, soit l'approche statistique et l'approche par apprentissage automatique. Ainsi, dans les sections qui suivent, différents développements, techniques et méthodes extractifs seront discutés à travers les principes de l'approche statistique, l'approche statistique enrichie et l'approche par apprentissage automatique.

2.1 Les principes de l'approche statistique

Cette approche est largement utilisée depuis les origines de l'automatisation du résumé (Jones, 1999; Jones, 2007). En général, elle consiste à assigner un poids aux phrases du document ou des documents sources en fonction de sa fréquence dans la source, ou de sa fréquence inverse, ou en utilisant des heuristiques pour le calcul d'attributs ou même en essayant de construire la structure sémantique du document. Les phrases sont ensuite ordonnées par poids et extraites selon leur valeur et la taille souhaitée du résumé.

En traitant la phrase indépendamment, on peut créer un problème de redondance. Pour résoudre ce problème, on utilise des méthodes comme par exemple la méthode d'importance marginale maximale (Carbonell et Goldstein, 1998), qui essaie de réduire la redondance en maintenant la pertinence de la recherche dans les documents réorganisés et en sélectionnant des passages adéquats pour le résumé.

Dans le cas de résumés construits à partir de plusieurs documents, les méthodes utilisées identifient des ensembles de documents avec un contenu semblable ou en rapport. Ces regroupements par sujets ou sous-sujets sont obtenus par des assemblages de données lexicales en précisant leurs caractéristiques (Radev *et al.*, 2000), ou en appliquant des algorithmes (Fahim *et al.*, 2009; Zhong, 2005). Les documents individuels ou les phrases sont comparés ensuite avec les vecteurs sujets en fonction d'une valeur assignée. Un

regroupement de documents sur un sujet est généralement pris comme base pour construire le résumé et on présume que ce résumé tient compte des sous-sujets statistiquement identifiés.

Par rapport à des logiciels pour résumer à partir d'une requête on analyse un programme qui est à la fois un cadre de référence pour la récupération d'information d'un ensemble de documents. Ce logiciel est conçu pour, qu'à partir d'une requête, on classe un ensemble de documents par sujet et qu'un résumé soit produit pour chaque sujet (Dunlavy *et al.*, 2007). Ils présentent la méthode QCS (Query, Clustering and Summarizing) qu'emploie un modèle d'espace vectoriel pour représenter un ensemble de documents (Salton *et al.*, 1975). Le poids des termes est fait de manière locale et de manière globale en utilisant la fréquence des termes, ainsi que leur fréquence inverse.

Il existe des méthodes pour récupérer un ensemble de documents qui coïncident mieux avec une requête, c'est–à-dire pour effectuer le regroupement des documents par sujet et créer un résumé correspondant à la requête. Par exemple, la méthode LSI (Latent Semantic Indexing) dérive la structure sémantique latente du document représenté par une matrice (Deerwester *et al.*, 1990). Cette matrice correspondant à une transformation du document original en un ensemble de vecteurs de base linéairement indépendants exprimant ce qui est le plus important dans les sujets du document. La transformation peut capturer des relations entre les termes, donc des termes et des phrases peuvent être groupés dans une base sémantique et pas seulement sur la base de mots. D'autres méthodes ont été utilisées pour mettre en application ce principe. Citons notamment les méthodes des plus proches voisins sphériques (Fahim *et al.*, 2009), des modèles de Markov cachés, des k plus proches voisins (Shakhnarovish *et al.*, 2005) et la décomposition QR (Golub et Van Loan, 1996).

Les travaux de Rogati et Yang (2002) présentent une étude sur 100 variantes des principales méthodes de sélection d'attributs de type filtre. Ces variantes sont basées sur les algorithmes de classification de Bayes (Hastie *et al.*, 2009), de Rocchio (Manning *et al.*, 2008), des plus proches voisins et de la machine à vecteur de support (Bosser *et al.*, 1992). Les méthodes testées, à la fois computationnelles et évolutives, ont une bonne performance en les comparants par exemple avec ceux obtenus par Yang et Pedersen (1997). Cette étude avait pour but de répondre aux questions suivantes :

- Vu la grande variance des collections de texte, quelles méthodes se sont le mieux comportées ?
- La combinaison de méthodes non reliées augmenterait-elle la performance ?

Diverses méthodes de sélection d'attributs utilisées ont été prises de Yang et Pedersen (1997), comme celle basée sur la fréquence d'un terme dans un document, le gain d'information ou le test du $\chi 2$. L'étude a montré que les méthodes de sélection d'attributs basées sur le test du $\chi 2$ obtiennent une performance qui dépasse de manière consistante celles des méthodes de sélection d'attributs qui utilisent d'autres filtres.

La fréquence d'un terme, notée TF, et la fréquence inverse de document, notée TF-IDF ont été les premières méthodes avec assignation de poids utilisées pour mesurer l'importance des mots dans un document. Elles sont toujours utilisées de nos jours dans diverses méthodes et approches en combinaison avec des mesures de similitude et des techniques d'apprentissage supervisé et non supervisé (Berger *et al.*, 2000). Dans son travail sur la force de la mesure TF-IDF, Ramos (2004) a examiné le résultat obtenu après l'application de cette mesure pour déterminer les mots les plus significatifs à utiliser dans une requête. Les mots avec une importante valeur TF-IDF supposent une relation forte avec le document où ils apparaissent. Ainsi, si un mot d'un document ayant une importante valeur TF-IDF fait partie de la requête, alors l'utilisateur pourrait être intéressé par ce document. L'étude a été effectuée sur 1400 documents et a permis de conclure que TF-IDF est un heuristique simple et efficace pour trouver des documents qui sont significatifs pour une requête. Le pouvoir discriminant de cet algorithme facilite la tâche d'un moteur de recherche par exemple, en trouvant rapidement des documents significatifs qui très probablement satisferont l'usager.

Les méthodes qui utilisent l'approche statistique sont basées en général sur le calcul de la fréquence d'apparition des termes de la phrase et utilisent conjointement des heuristiques pour « filtrer » les phrases les plus importantes. On peut dire aussi que ces phrases, issues d'un document ou d'un ensemble de documents, contiennent davantage d'information et seront postérieurement choisies pour faire partie du résumé. Bien que ces méthodes soient la base de beaucoup de logiciels commerciaux et non commerciaux actuels, et présentent

une performance relativement bonne dans quelques cas, on précise à plusieurs occasions que l'intervention humaine est nécessaire pour « adoucir » le langage de façon significative pour ne pas inclure des éléments de connaissance du domaine (McCargar, 2004) et des procédures pour essayer d'identifier la sémantique du texte. Plusieurs travaux de recherche ont tenté d'enrichir cette approche, c'est d'ailleurs ce que nous allons découvrir dans la prochaine section.

2.2 L'approche statistique enrichie

Nous présentons huit méthodes, qui sont aussi basées sur l'obtention de certaines mesures statistiques des termes de la phrase, dont leur signification a été renforcée en identifiant des termes ou des phrases en rapport avec eux sémantiquement. Cette relation sémantique est faite au moyen de thésaurus, de dictionnaires, de bases de données spécialisées et plus généralement d'ontologies.

Les ontologies sont connues comme un outil de modélisation de concepts pour décrire un système d'information par rapport à sa sémantique et ses connaissances (Neches *et al.*, 1991). Après que les ontologies furent introduites dans le domaine de l'intelligence artificielle (Gruber, 1995), elles ont été combinées avec le traitement automatique du langage naturel et appliquées dans beaucoup de domaines comme par exemple, l'ingénierie de la connaissance, la recherche d'information et le Web sémantique (W3C, 2001). Pour une proposition pour classifier les ontologies, le lecteur pourra se référer au document de Motta (2006) et à l'article de Ruder *et al.* (1997). Plus précisément, l'utilisation d'ontologies avec des méthodes statistiques a permis d'améliorer la précision de la classification de textes (Yu *et al.*, 2006). Dans ce cas, on utilise la connaissance ontologique linguistique.

Les huit modèles ou méthodes sont maintenant décrits en suivant à travers les travaux qui nous ont semblé les plus pertinents pour comprendre l'approche statistique enrichie.

2.2.1 Modèle statistique du langage

L'objectif de ce modèle est d'estimer la probabilité de chaînes de mots, de phrases et de documents (Gao, 2004). L'information statistique comme la fréquence des mots et des documents est un composant du modèle qui est appliqué à des domaines comme la

reconnaissance de la voix, la traduction automatique et la récupération d'information. Les travaux de Yu et al. (2006) et Gao (2004) essaient de déterminer une structure en combinant des ontologies avec des méthodes statistiques. Cette structure est construite à partir d'un cadre de description de l'ontologie et d'une ontologie de connaissance linguistique. Ensuite, différents types de connaissance ontologique linguistique sont appris d'un corpus d'entraînement. Dans le traitement d'un document non vu (donc à classifier), la valeur de son évaluation sémantique est obtenue à partir des différents types de connaissances linguistiques des ontologies.

Les résultats obtenus en appliquant la méthodologie proposée sont comparés avec ceux des classeurs de Bayes, de la machine à vecteur de support et des plus proches voisins. Cette comparaison a permis de mettre en évidence une augmentation relative dans leur précision (Yu *et al.*, 2006).

2.2.2 Mise en correspondance de phrases

L'essence de ce méthode est la mise en correspondance de phrases d'un document avec les nœuds d'une ontologie hiérarchique pour transformer la représentation sémantique du contenu d'information d'une phrase (Hennig, 2008).

Le modèle est induit au moyen d'un classeur de machines à vecteur de support qui utilise les phrases d'un résumé obtenu par un moteur de recherche. Le classeur fait une mise en correspondance d'une phrase avec la taxonomie en choisissant le sous-arbre qui représente le mieux la phrase. La mesure de distance est calculée avec le cosinus entre le vecteur d'attributs et la classe.

Le processus d'extraction de phrases du document à résumer est fait en créant une collection non ordonnée de mots, en écartant les mots d'arrêt (par exemple : et, ou, le, etc.), les mots mal écrits ou ayant la même racine. Les scores de chaque phrase peuvent être calculés en faisant une moyenne des mesures TF et TF-IDF de chacune d'elles (Mani et Bloedorn, 1998). En outre, la similitude peut être calculée par le cosinus de la phrase au document. Le modèle construit correspond donc à une collection non ordonnée de mots avec une mesure TF-IDF associée.

Suite aux résultats expérimentaux du modèle, l'auteur Hennig (2008) conclut qu'un classeur, formé avec des attributs basés sur une ontologie, offre plus de potentiel que les classeurs formés seulement avec les corpus habituels utilisés pour la recherche dans des résumés. Il est important de remarquer ici que les auteurs utilisent une taxonomie pour effectuer une recherche de documents dans le réseau Internet afin de renforcer la sémantique des termes des documents trouvés et que les phrases seront ensuite utilisées par le classeur.

2.2.3 Utilisation de WordNet ou UMLS (Verma *et al.*, 2007)

Dans ces travaux, les sources de connaissance ontologique comme WordNet (Fellbaum, 1985) et UMLS (NIH, 2013) sont utilisées pour déterminer et enrichir le sens des mots contenus dans les documents à résumer.

WordNet est une base de données lexicale, lisible par des machines, pour la langue anglaise, largement utilisée en linguistique et développée par Princeton University. Cette base de données est composée de mots reliés entre eux, qui sont des substantifs, des verbes, des adjectifs et des adverbes. Les mots sont organisés dans des ensembles de synonymes, appelés synsets, et reliées par trois relations sémantiques : hyperonyme, méronyme et pertainyme. Le système UMLS (Unified Medical Language System) de la NLM (National Library of Medicine) est conçu pour aider les systèmes d'information médicale à comprendre le sens des termes et concepts en biomédecine et d'autres domaines de la santé (NIH, 2013).

Le système proposé par Verma *et al.* (2007) effectue d'abord une révision de la requête de l'utilisateur en employant WordNet et UMLS. Dans ce processus, on ajoute des mots porteurs de sens et on élimine ce qui est redondant, puis on les retourne à l'utilisateur pour sa révision. Ensuite, on calcule la distance de chaque phrase dans le document par rapport à la requête révisée par l'utilisateur. La fonction de distance (dans le sens d'Euclide) est une mesure de dissimilitude, qui a les propriétés suivantes :

- positivité ($d(x, y) \geq 0$ pour toute x, y ; $d(x, y) = 0$, si x=y);
- symétrie ($d(x, y) = d(y, x)$);
- inégalité triangulaire ($d(x, z) \leq d(x, y) + d$ (et, z) pour tous les points x, et, z).

Ayant calculé cette distance, on vérifie si elle est plus petite qu'un certain seuil établi. Si elle est bien plus petite, la phrase devient candidate pour être incluse dans le résumé. On calcule à nouveau les distances entre des paires de phrases candidates pour obtenir des groupes de phrases. On choisit finalement les phrases des groupes ayant les valeurs maximales.

En comparant ce système avec d'autres, on observe une certaine performance importante, mais à son tour on souligne quelques limites. En effet, il y a une réduction insuffisante de la redondance et, pour ne pas avoir à réaliser une analyse syntaxique du document original, on baisse la qualité du résumé. De toutes manières, il est important de mettre l'accent sur le fait que l'utilisation de connaissance ontologique est une manière effective d'aller au-delà de la simple méthode qui permet de récupérer une information à partir d'une requête.

2.2.4 Système de Bellare (Bellare *et al.*, 2004)

Ces travaux utilisent aussi WordNet, mais pour extraire des sous-chaines de mots. Dans les travaux de Bellare et al. (2004), on présente un autre système basé sur WordNet dans lequel est créée une matrice qui consiste en des phrases du document à résumer et de sous-graphes porteurs de sens extraits de ce système de référence. Un algorithme principalement essaye de capturer l'information sémantique globale du texte à partir de synsets extraits de WordNet. Cet algorithme commence avec un prétraitement du texte : il effectue une segmentation des phrases afin de choisir la signification correcte à partir des synsets trouvés de WordNet. L'identification de *collocations* ou *cooccurrences* de mots fait partie de ce même processus; elles aident à mieux capturer la sémantique du texte, avant de considérer les mots individuellement. De même, les mots d'arrêt sont écartés (des mots qui sont répétées comme le, il, à, et, ou, etc.).

Ensuite, un sous-graphe est construit à partir de WordNet, pour trouver la portion du graphe de WordNet qui est significative pour le texte. Pour effectuer ceci, les mots présents dans le texte à résumer sont marqués dans le graphe de WordNet. Le graphe est ensuite parcouru afin d'associer ces mots avec les *synsets* de WordNet. Le parcours s'effectue par une recherche en largeur d'abord jusqu'à une profondeur fixe, là où les *synsets* deviennent trop

généraux pour être considérés. Finalement, le graphe G est construit : les mots et les synsets marqués forment les nœuds et les généralisations constituent les arcs.

Le prochain pas consiste à assigner la valeur aux *synsets* en fonction de son importance dans le texte. L'idée générale est que si plusieurs mots dans le texte correspondent au même *synset*, ce dernier sera considéré comme plus significatif et obtiendra une plus grande valeur (Ramakrishnan et Bhattacharya, 2003). Une matrice M, m*n, est ensuite construite, dans laquelle les m sont les phrases et les n les nœuds dans le graphe G. Une analyse de composants principaux sur cette matrice M permet d'obtenir les vecteurs propres correspondants. Les projections, ou les images de chaque vecteur propre sur toutes les phrases, sont calculées puis classées de la plus grande à la plus petite afin de pouvoir choisir les phrases ayant les plus hautes projections. Les phrases ainsi obtenues, sont les phrases d'une plus grande importance dans le texte.

Ce système a été comparé avec le logiciel commercial pour résumer appelé *Copernic* (Bouchard et Bouchard, 1996) sur un corpus de DUC 2002 (Document Understanding Conference) (NIST, 2002). Malgré l'utilisation de WordNet pour construire des chaînes porteuses de sens par rapport au texte et la capture de l'information sémantique au moyen des vecteurs singuliers, les résultats obtenus sont inférieurs à ceux du logiciel *Copernic*.

2.2.5 Travaux de Gong et Liu (Gong et Liu, 2001)

Les auteurs Gong et Liu (2001) proposent des méthodes pour produire des résumés génériques par sélection de phrases en utilisant les principes de l'analyse de la sémantique latente (Benzécri, 1973) .

Ces deux méthodes sont basées sur la mesure de la pertinence et sur une analyse de la sémantique latente (Deerwester *et al.*, 1990). Les deux méthodes décomposent le document en phrases individuelles pour créer ensuite des vecteurs de fréquence des termes.

Par exemple, $T_i = [t_{1i}\ t_{2i}. . .\ t_{ni}]^T$ est le vecteur de fréquence de termes du segment *i*, où t_{ij} dénote la fréquence dans laquelle le terme *j* se produit dans l'unité lexicale *i*. Une unité lexicale peut être une proposition, une phrase, un paragraphe ou le document complet. Ensuite, on calcule le vecteur $A_i = [a_{1i}\ a_{2i}. . .\ a_{ni}]^T$ qui correspond à l'unité lexicale *i*, étant

donné $a_{ji} = L(t_{ji}) * G(t_{ji})$, où $L(t_{ji})$ est le poids local pour le terme *j* dans l'unité lexicale *i* et $G(t_{ji})$ est le poids global du terme *j* dans le document.

Pour obtenir un résumé avec la mesure de la pertinence, on calcule la valeur de la pertinence de chaque phrase dans le document. On choisit ensuite la phrase la plus importante du document, elle s'ajoute au résumé et elle est retirée de l'ensemble de phrases candidates. Ainsi on élimine les termes contenus dans la phrase qui apparaissent dans le document original à résumer. La procédure est répétée avec les phrases restantes, jusqu'à l'obtention du nombre de phrases souhaitées pour le résumé.

Afin d'obtenir un résumé au moyen d'une analyse de sémantique latente, les auteurs s'inspirent de la méthode LSI (Latent Semantic Indexing), qui pour sa part applique une décomposition en valeurs singulières à la matrice d'occurrences de termes (Riley *et al.*, 2006; Steinbach *et al.*, 2006). Le processus commence par la création d'une matrice de termes par des phrases, $A = [A_1\ A_2. . .\ A_n]$, où chaque colonne du vecteur A_i représente le vecteur de fréquence de termes de la phrase *i* dans le document en considération. S'il y a un total de *n* termes et de *m* phrases dans le document, nous aurons alors une matrice A, *m* * *n* du document. À partir de cette matrice, les vecteurs singuliers de la matrice A sont calculés. Sans perte de généralité soit $m \geq n$, la matrice A, exprimée dans les termes de ses valeurs singulières, est définie par :

$$A = U \square V^T \tag{2.1}$$

où $U = [u_{ij}]$ est une matrice orthonormée dont les colonnes sont appelées vecteurs singuliers gauches, $\sum = \text{diag}\ [\sigma_1\ \sigma_2. . .\ \sigma_n]$ dont les éléments sont les valeurs singulières non négatives ordonnées de manière descendante et $V = [v_{ij}]$ est une matrice orthonormée *n* * *n* dont les colonnes sont appelées vecteurs singuliers droits. En outre si le rang de A est rang (A) = r, donc $\sum$ satisfait $\sigma_1 \geq \sigma_2 \geq . . . \geq \sigma_n \geq \sigma_{n+1}$. D'un point de vue sémantique, la décomposition en valeurs singulières dérive la structure sémantique du document représenté par la matrice A (Deerwester *et al.*, 1990). Le processus de sélection de phrases pour construire le résumé est effectué en choisissant le vecteur singulier droit k-nième de la matrice V^T et en

choisissant la valeur v_{ir} correspondant du vecteur ψ_i de la matrice V^T, jusqu'à atteindre une valeur prédéfinie de *k*.

Les évaluations des deux méthodes ont été faites sur un corpus de 549 textes de nouvelles du canal de télévision CNN avec des longueurs de 3 à 105 phrases. Trois évaluateurs humains ont également effectué des résumés sur 243 documents contenus dans le corpus mentionné. Les résultats ont montré que même si les approches sont très différentes, elles ont produit des performances semblables. Mais on observe aussi qu'il y a une importante disparité avec les résumés produits manuellement, ce qui aurait mérité une étude plus approfondie du traitement de l'espace d'attributs ainsi que la description des patrons sous-jacents.

2.2.6 Modèle d'espace vectoriel sémantique

Dans les deux sections suivantes, nous présentons des travaux de recherche basés sur un schéma de modèle vectoriel sémantique, qui consiste à représenter les phrases des documents comme des vecteurs, où chaque dimension correspond en général à un terme du document.

2.2.6.1 Décomposition en valeurs singulières (Berry *et al.*, 1995)

Cette technique a été utilisée dans d'autres domaines, notamment dans les domaines de l'extraction de d'information et de la recherche d'information. Mais nous la considérons comme très importante car nous pensons qu'elle est une option efficace pour l'optimisation d'espaces de recherche, c'est pourquoi nous la présentons en premier.

Ce sont les travaux de Berry et al. (1995) qui nous intéressent plus particulièrement. L'utilisation de la décomposition en valeurs singulières est présente ici pour pouvoir profiter de la structure implicite de haut niveau dans l'association de termes d'un document que celle-ci produit. Les termes et les documents, représentés pour les plus grandes valeurs singulières d'une matrice de termes par document, sont associés avec les requêtes d'un utilisateur. Cette méthode est appelée l'analyse sémantique latente (Deerwester *et al.*, 1990).

Ce travail montre comment, en utilisant des indices conceptuels dérivés statistiquement, on peut gérer le problème de la correspondance lexicale dans les systèmes de récupération de textes, en considérant que l'information est récupérée, typiquement, par correspondance littérale des termes dans les documents avec les termes de l'utilisateur, dans la *requête*. Ceci entraîne des problèmes éventuels qui sont reliés à la *synonymie* (différents mots avec la même signification) et la *polysémie* (différentes significations pour le même mot) des termes. Les auteurs effectuent une analyse sémantique latente à la matrice initiale, pour ensuite utiliser les valeurs et les vecteurs singuliers en vue d'effectuer la récupération d'information correspondante, et ainsi ajouter des termes nouveaux ou des documents à une base de données existante (O'Brien, 1994).

La base mathématique présentée dans ces travaux a ses origines dans les travaux d'Eugene Beltrami (1835-1899) (Beltrami, 1868) et Camille Jordan (1831-1921) (Jordan, 1870)et est utilisée pour la décomposition matricielle et la projection de nouvelles valeurs à partir d'une base vectorielle obtenue avec cette transformation. On suppose qu'une des matrices de la transformation contient les concepts les plus importants d'un ensemble de textes par rapport à une requête d'un utilisateur. La matrice est décomposée en facteurs. Si A est la *matrice de termes par document*, alors $A = U\sum V^T$, où U est appelée la matrice de vecteurs singuliers gauches, $\sum$ est une matrice diagonale de valeurs singulières et V^T est la matrice de vecteurs singuliers droits. Cette équation, qui peut aussi être exprimée comme $V^TAU = diag\ (s1, s2,..., sm)$, connue comme le théorème de décomposition en valeurs singulières.

Les applications de l'analyse sémantique latente sont nombreuses. Soulignerons l'extraction d'information, la réduction de bruit (fonction du nombre de facteurs de la transformation), le filtrage d'information (pour comparer l'importance d'un document donné), la recherche d'information dans de grandes bases de données et la modélisation de la mémoire humaine. Par exemple, pour récupérer une information dans de grandes bases de données comme celle de TREC (Text Retrieval Conference) (NIST, 1993), à partir d'un échantillon qui consiste de 70000 documents et 90000 termes différents, l'analyse sémantique latente a requis 18 heures de traitement dans un système Sun Sparc. C'est un temps relativement faible en tenant compte de l'époque et de la taille des données de travail qui est de l'ordre du milliard.

De l'utilisation de l'analyse sémantique latente - on suppose qu'il y a une structure latente dans l'utilisation des mots - pour analyser un texte, nous concluons que cette méthode est supérieure à la simple correspondance entre des mots. Cette amélioration est obtenue parce qu'on remplace les mots au cas par cas par des concepts sémantiquement robustes et statistiquement dérivés.

2.2.6.2 Travaux de Vikas (Vikas, 2008)

Dans ses travaux, Vikas (2008) utilise également la décomposition en valeurs singulières comme partie d'une analyse de composants principaux appliquée à un ensemble d'attributs, et ceci dans le but d'obtenir les valeurs singulières et les vecteurs propres à partir d'un modèle d'espace vectoriel. Ce modèle a été amélioré sémantiquement en modifiant les poids du vecteur de mots en fonction de la présence ou de l'absence de mots appelés mots d'action contextuelle, avec l'aide de WordNet. Un mot d'action contextuelle est un mot qui est introduit au début des phrases d'un résumé manuel pour augmenter son impact. Le modèle ainsi constitué est appelé modèle d'espace vectoriel sémantique.

Dans le travail mentionné, l'auteur propose d'améliorer l'espace d'attributs en utilisant un traitement sémantique, pour identifier les attributs (par rapport au thème) à travers ces mots-clés. Le schéma général que les auteurs proposent consiste à construire un modèle d'espace vectoriel à partir de l'ensemble de documents, puis de générer un contenu sémantique avec l'aide de WordNet en utilisant des mots d'action contextuelle. Ensuite, il y a une analyse de composants principaux de l'ensemble des mots afin d'obtenir les mots-clés les plus importants. Les phrases sont évaluées en fonction de différentes caractéristiques comme le poids du mot-clé, la longueur de la phrase ou encore les phrases en entrée. Le résumé est enfin généré par extraction des phrases ayant les scores les plus élevés.

En partant du modèle d'espace vectoriel, les processus insérés dans chacune des étapes mentionnées précédemment pourraient être résumés ainsi :

- création d'une matrice basée sur la fréquence du terme ;
- identification des mots d'action avec l'aide d'un ensemble de mot-clé. Si les mots action n'apparaissent pas dans cet ensemble, on fait appel à WordNet pour extraire

des synonymes.

- détermination de la liste d'objets associés (substantifs ou adjectifs pour l'action) avec chacun des mots action. Le poids de ces mots est ensuite modifié dans la matrice.

On effectue une analyse de composants principaux sur cette matrice, devenue une matrice sémantique. On obtient un vecteur *p*, qui est le document projeté au moyen des documents propres ou vecteurs propres. Ensuite, on divise la composante de *p* par sa valeur singulière correspondante afin de déterminer l'importance thème/document. Les mots ayant la plus grande valeur dans les vecteurs correspondants aux documents propres choisis correspondent donc aux mots clés par thème. Ces mots-clés sont ajoutés à l'ensemble des caractéristiques mentionnées (longueur de la phrase, position, etc.) à prendre en compte au moment de choisir les phrases pour l'obtention du résumé. Les auteurs comparent leur approche avec d'autres approches semblables. Ses résultats montrent une meilleure performance.

Dans ce travail, on observe une fois de plus que si on ajoute des composants sémantiques à un espace d'attributs, en partant de concepts prédéfinis ou des synonymes de ces attributs, on peut mieux déterminer l'information essentielle. Mais une fois de plus, les résultats ne sont pas comparables à ceux obtenus avec un logiciel populaire commercial dédié à l'extraction de résumés.

2.2.6.3 Travaux de Barsilay et Elhadad (Barzilay et Elhadad, 1997)

Dans l'article de Barsilay et Elhadad (1997), les auteurs proposent d'identifier des phrases porteuses de sens pour former le résumé, à partir de l'identification des chaînes lexicales "plus fortes", qui sont obtenues en mélangeant plusieurs sources de connaissance : WordNet, l'étiquetage de catégories grammaticales, l'identification de groupes nominaux et un algorithme de segmentation. Les résultats empiriques indiquent que la qualité est améliorée. La méthode présentée appartient à la famille de techniques qui utilisent la distribution de la fréquence entre les mots et les liaisons lexicales, pour approcher son contenu d'une manière plus robuste.

Certains concepts sont utilisés fréquemment pour la construction d'entités cohésives, notamment la notion de cohésion (Halliday et Hasan, 1976). La cohésion peut être atteinte avec l'utilisation de termes rapportés sémantiquement, la coréférence, les conjonctions, la cohésion lexicale (Hoey, 1991), classée dans les catégories de réitération (répétition et synonymes) et la collocation (mots qui co-apparaissent dans le même contexte lexical, par exemple, professeur-école). La cohésion lexicale se produit non seulement entre des mots, mais aussi entre des séquences de mots reliés appelées chaînes lexicales (Morris et Hirst, 1991). Ces chaînes lexicales fournissent une représentation de la structure cohésive lexicale du texte et sont aussi utilisées pour l'extraction d'information (Stairmand, 1996).

Barsilay et Elhadad (1997) construisent les chaînes lexicales au moyen d'un algorithme basé sur les principes généraux de la construction de chaînes lexicales et sur l'algorithme défini par Hirst et St-Onge (1998). Les mots candidats sont d'abord sélectionnés. Puis, pour chaque mot candidat, on cherche une chaîne appropriée selon un critère de relation entre des membres des chaînes. Si cette chaîne existe, on insère le mot dans la chaîne. Ensuite, on assigne des valeurs aux chaînes et on identifie les plus fortes. Les valeurs sont assignées en tenant compte de sa longueur, sa distribution dans le texte, sa densité, la topologie du graphe (diamètre du graphe des mots) et le nombre de répétitions. Seuls les paramètres, obtenus selon leur longueur et un indice d'homogénéité basé sur leur présence, sont de bons prédicteurs de la force de la chaîne. Une fois les chaînes choisies, les auteurs proposent d'effectuer la sélection des phrases, à inclure dans le résumé à partir des documents originaux, au moyen de trois heuristiques. La première consiste à choisir la phrase qui contient la première apparition d'une chaîne membre et la deuxième à choisir une chaîne représentante. Dans la troisième heuristique, on part du fait qu'un même thème est examiné dans plusieurs parties du texte, donc sa chaîne est distribuée dans tout le texte. Mais à son tour, ce thème est la partie centrale d'une unité textuelle (segments). On essaie d'identifier des unités de texte où ce thème se concentre. Cette concentration est calculée comme le nombre de présences de chaînes membres dans un segment divisé par le nombre de substantifs dans le segment. Pour chaque chaîne, on trouve les unités textuelles où la chaîne est hautement concentrée. On extrait la phrase qui contient la première chaîne dans cette unité. Cette heuristique donne le résultat le plus pauvre, contrairement à ce que nous indiquerait l'intuition.

Les résultats de cette approche, obtenus pour des résumés de longueurs de 10 % et 20 %, ont été comparés avec quelques produits commerciaux. On observe que ces résultats sont supérieurs tant en précision qu'avec le rappel. Ces résultats indiquent le fort potentiel des chaînes lexicales comme source de connaissance pour l'extraction de phrases, mais présentent des problèmes par rapport à la granularité des phrases (l'unité d'extraction est la phrase). En effet, les phrases longues ont plus de probabilité d'être choisies et, de ce fait, peuvent inclure des composants qui n'ont pas de mérite suffisant ou n'apportent pas d'information additionnelle. De plus, il n'est pas possible de contrôler la longueur et le niveau de détail du résumé car le nombre de chaînes fortes est très petit (cinq ou six), indépendamment de la taille du document à résumer.

En conclusion, les méthodes utilisant une approche statistique enrichie, que nous avons étudiées dans cette section, présentent en général une performance globale supérieure par rapport aux méthodes purement statistiques. Toutefois, certaines d'entre elles présentent d'importantes disparités par rapport aux résumés obtenus manuellement, ce qui nous indique qu'il est nécessaire d'étudier plus en profondeur les procédures mises en œuvre pour enrichir l'approche statistique.

2.3 Apprentissage automatique appliqué aux méthodes de résumé

Dans cette section, nous présentons une série de modèles, de développements et d'approches que nous considérons comme les plus importants depuis le début des années 90 jusqu'à nos jours, pour l'obtention de résumés extractifs en tenant compte des performances montrées. L'apprentissage automatique est sous-jacent à la question de comment concevoir des algorithmes et créer des programmes informatiques qui améliorent automatiquement leur performance avec l'expérience (Bishop, 2007). Herbert Simon définit l'apprentissage automatique comme « un certain changement dans un système qui lui permet de mieux effectuer une seconde fois ou de manière répétée la même tâche ou une autre sur la même population » (Hirst et St-Onge, 1998). L'apprentissage automatique a été formellement défini ainsi : on dit qu'un programme informatique apprend de l'expérience E pour une certaine classe de tâches T et une mesure de performance P, si sa performance dans la tâche T, mesurée comme P, s'améliore avec l'expérience E (Mitchell, 1997).

En général, l'apprentissage automatique peut être divisé en trois approches : apprentissage supervisé, apprentissage non-supervisé et apprentissage semi-supervisé. Dans la première, on suppose l'existence d'un enseignement d'une certaine mesure de l'efficacité et d'une certaine méthode ou algorithme pour classer des instances d'entraînement. Dans la deuxième, on suppose que ce qui apprend forme et évalue ses propres concepts. Il n'y a pas d'instances d'entraînement. Dans l'apprentissage semi-supervisé, on construit des modèles qui comportent des éléments des deux premières approches.

Il existe différents algorithmes d'apprentissage supervisé. Citons, par exemple, les algorithmes ID3 (Quinlan, 1986), ID4.5 (Quinlan, 1992), C5.0 (Kuhn et Johnson, 2013), la machine à vecteur de support, le classeur bayésien naïf, les réseaux de Bayes (Neapolitan, 2003), le perceptron (Rosenblatt, 1957), l'algorithme des plus proches voisins, la régression logistique (Agresti, 2007), l'analyse discriminante linéaire de Fisher (McLachlan, 2004). Comme exemple d'apprentissage non supervisé, nous citons le partitionnement de données (MacQueen, 1967), les cartes auto adaptives (Kohonen, 1990) et les réseaux neuronaux (McCulloch et Pitts, 1943).

Nous présentons maintenant une série de modèles, de développements et d'approches que nous considérons comme les plus importants depuis le début des années 90 jusqu'à nos jours, pour l'obtention de résumés extractifs par apprentissage automatique.

2.3.1 Travaux de Mani et Bloedorn

Un autre travail important à prendre en compte est présenté par Mani et Bloedorn (Mani et Bloedorn, 1998). Les auteurs décrivent l'utilisation de l'apprentissage automatique pour entraîner un *corpus* de documents et les résumés correspondants afin de découvrir des fonctions saillantes, qui permettent de déterminer la combinaison d'attributs optimale pour une certaine tâche de résumé. Les fonctions saillantes sont généralement obtenues à partir de l'évaluation des phrases du texte source en tenant compte de différentes caractéristiques comme par exemple la *localisation* (Edmunson, 1969), les *mesures statistiques de la proéminence du terme* (Luhn, 1958), la *structure rhétorique* (Miike *et al.*, 1994), la *similarité entre phrases* (Pollock et Zamora, 1975), la *présence ou l'absence de certaines caractéristiques syntaxiques* (Pollock et Zamora, 1975), la *présence de noms propres*

(Kupiec et Chen, 1995) et les mesures de *proéminence de certains concepts sémantiques* (Paice et Jones, 1993). En général, on suppose qu'un nombre de caractéristiques extraites de différents niveaux d'analyse peuvent être combinées pour contribuer à l'importance des termes ou des phrases.

Dans ce travail, les auteurs décrivent une approche d'apprentissage automatique qui apprend des résumés génériques en fonction d'une requête d'un utilisateur. L'objectif principal du travail est la comparaison du niveau de performance entre différentes méthodes d'apprentissage, la comparaison de la stabilité de l'apprentissage sous différents taux de compression (taille du résumé par rapport au document source) et la relation entre les règles apprises du résumé générique et celles apprises dans le cas d'une requête. Le résumé est traité comme la nécessité d'information de l'utilisateur. La méthode part d'une collection de textes avec ses résumés manuels. Chacune des phrases des documents sources est évaluée en fonction de certaines caractéristiques groupées selon trois classes :

a) *La localisation* : Cette classe utilise la structure du texte à différents niveaux d'analyse.
b) *Le sujet* : Cette classe est obtenue en calculant les fréquences inverses des termes du texte, ainsi qu'en calculant la mesure statistique G^2 (Cohen, 1995), qui indique la probabilité que la fréquence d'un terme dans un document soit plus grande que ce qui pourrait être attendu dans un *corpus*, vu la taille relative du document dans le *corpus*. Les auteurs utilisent la fréquence du terme dans le document, sa fréquence dans le corpus, le nombre de termes dans le document et la somme de tous les termes dans le corpus.
c) *La cohésion* (Riley *et al.*, 2006) : Cette classe se réfère aux relations entre les mots et entre les expressions pour déterminer la force de connexion avec le texte.

Ensuite, le processus d'entraînement consiste à étiqueter les phrases avec une valeur booléenne qui dépend de l'importance de la phrase par rapport au résumé. En d'autres termes, cette valeur montre si la phrase appartient ou non au résumé (exemples positifs ou négatifs). L'ensemble de phrases ainsi étiquetées est fourni à trois algorithmes d'apprentissage : l'analyse discriminante canonique (SCDF) (qui appartient aux techniques de l'analyse factorielle), C4.5 (Quinlan, 1992) et AQ15C (Wnek *et al.*, 1995). Il est

important de noter que l'analyse entre le résumé et le texte source, pour effectuer le processus d'étiquetage, est fait en comparant chacune des phrases du texte avec le résumé complet et non avec chacune de ses phrases. Cette procédure résout le problème de la redondance dans les phrases du résumé qu'on souhaite obtenir. Pour effectuer cette comparaison, on utilise une fonction d'étiquetage dont les paramètres sont le poids des mots des phrases du document source et des mots du résumé, le nombre de mots communs et le nombre total de mots de la phrase et du résumé.

Ce travail présente aussi une manière d'obtenir les résumés pour l'entraînement, à partir d'une requête d'utilisateur. À partir d'un ensemble de documents choisis par l'utilisateur, on calcule un vecteur centroïde en classant les mots des documents au moyen de la mesure statistique G^2 et en choisissant les mots avec une déviation typique plus grande ou égale à 2.5, comme les représentants du thème. On applique ensuite un autre algorithme pour découvrir dans chaque document les mots en rapport avec le thème pour constituer le résumé correspondant.

Enfin, on applique les trois algorithmes d'apprentissage, SCDF, C4.5 et AQ15C, à ces ensembles d'entraînements, en obtenant des résultats de précision, rappel et F_measure qui varient entre 56 % et 89 %. Aucune comparaison n'est présentée avec un quelconque logiciel spécialisé. Bien que ces résultats soient importants, sûrement par l'introduction d'algorithmes d'apprentissage, ils montrent la nécessité d'approfondir encore plus l'étude de méthodes et de techniques pour choisir des ensembles d'entraînement plus performants.

2.3.2 Travaux de Sharan et Imran (Sharan et Imran, 2009)

Les auteurs proposent une approche pour le résumé automatique en utilisant les algorithmes C4.5 et le classeur bayésien naïf. La sélection des attributs pour la création de l'ensemble d'entraînement est réalisée au moyen des huit métriques suivantes :

- Edmunson : Cette mesure assigne un score à chacune des phrases en se basant sur les fréquences des mots significatifs.
- Luhn : Cette mesure ne prend pas en compte la fréquence des mots significatifs. Par contre, elle distingue les mots « significatifs » de ceux « non significatifs ». Pour bien réaliser cela, on génère une liste de termes candidats du corps de documents en

ordre descendant selon sa fréquence. Les termes avec la fréquence la plus haute et la plus basse fréquence sont pris comme « non-significatifs ».

- Localisation : On prend en compte la position de la phrase dans le document pour déterminer son importance. Par exemple, si la phrase est au début ou à la fin d'un paragraphe ou du document.
- Phrases CUE : Cette métrique est basée sur l'hypothèse que l'importance de la phrase repose sur la présence de certaines locutions « pragmatiques » telles que par exemple « Dans cet article », « On peut conclure », etc.
- Titre : Les phrases qui contiennent le titre sont prises comme étant significatives.
- Première phrase : On prend toujours la première phrase du document comme étant significative.
- Longueur de la phrase : En tenant compte que les phrases courtes ne seront pas incluses dans le résumé, on prend un nombre minimal de mots que doit contenir chacune des phrases (par exemple cinq mots).
- Occurrence de noms : On définit cette mesure basée sur l'idée que les noms sont porteurs de sens.

L'étiquetage des phrases de l'ensemble d'entraînement est fait en comparant chaque phrase du corpus de documents d'entraînement avec son résumé et en vérifiant si elle appartient ou pas à celui-ci, afin de l'affecter avec l'étiquette « importante» ou « non importante ». La performance de la fonction d'induction produite, est évaluée en calculant sa précision, son rappel et son F_measure. Ces valeurs sont comparées aux logiciels pris comme référence « First Sentence » et « Word Summarizer ».

Si la méthode montre une bonne performance par rapport aux logiciels de référence (qui ne sont pas basés sur l'apprentissage automatique), surtout pour l'algorithme C4.5, on ne fait pas la comparaison avec des résumés produits par des humains. De plus, les courbes ROC obtenus ne présentent pas de bons résultats.

2.3.3 Travaux de Larocca et al. (Larocca et al., 2002)

Dans ce travail, on commence par mettre en évidence le peu de recherches qui sont faites au sujet des méthodes pour le choix des attributs importants, à partir de corpus d'entraînement.

Les auteurs emploient également les algorithmes C4.5 et le classeur bayésien naïf, mais en utilisant un ensemble d'attributs différents pour la caractérisation de l'ensemble de documents et la définition de l'ensemble d'entraînement. Ces attributs sont de deux types : statistiques et linguistiques. Ils font appel aux méthodes heuristiques suivantes pour l'extraction des attributs :

- moyenne TF-ISF : en multipliant la fréquence des termes de la phrase par sa fréquence inverse;
- longueur de la phrase : en utilisant la longueur normalisée qui est obtenue comme la raison du nombre de mots de la phrase sur le nombre de mots de la phrase la plus longue;
- position de la phrase : en donnant une valeur à la phrase selon sa position dans une section, un paragraphe, etc.;
- similarité au titre : en attribuant à chaque phrase sa similarité avec le titre au moyen du cosinus;
- similarité aux mots-clés : en considérant la similarité entre l'ensemble de mots-clés et chaque phrase du document, au moyen du cosinus;
- cohésion entre phrases : en calculant la similarité de chaque mot du document avec les autres et en l'ajoutant afin d'obtenir une valeur pour tous les mots. Après on normalise en divisant la valeur obtenue pour chacune des phrases et la valeur plus grande obtenue. Les valeurs proches de 1 signalent une grande cohésion.
- cohésion de la phrase au centroïde : en calculant le centroïde du document, qui est la moyenne arithmétique des valeurs des coordonnées des phrases du document. Après on calcule la similarité entre le centroïde et chacune des phrases. Et ainsi de suite, on normalise chaque phrase en divisant sa valeur obtenue par la plus grande valeur trouvée. Les phrases avec des valeurs proches de 1 ont une grande cohésion par

rapport au centroïde du document, qui est supposé représenter les idées de base du document.

L'article présente aussi un ensemble d'heuristiques en essayant d'identifier la structure argumentative du texte. La première étape que les auteurs réalisent est un processus de partitionnement en deux groupes, qui s'applique successivement pour obtenir un arbre binaire basé sur sa similarité lexicale. Après, on extrait les valeurs pour cinq attributs :

- profondeur de l'arbre : la position à un niveau déterminé de l'arbre (positions 1, 2, 3, 4), utilisé pour identifier les phrases avec une profondeur inférieure à 4;
- indicateur de concepts principaux : indicateur binaire qui signale quand une phrase capture les concepts principaux du document, en tenant compte que les concepts principaux sont des noms. Les 15 premiers noms avec la plus grande occurrence sont choisis. Finalement, pour chacune des phrases, la valeur de cet attribut est « vrai » si la phrase contient au moins un des noms et « faux » sinon;
- occurrence de noms propres : si une phrase contient un nom propre, elle est importante pour le résumé. Ainsi, on attribue une valeur « vrai » si la phrase contient au moins un des noms et « faux » sinon;
- occurrence d'anaphores : Si une phrase contient une anaphore, elle est non importante. Ainsi on attribue une valeur « vrai » si la phrase contient au moins une anaphore dans ses premiers six mots et « faux » sinon;
- occurrence d'information non essentielle : certains mots sont indicateurs d'information non essentielle par exemple « parce que », « en plus », « de même ». Ainsi si une phrase contient l'un de ces mots (appelés marqueurs de discours), elle sera étiquetée comme « vrai » et « faux » sinon.

Après avoir calculé l'ensemble des valeurs des attributs pour toutes les phrases, les auteurs réalisent la discrétisation correspondante. Ensuite, ils utilisent les algorithmes C4.5 et le classeur bayésien naïf pour l'induction des fonctions de classification à partir de la collection de documents TIPSTER (Harman, 1994). Pour le test correspondant, ils utilisent deux types de références : automatique et manuelle produite par un spécialiste.

L'article présente les résultats avec des taux de compression de 10 % et 20 % pour le résumé. Bien que les valeurs obtenues de performance pour la précision, le rappel et la F_measure soient relativement bonnes pour le classeur bayésien naïf (51 %), elles ne le sont pas pour l'algorithme C4.5 (35 %). On remarque ici que les résultats sont opposés à ceux du travail de la section précédente pour qui la meilleure performance était avec l'algorithme C4.5. On remarque aussi que l'article ne présente pas d'autres méthodes pour l'évaluation de la qualité des fonctions induites ni pour les résumés obtenus.

2.3.4 Travaux de García-Hernández et al.(García-Hernandez *et al.*, 2008)

Dans ce travail, les auteurs proposent une approche pour l'obtention du résumé automatique par extraction au moyen d'un algorithme d'apprentissage non supervisé qui utilise le partitionnement. Les auteurs partent de l'hypothèse que l'utilisation d'un algorithme d'apprentissage peut détecter des groupes de phrases similaires et ensuite sélectionner les phrases les plus représentatives.

Pour la sélection des termes de chacune des phrases, on utilise n-gramme, un modèle probabiliste qui permet de faire une prédiction du prochain élément à partir de certaines fréquences d'éléments survenues jusqu'au présent ou des éléments antérieurs.

Les valeurs $w_i\ (t_j)$ des attributs que les auteurs ont choisies pour la caractérisation des phrases sont :

- Boole : en attribuant la valeur 1 si le terme t_j apparaît dans le document i et 0 sinon;
- TF : en calculant la fréquence f_{ij} du terme i dans le document j;
- IDF : en appliquant la formule : $w_i\ (t_j) = \log\ (N/n_j)$, où N est le nombre de documents dans la collection et n_j est le nombre de documents où le terme *j* apparait ;
- TF-IDF : cette mesure donne plus de pertinence aux termes qui sont moins fréquents dans la collection, mais plus fréquents dans le document. On la calcule ainsi : $wi\ (t_j) = f_{ij} \times \log\ (N/n_j)$.

En ayant obtenu les valeurs pour chacune des phases du document (l'article présente la méthode pour un seul document), on applique l'algorithme des k-moyennes avec un nombre de groupes égal à cinq et une valeur initiale pour les groupes correspondante aux

valeurs initiales des premières phrases. On tient compte que ces premières phrases sont une référence initiale et pourtant de bonnes candidates pour faire partie du résumé.

Pour mesurer la similarité entre deux phrases, les auteurs utilisent la distance euclidienne définie comme étant: Distance(X, Y)= $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$, où X et Y sont des phrases représentées par des vecteurs avec n attributs.

Quand le processus de partitionnement est fini, on sélectionne les phrases les plus proches du centroïde comme les plus représentatives pour faire partie du résumé.

Les auteurs évaluent leurs résultats en utilisant le modèle statistique n-gramme du logiciel ROUGE (ROUGE-n) et en obtenant des valeurs à partir de 1-gramme et jusqu'au 11-gramme. Les valeurs montrées sont situées autour de 47 % pour la mesure 10-gramme.

Il est important de remarquer que cette recherche est un effort important pour la définition d'un modèle d'apprentissage non supervisé pour le résumé automatique et que sa performance est acceptable. Cependant, le travail mise sur l'obtention du résumé à partir d'un seul document. On ne fait pas l'évaluation du modèle proprement dit et les résultats n-gramme ne montrent pas une différence significative pour les différentes valeurs de n. Toutes les valeurs sont placées autour de 0.47, et montrent ainsi que le modèle ne trouve pas une différence remarquable entre les séquences de mots 1-gramme et 11-gramme (on présente une différence de millésimes).

Cette étude nous a permis de nous pencher plus particulièrement sur des questions de performance. Comment améliorer les moyens mis en œuvre pour extraire automatiquement des résumés ? Nous décrivons en détail cette problématique dans les pages suivantes.

2.4 Problématique

Nous venons de présenter les principaux avancements qui sont connus présentement dans le domaine du résumé automatique. Les approches utilisées sont basées sur des techniques statistiques simples ou enrichies par l'utilisation de dictionnaires, de thésaurus, d'ontologies et de bases de données ou bien font appel au domaine de l'apprentissage automatique. À partir de l'étude que nous avons effectuée, nous sommes en mesure de

présenter notre problème principal à résoudre, ainsi que les problèmes accessoires reliés à ce problème.

Bien que certaines des techniques utilisées aient des performances acceptables, surtout celles basées sur les approches statistiques enrichies et l'apprentissage automatique supervisé, nous pouvons remarquer que, de façon générale :

- On utilise surtout la fréquence des termes comme filtre pour identifier les phrases et les attributs les plus importants.
- Les bases de données, les dictionnaires, les thésaurus et les ontologies sont utilisés pour comparer les phrases ou certaines parties de la phrase qui feront partie du résumé, mais pas pour mesurer la quantité d'information apportée.
- Lorsque l'apprentissage automatique est utilisé, seule l'approche supervisée est mise en œuvre. L'ensemble d'entraînement est basé sur les résumés des documents utilisés pour la phase d'entraînement. À notre connaissance, il n'existe pas de travaux de recherche portant sur la mise en œuvre des deux autres approches d'apprentissage automatique, qui pourtant présentent certains avantages.
- En général, on observe une utilisation très limitée des techniques d'optimisation des espaces de recherche ainsi que des techniques d'induction des fonctions d'apprentissage pour obtenir les résumés extractifs.

Plus particulièrement et à partir de notre étude de l'approche basée sur des modèles appris, nous avons constaté qu'il existe beaucoup d'efforts dans le milieu scientifique pour concevoir des modèles et/ou méthodes qui permettent d'obtenir des résumés de qualité, c'est-à-dire semblables à ceux produits par des humains. Malgré une certaine qualité, beaucoup d'entre eux manquent encore d'exactitude et de précision. La majorité d'entre eux sont produits à partir d'un seul document, les problèmes étant augmentés avec plusieurs documents à cause de la redondance possible des phrases.

Dans la construction de ces modèles d'apprentissage automatique pour la production automatique de résumés, le problème qui apparaît comme majeur est le problème de l'identification des attributs les plus représentatifs. Cette identification se fait à partir des phrases des documents d'où on extrait les caractéristiques. Ensuite, il faut apprendre un

modèle qui sera postérieurement appliqué à de nouvelles phrases. Cet ensemble de phrases, choisies et porteuses des attributs les plus importants, constitue l'ensemble d'entraînement. Le choix de cet ensemble d'entraînement est un élément clé dans la détermination du modèle d'apprentissage et par conséquent de sa performance globale, dans la mesure où celui-ci pourrait réaliser une meilleure classification des phrases des nouveaux documents à résumer.

L'analyse des différentes méthodes utilisant l'approche par apprentissage automatique nous a permis de faire les constations suivantes :

- Les méthodes ne sont basées que sur l'approche d'apprentissage supervisé et quelques-uns sur l'apprentissage non supervisé, comme nous venons de le mentionner. Aucune méthode n'est basée sur l'approche semi-supervisée pourtant fort avantageuse.
- Le résumé du document est souvent utilisé pour comparer l'appartenance ou la non-appartenance des phrases du document à l'ensemble de phrases qui formera l'ensemble d'entraînement.
- Les méthodes les plus utilisées pour l'identification des attributs importants se basent sur l'obtention de certaines métriques ou mesures statistiques obtenues de chacun des attributs. Certaines mesures aussi pertinentes qui sont mises à profit actuellement comme : la fréquence des termes, le rapport χ^2, la mesure du gain d'information, le ratio OddsRatio, le rapport probabiliste logarithmique, le rapport de probabilité ou encore la séparation bi-normale, continuent à être utilisées sans changements importants. De même, on observe peu de recherches au sujet de méthodes performantes pour l'extraction d'attributs des espaces de recherche.
- Il existe d'autres méthodes basées sur certaines caractéristiques des phrases ou heuristiques telles que : les premiers N termes, les derniers N termes, le commencement de paragraphe, la fin de paragraphe, la ressemblance de titres, l'occurrence de noms, la position de l'attribut dans le document, les attributs de Luhn (1958), les attributs d'Edmunson (1969), etc. Comme dans la constatation précédente, on observe peu de recherches au sujet de l'obtention d'heuristiques performantes pour l'extraction d'attributs des espaces de recherche.

- Aucune méthode alternative n'est proposée pour améliorer le choix de l'ensemble d'entraînement.
- Aucune méthode n'est proposée pour améliorer l'efficacité de l'optimisation des espaces de recherche des attributs les plus performants.
- Les méthodes utilisées pour l'induction des fonctions d'apprentissage sont limitées à quelques-unes probabilistes et linéaires.
- Il reste encore des lacunes par rapport à deux problèmes classiques liés au résumé automatique : la redondance et le sur-ajustement (ou sur-apprentissage) en raison des méthodes peu performantes pour éliminer le bruit dans l'espace d'entraînement.
- Il n'existe pas de modèles généraux pour l'extraction de résumés.

Nous considérons les efforts pour utiliser certaines techniques d'apprentissage automatique pour obtenir des résumés extractifs comme intéressants et c'est un domaine en plein essor. En tenant compte des contraintes établies dans le théorème « pas de repas gratuit » (No free Lunch) (Wolpert et Macready, 1997), nous avons concentré nos efforts sur comment concevoir et tester des modèles pour extraire des résumés en nous basant sur des ensembles d'entraînement composés de phrases porteuses d'information et avec le minimum de bruit.

En ce qui concerne l'évaluation de la performance des méthodes d'apprentissage automatique, nous avons trouvé que, de façon générale, ce sont les métriques telles que la précision et le rappel qui sont utilisées. Parfois, ces deux métriques sont mises à profit pour calculer la F_measure. Seuls les travaux de Hanley et McNeil (1983) font appel aux courbes ROC pour cette évaluation de performance. Mais, aucun modèle ne mesure la performance de leurs méthodes en reliant les différentes métriques de précision, de rappel et de F_measure, avec les courbes ROC et l'outil d'évaluation ROUGE, élaboré par Lin et Hovy (2003) et habituellement utilisé pour évaluer la qualité de résumés produits automatiquement par rapport à ceux produits par des humains.

La problématique de notre recherche consistait donc à répondre à la question : quel modèle proposer pour améliorer le processus de résumé extractif basé sur l'approche par apprentissage automatique? Répondre à cette question suscite une série de problèmes fortement reliés qu'il est nécessaire de résoudre :

- Comment déterminer un ensemble d'entraînement à partir d'un document ou d'un ensemble de documents sans nécessairement faire appel aux résumés déjà produits ?
- Comment optimiser cet ensemble d'entraînement afin de réduire le bruit et identifier les phrases les plus porteuses d'information qui en font partie ?
- Comment déterminer que l'optimisation a atteint un niveau suffisant ?
- Quel(s) algorithme(s) d'apprentissage automatique utiliser pour abduire une fonction de classification ?
- Comment mesurer la performance du modèle d'extraction de résumés ainsi élaboré?
- Comment analyser les résultats obtenus par des mesures de performance ?
- Comment appliquer ce modèle sur de nouveaux documents à résumer ?
- Comment déterminer les paramètres les plus adéquats pour appliquer le logiciel ROUGE pour mesurer la qualité des résumés produits avec le modèle ?
- Comment rendre le modèle applicable à différents contextes et pour différents langages ?

Dans ce chapitre, nous avons présenté une série de travaux qui nous ont paru présentement représentatifs de l'état de l'art en résumé automatique extractif, en tenant compte des différentes méthodes et métriques pour la sélection ou extraction des attributs les plus importants ainsi comme la présentation et comparaison des mesures de performance des modèles obtenus. Ces travaux sont issus de trois approches qui sont l'approche statistique, l'approche statistique enrichie et l'approche par apprentissage automatique.

Comme nous l'avons établi, les travaux ont des forces mais aussi des faiblesses et des lacunes. Ces dernières amènent des questions de recherche pour proposer de nouvelles méthodes et améliorations. Parmi ces questions, nous avons tenté de proposer un modèle pour améliorer le processus de résumé extractif à partir de l'approche par apprentissage automatique, dont les ensembles d'entraînement ont été amélioré par la connaissance ontologique. Cette affirmation représente notre objectif principal de recherche qui nous a guidé tout au long de notre travail de recherche. Cet objectif est détaillé dans le prochain chapitre ainsi que la démarche suivie pour l'atteindre.

CHAPITRE 3. Objectifs de recherche et démarche suivie

Maintenant que nous avons exposé notre problématique, nous présentons l'objectif général de notre travail de recherche ainsi que les objectifs spécifiques reliés. Nous expliquerons ensuite les différentes étapes que nous avons suivies de façon générale et qui nous ont permis de réaliser notre recherche. Pour cela, nous avons utilisé une méthodologie basée sur l'approche action-recherche et expérimentation présentées par (Dawson, 2005). Celle-ci s'appuie en principe sur l'apprentissage par la pratique : on identifie un problème, on prend une action pour le résoudre, on voit si l'effort a été fructueux, s'il ne l'a pas été alors on essaie à nouveau.

C'est ainsi, que dans les sections suivantes nous exposons les éléments théoriques et les processus dans les différentes étapes pour la création et l'optimisation de l'espace d'attributs ainsi que dans l'induction des différents modèles d'apprentissage automatique pour définir notre modèle général.

3.1 Objectifs

Notre objectif principal était de proposer un nouveau modèle d'apprentissage automatique destiné à la production extractive et automatique de résumés, plus efficace que ceux actuellement proposés dans les travaux de recherche. Cette efficacité est dans le sens d'identifier les concepts les plus importants ou plus porteurs d'information d'un document ou d'un ensemble de documents.

Même si la construction de notre modèle s'est fait par tentatives successives dans le but constant de proposer un modèle plus performant que ceux considérés dans notre revue de littérature, nous avons au départ identifié trois objectifs spécifiques pour guider notre démarche tout au long de ce travail de recherche. Nous voyons maintenant en détail chacun de ces objectifs.

3.1.1 Déterminer un ensemble d'entraînement performant

Dans un premier temps, il est important de déterminer comment représenter l'espace d'attributs/concepts. En d'autres termes, l'ensemble des phrases doivent être représentées et reliées entre elles dans un espace de recherche, dans lequel on sélectionne ou on extrait l'ensemble d'entraînement.

Étant donné la présence de données entropiques dans cet espace de recherche, une phase de réduction/optimisation s'avère nécessaire. Pour cela, nous avons décidé d'utiliser les techniques d'exploration de données et de reconnaissance de formes.

Mais, à notre avis, cette phase de réduction n'est pas suffisante pour obtenir un ensemble d'entraînement performant c'est-à-dire qui soit vraiment capable de capturer les concepts les plus importants et les moins importants avec le minimum possible d'entropie. L'utilisation de la connaissance ontologique, comme effectuée dans divers travaux du domaine, nous a paru une excellente alternative pour améliorer cette performance, en tenant compte du renforcement sémantique qu'elle pourrait apporter aux concepts importants d'un document. Les résultats de ces travaux étaient importants dans la mesure qu'ils montrent la bonté d'utiliser des concepts reliés entre eux. Nous en avons tenu compte pour une utilisation différente mais tout aussi intéressante, voire plus fructueuse. C'est ainsi qu'est apparue une importante question pour notre recherche : l'introduction de connaissance ontologique dans l'espace d'attributs permet elle d'optimiser le choix de l'ensemble d'entraînement pour l'abduction de fonctions de classification de phrases ? En d'autres mots, l'introduction de connaissance ontologique influence-t-elle positivement la performance du modèle de résumés extractifs ? Il a été alors important de nous demander comment appliquer la connaissance ontologique à des espaces de recherche dans le but d'optimiser le choix de l'ensemble d'entraînement ? Nous avons donc élaboré un algorithme pour ajouter cette connaissance ontologique, le but étant de renforcer la sémantique des phrases en ajoutant des ensembles de mots synonymes, hyponymes et méronymes. Cet algorithme accède à l'ontologie de WordNet pour extraire les ensembles souhaités.

Même si l'ajout de connaissance ontologique présente théoriquement des avantages, il nous a paru nécessaire de tester cet avantage dès l'obtention des arbres de l'ontologie WordNet en mesurant la quantité l'information apportée. Nous avons proposé d'utiliser les différentes métriques de similarité, suggérées dans la littérature et de choisir celle ou celles qui correspondaient le mieux à l'objectif de notre travail.

3.1.2 Élaborer un processus d'abduction efficace

Les fonctions d'apprentissage ou de classification des futures phrases sont abduites des ensembles d'entraînement. Nous avons donc dû définir le processus d'abduction de ces fonctions et nous assurer qu'il soit efficace. Pour cela, nous avons décidé d'expérimenter un certain nombre d'algorithmes d'apprentissage automatique, largement utilisés dans les domaines de l'exploration de données et la reconnaissance de formes et qui ont montré leur efficacité. Nous pensons qu'il est préférable de mettre l'accent sur l'optimisation de l'ensemble d'entraînement pour trouver des bonnes fonctions de classification plutôt que sur le développement d'un nouvel algorithme d'apprentissage automatique, en sachant qu'un ensemble d'entraînement optimisé est garant de trouver les meilleurs paramètres pour ces fonctions.

Toutefois, il nous a paru nécessaire d'évaluer le pouvoir discriminant des fonctions d'apprentissage abduites. Nous avons proposé d'évaluer ce pouvoir discriminant au moyen de métriques couramment utilisées dans les domaines de l'extraction d'information et de la récupération d'information, par exemple, la précision, le rappel, la F_mesure (Manning *et al.*, 2008) et les courbes ROC (Lasko *et al.*, 2005), lesquelles seront analysées par différentes itérations avec divers paramètres. Pour appliquer ces métriques, nous devons créer des matrices de confusion ou des matrices de contingence, dont chaque ligne représente le nombre d'occurrences d'une classe réelle (par exemple la classe des phrases importantes), tandis que les colonnes représentent le nombre d'occurrences d'une classe estimée. À partir de ces données, nous pouvons calculer facilement le nombre de succès ou d'échecs de la fonction d'apprentissage dans son but d'identifier les phrases importantes ou porteuses d'information. Nous nous assurons ainsi d'avoir choisi le meilleur algorithme d'apprentissage.

3.1.3 Définir une procédure d'évaluation du modèle global

Afin de finaliser notre travail de recherche, il est important de mettre en place une procédure d'évaluation des résultats globalement obtenus. La procédure consiste donc à expérimenter le modèle proposé sur un corpus significatif et évaluer les résumés obtenus avec une méthode d'évaluation reconnue dans le domaine du résumé automatique, afin de pouvoir comparer nos résultats à ceux trouvés dans la littérature.

Nous avons expérimenté le modèle proposé sur des corpus de DUC -Document Understanding Conference- (NIST, 2006), habituellement utilisés pour ce type de recherche. Ces corpus sont intéressants car les utilisateurs peuvent accéder à un ensemble de documents textuels sur différents sujets, ainsi qu'aux résumés de ces mêmes documents produits par des spécialistes humains. L'idée est donc d'appliquer les fonctions de classification obtenues sur des ensembles de phrases non vues afin d'identifier celles qui sont les plus importantes et qui feront partie du résumé.

Pour compléter la procédure d'évaluation et ajuster le modèle au besoin, nous proposons d'évaluer les résultats avec le logiciel ROUGE, élaboré par Lin (2004). Ce dernier permet en effet de comparer le résumé produit automatiquement avec celui produit par un humain. En fonction des résultats obtenus, des ajustements peuvent toujours être faits dans la construction des ensembles d'entraînement.

Les sections suivantes montrent comment nous avons atteint l'ensemble de ces objectifs spécifiques en utilisant une démarche de recherche-action avec expérimentation. Les étapes cruciales de notre démarche ont fait l'objet de trois publications qui seront à leur tour présentées. Pour chacune d'elles, une expérimentation a été menée et a permis de spécifier notre modèle de résumé automatique extractif.

3.2 Préparation de l'ensemble d'entraînement

Cette première étape explique comment l'ensemble d'entraînement initial a pu être construit. D'abord, nous avons défini un modèle de l'espace de variables d'entraînement ainsi que les éléments de cet espace, puis une méthode de créer un ensemble d'entraînement.

3.2.1 Définition du modèle de l'espace de variables d'entraînement

Le premier problème à résoudre a été celui de la représentation de l'espace d'exemples possibles d'où on extraira la connaissance nécessaire pour induire un modèle qui classera les phrases (des plus importantes aux moins importantes) d'un texte quelconque. Il est nécessaire de disposer d'un modèle pour représenter et mettre en relation l'ensemble d'unités lexicales du texte, que ce soit pour les mots ou les phrases. Nous avons choisi de

représenter cet espace au moyen d'un modèle algébrique linéaire, pour fournir un ensemble d'éléments théoriques pour la manipulation et la transformation de l'espace.

Ensuite, il a été important d'identifier les éléments qui composent l'espace d'entraînement initial et qui sont la base pour la configuration de l'espace d'attributs. Il a été ici nécessaire de définir ce qu'allait être notre unité de texte et ses éléments, de telle sorte que ces unités devenaient nos éléments d'analyses primaires pour déterminer leur teneur en information. Dans notre modèle, les phrases des documents sont les unités lexicales et les mots les éléments de ces unités. Cet arrangement peut facilement être représenté par un tableau, les lignes indiquant les phrases qui contiennent un mot donné et les colonnes les mots contenus dans une phrase donnée. Nous avons pu ainsi tester ce type de représentation sur un corpus de documents.

De plus dans les phrases, il existe souvent une quantité très grande de mots qui ne contiennent pas d'information significative et qui au contraire contribuent à l'augmentation de l'entropie de l'espace d'entraînement. L'entropie ici correspond à des éléments redondants et non pertinents qui produisent beaucoup d'erreurs et de biais significatifs dans l'estimation des fonctions. Il est alors nécessaire d'identifier ces mots pour les éliminer. Par exemple, on citera, dans la langue anglaise, les mots tels que « and, or, the, etc. » et, dans la langue espagnole, les mots « y, o, el, etc. » ou encore dans la langue française les mots « et, ou, le, etc. ». Comme nous le mentionnons, ces mots, qui sont répétés dans beaucoup de phrases, en plus de produire de la redondance et d'une certaine manière affecter la fonction d'induction de connaissances, sont une charge pour la complexité computationnelle et spatiale, conduisant ainsi au phénomène du fléau de la dimension ou malédiction de la dimension (Bellman, 1970). En d'autres termes, le volume de l'espace augmente rapidement si bien que les informations se retrouvent de manière isolée et deviennent éparses, donc peu efficaces pour notre recherche d'information. Dans cette optique, nous avons ajouté un processus d'élimination des mots ne contenant pas d'information significative.

Nous nous sommes également demandé s'il serait profitable d'éliminer des mots avec une racine commune. En tenant compte des réflexions de Sebastiani (2002) qui affirme qu'une telle action pourrait, au contraire, affecter de manière importante l'extraction de la

connaissance la plus significative ou éventuellement influencer négativement l'identification des attributs les plus significatifs, nous nous sommes résolus à ne pas le faire.

3.2.2 Création de l'ensemble d'entraînement

Une fois une partie de l'entropie de notre ensemble de phrases éliminée comme proposé dans la section précédente, nous avons suggéré de construire un arrangement qui contient des valeurs pour chacun des mots (éléments) qui composent chaque phrase et qui formeront l'espace initial d'entraînement. Un problème immédiat est apparu et concerne la définition de la fonction qui assigne des valeurs à chacun des éléments de cet arrangement, afin que ressorte l'importance de cet élément non seulement dans la phrase, mais aussi dans l'ensemble de phrases. En d'autres termes, si, par exemple, nous prenons la fréquence d'apparition du mot dans la phrase comme l'indicatif de son importance dans celle-ci, nous serions intéressé à la relier à son degré de dispersion dans l'ensemble de toutes les phrases, sachant qu'un haut degré de dispersion (peu d'entropie) indiquera un fort contenu d'information. De la même manière, nous pourrions simplement assigner une valeur booléenne à chaque élément de l'arrangement en tenant compte du fait qu'un mot pourrait apparaître ou non dans une phrase donnée.

Pour notre modèle, nous avons élaboré un algorithme pour assigner les valeurs en tenant compte des considérations précédentes, en nous basent sur la formule de Salton et Buckley (1988). Selon ces auteurs, la valeur discriminatoire d'un terme, par rapport à un corpus de documents peut être mesurée au moyen de la formule : tf*idf, où : tf est la fréquence du terme dans un document et idf est la fréquence inverse, calculée comme ln (N/n), N étant le nombre total de documents et n le nombre de documents dans lequel est présent le terme.

En apprentissage automatique, il est maintenant crucial de définir un sous-espace d'entraînement à partir de tout l'espace de variables disponible, étant donné que nous ne pouvons pas sélectionner tout cet espace, puisque cette sorte de sélection rendrait le problème intraitable. Ce sous-espace est un ensemble d'exemples utilisés par un algorithme pour identifier les patrons les plus importants qui mettent en relation l'exemple avec une classe déterminée. Cet algorithme permet ensuite de produire le modèle correspondant qui

sera appliqué ultérieurement à de nouvelles instances. En général, pour résoudre ce problème dans le contexte du résumé automatique de textes, les stratégies proposées sont nombreuses et sont principalement basées sur des métriques qui sont appliquées à chacune des phrases. Ces stratégies sont par exemple le gain d'information (IG, terme en anglais), le test du $\chi 2$, le ratio de probabilité logarithmique (LogProbRatio), la position de la phrase dans le texte, la longueur de la phrase, le contenu des mots par rapport au titre, les phrases d'entrée ou phrases-repère, etc. Dans notre modèle, les exemples sont un ensemble de phrases d'un corpus de documents, divisées en deux classes en lien avec leur importance dans le texte évaluée en fonction de la quantité d'information que ces phrases peuvent apporter. Cette importance est mesurée à partir de la variance des termes des phrases et le degré d'entropie entre ces termes. En d'autres termes, la variance dépend de la teneur en information en se basant sur l'idée classique d'entropie de Shannon (1948). Les phrases ayant le plus grand contenu d'information sont définies comme importantes (classe 1) et les phrases avec moins de teneur en information sont définies comme non importantes (classe -1). Pour atteindre cet objectif, nous avons utilisé des méthodes algébriques qui sont décrites plus précisément dans le chapitre 4, section 4.4

3.3 Ajout de la connaissance ontologique

L'étape suivante dans notre recherche consistait à déterminer la connaissance ontologique à introduire et comment l'ajouter à notre espace d'entraînement. Notre hypothèse, rappelons-la, est que si l'espace d'entraînement est renforcé avec cette connaissance, les termes ou les éléments de cet espace contiendront plus de sens et par conséquent un plus fort potentiel discriminant. L'introduction de cette connaissance a aussi pour but de contribuer à résoudre en partie le problème de la synonymie un des grands problèmes du domaine de la recherche d'information (Sharan et Imran, 2009). Avant de présenter l'algorithme nécessaire à l'introduction de la connaissance ontologique, nous décrivons ce que nous entendons par connaissance ontologique.

3.3.1 La connaissance ontologique

Une ontologie est une représentation explicite d'une conceptualisation ou encore une description des concepts et les relations qui peuvent exister pour un agent ou une communauté d'agents (Gruber, 1995). En d'autres termes, une ontologie est le résultat de la

sélection d'un domaine et de l'application d'une méthode sur ce dernier, pour obtenir une représentation formelle des concepts qu'il contient et des relations qui existent entre ces derniers.

Pourquoi une ontologie est-elle développée ? Selon Noy et Hafner (1997), les buts sont les suivants:

- partager une compréhension commune des structures d'information entre des personnes ou des agents logiciels ;
- aider la réutilisation de domaines de connaissance (Si par exemple un groupe de chercheurs développent une ontologie détaillée, d'autres chercheurs peuvent la réutiliser pour construire leur domaine) ;
- permettre de faire des suppositions explicites d'un domaine (Il est possible de changer facilement les suppositions si notre connaissance du domaine est modifiée);
- séparer des domaines de connaissance de la connaissance opérationnelle (On peut décrire une tâche pour configurer un produit et ses composants en fonction d'une spécification qui est requise et implémenter un programme indépendant de ces produits. On peut par exemple développer une ontologie des composants d'un ordinateur et en vue de l'utiliser pour des commandes d'ordinateurs sur mesure) ;
- analyser des domaines de connaissance (Les méthodes pour la résolution de problèmes, les applications indépendantes du domaine et les agents logiciels utilisent les ontologies et les bases de connaissances construites à partir d'une ontologie donnée. Par exemple, si on construit une ontologie des vins et des repas, on peut l'utiliser comme base pour différentes applications reliées aux activités d'un restaurant).

Les ontologies peuvent être classifiées selon différents niveaux et catégories. Cinq niveaux sont proposés selon l'utilité des ontologies, soit les niveaux d'abstraction, de formalisme, de complétude et de détail. Le premier niveau, le niveau d'abstraction se décompose en sept sous-niveaux soit niveau application, niveau du domaine, niveau générique, niveau de représentation, niveau des tâches, niveau des inférences, niveau supérieur. Le deuxième niveau, le niveau de formalisme, est en lien avec le formalisme du langage de représentation utilisé pour rendre l'ontologie fonctionnelle. Le troisième niveau, le niveau

de complétude, est en lien avec le sens du concept. Le quatrième niveau, le niveau de détail, tient compte de la granularité de l'ontologie, fine ou grosse, selon l'objectif opérationnel prévu de l'ontologie. Les ontologies peuvent également être classées en accord avec des aspects généraux du monde, selon quatre catégories : statiques, dynamiques, sociales et intentionnelles. Le lecteur pourra trouver une description détaillée de ces différents niveaux et catégories dans le document synthèse de Motta (2006).

Depuis le début des années 90 et comme nous l'avons mentionné dans des différentes sections du chapitre 2, la connaissance ontologique a été ajoutée dans quelques développements pour renforcer l'utilisation d'heuristiques dans la sélection des phrases les plus représentatives d'un document. C'est ainsi, que nous avons mis au point un nouvel algorithme pour rendre cet ajout plus efficace.

3.3.2 Algorithme d'introduction de la connaissance ontologique

Pour notre modèle, nous avions besoin de concevoir un algorithme pour accéder, récupérer et calculer des structures conceptuelles à partir d'ontologies, afin que ces structures soient introduites dans chacun des termes de l'espace initial d'entraînement. Le but d'un tel algorithme est d'être capable de donner en sortie une chaîne lexicale ou des chaînes grammaticales formées à partir des mots des phrases de l'espace initial et d'intégrer leurs valeurs à cet espace. La ou les chaînes doivent être optimales, ce qui suppose une analyse et une comparaison entre elles. Elles doivent aussi être composées de nouveaux concepts reliés entre eux et avec le concept (phrases) de cet espace. D'une certaine manière, ces chaînes sont un facteur qui ajoutera une valeur sémantique aux éléments de cet espace, en renforçant leur classification : significative (phrase importante) ou non significative (phrase non importante).

L'algorithme, de manière générale, sera composé des processus suivants :

- accès à l'ontologie;
- accès aux différents concepts qu'elle contient en parcourant différents sous-arbres;
- évaluation des différents arbres trouvés;
- choix du meilleur arbre;
- intégration de l'arbre choisi à l'espace de variables.

La première étape que nous avons réalisée consiste en l'identification des catégories lexicales principales de la phrase (noms, verbes et adjectifs) comme le propose Robins (1989). Ce processus est en général appelé POST (Part-of-Speech-Tagging) ou simplement analyse grammaticale et est normalement utilisé en traitement automatique du langage naturel. Une fois les catégories lexicales identifiées dans chacune des phrases, l'algorithme mis au point accède à l'ontologie en faisant correspondre chaque mot de la phrase au nœud adéquat dans l'ontologie. Ainsi, les sous-arbres sémantiques correspondants associés à chaque mot, en fonction de sa catégorie lexicale, sont identifiés et peuvent être ajoutés à l'espace d'attributs. Ces sous-arbres sont notés par $G = (V, E)$, où V est l'ensemble de mots de la phrase et E leurs relations lexicales. La profondeur des sous-arbres correspondants sera variée afin d'analyser son impact sur la performance générale du modèle. La relation sémantique entre les mots et les arbres peut être déterminée de manière générale par synonymie, hypéronymie, hyponymie et méronymie (Robins, 1989). Nous avons analysé cette situation en fonction de la complexité du calcul et de l'espace.

Dans la figure 1, nous pouvons observer la représentation de la phrase « le chat est un animal domestique » en un arbre avec deux catégories lexicales principales (nom et adjectif). Par exemple, les ensembles du sous-arbre sémantique G associés à l'adjectif « animal » sont : $V = [il\ est\ animal, bête, brute, créature, faune]$ et $E = [synonyme]$. Ainsi, le sous-arbre associé au nom « animal » sera introduit à l'espace initial d'entraînement des attributs.

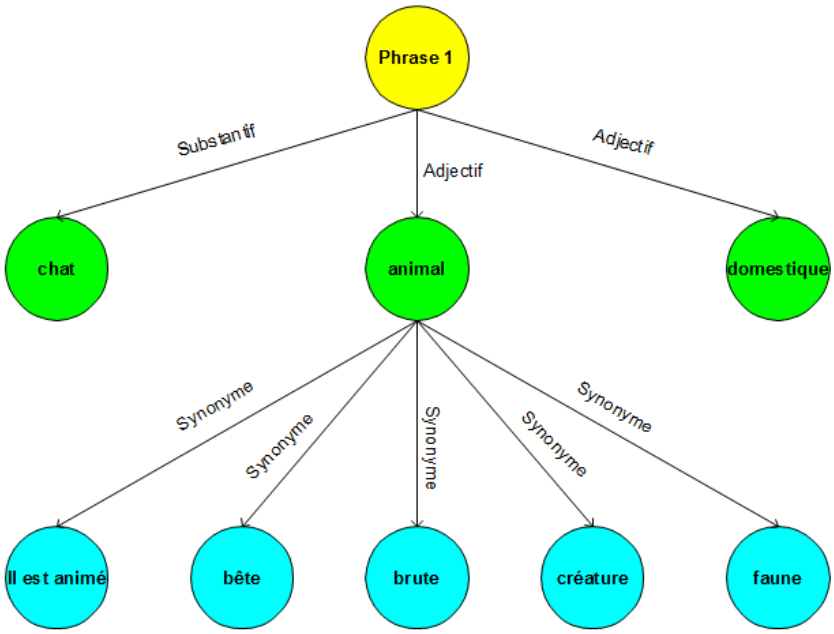


Figure 3.1 *Représentation d'une phrase par un arbre sémantique*

Après avoir introduit ces sous-arbres dans l'espace d'attributs, notre algorithme identifie toutes les relations avec chacune des phrases de cet espace, ce qui renforce le contenu et le degré de cohésion sémantique de l'espace (Motta *et al.*, 2011). L'algorithme utilisé pour ajouter de la connaissance ontologique est décrit en détail dans la section 5.2. Dans cette section, le lecteur pourra trouver toute l'information relative à la mise au point de cet algorithme. L'expérimentation décrite est une évaluation de l'amélioration obtenue en ajoutant la connaissance ontologique à l'ensemble d'entraînement. Pour cela, nous avons utilisé quatre algorithmes d'apprentissage automatique parmi ceux couramment utilisés : Bayes naïf, la machine à vecteurs de support, les arbres de décision et le perceptron multicouche. Ces algorithmes ont été appliqués sur des ensembles d'entraînement pour lesquels nous avions appliqué l'algorithme proposé et sur les mêmes ensembles d'entraînement sans cette application. Une comparaison en utilisant les métriques habituelles, qui seront plus amplement décrites dans la démarche suivie (section 5.6) a montré une nette amélioration des résultats obtenus lorsqu'il y a renforcement avec la

connaissance ontologique. La plus grande amélioration est toutefois obtenue avec les algorithmes Bayes naïf et de la machine à vecteurs de support (Motta *et al.*, 2011).

Même si l'ajout de connaissance ontologique permet d'améliorer le pouvoir discriminant des algorithmes d'apprentissage automatique, il n'en reste pas moins que l'extraction des concepts de l'ontologie pour créer les sous-arbres sémantiques dépend de la similarité des concepts avec l'attribut considéré. Nous avons décidé de tester cinq mesures de similarité connues afin de déterminer celle qui était la plus appropriée pour notre modèle. Ces cinq mesures sont : Path (Fellbaum, 1985), Leacock-Chodorow (Leacock *et al.*, 1998), Wu-Palmer (Wu et Palmer, 1994), Resnick (Resnick, 2000), Jiang-Conrath (Jiang et Conrath, 1997) et Lin (Lin, 1998).

a. Path : Cette mesure permet d'évaluer la similarité en calculant le chemin le plus court séparant les synsets (figure 3.2).

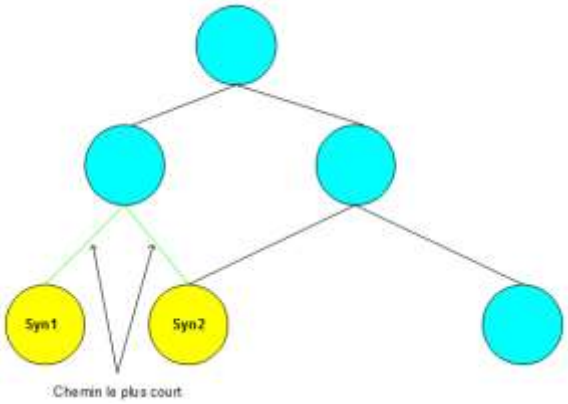


Figure 3.2 *Calcul de la similarité avec Path*

b. Leacock-Chodorow (LCH) : Cette mesure permet de trouver la similarité en considérant le chemin et la profondeur de l'ontologie. Cette mesure est calculée à l'aide de l'équation suivante, dans laquelle *length* est le chemin le plus court et D est la profondeur maximale :

$$LCH = \frac{-\log(lenght)}{2 \times D} \tag{3.1}$$

c. Wu-Palmer (WUP) : Cette mesure tient compte de la profondeur (*depth*) des deux synsets et de la profondeur de l'ancêtre commun le plus proche (*lcs*). On utilise l'équation suivante :

$$WUP = \frac{2 \times depth(lcs)}{depth(syn1) + depth(syn2)} \tag{3.2}$$

d. Jiang-Conrath (JCN) : C'est une combinaison du nombre d' "arches" dans la hiérarchie "est-un" de WordNet et le contenu d'information (IC) des concepts. L'équation est la suivante :

$$JCN = \frac{1}{jcn\_dist}$$
$$jcn\_dist = \mathrm{IC}(\mathrm{syn1}) + \mathrm{IC}(\mathrm{syn2}) - 2 * \mathrm{IC}(\mathrm{lcs}) \tag{3.3}$$

e. Lin. (Lin) : Cette mesure permet de calculer la relation sémantique en utilisant le contenu d'information des concepts et le "théorème de similitude" qui le soutient, soit :

$$Lin = \frac{2 \times IC(lcs)}{IC(syn1) + IC(syn2)} \tag{3.4}$$

f. Resnick (Res). Cette mesure utilise le contenu de l'information des concepts, qui est calculée à partir de la probabilité de leur occurrence dans un vaste corpus de documents. Ainsi, on utilise le contenu d'information de l'ancêtre commun le plus proche (LCS) des deux synsets d'entrée, comme le montre la figure 3.3

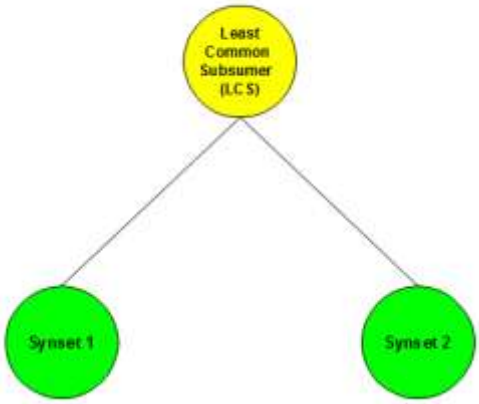


Figure 3.3 *Calcul de la similarité avec Resnick*

Dans le chapitre 6, la section 6.6.1 décrit l'expérimentation que nous avons menée pour déterminer la mesure de similarité la plus appropriée, en calculant la quantité d'information apportée par les lexèmes introduits dans la phrase. Cette expérimentation nous a permis d'opter pour la mesure de similarité de Wu-Palmer, car elle donne la meilleure valeur pour le contenu d'information et permet de déterminer quand l'ajout de connaissance ontologique est suffisant.

3.4 Optimisation de l'ensemble d'entraînement

Nous avons expliqué comment construire un ensemble d'entraînement à partir d'un corpus de document(s) et comment le renforcer en introduisant une certaine connaissance ontologique. Mais notre travail de recherche nous a permis de conclure que cet ensemble d'entraînement ne pouvait pas être utilisé directement par un algorithme d'apprentissage pour abduire une fonction de classification. En effet, il pourrait être optimisé dans le but d'augmenter le pouvoir discriminant de la fonction de classification abduite. Dans cette section, nous décrivons comment optimiser l'ensemble d'entraînement, soit en filtrant les attributs entropiques et en classifiant les éléments du nouvel ensemble d'entraînement.

3.4.1 Filtrage d'attributs entropiques

L'ensemble d'entraînement considéré est généralement entropique et très grand. La quantité de données est de l'ordre de dizaines de milliers voire plus. Ces données contiennent de l'information sur les aspects les plus significatifs de cet ensemble qu'il est nécessaire d'identifier et de discriminer, ainsi que de quantifier et de retenir cette information. Cette information sera ensuite transmise aux fonctions d'induction de connaissance pour obtenir notre modèle.

Rappelons que notre ensemble d'entraînement est composé d'un arrangement de phrases enrichi par de la connaissance ontologique. Dans cet espace de recherche, il est nécessaire d'identifier les caractéristiques les plus significatives et celles non significatives associées avec chacune des phrases, pour que le programme envisagé « apprenne » à les distinguer et, postérieurement, qu'il puisse profiter de cette nouvelle connaissance pour discriminer les mêmes caractéristiques à partir d'un nouvel espace de recherche. Nous avons proposé de filtrer les attributs entropiques au moyen de méthodes algébriques linéaires. En effet, à partir de la représentation de l'espace d'attributs proposée, nous utilisons des transformations matricielles de ressemblance pour trouver des espaces plus réduits et moins redondants et dans lesquels il est possible d'identifier facilement deux classes d'attributs : significatif et non significatif.

Une transformation matricielle de ressemblance peut être définie comme :

« Étant donnée une matrice carrée A et une matrice inversible T (matrice avec son inverse), alors la matrice $B=T^{-1}AT$ est appelée matrice semblable à A et l'opération $T^{-1}AT$ est une transformation de ressemblance ».

Cette transformation, en plus de conserver ses propriétés de base de réflexivité, de symétrie et de transitivité, a des caractéristiques invariantes : le déterminant, la trace (somme de ses valeurs propres), ses valeurs propres et ses vecteurs propres. Or, si la matrice A, $n \times n$, a n vecteurs propres linéairement indépendants et qu'ils forment une matrice T dont les colonnes sont ces vecteurs, alors la transformation $D = T^{-1}AT$ produit une matrice diagonale D (*Théorème Spectral*). En outre, les éléments de D seront justement les valeurs propres (*vecteurs propres*) de la matrice A.

Le théorème spectral montre l'importance des valeurs propres et des vecteurs propres pour caractériser une transformation linéaire de manière unique. Dans sa version la plus simple, le théorème spectral établit que, sous des conditions déterminées, une transformation linéaire d'un vecteur peut être exprimée comme la combinaison linéaire de ses vecteurs propres avec des coefficients de valeur égale à ses valeurs propres par le produit des vecteurs propres par le vecteur auquel on applique la transformation, ce qui peut être écrit comme suit:

$$\mathcal{T}(\mathbf{v}) = \lambda_1(\mathbf{v}_1 \cdot \mathbf{v})\mathbf{v}_1 + \lambda_2(\mathbf{v}_2 \cdot \mathbf{v})\mathbf{v}_2 + \dots \tag{3.5}$$

où $\mathbf{v}_1, \mathbf{v}_2, \dots$ et $\lambda_1, \lambda_2, \dots$ représentent les vecteurs propres et les valeurs propres de $\mathcal{T}$. Le cas le plus simple qui valide le théorème est quand la transformation linéaire est donnée par une matrice symétrique réelle, une matrice *A* étant symétrique si un élément a_{ij} est égal à l'élément a_{ji}, ou une matrice *hermitique* (ou *autoadjointe*) complexe, soit lorsqu'elle est égale à sa propre *adjointe* (Fraleigh et Beauregard, 1994)

Nous pouvons calculer les valeurs propres d'une matrice carrée *A* à partir de l'équation $AX = \lambda X$, où λ est un vecteur de valeurs scalaires et $X \neq 0$ est un arrangement de vecteurs scalaires. En réarrangeant l'équation $AX-\lambda X = 0$ *en* $(A-\lambda) X = 0 = (A-\lambda I) X$, où *I* est la matrice identique. C'est un système homogène de la forme $BX = 0$, qui a une solution unique $X = 0$, quand $det\ (B) \neq 0$.

Comme nous avons établi que $X \neq 0$, alors, pour que $(A-\lambda I)\ X = 0$, le déterminant de $A-\lambda I$ doit être égal à zéro, soit $det\ (A-\lambda I) = 0$ et correspond à *l'équation caractéristique* d'*A*. Ce déterminant est un polynôme en puissances de λ. Pour cette raison, on l'appelle le *polynôme caractéristique* de *A*. Ainsi, si λ est une *valeur propre* de *A* et si *X* est un vecteur non nul tel que $AX = \lambda X$, on dit alors que *X* est un vecteur propre de *A*.

À une *valeur propre* λ correspond en général des infinis *vecteurs propres*, qui composent un espace vectoriel. Cet espace est appelé *espace* propre de λ. Nous pouvons mentionner quelques propriétés de base intéressantes des valeurs propres :

a) La somme des n valeurs propres de la matrice A, est égale à sa trace : $\lambda_1+\lambda_2+\dots+\lambda n = trace\ (A)$.

b) Le produit des valeurs propres est égal à son déterminant *:* $\lambda_1\lambda_2 \ldots \lambda n = det\ (A)$.

c) Les valeurs propres d'une matrice triangulaire supérieure (matrice dont les éléments sous la diagonale principale sont égaux à zéro) sont les éléments de leur diagonale principale.

Il existe plusieurs méthodes algébriques linéaires et non linéaires qui peuvent permettre de réduire l'espace de recherche. Afin de déterminer quelle méthode serait la plus appropriée, nous avons mené une expérimentation et étudié ainsi les performances de cinq méthodes (Motta *et al.*, 2012). Ces méthodes, utilisées dans le partitionnement de données et l'apprentissage automatique non supervisé, agissent selon deux approches, soit en sélectionnant un ensemble d'attributs considérés comme les plus pertinents, soit en extrayant certains attributs pour former un nouvel ensemble d'entraînement. Pour la première approche, nous avons étudié la décomposition en valeurs singulières, l'algorithme des k-moyennes et les réseaux de neurones de Kohonen. Quant à la deuxième, nous avons retenu l'analyse en composantes principales et l'analyse factorielle. Le chapitre 4 décrit cette expérimentation ainsi que les résultats obtenus. Tout d'abord, nous avons expliqué comment ces cinq méthodes ont été utilisées dans le contexte du résumé automatique et ont permis d'optimiser l'espace d'attributs pour enfin abduire une fonction de classification. Le processus d'abduction a été testé avec six algorithmes bien connus d'apprentissage automatique et l'exploration de données. Ce test s'est avéré utile pour s'assurer de ne pas créer de biais pour cette dernière étape du processus de résumé automatique proposé. La phase de validation décrite a permis de mettre en évidence le pouvoir de discrimination de la fonction de classification abduite, et ceci en utilisant la F_measure et les courbes ROC. Les résultats obtenus montrent que l'application des cinq méthodes linéaires choisies pour optimiser l'espace d'attributs est une option pertinente pour le processus de résumé automatique par extraction (Motta *et al.*, 2012). En conclusion, cette expérimentation nous a permis de faire un choix éclairé pour les méthodes linéaires à utiliser pour réduire l'ensemble d'entraînement. Nous avons donc opté pour la décomposition en valeurs singulières et l'analyse en composantes principales pour notre modèle VENCE de résumé automatique de textes. Bien que ces deux méthodes linéaires soient décrites dans le chapitre 5, nous pensons que leurs utilisations dans notre contexte méritent une attention particulière. En effet, dans les deux cas, la procédure mathématique a dû être convertie en

une méthode d'identification des phrases porteuses d'information. De même, nous mettons en évidence le concept de variance des données et de l'information apportée, concept utile pour les différentes expérimentations. Ces aspects ne sont pas aussi détaillés dans le chapitre 5, aussi nous préférons le faire dans cette section.

3.4.1.1 Décomposition en valeurs singulières (SVD)

Pour le cas de matrices générales m×n, on applique le *théorème de décomposition* (Horn et Johnson, 1985), qui est une version du *théorème spectral* (Halmos, 1963; Helson, 1986; Hilbert *et al.*, 1927; Reed et Simon, 1975). Dans l'ensemble, il peut être vu comme une transformation d'un système de coordonnées à un autre, où les colonnes d'une matrice inversible U sont les composants de la nouvelle base de vecteurs exprimés dans des termes de la base précédente. Dans ce nouveau système, les coordonnées du vecteur v sont représentées par v', qui peut être obtenu par la relation $v' = Uv$ et, d'autre part, on a $v = U^{-1}v'$. En appliquant successivement $v' = Uv$, $w' = Uw$ et $U^{-1}U = I$, à la relation $Av = w$, on obtient $A'v' = w'$ avec $A' = UAU^{-1}$, qui est la représentation de A dans la nouvelle base. Dans ce cas, on dit que les matrices A et A' sont semblables.

Le *théorème de décomposition* spécifie que, si on choisit comme colonnes de U^{-1} n vecteurs propres linéairement indépendants de A, la nouvelle matrice $A' = UAU^{-1}$ est diagonale et ses éléments dans la diagonale sont les valeurs propres de A. La matrice A est alors une matrice *diagonalisable* (Horn et Johnson, 1985).

Il est important de remarquer que les matrices n'ont pas toutes une forme diagonale. Pour traiter ces cas, on a développé différentes généralisations de la décomposition précédente, comme par exemple:

- triangulation de Schur (Golub et Kahan, 1965): toute matrice peut être exprimée comme une matrice triangulaire. Si A est une matrice carrée de valeurs complexes, alors A peut être décomposée comme $A = QUQ^T$, où Q est une matrice unitaire (matrice dont le produit par sa transposée est égal à la matrice identique) et U une matrice triangulaire supérieure dont les entrées diagonales sont précisément les valeurs propres de A. Si la matrice A a des valeurs réelles, on dit alors que c'est une *matrice orthogonale* (Hastie et al., 2009).

- décomposition dans des valeurs singulières : la matrice *A,* une matrice m×n, peut être exprimée comme $A = U\Sigma V^T$, où Σ est une matrice diagonale, U et V sont des matrices unitaires. Les éléments de la diagonale de $A = U\Sigma VT$, soit la matrice Σ, ne sont pas négatifs et reçoivent le nom de valeurs singulières de la matrice A (Golub et Van Loan, 1996).

À travers une interprétation géométrique, nous pourrions avoir une idée plus intuitive des formulations que nous venons de mentionner. En fin de compte, l'idée est de trouver une base vectorielle pour exprimer les éléments d'une matrice, à travers un processus de diagonalisation.

Ainsi nous pouvons dire que les valeurs singulières d'une matrice A sont les longueurs des demi-axes d'une hyper-ellipse dans laquelle se transforme la sphère unité au moyen de la matrice A. Ce sont par conséquent des nombres réels, non négatifs : $s_1 \geq s_2 \geq ... \geq 0$ (Stewart, 1993).

À chaque matrice, nous pouvons associer les valeurs singulières, mais en général il y a beaucoup de matrices qui produisent la même ellipse comme image de la sphère unité. Toutes ces matrices auront les mêmes valeurs singulières. On dit alors qu'elles sont orthogonalement équivalentes. En général, elles sont définies dans l'espace des nombres complexes. Or, nous travaillons avec l'espace des réels.

Nous pouvons imaginer une matrice A et une paire de vecteurs unitaires (de longueur 1) qui sont perpendiculaires (o*rthogonaux*), x, y, sur un plan euclidien, avec l'origine dans le point (0.0) et pour lesquels on a calculé leurs images Ax et Ay au moyen de la matrice A. Maintenant, imaginons que nous tournons les vecteurs x et y en maintenant leur *orthogonalité*. Comme ceux-ci partent de l'origine du système de coordonnées, tous les deux décriront la circonférence unité et leurs images Ax et Ay décriront une ellipse avec aussi comme centre l'origine de coordonnées. Un fait remarquable à indiquer est qu'il y a

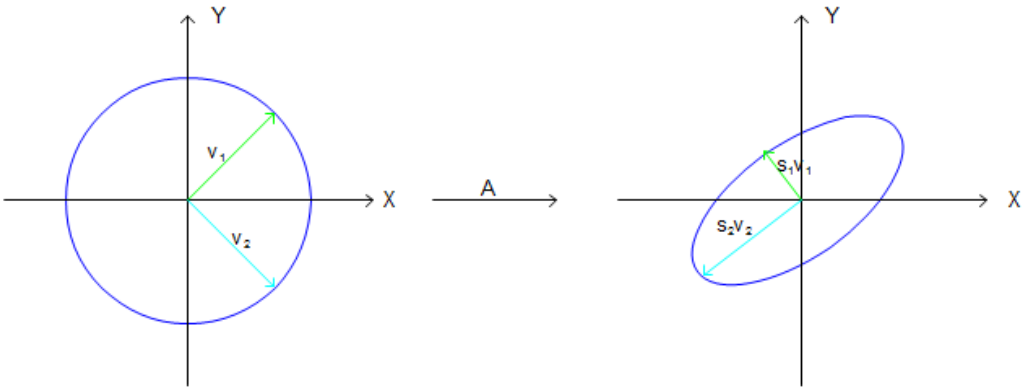


Figure 3.4 *Transformation au moyen des vecteurs singuliers*

une position des vecteurs x et y telle que leurs images sont les demi-axes de l'ellipse, c'est-à-dire que Ax et Ay sont perpendiculaires. De fait, il y a seulement deux positions des vecteurs x et y pour lesquelles Ax et Ay seraient les demi-axes.

Or, nous pouvons donner l'interprétation algébrique suivante:

si s_1 et s_2 sont les valeurs singulières de A, qui comme nous avons précisé sont les longueurs des demi-axes de l'ellipse, il existe alors des vecteurs u_1 et u_2 de longueur 1 tels que s_1u_1 et s_2u_2 sont les vecteurs qui forment les demi-axes de l'ellipse et par conséquent seront aussi *orthogonaux* (voir figure 3.4). Maintenant, si nous notons par v_1 et v_2 les vecteurs x et y, orthonormaux comme nous avons déterminé, alors Av_1 et Av_2 sont aussi les demi-axes de l'ellipse, avec $Av_1 = s_1v_1$ et $Av_2 = s_2v_2$, que nous pouvons écrire en manière matricielle :

$$A[v_1\, v_2] = [u_1\, u_2] * \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \tag{3.6}$$

En sachant que les vecteurs u_1 et u_2 sont orthogonaux, alors :

$$Q_1^T = \begin{bmatrix} u_1^T \\ u_2^T \end{bmatrix} \text{ y } Q_2^T = \begin{bmatrix} v_1 & v_2 \end{bmatrix} \tag{3.7}$$

et comme Q_1 et Q_2 sont orthogonaux, alors

$$\begin{aligned} y_j &= a_{j1}x_1 + a_{j2}x_2 + \ldots + a_{jp}x_p = a_j{}^T x \\ a_j{}^T &= (a_{1j}, a_{2j}, \ldots, a_{pj}) \end{aligned} \tag{3.8}$$

et

$$Q_2^T Q = Q_2 Q^T = I \tag{3.9}$$

En multipliant les deux côtés de l'équation (3.6) par Q1, nous obtenons un résultat, que nous pouvons exprimer comme suit:

$$A = \begin{bmatrix} u_1 u_2 \end{bmatrix} * \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} * \begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} = U \Sigma V^T \tag{3.10}$$

Les demi-axes de cette *hyper-ellipse* signalent toujours les directions de moindre redondance, soit de moindre entropie. Cela signifie les directions de variance maximale des données ou d'information maximale, dans le sens que la teneur en information d'un ensemble de données est intimement liée à son improbabilité ou valeur surprise. Plus elle nous surprend ou plus nous la jugeons peu probable ou inattendue, plus elle renferme d'information (Shannon, 1948).

Une autre analyse mathématique nous a paru significative et très importante pour notre recherche afin de caractériser et d'analyser notre espace initial d'entraînement. Elle est en rapport avec l'analyse mathématique précédente, dans le sens où son principe repose aussi sur l'obtention des *vecteurs propres* mais, contrairement à la première, la transformation des données est faite à partir d'un arrangement matriciel de *covariances* ou de corrélations. La *covariance* de deux attributs est une mesure de la force de sa variation conjointe.

Cette analyse permet de trouver une transformation des données qui satisfont les propriétés suivantes:

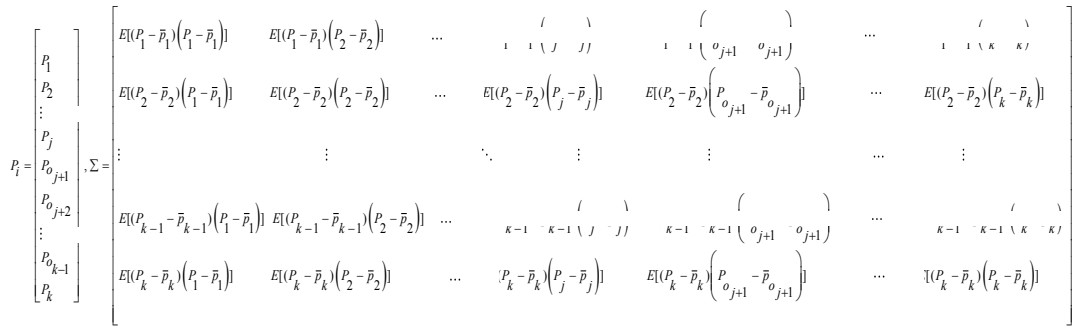
- Chaque paire de nouveaux attributs a une covariance égale à zéro.
- Les attributs sont ordonnés en fonction de la quantité de variance qu'ils capturent.
- Le premier attribut capture la plus grande quantité possible de variance.
- Soumis à la restriction d'*orthogonalité*, chaque attribut suivant capture la plus grande quantité de variance restante.

Dans la prochaine section, nous essayons d'appliquer cette technique statistique à notre ensemble de phrases pour créer des groupes de variables les plus porteuses d'information avec une forte corrélation dans le groupe, mais sans aucun rapport entre les groupes.

3.4.1.2 Analyse en composantes principales (PCA)

Pour l'analyse en composantes principales (Golub et Van Loan, 1996; Pearson, 1901), nous partons d'un espace d'entraînement initial que nous représentons par un arrangement E de valeurs indiquant la variation entre chaque paire d'attributs i (mots) des différentes phrases j qui sont pris des différents documents k. Nous supposons que l'apparition d'un mot i dans une phrase j du document k est aléatoirement et uniformément distribuée.

Nous pouvons représenter les valeurs aléatoires de l'apparition des mots, dans chaque document et aux mots récupérés de l'ontologie, au moyen d'un vecteur P_i, où i indique le mot et k le document où il apparaît, ou l'ordre assigné au mot qui provient de l'ontologie. De même, nous pouvons représenter la variation s_{ik} entre chaque paire d'attributs i, k (le mot i du document k ou de l'ontologie), dans un arrangement Σ, où chaque élément i, j de cet arrangement sont les variations entre les éléments du vecteur P. Nous pouvons alors représenter de manière matricielle le vecteur P et l'arrangement Σ (à partir de l'espace E). L'ensemble d'éléments du vecteur P_i, correspondant aux mots de l'ontologie, sont les mots P_o, avec $o = j+1, j+2, \ldots, \leq k$.



(3.11)

Une matrice obtenue de cette manière est une matrice semi-définie positive, avec ses éléments supérieurs ou égaux à zéro. Nous pouvons aussi l'exprimer comme $\Sigma = E\left(XX^T\right) - \bar{X}\bar{X}^T$ et elle possède les huit propriétés suivantes selon Eaton (1983):

- $\Sigma = E\left(XX^T\right) - \bar{X}\bar{X}^T$
- La matrice A est *hermitique,* car elle est égale à A^T.
- *var (AX + a) = A var (x)* A^T
- *COV (X, Y) = COV (Y, X)*^{T}
- *COV (*X_1*+*X_2*, Y) = COV (*X_1*, Y) + COV (*X_2*, Y)*
- Si $p = q$, alors *var (X + Y) = var (X) + COV (X, Y) + COV (Y, X) + var (Y)*
- *COV (AX, BY) = A COV (X, Y)* B^T
- Si X et Y sont indépendants, *COV (X, Y) = 0*

sachant que X, X_1 et X_2 sont des vecteurs aléatoires de dimension (p×1), que Y est un vecteur aléatoire (q×1), que a est un vecteur (p×1) et que A et B sont des matrices de dimension p×q.

En outre, comme la matrice est symétrique, elle a les propriétés suivantes :

- Ses valeurs propres sont réelles et ses vecteurs propres sont toujours linéairement indépendants.
- Les vecteurs propres associés avec des valeurs propres différents sont orthogonaux ($A^TA = AA^T$).

- Si la matrice ($n \times n$) a n vecteurs propres et forme une matrice T dont les colonnes sont ces vecteurs, alors la transformation $T^{-1}T$ produit une matrice diagonale D, dont les éléments de sa diagonale seront les valeurs propres de la matrice.
- Les valeurs propres $\lambda_1, \lambda_2, \ldots, \lambda_n$ correspondent aux carrés des valeurs singulières.

Or, notre problème consiste à savoir comment ordonner, par groupes, l'ensemble des mots de l'arrangement précédent. Selon la quantité d'information, que contiennent-ils, sont-ils fortement en rapport dans le groupe, sans qu'il n'y ait aucune relation entre les groupes, soit sans redondance entre des groupes? La solution à ce problème permettra d'identifier les *mots* (et les *phrases* correspondantes), qui apportent la plus grande quantité possible d'information, ainsi que celles qui n'en apportent pas de manière importante. Autrement dit, la solution à ce problème nous permettra de discriminer les phrases importantes de celles les moins importantes, ce qui est justement ce dont nous avons besoin pour créer notre espace d'entraînement.

En termes d'algèbre vectorielle, nous avons besoin de construire une nouvelle base vectorielle pour exprimer notre ensemble de mots et de phrases, c'est-à-dire exprimer notre arrangement Σ en un ensemble de vecteurs unitaires et orthogonaux, soit orthonormaux.

En sachant que nous avons l'égalité $Av=\lambda v$ (*théorème spectral*), où A est une matrice carrée $n \times n$, v une matrice carrée de vecteurs propres de A et λ une matrice de valeurs propres de la matrice A, alors nous pouvons appliquer ces mêmes transformations à notre matrice $\sum$ pour trouver l'ensemble de vecteurs propres, orthogonaux. Nous résolvons ainsi une première partie de notre problème qui est de trouver un ensemble de vecteurs qui doivent être orthogonaux. Il reste ensuite à résoudre encore deux problèmes additionnels à savoir que les vecteurs, en plus d'être orthogonaux, doivent être normaux et contenir la variance maximale (la plus grande information). Nous nous trouvons ainsi devant un problème d'optimisation mathématique, spécifiquement de type quadratique avec une fonction quadratique à optimiser soumise à un ensemble de restrictions linéaires.

Considérons alors notre espace initial d'entraînement $\sum$, qui est un ensemble de variables ou d'attributs $P_i = \left[p_1, p_2, \ldots \quad , \quad , p_{o_{j+2}}, p_{k-1}, p_k \right]$, $i = 1,2,\ldots, k$ (notre ensemble de mots)

qui agissent sur un groupe d'objets, soit les phrases prises du corpus de documents. À partir de ces phrases, il s'agit de calculer un nouvel ensemble de variables y_1, y_2,…, y_p, dont les variances diminuent progressivement et n'ont pas de corrélation entre elles. Chaque y_j (j = 1,…., p) sera une combinaison linéaire des x_1, x_2, … , x_p variables originales, sachant que nous avons mentionné que les groupes de variables sont reliés dans le groupe, c'est-à-dire :

$$y_j = a_{j1}x_1 + a_{j2}x_2 + ... + a_{jp}x_p = a_j{}^T x \quad \text{où} \quad a_j{}^T = (a_{1j}, a_{2j}, ..., a_{pj}) \tag{3.12}$$

est un vecteur de constantes et

$$\mathrm{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \tag{3.13}$$

Comme les vecteurs a_{ij} doivent être orthonormaux, alors le module du vecteur doit être égal à 1, c'est-à-dire :

$$a_j^T a_j = \sum_{k=1}^{p} a_{kj}^2 = 1 \tag{3.14}$$

Nous calculons la première y_j, correspondant à y_1, en choisissant le vecteur a_1 qui lui apporte la variance maximale soumise à la contrainte $a_1^T a_1 = 1$. La deuxième y_2 est calculée en obtenant a_2 de sorte qu'elle ne soit pas reliée avec y_1 et ainsi de suite, de sorte que les y_j qui sont obtenues contiennent chaque fois moins de variance.

Or, dans notre matrice initiale $\sum$, nous pouvons observer que les éléments de la diagonale principale correspondent à $\mathrm{E(X\text{-}(X))}^2$, et que dans cette matrice de covariance Σ, ils correspondent à la variance des données, soit la dispersion des mots dans les phrases.

Nous avons précédemment indiqué que nous pouvons réduire une matrice A à une autre matrice diagonale D, en la pré-multipliant par une matrice inverse T^{-1} de vecteurs propres et en la post-multipliant par la même T ($T^{-1}AT = D$). En tenant compte du fait que, par

définition, une matrice orthogonale est telle que $A^{\mathrm{T}}A = AA^{\mathrm{T}}$ et en tenant compte aussi qu'une matrice A est réversible alors $A^{-1}A = AA^{-1}$, ce qui implique que dans ces matrices $A^{\mathrm{T}} = A^{-1}$. Si comme nous l'avons précisé alors, nous voulons choisir a_{j} de sorte que la variance de y_{j} soit maximisée, soumise à $a_j^T a_j = 1$, alors *var* (y_{j}) = *var* ($a_j^T \boldsymbol{x}$) = $a_j^T \sum a_j$ = diag ($\lambda_1, \lambda_2, \ldots, \lambda_{\mathrm{p}}$), où les λi correspondent aux *valeurs propres* de $\sum$ et les vecteurs a_{j} sont ses *vecteurs propres*.

Si nous ordonnons les valeurs propres de la plus grande à la plus petite, nous obtenons les variances ordonnées de la plus grande à la plus petite qui seront à l'origine des vecteurs a_{j} de variances décroissantes. Ainsi, en choisissant le premier vecteur, nous obtenons celui qui nous apporte la variance maximale. Mais notre problème n'est pas seulement ceci, car le vecteur choisi doit être orthogonal au premier. Comme nous l'avons écrit, nous nous trouvons alors devant un problème d'optimisation, avec une fonction à maximiser (le vecteur de variance maximale sujet à des restrictions).

Pour cela, nous pouvons utiliser une méthode basée sur le gradient d'une fonction. Nous avons choisi le théorème de Lagrange. Nous l'appliquons pour l'espace , mais il peut être généralisé à l'espace .

Soit f et g des fonctions différenciables aux points x et y. Si la fonction f a une extrémité (point stationnaire) soumise à la restriction $g(x, y) = c$, au point (x_0, y_0) avec les gradients $\nabla_{x,y} f$ et $\nabla_{x,y} g$ différents de zéro, alors $\nabla_{x,y} f = \lambda \nabla_{x,y} g$, où $\nabla_{x,y} f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$ et $\nabla_{x,y} g = (\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y})$, selon les définitions de Riley et al. (2006). Nous pouvons introduire une fonction auxiliaire $\Lambda(x, y, \lambda) = f(x, y) + \lambda(g(x, y) - c)$ et résoudre l'équation $\nabla_{x,y,\lambda} \Lambda(x, y, \lambda) = 0$. La fonction Λ est appelée le Lagrangien. Ayant obtenu les valeurs $\lambda_1, \lambda_2, \ldots$, où k est le nombre de restrictions, nous pouvons évaluer la fonction objectif f et déterminer leurs valeurs maximales ou minimales.

Dans le cas qui nous occupe et comme nous l'avons mentionné, nous devons maximiser la fonction $y_j = f = a_1^T \sum a_1$, qui est la fonction à maximiser (variance), avec $g = a_1^T a_1 - 1$. La fonction Λ est alors :

$$\Lambda(a_1, \lambda) = a_1^T \Sigma a_1 - \lambda(a_1^T a_1 - 1) \tag{3.15}$$

En remarquant que la fonction $a_j^T \sum a_j$ a une forme quadratique selon la définition de Riley et al. (2006),

$$Q(x) = x^T A x = a_j^T a_j = \sum_{k=1}^{p} a_{kj}^2 = 1 \tag{3.16}$$

Alors,

$$\frac{\partial \Lambda}{\partial a_1} = 2\Sigma a_1 - 2\lambda a_1 \text{ et } \frac{\partial \Lambda}{\partial \lambda} = -a^2 + 1 = 0 \tag{3.17}$$

En faisant $2\Sigma a_1 - 2\lambda a_1 = 2\Sigma a_1 - 2\lambda I a_1 = 0$, alors $(\Sigma - \lambda I) a_1 = 0$. Nous observons que ce résultat est une formulation du *théorème spectral*, où a_1 est le vecteur propre correspondant de λ. Comme cette valeur dans l'équation précédente doit être plus grande que zéro, ceci implique que $(\Sigma - \lambda I)$ doit être égal à zéro. Pour trouver le vecteur a_1, nous calculons d'abord λ, avec le déterminant de $(\Sigma - \lambda I)$ et en le posant égal à zéro, soit $\det(\Sigma - \lambda I) = 0$, selon le théorème de Roche-Frobenius (Rao, 2002). Ayant trouvé cette valeur, nous la remplaçons dans l'équation $(\Sigma - \lambda I) a_1 = 0$, et ainsi obtenons les éléments du vecteur a_1.

La matrice de covariances de Σ est d'ordre *p* et comme elle est semi-définie positive, nous aurons *p* valeurs propres différentes $\lambda_1, \lambda_2, \ldots$, telles que $\lambda_1 > \lambda_2 > \ldots$.

En continuant avec notre développement, si $(\Sigma - \lambda I) a_1 = 0$, alors $\Sigma a_1 - \lambda I a_1 = 0$, ce qui donne $\Sigma a_1 = \lambda I a_1$ alors *var* (y_1) = *var* ($a_1^T x$) = $a_1^T \Sigma a_1 = a_1^T \lambda I a_1 = = \lambda a_1^T a_1 = \lambda \cdot 1 = \lambda$

Donc, pour maximiser la variance y_1, nous devons prendre la plus grande valeur propre, disons λ_1 et son vecteur propre correspondant a_1.

Ainsi, si

$$a_1^T = \left[a_{11}, a_{12}, \cdots \right] \tag{3.18}$$

Alors,

$$y_1 = a_1^T x = a_{11}x_1 + a_{12}x_2 + \cdots \tag{3.19}$$

Ce résultat est une combinaison des variables originales avec la plus grande variance possible.

Nous obtenons la deuxième variable $y_2 = a_2^T x$ au moyen de raisonnements semblables. Il est en outre requis que y_2 ne soit pas co-rapportée avec la variable précédente, c'est-à-dire $cov\ (y_1, y_2) = 0$. Comme $cov(y_1, y_2) = \text{cov}(a_2^T x, a_1^T x)$ et en sachant que $\text{cov}\left(Ax, Bx\right) = A\,\text{cov}\left(x, y\right)B^T$,étant A, B deux matrices, nous avons alors :

$$\text{cov}\ (y_1, y_2) = \text{cov}(a_2^T x, a_1^T x) \tag{3.20}$$

$$\text{cov}(a_2^T x, a_1^T x) = a_2^T\, \text{cov}(x, x) a_1^T \tag{3.21}$$

$$= a_2^T\, \text{cov}(x, x)(a_1^T)^T \tag{3.22}$$

$$= a_2^T\, \text{cov}(x, x) a_1 \tag{3.23}$$

$$= a_2^T \cdot E\left[\left(x - \bar{x}\right)\left(x - \bar{x}\right)^T \right] \cdot a_1 \tag{3.24}$$

$$= a_2^T \Sigma a_1 \tag{3.25}$$

Notre restriction est donc $cov(y_1, y_2) = a_2^T \Sigma a_1 = 0$.

Comme nous avons établi que, $\Sigma a_1 = \lambda a_1$, alors $a_2^T \Sigma a_1 = a_2^T \lambda a_1 = \lambda a_2^T a_1 = 0$

Comme $\lambda \neq 0$, alors $a_2^T a_1 = 0$, ce qui revient à dire que les vecteurs a_2^T y a_1 sont orthogonaux.

En continuant avec notre processus d'optimisation, nous devons maximiser la variance $y_2 = a_2^T \Sigma a_1$, soumise aux restrictions suivantes :

$$a_2^T a_2 = 1 \tag{3.26}$$

$$a_2^T a_1 = 0 \tag{3.27}$$

En appliquant à nouveau le *théorème de Lagrange*, nous obtenons :

$$\Lambda(a_2, \lambda_1, \lambda_2) = a_2^T \Sigma a_2 - \lambda_1 (a_2^T a_2 - 1) - \lambda_2 a_2^T a_1 \tag{3.28}$$

Comme dans le cas précédent, Q est une forme quadratique $Q = \sum_{k=1}^{p} a_k^2 = 1$, alors :

$$\frac{\partial \Lambda}{\partial a_2} = 2\Sigma a_2 - 2\lambda_1 a_2 - \lambda_2 a_1 = 0 \tag{3.29}$$

$$\frac{\partial \Lambda}{\partial \lambda_1} = -a_2^T a_1 + 1 = 0 \tag{3.30}$$

$$\frac{\partial \Lambda}{\partial \lambda_2} = a_2^T a_1 = 0 \tag{3.31}$$

Si nous pré-multiplions la première équation par a_1^T et en sachant que $a_1^T a_2 = 0$ et $a_1^T a_1 = 1$, alors nous obtenons :

$$2a_1^T \Sigma a_2 - \lambda_2 = 0 \tag{3.32}$$

Sachant que si a_2^T et a_1 sont orthogonaux, alors $a_2^T a_1 = a_1^T a_2$ et que $COV(y_2, y_1) = 0$, donc,

$$\lambda_2 = 2a_1^T \Sigma a_2 = 2a_2^T \Sigma a_1 = 0 \tag{3.33}$$

En remplaçant λ_2 dans

$$\frac{\partial \Lambda}{\partial a_2} = 2\Sigma a_2 - 2\lambda_1 a_2 - \lambda_2 a_1 = 0 \tag{3.34}$$

il reste finalement :

$$\frac{\partial \Lambda}{\partial a_2} = 2\Sigma a_2 - 2\lambda_1 a_2 = \left(\Sigma - \lambda I\right) a_2 = 0 \tag{3.35}$$

À nouveau, nous nous trouvons avec une formulation du *théorème spectral*, donc λ_2 est la deuxième valeur propre de la matrice avec son vecteur propre correspondant a_2.

Les raisonnements précédents peuvent être appliqués au composant y_j, auquel correspond le *j-ième* vecteur propre.

Ainsi, l'ensemble des *p* variables *y* peut être exprimé comme le résultat d'une matrice formée par les vecteurs propres, multipliée par le vecteur *x* qui contient les variables originales : $y = Ax$

où :

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix},\ A = \begin{bmatrix} a_{11} & a_{12} & \cdots & \\ a_{21} & a_{22} & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & & a_{pp} \end{bmatrix},\ x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \tag{3.36}$$

Comme nous avons :

$$\begin{aligned}
&\text{var}(y_1) = \lambda_1 \\
&\text{var}(y_2) = \lambda_2 \\
&\text{var}(y_3) = \lambda_3 \\
&\cdots \\
&\text{var}(y_p) = \lambda_p
\end{aligned} \tag{3.37}$$

alors la matrice de covariances **y** est :

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & \\ 0 & \lambda_2 & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & & \lambda_p \end{bmatrix} \tag{3.38}$$

Les valeurs zéro de cette matrice *diagonale* correspondent à la valeur des variables non co-rapportées (covariance 0). On a alors :

$$\Lambda = \text{var}(Y) = A^T \text{ var}(X) A = A^T \Sigma A \tag{3.39}$$

En sachant que A est une matrice orthonormée (parce qu'elle est orthogonale et que pour toutes ses colonnes on a $a_i^T a_i = 1$), alors $\mathrm{A^T A = A A^T = I}$), et donc : $\Sigma = A \Lambda A^T$

Les modèles précédents nous ont paru des modèles efficaces à utiliser comme base pour la description d'un ensemble de phénomènes d'entropie qui caractérisent le traitement du langage naturel. Dans ce cas spécifique de synthèse d'information, nous partons d'un ensemble de documents que nous souhaitons synthétiser en identifiant les unités essentielles d'information à extraire, soit les phrases qui contiennent cette information, et en rejetant les données redondantes, soit les phrases qui contiennent une information non significative.

3.4.2 Classification des éléments du nouvel espace d'entraînement

Rappelons qu'avec notre méthode, pour identifier les valeurs singulières de notre matrice $\sum$, nous exprimons cette matrice comme suit, en sachant que chaque valeur s_i représente la variance de chacune des phrases dans l'ensemble de phrases.

$$A = \left(u_1 u_2 \dots \right. \quad \begin{pmatrix} s_1 & 0 & \dots \\ \vdots & \ddots & \vdots \\ 0 & \dots & \end{pmatrix} \quad \begin{pmatrix} v_1 \\ \\ \vdots \\ v_n \end{pmatrix} \quad \dots \qquad (3.40)$$

Cette variance est associée à la quantité d'information de chacune des phrases, soit son importance. Nous ordonnons les valeurs s_i avec les phrases correspondantes de la plus grande à la plus petite.

Ayant ordonné les phrases, nous initions le processus de construction de l'ensemble d'entraînement en étiquetant les phrases à 70% du total, en commençant par celles ayant les plus grandes variances, avec la valeur +1 pour indiquer qu'elles appartiennent à la classe +1 (phrases importantes). Les 30 % de phrases restantes sont marquées avec la valeur -1 pour indiquer qu'elles appartiennent à la classe -1 (non importantes). Ces proportions seront ajoutées en fonction de la performance des fonctions de classification obtenues.

Rappelons aussi que l'obtention des *composants principaux* de notre matrice nous a permis d'identifier des groupes de variables-mots (*composantes principales*) qui sont fortement reliés dans le groupe, mais sans aucune corrélation entre les groupes. À son tour, le facteur déterminant pour ce groupement correspond aussi la *variance* et comme nous l'avons indiqué cette variance représente la teneur en information ou *importance.* Nous pourrions donc à partir de cette idée choisir les premières phrases. Les phrases représentant la plus haute variance sont marquées comme phrases significatives ou importantes et appartiennent à la classe +1 et les dernières phrases comme non significatives ou non importantes avec la valeur -1.

3.5 Application des algorithmes d'apprentissage

Dans cette dernière phase, il reste à appliquer un algorithme d'apprentissage sur l'ensemble d'entraînement optimisé qui permettra d'abduire la fonction de classification. En d'autres termes, cette phase a pour but d'appliquer, sur l'espace d'entraînement avec ses classes correspondantes, des algorithmes linéaires ou non linéaires pour abduire des fonctions d'apprentissage, qui par la suite seront appliquées à des nouvelles instances pour obtenir leurs classifications correspondantes. À partir de l'espace de phrases significatives (porteuses de l'information maximale) et non significatives (porteuses de peu d'information), on déduira un modèle qui sera ensuite appliqué à une nouvelle phrase pour décider si celle-ci est porteuse d'information significative et si on doit l'inclure ou non dans notre résumé.

Afin de choisir les algorithmes les plus pertinents pour notre modèle de résumé automatique, VENCE, nous avons évalué ceux basés sur les théories de l'apprentissage Bayésien, l'apprentissage statistique et l'exploration de données. Ces algorithmes sont largement utilisés en apprentissage automatique, exploration de données et reconnaissance de formes et avaient donc leur pertinence pour notre recherche. Nous avons mené une expérimentation avec six algorithmes, puis évalué le pouvoir discriminant des fonctions de classification abduites. Cette expérimentation est décrite dans notre troisième article (Chapitre 6) à la section 6.5. Dans les trois prochaines sections, nous détaillons la méthode utilisée pour comparer les résultats obtenus avec ces différents algorithmes puis nous approfondirons les composantes mathématiques des deux algorithmes retenus, le classeur bayésien naïf et la machine à vecteurs de support, ainsi que la façon dont ils sont appliqués pour le résumé automatique extractif. Il est important de noter que tandis que le premier algorithme retenu est nettement probabiliste, le deuxième est basé sur l'optimisation d'un sous-espace vectoriel.

3.5.1 Méthode d'évaluation des algorithmes d'apprentissage

L'ensemble d'entraînement optimisé est maintenant prêt à être utilisé par un algorithme d'apprentissage qui permettra d'abduire une fonction de classification. Cette fonction sera appliquée à de nouvelles phrases (phrases non vues) afin de déterminer si ces phrases sont ou ne sont pas significatives en vue de la construction du résumé. En mesurant le degré

d'efficacité de cette fonction ou autrement dit son pouvoir discriminant, on peut déterminer la performance de l'algorithme d'apprentissage utilisé. Cette section explique donc comment nous avons déterminé quel était les meilleurs algorithmes à utiliser pour notre modèle de résumé automatique.

Nous avons mené une expérimentation afin de répondre à cette question. Cette expérimentation est décrite par la section 6.6.2. Elle porte sur l'évaluation de six algorithmes de classification bien connus et largement utilisés dans les domaines de l'apprentissage automatique, le reconnaissance de formes et l'exploration de données et qui sont la machine à vecteurs de support (SVM) (Cortes et Vapnik, 1995; Vapnik, 1999), la régression logistique (LR) (Agresti, 2007), l'arbres aléatoires (RT) (Breiman, 2001), le classeur bayésien naïf (NB) (Hastie *et al.*, 2009), le perceptron multicouche (MLP) (Rosenblatt, 1957; Rumelhart *et al.*, 1986) et les réseaux neuronaux de type RBF (Radial Basis Function) (RBF-NN) (Buhmann, 2003). Nous avons utilisé quatre métriques. Trois d'entre elles sont issues du domaine de la recherche d'information et sont notées : *Précision* (P), *Rappel* (R), *F_measure* (Korfhage, 1997). La dernière métrique correspond aux courbes *ROC, Receiver Operating Characteristic* en anglais, (Fogarty *et al.*, 2005), de la théorie de la détection des signaux pour des classeurs binaires.

Tout d'abord, nous avons déterminé des catégories des prédictions dans notre classification binaire. En effet, il fallait savoir si la phrase avait été :

- correctement classée en accord avec sa catégorie, soit un vrai positif ou vrai négatif que nous notons TP ou TN respectivement;
- incorrectement assignée à une catégorie, soit un faux positif ou un faux négatif que nous notons FP ou FN respectivement;

L'équation suivante permet de calculer la précision, soit la proportion de phrases correctement classées par rapport au total des phrases à classer. Soit *i* la catégorie de la classe.

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{3.41}$$

Le rappel est calculé en utilisant l'équation suivante, soit la probabilité qu'une phrase soit classée dans la catégorie *i*.

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{3.42}$$

Enfin, la F_measure est calculée avec cette équation :

$$F_\beta = (1+\beta^2)\frac{P_i \times R_i}{(\beta^2 \times P_i) + R_i} \tag{3.43}$$

C'est une mesure moins sensible aux variations que les deux précédentes, la précision et le rappel. Cette mesure est une espèce de moyenne harmonique entre la précision et le rappel qui peut être interprétée comme une moyenne pondérée entre elles. Nous la calculons en faisant β=1 soit :

$$F_1 = (1+1^2)\frac{P_i \times R_i}{(1^2 \times P_i) + R_i} = 2\frac{P_i \times R_i}{R_i + P_i} \tag{3.44}$$

Nous avons également utilisé les courbes *ROC* afin de montrer les variations des phrases classées comme vraies positives par rapport aux fausses positives. En sachant que nous devons produire des résumés de différents documents *i*, nous avons défini les valeurs suivantes :

$$\begin{aligned} tpr_i &= \frac{TP}{TP+FN} \ (taux\ de\ vraies\ \ positives) \\ fpr_i &= \frac{FP}{FP+TN} \ (taux\ de\ \ fausses\ \ positives) \end{aligned} \tag{3.45}$$

Ces métriques nous ont permis d'évaluer le pouvoir discriminant des fonctions de classification abduites et donc de déterminer que le classeur bayésien naïf et la machine à vecteurs de support étaient les meilleurs candidats comme algorithmes d'apprentissage.

3.5.2 Classeur bayésien naïf

En apprentissage automatique, on souhaite souvent déterminer la meilleure hypothèse d'un espace *H*, étant donné un ensemble d'entraînements *D*. La meilleure hypothèse est comprise dans le sens du *plus probable*, étant donné un ensemble de données *D*, avec une connaissance initiale ajoutée sur les probabilités antérieures ou *a priori* d'un ensemble d'hypothèses en *H*. Le *Théorème de Bayes* fournit une méthode directe pour calculer de telles probabilités. En effet, il fournit une méthode pour calculer la probabilité d'une hypothèse, basée sur sa probabilité antérieure, selon la probabilité d'observer plusieurs données vues l'hypothèse (preuves) et les données elles-mêmes (Mitchell, 1997). Ce théorème est la pierre angulaire des méthodes d'apprentissage bayésiens, parce qu'il fournit une manière de calculer les probabilités postérieures ou *a posteriori* P (h | D) à partir des probabilités antérieures *P (h),* avec *P (D)* et *P (D | h).* Le théorème de Bayes peut être alors écrit comme suit :

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{3.46}$$

ou bien défini de manière générale de la façon suivante :

« Soit, $\{A_1, A_2, \ldots$, un ensemble d'événements mutuellement exclusifs et exhaustifs tels que la probabilité de chacun d'eux est différente de zéro. Soit B un événement quelconque tel qu'on connaît ses probabilités conditionnelles $P(B|A_i)$, alors la probabilité $P(A_i|B)$ est donnée par l'expression :

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \tag{3.47}$$

En sachant que la P (B) est la probabilité totale, définie de la façon suivante (Navidi, 2010):

Si $A_1, A_2, \ldots$ sont des événements mutuellement exclusifs et exhaustifs et *B* est un certain événement, donc :

$$P(B) = P\left(A_1 \cap B\right) + \cdots \qquad \cap B\ ) \tag{3.48}$$

Si P (A | B) $= \dfrac{P\left(A \cap B\right)}{P(B)}$, alors $P\left(A \cap B\right)$ = P (A | B) P (B) (3.49)

De même, si P (A | B) $= \dfrac{P\left(A \cap B\right)}{P(B)}$, alors $P\left(A \cap B\right)$= P(A | B) P(B) (3.50)

Ainsi, si P (A_i) ≠ *0* pour chaque A_i, alors :

$$P(B) = P(B\left|A_1\right)P(A_1) + \cdots \qquad A_n)P(A_n) \tag{3.51}$$

En remplaçant *P*(*B*) dans la formule de Bayes (eq. 3.7), nous obtenons alors :

$$P(A_i\left|B\right) = \frac{P(B\left|A_i\right)P(A_i)}{\sum_{j=1}^{n} P(B\left|A_j\right)P(A_j)} \tag{3.52}$$

L'obtention des probabilités $P(A_i)$, ainsi que la *Probabilité Totale,* ne représentent pas un plus grand problème alors que l'estimation de la probabilité $P(B\left|A_i\right)$ s'avère plus problématique. Pour l'obtenir, nous devons prendre en considération la notion d'*indépendance conditionnelle*, qui est définie par Steinbach et al. (2006) de la manière suivante :

Soient **X**, **Y**, **Z**, trois ensembles de variables aléatoires. Les variables en **X** sont dites conditionnellement indépendantes de **Y**, vu **Z,** si la condition suivante est remplie:

$$P(X\left|Y,Z\right) = P(X\left|Z\right) \tag{3.53}$$

Autrement dit, le fait de connaître **Z** permet de dire que **X** et **Y** sont indépendants.

Or, soit :

$$P\left(X,Y\left|Z\right.\right) = \frac{P(X,Y,Z)}{P(Z)} \tag{3.54}$$

en multipliant le numérateur et le dénominateur par *P (Y, Z),* alors :

$$P\left(X,Y\middle|Z\right)=\frac{P(X,Y,Z)}{P(Z)}\times\frac{P(Y,Z)}{P(Y,Z)} \tag{3.55}$$

$$P\left(X,Y\middle|Z\right)=\frac{P(X,Y,Z)}{P(Y,Z)}\times\frac{P(Y,Z)}{P(Z)} \tag{3.56}$$

$$P\left(X,Y\middle|Z\right)=P\left(X\middle|Y,Z\right)\times P(Y|Z) \tag{3.57}$$

$$P\left(X,Y\middle|Z\right)=P\left(X\middle|Y\right)\times P(Y|Z) \tag{3.58}$$

Maintenant, soient $X_1,X_2,\ldots$ des attributs de l'espace d'entraînement, qui sont conditionnellement indépendants l'un de l'autre, et étant donné l'ensemble Y. Considérons par convenance que $X=\{X_1,X_2\}$, alors :

$$\begin{aligned}&P\left(X\middle|Y\right)=P\left(X_1,X_2\middle|Y\right)\\&P\left(X\middle|Y\right)=P\left(X_1\middle|X_2,Y\right)P\left(X_2\middle|Y\right)\\&P\left(X\middle|Y\right)=P\left(X_1\middle|Y\right)P\left(X_2\middle|Y\right)\end{aligned} \tag{3.59}$$

De manière générale, quand **X** contient *n* attributs qui sont conditionnellement indépendants l'un de l'autre étant donné **Y**, nous avons:

$$P\left(X_1,X_2,\ldots\right)=\prod_{i=1}^{n}P\left(X_i\middle|Y\right) \tag{3.60}$$

En retournant au théorème de Bayes,

$$P(A_i|B)=\frac{P(B|A_i)P(A_i)}{\sum_{j=1}^{n}P(B|A_j)P(A_j)} \tag{3.61}$$

et en tenant compte du résultat précédent, alors :

$$P(A_i \mid B) = \frac{\prod_{j=1}^{d} P(B_j \mid A_i) P(A_i)}{\sum_{j=1}^{n} P(B \mid A_j) P(A_j)} \tag{3.62}$$

où chaque ensemble d'attributs est égal à : $B = \{B_1, B_2, \ldots$.

Avant de présenter comment la formule précédente nous a permis de classer de nouvelles instances (phrases) de documents comme significatives ou non significatives, il est très important de souligner que la présomption d'indépendance conditionnelle nous a permis de réduire notre espace d'hypothèse et par conséquent notre complexité de calcul temporaire d'une manière radicale, en passant d'une complexité exponentielle d'ordre $O(2^n)$ à une complexité polynomiale d'ordre O(n). En sachant que notre espace de phrases **X** est composé de *n* valeurs booléennes (chaque phrase est importante ou ne l'est pas) et **Y** est notre vecteur de classes qui contient seulement deux types de classes (significative et non significative), nous avions la nécessité de calculer un ensemble de paramètres $\beta_{ij} = \mathrm{P}\left(\mathrm{X} = \mathrm{x_i} \mid \mathrm{Y} = \mathrm{y_i}\right)$. Ainsi, l'indice *i* prend 2^n valeurs (un pour chaque valeur de X) et *j* peut prendre deux valeurs, de telle sorte que nous avions besoin de 2^{n+1} paramètres. Comme pour une certaine valeur *j*, la somme des i en β_{ij} doit être égal à 1, alors, pour une certaine valeur y_j, nous avions besoin de calculer 2^{n-1} paramètres. En tenant compte des deux valeurs possibles de Y, le total de paramètres à calculer est alors $2\times(2^{n-1})$, en se transformant ainsi en un problème *intraitable*. Si nous introduisons la présomption d'indépendance conditionnelle et comme X et Y sont des vecteurs booléens, nous avions alors besoin seulement de $2\times n$ paramètres pour définir $P\left(X_i = x_{ik} \mid Y = y_j\right)$.

Le théorème de Bayes modifié avec l'introduction de la notion d'indépendance conditionnelle nous a permis de classer de nouvelles instances de la manière suivante :

Vu une nouvelle phrase X, il est important de déterminer la valeur la plus probable de Y (en maximisant sa probabilité), alors :

$$P(B) = P\left(A_1 \cap B\right) + \cdots \qquad \cdots \tag{3.63}$$

$$Y \leftarrow \arg\max_{y_k} = \frac{\prod_{j=1}^{d} P(B_j \mid A_i = A_k) P(A_i = A_k)}{\sum_{j=1}^{n} P(B \mid A_j) P(A_j)} \tag{3.64}$$

où k=0,1 (significative et non significative)

Comme le dénominateur est constant dans tous les cas, nous pouvons le supprimer et nous obtenons:

$$Y \leftarrow \arg\max_{y_k} = \prod_{j=1}^{d} P(B_j \mid A_i = A_k) P(A_i = A_k) \tag{3.65}$$

Ainsi, selon la valeur obtenue, la phrase sera ou non sélectionnée.

Dans cette section, nous venons de présenter comment nous avons fait pour entraîner une fonction basée sur les probabilités antérieures ou a priori, en utilisant la méthode Bayes naïve.

Dans la prochaine section, nous présentons comme utiliser la programmation mathématique pour entraîner une fonction qui est basée sur des vecteurs de support dans l'espace n-dimensionnel.

3.5.3 Machine à vecteurs de support

En apprentissage automatique en général, l'espace d'entraînement peut être initialement vu comme une boîte noire (figure 3.5), contenant une fonction inconnue f qui reçoit un ensemble de valeurs associées à un ensemble de variables ou d'attributs *X* et qui produit en sortie un ensemble de valeurs associées à des classes Y.

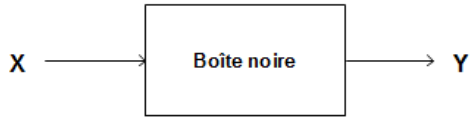


Figure 3.5 *Le processus général d'apprentissage automatique*

En supposant que cette fonction est d'un type déterminé, la tâche consiste à découvrir ou à induire les paramètres qui régissent la fonction correspondante, pour apprendre le modèle respectif à partir d'un ensemble de données qui sont associées à des instances ou à un ensemble d'entraînement différencié ou classé (figure 3.6).

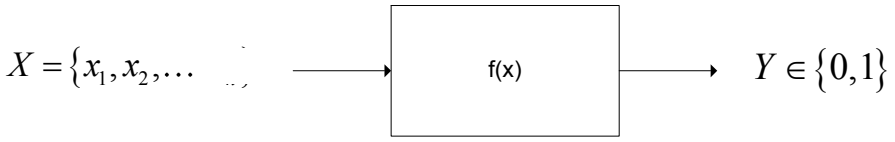


Figure 3.6 *L'induction/abduction de la fonction*

Le problème ainsi posé permet de transformer la tâche d'apprendre en un problème de classification. Si la fonction induite est linéaire, alors ce problème en est un de classification linéaire et le modèle trouvé est un classeur linéaire. Si l'ensemble qui forment les classes est un ensemble binaire, qui correspond à un ensemble d'entraînement sur lequel nous avons fait une *mise en correspondance avec* un plan ou un hyperespace, nous pouvons essayer de séparer ces classes en deux groupes au moyen d'une ligne droite, d'un plan ou d'un hyperplan, selon le nombre de variables ou d'attributs qui composent chacun des éléments de cet ensemble ou d'instances d'entraînement (figure 3.7).

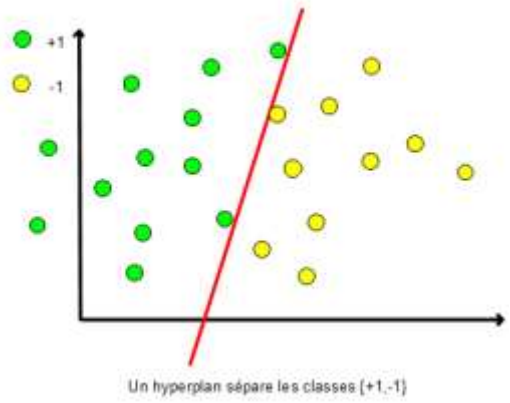


Figure 3.7 *Séparation des classes par un hyperplan*

Le nombre d'hyperplans que nous pouvons tracer est infini. Nous pouvons visualiser ces situations dans la figure 3.8.

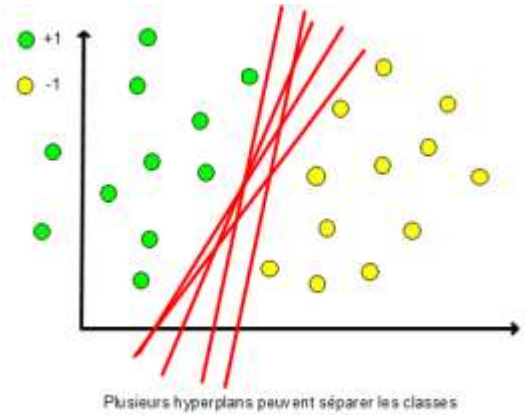


Figure 3.8 *Des infinis plans peuvent séparer les classes*

De même, il peut y avoir des mesures infinies de la distance entre les deux groupes. Nous appelons cette distance *marge* du classeur. Nous pouvons aussi imaginer que nous traçons deux hyperplans parallèles et équidistants du premier hyperplan, qui délimitent les limites de cette marge, comme nous pouvons le voir dans la figure 3.9.

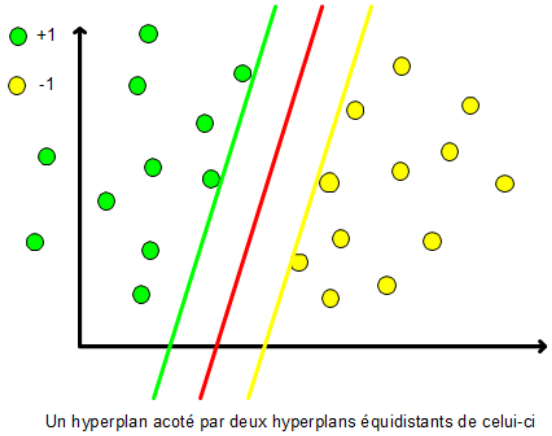


Figure 3.9 *Deux plans symétriquement séparés d'un plan*

Appelons l'hyperplan central *limite de la décision* et appelons la marge, *marge de la limite de décision*. Nous pouvons intuitivement penser que si la marge est petite, de légères perturbations dans la limite peuvent significativement affecter la classification. Les classeurs qui produisent des limites de décision avec des petites marges sont plus

susceptibles de produire un *« sur-ajustement »* et tendent à généraliser pauvrement les nouveaux exemples. Le sur-ajustement se produit quand le modèle n'a pas appris correctement les données d'entraînement. Ainsi, la capacité d'un modèle linéaire est inversement en rapport avec sa *marge.* Des modèles avec des petites marges ont une haute capacité de classification car ils sont plus flexibles et peuvent recevoir beaucoup d'exemples d'entraînement contrairement aux classeurs avec de plus amples marges. Toutefois, en accord avec le principe *d'apprentissage statistique de* minimisation du risque structurel, lorsque la capacité est augmentée, les erreurs de généralisation le sont aussi. Il est ensuite nécessaire, dans les classeurs linéaires, d'*optimiser* la marge de ces limites de décision pour assurer que les erreurs de généralisation soient minimales (Steinbach *et al.*, 2006). Dans la figure 3.10, nous pouvons visualiser cette idée.

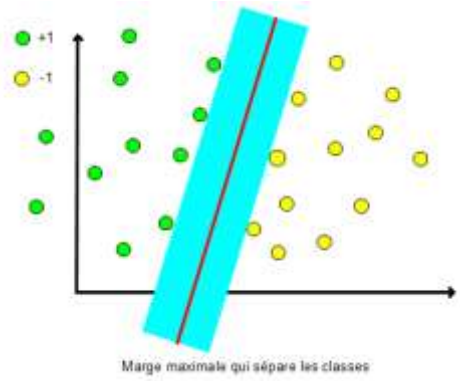


Figure 3.10 *Plan de marge maximale*

En général, dans les classeurs linéaires, nous pouvons distinguer deux types selon qu'ils peuvent totalement ou non séparer les classes : cas *séparable* ou *sans bruit* et cas *non séparable* ou *avec bruit* (Hastie et al., 2009). Dans notre recherche, nous avons assumé que notre espace de phrases n'est pas totalement séparable, étant donné la difficulté réal à bien distinguer les phrases importantes de celles qui ne le sont pas.

Ensuite, nous formalisons les concepts exposés précédemment en nous basant principalement sur les travaux de Hastie et al. (2009), McQueen (1967), Mitchell (1997), Riley et al. (2006) et Steinbach et al. (2006). Supposons que nous partons d'un espace d'entraînement qui est composé de *n* exemples et que notre fonction *f* de la boîte noire, qui

régit le comportement de ces données, est égale à $\boldsymbol{w}\bullet\boldsymbol{x}+b$, où $\boldsymbol{w}$ et b sont les paramètres du modèle et $\boldsymbol{x}$ est l'ensemble d'attributs. Chaque exemple est représenté par le tuple ($\mathbf{x}_i$, $\mathbf{y}_i$), avec $i = 1,\ldots, n$, où $x_i = (x_{i1}, x_{i2}, \ldots$ et correspond à l'ensemble d'attributs pour le i^e exemple et $y_i \in \{-1,1\}$. Ainsi, la limite de la décision (la fonction f) de ce classeur linéaire peut être écrite par $\boldsymbol{w}\bullet\boldsymbol{x} + b = 0$.

Tout exemple situé sur cette limite doit satisfaire l'équation précédente. Ainsi, par exemple, si x_a et x_b sont deux points situés sur cette limite, alors :

$$\begin{aligned} w\cdot x_a + b &= 0 \\ w\cdot x_b + b &= 0 \end{aligned} \tag{3.66}$$

En soustrayant les deux équations, nous obtenons $w\cdot(x_b - x_a) = 0$, où évidemment $x_b - x_a$ est un vecteur parallèle à la limite de la décision. Comme le *produit scalaire* est zéro, alors $\mathbf{w}$ est perpendiculaire à $x_b - x_a$ et par conséquent, perpendiculaire à la limite de la décision. Il peut être montré qu'une certaine valeur x_u au-dessus de cette limite satisfait l'équation $w\cdot x_u + b = c$ (c > 0) et de manière égale une valeur x_l située sous cette limite satisfait l'équation $w\cdot x_l + b = c'$ (c < 0). Si nous marquons toutes les valeurs au-dessus de cette fonction comme de la classe +1 et tous ceux qui sont trouvés au-dessous comme de la classe -1, nous pouvons donc prédire la classe (+1 ou -1) pour un nouvel exemple z de la manière suivante :

$$y = \begin{cases} 1 \ \ si \ \ w\cdot z + b > 0 \\ -1 \ \ si \ w\cdot z + b < 0 \end{cases} \tag{3.67}$$

Considérons maintenant les deux valeurs qui sont trouvées au-dessus et au-dessous des plus proches valeurs de la limite de la décision. Ces valeurs doivent satisfaire les équations $w\cdot x_u + b = c$ et $w\cdot x_l + b = c'$ pour une valeur positive et une valeur négative respectivement. Si nous calculons à l'échelle les paramètres $\mathbf{w}$ et b, nous pouvons exprimer les hyperplans h_1 et h_2 de la manière suivante :

$$\begin{aligned} h_1 &= w \cdot x + b = \ \ 1 \\ h_2 &= w \cdot x + b = -1 \end{aligned} \tag{3.68}$$

Or, la marge de la limite de la décision est donnée par la distance d entre ces deux hyperplans. Soient x_1 et x_2 deux points situés sur les hyperplans h_1 et h_2 respectivement. En remplaçant ces points dans les équations précédentes et en soustrayant la seconde équation de la première, alors :

$$\begin{aligned} h_1(x_1) &= w \cdot x_1 + b = \ \ 1 \\ h_2(x_2) &= w \cdot x_2 + b = -1 \\ h_1(x_1) - h_2(x_2) &= w(x_1 - x_2) = 1 - (-1) = 2 \end{aligned} \tag{3.69}$$

En sachant que la norme de w est $\|w\|$ et que la distance d est justement x_1 - x_2, nous pouvons trouver une telle distance en remplaçant ces valeurs dans la dernière équation :

$$\|w\| \times d = 2, \Rightarrow d = \frac{2}{\|w\|} \tag{3.70}$$

Jusqu'à ce point comme nous pouvons le voir, nous avons déterminé la manière d'obtenir les hyperplans h_1 et h_2, qui délimitent la marge de la limite de la décision, ainsi que la largeur d de cette marge (la *distance* entre des hyperplans). Comme nous l'avions mentionné, non seulement il est nécessaire de trouver la manière de séparer les deux groupes qui représentent les classes, mais il est aussi nécessaire de trouver la largeur optimale de cette séparation. Cela signifie qu'il faut maximiser la marge de la limite de la décision. Donc les paramètres de la fonction f à induire, $\boldsymbol{w}$ et b doivent être choisis de la manière suivante :

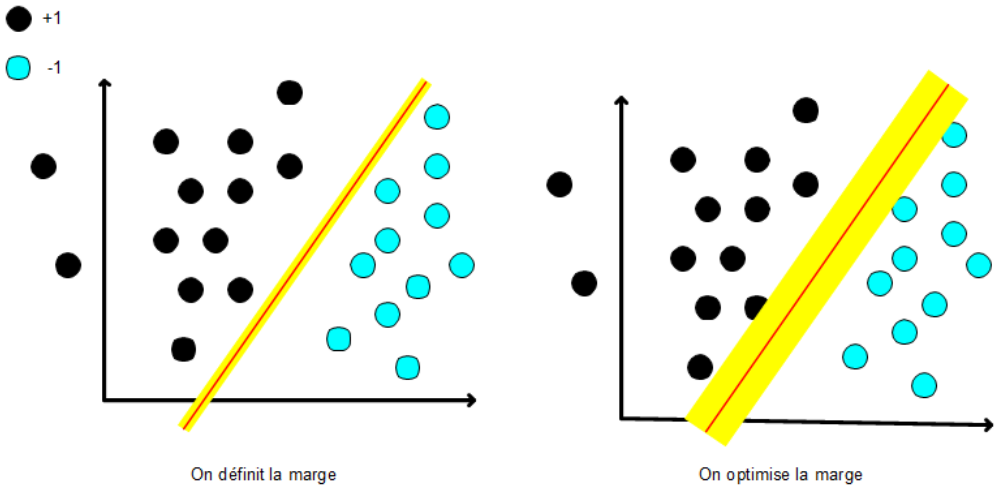
$$\begin{aligned} w \cdot x_i + b &\geq 1 \quad si \ \ y_i = 1 \\ w \cdot x_i + b &\leq -1 \quad si \ \ y_i = -1 \end{aligned} \tag{3.71}$$

Nous pouvons ainsi écrire ces équations d'une manière plus compacte comme suit:

$$y_i = (w \cdot x_i + b) \geq 1, \ \ i = 1, \ldots \tag{3.72}$$

Aux deux conditions précédentes, nous devrions ajouter une contrainte additionnelle, soit que la marge de la limite de décision doit être maximale. Nous aurions alors un problème qui consiste à optimiser une fonction $f(w) = \frac{2}{\|w\|}$, soumise à deux restrictions linéaires. Nous nous trouvons ainsi devant un nouveau problème de programmation quadratique, que nous pouvons résoudre avec les techniques et les méthodes correspondantes.

Dans les deux figures qui suivent (figures 3.11a et 3.11b), nous pouvons visualiser l'ensemble des classes divisé par la limite de la décision $\boldsymbol{w}\bullet\boldsymbol{x} + b = 0$ avec une petite marge, ainsi que cette même limite avec sa marge maximisée. Nous observons que les limites de cette marge touchent un ensemble de points qui sont appelés *vecteurs de support*.



Figures 3.11a et 3.11b *Optimisation de la marge qui sépare les classes*

Dans notre recherche, nous avons induit un modèle qui apprend à séparer les phrases d'un ensemble de documents en deux classes : significative (importante) et non significative (non importante), selon la teneur en information de ces phrases. Nous avons ainsi une fonction qui identifie les phrases qui se trouvent au-dessus ou au-dessous d'elle, mais avec la séparation maximale (optimale), en ayant une marge moins susceptible au problème mentionné de sur ajustement. Nous avons supposé aussi que notre fonction fera éventuellement des erreurs de classification, en introduisant du bruit dans la classification. En d'autres termes, des phrases qui doivent être classées comme significatives sont classées

comme non significatives et vice versa. Nous pouvons formellement établir que les vecteurs, représentant quelques phrases *i,* se trouvent dans une région qui ne leur corresponde pas et par conséquent ils ne pourront pas être séparés au moyen d'hyperplans comme nous l'avons mentionné. Nous pouvons visualiser la situation décrite dans la figure 3.12.

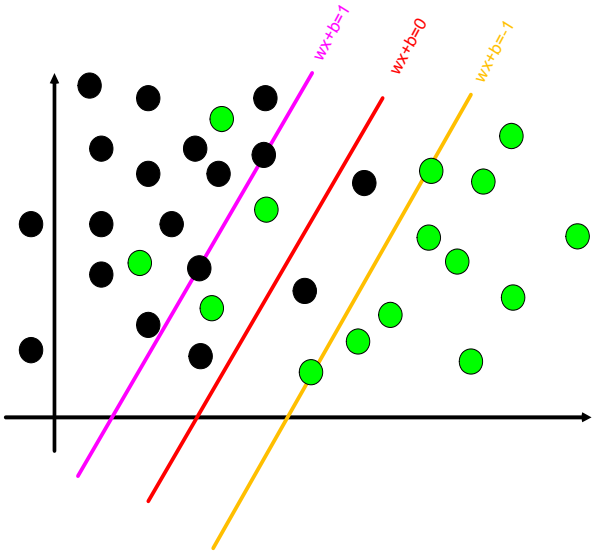


Figure 3.12 *Des erreurs dans la marge*

Pour remédier à cette situation, nous devons introduire une valeur $\xi > 0$ qui nous permet d'ajuster ces cas non séparables aux restrictions de notre modèle, ce qui donne :

$$\begin{aligned} w \cdot x_i + b &\geq +1 - \xi_i \\ w \cdot x_i + b &\leq -1 + \xi_i \end{aligned} \tag{3.73}$$

Nous pouvons visualiser cette situation dans la figure 3.13.

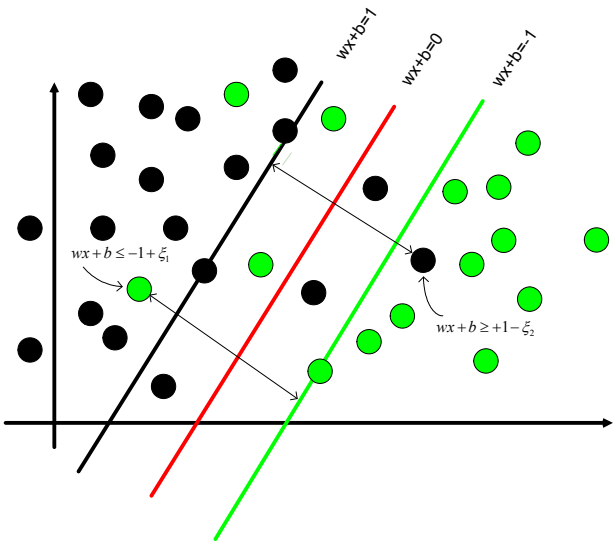


Figure 3.13 *Estimation des erreurs*

Notre modèle à optimiser peut alors être présenté de la manière suivante :

$$\text{Maximiser } f(w) = \frac{2}{\|w\|} \tag{3.74}$$

$$\text{Soumise à: } \begin{matrix} w \cdot x_i + b \ge +1 - \xi_i \\ w \cdot x_i + b \le -1 + \xi_i \end{matrix} \text{ et } \xi_i > 0 \tag{3.75}$$

Notre fonction peut être écrite par convenance comme $f(w) = \frac{2}{\|w\|^2}$, nous évitant ainsi de manipuler une racine carrée dans le calcul de la norme $\|w\|$. En sachant qu'en programmation mathématique, maximiser la fonction Z est égal à minimiser $\frac{1}{Z}$, posons alors :

$$\max f(w) = \min \frac{1}{f(w)} \tag{3.76}$$

Exprimons aussi notre système,

$$\begin{aligned} w \cdot x_i + b &\geq +1 - \xi_i \\ w \cdot x_i + b &\leq -1 + \xi_i \end{aligned} \tag{3.77}$$

d'une manière plus compacte, soit :

$$y_i(x_i \cdot w + b) \geq 1 + \xi_i \tag{3.78}$$

Or, si dans le vecteur x_i on commet une erreur, alors $\xi_i > 1$ et alors, $\sum_i \xi_i$ est un niveau supérieur du nombre d'erreurs qui sont commises dans l'ensemble d'entraînement.

Nous pouvons aussi inclure une valeur C dans la *fonction-objectif* en pénalisant les erreurs, ce qui pourrait être interprété comme un coût additionnel à cette fonction. Ce paramètre peut être choisi en se basant sur la performance du modèle sur *l'ensemble de validation mis en évidence par* Steinbach et al. (2006).

De manière additionnelle, nous pouvons avoir un exposant *k* pour la somme des erreurs $\sum_i \xi_i$, que nous pouvons interpréter comme une manière de pondérer les erreurs en fonction de leur quantité ainsi que déterminer la convexité de la fonction, aspect très important pour garantir qu'un *minimum local* est aussi *global*. Si nous définissons par exemple cette valeur comme 1 ou 2, nous avons une convexité quadratique. Par addition, si nous la définissons comme égale à 1, nous garantissons qu'aucune valeur ξ_i, ni aucun de ses correspondants multiplicateurs de Lagrange, apparaissent dans le *problème dual*, comme nous verrons plus loin.

Alors, la fonction à optimiser *f (w)* peut initialement être exprimée par :

$$f(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i^k \tag{3.79}$$

Comment pouvons-nous interpréter cette *fonction-objectif* ? Si nous considérons que C a une grande valeur, nous assignons un poids élevé aux erreurs face à $\|w\|^2$ et, au contraire, si C est petite, nous assignons un poids plus élevé à $\|w\|^2$. D'autre part, si *k* est grand, ce que

nous faisons revient à donner beaucoup plus de poids aux erreurs. Dans notre recherche, nous avons assumé une valeur pour *k* égale à 1.

Nous pouvons poser l'ensemble de restrictions comme :

$$\left.\begin{array}{l} w\cdot x_i + b \geq +1-\xi_i \\ w\cdot x_i + b \leq -1+\xi_i \end{array}\right\}, y_i \in \{+1,-1\} \qquad \xi_i \geq 0 \;\; \forall_i = 1,\ldots \tag{3.80}$$

Ainsi, notre modèle à optimiser est :

$$\min_{w\in} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \tag{3.81}$$

$$s.à: \quad \begin{array}{l} y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0, \; \forall_i \\ \xi_i \geq 0, \; \forall_i \end{array} \tag{3.82}$$

où *d* est le nombre de composants du vecteur.

Comme nous l'avons précédemment mentionné, pour résoudre ce type de problèmes de programmation quadratique nous avons utilisé le théorème de Lagrange. La nouvelle fonction-objectif ou *Lagrangien*ne est :

$$\Lambda_p(w,b,\xi_i,\lambda_i,\mu_i) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\lambda_i\{y_i(x_i\cdot w+b)-1+\xi_i\} - \sum_{i=1}^{n}\mu_i\xi_i \tag{3.83}$$

avec λ_i et μ_i étant les *multiplicateurs de Lagrange.*

Afin de réduire les inéquations à des égalités, nous avons introduit les conditions complémentaires de Karush - Kuhn - Tucker (KKT) (Kuhn et Tucker, 1951):

$$\begin{aligned}
&\xi_i \geq 0 \\
&\lambda_i \geq 0 \\
&\mu_i \geq 0 \\
&\lambda_i \left\{ y_i (x_i \cdot w + b) - 1 + \xi_i \right\} = 0 \\
&\mu_i \xi_i = 0
\end{aligned} \tag{3.84}$$

En continuant avec l'application du théorème de Lagrange, nous obtenons la première dérivée de Λ par rapport à w, b, ξ_i et que nous égalons à zéro :

$$\frac{\partial L}{\partial w_j} = w_j - \sum_{i=1}^{n} \lambda_i y_i x_{ij} = 0 \Rightarrow w_j = \sum_{i=1}^{n} \lambda_i y_i x_{ij} \tag{3.85}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \lambda_i y_i = 0 \Rightarrow \sum_{i=1}^{n} \lambda_i y_i = 0 \tag{3.86}$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \mu_i = 0 \Rightarrow \lambda_i + \mu_i = 0 \tag{3.87}$$

En sachant que pour le *Théorème de la dualité*, toute solution *primale* a une solution *duale*, nous avons utilisé cette propriété pour exprimer notre problème en fonction seulement des multiplicateurs de Lagrange, ce qui nous a permis de réduire la complexité temporelle-spatiale. Ainsi, nous avons pu remplacer les équations (3.85), (3.86), et (3.87) précédentes dans la fonction Λ_p, pour obtenir :

$$\begin{aligned}
\Lambda_d &= \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i \cdot x_j + C \sum_i \xi_i - \sum_i \lambda_i \left\{ y_i \left(\sum_j \lambda_j y_j x_i \cdot x_j + b \right) - 1 + \xi_i \right\} - \sum_i (C - \lambda_i) \xi_i \\
&= \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i \cdot x_j
\end{aligned} \tag{3.88}$$

C'est une solution duale à notre problème initial primal Λ_p. Il est important de remarquer, qu'en accord avec l'équation (3.87) précédente, puisque les multiplicateurs λ_i y μ_i sont non négatifs alors $0 \leq \mu_i, \lambda_i \leq C$.

Ayant obtenu les valeurs des multiplicateurs de Lagrange en utilisant une certaine méthode numérique de programmation quadratique et en tenant compte de la restriction précédente, nous avons pu remplacer cette restriction dans l'équation (3.85) précédente pour trouver les valeurs de w_j.

Nous pouvons utiliser le résultat précédent pour choisir les *phrases significatives* qui formeront une partie du résumé ainsi :

Soit *z* une phrase qui doit être classée, alors,

$$f(z) = sign\left(\boldsymbol{W} \cdot \boldsymbol{Z} + b\right) = sign\left(\sum_{i=1}^{n} \lambda_i y_i \boldsymbol{X}_i \cdot \boldsymbol{Z}\right) \tag{3.89}$$

Si la valeur de la fonction précédente est égale à 1, la phrase sera classée comme phrase significative. Dans le cas contraire, ce sera une phrase non significative. Les valeurs de *W* sont celles obtenues par le modèle comme nous l'avons déjà montré. Les valeurs de *Z* sont les valeurs *tf* - *idf* calculées pour cette phrase. La valeur de la constante *b* est la constante de notre modèle linéaire, que nous avons obtenue en remplaçant les valeurs de w_j dans l'équation $\lambda_i \left\{ y_i(\boldsymbol{X}_i \cdot \boldsymbol{W} + b) - 1 + \xi_i \right\} = 0$. Nous pouvons aussi prouver notre modèle en utilisant une instance Z qui a déjà été classée.

Dans les sections précédentes, nous venons de présenter les aspects mathématiques des méthodes que nous avons utilisées pour obtenir des espaces d'entraînement et ses fonctions correspondantes de classification des modèles à comparer. Nous avons aussi montré comme appliquer ces méthodes basées sur des probabilités a priori et a posteriori et sur des techniques d'optimisation mathématique, pour bien discriminer les phrases significatives ou « importantes » de celles qui ne le sont pas.

3.6 Expérimentation du modèle proposé

Nous avons mené plusieurs expérimentations pour proposer un modèle de résumé automatique extractif performant. Chacune de ces expérimentations nous a permis de montrer des résultats intermédiaires, d'affiner notre modèle et d'avancer vers un modèle

global. Une dernière expérimentation sur un corpus a encore permis de montrer l'efficacité du modèle en utilisant des métriques utilisées pour mesurer la qualité des résumés obtenus. Cette expérimentation est décrite en détail dans la section 6.5 (article 3). Nous en reprenons ici les éléments principaux ainsi que les résultats obtenus.

3.6.1 Description de l'expérimentation

Nous avons utilisé les métriques de ROUGE (Recall Oriented-Understudy of Gisting Evaluation) pour mesurer la qualité des résumés automatiques proposé par Lin (2004). Le logiciel ROUGE est en effet accessible à la communauté universitaire pour comparer les résumés générés par un programme avec des résumés préparés par des experts humains. Pour effectuer cette comparaison, le logiciel ROUGE utilise diverses méthodes telles que ROUGE-N (n-grammes), ROUGE-L (la sous-séquence commune la plus longue – LCS), ROUGE-SU (un couple de mots en ordre dans la phrase, plus 1-gramme) et ROUGE-W (basé sur les sous-séquences communes les plus longues pondérées). Pour chacune de ces méthodes, on calcule la précision, le rappel et la F_measure. Ce logiciel est considéré comme la référence pour évaluer la qualité de résumés produits automatiquement.

Nous avons appliqué notre modèle sur un corpus de documents de DUC (NIST, 2006). Ce corpus, est un ensemble de documents sur des différents sujets qui comportent les résumés correspondants faits par des spécialistes et qui sont disponibles pour les chercheurs en résumé automatique. Nous avons ensuite évalué la performance des fonctions de classification ainsi que la qualité des résumés obtenus.

3.6.2 Résultats obtenus

Notre projet de recherche a débuté par l'optimisation d'un espace de recherche renforcé par la connaissance ontologique (Motta *et al.*, 2011). Ensuite, il a fallu considérer la définition de procédures pour l'identification de phrases les plus porteuses d'information ainsi que celles les moins porteuses d'information, pour la création d'un ensemble d'entraînement à partir duquel on a abduit des fonctions d'apprentissage (Motta *et al.*, 2012). Ces fonctions ont été soumises à un processus d'évaluation pour déterminer celles qui ont la meilleure capacité de discriminer les phrases les plus importantes (significatives) de celles qui ne le sont pas. L'étape suivante a consisté à appliquer les fonctions obtenues à un corpus de

documents pour extraire les phrases les plus importantes, puis les ordonner et produire le résumé correspondant. Pour compléter, la qualité du résumé obtenu a été évalué au moyen de la suite logicielle spécialisée ROUGE (Lin, 2004).

Les résultats obtenus montrent que les objectifs proposés ont été largement atteints. En effet, les performances des fonctions abduites, dont le calcul se base sur les métriques de précision, de rappel et de F_measure, donnent des valeurs de 100 % ou proches de celle-ci. Les valeurs trouvées pour évaluer la qualité de la classification, au moyen des courbes ROC en calculant l'aire sous la courbe (AUC), sont aussi égales ou proches de 100 %. En ce qui concerne l'évaluation des résumés obtenus proprement dite, les résultats obtenus sont bien meilleurs que ceux obtenus avec certains logiciels disponibles sur le marché, ou encore bien meilleurs que les résultats obtenus par des travaux de recherche similaires. Le détail de ces résultats est présenté au complet dans le chapitre 6.

Dans le présent chapitre, nous venons de présenter la manière dont nous avons conçu un nouveau modèle d'apprentissage automatique pour la production de résumés extractifs. Nous avons commencé par représenter un corpus de documents au moyen d'états d'un espace vectoriel pour y extraire un ensemble d'entraînement performant à partir de l'ajout de la connaissance ontologique et l'optimisation de cet espace.

Nous avons présenté aussi comment nous avons fait pour abduire les paramètres de fonctions de classification prises des domaines de la reconnaissance de formes et l'exploration de données qui apprennent de cet espace d'entraînement pour configurer des fonctions d'apprentissage qui seront utilisées pour distinguer les phrases porteuses et non porteuses d'information d'après des nouveaux exemples. C'est ainsi que cette classification, nous servira pour la construction d'un algorithme d'ordonnancement qui identifiera les phrases qui feront partie d'un résumé.

Étant donné qu'il est nécessaire d'évaluer la performance des fonctions obtenues, nous avons présenté les différentes méthodes pour mesurer cette performance: des matrices de confusion, des métriques telles comme la précision, le rappel et la mesure F et des courbes ROC.

Nous avons montré aussi dans le présent chapitre comment évaluer la qualité des résumés obtenus d'après l'application des meilleures fonctions à des ensembles de phrases non vues, au moyen du logiciel ROUGE et ses méthodes ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-SU et ROUGE-L.

Les prochains chapitres 4, 5 et 6 sont composés respectivement de nos trois articles. Le premier présente les différentes méthodes et expérimentations menés pour l'optimisation de l'espace de recherche pour l'extraction d'ensembles d'entraînement. Le deuxième décrit comment obtenir des fonctions de classification plus efficaces dans le contexte du résumé automatique par extraction. Enfin, le troisième article présente le modèle VENCE et l'évaluation de sa performance tant pour l'abduction des fonctions d'apprentissage que pour les résumés obtenus.

CHAPITRE 4. Évaluation de l'efficacité de techniques linéaires pour optimiser l'espace d'attributs dans l'apprentissage automatique utilisée pour le résumé automatique extractif

4.1 Détails de l'article

Evaluation of Efficiency of Linear Techniques to Optimize Attribute Space in Machine Learning: Relevant Results for Extractive Methods of Summarizing

Jesus Antonio Motta, Laurence Capus et Nicole Tourigny

Article publié dans **Computer and Information Science (**Volume 5, No. 6, pp. 58-72, Novembre 2012)

4.2 Résumé

Un défi majeur dans le domaine de l'apprentissage automatique, en particulier dans les problèmes de classification, est d'optimiser l'espace d'attributs afin d'obtenir une fonction de classification, qui sera utilisée pour discriminer les objets non vus. Plusieurs méthodes pour optimiser l'espace d'attributs peuvent être utilisées : certaines d'entre elles choisissent les attributs les plus importants et les autres extraient certains attributs pour créer un nouvel ensemble plus petit de variables. Ces méthodes, permettent soit la sélection d'attributs (décomposition en des valeurs singulières, K-Means, réseaux neuronaux de Kohonen) ou une nouvelle extraction d'attributs (analyse en composantes principales, analyse factorielle). Six techniques d'apprentissage automatique ont été utilisées pour abduire la fonction de classification. Les résultats montrent que l'application des cinq méthodes linéaires choisies pour optimiser l'espace d'attributs dans le processus de résumé automatique par extraction est pertinente. Ils montrent aussi quelle technique d'apprentissage automatique est préférable d'utiliser avec chaque méthode linéaire pour obtenir une meilleure efficacité.

Mots clé : Espace d'attributs, apprentissage automatique, classification, abduction, résumé automatique

4.3 Abstract

One major challenge in the field of machine learning, especially in classification problems, is to optimize the attribute space in order to obtain a classification function. Several methods to optimize the attribute space can be used: some of them select the most relevant attributes and the other ones extract certain attributes to create a new smaller set of variables. These methods, allow either attribute selection (Singular Value Decomposition, K-Means, Kohonen Neural Networks) or new attribute extraction (Principal Component Analysis, Factor Analysis). The validation phase was focused on the discrimination power of the obtained classification function. For that, six techniques of machine learning were used to abduce the classification function. The results show that the application of the five chosen linear methods for optimizing attribute space in the automatic summarization process by extraction is relevant. They also show which machine learning technique is preferable to use with each linear method to obtain a better efficiency.

Keywords: attribute space, machine learning, classification, abduction, automatic summarization.

4.4 Introduction

In machine learning problems and specially classification problems, a space of concepts, variables or features, is used to induce or to abduce the classification function. This space may be used to extract the training set during the inductive/abductive process. Unfortunately, this initial space is generally too large and entropic. In other words, this space contains too much noise and numerous irrelevant features, thus creating the well-known problem of the curse of dimensionality (Smale, 1997). When data is too scattered, good estimations and consequently the elaboration of good classification models are not possible. If one applies a machine learning method to such spaces, one could have an over-fitting problem of the training set, which will generalize poorly the new items or examples to be classified (Langley, 1994). In contrast, if one tries to travel through the entire space or a large part of this space in order to find optimal solutions, one will be face to a polynomial, exponential or combinatorial explosion problem of the search time and the saving space of intermediate states, making this problem intractable, in most of cases. Let us remind that an intractable problem is a polynomial problem with a solution in theory but not in practice (Hastie *et al.*, 2009).

Several approaches are used to optimize the attribute space either by selecting the most relevant attributes or by extracting certain attributes to create a new smaller set of variables. This way of doing has been recently apply to automatic summarization process (Motta *et al.*, 2011). In the present paper, we propose to enrich these first results by adding an experiment we conducted with methods for optimizing attribute space, well-known but not enough used in this field. We have chosen five methods: three to select attributes that are Singular Value Decomposition (SVD) (Golub et Van Loan, 1996; Stewart, 1993), K-Means (KM) (Hastie *et al.*, 2009) and Kohonen Neural Networks (KNN) (Kohonen, 1990), and two others that allow attributes extraction, i.e. Principal Component Analysis (PCA) (Jolliffe, 2002) and Factor Analysis (FA) (Kim et Mueller, 1978). These methods commonly used for data clustering and unsupervised machine learning will be described in the next section.

From a corpus of 1250 textual documents from DUC 2006 (NIST, 2006), we applied these five methods to optimize the space of attributes. In fact, this attribute space is a matrix, where the rows represent the sentences of the documents and the columns are the words of

these sentences. Each item of the matrix corresponds to the frequency of the word in the sentence in a vector space model (Motta *et al.*, 2011). By applying one of the five methods on this matrix, we obtained an optimized attribute space composed of sentences with important information. Rather than evaluating produced summaries, we induced a classification function and assessed its performance. This way, we were able to determine the power to discriminate of the five chosen methods in order to optimize the attribute space, given that a good performance of these functions depends largely on the choice of the training set (Ikonomakis *et al.*, 2005). By focusing our interest on this principle, we did not thus used methods that allow the evaluation of the quality of produced summaries, as ROUGE (Lin et Hovy, 2003) for instance. We preferred this other protocol of validation. First, we applied six current techniques of data mining on the optimized attribute space to induce a classification function. Next, we evaluated the classification performance with the metric F_measure (harmonic mean of precision and recall) and ROC curves. We can conclude that the five chosen methods are relevant and appropriate with the application of machine learning techniques for automatic summarization process by extraction.

Section 4.5 presents an overview of the chosen reduction methods. Section 4.6 describes the experiment that we conducted. Section 4.7 shows the abduction process. Section 4.8 describes the evaluation criteria. Section 4.9 shows the tables and graphs produced with the experimental values and the discussion of the results obtained. Section 4.10 concludes on the importance of such an experiment.

4.5 The Five Methods Chosen for Our Experiment

This section presents the five methods we used for the reduction of the attribute space and how we applied them. To well understand, it is important to know that each method was applied to a matrix E in the vector space model, where the rows represent the sentences of the documents and the columns are the set of attributes or vocabulary of the corpus of documents. The presence of every word in every sentence of the corpus of documents has been measured by multiplying the word frequency with the frequency inverse of the words in the document, i.e.:

$$tf_t \times idf = tf_t \times \ln\left(\frac{|P|}{|\{t \in p\}|}\right) \quad (4.1)$$

where tf_t is the frequency of word *t* in the sentence *f*, $|P|$ is the total number of sentences in the corpus and $|\{t \in p\}|$ is the number of sentences containing the word *t*.

To apply the five methods on this matrix, we then created algorithms based on them. Following, we explain the principle of each method and how we applied it to our research.

4.5.1 K-means (KM)

K-means (Hastie *et al.*, 2009) is a kind of clustering based on centroids , which groups *n* objects or points $(x_1, x_2, \dots, x_n)$ that have certain characteristics in *k* partitions, groups or clusters$(k < n)$. The objects in the cluster are correlated with it, but they are not correlated with other clusters. The optimization problem that arises is to find those *k* clusters of the set $S = [S_1, S_2, \dots, S_k]$ and their centers, to assign the objects to the nearest cluster center, so that the square of their distances from the cluster is minimal. That is, it tries to minimize total intra - cluster variance or square error function:

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} \left(x_j - \mu_i\right)^2 \tag{4.2}$$

where there are *k* **clusters** S_i, $i = 1, 2, \dots, k$ and μ_i is the centroid or mean point of all points x_j in S_i. A simple algorithm would consist in determining the centroid coordinates and the distance between each object and the centroid, and next grouping the objects with the minimum distance. This is a heuristic algorithm and there is no guarantee that it converges to a global optimum. The distance between points and the centroid can be obtained in several ways: Euclidean, sum of absolute differences, cosine (*1 – angle between points*), correlation (*1 – correlation between points*), Hamming (*percentage of bits that differ*) and others. Obviously the cluster centroids are different depending on the distance measure used.

In our research, as already mentioned, the items of the matrix *E* are calculated by using the formula $tf * idf$. Our clustering technique is applied to this matrix defining $k = 2$ groups. The idea is to partition the matrix *E* into two groups that correspond to the very important sentences in an information viewpoint and the less important sentences. This way, we can discriminate the relevance of the sentences of the corpus in order to eliminate those that provide less information. For the process of discrimination of sentences, we find the silhouette (Rousseeuw, 1986) of each cluster and identify the sentences that have greater

dissimilarity with the neighboring cluster. Actually, separating the important sentences from those unimportant is a difficult process because in reality it can be subjective and involve a high degree of abstraction. For this reason, we preferred to use the concepts of 'very important sentences', and 'less important sentences', and identify more clearly their group membership. To represent the silhouette $s(i)$ of a cluster, we proceed as follows:

Let:

- *i* an object in the data set
- *A* the cluster to which *i* has been assigned
- $a(i)$ the mean dissimilarity of *i* to all other objects of *A*
- *C* a different cluster of *A*
- $d(i, C)$ the average dissimilarity of *i* to all objects of *C*
- $b(i) = \min_{C \neq A} d(i, C)$
- *B* the cluster, where the minimum is reached (i.e.$d(i, B) = b(i)$)

Then the silhouettes(i) obtained by combining $a(i)$ and $b(i)$ is as follows:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & if\ a(i) < b(i) \\ 0 & if\ a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & if\ a(i) > b(i) \end{cases} \tag{4.3}$$

one can see clearly that $1 \leq s(i) \leq 1$ for each object *i*.

To analyze how $s(i)$ works, one takes for instance the case when $s(i)$ is close to 1, that is $s(i)$ has its maximum value. This implies that the intra-cluster dissimilarity $a(i)$ is much smaller than the inter-cluster dissimilarity $b(i)$. One can say 1 is 'well clustered' and doubt is small compared to the question whether it was well clustered, because the other cluster in any case is far from the first. When $s(i)$ is close to 0, then $a(i)$ and $b(i)$ are approximately equal and there is no clarity to what cluster allocates *i*. When $s(i)$ is negative and close to -1, then $a(i)$ is much larger than $b(i)$, so on average *i* is closer to *B* than *A*. It would have been then better to assign *i* to *B* before than *A*. One could conclude that this object has been wrongly classified.

4.5.2 Kohonen Neural Networks (KNN) (Kohonen, 1990)

It is an artificial neural network and as such it must find common features, regularities, correlations or categories in the input data and incorporates them into its internal structure of connections. Therefore, neurons are self-organized according to stimuli from outside.

In this type of process, neurons compete with one another to advance a given task. When they discover an input pattern, only one neuron (BMU, Best Matching Unit) or group of neighboring neurons is activated. Neurons compete for being activated and at the end only one is winner and the others are forced to produce a minimum response level. The ultimate goal of this process is categorizing the data which enters the network. Similar values are classified into the same category and thus should activate the same output neuron. In essence, it is intended to map similar patterns in the input signal space (vectors patterns) in contiguous locations in the output space that is much smaller.

The K-NN method consists of an input layer composed of *N* neurons (one for each input variable), which receives and transmits to the output layer (made up of *M* neurons) outside information. This last layer processes information and forms the pattern map. Each input neuron *i* is connected to an output neuron *j* through aW_{ji}weight. In this way, the output neurons have an associated weighting vector W_j,called reference vector that corresponds to an average of the category represented by the output neuron *j*. As one can see in Figure 4.1, each node has a specific topological position (*x* and *y*, its coordinates in the lattice) and contains a weight vector of the same dimension of input vectors.

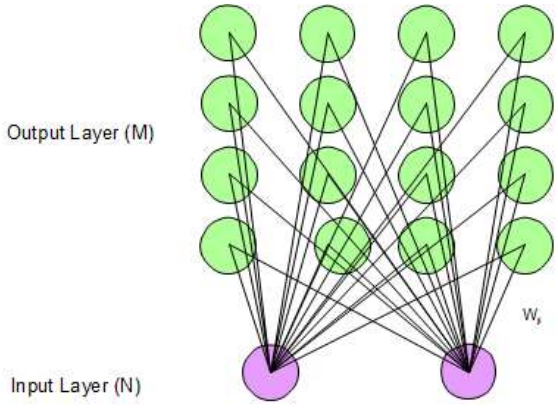


Figure 4.1 *A simple Kohonen Neural Network*

The classification process of the network corresponds to the SOM algorithm. First, it chooses at random a weight vector of the nodes W_j. It makes a loop through the nodes to choose an input vector x using Euclidean distance to find the similarity between this vector and vector weights W_j. Next, it identifies the node that produces the smallest distance (node BMU) and moves this node BMU and its neighbors near the input vector *x*. This approach as a time function *t* is called the learning rate $\alpha(t)$. As this approach is being updated and new vectors are assigned to the map, the learning rate approaches 0. Along with it, the radius of neighborhood also decreases. This update is performed using the following formula:

$$W_j(t+1) = \begin{cases} W_j(t) + \alpha(t)\Theta(t)(x(t) - W_j(t)) & if\ j \in N_j(t) \\ W_j(t) & if\ j \notin N_j(t) \end{cases} \tag{4.4}$$

where *t* is the current iteration, N_j is the vicinity of the neuron *j*, $\Theta(t)$ is the neighborhood function. Finally, the SOM algorithm increases *t* and repeats from the loop to $t < \lambda$ (time limit). The number of iterations is fixed a priori. Once the process is finished, the map is sorted topologically speaking, grouping *n* vectors corresponding to *n* next adjacent neurons or in the same neuron.

In our research, we started from the matrix *E* and defined the number of neurons equal to the number of words (variables). We tried to discriminate the important sentences of those less significant from the vector of centroids of the output neurons, in order to find the correlation of each variable to the neuron and then collect the sentences that contain these variables.

4.5.3 Factor Analysis (FA) (Kim et Mueller, 1978)

This statistical method attempts to describe the influence between correlated variables and uncorrelated latent variables or constructs called factors. It distinguishes between common variance and unique variance. Common variance is the part of the variation of the variable that is shared with the other variables. The unique variance is the variation of the variable that is unique to that variable. The FA method aims to find a new set of variables, fewer in number than the original variables, to express what is common to these variables.

There are basically two types of factor analysis: exploratory, which attempts to discover the nature of the latent variables that influence a set of responses, and confirmatory that tests whether a set of constructs has influence over a set of answers. These two types are based on the model shown in Figure 4.2 (Joreskog et Van Thillo, 1972; Kim et Mueller, 1978). This model proposes that each response observed from measure 1 to 5 is influenced in part by the latent factors Factor 1 and Factor 2 and the errors E1 to E5.

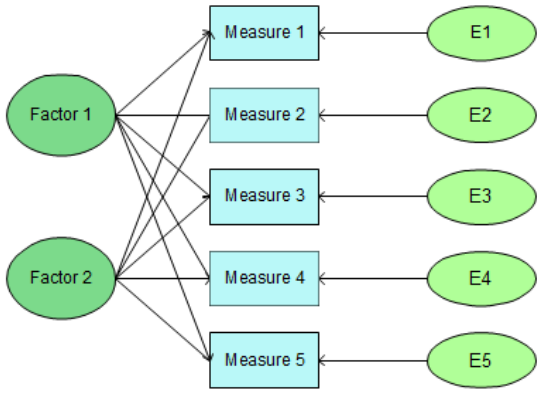


Figure 4.2 *Model for the two types of factor analysis*

The FA method is performed by finding the correlation patterns or covariance between the values of measured variables. The high correlation measures are the most likely influenced by the same factors, while measures with low correlation are probably influenced by other factors.

Formalizing these above ideas:Let $X_1, X_2, \ldots, X_p$ be the variables under analysis, then one can express the components (measurements) of the first variable (vector) X_1 as:

$$\begin{aligned} x_{11} - \mu_1 &= \lambda_{11}F_1 + \lambda_{12}F_2 + \cdots + \lambda_{1k}F_k + \varepsilon_1 \\ x_{21} - \mu_1 &= \lambda_{21}F_1 + \lambda_{22}F_2 + \cdots + \lambda_{2k}F_k + \varepsilon_2 \\ &\cdots \\ x_{p1} - \mu_1 &= \lambda_{p1}F_1 + \lambda_{p2}F_2 + \cdots + \lambda_{pk}F_k + \varepsilon_p \end{aligned} \tag{4.5}$$

where $F_1, \ldots, F_k$ are the common factors, μ_1 is the mean of the variable X_1 and the coefficients $\lambda_{ij} (i = 1, \ldots, p;\ j = 1, \ldots, k)$ are the factor loadings.

Generally, one can write:

$$x_i = \sum_{j=1}^{k} \lambda_{ij} f_j + \varepsilon_i + \mu_i \quad (4.6)$$

The k factors $F_1, \ldots, F_k$ are assumed to be uncorrelated random variables with mean 0 and variance 1. The errors $\varepsilon_1, \ldots, \varepsilon_p$ are uncorrelated random errors with mean 0and variances ψ_1 ... ψ_p, as well as the ε_i are uncorrelated with the factors F_j.

One can write the above equation in matrix form as follows:

$$X = \Lambda f + u + \mu \quad (4.7)$$

Where X, ε and μ are vectors of dimension $(p \times 1)$ and f is also a vector of dimension $(k \times 1)$. Λ is a matrix of unknown constants $p \times k$ $(k < p)$, called the matrix of factor loadings.

As stated:

$$Var(\varepsilon_i) = \psi_i$$

then:

$Var(\varepsilon) = \psi = Diag\big(Cov(\varepsilon)\big) = \psi_1, \psi_2, \ldots, \psi_p$.(4.8)Assuming also that $Cov(F) = 1$and, as established $Cov(j, u) = 0$, then a solution to the set of equations defined above subject to the restrictions for F, is the factors and Λ is the loading matrix.

In our research, as proceeding, we applied the FA method on the matrix *E*. We tried to discover the latent patterns that could discriminate the sentences carrying information of those phrases that do not carry it.

4.5.4 Principal Components Analysis (PCA*)* (Jolliffe, 2002)

This method studies the relationships that exist between p correlated variables, finding another set of new uncorrelated variables called principal components. These components successively explain most of the total variance, unlike the previous method that distinguishes two types of variance: common and unique. The new variables are linear combinations of the foregoing and are constructed successively in order of significance, determined by extracting the total variability of the sample. In other words, the method

seeks to find $m < p$ variables that are combinations of the original p variables which are not correlated and containing the most of the information or data variability.

By specifying the above ideas, one could have:

Consider a number of variables $x_1, x_2, \ldots, x_p$ on a group of objects or individuals. From this group, calculate a new set of variables $y_1, y_2, \ldots, y_p$ that do not correlate with each other and whose variances successively will decrease. The first variable contains all the possible variation of the total amount of variation; the second contains the largest possible amount of the remaining variation and so on.

Each y_j $(j = 1, \ldots, p)$ is a linear combination of the original x_i as follows:

$$y_j = a_{j1}x_1 + a_{j2}x_2 + \cdots + a_{jp}x_p = a_j x \tag{4.9}$$

where $a_j^{'} = \left[a_{1j}, a_{2j}, \ldots, a_{pj}\right]$ is a constant vector and $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$. (4.10)

As it is necessary to ensure the orthogonality of the transformation (the new variables are uncorrelated, and its magnitude is unitary), then the norm of the vector $a_j^{'}$ must be equal to 1, i.e.:

$$a_j^{'} a_j = \sum_{k=1}^{p} a_{kj}^2 = 1 \tag{4.11}$$

The first component is calculated by choosing a_1 such as y_1 so that it has the greatest variance. The second component is calculated by choosing a_2 so that it is uncorrelated with y_1 and forth so that the variables obtained will have increasingly less variance. For the first component, for example, we obtain a_1 such as it maximizes the variance y_1 subject to the constraint $a_1^{'} a_1 = 1$. Knowing that $a_1^{'} x$ is a linear combination, then $Var\left(a_1^{'} x\right) = a_1^{'} \sum a_1$. Thus, the problem is to maximize this function subject to the restriction $a_1^{'} a_1 = 1$. It is noted that the unknown is precisely the vector a_1, the vector that will give the optimal linear combination.

Applying quadratic programming techniques finally leads to a diagonal matrix of eigenvalues, whose elements represent the obtained major variances that are associated with the corresponding eigenvectors, i.e. the principal components sought.

By applying this method to our matrix *E*, we identified the first principal components (which contain more variance), and the words of highest correlation with these components for later use them, that is to discriminate the important phrases of those unimportant.

4.5.5 Singular Value Decomposition (SVD) (Golub et Van Loan, 1996; Stewart, 1973)

This method is based on the Spectral Theorem to factor matrices. Thus, the singular values of a given matrix *A* are obtained from a diagonalization process of this matrix and its singular values are precisely the elements of this diagonal. These values are sorted from highest to lowest and are associated with corresponding singular vectors.

By specifying the above ideas, one has:

Let A be a $m \times n$ matrix, then there are two orthogonal matrices *U* and *V* such that $D = U^T AV$, where D is a diagonal matrix whose entries d_i are called singular values of matrix *A* and are arranged in decreasing order. *U* is an $m \times m$ square matrix whose columns are the left singular vectors and *V* is also a square matrix of order $n \times n$ whose columns contain the right singular vectors. *D* is a diagonal matrix of order $m \times n$.

One could say that the singular values of matrix *A* are the lengths of the semi-axes of the hyperellipse that maps the unit sphere by the matrix *A*. They are therefore non-negative real numbers. To better understand this statement, consider a pair of vectors *x* and *y* in Euclidean space $\mathbf{R}^2$ which are orthogonal. Imagine further that these vectors have the images *Ax* and *Ay* by the matrix *A*. If one rotates these vectors, they will describe the unit circle and their images *Ax* and *Ay* will describe an ellipse centered at the origin of coordinates (See Figure 4.3).

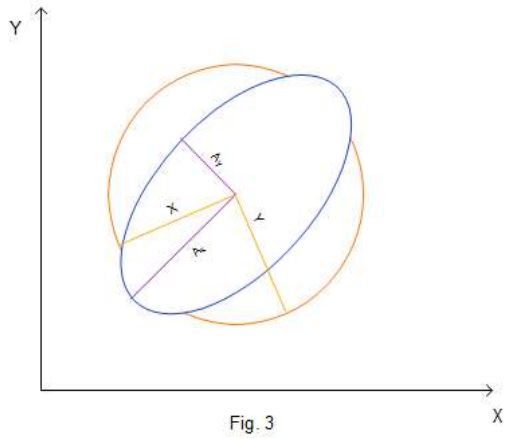


Figure 4.3 *Sphere transformation for matrix A*

In fact, there is a position of the vectors x and y such as their images are the semi-axes of the ellipse, i.e., they are perpendicular too. This fact can be exploited to the deductions that follow:

Set s_1 and s_2 are the singular values of A (the lengths of the semi-axes), then there are unit vectors u_1 and u_2 such as s_1u_1 and s_2u_2 are the vectors that represent the semi-axes of the ellipse. Therefore, u_1 and u_2 are orthogonal.

Let $x = v_1$ and $y = v_2$, then Av_1 and Av_2 are the semi-axes of the ellipse. Then $Av_1 = s_1u_1$ and $Av_2 = s_2u_2$. By expressing these equations in matrix form, one has:

$$A\left[v_1 v_2\right] = \left[u_1 u_2\right]\begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} = AV = US \tag{4.12}$$

Knowing that U and V are orthogonal and the diagonal matrix $S = \sum$ and by doing $U^T = \begin{bmatrix} u_1^T \\ u_2^T \end{bmatrix}$ and $V = [v_1 v_2]$, then $U^T AV = diag(\sigma_1, \sigma_2) = \Sigma$ and $A = U\Sigma V^T$. (4.13)

Obviously the above result can be generalized to the space $\mathbf{R}^n$.

To apply this method in our research, we also used the matrix E. We factored this matrix, we next founded the largest singular values associated with the right singular vectors V and finally we identified the components of these vectors containing the highest correlation. In other words, we started to identify the elements of highest variation to determine the information-bearing phrases.

4.6 Experiment

Our experiment was performed on a corpus of documents from DUC 2006 (NIST, 2006). This collection consists of 1250 documents that are grouped into 50 topics. We can resume the experiment by the following steps. We built the matrix $tf * idf$ and applied each of the chosen methods on this matrix. The training set was then created by selecting the sentences that contain more information to label them as important sentences and sentences less carriers of information as unimportant. From this training set, we used machine learning techniques to abduce classification functions. We selected six machine learning techniques based on different approaches in order to obtain representative results. These six techniques are efficient and then largely used in the data mining field. For each chosen method, six classification functions were abduced and evaluated. We compared the results between them.

Following, we present briefly the abduction functions retained. We described next the evaluation criteria used to calculate the performance of the classification obtained. The results are described and discussed in Section 4.

4.6.1 Abduction Functions

We wanted to try techniques from several approaches: probabilistic, linear regression, trees and neural networks. Our idea is to determine not only the most appropriate technique but also its relationship with the most appropriate approach. With the application of these machine learning techniques, we abducted a function that enables to measure the discriminatory power according to the space reduction method used, as well as establish the effectiveness of the reduction-optimization methods chosen. For the application of each classification technique, we started from a set of instances which will carry information labeled each as important class (the subspace selected by applying the reduction technique) and instances of less information carriers classed as are not important (the subspace discarded by applying the reduction technique).

The first technique used is Support Vector Machine (SVM), which may be linear or nonlinear, using polynomial functions for example (Cortes et Vapnik, 1995; Vapnik, 1999). In our case, we used it as a linear technique. The second technique, Naïve Bayes (NB), based on the Bayes' theorem provides a way of calculating the probability of a hypothesis

(a posteriori) based on their previous or a priori probability (Hastie *et al.*, 2009). The Logistic Regression (LR), the third technique, is a probabilistic regression model whose dependent variable Y can only take two values that are explained by a set of predictors $x_1, x_2, \ldots, x_k$ (Agresti, 2007). The fourth technique, Random Forest (RF), combines a selection method called bagging with a tree induction technique (Breiman, 2001). The bagging produces replicas of the training set sampling with replacement of training instances. The classifier Multilayer Perceptron (MLP), the fifth technique, is a neural network containing hidden layers that interact with the input layer and output layer of the network, allowing classifying elements of a state space that are not linearly separable (Rosenblatt, 1957; Rumelhart *et al.*, 1986). The sixth and last technique is the Radial Basis Function Neural Networks (RBFNN). This network is a universal classifier composed of three layers: input, hidden and output (Buhmann, 2003). Some processing is not performed in the input layer. The hidden layer performs a nonlinear and local transformation on local data or input signals. The output layer allows a linear combination of activations of the hidden layer, that is gives the output of the network.

4.6.2 Evaluation Criteria

We used different metrics to evaluate the performance and the quality of classification for different machine learning techniques, which are implemented on the training sets obtained from spaces subject to the attribute optimization-reduction process. It is very important to note that a good performance of a machine learning technique is based on a good choice of the training set. Given our classifiers are binary, we use a contingency table where the entries are the two classes obtained by classification functions: important sentence and non-important sentence. This type of table, called confusion matrix (Table 4.1), displays four sets of sentences:

- True Positive (TP): if the function has correctly predicted the phrase labeled as important;
- True Negative (TN): if the function has predicted a phrase labeled as not important when it is not important;
- False Positive (FP): if the function has predicted a phrase not important as being important; and,

- False Negative (FN), if the function has predicted that a sentence is important when it is not important.

The values obtained by grouping the sentences in this way will be used to obtain three metrics for performance evaluation of classifiers: precision, recall and $F\_{measure}$. Also from these data, we calculated the additional metrics sensitivity and specificity to construct ROC curves (Receiver Operation Characteristic) to try to visualize the quality of classifiers (Hastie *et al.*, 2009; Lasko *et al.*, 2005).

Table 4.1 *Confusion Matrix*

Sentence Class	Predicted Class	
	Important	***Not important***
Important	True Positive Cases	False Positive Cases
Not important	False Negative Cases	True Negative Cases

Recall (R) is the proportion of positive cases in the total of cases classified as positive (True Positives + False Negatives). It informs about the ability of the classification function to correctly classify the important phrases. We calculate the Recall as:

$$R = \frac{TP}{TP + FN} \tag{4.14}$$

Precision (P) informs us of the ability of the function to classify the important phrases when they are truly, i.e. the proportion of correctly classified phrases in relation to the total important phrases (True Positives + False Positives). Precision is calculated as:

$$P = \frac{TP}{TP + FP} \tag{4.15}$$

The average of the two previous metrics calculated with its harmonic mean that is called F_measure (also known as F_{score} or F1 score) and it is obtained as:

$$F\_{mesure} = 2\frac{R \times P}{R + P} \tag{4.16}$$

In addition to the above metrics, we thought it is necessary to use another tool to analyze the quality of the classification functions. This tool has its origin in the analysis of signals from radar to distinguish true signals from which are not (noise). It is called curve analysis of Receiver Operation Characteristic (ROC). Its implementation in our research enables us to assess the accuracy of the classification models obtained, and to obtain a unified model of the evaluation process.

The curves are produced by calculating the sensitivity (true positive rate) versus specificity (1-false positive rate) and show a continuous variation of the observation points obtained (Lasko *et al.*, 2005). From the observation of several curves, we obtain a qualitative comparison knowing that a curve on the top and to the left has the greatest accuracy. Additionally, obtaining the area under the curve (AUC) indicates the probability of success of the function to identify a sentence that is important. These metrics are obtained as follows (Spackman, 1989) :

$$Sensitivity = TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{4.17}$$

$$specificity = TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR \tag{4.18}$$

$$FPR = 1 - specificity \tag{4.19}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.20}$$

Thus, Recall, Precision, F_measure , sensitivity, specificity and ROC curves, provide us the values necessary for assessing the quality of the selected training set, by analyzing the performance of classifiers applied on them.

4.7 Results and Discussion

This section aims to show the results obtained from our experiment. As first results, we present the confusion matrix for each method chosen to optimize the attribute space. We also give the values computed for Recall, Precision and F_measure. Next, all the ROC curves are given and discussed. We then conclude on the relevance of using the five chosen methods (KM, K-NN, FA, PCA, and SVD) to optimize the attribute space in extractive summarization.

4.7.1 Predictions and Confusion Matrices for the Different Method Analyzed

For the selection of instances to be classified, we used sub-sampling technique that selects the 2/3 of the training set for inducing the classification function. The remaining 1/3 was used as a sub-sample for the corresponding test. To obtain the evaluation metrics, we relied on three different software that yielded similar results: Tanagra (Rakotomalala, 2005), Weka (Holmes *et al.*, 1994), and Orange (Demzar et Zupan, 2010).

The five following tables (Tables 4.2 to 4.6) show the values of the performance of classifiers applied to the spaces which have been optimized through the use of the five chosen methods. Each table presents the corresponding confusion matrix that has two values as input, class important and non-important class, and the values of the predictions: Recall, Precision and F_measure . As we already mentioned, we wanted to examine the predictive power of each function associated with each classifier with respect to the attribute space that acts on each function.

More precisely, Table 4.2 presents the confusion matrix and the predicted values for the algorithms applied to the matrix $tf * idf$ of attributes optimized by applying the KM method. By noting the values of the table, we realize that the values of Precision, Recall and F_measure for each algorithm are very high, with values of 1 or approaching this maximum. This would indicate that the KM method is quite efficient for the selection of spaces that can be used by all these algorithms in the way of classifying the sentences of a set of documents as important and not important. It is important to note here that the KM method can be used as a classification technique in an unsupervised machine learning approach.

Table 4.2 *K-means method (prediction and confusion matrices)*

Algorithm	*Class*	*Confusion Matrix*		*Predictions*		
		Important	***Not Important***	***Recall***	***Precision***	F_{mesure}
Support Vector Machines (SVM)	Important	91	1	0.9891	1	0.9945
	Not important	0	94	1	0.9895	0.9949
Naive Bayes (NB)	Important	91	1	0.9891	1	0.9945
	Not important	0	94	1	0.9895	0.9947
Logistic Regression (LR)	Important	92	0	1	1	1
	Not important	0	94	1	1	1
Random Forest (RF)	Important	92	0	1	1	1
	Not important	0	94	1	1	1
Multilayer Perceptron (MLP)	Important	91	1	0.9891	1	0.9945
	Not important	0	94	1	0.9895	0.9947
RBF Neural Networks (RBF-NN)	Important	92	0	1	1	1
	Not important	0	94	1	1	1

From observation of Table 4.3, we conclude that the method KNN applied to spaces of attributes may produce smaller spaces with high information content. The discriminatory task of classification algorithms is made easier as demonstrated with the F_measure calculated values for each of them: all values are above 93%, with values 1 or close to it.

Table 4.3 *Kohonen Neural Networks method (prediction and confusion matrices)*

Algorithm	*Class*	*Confusion Matrix*		*Predictions*		
		Important	***Not Important***	***Recall***	***Precision***	F_{mesure}
Support Vector Machines (SVM)	Important	42	0	1	1	1
	Not important	0	42	1	1	1
Naive Bayes (NB)	Important	42	0	1	0.9333	0.9655
	Not important	3	39	0.9286	1	0.9630
Logistic Regression (LR)	Important	42	0	1	1	1
	Not important	0	42	1	1	1
Random Forest (RF)	Important	42	0	1	0.9767	0.9882
	Not important	1	41	0.9762	1	0.9879
Multilayer Perceptron (MLP)	Important	37	5	0.8810	0.9737	0.9250
	Not important	1	41	0.9762	0.8913	0.9318
RBF Neural Networks (RBF-NN)	Important	42	0	1	1	1
	Not important	0	42	1	1	1

Table 4.4 shows the performance of classifiers applied to an attribute space that is optimized with the FA method. The observed values of F_measure are 1 or close to this value.

Table 4.4 *Factor analysis method (prediction and confusion matrices)*

Algorithm	*Class*	*Confusion Matrix*		*Predictions*		
		Important	***Not Important***	***Recall***	***Precision***	F_{mesure}
Support Vector Machines (SVM)	Important	56	0	1	1	1
	Not important	0	44	1	1	1
Naive Bayes (NB)	Important	55	1	0.9821	1	0.9909
	Not important	0	44	1	0.9777	0.9887
Logistic Regression (LR)	Important	56	0	1	1	1
	Not important	0	44	1	1	1
Random Forest (RF)	Important	56	0	1	0.9180	0.9572
	Not important	5	39	0.8864	1	0.9397
Multilayer Perceptron (MLP)	Important	56	0	1	0.875	0.9333
	Not important	8	36	0.8182	1	0.9001
RBF Neural Networks (RBF-NN)	Important	56	0	1	1	1
	Not important	0	44	1	1	1

Table 4.5 illustrates the values of performance obtained by applying classification algorithms to attribute space optimized by the PCA method. The values obtained for F_measure , except for the values obtained for MLP (92.03% and 92.55%), are all equal to 1 or close to this maximum value.

Table 4.5 *Principal components analysis method (prediction and confusion matrices)*

Algorithm	*Class*	*Confusion Matrix*		*Predictions*		
		Important	***Not Important***	***Recall***	***Precision***	F_{mesure}
SVM	Important	61	0	1	1	1
	Not important	0	56	1	1	1
Naive Bayes	Important	61	0	1	0.9531	0.9759
	Not important	3	53	0.9464	1	0.9724
Logistic Regression	Important	61	0	1	1	1
	Not important	0	56	1	1	1
Random Forest(RF)	Important	60	1	0.9836	1	0.9917
	Not important	0	56	1	0.9825	0.9911
Multilayer Perceptron	Important	52	9	0.8525	1	0.9203
	Not important	0	56	1	0.8615	0.9255
RBF Neural Networks	Important	61	0	1	1	1
	Not important	0	56	1	1	1

The performance values obtained based on F_measure applying the SVD method, presented in Table 4.6, show that all are close to 1, except for the values obtained for unimportant class of the classifiers RF (90.91%) and MLP (89.76%).

Table 4.6 *Singular value decomposition (prediction and confusion matrices)*

Algorithm	*Class*	*ConfusionMatrix*		*Predictions*		
		Important	*Not Important*	*Recall*	*Precision*	*F_measure*
SVM	Important	196	0	1	0.9949	0.9974
	Not important	1	67	0.9853	1	0.9925
NaiveBayes	Important	196	0	1	0.9849	0.9923
	Not important	3	65	0.9559	1	0.9774
LogisticRegression	Important	196	0	1	0.9949	0.9974
	Not important	1	67	0.9853	1	0.9925
RandomForest	Important	192	4	0.9796	0.9600	0.9697
	Not important	8	60	0.8824	0.9375	0.9091
MultilayerPerceptron	Important	194	2	0.9898	0.9463	0.9675
	Not important	11	57	0.8382	0.9661	0.8976
RBF Neural Networks	Important	194	2	0.9898	1	0.9949
	Not important	0	68	1	0.9714	0.9854

4.7.2 ROC Curves

Figures 4.4 to 4.8 show the ROC curves obtained for each classifier on different spaces of attributes. AUC values (Area Under Curve) are shown in the box located at the bottom right. We thought it is important to present these graphs, in order to analyze the results in a different viewpoint. If we fix our gaze for example the important class, these graphs would show the probability that the function classifies an important phrase that has been labeled as such. The optimal point of each classifier can be found by identifying the highest and farthest to the left. Thus, this measure can distinguish performance between algorithms.

If we look at the ROC curves in Figure 4.4, we find that all AUC registered values above 75%. The maximum values were obtained with the algorithms SVM (99.88%), MLP (98.41%) and the LR (97.44%), which means that these last three classifiers should be used in conjunction with the KM method. In Figure 4.5, we verify that all AUC values of the graphics are above 75%. The maximum values are for the algorithms NB (99.85%), RT (99.85%) and SVM (94.69%), indicating that the latter three algorithms should be used with the KNN method.

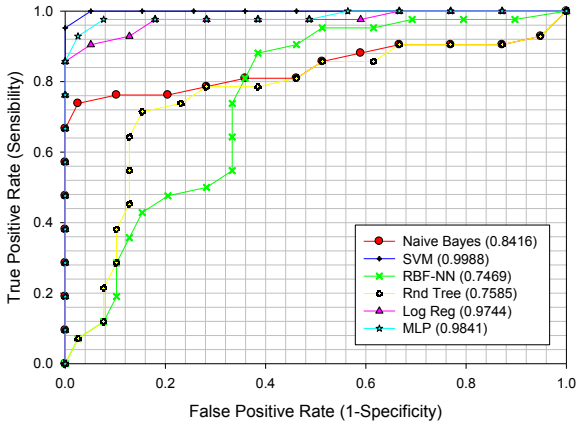


Figure 4.4 *K-means (ROC Curves)*

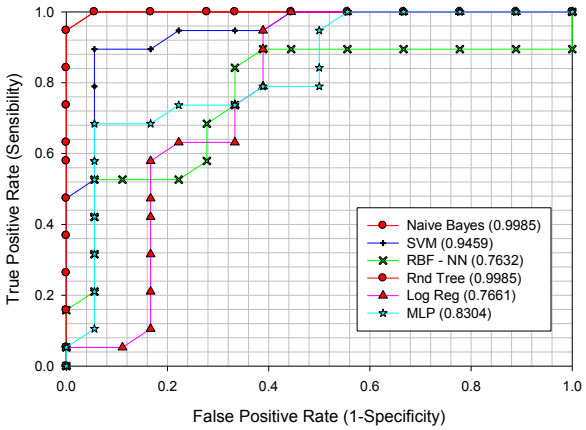


Figure 4.5 *KNN (ROC Curves)*

Unlike the previous methods, the AUC values observed in Figure 4.6 show different results. However, the values of the NB, RBF and RT algorithms are really acceptable: 90.8%, 72.5% and 72.2% respectively. Therefore, the FA method is recommended to be

applied only with these two classifiers. By analyzing the graphs of Figure 4.7, we note that AUC values ranged from 72.1% (SVM) and 81% (RT). We conclude the PCA method could be well applied in conjunction with all the algorithms analyzed with similar results.

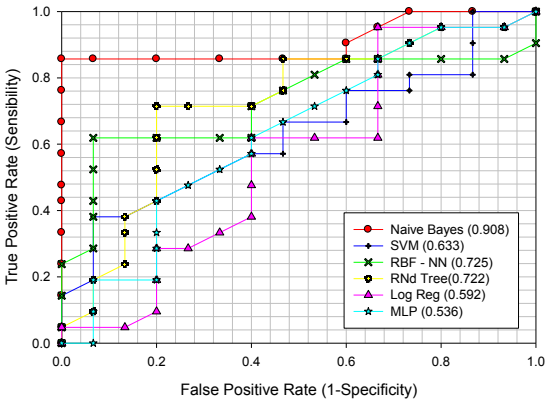


Figure 4.6 *Factor Analysis. (ROC Curves)*

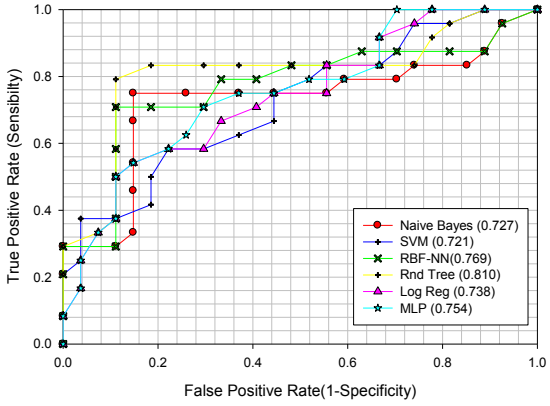


Figure 4.7 *Principal Components Analysis (ROC Curves)*

If we observe AUC values for the different algorithms in Figure 4.8, we note that their values are between 72.1% for RBFNN to 84.8% for MLP and LR. Thus the SVD method may be well applied with all algorithms studied.

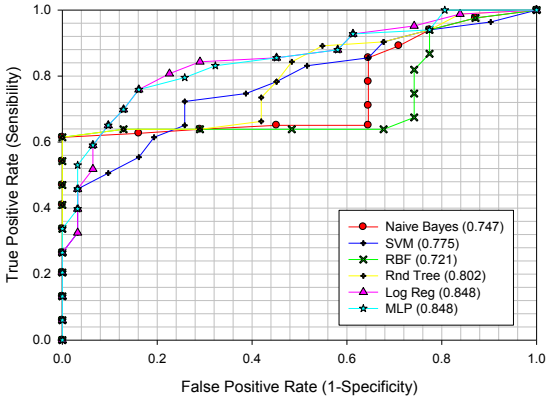


Figure 4.8 *Singular Value Decomposition (ROC Curves)*

Table 4.7 shows the AUC values for each classification algorithm based on the reduction method applied. In the rows, one can observe the AUC values obtained for each machine learning algorithm. The columns give the AUC values of each algorithm for one reduction method given.

Table 4.7 *AUC values by reduction/optimization method for the classification algorithms.*

Algorithm	***Method***				
	K-means	***Kohonen-NN***	***FA***	***PCA***	***SVD***
SVM	0.999	0.946	0.967	0.721	0.775
Naive Bayes	0.842	0.999	0.560	0.727	0.747
Logistic Regression	0.974	0.766	0.510	0.738	0.848
Random Forest	0.759	0.999	0.814	0.810	0.802
Multilayer Perceptron	0.984	0.830	0.536	0.754	0.848
RBF-Neural Networks	0.747	0.763	0.455	0.769	0.721

Generally the average value of the exactness and completeness is represented by the metric F_β (F_1 in our case). We calculated this metric for each classification algorithm applied on each method of reduction and the values obtained are quite high for all. We also use the ROC values of Table 4.7 to conclude about the best methods to use. By observing this table, we could say that the reduction methods, which give greater performance to the classification algorithms, are the KNN and KM methods. The values obtained are above 75%. We could even select one of two methods depending on the algorithm used. For instance, by considering that NB presents a performance of 84.2% with the KM method, then NB may be used with the KNN method with a performance of 99.9%. It is important to note also that the use of the other methods give acceptable results with some algorithms. For instance, SVM with the FA method presents a performance of 96.7% and LR and MLP a performance of 84.8% with the SVD method.

4.8 Conclusion

In order to obtain smaller spaces or attribute subsets with an important information content and abduce classification functions, we used five different methods (KM, KNN, FA, PCA

and SVD) that fall within the attribute selection and extraction approaches. As mentioned at the beginning of this article, the classification functions obtained will be used to produce automatically summaries in an unsupervised approach. The methods used to optimize the attribute space are based on low-dimensional projections, transformations and partitioning of this space for the selection of a representative subset. We call this process reduction/optimization in the sense that, to obtain the corresponding subsets, we use known techniques of mathematical optimization and local optimization. It is important to note that the chosen methods are not commonly used for the reduction of attribute spaces. We then created algorithms based on these methods for this purpose.

By considering that a good subset of attributes should be highly correlated with the classification process (Hall, 1999; Ikonomakis *et al.*, 2005), we relied on the values of performance of classification functions obtained from these subsets to estimate the quality of the reduction method used. We studied the application of six current machine learning techniques, which are SVM, NB, LR, RF, MLP and RBFNN, in order to obtain significant results.

From the analysis of the experimental results, we conclude that the two best methods for reducing the space of attributes, between the methods we have studied, are KM and KNN, which produced excellent results for the six machine learning techniques applied. In addition, the better results were obtained with SVM, MLP and LR for KM, and NB, RT and SVM for KNN.

References

1. Agresti, A. (2007). *Building and applying logistic regression models. An Introduction to Categorical Data*
2. *Analysis*. Hoboken, New Jersey: Wiley.
3. Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32. http://dx.doi.org/10.1023/A:1010933404324
4. Buhmann, M. D. (2003). *Radial Basis Functions: Theory and Implementations*. Cambridge University Press. http://dx.doi.org/10.1017/CBO9780511543241
5. Lin, C. Y. (2004). ROUGE: a Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, Barcelona, Spain, July 25-26, 2004.
6. Cortes, C., & Vapnik, V. (1995). *Support-Vector Networks*. Holmdel, NJ 07733, USA: AT & T Bell Labs.
7. Demzar, J., & Zupan, B. (2010). *ORANGE*: A Software Developed at Laboratory of Artificial Intelligence.
8. Retrieved from http://orange.biolab.si/
9. Golub, G. H., & Van Loan, C. F. (1996). *Matrix Computations*. *Johns Hopkins Studies in Mathematical Sciences*(3rd ed.). The Johns Hopkins University Press.
10. Hall, M. A. (1999). *Correlation-based Feature Subset Selection for Machine Learning*. PhD dissertation, Department of Computer Science, University of Waikato.
11. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). New York: Springer. http://dx.doi.org/10.1007/978-0-387-84858-7
12. Holmes, G., Donkin, A., & Witten, I. H. (1994). *WEKA*: A Software Developed by Machine Learning Group).
13. Retrieved from www.cs.waikato.ac.nz/ml/weka
14. Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text Classification Using Machine Learning Techniques. Paper presented at the WSEAS Transations on Computers.
15. Joreskog, K. G., & Van Thillo, M. (2010). *LISREL: A General Computer Program for Estimating a Linear Structural Equation System Involving Multiple Indicators of Unmeasured Variables*. Princeton, NJ: Educational Testing Service.
16. Kim, J., & Mueller, C. (1978). *Factor analysis: statistical methods and practical issues*. Newbury Park, California: Sage Publications Inc.
17. Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of the IEEE, 78*(9), 1464-1480. http://dx.doi.org/10.1109/5.58325
18. Langley, P. (1994). Selection of Relevant Features in Machine Learning. *Proceedings of the AAAI Fall Symposium on Relevance*. New Orleans, LA: AAAI Press
19. Lasko, T. A., Bhagwat, J. G., Zou, K. H., & Ohno-Machado, L. (2005). The use of receiver operating characteristic curves in biomedical informatics. *Journal of Biomedical Informatics, 38*(5), 404-415. http://dx.doi.org/10.1016/j.jbi.2005.02.008
20. Motta, J. A., Capus, L., & Tourigny, N. (2011). Insertion of Ontological Knowledge to Improve Automatic Summarization Extraction Methods. *Journal of Intelligence Learning Systems and Applications, 3*, 131-138. http://dx.doi.org/10.4236/jilsa.2011.33015
21. NIST. (2006). *Document Understanding Conferences – DUC2006*. Retrieved from http://www-nlpir.nist.gov/projects/duc

22. Rakotomalala, R. (2005). Tanagra: un logiciel gratuit pour l'enseignement et la recherche. *Proceedings of the EGC'2005 Conference*, Amsterdam, RNTI-E-3, vol. 2, 697-702.
23. Rosenblatt, F. (1957). *The Perceptron-a perceiving and recognizing automaton*. New York: Cornell Aeronautical Laboratory.
24. Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*(1), 53-65. http://dx.doi.org/10.1016/0377-0427(87)90125-7
25. Rumelhart, D., Hinton, G., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*, 533-536. http://dx.doi.org/10.1038/323533a0
26. Smale, S. (1997). Complexity Theory and Numerical Analysis. *Acta Numerica, 6*, 523-551. http://dx.doi.org/10.1017/S0962492900002774

CHAPITRE 5. Ajout de la connaissance ontologique pour améliorer le résumé automatique extractif

5.1 Détails de l'article

Ajout de connaissance ontologique pour améliorer le résumé automatique extractif

Jésus Antonio Motta, Laurence Capus et Nicole Tourigny

Article publié dans **Journal of Intelligent Learning Systems and Applications** (Volume 3, No. 3, pp. 131-138, Août, 2011)

5.2 Résumé

Le contexte de cet article est celui du résumé automatique par extraction à l'aide d'une fonction de classification. Cette fonction classe les phrases en deux groupes selon leurs contenus d'information: phrases importantes et phrases non importantes. Les phrases importantes forment alors le résumé. Mais l'efficacité de cette fonction de classification dépend directement de l'ensemble d'entraînement utilisée pour induire la fonction. Cet article propose une façon originale d'optimiser cet ensemble en insérant des lexèmes obtenus à partir de bases de connaissances ontologiques. L'expérience décrite a été réalisée avec quatre algorithmes d'apprentissage automatique, soit le Bayes naïf, les machines à vecteurs de support, les arbres de décision et le perceptron multicouche. Une comparaison des résultats a été effectuée à l'aide des métriques habituelles. L'amélioration obtenue avec les ensembles renforcés est nettement significative pour le processus de résumé automatique. La plus grande amélioration est toutefois celle obtenue avec l'utilisation de Bayes naïf et des machines à vecteurs de support.

Mots clé : Résumé automatique, ontologie, apprentissage automatique, méthode par extraction

5.3 Abstract

The vast availability of information sources has created a need for research on automatic summarization. Current methods perform either by extraction or abstraction. The extraction methods are interesting, because they are robust and independent of the language used. An extractive summary is obtained by selecting sentences of the original source based on information content. This selection can be automated using a classification function induced by a machine learning algorithm. This function classifies sentences into two groups: important or non-important. The important sentences then form the summary. But, the efficiency of this function directly depends on the used training set to induce it. This paper proposes an original way of optimizing this training set by inserting lexemes obtained from ontological knowledge bases. The training set optimized is reinforced by ontological knowledge. An experiment with four machine learning algorithms was made to validate this proposition. The improvement achieved is clearly significant for each of these algorithms.

Keywords: Automatic summarization, Ontology, Machine learning, Extraction method.

5.4 Introduction

Research works on automatic summarization have greatly increased in recent years. Indeed, digital sources of information have become increasingly available. When a user runs a query on Internet, s/he must choose among the retrieved documents those containing relevant information for her/him. The task becomes more difficult when the number of documents increase. An automated system able to 'discover' the essential information is one of the challenges of artificial intelligence, especially in natural language processing. In some cases, methods of machine learning based on symbols are used to tackle this problem.

Automatic summarization can be seen as a problem of transforming one or more documents in a shorter version with preserving information content (Jones, 1999) . The methods used are divided into two main approaches: extraction and abstraction, respectively surface methods and deep methods in a more linguistic viewpoint. A summary obtained by extraction is composed of a set of sentences selected from the source document(s) by using statistical or heuristic methods based on information entropy of sentences. The summarization process by extraction is a relevant alternative, robust and independent of language, compared with the summarization process by abstraction (Mani et Bloedorn, 1998). An abstractive summary is obtained by semantic analysis in order to interpret the source text, and find new concepts to generate a new text that will be the summary. This method requires linguistic processing at a certain level (Salton et Buckley, 1988). In addition, a summary can be produced in a generic way to give a general idea of the contents of documents to be summarized. It can also be based on keywords supplied by the user. In this case, it will contain the most relevant information related to these keywords (Goldstein *et al.*, 1999). Automatic summarization process by abstraction is usually decomposed into three steps: interpretation of source document(s) to obtain representation, transformation of this representation, and production of a textual synthesis(Mani, 2001). Both approaches have their advantages and drawbacks. For this research, we are only interested in automatic summarization process by extraction and how to improve it.

The main problem of this kind of automatic summarization by extraction lies in identifying the most important information of the source documents (Mani et Bloedorn, 1998). Different methods have been used until now with more or less successful results according to measurements based on recall (the number of correct sentences selected on the total number of correct sentences) and precision (the number of correct sentences on the total number of selected sentences) (Steinbach *et al.*, 2006). Some methods use an ontology or ontological knowledge to analyze terms and relations (Korfhage, 1997). More recently, other methods have been reported using machine learning algorithms for determining the description of concepts. These methods build a training set divided into two subsets: important sentences and non-important sentences (Bellman et Kenneth, 1970). This training set is next used to induce a classification function from the concepts description. This function will serve to classify future sentences to produce new summaries. Generally in classification problems, the set of attributes is very large and entropic, with much noise and irrelevant attributes. The well-known underlying problem is named the curse of dimensionality (Bellman et Dreyfus, 1962). Indeed, data too scattered do not facilitate a good estimate, nor obtain good classification models. This problem is actually tackled by using heuristic methods based on linear approximations, which optimize the training set by reducing it or constructing a new smaller set from another series of attributes (Rosenblatt, 1957). The obtained results so far, even if they have progressed, could be further improved.

In this paper, we propose to optimize the training set in an original way. We insert lexemes of ontological knowledge bases into the training set to form a conceptual space, which will be used by the learning algorithm. Our hypothesis is that it is possible to obtain a reinforced set, by using ontological knowledge to select or transform the characteristics of the set. We validated our hypothesis with four machine learning algorithms. We compared their performance by using various evaluation indicators. The obtained results showed that our solution improves the performance; it is then promising for the suite of this research. In Sections 5.5, 5.5.1 and 5.5.2 we will describe the solution proposed. In section 5.6 we will show the evaluation methods. In Section 5.7, we will present the conducted experimentation to validate our solution. In Section 5.8, we will conclude our paper by giving future work.

5.5 Insert ontological knowledge in summary extraction process

Automatic summarization by extraction is a broad topic that uses different approaches, methods or techniques. It seems important at first to give our research framework, i.e. the process that we have considered and decided to improve. Then, we explain what we mean by the insertion of ontological knowledge and how this insertion fits into the summarization process. Finally, we give the evaluation methods that have allowed us to validate our hypothesis.

5.5.1 Summarization process considered

The different methods used for automatic summarization by extraction can be grouped into three approaches: statistical, enriched statistical and machine learning (Bellman, 1970). In this research work, we are interested especially in machine learning approaches because the results obtained are relevant and promising. The key item of these approaches lies on the choice of the training set and its optimization, which will be used to induce the classification function for summarizing futures documents in function of information content. More precisely, the sentences of the documents are represented by vectors, which constitute an initial matrix (Demzar et Zupan, 2010; Hennig, 2008; Xuexian *et al.*, 2004). This matrix corresponds to the training set. The induced classification function enables to classify sentences into two classes: class 1 for important sentences and class -1 for non-important sentences. The summary will be then composed of the sentences of class 1. The crucial problem of this process is the fact that the sentences of this matrix are very entropic. It is necessary to optimize the matrix in order that it becomes an efficient training set.

Although many efforts have been made to improve the quality of summaries obtained, thus approaching those achieved by humans, there are still gaps in terms of accuracy and precision of results. Moreover, most of the summaries obtained are built from a single document. The most evident explanation is that the problems of redundancy in the summary, increases along with the number of documents to be summarized.

The idea of our research work is then to propose a solution to better optimize the training set, i.e. the set of selected sentences forming the initial matrix needed for inducing the classification function. We wanted to find a solution more efficient, which do not need

initially summaries already written to constitute the training set and can be applied on several documents to be summarized.

5.5.2 Insertion of ontological knowledge

Before inserting ontological knowledge, we identify the sentences of the document(s) to be summarized. Next, we make a grammatical categorization to identify the structure of the sentence and then delete stop words. We then create a matrix E, formed of words by sentences. Each item of the matrix contains the value $tf \times idf$ of the word i in the sentence j. This discriminatory value is based on the Salton et al.'s formula (Salton et Buckley, 1988), which evaluates the value of a term compared to a corpus of documents. We insert ontological knowledge to this matrix in order to obtain the new matrix E_0. Our hypothesis lies on the fact that the training set is reinforced by new information, i.e. terms or items with more semantic content and potentially discriminatory. Such an insertion also enables to solve partially the problem of synonymy (Bellman et Kenneth, 1970), one of main open problems of information retrieval. Briefly, the initial matrix is improved by adding a set of sub-trees of hypernyms and hyponyms, for each word.

We elaborated an algorithm to insert ontological knowledge that enables to find conceptual structures from ontology, to compute their importance in an information viewpoint and introduce them in each term of the matrix. This algorithm gives lexemes in function of the words of sentences and inserts their semantic values to the matrix. The optimum lexemes have a factor that adds a semantic value to items of the matrix, improving its performance for classification. More precisely, the algorithm begins to do a search in the ontology by subject and verb. Next, it identifies the various concepts of each sentence by analyzing different sub-trees of parts of the sentence. The sub-trees are built in function of semantic relations of hypernym and hyponym. The algorithm evaluates the various sub-trees and chooses the best one. To finish, it inserts the selected sub-tree in the training set. When all new components of the training set are inserted, we have a new conceptual space enriched.

After inserting ontological knowledge, we do different steps to obtain the final training set and then induce the classification function. First, we filter entropic attributes with algebraic methods. To obtain our new set, we used a similarity transformation matrix, which enables

to find smaller and less redundant subsets of attributes. By applying a transformation matrix to the matrices *E* and E_0, we identify principal components (García-Hernandez *et al.*, 2008) and singular values (Golub et Kahan, 1965) in order to reduce the entropy of the matrix and sort sentences in function their information content. The principal components of a matrix enable to identify groups of variables/words (principal component), greatly connected in the group, but without correlation between groups. The determining factor of this grouping is the variability, which represents the information or importance. We can then choose the first sentences, with the greatest variability, as important sentences and the last ones as non-important.

In detail, we represent the matrix *E* (for singular values for instance), by:

$$E = (u_1 u_2 \cdots u_n) \times \begin{pmatrix} s_1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & & \cdots & s_n \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = U\Sigma V^T = diag(s_1 s_2 \dots s_n) \quad (5.1)$$

Each value s_i corresponds to the variability of each sentence in all sentences. This variability is correlated to the information content of each sentence. We associate the values s_i, the greatest to the smallest, with the corresponding words of the matrix. Next, we label the sentences that contain high values s_i as important (class 1) and those that contain low values s_i as non-important (class -1). Finally, we use machine learning algorithm to induce the classification function, among those proposed by literature.

The induced classification function is able to differentiate sentences with much information (important) to sentences with insufficient information (non-important). So by applying the induced function on new documents, only the sentences containing the most important information are chosen to produce summaries.

5.6 Evaluation method

To evaluate our solution and verify its efficacy, we created a contingency table, named confusion matrix. The inputs of this table correspond to considered classes, knowing that these values are given after applying the classification function to the training set. As the training set is a binary set, we obtain Table 1 elaborated in function of the two classes to be determined: important and non-important.

When the process of classification is finished, we identify four categories of sentences among all those analyzed:

- TP: if the function predicts correctly a sentence labeled as important;
- TN: if the function predicts correctly a sentence labeled as non-important;
- FP: if the function predicts incorrectly a sentence labeled as important;
- FN: if the function predicts incorrectly a sentence labeled as non-important.

Table 5.1 *Confusion matrix used to evaluate efficacy.*

Class	**Predicted Class**	
	Important	***Non-important***
Important	True Positive Case (TP)	False Negative Case (FN)
Non-important	False Positive Case (FP)	True Negative Case (TN)

We used the information given by Table 5.1 to obtain the values of three evaluation indicators known in automatic summarization, that are recall, precision and F-score, as well as ROC curves (Receiver Operation Characteristic) (Ohno-Machado *et al.*, 2005).

Recall (R) is the number of predictions TP divided by the true number of positive instances classified as positives (Korfhage, 1997). It informs about the capability of the classification function to identify a sentence as important when it is really important. The following formula enables to compute recall:

$$R = \frac{TP}{TP + FN} \tag{5.2}$$

Precision (P) corresponds to the number of predictions TP divided by the total number of instances classified as positives (Korfhage, 1997). This indicator informs about the capability of the function to classify correctly a sentence according to all the sentences added to this category. It is computed with the following formula:

$$P = \frac{TP}{TP + FP} \tag{5.3}$$

F-score corresponds to a harmonic average of recall and precision (Korfhage, 1997) and it is defined by:

$$F_{-score} = 2 \times \frac{R \times P}{R + P} \tag{5.4}$$

We also represented our results by using ROC curves. These graphs enable to represent all the pairs of values TPR (True Positive Rate) or *sensitivity* and FPR (False Positive Rate) or 1-*specificity*, resulting of a continuous variation of the observation points in the whole row of observed results (Ohno-Machado *et al.*, 2005). By simple observation of these graphs, we obtain a qualitative comparison. When we apply each model to be evaluated on the training set, the curve placed on the top and to the left has the greatest accuracy. Likewise, the area under the curve indicates the success probability of the model by identifying a sentence as important. The ROC curves then give indications on the accuracy of the classification model, as well as a unified criterion in the evaluation process. The mentioned values are obtained by the following formulas:

$$sensitivy\left(TPR\right) = \frac{TP}{TP + FN} \tag{5.5}$$

$$specificity\left(TNR\right) = \frac{TN}{TN + TP} = 1 - FPR \rightarrow FPR = 1 - specificity \tag{5.6}$$

$$FPR = \frac{FP}{FP + TN} \tag{5.7}$$

Recall, Precision and F-score as well as ROC curves allowed us to evaluate the improvement rates obtained by our solution.

5.7 Experiment and results

The experiment is conducted on a set of documents from Reuters Corpus (Saleh, 2004), a news database that contains approximately 11000 documents, classified into 90 currents events subjects and grouped into two sets, respectively named training and test. Each document contains on average 120 words and 15 sentences. We chose a total number of 2000 documents for our experiment.

For the extraction of ontological knowledge, we used WordNet database (Fellbaum, 1985), developed at Princeton University. This is a database oriented semantically with a very rich frame, greatly used in computational linguistic. It is composed of words related to names, verbs, adjectives and adverbs. Words are organized into sets of synonyms named *synsets*, related by semantic relations of hypernym, hyponym, meronymy and holonomy. WordNet database thus contains 155287 words and 117659 *synsets*.

We also chose four machine learning algorithms greatly used. The first algorithm is named *Support Vector Machine* (Vapnik, 1999). It builds a hyper-plane in an *n*-dimensions space to classification, regression or other tasks. Intuitively, a good separation between classes is obtained when a hyper-plane has the greatest distance for all the nearest points of the training set. The second algorithm is a probabilistic classifier based on Bayes' theorem, *Bayesian Classifier* or *Naïve Bayes (Steinbach et al., 2006)*, but with a great independence hypothesis. In other words, it assumes that the presence or absence of a characteristic is not related to the presence or absence of another characteristic. The third algorithm, *Random Tree* (Bellman et Kenneth, 1970), realizes its classification by building a tree in which the total number of selected nodes is randomly chosen while being equal to:

$$\log_2(attribute\,number+1) \tag{5.8}$$

Finally, the fourth algorithm is Multilayer Perceptron. This is an artificial neuronal network with many layers. The activation function of each neuron is not linear. This neuronal network can be used to identify linearly inseparable classes. The function is learned from multilayers that are totally connected to each other.

The experiment conducted is composed of two parts. In the first part, we produced summaries by extraction from the chosen corpus without inserting ontological knowledge to the machine learning algorithm. We evaluated the obtained results for each algorithm in function of the evaluation methods given. In the second part, we produced summaries from the same corpus, this way by inserting ontological knowledge. We also evaluated the results obtained. By following, we present these results and discuss them.

5.7.1 Results for Recall, Precision and F-score

We used methods of random sub sampling (1/3 for the test set and 2/3 for the training set) and cross-validation (10 crossings). We also based on Tanagra software of Lyon University (Rakotomalala, 2005), Weka software of Waikato University (Holmes *et al.*, 1994) and Orange software of Ljubljana University (Demzar et Zupan, 2010). The two methods used gave similar results.

Tables 5.2 and 5.3 present the values of recall, precision and F-score as well as the confusion matrix for each of the four algorithms evaluated. In Table 5.2, the algorithms were applied to the training set obtained from principal components of the word matrix. In Table 5.3, the training set was obtained from singular values of the matrix. The values are given for each part of the experiment, i.e. before inserting ontological knowledge and after inserting it.

Table 5.2 *Predictions and confusion matrix with principal components.*

Algorithm	With ontological knowledge					Without ontological knowledge				
	Prediction			Confusion matrix		Prediction			Confusion matrix	
	Recall	Precision	F-score	Important	Non-Important	Recall	Precision	F-score	Important	Non-important
Naïve Bayes			0.993					0.978		
Important	1.00	0.986		488	0	0.994	0.964		480	3
Non-important	0.837	1.00		7	36	0.818	0.964		18	81
SVM			1.00					0.999		
Important	1.00	1.00		488	0	1.00	0.998		483	0
Non-important	1.00	1.00		0	43	0.999	1.00		1	98
Random Tree			0.982					0.952		
Important	1.00	0.972		488	0	0.977	0.909		472	11
Non-important	0.674	1.00		14	29	0.579	0.905		42	57
ML Perceptron			0.993					0.983		
Important	0.998	0.988		487	1	0.992	0.974		479	4
Non-important	0.861	0.974		6	37	0.869	0.956		13	86

Table 5.3 *Predictions and confusion matrix with Singular Values.*

Algorithm	With ontological knowledge					Without ontological knowledge				
	Prediction			Confusion matrix		Prediction			Confusion matrix	
	Recall	Precision	F-score	Important	Non-important	Recall	Precision	F-score	Important	Non-important
Naïve Bayes			0.997					0.958		
Important	1.00	0.994		486	0	0.998	0.956		483	1
Non-important	0.858	1.00		3	18	0.776	.987		22	76
SVM			1.00					0.999		
Important	1.00	1.00		486	0	1.00	0.998		484	0
Non-important	1.00	1.00		0	21	1.0	1.00		1	97
Random Tree			0.992					0.957		
Important	1.00	0.984		486	0	1.0	0.917		484	0
Non-important	0.61	1.00		8	13	0.551	1.0		44	54
ML Perceptron			0.992					0.980		
Important	1.00	0.984		486	0	0.971	0.999		470	14
Non-important	0.619	1.00		8	13	0.949	0.869		5	93

By observing Tables 5.2 and 5.3, we note that the performances in terms of recall and precision are high. Naïve Bayes algorithm presents the greatest performance in the case of using principal components. Its performance is followed by the two other algorithms Support Vector Machine and Multilayer Perceptron. When the ontological knowledge is inserted then the greatest performance is those of Support Vector Machine algorithm followed by those of Multilayer Perceptron algorithm. In the case where the space was based on singular values, this is Support Vector Machine algorithm that obtains the greatest performance, followed by those of Naïve Bayes algorithm. When the ontological knowledge is inserted, Support Vector Machine algorithm continues to occupy the first place. The performance of Multilayer Perceptron algorithm is improved at the expense of those of Naïve Bayes algorithm.

5.7.2 Results for ROC curves

In Table 5.4, we observe the AUC values (Area Under Curve) of each ROC curves, for each algorithm applied at a training set obtained from principal components. The results are given before and after inserting ontological knowledge. The last column indicates the relative improvement obtained.

Table 5.4 *Improvement when using principal components.*

Algorithm	Values of ROC		Improvement (%)
	Without ontological knowledge	With ontological knowledge	
Naïve B.	0.668	0.737	10.33
SVM	0.682	0.783	14.8
Random	0.555	0.661	19.1
ML Perceptron	0.449	0.713	58.8

Table 5.5 gives the same values, but this way by considering that the training set is obtained from singular values.

Table 5.5 *Improvements when using Singular Values.*

Algorithm	Values of ROC curves		Improvement(%)
	Without ontological knowledge	With ontological knowledge	
Naïve B.	0.683	0.820	20.06
SVM	0.673	0.811	20.51
Random Tree	0.697	0.740	6.17
ML Perceptron	0.589	0.748	26.99

From the joint observation of Tables 5.4 and 5.5, we can say that the introduction of ontological knowledge in the training set, obtained using principal components or singular values, increases the quality of all algorithms. For instance, we note an improvement of 58.8% and 19.1% for respectively *MultiLayer Perceptron* and *Random Tree* algorithms when using principal components. *Support Vector Machine* and *Naïve Bayes* algorithms are improved of 14.8% and 10.33% respectively, when using singular values. The algorithm with the best performance is *Support Vector Machine* followed by *Naïve Bayes*, when using principal components. In the case of singular values are used, we also observe a very great improvement of the algorithm quality when the ontological knowledge is inserted. *Multilayer Perceptron* algorithm obtains again the highest value with 27%. *Support Vector Machine* and *Naïve Bayes* algorithms occupy respectively the second and the third places with 20.5% and 20.1%. The best quality algorithm is *Naïve Bayes* followed by *Support Vector Machine*.

Table 5.6 shows a comparison of AUC values of each algorithm after inserting ontological knowledge, depending on whether the training sets are obtained from principal components or singular values.

Table 5.6 *Difference of improvement between using pca and using singular values*.

Algorithm	Values of ROC curves		Improvement(%)
	Principal components	Singular values	
Naïve B.	0.737	0.820	11.30
SVM	0.783	0.811	3.60
Random Tree	0.697	0.740	12.00
ML Perceptron	0.589	0.748	4.90

Table 5.6 enables to estimate the difference of improvement between a space based on principal components and another based on singular values. First, we observe that all algorithms studied improve their qualitative performance moving from a space of principal components to another of singular values. The greatest improvement is obtained by *Random Tree* algorithm with 12% followed by *Naïve Bayes* with 11.3%. From this table, we also conclude that the two best algorithms and the most promising to extract summaries with spaces ontologically reinforced, among the four ones studied, are *Naïve Bayes* and *Support Vector Machine*.

Figures 5.1 and 5.2 correspond to ROC curves obtained for each algorithm with training sets produced from principal components and singular values. In these two cases, the results are given without inserting ontological knowledge (5.1), and with inserting ontological knowledge (5.2).

As we already mentioned, the ROC curves offer a way of evaluating the quality of a classification algorithm in function of its capability to give good predictions. In our case, a good prediction corresponds to sentences classified as important, that should take part of summary, and discriminated to non-important sentences.

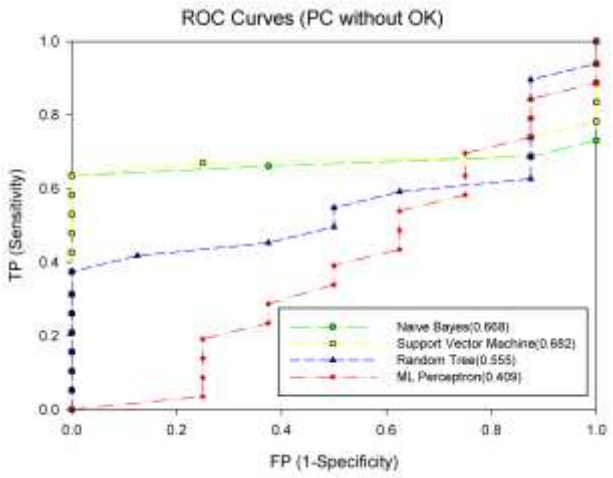


Figure 5.1 *Principal components before inserting ontological knowledge.*

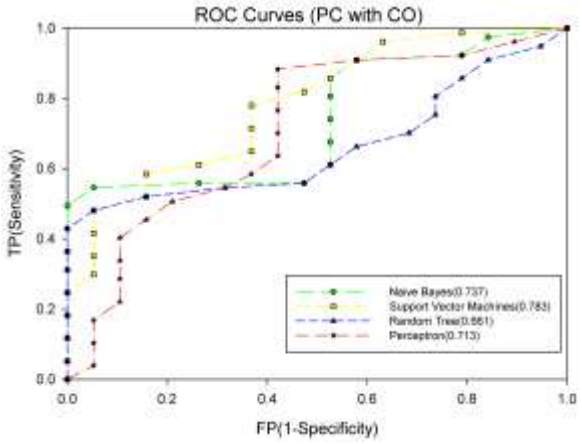


Figure 5.2 *Principal components after inserting ontological knowledge*.

In addition to the information obtained by Tables 5.4, 5.5 and 5.6, careful observation of figures enables to identify the optimum points of each algorithm, simply by placing the point at the highest position to the left. We also compare the accuracy between algorithms. For instance, if we want to compare *Support Vector Machine* algorithm when it reaches a little more than 90% of TP cases face to *Random Tree* algorithm then we see on the Figure

5.4 that *Support Vector Machine* algorithm has an approximate FP rate of 37% and *Random Tree* algorithm has a rate of 70%.

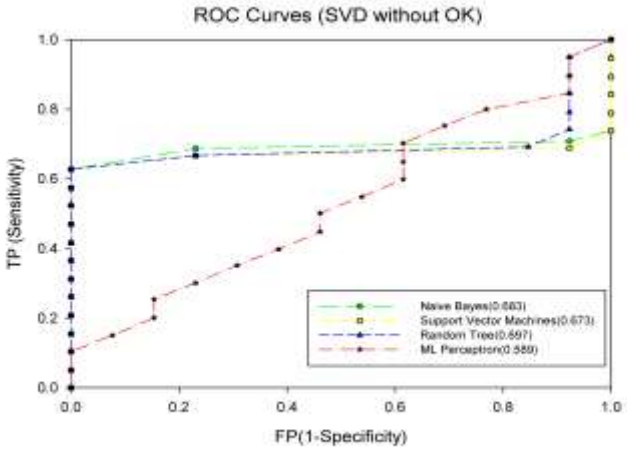
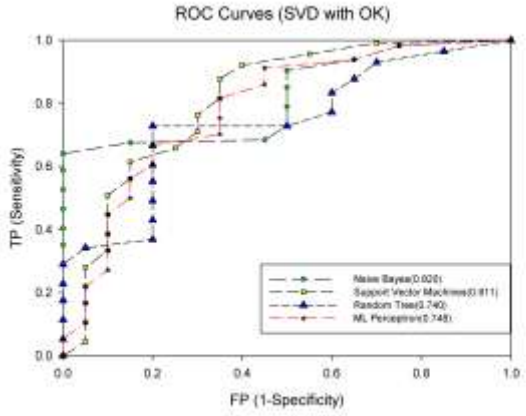


Figure 5.3 *Singular values before inserting ontological knowledge.*



Fi**gure 5.4** *Singular values after inserting ontological knowledge.*

The obtained results show a significantly improvement to the classification function after inserting ontological knowledge, whatever the machine learning algorithm used. More, these results give the two best algorithms. The *Naïve Bayes* and *Support Vector Machine* algorithms should be then applied to automatic summarization with a training set produced from singular values and reinforced by ontological knowledge.

5.8 Conclusion

There are still great opportunities for deepening and development research to find suitable methods for summarizing. In this paper, we studied the behaviour of four machine learning algorithms that induce classification function from training sets. These sets were reinforced by inserting ontological knowledge and used to discriminate the important sentences, from one or several documents, of those which are not. The used algorithms are *Naïve Bayes*, *Support Vector Machine*, *Random Tree* and *Multilayer Perceptron*. By analyzing the results of experimentation, we concluded that all considered algorithms may be used to produce summaries. We also note that using principal components or singular values to select the training set may be successfully retained to induce the learning functions of the four studied algorithms. The insertion of ontological knowledge gives qualitative improvements of performance remarkable. This insertion enables to propose good classification functions, which are able to discriminate sentences between important and non-important. The sentences discriminated as important constitute the future summary. Likewise, we observe that ontological knowledge produces more great effects on the classifier quality, if the training set is obtained from singular values rather than from principal components. From this final analysis, we conclude that the two best algorithms that should be applied to automatic summarization by extraction are *Naïve Bayes* and *Support Vector Machine*, from a set of singular values reinforced by ontological knowledge.

As future work, we think that it would be interesting to evaluate the performance of classification algorithms on more reduced spaces, i.e. optimized by means of techniques different to those used in this experimentation. It would be also interesting to explore their behaviour on sets reinforced by ontological knowledge.

5.9 Acknowledgements

The authors would thank Natural Sciences and Engineering Research Council of Canada (NSERC) for its financial support.

References

1. A. Sharan and H. Imran, "Machine Learning Approach for Automatic Document Summarization," *Proceedings of World Academy of Science*, *Engineering and Techno- logy*, 2009, pp. 103-109.
2. R. A. García-Hernandez, R. Montiel, Y. Ledeneva, E. Rendón, A. Gelbukh and R. Cruz, "Text Summarization by Sentence Extraction Using Unsupervised Learning," *Proceedings of the 7th Mexican International Conference on Artificial Intelligence*: *Advances in Artificial Intelligence*, 2008, pp. 133-143.
3. I. Mani and E. Bloedorn, "Machine Learning of Generic and User-Focused-Summarization," *Proceedings of the Tenth Conference on Innovative Applications of Artificial Intelligence*, Menlo Park, 1998, pp. 821-826.
4. J. Goldstein, "Evaluating and generating summaries using normalized probabilities," *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, 1999, pp. 121-128. doi:10.1145/312624.312665
5. K. S. Jones, "Automatic Summarising: The State of Art," *Information Processing and Management*, Vol. 43, No. 6, 2007, pp. 1449-1481. doi:10.1016/j.ipm.2007.03.009
6. R. R. Korfhage, "Information Storage and Retrieval," Wiley, New York, 1997.
7. L. Hennig, W. Umbrath and R. Wetzker, "An Ontology- Based Approach to Text Summarization," *IEEE/WIC /ACM Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology Work- shops*, 2008, pp. 291-294.
8. R. Bellman, "Introduction to Matrix Analysis," McGraw- Hill, New York, 1997.
9. M. Steinbach, "Introduction to Data Mining", Pearson Education, Boston, 2006.
10. M. Ikonomakis, S. Kotsiantis and V. Tampakas, "Text Classification Using Machine Learning Techniques," *Proceedings of the 9th WSEAS International Conference on Computers*, Stevens Point, 2005, pp. 966-974.
11. I. Mani, "Recent Development in Text Summarization," *Proceedings of the Tenth International Conference on Information and Knowledge Management*, McLean, 2001, pp. 529-531.
12. H. Xuexian, "Accuracy Improvement of Automatic Text Classification Based in Feature Transformation and Mul- ti-classifier Combination," *Proceedings of AWCC*'2004, Zhenjiang, 2004, pp. 463-464.
13. G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," *Information Processing and Management*, Vol. 24, No. 5, 1988, pp. 513-523. doi:10.1016/0306-4573(88)90021-0
14. G. H. Golub, "Calculing the Singular Values and Pseudo- Inverse of a Matrix," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 2, No. 2, 1965, pp. 205-224. doi:10.1137/0702016

15. T. A. Lasko, J. G. Bhagwat, K. H. Zou and L. Ohno- Machado, “The Use of Receiver Operating Characteristic Curves in Biomedical Informatics,” *Journal of Biomedi-cal Informatics*, Vol. 38, No. 5, 2005, pp. 404-415. doi:10.1016/j.jbi.2005.02.008
16. A. Saleh, “Reuters Corpus (Offered by Reuters News Agency),” 2004. http://about.reuters.com/researchandstandards/corpus/
17. C. D. Fellbaum, “WordNet (A Lexical Database for Eng-lish),” Princeton University, 1985. http://wordnet.princeton.edu//
18. V. Vapnik, “The Nature of Statistical Learning Theory,” Springer-Verlag, New York, 1995.
19. R. Bellman, “Algorithms, Graphs and Computers”, Aca-demic Press, New York, 1970.
20. F. Rosenblatt, “Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms,” Spartan Books, Washington DC, 1961.
21. R. Rakotomalala, “Tanagra: Un Logiciel Gratuit Pour L’enseignement et la Recherche,” *Proceedings of the EGC*’2005 *Conference*, Amsterdam, 2005, pp. 697-702.
22. G. Holmes, A. Donkin and I. H. Witten, “Weka (A Soft-ware Developed by Machine Learning Group),” University of Waikato, 1994. http://www.cs.waikato.ac.nz/ml/weka/
23. J. Demzar and B. Zupan, “Orange (A Software Devel-oped at Laboratory of Artificial Intelligence),” Faculty of Computer and Information Science, University of Ljubl-jana, 2010. http://orange.biolab.si/

CHAPITRE 6. VENCE: une nouvelle méthode basée sur l'apprentissage automatique renforcé de connaissance ontologique pour extraire des résumés

6.1 Détails de l'article

VENCE: *un nouveau modèle basé sur l'apprentissage automatique renforcé par de la connaissance ontologique pour extraire des résumés*

Jésus Antonio Motta, Laurence Capus et Nicole Tourigny

À soumettre

6.2 Résumé

L'obtention de résumés extractifs en utilisant des fonctions de classification abduites à partir d'un ensemble d'entraînement continue à être un grand défi pour le processus de résumé de textes automatique. Les fonctions de classification obtenues permettent de classer les phrases selon deux catégories, celles importantes qui feront partie du résumé et celles non importantes. Cet article présente le modèle *VENCE*[4] basé sur ce principe. Le pouvoir discriminant des fonctions abduites est amélioré par l'utilisation des relations sémantiques des mots de l'ensemble d'entraînement qui sont extraites d'une ontologie. L'ensemble d'entraînement est également renforcé par l'optimisation de l'espace d'attributs au moyen de techniques statistiques et l'introduction de l'indice Jaccard, qui est calculé à partir des résumés manuels du corpus des documents choisis. Les principes du modèle *VENCE* sont expliqués de manière détaillée ainsi que les différentes expérimentations qui ont mené à l'obtention d'un modèle optimal. Les résultats obtenus montrent une évidente efficacité par rapport aux autres méthodes ou logiciels mis en œuvre pour construire des résumés par extraction.

Mots clé : Index Jaccard, apprentissage automatique semi-supervisé, contenu d'information

[4] VENCE : **V**ectorial **E**nhancing **N**oise free by means of **C**onceptual **E**xtracting

6.3 Abstract

Obtaining extractive summaries by using functions induced from a training set continues to be a great challenge in the domain of the automatic text summary. This paper presents the VENCE[5] model based on this approach and improves the quality of the abduced functions, using semantic relations of the words (attributes) of the training set that are fetched from a ontology to be inserted in this set. The choice of this training set is reinforced with the optimization of the space of attributes by means of statistical techniques, as well as with the introduction of the Jaccard index, calculated from considering a manual summary that is extracted from the corpus of the chosen documents. The VENCE model is explained in details as well as the different experiments conducted to propose an optimal process. Its application to a text document corpus highlighted its efficiency. The results obtained are very satisfactory for the assessment of discriminating power of the abduced classification function as well as for the quality of summaries produced.

Key words: Jaccard index, semi-supervised machine learning, information content

[5] VENCE : **V**ectorial **E**nhancing **N**oise free by means of **C**onceptual **E**xtracting

6.4 Introduction

The importance of the research to develop and improve methods and techniques for the identification of relevant information from text documents, thus production of summaries, continues to increase in direct proportion to the needs of a changing world that is facing the massive amount of information to be handled, for an opportune and efficient decision making. More particularly, the methods for producing automatic summaries can be divided in two great categories: extraction or surface methods and abstraction or deep methods from a more linguistic point of view. A summary produced by extraction is made up of selected sentences of the source documents, using probabilistic and/or heuristic methods. In order to obtain a summary by abstraction, an analysis to interpret the source document is necessary to find new concepts that will replace the initial ones. These methods require a linguistic process at a certain level. The extractive methods are an efficient and robust alternative because they are more independent of language. The task of summarizing one or more documents in an extractive way can be seen like the problem of identifying the parts containing the most relevant information, and so replace the original document or documents, in order to generally produce a quickly synthesis of the text. At the present, the most of the research is achieved with the extraction methods (Chandra *et al.*, 2011; García-Hernandez *et al.*, 2008; Neto *et al.*, 2002; Sanghoon, 2013; Shen et Li, 2011)

A crucial problem in producing a summary by extraction is then the identification of the sentences with the highest content of information of the source document. Different heuristic and probabilistic methods have been used until now, as well as methods that use ontologies to analyze the terms and their relations in the original document. More recently, the use of learning algorithms has emerged out for learning to distinguish the important sentences from those they are not (Motta *et al.*, 2012; Shen et Li, 2011). The idea can be formalized as: let p_i a sentence of the set of sentences P of a document or set of documents and the classes $[C_1, C_2]$, C_1 the class of important sentences and C_2 the class of not important sentences; the task of summarizing consists in assigning a class c_j to the sentence p_i.

To achieve this idea of classification, the document corpus is represented as a workspace, also called the space of attributes, which is reduced and/or optimized to constitute a training

set. Next, a machine learning algorithm uses this training set to abduce a classification function. Once the classification function abduced, it can be applied to new documents in order to produce their summaries. However, in a classification problem, the space of attributes can be entropic, i.e. it can contain a great amount of attributes that are irrelevant, redundant or superfluous. Moreover, these attributes affect a good classification and in some cases they turn the problem intractable under the viewpoint of time and space complexity.

In this paper, we describe the VENCE model, based on this idea of classification in order to produce summaries. In a first time, we used linear transformations to select or to transform the space of attributes to reduce it to a much smaller space, conserving or defining new attributes with high content of information (Motta *et al.*, 2012). The way of learning is semi-supervised: for the induction of the functions, there are labeled and not labeled training sets. The no-labeled set is obtained by identifying the most information carrying sentences by means of linear transformations, and the labeled set is chosen from manual summaries. In order to improve the performance of the classification process, the training set is reinforced with the insertion of ontological knowledge (Motta *et al.*, 2011). The selection of this knowledge is determined by a similarity index that in this case is the Jaccard index. Each attribute of the attribute space is reinforced with a set of hyponyms and meronyms obtained from sub-trees extracted from the WordNet ontology according to the Wu-Palmer similarity measure (Wu et Palmer, 1994). The classification function is next abduced with a machine learning algorithm and ready to be used to produce summaries of new documents.

In this paper, we explain why we chose the Wu-Palmer similarity measure rather than other measures and show that in general all machine learning algorithms can be applied to abduce an efficient classification function with the VENCE model. We also describe an experiment conducted with six algorithms, and the obtained results were similar. But, to continue our research work, we opted for Support Vector Machine (Cortes et Vapnik, 1995; Vapnik, 1999), because among the tested algorithms this one gave us the best results. We used the VENCE model on the DUC2006 (NIST, 2006) corpus in order to produce summaries. The

quality was evaluated with the ROUGE[6] tool kit proposed by Lin (2004), that allows highlighting on a remarkable performance of our model.

The next section, Section 6.5, presents the three phases of the VENCE model. Section 6.6 describes how we optimized this model. Section 6.7 shows its evaluation through its application on the DUC2006 corpus and the discussion of the results obtained. Finally, in section 6.8 we conclude on the performed method and we give some research perspectives.

6.5 The VENCE Model

In this section, we describe the three phases of the VENCE model, which takes in input a document corpus and gives in output an abduction function. This function will next be used to select important sentences in future documents. We note that the document corpus is a set of documents with their abstracts made by human experts. Figure 6.1 presents the three phases elaborated, which are explained in detail in the following of this section.

[6] ROUGE: Recall-Oriented Understudy for Gisting Evaluation

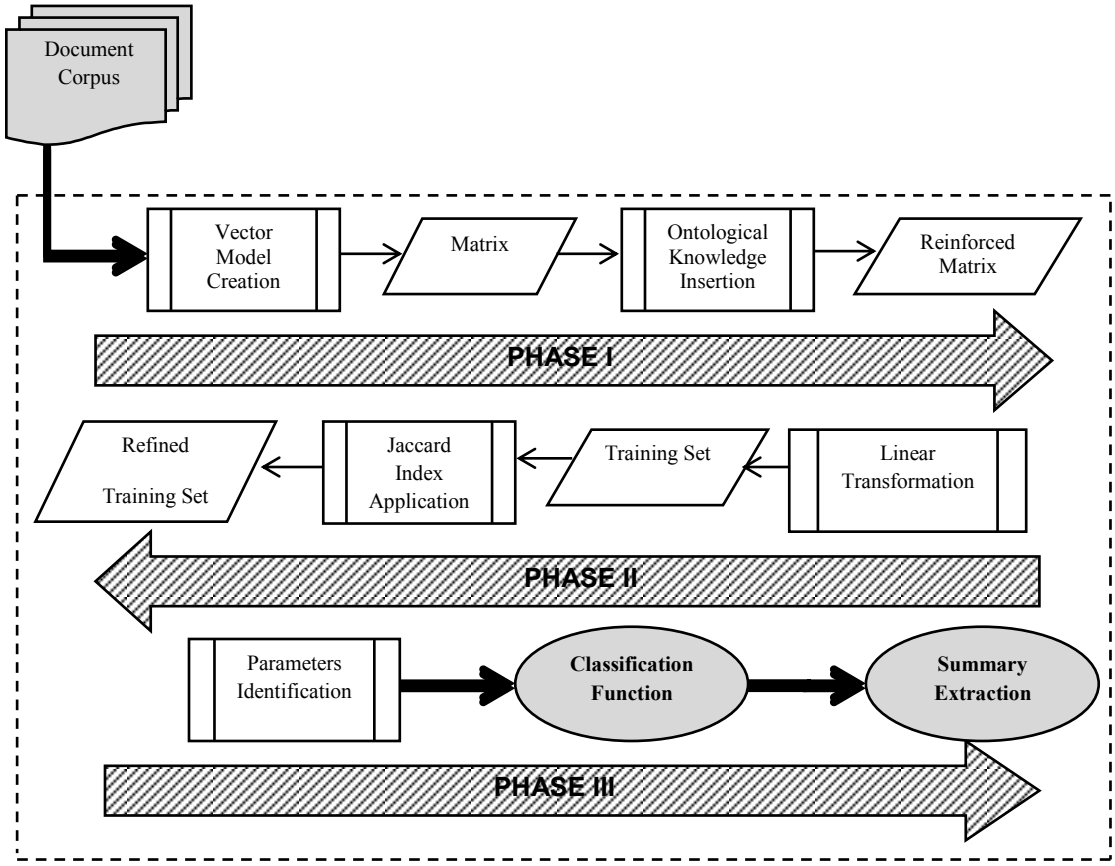


Figure 6.1 *The 3 phases of the VENCE model.*

6.5.1 Phase I: Preparation of the workspace

The first phase consists in the preparing of the workspace on which the different processes are applied. Above all, a *vector model is created*. This workspace is a $tf \times idf$ matrix whose rows are the sentences of the documents and the columns the terms of each of the sentences. Each sentence has undergone a process of tokenization and pruning of stop-words (Figure 6.2).

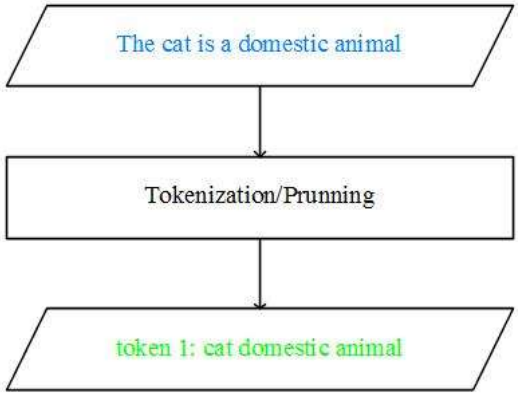


Figure 6.2 *A simple sentence tokenization/pruning example.*

This matrix is next reinforced by *insertion of ontological knowledge*. Into each sentence, we introduce lexemes synonyms, hyponyms and meronyms of each of the singular and plural substantive terms. To complete this process, the algorithm travels through the corresponding sub-trees to extract WordNet ontology terms. Figure 6.3 gives a sub-tree for a term of the sentence.

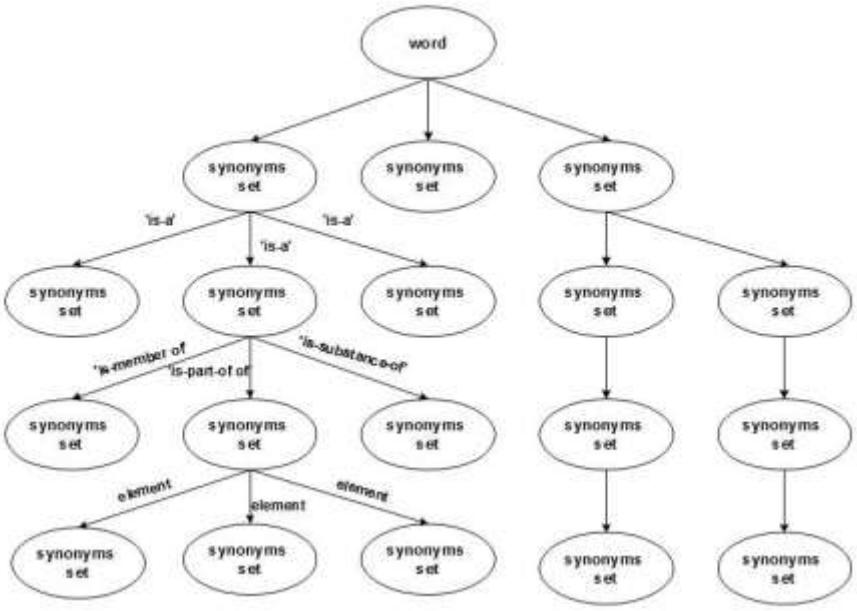


Figure 6.3 *Ontology sub-trees for a word.*

The ontology terms are introduced within the corresponding sentence according to a score calculated by similarity indices. This calculation uses scores based measures, from the works of Resnick (1999), Leacock et al. (1998), Lin (1998), Jiang & Conrath (1997), Wu & Palmer (1994) and Fellbaum (1985).

The steps of the algorithm of ontological knowledge insertion can be seen in Figure 6.4. Each attribute or term of each sentence is read one after the other. For each attribute, a set of synonyms is extracted from the WordNet ontology. These synonyms are added to the sentence to reinforce it according to the similarity measure between the attribute and them. The synonyms with the relation "is-a", if they exist, are firstly examined. Next, the test continues for the synonyms with the relation "is part-of", "is-member-of" and "is-substance-of", if they exist.

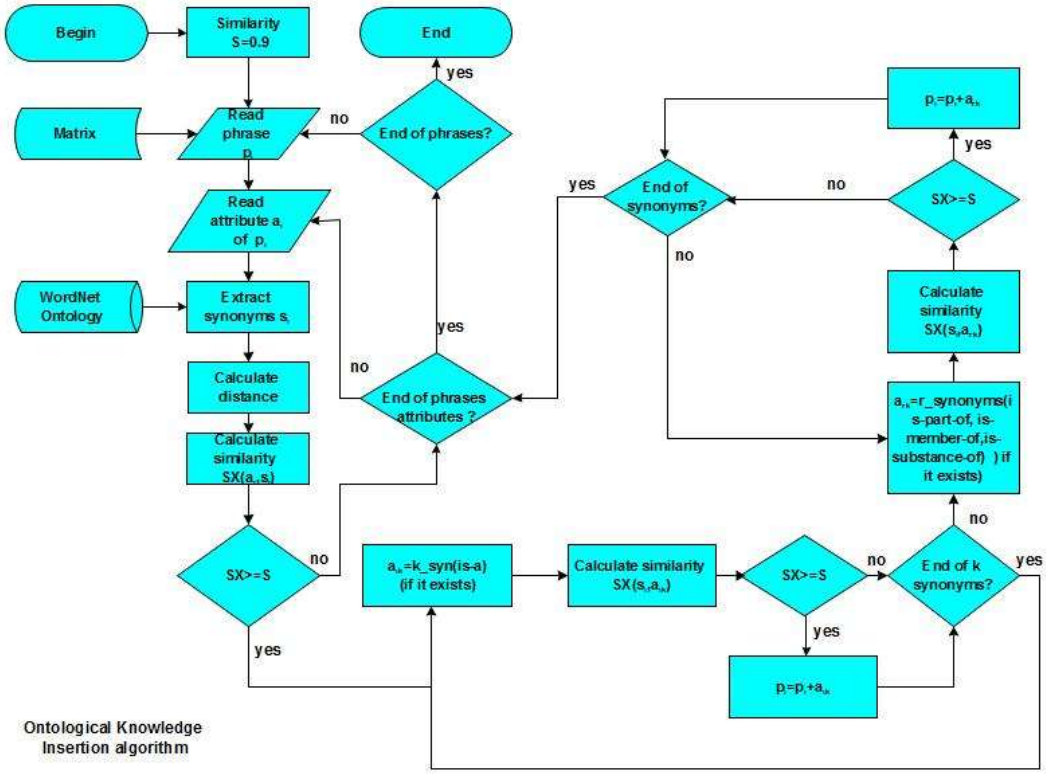


Figure 6.4 *Ontological knowledge insertion algorithm.*

To evaluate the similarity between the attributes and their lexemes, it is important to choose an appropriate measure. To determine the one it is better to use, we made a comparison between different measures which allow the evaluation of semantic distance between two terms. From this comparison, we decided to retain the Wu-Palmer measure (Wu et Palmer, 1994), because its great capacity for information extraction in this model. Section 6.5.1 describes how we compared the different similarity measures and how to make our choice.

6.5.2 Phase II: Getting the training set

Once the ontological knowledge is inserted, *linear transformations* were applied to the reinforced matrix using optimization methods. The workspace was optimized by reducing its size while preserving all information. By applying such methods, one obtains a training set, which will be later usable in the abduction process. In our context, linear transformations allowed us to get a sentences subset representative of the whole space of attributes (Motta *et al.*, 2012). For linear transformations, we selected two mathematical procedures: Principal Components Analysis (PCA) (Golub et Van Loan, 1996; Pearson, 1901) and Singular Vector Decomposition (SVD) (Golub et Van Loan, 1996; Stewart, 1993). The PCA procedure is part of the space attributes transforming methods, also called extraction methods. From the initial variable space, this application creates a subspace of much smaller variables and high-variance (information), such that there is a high correlation among the variables of the new subspace but no correlation with the original variables (Jolliffe, 2002). Unlike previous one, the SVD procedure belongs to the variable selection methods (Golub et Kahan, 1965), and are based on the spectral theorem (Halmos, 1963; Helson, 1986; Hilbert *et al.*, 1927; Reed et Simon, 1975) . One obtains a decomposition of the initial space (initial matrix) based in eigenvector matrices and a diagonal matrix of eigenvalues, providing this manner a shortened version of the initial space with high information content. As one can see in Section 6.3, we showed that the results are similar whatever the optimization method used.

At the end of this phase, we obtained a training set usable to the abduction process. But, first results showed that it was not satisfying. So, we decided to add another phase of optimization. This second phase of optimization consists in applying the *Jaccard index* (Jaccard, 1901), commonly used to compare similarity between samples in research. This index is calculated as follows:

$$J(A,B) = \frac{A \cdot B}{|A|^2 \cdot |B|^2 - A \cdot B} \tag{6.1}$$

where A and B are the samples to be compared. The J value is between 0 and 1. We use it to measure the degree of similarity between the sentences of the training set and those of

manual summaries of the input document corpus. More precisely, A is the manual summary and B is a sentence of the training set. This phase aims to refine the training set, by selecting only the sentences that are most similar to those of manual summaries.

Now, we obtain two subsets of sentences. The sentences with the highest index are identified as 'important' and those with the lower index are labelled as 'not important'. Figure 6.5 resumes how the document corpus is divided into these two sets of sentences.

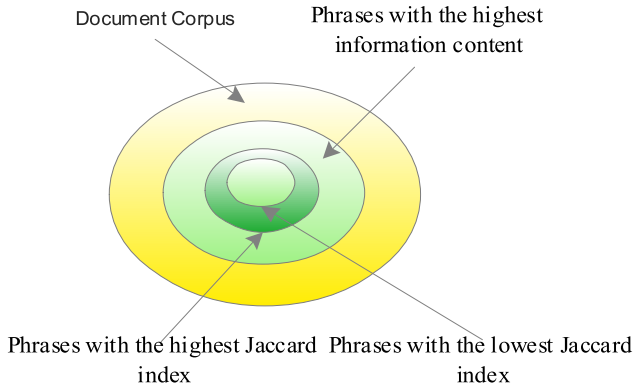


Figure 6.5 *Different sets of sentences.*

From these subsets, we built a satisfying training set that is a binary set made by identifying the two categories of sentences: 'important' and 'not important' sentences. We set a threshold to extract the upper and lower values, while taking care to keep a balance between the two classes of sentences. This training set was next used to abduce the classification function.

6.5.3 Phase III: Abduction of the classification function

The training set produced enables the last phase of the VENCE model, which consists in identifying the *parameters* to obtain the abduction function. It will be next used to identify important sentences, i.e. a summary, of one document or a set of documents that have not been seen yet. The literature gives us several Machine Learning (ML) algorithms to identify the abduction function. During this research work, we applied six algorithms well known in ML and data mining domains in order to compare the discriminating capacity of the classification functions obtained. These results are described in Section 6.6.2 and allowed us to decide what algorithm is more appropriate for the VENCE model. So, we opted for the algorithm SVM (Support Vector Machines) because its best performance.

Thereby, we presented the way of analyzing a corpus of documents and their corresponding abstracts, introducing reduction techniques on the attribute space as well as an unsupervised learning process. We showed how to find a learning function that will be used in identifying in a document or set of documents the important sentences that will compose the corresponding summary.

6.6 Optimization of the VENCE model

To provide us with the performance of the model, we conducted related experiments. Indeed, as we have already mentioned, we evaluated different similarity measures in order to choose the most appropriate one, the Wu-Palmer measure in this case. This evaluation is described in the following section 6.6.1 In the same way, we compared different machine learning algorithms used to abduce the classification function in order to choose the better one (See Section 6.6.2).

6.6.1 How to choose the most appropriate similarity measure

Before selecting the most appropriate similarity measure, we evaluated the impact of the information content by insertion of the different sub-trees of the ontology as well as different combinations of them. We used the works of Wu and Palmer (Wu et Palmer, 1994) to identify the differences in the variances of attributes achieved with the introduction of ontological knowledge. We constructed the following tables and graphs:

- Singular values plot vs. Variance: attributes space without ontological knowledge and attributes space with ontological knowledge;
- Principal components vs. Variance: attributes space without ontological knowledge and attributes space with ontological knowledge;
- Cumulative variance graphs of variance with and without ontological knowledge;
- Comparative table of different information content similarity scores.

The graphics performance was studied to decide if the values were acceptable or otherwise changes were needed in order to obtain the parameters of a better training set from the insertion of ontological knowledge. Figures 6.6, 6.7, 6.8 and 6.9 show the obtained change in the content of variance by inserting ontological knowledge. More precisely, Figure 6.6 presents the variance of the principal components without insertion of ontological knowledge and with this insertion. We can then see the significant impact of this insertion. Figure 6.7 shows the cumulative variance. Note that more there is variance in content model attributes, the more relevant it is (Motta *et al.*, 2011). Observing the behavior of the graphs shown, we can conclude that the impact on the first 200 components is quite noticeable.

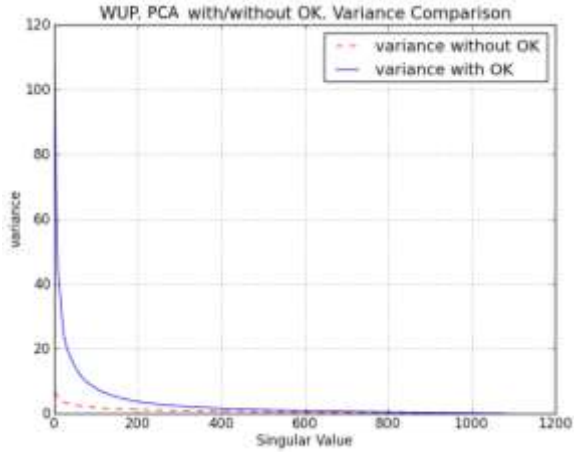


Figure 6.6 *Principal Components Variance.*

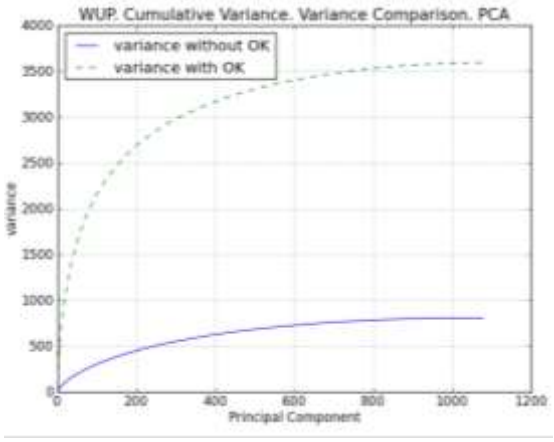


Figure 6.7 *Principal Components Cumulative Variance.*

In a same way, Figure 6.8 shows the variation of the variance of the singular values with insertion of ontological knowledge. Figure 6.9 presents also the cumulative variance corresponding. From the observation of these graphs, we can conclude that the insertion of ontological knowledge produces significant effect on the first 400 singular values. It turns

considerably higher compared to the principal components. Consequently, the application of SVD on the attribute space enables to identify more elements carriers of information (more information, more variance) than those achieved with the application of the PCA method.

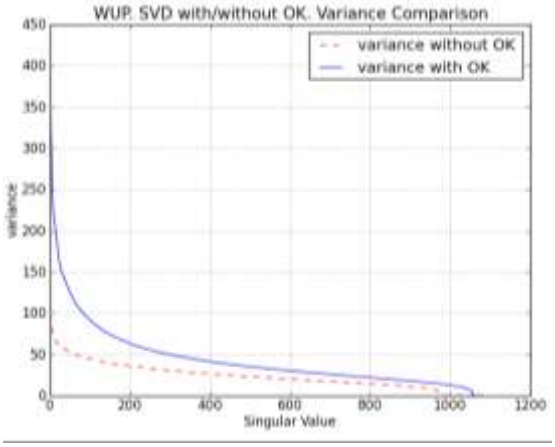


Figure 6.8 *Singular Values Variance.*

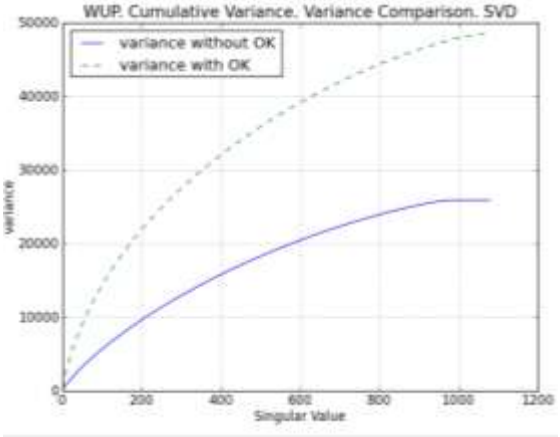


Figure 6.9 *Singular Values Cumulative Variance.*

Whether the PCA or the SVD method is used, the insertion of ontological knowledge is significant to identify the subset of sentences. With such a result, we decided to compare the different similarity measures in research works in order to select the most appropriate for our model. We then identified six measures usable for our model, which were Path (Fellbaum, 1985), Leacock-Chodorow (Leacock *et al.*, 1998), Wu-Palmer (Wu et Palmer, 1994), Resnick (Resnick, 2000), Jiang-Conrath (Jiang et Conrath, 1997), and Lin (Lin, 1998), and compared their impact on the information content.

Table 6.1 presents the different similarity scores obtained for the insertion of ontological knowledge, the amount of lemmas and the information content for each of the synsets synonyms, hyponyms of synonyms and hyponyms plus meronyms, as well as improvement relative of the score with the insertion of meronyms, in relation to the amount of information. For example, the improvement in the similarity according Jiang-Conrath (1997) is obtained with the ratio 0.0156742/0.0001902, which produces a value of 82 (times).

Table 6.1 *Lemma Information Content for different scores.*

Score	Synonyms		Hyponyms		Hyponyms +Meronyms		Improvement
	Lemma/ Sentence	*IC/ Lemma*	*Lemma/ Sentence*	*IC/ Lemma*	*Lemma/ Sentence*	*IC/ Lemma*	
Path	87	0.00030131	387	0.0182471	427	0.0183670	60
Leacock-Chodorow	13	0.0002393	33	0.01685180	36	0.0168550	70
Wu-Palmer	87	0.00030131	387	0.01824690	427	0.0183688	60
Resnick	33	0.0006010	115	0.01708576	126	0.0172401	28
Jiang-Conrath	38	0.0001902	100	0.0156134	106	0.0156742	82
Lin	27	0.0006993	83	0.0168285	89	0.0172140	24

From this table, we conclude that, although the Jiang-Conrath similarity presents the best relative improvement inserting meronyms term, the Wu-Palmer similarity gives greater information content in average. It is for this reason that this similarity measure was chosen for the ontological knowledge insertion. It is obtained with the following formula:

$$WUP = \frac{2 \times depth(lcs)}{depth(syn1) + depth(syn2)} \tag{6.2}$$

where *depth (sync1)* and *depth (sync2)* are the depths of the two synsets to be compared and *depth (lcs)* is the depth of the nearest common ancestor (*Least Common Subsumer*).

Thus, we consider that the similarity measure established by Wu-Palmer between the word/attribute and the corresponding synsets of the ontology is the best measure in our research, because the Wu-Palmer similarity measure allows us to decide if the insertion of ontological knowledge is sufficient or not.

6.6.2 How to evaluate the best algorithm to abduce the classification function

For this phase of the VENCE model, we could use different ML algorithms. To complete our model and make it more optimized, we conducted another experiment to compare the results of six ML and data mining well-known techniques: Support Vector Machine (Cortes et Vapnik, 1995), Logistic Regression (Agresti, 2007), Random Tree (Breiman, 2001), Naïve Bayes (Hastie *et al.*, 2009), MultiLayer Perceptron (Rosenblatt, 1957), and Radial Basis Function Neural Networks (Buhmann, 2003). With such an experiment, we can suggest a more appropriate technique, which will allow to get a classification function with

the better discrimination ability. We describe briefly these six techniques and how they were applied. Next, we expose the experiment and the results obtained.

Support Vector Machine (SVM) can use a linear or nonlinear kernel (Cortes et Vapnik, 1995; Vapnik, 1999). In our research, we used a linear kernel, based on separate instances of binary values in the hyperspace (it is possible to have more than two classes) using a maximum margin function as shown on Figure 6.10. In the context of our research, the kernel linear complexity is equal to 1, equal to tolerance precision and normalization of attributes 0001.

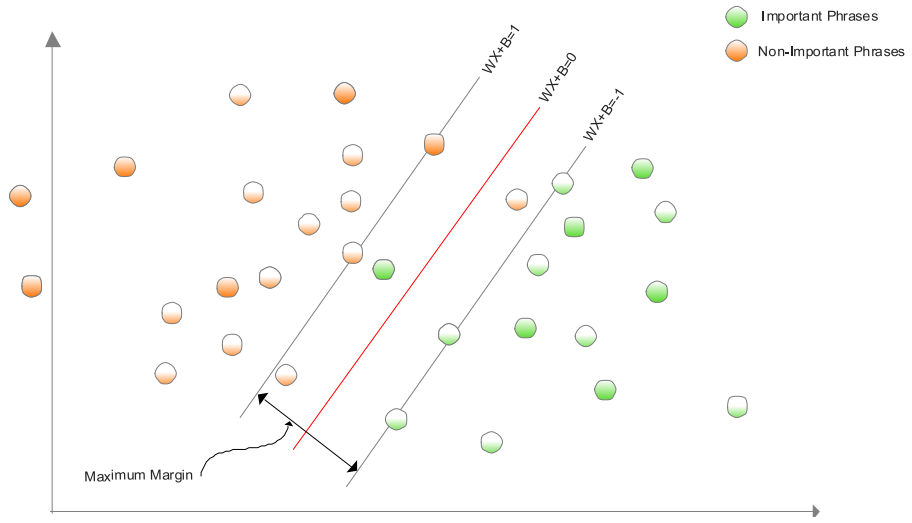


Figure 6.10 *SVM Classification.*

The Logistic Regression (LR) technique is based on a probabilistic regression model whose dependent variable Y discriminates classes based in the binary *logit* function and a set of independent variables $x_1, x_2, \ldots, x_k$ (Agresti, 2007). Random Tree (RT) is a tree induction method from samples taken from the training set with replacement, also called bagging (Breiman, 2001). For our research, the number of attributes split was 20. Naïve Bayes (NB) is a classification technique built from the Bayes' theorem, which provides a way of calculating the probability of a hypothesis based on its prior probability (Hastie *et al.*, 2009). We applied the Laplace probability estimate. Multilayer Perceptron is a universal classifier with hidden layers that interact with the input and output layer of the network to

discriminate elements of a state space that are not linearly separable (Rosenblatt, 1957; Rumelhart *et al.*, 1986). An example is given by Figure 6.11. We did several tests. Finally, we used 10 neurons and two hidden layers with a learning rate of 15%. The proportion of the validation set was of 20%. The maximum number of iterations was 100 and the threshold error rate 0.1%.

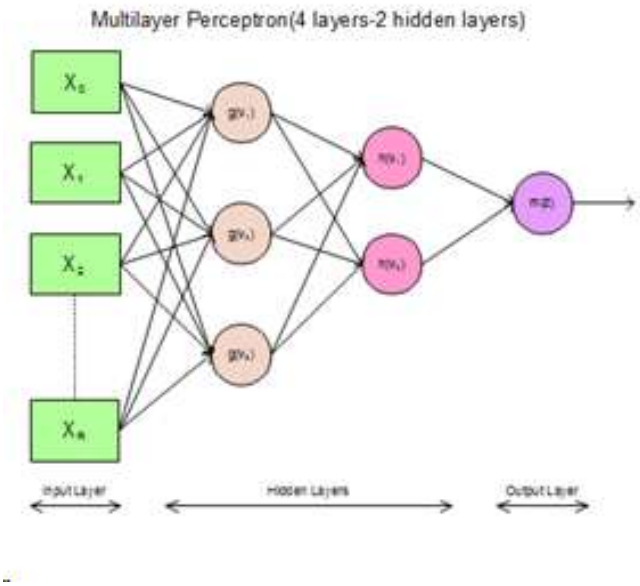


Figure 6.11 *Multilayer Perceptron.*

The last technique, Radial Basis Function Neural Networks (RBFNN), is also a universal classifier consisting of three layers: input layer, output layer and hidden layer (Buhmann, 2003). The hidden layer performs local nonlinear transformations of the input data or local data. In the output layer are produced linear combinations of hidden layers activations, which are the output of network, as shown by Figure 6.12. For our context, the learning rate was of 15%, the proportion of test sample 20%. The maximum number of iterations was 100 and the threshold error rate 0.1%.

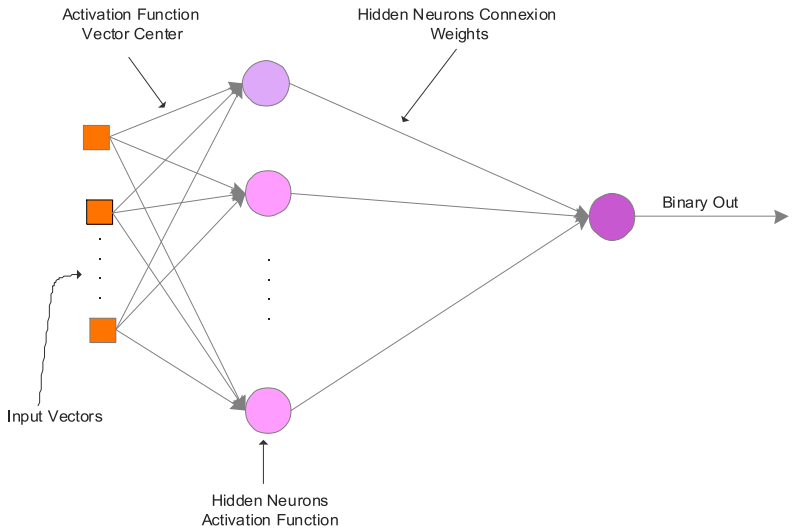


Figure 6.12 *Radial Basis Function Neural Network.*

To evaluate the discrimination ability of the classification function, we used the usual metrics *Precision*, *Recall* and *F_measure* or *F-score*. The *Precision* metric enables to examine the ability of the classification function to identify important sentences when they are really important. The value of *Recall* can be seen as the proportion of sentences correctly classified by the function. The *F_measure* corresponds to the harmonic mean of the two previous values. Detailed definitions of these metrics can be found in literature related to the retrieval information domain. The values used are taken from confusion matrices created for important and not important classes, from the identification of four sets of sentences: true positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) as shown on Table 6.2. This matrix serves in turn to demonstrate further the performance of the classification function.

Table 6.2 *Confusion Matrix.*

Class	**Predicted Class**	
	Important	***Not important***
Important	True Positive Cases	False Positive Cases
Not important	False Negative Cases	True Negative Cases

To evaluate the quality of this discrimination we built *ROC (Receiver Operating Characteristic) Curves* (Fawcett, 2006). These curves are obtained by calculating the *sensitivity* (true positive rate) versus *specificity* (1-false positive rate), showing a continuous variation of observed points. From the observation of several curves, we obtain a qualitative comparison, knowing that curve upward and to the left has the largest accuracy. Additionally obtaining the area under the curve (AUC) indicates the probability of success of the function to identify a sentence that is important. The metrics mentioned and the false positive rate (FPR), are obtained as follows:

$$sensitivity = TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{6.3}$$

$$specificity = TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR \tag{6.4}$$

$$FPR = 1 - specificity \tag{6.4.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{6.5}$$

The selection of samples for the abduction of the classification functions of the different methods was analyzed using the sub-sampling technique choosing 2/3 of the training set for this effect and the remaining third for the corresponding test. To obtain different performance metrics, we used the programs Tanagra (Rakotomalala, 2005), Orange (Demzar et Zupan, 2010) and Weka (Holmes *et al.*, 1994). In a first time, we analyze the predictions obtained with the metrics Recall, Precision and F_measure as well as confusion matrices for each of the machine learning methods applied according to the attribute space optimization technique used: PCA and SVD. In a second time and to get another element of decision, we study the ROC curves built for each one of the analyzed methods.

Table 6.3 gives the predictions and confusion matrices for the PCA method. The analysis of these results allows the conclusion that generally all methods give an important performance. So, in principle, they could be used in conjunction with the PCA method to abduce classification functions for extracting automatically summaries. It is worth

mentioning that, with SVM, LR, RT and RBFNN methods, the F_measure values are equal to 1, setting β=1 (importance) in the original formula,

$$F_{\beta} = \frac{\left(1+\beta^2\right)\times\left(precision\times recall\right)}{\beta^2 \times precision + recall} \tag{6.6}$$

Table 6.3 *Prediction and Confusion Matrices for Principal Components Analysis method.*

		Confusion Matrix		***Predictions***		
ML Method	***Class***	**Imp**	**Not Imp**	**Recall**	**Precision**	**F_measure**
SVM	Imp	50	0	1	1	1
	Not imp	0	59	1	1	1
NB	Imp	42	8	0.840	1	0.9130
	Not imp	0	59	1	0.8806	0.9365
LR	Imp	50	0	1	1	1
	Not imp	0	9	1	1	1
RT	Imp	50	0	1	1	1
	Not imp	0	59	1	1	1
MLP	Imp	44	6	0.9891	0.9167	0.8979
	Not imp	4	55	1	0.9016	0.9166
RBF-NN	Imp	50	0	1	1	1
	Not imp	0	59	1	1	1

Meanwhile, Table 6.4 provides predictions and confusion matrices for the SVD method. We infer as in the previous case, all methods could be used for discrimination of significant sentences in order to extract automatically summaries. All methods analyzed get a F_measure value of 1 except for RT algorithm that anyway presents values higher than 95%. Importantly, these values based on the SVD method are superior to those found with the previous method PCA, suggesting that, for this particular context, it should be preferred.

Table 6.4 *Predictions and Confusion Matrices for Singular Value Decomposition method.*

		Confusion Matrix		***Predictions***		
ML Method	***Class***	**Imp.**	**Not Imp.**	**Recall**	**Precision**	**F_measure**
SVM	Important	48	0	1	1	1
	Not imp.	0	51	1	1	1
NB	Important	54	0	1	1	1
	Not imp	0	45	1	1	1
LR	Important	54	0	1	1	1
	Not imp	0	45	1	1	1
RT	Important	46	4	0.9200	1	0.9583
	Not imp	0	49	1	0.9245	0.9607
MLP	Important	54	0	1	1	1
	Not imp	0	45	1	1	1
RBF-NN	Important	48	0	1	1	1
	Not imp	0	51	1	1	1

The two following figures, 6.12 and 6.13, show the ROC Curves for the two methods of attribute space optimization, PCA and SVD. In Figure 6.12, we see that all the ML methods, applied on a space optimized by identifying its principal components, could be used for extracting the most important sentences of document corpus. The SVM and RT methods are noteworthy with an area under the curve (AUC) around 90% and 85% respectively.

In the case of Figure 13, we highlight first all AUC values found for the methods applied on the selected attribute space, i.e. with the SVD method, are superior to those found with the PCA method. It is also important to note also that all the methods have values above 80%, and a maximum value is found above 95% for the SVM algorithm.

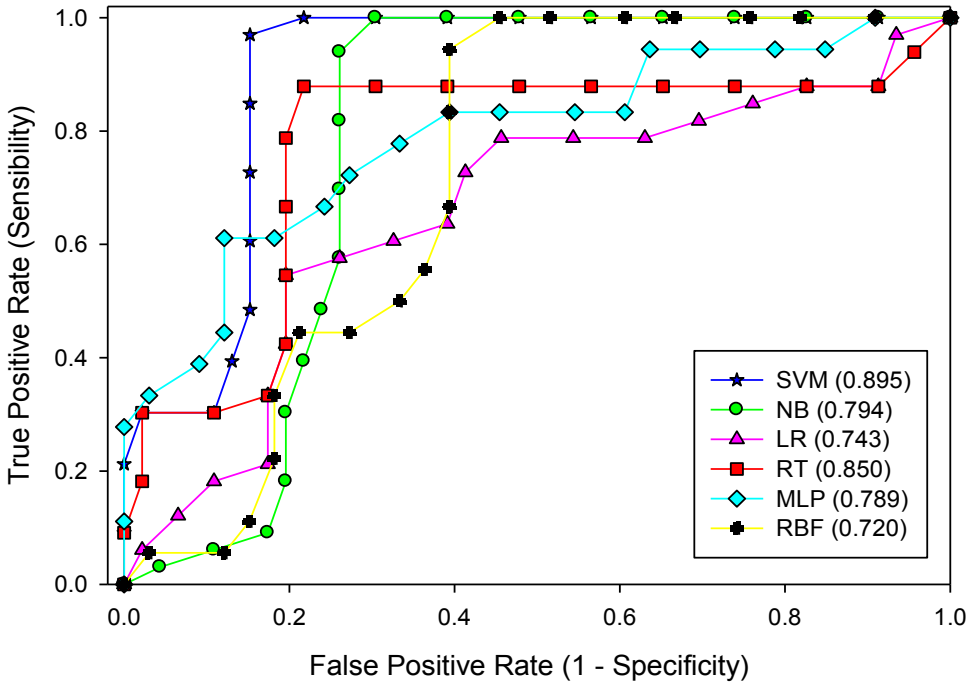


Figure 6.12 *ROC Curves with PC method.*

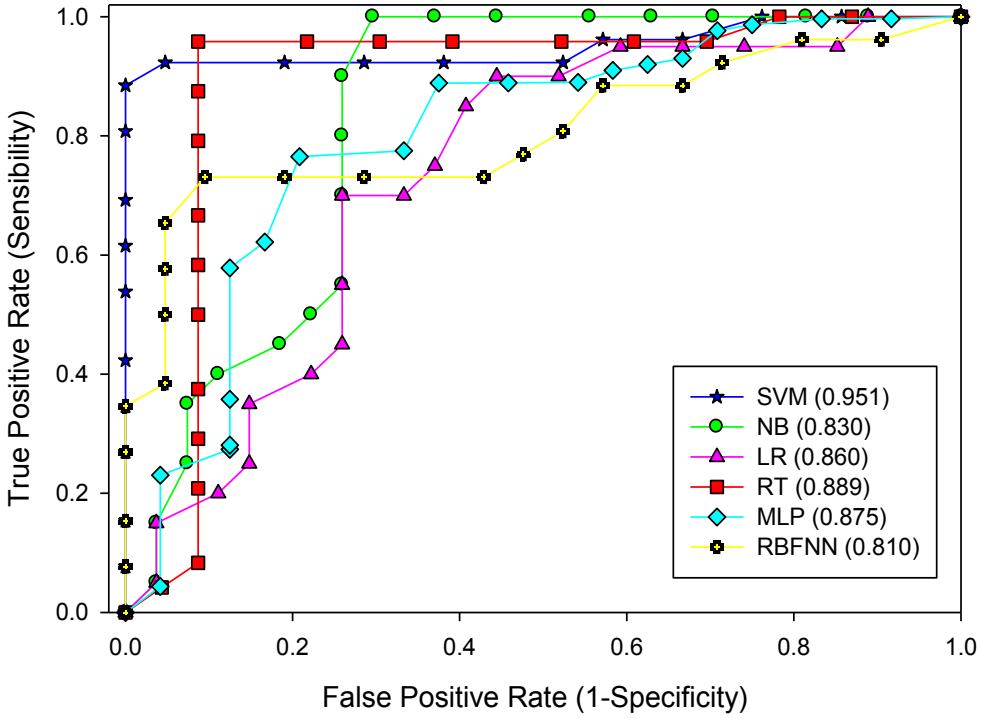


Figure 6.13 *ROC Curves with SVD method.*

Table 6.5 shows the AUC values obtained for each of the ML methods studied according to space optimization used: PCA or SVD.

Table 6.5 *AUC values for the PCA and SVD methods.*

ML Method	*Optimization method*	
	PCA	**SVD**
SVM	0.895	0.951
NB	0.794	0.830
LR	0.743	0.860
RT	0.850	0.889
MLP	0.789	0.875
RBF-NN	0.720	0.810

By having the highest values in both the F_measure as its AUC, the SVM method was chosen to complete our VENCE model. This method obtained the best discriminating power and can be now applied to other documents, in order to identify the important sentences that will be candidates to construct the document's summary.

An interface was programmed to facilitate the use of the VENCE model and its evaluation. To choose the corpus to be summarized among a list of documents, the interface offers three ways of doing: a percentage of this list, a sub-list of documents or a set of documents chosen by the user. The size of the future extract must be also validated through this interface by indicating the proportion of important sentences to be preserved. The different processes of the VENCE model allow the classification of all the sentences of the chosen corpus according to their degree of importance. The interface displays the *n* first sentences, *n* being the chosen size.

6.7 Evaluation of the VENCE model

To conduct experiments and then evaluate the VENCE model, we used the set of sentences of DUC2006. This collection consists of 1250 documents grouped into 50 topics. Each document has a summary performed manually by human experts. We applied the VENCE model to abduce classification functions and also produce new summaries. Now, we present the results of these experiments in order to show the efficiency of the VENCE model.

To assess the quality of produced summaries, we use the specialized toolkit ROUGE (Recall-Oriented Understudy for Gisting Evaluation) 1.5.5, elaborated by Lin (2004). This program has as entries automatically produced summary and abstracts of documents made manually by human specialists. The ROUGE metrics used to measure the quality of the abstract are: Precision, Recall and F_measure. In our model, if the F_measure is not near than 70%, we revise the training set and go back to the step of ontological knowledge insertion. This return aims to modify the sub-tree extracted from the ontology in terms of the type and value of similarity index and the depth of the extracted sub-tree. The process continues in this manner until the quality value specified for summary extracted by our model. More precisely, with the ROUGE toolkit, we used ROUGE-L, ROUGE-SU and ROUGE-N with a confidence interval of 95%.

6.7.1 Results obtained with the ROUGE assessment tool

According to the previous sections, the ML method applied to abduce the classification function is SVM, which reached an AUC of 95.1% with 0% errors in the selection of sentences considered important (% error in selecting cases TP). This method is applied on the DUC2006 corpus of documents on climate change containing 1080 sentences in order to be classified as *important* (which will be part of the summary) and not important sentences (which will not be part of the summary). From our point of view, the produced summaries matched fairly closely with the summaries manuals on the subject written by human specialists for DUC2006. To confirm our intuition, we computed the performance of the classifier function through the Precision, Recall and F_measure metrics, and also the values obtained with ROUGE.

The inputs to the program are the summary produced by the machine and the abstracts from experts on the subject. Table 6.6 shows the values of Recall, Precision and F_measure obtained by applying different methods of ROUGE assessment tool for automatic summaries: ROUGE-L (the longest common subsequence - LCS), ROUGE-SU (skip-bigram plus unigram), ROUGE-1 (1-gram), ROUGE-2 (2-gram) and ROUGE-3 (3-gram). As we can see, the performance measurement values reported by ROUGE-L and ROUGE-1 show a fairly high values of performance that are well above the average obtained on other

work on the topic: a Recall close to 84% and an approximate Precision of 59% to produce a balance in performance exhibited by a F_measure approximately of 70%.

Table 6.6 *ROUGE metrics for the VENCE Model.*

ROUGE Method	***Metrics***		
	Recall	**Precision**	**F_measure**
ROUGE-L	0.838	0.581	0.686
ROUGE-SU	0.619	0.429	0.507
ROUGE-1	0.847	0.587	0.694
ROUGE-2	0.513	0.355	0.420
ROUGE-3	0.372	0.258	0.305

These values mean that the summary produced by the VENCE model is very similar to the summaries written by experts, which in our view represents a major advance in the use of machine learning approach to extract summaries.

6.7.2 Comparison with other research works

We also compare the results obtained with the VENCE model to those of other research works. Such results are not evident to find but three works retained our attention and allowed an interesting comparison.

The work of Shen and Li (2011), that we call W1, presents how to use the SVM ranking to train the feature weights for query-focused multi-document summarization process. The authors applied a supervised learning method to extract sentences from multi-documents, by deriving the labels of sentences for training corpus from an existing human labeling. In this study, the authors tested SVM-CSL models (Ranking SVM with Cost Sensitive Loss), SVM and SVR (Support Vector Regression) which are evaluated with ROUGE. They obtained the best performing for SVM-CSL model with a Recall value for ROUGE-1 of 0.4221 that represents only 49.83% of the result of VENCE with ROUGE-1, which is 0.8470 (See Table 6.7).

Table 6.7 *Comparison between the VENCE model and models proposed by W1.*

	VENCE	***Models W1***		
		SVM-CSL	**SVM**	**SVR**
ROUGE-SU	0.6190	0.1542	0.1533	0.1517
ROUGE-1	0.8470	0.4221	0.4215	0.4166
ROUGE-2	0.5130	0.0994	0.0983	0.0952

The work of García-Hernandez et al., (2008) referred as W2 in Table 6.9, proposes an automatic text summarization approach independent of language and domain by sentence extraction using an unsupervised learning algorithm. The authors' hypothesis is that an unsupervised algorithm can help for clustering similar ideas (sentences). Then, for composing the summary, the most representative sentence is selected from each cluster. The results of the work W2 correspond to the values from ROUGE-1 to ROUGE-10 and are very similar (around 0.47). But, we did not obtain the values from ROUGE-10 for the VENCE model. Then, to compare the model W2 to our model, we calculated the value of the F_measure by using an arithmetic mean of the metrics ROUGE-L, ROUGE-SU and ROUGE-1. Table 6.8 gives these results. The F_measure value of ROUGE-10 is 0.47906 for the W2 model, which is 31.15% lower than the 0.6290 value of the VENCE model.

Table 6.8 *Comparison between the VENCE model and the model W2.*

	VENCE	***Model W2***
Recall	0.76800	0.48155
Precision	0.53230	0.47633
F_measure	0.62900	0.47906

The research work, named W3 in Table 6.9, (Larocca *et al.*, 2002) presents a summarization procedure based on the application of trainable machine learning algorithms. These algorithms employ a set of features extracted directly from the original text. These features are of two kinds: statistical, based on the frequency of some elements in the text, and linguistic, extracted from a simplified argumentative structure of the text. Table 6.10 shows the Recall values obtained for the models in this research, C4.5 and Bayes, and the average Recall value of the VENCE model. It can be seen that the value obtained for the VENCE model is 2.21 times higher compared to C4.5 method and 49.33% higher than the Bayes model.

Table 6.9 *Comparison between the VENCE model and the models W3.*

	VENCE	*Models W3*	
		C4.5	**Bayes**
Recall	0.76800	0.3468	0.5143

Consequently, with the VENCE model, we obtained better results than those obtained by important similar research works.

6.7.3 Comparison with other automatic summarizers

To evaluate the VENCE model, we also decided to compare the results obtained with the results of other automatic summarizers. We used a free summarizer because of its accessibility: FreeSummarizer (Xmarks, 2012). It was easier to make several tests with this tool. We also used a commercial summarizer, the Copernic™ Inc. company's one (Bouchard et Bouchard, 1996).

In Table 6.10, we can observe the F_measure values for ROUGE-L, ROUGE-SU, ROUGE-1, ROUGE-2 and ROUGE-3, applied on summaries produced with FreeSummarizer and Copernic Summarizer, as well as those produced with the VENCE model. This table also shows the percentage of superiority of our model against the other two models. As can be seen, these values are far superior from a low value of 33.87% to reach values up to 250% (relative to FreeSummarizer).

Table 6.10 *Comparison with others automatic summarizers.*

	FreeSum.	*Sup. (%)*	*Copernic*	*Sup. (%)*	*VENCE*
ROUGE-L	0.4523	51.66%	0.5089	34.80%	0.6860
ROUGE-SU	0.2905	74.52%	0.3491	45.23%	0.5070
ROUGE-1	0.4633	49.79%	0.5184	33.87%	0.6940
ROUGE-2	0.2110	99.05%	0.2680	56.71%	0.4200
ROUGE-3	0.1217	250.00%	0.1749	74.38%	0.3050

As described, the VENCE model was evaluated by using five different ways:

- use of the metrics Recall, Precision and F_measure to the abduced classification function (learning function);
- construction of the ROC curves and areas under the curves (AUC) of the classification function;

- application of the ROUGE (L, SU, 1, 2, 3) metric on produced summaries;
- comparison of the performance with the results of other related research works;
- comparison of the performance of produced summaries with those obtained with FreeSummarizer® and Copernic® automatic summarizers.

As seen in all these evaluations, all values are pretty good, starting with the evaluation of the mathematical model (SVM) having an AUC value of 0.951, until the ROUGE calculated values and comparisons with other models that confirm this initial result and guarantees a remarkable performance for the VENCE model.

6.8 Conclusion

In this paper, we presented the VENCE model based on machine learning for extracting automatically summaries. The training set for the learning function is strengthened with the introduction of ontological knowledge of high information content and its similarity to abstractive summaries made by specialists. The training set was obtained primarily from spaces of variables that have been optimized and reduced by PCA and SVD methods. In this sense, the VENCE model fits the paradigm of semi-supervised learning.

We evaluated the performance of the VENCE model with six ML algorithms by using the Recall, Precision and F_measure metrics and the quality of the classification by means of ROC curves on the two types of spaces (PCA, SVD). The result of this evaluation showed that in general all machine learning algorithms perform well on SVD optimized space. However, we retain the SVM algorithm, which reached the first place.

We applied the VENCE model on the DUC2006 corpus in order to produce automatically summaries. We used the ROUGE 1.5.5 tool to assess their quality. Given the analysis of the scores reported by this tool, we can conclude that the VENCE model can be used to obtain extractive summaries ensuring high quality and performance.

References

1. Agresti, A. (2007). *An Introduction to Categorical Data Analysis. Building and Applying Logistic Regression Models.* Hoboken, New Jersey: John Wiley & Sons.

2. Bouchard, M. & E. Bouchard. (1996). "Copernic Summarizer". http://www.copernic.com/.

3. Breiman, L. (2001). "Random Forests." *Machine Learning* 45, no. 1 5-32.

4. Buhmann, M. D. (2003). *Radial Basis Functions: Theory and Implementations.* United Kingdom: Cambridge University Press.

5. Chandra, M., V. Gupta & S. K. Paul. (2011). "*A Statistical Approach for Automatic Text Summarization by Extraction "* In *International Conference on Communication Systems and Network Technologies (CSNT)*, 268-271. Kattra, India: IEEEXplore.

6. Cortes, C. & V. Vapnik. (1995). *Support-Vector Networks*. Holmdel, NJ, USA.: AT&T Bell Labs.

7. Demzar, J. & B. Zupan. (2010). "Orange (a Software Developed at Laboratopry of Artificial Intelligence)". http://orange.biolab.si/.

8. Fawcett, T. (2006). "An Introduction to Roc Analysis." *Pattern Recognition Letters* 27, no. 2006 861-874.

9. Fellbaum, C.D. (1985). "Wordnet (a Lexical Database for English)", Princeton University. http://wordnet.princeton.edu.

10. García-Hernandez, R. A, R. Montiel, Y. Ledeneva, E. Rendón, A. Gelbuck & R. Cruz. (2008). "*Text Summarization by Sentence Extraction Using Unsupervised Learning "* In *Mexican International Conference on Artificial Intelligence. MICAI 2008*. Ciudad de Mexico: Springer-Verlag.

11. Golub, G. H. & W. Kahan. (1965). "Calculing the Singular Values and Pseudo-Inverse of a Matrix." *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis* 2, no. 2 205-224.

12. Golub, G. H. & C. F. Van Loan. (1996). *Matrix Computations.* 3 ed. Baltimore, Mariland, USA: Johns Hopkins University Press.

13. Halmos, P. R. . (1963). "What Does the Spectral Theorem Say?" *The American Mathematical Monthly - Mathematical Association of America* 70, no. 3 241-47.

14. Hastie, T., R. Tibshrani & J. Friedman. (2009). *The Elements of Statistical Learning. Data Mining, Inference and Prediction* Second ed., Edited by Springer Series in Statistics. New York, USA: Springer.

15. Helson, H. . (1986). *The Spectral Theorem.* Edited by University of California. Berkeley, CA, USA: Springer.

16. Hilbert, D., W. Lothar & J. Von Neumann. (1927). "Über Die Grundlagen Der Quantenmechanik." *Mathematische Annalen* 98, 1-30.

17. Holmes, G., A. Donkin & I. H. Witten. (1994). "Weka (a Software Developed by Machine Learning Group)". www.cs.waikato.ac.nz/ml/weka.

18. Jaccard, P. (1901). "Étude Comparative De La Distribution Florale Dans Une Portion Des Alpes Et Des Jura." *Bulletin de la Société Vaudoise des Sciences Naturelles* 37, no. 1901 547-579.

19. Jiang, J. & D. W. Conrath. (1997). "*Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy."* In *International Conference Research on Computational Linguistics (ROCLING X)*. Taiwan.

20. Jolliffe, I. T. (2002). *Principal Component Analysis* 2ed. Springer Series in Statistics. New York: Springer-Verlag.

21. Larocca, J. N., A. A. Freitas & C. A. Kaestner. (2002). "*Automatic Text Summarization Using a Machine Learning Approach."* In *16th Brazilian Symposium on Artificial Intelligence, SBIA 2002*, edited by Pontifical Catholic Univ. of Parana (PUCPR), 2507, 205-15. Curitiba, Brazil: Springer-Verlag, Berlin, Germany.

22. Leacock, C., G. A. Miller & M. Chodorow. (1998). "Using Corpus Statistics and Wordnet Relations for Sense Identification." *Computational Linguistics* 24, no. 1 147-165.

23. Lin, D. (1998). "*An Information-Theoretic Definition of Similarity."* In *Fifteenth International Conference on Machine Learning (ICML'98)* edited by Manitoba Univ. Dept. of Comput. Sci., Winnipeg, Man., Canada, 296-304. Madison, Wisconsin.

24. Motta, J. A., L. Capus & N. Tourigny. (2011). "Insertion of Ontological Knowledge to Improve Automatic Summarization Extraction Methods." *Journal of Intelligence Learning Systems and Applications* 3, no. 3 131-138.

25. Motta, J. A., L. Capus & N. Tourigny. (2012). "Evaluation of Eficiency of Linear Techniques to Optimize Attributte Space in Machine Learning: Relevant Results for Extractive Methods of Summarizing." *Computer and Information Science* 5, no. 6 58-72.

26. Neto, J. L., A. A. Freitas & C. A. Kaestner. (2002). "*Automatic Text Summarization Using a Machine Learning Approach."* In *16th Brazilian Symposium on Artificial Intelligence, SBIA 2002*, 2507, 205-15. Curitiba, Brazil: Pontificial Catholic University of Parana (PUCPR).

27. NIST. (2006). "Document Understanding Conferences - Duc. 2006". http://www-nlpir.nist.gov/projects/duc.

28. Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in the Space." *Philosophical Magazine*, 559-572.

29. Rakotomalala, R. (2005). "*Tanagra: Un Logiciel Gratuit Pour L'enseignement Et La Recherche."* In *European Grid Conference. EGC'2005*, 2, 697-702. Amsterdam. Netherlands.

30. Reed, M. & V. Simon. (1975). *Methods of Modern Mathematical Physics.* San Diego, California: Academic Press Inc. Hardcourt Brace Jovanovich.

31. Resnick, P. (2000). "Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity in Natural Language." *Journal of Articial Intelligence Research* 11, 95-130.

32. Rosenblatt, F. (1957). *The Perceptron-a Perceiving and Recognizing Automaton*. New York: Cornell Aeronautical Laboratory.

33. Rumelhart, D., G. Hinton & R. J. Williams. (1986). "Learning Representations by Back-Propagating Errors." *Nature* 323, 533 - 536.

34. Sanghoon, L. (2013). "*Multi-Document Text Summarization Using Topic Model and Fuzzy Logic."* In *Machine Learning and Data Mining in Pattern Recognition. 9th International Conference, MLDM 2013*, edited by Georgia State Univ. Comput. Sci., 2013, 159-68. Atlanta, GA, USA.

35. Shen, C. & T. Li. (2011). "*Learning to Rank for Query-Focused Multi-Document Summarization."* In *11th IEEE International Conference on Data Mining. ICDM 2011*, 626-34. Miami, Fl, USA: School of Computing & Information Sciences, Florida International University

36. Stewart, G. W. (1993). "On the Early History of the Singular Value Decomposition." *SIAM Review* 35 no. 4 551-566.

37. Vapnik, V. (1999). *The Nature of Statistical Learning Theory.* Information Science and Statistics. New York: Springer-Verlag.

38. Wu, Z. & M. S. Palmer. (1994). "*Verb Semantics and Lexical Selection."* In *32nd. Annual Meeting of the Association for Computational Linguistics*, 133-138. San Francisco, California: Morgan Kaufmann.

39. Xmarks. (2012). "Freesummarizer". http://freesummarizer.com/.

CHAPITRE 7. Conclusion générale

Notre objectif principal était de proposer un nouveau modèle d'apprentissage automatique plus performant destiné à la production extractive et automatique de résumés multi document. Nous avons montré montrer comment nous avons atteint cet objectif en améliorant les modèles d'apprentissage automatique utilisés pour extraire les phrases les plus porteuses d'information d'un corpus en vue de produire des résumés. Plusieurs améliorations ont été proposées et validées. Rappelons que l'approche par apprentissage automatique prend en entrée un corpus de documents qui va servir d'ensemble d'entraînement pour l'algorithme d'apprentissage. Cet algorithme permet d'apprendre une fonction qui dans notre cas est une fonction de classification, qui détermine si une phrase est importante ou non, c'est-à-dire si elle est ou non porteuse de sens. Cette fonction de classification peut ensuite être utilisée sur de nouveaux corpus pour produire des résumés.

Nous nous étions fixé trois objectifs spécifiques, soit la création d'un ensemble d'entraînement performant, l'élaboration d'un processus d'abduction efficace qui à partir d'un ensemble d'entrainement détermine une fonction de classification et la définition d'une procédure pour l'évaluation du modèle. Ces trois objectifs ont guidé notre démarche dans le but de répondre à notre question de recherche qui était : l'introduction de connaissance ontologique dans l'espace d'attributs permet-elle d'optimiser le choix de l'ensemble d'entraînement pour l'abduction de fonctions de classification de phrases ? En d'autres mots, l'introduction de connaissance ontologique influence-t-elle positivement la performance du modèle de résumés extractifs ?

Dans le nouveau modèle proposé, VENCE, l'ensemble de phrases d'entrée est représenté au moyen d'un espace vectoriel. Cet espace est ensuite optimisé à l'aide de méthodes linéaires connues telles que l'extraction de composantes principales et la décomposition en vecteurs singuliers ainsi qu'à l'aide de méthodes issues de l'exploration de données telles que les cartes auto-adaptatives, l'analyse de facteurs et des k-moyennes. Comme nous l'avons démontré dans notre premier article (Motta *et al.*, 2012), l'adaptation de ces méthodes à notre contexte est très importante car cela permet d'augmenter l'efficacité de tâche de classification. Nous avons montré que l'ajout de connaissance ontologique, en utilisant un nouveau processus d'insertion qui renforce l'espace vectoriel optimisé, était également efficace. Les synonymes, les hyponymes et les hyperonymes de chaque attribut

de l'espace sont extraits de l'ontologie pour réaliser ce renforcement. À partir d'un espace renforcé par de la connaissance ontologique, nous avons construit un ensemble d'entraînement qui sert à obtenir des modèles d'apprentissage automatique. Nous avons expérimenté des techniques d'exploration de données et de reconnaissance de formes telles que la machine à vecteurs de support, le classeur bayésien naïf, le perceptron multicouche et les arbres décisionnels, afin de déterminer la technique la plus appropriée à ce contexte. De ces expérimentations, nous avons conclu que la machine à vecteurs de support et le classeur bayésien naïf présentaient les meilleures performances. C'est ainsi que ces deux techniques ont été choisies pour compléter notre modèle d'apprentissage automatique, comme nous l'avons montré dans notre deuxième article (Motta *et al.*, 2011). Dans ces expérimentations, nous avons utilisé le paradigme de l'apprentissage automatique non supervisé. Mais, il nous a semblé important d'introduire le facteur humain pour obtenir l'ensemble d'entraînement et ainsi se situer dans un approche d'apprentissage semi-supervisé. Pour la définition de l'ensemble d'entraînement, nous avons donc fait appel à une mesure normalement utilisée dans l'expérimentation biologique, appelé le coefficient de Jaccard (une variation de la similarité pour cosinus). Ce coefficient nous a permis de mesurer le degré de similarité entre les phrases de l'ensemble d'entraînement et les résumés manuels des documents étudiés en entrée.

Pour évaluer la performance du modèle proposé, nous avons utilisé non seulement les matrices de confusion, les métriques de précision, de rappel et de F_measure, les courbes ROC, mais aussi l'ensemble d'outils ROUGE, largement utilisé pour l'évaluation de résumés. Nous avons testé les moyens définis dans le modèle VENCE en utilisant la machine à vecteurs de support pour abduire les fonctions d'apprentissage sur un corpus de documents de DUC 2006. Avec ces fonctions d'apprentissage, nous avons ainsi pu extraire les phrases les plus porteuses d'information de nouveaux corpus. Nous avons mesuré la précision, le rappel et la F_measure pour 1-gram, 2-grams, 3-grams, ROUGE-L et ROUGE-SU des phrases extraites. La qualité de l'extrait obtenu est très bonne. Par exemple, citons les métriques ROUGE-L pour le rappel, la précision et la F_measure, qui sont de 0.838, 0.581 et 0.686 respectivement. Ces résultats ont été comparés avec des résultats publiés récemment dans des publications scientifiques de modèles similaires à VENCE. Nous avons pu constater que nos résultats sont bien supérieurs. Nous avons

également comparé les extraits de documents obtenus avec VENCE avec ceux donnés par les logiciels Copernic et FreeSummarizer d'un point de vue qualitatif. Nous avons pu également noter la puissance de notre modèle. L'expérimentation ainsi que les résultats ont été présentés dans le troisième article de la présente thèse.

La contribution à la recherche de cette thèse est importante puisqu'elle propose un nouveau modèle pour extraire des phrases importantes en vue de produire des résumés, à partir de nouvelles applications de techniques qui en améliorent les processus mis en œuvre. De plus, plusieurs améliorations à l'intérieur du modèle peuvent être utilisées pour construire d'autres modèles existants ou d'autres contextes. Ceci ouvre la voie à des perspectives de recherche futures. En effet, nous pensons qu'il serait intéressant de concevoir des modèles d'apprentissage automatique non-supervisé pour produire des résumés par extraction, avec l'aide des techniques d'optimisation décrites dans le premier article. De plus, il serait intéressant d'intégrer des techniques et des idées présentées dans cette thèse dans les domaines de la reconnaissance de formes et l'exploration de données. De même, certains principes de ce travail pourraient être utilisés pour définir des modèles d'analyse de sentiments. Enfin, il serait avantageux de profiter de la méthodologie présentée pour tester de nouvelles techniques d'optimisation et de classification dans le domaine du résumé automatique.

Bibliographie

Agresti, A. (2007). *An Introduction to Categorical Data Analysis. Building and Applying Logistic Regression Models.* Hoboken, New Jersey: John Wiley & Sons.

Barzilay, R. et M. Elhadad. (1997). *Using Lexical Chains for Text Summarization. Intelligent Scalable Text Summarization.*: Mathematics and Computer Science Department. Ben Gurion University.

Bellare, K., A. Das Sarma, Atish Das Sarma, N. Loiwal, V. Mehta, G. Ramakrishnan et P. Bhattacharyya. (2004). "Generic Text Summarization Using Wordnet." http://i.stanford.edu/~anishds/publications/lrec04/lrec04.ps.

Bellman, R. E. (1970). *Introduction to Matrix Analysis.* New York: McGraw-Hill.

Bellman, R. E. et S. E. Dreyfus. (1962). *Applied Dynamic Programming.* Princeton, NJ: Princeton University Press.

Bellman, R. E. et L. C. Kenneth. (1970). *Algorithms Graphs and Computers.* London: Academic Press Inc.

Beltrami, E. (1868). "Saggio Di Interpretazione Della Geometria Non-Euclidea." *Giornale di Mathematiche* VI, 285-315.

Benzécri, J. P. . (1973). *L'analyse Des Données. L'analyse Des Correspondences.* Vol. II. Paris: Dunod.

Berger, A., R. A. Caruana, D. L. Cohn, D. B. Freitag et V. O. Mittal (2000). "*Bridging the Lexical Chasm : Statistical Approaches to Answer Finding."* In *SIGIR '00 23rd annual international ACM SIGIR conference on Research and development in information retrieval.*, 34, 192-9. New York: ACM, USA.

Berry, M. W., S. T. Dumais et G. W. O'Brien. (1995). "Using Linear Algebre for Intelligent Information Retrieval." *SIAM Review* 37 no. 4 573-595.

Bishop, C. M. (2007). *Pattern Recognition and Machine Learning.* Information Science and Statistics. New York: Springer.

Bosser, B., I. Guyon et V. Vapnik. (1992). "*A Training Algorithm for Optimal Margin Classifiers."* In *5th Annual ACM Workshop on Computational Learning Theory*, 114-152. Pittsburg, USA.

Bouchard, M. et E. Bouchard. (1996). "Copernic Summarizer". http://www.copernic.com/.

Breiman, L. (2001). "Random Forests." *Machine Learning* 45, no. 1 5-32.

Buhmann, M. D. (2003). *Radial Basis Functions: Theory and Implementations.* United Kingdom: Cambridge University Press.

Carbonell et Goldstein. (1998). "*The Use of Mmr, Diversity-Based Reranking for Reordering Documents and Producing Summaries."* In *21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR'98*, edited by ACM, 335-6. Melbourne, Australie: ACM.

Chandra, M., V. Gupta et S. K. Paul. (2011). "*A Statistical Approach for Automatic Text Summarization by Extraction "* In *International Conference on Communication Systems and Network Technologies (CSNT)*, 268-271. Kattra, India: IEEEXplore.

Cohen, J D. (1995). "Hights : Language and Domain Independent Automatic Indexing Terms for Abstracting." *Journal of the American Society for Information Science* 43, no. 3 162-174.

Cortes, C. et V. Vapnik. (1995). *Support-Vector Networks*. Holmdel, NJ, USA.: AT&T Bell Labs.

Dawson, C. (2005). *Projects in Computing and Information Systems.* First ed., Edited by Addison-Wesley Educational Publications. Boston, USA: Addison-Wesley.

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer et R. Harshman. (1990). " Indexing by Latent Semantic Analysis." *Journal of the American for Information Science* 41, no. 6 391-407.

Demzar, J. et B. Zupan. (2010). "Orange (a Software Developed at Laboratopry of Artificial Intelligence)". http://orange.biolab.si/.

Dunlavy, D. M. , J. Conroy, D. P. O'Leary et J. Schlesinger. (2007). "Qcs: A Tool for Querying, Clustering, and Summarizing Documents." *Information Processing and Management: an International Journal* 43, no. 6 1588-1605.

Edmunson, H. P. (1969). "New Methods in Automatic Abstraction." *Journal of the Association for Computing Machinery* 16, no. 2 264-285.

Fahim, A. M., G. Saake, A. M. Salem, F. A. Torkey et M. A. Ramadan. (2009). "K-Means for Spherical Clusters with Large Variance in Sizes." *International Journal of Computer Science* 4, no. 3 145-150.

Fawcett, T. (2006). "An Introduction to Roc Analysis." *Pattern Recognition Letters* 27, no. 2006 861-874.

Fellbaum, C.D. (1985). "Wordnet (a Lexical Database for English)", Princeton University. http://wordnet.princeton.edu.

Fogarty, J., R. Baker et S. Hudson. (2005). "*Case Studies in the Use of Roc Curve Analysis for Sensor-Based Estimates in Human Computer Interaction "* In *ACM International Conference, Proceedings of Graphics Interface 2005*, edited by Canadian Human-Computer Communications Society. Waterloo, Ontario, Canada.

Fraleigh, J. B. et R. A. Beauregard. (1994). *Linear Algebra.* 3 ed. Boston: Addison-Wesley.

Gao, J. F (2004). "Introduction to the Special Issue on Statistical Language Modeling." *ACM Transations on Asian Language Information Processing* 3, no. 2 87-93.

García-Hernandez, R. A, R. Montiel, Y. Ledeneva, E. Rendón, A. Gelbuck et R. Cruz. (2008). "*Text Summarization by Sentence Extraction Using Unsupervised Learning "* In *Mexican International Conference on Artificial Intelligence. MICAI 2008*, 133-143. Ciudad de Mexico: Springer-Verlag.

Goldstein, J., M. Kantrowitz, V. O. Mittal et J. G. Carbonell. (1999). "*Summarizing Text Documents: Sentence Selection and Evaluation Metrics."* In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 121-128. Berkeley, California, USA.

Golub, G. H. et W. Kahan. (1965). "Calculing the Singular Values and Pseudo-Inverse of a Matrix." *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis* 2, no. 2 205-224.

Golub, G. H. et C. F. Van Loan. (1996). *Matrix Computations.* 3 ed. Baltimore, Mariland, USA: Johns Hopkins University Press.

Gong, Y. et X. Liu. (2001). "*Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis."* In *SIGIR '01 Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* New York, USA: ACM.

Gruber, T. R. (1995). "Toward Principles for the Design of Ontologies Used for Knowledge Sharing." *International Journal of Human-Computer Studies* 43, no. 5-6 907-928.

Hall, M. A. (1999). "Correlation-Based Feature Selection for Machine Learning " Ph.D Thesis, University of Waikato.

Halliday, M. et R. Hasan. (1976). *Cohesion in English.* London: Longman Publishing Group.

Halmos, P. R. . (1963). "What Does the Spectral Theorem Say?" *The American Mathematical Monthly - Mathematical Association of America* 70, no. 3 241-47.

Harman, D. (1994). "*Data Preparation."* In *TIPSTER Text Program Phase I.* Fredricksburg, Virginia: Morgan Kaufmann Publishing Co.

Hastie, T., R. Tibshrani et J. Friedman. (2009). *The Elements of Statistical Learning. Data Mining, Inference and Prediction* Second ed., Edited by Springer Series in Statistics. New York, USA: Springer.

Helson, H. . (1986). *The Spectral Theorem.* Edited by University of California. Berkeley, CA, USA: Springer.

Hennig, L. (2008). "*An Ontology-Based Approach to Text Summarization."* In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops*, 291-294.

Hilbert, D., W. Lothar et J. Von Neumann. (1927). "Über Die Grundlagen Der Quantenmechanik." *Mathematische Annalen* 98, 11-30.

Hirst, G. et D. St-Onge. (1998). "Lexical Chains as Representation of Context for the Detection and Correction of Malapropisms." *The MIT Press*, 305-332.

Hoey, M. (1991). *Patterns of Lexis in Text.* First ed. Oxford: Oxford University Press.

Holmes, G., A. Donkin et I. H. Witten. (1994). "Weka (a Software Developed by Machine Learning Group)". www.cs.waikato.ac.nz/ml/weka.

Horn, R. A. et C. R. Johnson. (1985). *Matrix Analysis.* Cambridge: Cambridge University Press.

Ikonomakis, M., S. Kotsiantis et V. Tampakas. (2005). "Text Classification Using Machine Learning Techniques." *WSEAS Transations on Computers* 4, no. 8 966-974.

Jaccard, P. (1901). "Étude Comparative De La Distribution Florale Dans Une Portion Des Alpes Et Des Jura." *Bulletin de la Société Vaudoise des Sciences Naturelles* 37, no. 1901 547-579.

Jiang, J. et D. W. Conrath. (1997). "*Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy."* In *International Conference Research on Computational Linguistics (ROCLING X)*. Taiwan.

Jolliffe, I. T. (2002). *Principal Component Analysis* 2ed. Springer Series in Statistics. New York: Springer-Verlag.

Jones, K. S. (1999). *Automatic Summarising: Factors and Directions*. Cambridge, England: Computer Laboratory, University of Cambridge.

Jones, K. S. (2007). "Automatic Summarising: The State of Art." *Information Processing and Management* 43, no. 6 1449-1481.

Jordan, C. (1870). *Traité Des Substitutions Et Des Équations Algébriques.* Paris - France: Gauthier-Villars.

Kim, J. et C. Mueller. (1978). *Factor Analysis: Statistical Methods and Practical Issues.* Sage University Paper Series on Quantitative Applications in the Social Sciences. Newbury Park, California: Sage Publications Inc.

Kohonen, T. (1990). "The Self-Organizing Map." *Proceedings of the IEEE* 78, no. 9 1464-1480.

Korfhage, R. R. (1997). *Information Storage and Retrieval.* Lansing, Michigan: Wiley.

Kuhn, H. W. et A. W. Tucker. (1951). "*Nonlinear Programming."* In *2nd Berkeley Symposium*, 481–492. Berkeley. University of California.

Kuhn, M. et K. Johnson. (2013). *Applied Predictive Modeling.* New York: Springer.

Kupiec, J. et F. Chen. (1995). "*A Trainable Document Summarizer."* In *18th annual international ACM SIGIR Conference on Research and Development in Information Retrieval* Seatle, WA.

Langley, P. (1994). "*Selection of Relevant Features in Machine Learning."* In *AAAI Fall Symposium on Relevance*, edited by AAAI Press. New Orleans, LA.

Larocca, J. N., A. A. Freitas et C. A. Kaestner. (2002). "*Automatic Text Summarization Using a Machine Learning Approach."* In *16th Brazilian Symposium on Artificial Intelligence, SBIA 2002*, edited by Pontifical Catholic Univ. of Parana (PUCPR), 2507, 205-15. Curitiba, Brazil: Springer-Verlag, Berlin, Germany.

Lasko, T. A., J. G. Bhagwat, K. H. Zou et L. Ohno-Machado. (2005). "The Use of Receiver Operating Characteristic Curves in Biomedical Informatics." *Journal of Biomedical Information* 38, no. 5 404-15.

Leacock, C., G. A. Miller et M. Chodorow. (1998). "Using Corpus Statistics and Wordnet Relations for Sense Identification." *Computational Linguistics* 24, no. 1 147-165.

Lin, C. Y. (2004). "*Rouge: A Package for Automatic Evaluation of Summaries."* In *Proceedings of Workshop on Text Summarization Branch Out (WAS 2004)*, 1-8. Barcelona, Spain.

Lin, C. Y. et E. Hovy. (2003). "*Automatic Evaluation of Summaries Using N-Gram Co-Occurrence Statistics. Rouge: A Package for Automatic Evaluation of Summaries."*

In *Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003)*. Edmonton, Canada.

Lin, D. (1998). "*An Information-Theoretic Definition of Similarity."* In *Fifteenth International Conference on Machine Learning (ICML'98)* edited by Manitoba Univ. Dept. of Comput. Sci., Winnipeg, Man., Canada, 296-304. Madison, Wisconsin.

Luhn, H. P. (1958). "The Automatic Creation of Literature Abstracts." *IBM Journal of Research and Development* 2, no. 2 159-165.

MacQueen, J B. (1967). "*Some Methods for Classification and Analysis of Multivariate Observations."* In *5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281-297

Mani, I. et E. Bloedorn. (1998). "*Machine Learning of Generic and User-Focused-Summarization."* In *Fifteenth national conference on Artificial intelligence/Innovative applications of artificial intelligence. AAAI '98*, 820-826: ACM.

Mani, I. J. (2001). *Automatic Summarization.* Natural Language Processing (Book 3), Edited by Amsterdam Publications. Amsterdam, The Netherlands: John Benjamin Publishing Company.

Manning, C. D., P. Raghavan et S. Schütze. (2008). *An Introduction to Information Retrieval.* Cambridge University Press.

McCargar, V. (2004). "Statistical Approaches to Automatic Text Summarization." *Bulletin of the American Society for Information Science and Technology* 30, no. 4 10.

McCulloch, W. et W. Pitts. (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics* 5, no. 4 115-133.

McLachlan, G J. (2004). *Discriminant Analysis and Statistical Pattern Recognition.* Wiley Interscience.

Miike, S., E. Itoh, K. Ono et K. Sumita. (1994). "*A Full Text Retrieval System with and Dynamic Abstract Generation Function."* In *Seventeenth Annual International*

ACM-SIGIR Conference on Research and Development in Information Retrieval, 152-161. Dublin: Springer, London.

Mitchell, T M. (1997). *Machine Learning.* Edited by McGraw - Hill.

Morris, J. et G. Hirst. (1991). "Cohésion Lexicale Computed by Thesaural Relations as a Indicator of the Structure of Text." *Computational Linguistics* 17, no. 1 21-48.

Motta, J. A., L. Capus et N. Tourigny. (2011). "Insertion of Ontological Knowledge to Improve Automatic Summarization Extraction Methods." *Journal of Intelligence Learning Systems and Applications* 3, no. 3 131-138.

Motta, J. A., L. Capus et N. Tourigny. (2012). "Evaluation of Eficiency of Linear Techniques to Optimize Attributte Space in Machine Learning: Relevant Results for Extractive Methods of Summarizing." *Computer and Information Science* 5, no. 6 58-72.

Navidi, W. (2010). *Statistics for Engineers and Scientists.* 3 ed.: McGraw-Hill Science/Engineering/Math.

Neapolitan, R. E. (2003). *Learning Bayesian Networks.* 1 ed.: Prentice Hall.

Neches, R., R. E. Fikes, T. Finin, T. R. Gruber, R. S. Patil, T. E. Senator et W. R. Swartout (1991). "Enabling Technology for Knowlwdge Sharing." *AI Magazine* 12, no. 3 16-36.

Nenkova, A et K McKeown. (2011). "Automatic Summarization." *Foundations and Trends in Information Retrieval* 5, no. 2-3 103-233.

Neto, J. L., A. A. Freitas et C. A. Kaestner. (2002). "*Automatic Text Summarization Using a Machine Learning Approach."* In *16th Brazilian Symposium on Artificial Intelligence, SBIA 2002*, 2507, 205-15. Curitiba, Brazil: Pontificial Catholic University of Parana (PUCPR).

NIH. (2013). "Unified Medical Language System® (Umls®)". http://www.nlm.nih.gov/research/umls.

NIST. (1993). "Text Retrieval Conference (Trec)". http://trec.nist.gov/.

NIST. (2002). "Document Understanding Conference - Duc2002". http://duc.nist.gov/pubs.html#2002.

NIST. (2006). "Document Understanding Conference - Duc2006". http://www-nlpir.nist.gov/projects/duc.

O'Brien, G .W. (1994). *Information Management Tools for Updating an Svd-Indexing Scheme - Report*. Tennessee: The University of Knoxville.

Ohno-Machado, L, T. A. Lasko, J. G. Bhagwat et K. H. Zou. (2005). "The Use of Receiver Operating Characteristic Curves in Biomedical Informatics " *Journal of Biomedical Informatics* 38, no. 5 401-15.

Paice, C. et P. Jones. (1993). "*The Indentification of Importants Concepts in Highly Structured Technical Papers."* In *ACM-SIGIR'93*, 69-78. Pittsburg, PA: ACM.

Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in the Space." *Philosophical Magazine*, 559-572.

Pollock, J. et A. Zamora. (1975). "Automatic Abstracting Research at Chemical Abstracts Service." *Journal of Chemical Information and Computer Sciences* 4, no. 15.

Quinlan, J. R. (1986). "Induction of Decision Trees." *Machine Learning* 1 no. 1 81-106.

Quinlan, J. R. (1992). *C4.5 : Programs for Machine Learning.* 1 ed. San Francisco. Californie.: Morgan-Kaufman.

Radev, D, H. Jing et M. Budzikouska. (2000). "*Centroid-Based Summarisation of Multiple Documents : Sentence Extraction, Utility-Based Evaluation and User Studies."* In *6th Applied Natural Language Processing Conference. (ANLPANLP/NAACL-2000*, 1, 21-30. Seattle, Washington, USA.

Rakotomalala, R. (2005). "*Tanagra: Un Logiciel Gratuit Pour L'enseignement Et La Recherche."* In *European Grid Conference. EGC'2005*, 2, 697-702. Amsterdam. Netherlands.

Ramakrishnan, G. et P Bhattacharya. (2003). "*Text Repersentation of Wordnet Synsets."* In *8th Conference on Application of Natural Language to Informations Systems*, 29, 214-227. Burg, Germany.

Rao, S. S. (2002). *Applied Numerical Methods for Engineers and Scientists.* Upper Saddle River, NJ: Prentice Hall.

Reed, M. et V. Simon. (1975). *Methods of Modern Mathematical Physics.* San Diego, California: Academic Press Inc. Hardcourt Brace Jovanovich.

Resnick, P. (2000). "Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity in Natural Language." *Journal of Articial Intelligence Research* 11, 95-130.

Riley, K. F, M. P Hobson et S. J Bence. (2006). *Mathematical Methods for Physics and Engineering.* New York: Cambridge University Press.

Robins, R. H. (1989). *General Linguistics.* 4 ed., Edited by Longman. London: Longman Publishing Group.

Rosenblatt, F. (1957). *The Perceptron-a Perceiving and Recognizing Automaton - Report*. New York: Cornell Aeronautical Laboratory.

Rousseeuw, P. J. (1986). "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics* 20(1987), 53-65.

Rumelhart, D., G. Hinton et R. J. Williams. (1986). "Learning Representations by Back-Propagating Errors." *Nature* 323, 533 - 536.

Saleh, A. (2004). "Reuters Corpus (Reuters News Agency)". http://about.reuters.com/researchandstandards/corpus/.

Salton, G. et C. Buckley. (1988). "Term-Weighting Approaches in Automatic Rext Retrieval." *Information Processing and Management*, 513-523.

Salton, G., A. Wong et C. S. Yang. (1975). "A Vector Space Model for Information Retrieval." *Communications of the ACM* 18, no. 11 613-620.

Sanghoon, L. (2013). "*Multi-Document Text Summarization Using Topic Model and Fuzzy Logic."* In *Machine Learning and Data Mining in Pattern Recognition. 9th International Conference, MLDM 2013*, edited by Georgia State Univ. Comput. Sci., 2013, 159-68. Atlanta, GA, USA.

Shakhnarovish, D., T. Darrel et P. Indyk. (2005). *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice.* Massachusets, USA: MIT Press.

Shannon, C. E. (1948). "A Mathematical Theory of Communication." *Bell System Technical Journal* 27, 379-423,623-656.

Sharan, A. et H. Imran. (2009). "*Machine Learning Approach for Automatic Document Summarization."* In *World Academy of Science, Engineering and Technology*, 39, 2070-3740: PWASET.

Shen, C. et T. Li. (2011). "*Learning to Rank for Query-Focused Multi-Document Summarization."* In *11th IEEE International Conference on Data Mining*. Miami, Fl.

Smale, S. (1997). "Complexity Theory and Numerical Analysis." *Acta Numerica* 6, 523-51.

Spackman, K. A. (1989). "*Signal Detection Theory: Valuable Tools for Evaluating Inductive Learning."* In *Sixth International Workshop on Machine Learning*. San Mateo, CA.: ACM.

Stairmand, M. (1996). "A Computational Analysis of Lexical Cohesion in Information Retrieval." PH.D, University of Manchester Institute of Science and Technology (UMIST).

Steinbach, M., P. Tan et V. Kumar. (2006). *Introduction to Data Mining.* Pearson Editions. Oxon, United Kingdom Addison-Wesley.

Stewart, G. W. (1993). "On the Early History of the Singular Value Decomposition." *SIAM Review* 35 no. 4 551-566.

Stewart, G. W. (1973). *Introduction to Matrix Computations.* New York: Academic Press.

Vapnik, V. (1999). *The Nature of Statistical Learning Theory.* Information Science and Statistics. New York: Springer-Verlag.

Verma, R., P. Chen et W. Lu. (2007). *A Semantic Free-Text Summarization System Using Ontology Knowledge*. Houston: University of Houston.

Vikas, O. (2008). *Multiple Document Summarization Using Principal Component Analysis Incorporating Semantic Vector Space Model*. Gwailor, India: Indian Institute of Information Technology and Management.

W3C. (2001). "World Wide Web Consortium". www.w3.org/2001/sw.

Wnek, K., K. Kaufman, E. Bloedorn et R. S Michalski. (1995). *Selective Inductive Learning Method "Aq15c" : The Method and User Guide. Machine Learning and Inference* Fairfax. Virginia: Laboratory Report ML95-4. George Mason University.

Wolpert, D. et W. Macready. (1997). "No Free Lunch Theorems for Optimization." *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 1, no. 1 16.

Wu, Z. et M. S. Palmer. (1994). "*Verb Semantics and Lexical Selection."* In *32nd. Annual Meeting of the Association for Computational Linguistics*, 133-138. San Francisco, California: Morgan Kaufmann.

Xmarks. (2012). "Freesummarizer". http://freesummarizer.com/.

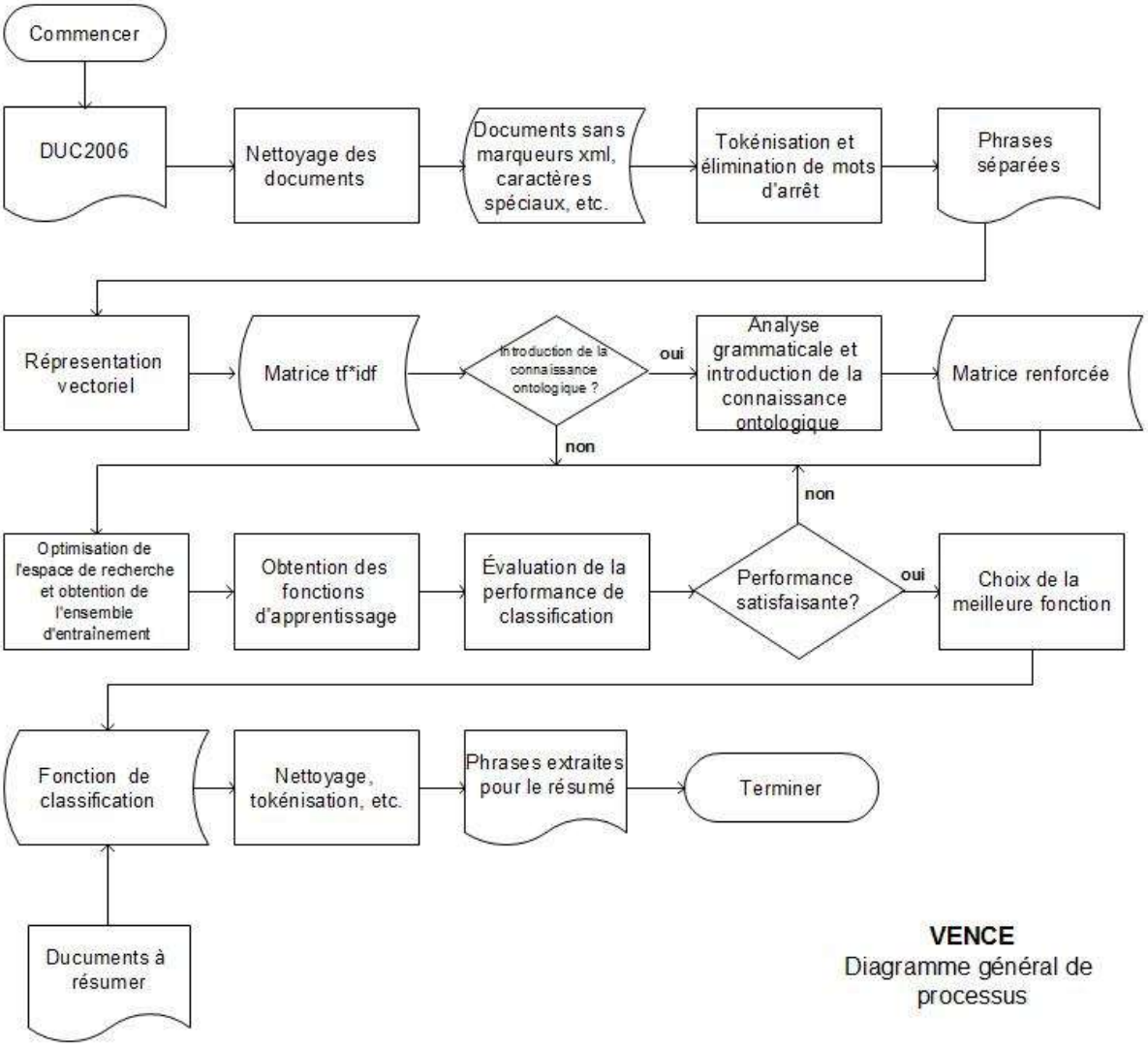
Xuexian, H., Z . Guowei, O. Wataru, W . Tetsushi et K. Fumitaka. (2004). "Accuracy Improvement of Automatic Text Classification Based on Feature Transformation and Multi-Classifier Combination " *LNCS* 3309/2004, 463-468.

Yu, F., D. Zheng, T. J. Zhao, S. Li et H. Yu. (2006). "*Text Classification Based on a Combination of Ontology with Statistical Method."* In *Fifth International Conference On Machine Learning and Cybernetics*, 6. Dalian, China: IEEE, Piscataway, NJ, USA.

Zhong, S. (2005). "*Efficient Online Spherical K-Means Clustering."* In *IEEE International Joint Conference on Neural Networks - IJCNN 2005*, 5, 3180-3185. Montréal, Canada: Institute of Electrical and Electronics Engineers Inc.

Annexes

Annexe 1. Diagramme général de processus



Annexe 2. Document DUC2006

<DOC>
<DOCNO> NYT19990406.0038 </DOCNO>
<DOCTYPE> NEWS STORY </DOCTYPE>
<DATE_TIME> 1999-04-06 03:33 </DATE_TIME>
<HEADER>
A7345 &Cx1f; ttd-z
r a &Cx13; &Cx11; BC-PAIN-DRUGS-SFCHRON(SI   04-06 0593
</HEADER>
<BODY>
<SLUG> BC-PAIN-DRUGS-SFCHRON(SIDEBAR </SLUG>
 TO PAIN-LIVING-SFCHRON)
 NEWEST DRUGS RELIEVE PAIN WITH LESS RISK &HT;   By CARL T. HALL
 &HT;   c.1999 San Francisco Chronicle
<TEXT>
 &LR;   &LR; New insights into the biochemistry of pain are starting to pay
off in safer, and perhaps even more effective, treatment options
for chronic sufferers.
<P>
   One new type of painkiller, known as COX-2 inhibitors, just hit
pharmacy shelves. The drugs appear to be no more effective at
fighting pain than traditional anti-inflammatory medications such
as ibuprofen and aspirin, but offer the advantage of fewer side
effects.
</P>
<P>
   That is a major plus for people who must use pain relievers
long-term and risk such problems as stomach bleeding and ulcers.
</P>
<P>
   The first of the new drugs, Celebrex, developed by Monsanto's
G.D. Searle & Co. pharmaceutical unit and co-marketed by Pfizer
Inc., was launched commercially last month.
</P>
<P>
   A different drug of the same type, Merck & Co.'s Vioxx, is
awaiting approval by the Food and Drug Administration and could be
on the market by summer.
</P>
<P>
   Industry analysts view the drugs as potential blockbusters,
likely to command at least 10 percent of the market for common pain
relievers.
</P>
<P>
   COX-2 drugs are designed primarily for those with rheumatoid
arthritis, which afflicts about 2.1 million people in the United
States, and for painful flareups of osteoarthritis, the
``wear-and-tear'' form of the ailment, which affects an estimated
20.7 million people.
</P>
<P>
   Until now, these patients and others in chronic pain have relied
mostly on nonsteroidal anti-inflammatory drugs, or NSAIDs, sold
over-the-counter or at prescription strength under brand names such
as Advil, Daypro, Lodine and Naprosyn.
</P>
<P>
   The drugs are generally effective but can be harmful with
constant use. A 1997 study of arthritis patients showed at least
107,000 hospitalizations and 16,500 deaths a year from
NSAID-related problems.
</P>
<P>
   COX-2 drugs block production of an enzyme called
cyclooxygenase-2, which produces painful inflammation. Unlike
NSAIDs, however, they do not interfere with the closely related
COX-1 enzyme, which helps build a protective lining in the stomach
and gastrointestinal tract.
</P>

<P>
   Searle began selling Celebrex for $2.42 per 200 milligram pill, less than expected but still more than some HMOs might be willing to pay, especially for patients without a demonstrated problem with NSAIDs.
</P>
<P>
   The COX-2 inhibitors, meanwhile, are not the only new possibilities for fighting pain:
</P>
<P>
   -- Ziconotide, awaiting FDA approval, has been touted as a potent alternative to morphine in cases of severe pain. Derived from sea snail venom, it is designed to block calcium channels in nerve cells of the spine, interrupting the flow of pain signals to the brain.
</P>
<P>
   -- Enbrel, first in a new class of rheumatoid arthritis medicines called biologic response modifiers, soaks up excess ``tumor necrosis factor,'' a chemical secreted by immune cells involved in joint inflammations.
</P>
<P>
   The drug, made by Immunex Corp. of Seattle, was approved by the FDA last year1998cq &LR; for use in adults. New studies show that it may also be effective in childhood forms of arthritis.
</P>
<P>
   -- Neurontin, known generically as gabapentin, is an established epilepsy drug that has been shown to work against hard-to-treat nerve pain from shingles or diabetes.
</P>
<P>
   -- Tests are underway to establish whether memantine _ used in Germany to treat Parkinson's disease _ can ease the pain of diabetic neuropathy. The drug blocks NMDA receptors, nerve cell sites that help relay pain signals.
</P>
</TEXT>
</BODY>
<TRAILER>
NYT-04-06-99 0333EDT &QL;
</TRAILER>
</DOC>

Annexe 3. Nettoyage du document

D0605ENYT19990406.0038 One new type of painkiller, known as COX-2 inhibitors, just hit pharmacy shelves. The drugs appear to be no more effective at fighting pain than traditional anti-inflammatory medications such as ibuprofen and aspirin, but offer the advantage of fewer side effects.

D0605ENYT19990406.0038 That is a major plus for people who must use pain relievers long-term and risk such problems as stomach bleeding and ulcers.

D0605ENYT19990406.0038 The first of the new drugs, Celebrex, developed by Monsanto's G.D. Searle & Co. pharmaceutical unit and co-marketed by Pfizer Inc., was launched commercially last month.

D0605ENYT19990406.0038 A different drug of the same type, Merck & Co.'s Vioxx, is awaiting approval by the Food and Drug Administration and could be on the market by summer.

D0605ENYT19990406.0038 Industry analysts view the drugs as potential blockbusters, likely to command at least 10 percent of the market for common pain relievers.

D0605ENYT19990406.0038 COX-2 drugs are designed primarily for those with rheumatoid arthritis, which afflicts about 2.1 million people in the United States, and for painful flareups of osteoarthritis, the ``wear-and-tear'' form of the ailment, which affects an estimated 20.7 million people.

D0605ENYT19990406.0038 Until now, these patients and others in chronic pain have relied mostly on nonsteroidal anti-inflammatory drugs, or NSAIDs, sold over-the-counter or at prescription strength under brand names such as Advil, Daypro, Lodine and Naprosyn.

D0605ENYT19990406.0038 The drugs are generally effective but can be harmful with constant use. A 1997 study of arthritis patients showed at least 107,000 hospitalizations and 16,500 deaths a year from NSAID-related problems.

D0605ENYT19990406.0038 COX-2 drugs block production of an enzyme called cyclooxygenase-2, which produces painful inflammation. Unlike NSAIDs, however, they do not interfere with the closely related COX-1 enzyme, which helps build a protective lining in the stomach and gastrointestinal tract.

D0605ENYT19990406.0038 Searle began selling Celebrex for $2.42 per 200 milligram pill, less than expected but still more than some HMOs might be willing to pay, especially for patients without a demonstrated problem with NSAIDs.

D0605ENYT19990406.0038 The COX-2 inhibitors, meanwhile, are not the only new possibilities for fighting pain:

D0605ENYT19990406.0038 potent alternative to morphine in cases of severe pain. Derived from sea snail venom, it is designed to block calcium channels in nerve cells of the spine, interrupting the flow of pain signals to the brain.

D0605ENYT19990406.0038 medicines called biologic response modifiers, soaks up excess ``tumor necrosis factor,'' a chemical secreted by immune cells involved in joint inflammations.

D0605ENYT19990406.0038 The drug, made by Immunex Corp. of Seattle, was approved by the FDA last year1998cq for use in adults. New studies show that it may also be effective in childhood forms of arthritis.

D0605ENYT19990406.0038 epilepsy drug that has been shown to work against hard-to-treat nerve pain from shingles or diabetes.

Annexe 4. Document tokenizé

D0605ENYT19990406.0038 One new type of painkiller, known as inhibitors, just hit pharmacy shelves.

D0605ENYT19990406.0038 The drugs appear to be no more effective at fighting pain than traditional anti-inflammatory medications such as ibuprofen and aspirin, but offer the advantage of fewer side effects.

D0605ENYT19990406.0038 That is a major plus for people who must use pain relievers long-term and risk such problems as stomach bleeding and ulcers.

D0605ENYT19990406.0038 The first of the new drugs, Celebrex, developed by Monsanto's GD Searle Co pharmaceutical unit and co-marketed by Pfizer Inc, was launched commercially last month.

D0605ENYT19990406.0038 A different drug of the same type, Merck Co's Vioxx, is awaiting approval by the Food and Drug Administration and could be on the market by summer.

D0605ENYT19990406.0038 Industry analysts view the drugs as potential blockbusters, likely to command at least percent of the market for common pain relievers.

D0605ENYT19990406.0038 drugs are designed primarily for those with rheumatoid arthritis, which afflicts about million people in the United States, and for painful flareups of osteoarthritis, the wear-and-tear" form of the ailment, which affects an estimated million people.

D0605ENYT19990406.0038 Until now, these patients and others in chronic pain have relied mostly on nonsteroidal anti-inflammatory drugs, or NSAIDs, sold over-the-counter or at prescription strength under brand names such as Advil, Daypro, Lodine and Naprosyn.

D0605ENYT19990406.0038 The drugs are generally effective but can be harmful with constant use.

D0605ENYT19990406.0038 A study of arthritis patients showed at least hospitalizations and deaths a year from NSAID-related problems.

D0605ENYT19990406.0038 drugs block production of an enzyme called which produces painful inflammation.

D0605ENYT19990406.0038 Unlike NSAIDs, however, they do not interfere with the closely related enzyme, which helps build a protective lining in the stomach and gastrointestinal tract.

D0605ENYT19990406.0038 Searle began selling Celebrex for per milligram pill, less than expected but still more than some HMOs might be willing to pay, especially for patients without a demonstrated problem with NSAIDs.

D0605ENYT19990406.0038 The inhibitors, meanwhile, are not the only new possibilities for fighting pain potent alternative to morphine in cases of severe pain.

D0605ENYT19990406.0038 Derived from sea snail venom, it is designed to block calcium channels in nerve cells of the spine, interrupting the flow of pain signals to the brain.

D0605ENYT19990406.0038 medicines called biologic response modifiers, soaks up excess tumor necrosis factor a chemical secreted by immune cells involved in joint inflammations.

D0605ENYT19990406.0038 The drug, made by Immunex Corp.

D0605ENYT19990406.0038 of Seattle, was approved by the FDA last for use in adults.

D0605ENYT19990406.0038 New studies show that it may also be effective in childhood forms of arthritis.

D0605ENYT19990406.0038 epilepsy drug that has been shown to work against hard-to-treat nerve pain from shingles or diabetes.

D0605ENYT19990406.0038 Germany to treat Parkinson's disease can ease the pain of diabetic neuropathy.

D0605ENYT19990406.0038 The drug blocks NMDA receptors, nerve cell sites that help relay pain signals.

Annexe 5. Liste de mots outils (stopwords)

a
able
about
above
abroad
according
accordingly
across
actually
adj
after
afterwards
again
against
ago
ahead
ain't
all
allow
allows
almost
alone
along
alongside
already
also
although
always
am
amid
amidst
among
amongst
an
and
another
any
anybody
anyhow
anyone
anything
anyway
anyways
anywhere
apart
appear
appreciate
appropriate
are
aren't
around
as
a's
aside
ask
asking
associated
at
available
away
awfully
b
back
backward
backwards
be
became
because
become
becomes
becoming
been
before
beforehand
begin
behind
being
believe
below
beside
besides
best
better
between
beyond
both
brief
but
by
c
came
can
cannot
cant
can't
caption
cause
causes
certain
certainly
changes
clearly
c'mon
co
co.
com
come
comes
concerning
consequently
consider
considering
contain
containing
contains
corresponding
could
couldn't
course
c's
currently
d
dare
daren't
definitely
described
despite
did
didn't
different
directly
do
does
doesn't
doing
done
don't
down
downwards
during
e
each
edu
eg
eight
eighty
either
else
elsewhere
end
ending
enough
entirely
especially
et
etc
even
ever
evermore
every
everybody
everyone
everything
everywhere
ex
exactly
example
except
f
fairly
far
farther
few
fewer
fifth
first
five
followed
following
follows
for
forever
former
formerly
forth
forward
found
four
from
further
furthermore
g
get
gets
getting
given
gives
go
goes
going
gone
got
gotten
greetings
h
had
hadn't
half
happens
hardly
has
hasn't
have
haven't
having
he
he'd
he'll
hello
help
hence
her
here
hereafter
hereby
herein
here's
hereupon
hers
herself
he's
hi
him
himself
his
hither
hopefully
how
howbeit
however
hundred
i
i'd
ie
if
ignored
i'll
i'm
immediate
in
inasmuch
inc
inc.
indeed
indicate
indicated
indicates
inner
inside
insofar

instead
into
inward
is
isn't
it
it'd
it'll
its
it's
itself
i've
j
just
k
keep
keeps
kept
know
known
knows
l
last
lately
later
latter
latterly
least
less
lest
let
let's
like
liked
likely
likewise
little
look
looking
looks
low
lower
ltd
m
made
mainly
make
makes
many
may
maybe
mayn't
me
mean
meantime
meanwhile
merely
might
mightn't
mine
minus
miss
more
moreover
most
mostly
mr
mrs
much
must
mustn't
my
myself
n
name
namely
nd
near
nearly
necessary
need
needn't
needs
neither
never
neverf
neverless
nevertheless
new
next
nine
ninety
no
nobody
non
none
nonetheless
noone
no-one
nor
normally
not
nothing
notwithstanding
novel
now
nowhere
o
obviously
of
off
often
oh
ok
okay
old
on
once
one
ones
one's
only
onto
opposite
or
other
others
otherwise
ought
oughtn't
our
ours
ourselves
out
outside
over
overall
own
p
particular
particularly
past
per
perhaps
placed
please
plus
possible
presumably
probably
provided
provides
q
que
quite
qv
r
rather
rd
re
really
reasonably
recent
recently
regarding
regardless
regards
relatively
respectively
right
round
s
said
same
saw
say
saying
says
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
shan't
she
she'd
she'll
she's
should
shouldn't
since
six
so
some
somebody
someday
somehow
someone
something
sometime
sometimes
somewhat
somewhere
soon
sorry
specified
specify
specifying
still
sub
such
sup
sure
t
take
taken
taking
tell
tends
th
than
thank
thanks
thanx
that
that'll
thats
that's
that've
the
their
theirs
them
themselves
then
thence
there
thereafter
thereby
there'd
therefore
therein
there'll
there're
theres
there's
thereupon
there've
these
they
they'd
they'll
they're
they've
thing
things
think
third
thirty
this
thorough
thoroughly
those
though
three
through
throughout

thru
thus
till
to
together
too
took
toward
towards
tried
tries
truly
try
trying
t's
twice
two
u
un
under
underneath
undoing
unfortunately
unless
unlike
unlikely
until
unto
up
upon
upwards
us
use
used
useful
uses
using
usually
v
value
various
versus
very
via
viz
vs
w
want
wants
was
wasn't
way
we
we'd
welcome
well
we'll
went
were
we're
weren't
we've
what
whatever
what'll
what's
what've
when
whence
whenever
where
whereafter
whereas
whereby
wherein
where's
whereupon
wherever
whether
which
whichever
while
whilst
whither
who
who'd
whoever
whole
who'll
whom
whomever
who's
whose
why
will
willing
wish
with
within
without
wonder
won't
would
wouldn't
x
y
yes
yet
you
you'd
you'll
your
you're
yours
yourself
yourselves
you've
z
zero

Annexe 6. Des Phrases sans mots outils (stopwords) (l'étoile indique la place du mot enlevé)

d0605enyt19990406.0038
*
*
type
*
painkiller
*
*
inhibitors
*
hit
pharmacy
shelves.
//
d0605enyt19990406.0038
*
drugs
*
*
*
*
*
effective
*
fighting
pain
*
traditional
anti-inflammatory
medications
*
*
ibuprofen
*
aspirin
*
offer
*
advantage
*
*
side
effects.
//
d0605enyt19990406.0038
*
*
*
major
plus
*
people
*
*
*
pain
relievers
long-term
*
risk
*
problems
*
stomach
bleeding
*
ulcers.
//
d0605enyt19990406.0038
*
*
*
*
*
drugs
celebrex
developed
*
monsanto
gd
searle
*
pharmaceutical
unit
*
co-marketed
*
pfizer
*
*
launched
commercially
*
month.
//
d0605enyt19990406.0038
*
*
drug
*
*
*
type
merck
*
vioxx
*
awaiting
approval
*
*
food
*
drug
administration
*
*
*
*
*
market
*
summer.
//
d0605enyt19990406.0038
industry
analysts
view
*
drugs
*
potential
blockbusters
*
*
command
*
*
percent
*
*
market
*
common
pain
relievers.
//
d0605enyt19990406.0038
drugs
*
designed
primarily
*
*
*
rheumatoid
arthritis
*
afflicts
*
million
people
*
*
united
states
*
*
painful
flareups
*
osteoarthritis
*
wear-and-tear
form
*
*
ailment
*
affects
*
estimated
million
people.
//
d0605enyt19990406.0038
*
*
*
patients
*
*
*
chronic
pain

*
relied
*
*
nonsteroidal
anti-inflammatory
drugs
*
nsaids
sold
over-the-counter
*
*
prescription
strength
*
brand
names
*
*
advil
daypro
lodine
*
naprosyn.
//
d0605enyt19990406.0038
*
drugs
*
generally
effective
*
*
*
harmful
*
constant
*
//
d0605enyt19990406.0038
*
study
*
arthritis
patients
showed
*
*
hospitalizations
*
deaths
*
year
*
nsaid-related
problems.
//
d0605enyt19990406.0038
drugs
block
production
*
*
enzyme
called
*
produces
painful
inflammation.
//
d0605enyt19990406.0038
*
nsaids
*
*
*
*
interfere
*
*
closely
related
enzyme
*
helps
build
*
protective
lining
*
*
stomach
*
gastrointestinal
tract.
//
d0605enyt19990406.0038
searle
began
selling
celebrex
*
per
milligram
pill
*
*
expected
*
*
*
*
*
hmos
*
*
*
*
pay
*
*
patients
*
*
demonstrated
problem
*
nsaids.
//
d0605enyt19990406.0038
*
inhibitors
*
*
*
*
*
*
possibilities
*
fighting
pain
potent
alternative
*
morphine
*
cases
*
severe
pain.
//
d0605enyt19990406.0038
derived
*
sea
snail
venom
*
*
designed
*
block
calcium
channels
*
nerve
cells
*
*
spine
interrupting
*
flow
*
pain
signals
*
*
brain.
//
d0605enyt19990406.0038
medicines
called
biologic
response
modifiers
soaks
*
excess
tumor
necrosis
factor
*
chemical
secreted
*
immune
cells
involved
*
joint
inflammations.
//
d0605enyt19990406.0038
*
drug
*
*

immunex
corp.
//
d0605enyt19990406.0038
*
seattle
*
approved
*
*
fda
*
*
*
*
adults.
//
d0605enyt19990406.0038
*
studies
show
*
*
*
*
*
effective
*
childhood
forms
*
arthritis.
//
d0605enyt19990406.0038
epilepsy
drug
*
*
*
shown
*
work
*
hard-to-treat
nerve
pain
*
shingles
*
diabetes.
//
d0605enyt19990406.0038
germany
*
treat
parkinson
disease
*
ease
*
pain
*
diabetic
neuropathy.
//
d0605enyt19990406.0038
*
drug
blocks
nmda
receptors
nerve
cell
sites
*
*
relay
pain
signals.
//

Annexe 7. Analyse grammaticale (TreeTagger-POST)

```
d0605enyt19990406.0038	NN	<unknown>
one	CD	one
new	JJ	new
type	NN	type
of	IN	of
painkiller	NN	painkiller
known	VVN	know
as	IN	as
inhibitors	NNS	inhibitor
just	RB	just
hit	VVD	hit
pharmacy	NN	pharmacy
shelves	NNS	shelf
.	SENT	.
//	NN	<unknown>
0252	CD	@card@
d0605enyt19990406.0038	NN	<unknown>
the	DT	the
drugs	NNS	drug
appear	VVP	appear
to	TO	to
be	VB	be
no	DT	no
more	RBR	more
effective	JJ	effective
at	IN	at
fighting	VVG	fight
pain	NN	pain
than	IN	than
traditional	JJ	traditional
anti-inflammatory	JJ	anti-inflammatory
medications	NNS	medication
such	JJ	such
as	IN	as
ibuprofen	NN	ibuprofen
and	CC	and
aspirin	NN	aspirin
but	CC	but
offer	VV	offer
the	DT	the
advantage	NN	advantage
of	IN	of
fewer	JJR	few
side	NN	side
effects	NNS	effect
.	SENT	.
//	NN	<unknown>
0253	CD	@card@
d0605enyt19990406.0038	NN	<unknown>
that	WDT	that
is	VBZ	be
a	DT	a
major	JJ	major
plus	NN	plus
for	IN	for
people	NNS	people
who	WP	who
must	MD	must
use	VV	use
pain	NN	pain
relievers	NNS	reliever
long-term	JJ	long-term
and	CC	and
risk	VV	risk
such	JJ	such
problems	NNS	problem
as	IN	as
stomach	NN	stomach
bleeding	NN	bleeding
and	CC	and
ulcers	NNS	ulcer
.	SENT	.
//	NN	<unknown>
0254	CD	@card@
d0605enyt19990406.0038	NN	<unknown>
the	DT	the
first	JJ	first
of	IN	of
the	DT	the
new	JJ	new
drugs	NNS	drug
celebrex	NN	<unknown>
developed	VVN	develop
by	IN	by
monsanto	NN	<unknown>
gd	NN	<unknown>
searle	NP	Searle
co	NP	Co
pharmaceutical	JJ	pharmaceutical
unit	NN	unit
and	CC	and
co-marketed	JJ	<unknown>
by	IN	by
pfizer	NN	<unknown>
inc	NN	<unknown>
was	VBD	be
launched	VVN	launch
commercially	RB	commercially
last	JJ	last
month	NN	month
.	SENT	.
//	NN	<unknown>
0255	CD	@card@
d0605enyt19990406.0038	NN	<unknown>
a	DT	a
different	JJ	different
drug	NN	drug
of	IN	of
the	DT	the
same	JJ	same
type	NN	type
merck	NP	Merck
co	NP	Co
vioxx	NP	<unknown>
is	VBZ	be
awaiting	VVG	await
approval	NN	approval
by	IN	by
the	DT	the
food	NN	food
and	CC	and
drug	NN	drug
administration	NN	administration
and	CC	and
could	MD	could
be	VB	be
on	IN	on
the	DT	the
market	NN	market
by	IN	by
summer	NN	summer
.	SENT	.
//	NN	<unknown>
0256	CD	@card@
```

d0605enyt19990406.0038 NN <unknown>
industry NN industry
analysts NNS analyst
view VVP view
the DT the
drugs NNS drug
as IN as
potential JJ potential
blockbusters NNS blockbuster
likely JJ likely
to TO to
command VV command
at IN at
least JJS least
percent NN percent
of IN of
the DT the
market NN market
for IN for
common JJ common
pain NN pain
relievers NNS reliever
. SENT .
// NN <unknown>
0257 CD @card@
d0605enyt19990406.0038 NN <unknown>
drugs NNS drug
are VBP be
designed VVN design
primarily RB primarily
for IN for
those DT those
with IN with
rheumatoid JJ rheumatoid
arthritis NN arthritis
which WDT which
afflicts VVZ afflict
about RB about
million CD million
people NNS people
in IN in
the DT the
united VVN unite
states NNS state
and CC and
for IN for
painful JJ painful
flareups NNS flareup
of IN of
osteoarthritis NN osteoarthritis
the DT the
wear-and-tear JJ <unknown>
form NN form
of IN of
the DT the
ailment NN ailment
which WDT which
affects VVZ affect
an DT an
estimated JJ estimated
million CD million
people NNS people
. SENT .
// NN <unknown>
0258 CD @card@
d0605enyt19990406.0038 NN <unknown>
until IN until
now RB now
these DT these
patients NNS patient
and CC and
others NNS other
in IN in
chronic JJ chronic
pain NN pain
have VHP have
relied VVN rely
mostly RB mostly
on IN on
nonsteroidal JJ nonsteroidal
anti-inflammatory JJ anti-inflammatory
drugs NNS drug
or CC or
nsaids NNS <unknown>
sold VVD sell
over-the-counter JJ over-the-counter
or CC or
at IN at
prescription NN prescription
strength NN strength
under IN under
brand NN brand
names NNS name
such JJ such
as IN as
advil NN <unknown>
daypro NN <unknown>
lodine NN <unknown>
and CC and
naprosyn NN Naprosyn
. SENT .
// NN <unknown>
0259 CD @card@
d0605enyt19990406.0038 NN <unknown>
the DT the
drugs NNS drug
are VBP be
generally RB generally
effective JJ effective
but CC but
can MD can
be VB be
harmful JJ harmful
with IN with
constant JJ constant
use NN use
. SENT .
// NN <unknown>
0260 CD @card@
d0605enyt19990406.0038 NN <unknown>
a DT a
study NN study
of IN of
arthritis NN arthritis
patients NNS patient
showed VVD show
at IN at
least JJS least
hospitalizations NNS hospitalization
and CC and
deaths NNS death
a DT a
year NN year
from IN from
nsaid-related JJ <unknown>
problems NNS problem
. SENT .
// NN <unknown>
0261 CD @card@
d0605enyt19990406.0038 NN <unknown>

drugs NNS drug
block NN block
production NN production
of IN of
an DT an
enzyme NN enzyme
called VVD call
which WDT which
produces VVZ produce
painful JJ painful
inflammation NN inflammation
. SENT .
// NN <unknown>
0262 CD @card@
d0605enyt19990406.0038 NN <unknown>
unlike IN unlike
nsaids NNS <unknown>
however RB however
they PP they
do VVP do
not RB not
interfere VV interfere
with IN with
the DT the
closely RB closely
related VVN relate
enzyme NN enzyme
which WDT which
helps VVZ help
build VV build
a DT a
protective JJ protective
lining NN lining
in IN in
the DT the
stomach NN stomach
and CC and
gastrointestinal JJ gastrointestinal
tract NN tract
. SENT .
// NN <unknown>
0263 CD @card@
d0605enyt19990406.0038 NN <unknown>
searle NP Searle
began VVD begin
selling VVG sell
celebrex NN <unknown>
for IN for
per IN per
milligram NN milligram
pill NN pill
less RBR less
than IN than
expected VVN expect
but CC but
still RB still
more JJR more
than IN than
some DT some
hmos NNS <unknown>
might MD might
be VB be
willing JJ willing
to TO to
pay VV pay
especially RB especially
for IN for
patients NNS patient
without IN without
a DT a
demonstrated VVN demonstrate
problem NN problem
with IN with
nsaids NNS <unknown>
. SENT .
// NN <unknown>
0264 CD @card@
d0605enyt19990406.0038 NN <unknown>
the DT the
inhibitors NNS inhibitor
meanwhile RB meanwhile
are VBP be
not RB not
the DT the
only JJ only
new JJ new
possibilities NNS possibility
for IN for
fighting VVG fight
pain NN pain
potent JJ potent
alternative NN alternative
to TO to
morphine NN morphine
in IN in
cases NNS case
of IN of
severe JJ severe
pain NN pain
. SENT .
// NN <unknown>
0265 CD @card@
d0605enyt19990406.0038 NN <unknown>
derived VVN derive
from IN from
sea NN sea
snail NN snail
venom NN venom
it PP it
is VBZ be
designed VVN design
to TO to
block VV block
calcium NN calcium
channels NNS channel
in IN in
nerve NN nerve
cells NNS cell
of IN of
the DT the
spine NN spine
interrupting VVG interrupt
the DT the
flow NN flow
of IN of
pain NN pain
signals NNS signal
to TO to
the DT the
brain NN brain
. SENT .
// NN <unknown>
0266 CD @card@
d0605enyt19990406.0038 NN <unknown>
medicines NNS medicine
called VVD call
biologic JJ biologic
response NN response
modifiers NNS modifier
soaks VVZ soak

up RP up
excess JJ excess
tumor NN tumor
necrosis NN necrosis
factor NN factor
a DT a
chemical NN chemical
secreted VVN secrete
by IN by
immune JJ immune
cells NNS cell
involved VVN involve
in IN in
joint JJ joint
inflammations NNS inflammation
. SENT .
// NN <unknown>
0267 CD @card@
d0605enyt19990406.0038 NN <unknown>
the DT the
drug NN drug
made VVN make
by IN by
immunex NP Immunex
corp. NP Corp.
// NN <unknown>
0268 CD @card@
d0605enyt19990406.0038 NN <unknown>
of IN of
seattle NN Seattle
was VBD be
approved VVN approve
by IN by
the DT the
fda NP <unknown>
last JJ last
for IN for
use NN use
in IN in
adults NNS adult
. SENT .
// NN <unknown>
0269 CD @card@
d0605enyt19990406.0038 NN <unknown>
new JJ new
studies NNS study
show VVP show
that IN/that that
it PP it
may MD may
also RB also
be VB be
effective JJ effective
in IN in
childhood NN childhood
forms NNS form
of IN of
arthritis NN arthritis
. SENT .
// NN <unknown>
0270 CD @card@
d0605enyt19990406.0038 NN <unknown>
epilepsy NN epilepsy
drug NN drug
that WDT that
has VHZ have
been VBN be
shown VVN show
to TO to
work VV work
against IN against
hard-to-treat JJ <unknown>
nerve NN nerve
pain NN pain
from IN from
shingles NNS shingle
or CC or
diabetes NN diabetes
. SENT .
// NN <unknown>
0271 CD @card@
d0605enyt19990406.0038 NN <unknown>
germany NN <unknown>
to TO to
treat VV treat
parkinson NN <unknown>
disease NN disease
can MD can
ease VV ease
the DT the
pain NN pain
of IN of
diabetic JJ diabetic
neuropathy NN neuropathy
. SENT .
// NN <unknown>
0272 CD @card@
d0605enyt19990406.0038 NN <unknown>
the DT the
drug NN drug
blocks VVZ block
nmda NN <unknown>
receptors NNS receptor
nerve NN nerve
cell NN cell
sites NNS site
that WDT that
help VVP help
relay NN relay
pain NN pain
signals NNS signal
. SENT .
// NN <unknown>

Annexe 8. Des phrases renforcées par la connaissance ontologique

d0605enyt19990406.0038

type

painkiller

inhibitors

hit

pharmacy

shelves

//

d0605enyt19990406.0038

drugs

effective

fighting

pain

traditional

medications

ibuprofen

aspirin

offer

advantage

side

effects

//

d0605enyt19990406.0038

major

plus

people

pain

relievers

risk

problems

stomach

bleeding

ulcers

//

d0605enyt19990406.0038

drugs

celebrex

developed

monsanto

gd

searle

pharmaceutical

unit

pfizer

launched

commercially

month

//

d0605enyt19990406.0038

drug

type

merck

vioxx

awaiting

approval

food

drug

administration

market

summer

bladder

vesica

amnion

amniotic_sac

amnios

pericardial_sac

saccule

sacculus

chorion

brawn

brawniness

muscularity

sinew

pain

relievers

tendonitis

tenonitis

tennis_elbow

lateral_epicondylitis

lateral_humeral_epicondylitis

tenosynovitis

tendosynovitis

tendonous_synovitis

sinew

Achilles_tendon

tendon_of_Achilles

hamstring

hamstring_tendon

brawn

brawniness

muscularity

sinew

heftiness

//

d0605enyt19990406.0038

industry

analysts

view

drugs

potential

blockbusters

command

percent

market

common

heftiness

ivory

pearl

off-white

//

d0605enyt19990406.0038

drugs

designed

primarily

rheumatoid

arthritis

afflicts

million

people

united

states

painful

flareups

osteoarthritis

form

ailment

affects

estimated

million

people

sort

form

variety

description

stripe

color

colour

like

ilk

brand

make

genus

like

the_like

the_likes_of

art_form

type

genre

manner

style

antitype

model

flavor

flavour

species

gout

gouty_arthritis

urarthritis

osteoarthritis

degenerative_arthritis

degenerative_joint_disease

rheumatoid_arthritis

atrophic_arthritis

rheumatism

spondylarthritis

archer

bowman

shark

therapist

healer

efficiency_expert

efficiency_engineer

analyst

technocrat

horticulturist

plantsman

calculator

reckoner

figurer

estimator

computer

shot

shooter

ace

adept

champion

sensation

maven

mavin

virtuoso

genius

hotshot

star

superstar

whiz

whizz

wizard

wiz

observer

commentator

cosmetologist

nerd

black_belt

antiquary

antiquarian

archaist

authority

computer_expert

computer_guru

lapidary

lapidarist

pteridologist

jurist

legal_expert

anatomist

agronomist

cabalist

kabbalist

arbiter

supreme_authority

analyst

mythologist

past_master

exegete

talent

logician

logistician

out-and-outer

geographer

parliamentarian

mnemonist

technician

all-rounder

all_arounder

scout

pathfinder

guide

investigator

climatologist

veteran

old-timer

oldtimer

old_hand

warhorse

old_stager

stager

prosthetist

genealogist

specialist

specializer

specialiser

//

d0605enyt19990406.0038

patients

chronic

pain

relied

nonsteroidal

drugs

nsaids

sold

prescription

strength

brand

names

advil

daypro

lodine

naprosyn

//

d0605enyt19990406.0038

drugs

generally

effective

harmful

constant

//

d0605enyt19990406.0038

study

arthritis

patients

showed

hospitalizations

deaths

year

problems

//

d0605enyt19990406.0038

drugs

block

production

enzyme

called

produces

painful

inflammation

//

d0605enyt19990406.0038

nsaids

interfere

closely

related

enzyme

helps

build

protective

lining

stomach

gastrointestinal

tract

//

d0605enyt19990406.0038

searle

began

selling

celebrex

per

milligram

pill

expected

hmos

pay

patients

demonstrated

problem

nsaids

//

d0605enyt19990406.0038

inhibitors

possibilities

fighting

pain

potent

alternative

morphine

cases

severe

pain

//

d0605enyt19990406.0038

derived

sea

snail

venom

designed

block

calcium

channels

nerve

cells

spine

interrupting

flow

pain

signals

brain

//

d0605enyt19990406.0038

medicines

called

biologic

response

modifiers

soaks

excess

tumor

necrosis

factor

chemical

secreted

immune

cells

involved

joint

inflammations

//

d0605enyt19990406.0038

drug

immunex

corp

rule

ruler

size_stick

measuring_cup

//

d0605enyt19990406.0038

seattle

approved

fda

adults

//

d0605enyt19990406.0038

studies

show

effective

childhood

forms

arthritis

//

d0605enyt19990406.0038

epilepsy

drug

shown

work

nerve

pain

shingles

diabetes

hammer

hammering

pounding

corpulence

stoutness

adiposis

weighting

gene

cistron

lethal_gene

suppressor

suppresser

suppressor_gene

suppresser_gene

modifier

modifier_gene

repressor_gene

allele

allelomorph

proto-oncogene

mutant_gene

dominant_gene

transgene

linkage_group

linked_genes

polygene

nonallele

regulatory_gene

regulator_gene

recessive_gene

structural_gene

homeotic_gene

X-linked_gene

Y-linked_gene

holandric_gene

genetic_marker

operator_gene

oncogene

transforming_gene

//

d0605enyt19990406.0038

germany

treat

parkinson

disease

ease

pain

diabetic

neuropathy

extinction

experimental_extinction

aversive_conditioning

classical_conditioning

counter_conditioning

operant_conditioning

diagnosing

urinalysis

uranalysis

blood_typing

medical_diagnosis

promise

rainbow

historic_period

antiquity

reign

turn_of_the_century

golden_age

Jazz_Age

eld

age_of_consent

drinking_age

majority

legal_age

voting_age

old_age

years

eld

geezerhood

dotage

second_childhood

senility

mannequin

manikin

mannikin

manakin

//

d0605enyt19990406.0038

drug

blocks

nmda

receptors

nerve

cell

sites

relay

pain

signals

//

Annexe 9. Extraction du résumé

9.1 Documents à résumer

Document 1

```
<DOC>
<DOCNO> APW19990102.0062 </DOCNO>
<DATE_TIME> 1999-01-02 05:31:53 </DATE_TIME>
<BODY>
<CATEGORY> financial </CATEGORY>
<HEADLINE> Celebrex Ruling Benefits Merck </HEADLINE>
<TEXT>
<P>
          NEW YORK (AP) -- Monsanto will be first to sell a revolutionary
type of arthritis medication, but its battle with Merck &AMP; Co. in
the high stakes painkiller drug wars is far from over.
</P>
<P>
          Celebrex, made by Monsanto's Searle pharmaceutical unit, on
Thursday became the first ``cox-2'' inhibitor to win Food and Drug
Administration approval. The drugs have been touted as possibly the
next wonder pills because of their promise to be easier on
patients' stomachs than existing medications.
</P>
<P>
          But Monsanto was dealt a blow when the FDA declared there is no
proof that Celebrex actually is safer for patients' stomachs than
older painkillers. As a result, when Celebrex hits retail outlets
next month it must bear the same warning about side effects as many
of its older competitors.
</P>
<P>
          Merck, which is awaiting FDA approval for its Vioxx drug,
another cox-2 pill, hopes to get a less stern warning label. One
reason Merck was behind Monsanto in seeking federal approval is
because it was waiting for the results of some longer term studies.
</P>
<P>
          The warning label could be a key driver of physician use, and
whether either drug reaches the $1 billion in annual sales that
some Wall Street analysts had forecast.
</P>
<P>
          Merck also hopes to gain another advantage by getting approval
for Vioxx to be taken only once a day for arthritis users. The
recommended dose of Celebrex is once or twice a day for
osteoarthritis, and twice a day for rheumatoid arthritis.
</P>
<P>
          ``Merck may be able to use Monsanto's experience to get a better
label,'' said Sergio Traversa, an analyst with Mehta Partners in
New York.
</P>
<P>
          After the FDA ruling Thursday, shares of Monsanto rose $1.19 to
$47.50. Pfizer, which will be co-marketing Celebrex, fell $1.12 to
$125. Merck shares dropped $1.31 to $147.50.
</P>
<P>
          Merck spokesman John Bloomfield refused to speculate whether the
company has enough research results to get a better warning label
than Monsanto. Merck is working closely with the FDA to demonstrate
that Vioxx has safety advantages over existing drugs, he said.
</P>
<P>
          ``We are pretty darn happy,'' said Scarlett Lee Foster, a
Monsanto spokeswoman. ``I think we got a label we can use...to have
```

the drug widely accepted.''
</P>
<P>
Monsanto is continuing to study Celebrex in hopes of providing FDA with enough evidence to advertise Celebrex as a safer painkiller.
</P>
<P>
Millions of people now depend on aspirin, ibuprofen, and a host of other pills called ``non-steroidal anti-inflammatory drugs,'' or NSAIDs for arthritis and other pains. NSAIDS can cause ulcers, stomach bleeding and other gastrointestinal side effects, and are blamed for causing 107,000 Americans to be hospitalized every year, and for killing 16,500.
</P>
<P>
Monsanto said it will price Celebrex comparably to other prescription strength painkillers that cost about $2.40 a day. With the traditional safety warnings on Celebrex, Monsanto would have faced considerable difficulty convincing insurers to pay any more.
</P>
<P>
Despite the warning label issue, the government approval of Celebrex is a boost for Monsanto, which last summer had to drop plans to merge with American Home Products, maker of Robitussin cough syrup and Advil pain reliever. The $33.6 billion marriage would have been the biggest merger ever in the pharmaceutical industry.
</P>
<P>
Monsanto, based in St. Louis, makes the artificial sweetener Nutrasweet and is also heavily involved in genetic research for agriculture.
</P>
<P>
``This adds credibility to their pipeline,'' said Sano Shimda, president of BioScience Securities.
</P>
<P>
And the FDA approval is a boost for Pfizer, who some analysts feared would grow too dependent on its anti-impotence drug Viagra to spur sales growth.
</P>
<P>
Merck knows that being first to market with a new class of drugs is no guarantee of long-term success. Merck held the lead in cholesterol lowering drugs with its Zocor drug, but Warner Lambert in the last two years has taken the market lead with its Lipitor pill.

</P>
</TEXT>
</BODY>
</DOC>

Document 2

<DOC>
<DOCNO> APW19990318.0145 </DOCNO>
<DATE_TIME> 1999-03-18 05:35:08 </DATE_TIME>
<BODY>
<CATEGORY> financial </CATEGORY>
<HEADLINE> Drug Found To Curb Menstrual Pain </HEADLINE>
<TEXT>
<P>
NEW YORK (AP) -- Vioxx, the highly anticipated new pain pill from Merck & Co. Inc. that promises to be easier on the stomach than aspirin, works well in relieving menstrual and post-surgical pain, according to two company studies unveiled today at a medical meeting.
</P>
<P>
Vioxx is expected to reach pharmacies later this spring after obtaining approval from the federal Food and Drug Administration.
</P>
<P>
Vioxx was as effective as prescription-strength Aleve, an aspirin-type drug, in treating moderate to severe menstrual pain, according to a Merck study of 127 women being presented at the American Society of Clinical Pharmacology and Therapeutics in San Antonio, Texas. The study found pain relief was sustained for at least 12 hours.
</P>
<P>
In a second study of 151 patients who had oral surgery to remove their wisdom teeth, Vioxx proved as effective at postoperative pain relief as ibuprofen. The dental study also showed one 50 milligram Vioxx pill could provide relief for a full 24 hours. Ibuprofen must be taken at least three times a day.
</P>
<P>
Vioxx will likely become the second in a new class of pain medications which work as well as aspirin and other anti-inflammatory medicines but don't cause stomach irritation that can lead to ulcers and other dangerous ailments.
</P>
<P>
Merck, the world's largest drug company, will be playing catch-up to Monsanto Co.'s Celebrex drug, which won U.S. approval on Dec. 31. Celebrex, approved for osteoarthritis and rheumatoid arthritis, has already sold more than 1.1 million prescriptions, making it the biggest new drug of 1999. It has also has become the second-fastest-selling new drug in history after Pfizer Inc.'s impotence pill Viagra.
</P>
<P>
Like Celebrex, Merck is looking to get approval for Vioxx to treat osteoarthritis, which affects more than 16 million Americans. But instead of also seeking indication for rheumatoid arthritis, which affects about 2 million Americans, it wants the FDA to approve Vioxx for acute pain treatment -- a market of more than 40 million Americans.
</P>
<P>
It's for this reason that Merck is giving the FDA data showing how the drug works against different types of pain.
</P>
<P>
Though the risk of stomach ailments is higher when taking aspirin and other anti-inflammatory drugs for prolonged use, a dilemma faced most arthritis sufferers, the gastrointestinal risks exist even with short duration use, experts say.
</P>
<P>
Neil Sweig, an analyst with Southeast Research Partners, said getting the acute pain indication is vital for Merck, though the

```
market for the both Celebrex and Vioxx is bigger with arthritis
because patients with that condition take pain relievers more
often.
</P>
<P>
          An FDA advisory committee meets April 20 to review Merck's
application for Vioxx.


</P>
</TEXT>
</BODY>
</DOC>
```

Document 3

<DOC>
<DOCNO> APW19990420.0246 </DOCNO>
<DATE_TIME> 1999-04-20 10:19:57 </DATE_TIME>
<BODY>
<CATEGORY> usa </CATEGORY>
<HEADLINE> Paper: Celebrex Linked to 10 Deaths </HEADLINE>
<TEXT>
<P>
NEW YORK (AP) -- Monsanto's highly successful painkiller Celebrex has been linked to 10 deaths and 11 cases of gastrointestinal hemorrhages in its first three months on the market, The Wall Street Journal reported today.
</P>
<P>
Half of the 10 people who died suffered from gastrointestinal bleeding or ulcers, according to ``adverse incident'' reports submitted to the Food and Drug Administration that were obtained by the Journal under the Freedom of Information Act.
</P>
<P>
Two other deaths were attributed to heart attacks, one to drug interaction and one to kidney disorder. No cause of death was given for the 10th fatality.
</P>
<P>
A Monsanto spokeswoman said Tuesday that there is no evidence that Celebrex actually caused the deaths or other health problems in people taking the drug. More than 2 million consumers have taken Celebrex. ``You can't draw any conclusions from the adverse incident reports,'' said spokeswoman Scarlett Lee Foster.
</P>
<P>
The Journal did not specify the sources of the adverse event reports, which could come from health professionals, consumers or the drug company itself.
</P>
<P>
Celebrex, manufactured by St. Louis-based Monsanto Co.'s G.D. Searle & Co. subsidiary, went on the market in January to treat osteoarthritis and rheumatoid arthritis.
</P>
<P>
Celebrex was touted by Monsanto as an effective pain reliever much like ibuprofen, but was much less likely to cause severe stomach problems such as bleeding ulcers.
</P>
<P>
So far it has been a gigantic success: 2.5 million prescriptions have been filled in its first 13 weeks on the market, compared with the record 2.7 million prescriptions of anti-impotency drug Viagra filled during its first three months.
</P>
<P>
Robert DeLap, director of an FDA office of drug evaluation, told the Journal that more research needs to be done before coming to a conclusion about Celebrex's safety. ``Do we think there's a signal that the product poses some special risk? No, not at the moment.''
</P>
<P>
Searle officials told the Journal they remain excited about Celebrex's performance. ``We really feel the drug is performing as expected. The safety profile is what we would expect,'' said Steve Geis, the company's vice president for arthritis clinical research.
</P>
<P>
Geis declined to go into details about any cases of death linked to the drug, but said that many patients taking Celebrex have other illnesses and are taking multiple medications.

```
</P>
</TEXT>
</BODY>
</DOC>
```

9.2 Phrases extraites pour le résumé

9.2.1 Nombre de document(s) : 1, compression = 10%

Monsanto will be first to sell a revolutionary type of arthritis medication, but its battle with Merck & Co. in the high stakes painkiller drug wars is far from over.

Monsanto is continuing to study Celebrex in hopes of providing FDA with enough evidence to advertise Celebrex as a safer painkiller.

“We are pretty darn happy" said Scarlett Lee Foster, a Monsanto spokeswoman.

9.2.2 Nombre de document(s) : 1, compression = 20%

Monsanto will be first to sell a revolutionary type of arthritis medication, but its battle with Merck & Co. in the high stakes painkiller drug wars is far from over.

Monsanto is continuing to study Celebrex in hopes of providing FDA with enough evidence to advertise Celebrex as a safer painkiller.

``We are pretty darn happy," said Scarlett Lee Foster, a Monsanto spokeswoman.

Merck held the lead in cholesterol lowering drugs with its Zocor drug, but Warner Lambert in the last two years has taken the market lead with its Lipitor pill.

As a result, when Celebrex hits retail outlets next month it must bear the same warning about side effects as many of its older competitors.

9.2.3 Nombre de document(s) : 2, compression = 10%

Merck, the world's largest drug company, will be playing catch-up to Monsanto Co.'s Celebrex drug, which won U.S. approval on Dec. 31. Celebrex, approved for osteoarthritis and rheumatoid arthritis, has already sold more than 1.1 million prescriptions, making it the biggest new drug of 1999.

a market of more than 40 million Americans.

Monsanto will be first to sell a revolutionary type of arthritis medication, but its battle with Merck & Co. in the high stakes painkiller drug wars is far from over.

Monsanto is continuing to study Celebrex in hopes of providing FDA with enough evidence to advertise Celebrex as a safer painkiller.

9.2.4 Nombre de document(s) : 2, compression = 20%

Merck, the world's largest drug company, will be playing catch-up to Monsanto Co.'s Celebrex drug, which won U.S. approval on Dec. 31. Celebrex, approved for osteoarthritis and rheumatoid arthritis, has already sold more than 1.1 million prescriptions, making it the biggest new drug of 1999.

a market of more than million Americans.

Monsanto will be first to sell a revolutionary type of arthritis medication, but its battle with Merck & Co. in the high stakes painkiller drug wars is far from over.

Monsanto is continuing to study Celebrex in hopes of providing FDA with enough evidence to advertise Celebrex as a safer painkiller.

“We are pretty darn happy” said Scarlett Lee Foster, a Monsanto spokeswoman.

Merck held the lead in cholesterol lowering drugs with its Zocor drug, but Warner Lambert in the last two years has taken the market lead with its Lipitor pill.

As a result, when Celebrex hits retail outlets next month it must bear the same warning about side effects as many of its older competitors.

Celebrex, made by Monsanto's Searle pharmaceutical unit, on Thursday became the first “cox-2” inhibitor to win Food and Drug Administration approval.

9.2.5 Nombre de document(s) : 3, compression = 10%

Merck, the world's largest drug company, will be playing catch-up to Monsanto Co.'s Celebrex drug, which won U.S. approval on Dec. 31. Celebrex, approved for osteoarthritis and rheumatoid arthritis, has already sold more than 1.1 million prescriptions, making it the biggest new drug of 1999.

a market of more than million Americans.

Celebrex, manufactured by St. Louis-based Monsanto Co.'s G.D. Searle & Co. subsidiary, went on the market in January to treat osteoarthritis and rheumatoid arthritis.

So far it has been a gigantic success: 2.5 million prescriptions have been filled in its first 13 weeks on the market, compared with the record 2.7 million prescriptions of anti-impotency drug Viagra filled during its first three months.

Monsanto will be first to sell a revolutionary type of arthritis medication, but its battle with Merck & Co. in the high stakes painkiller drug wars is far from over.

Merck knows that being first to market with a new class of drugs is no guarantee of long-term success.

9.2.6 Nombre de document(s) : 3, compression = 20%

Merck, the world's largest drug company, will be playing catch-up to Monsanto Co.'s Celebrex drug, which won U.S. approval on Dec. 31. Celebrex, approved for osteoarthritis and rheumatoid arthritis, has already sold more than 1.1 million prescriptions, making it the biggest new drug of 1999.

a market of more than million Americans.

Celebrex, manufactured by St. Louis-based Monsanto Co.'s G.D. Searle & Co. subsidiary, went on the market in January to treat osteoarthritis and rheumatoid arthritis.

So far it has been a gigantic success: 2.5 million prescriptions have been filled in its first 13 weeks on the market, compared with the record 2.7 million prescriptions of anti-impotency drug Viagra filled during its first three months.

Monsanto will be first to sell a revolutionary type of arthritis medication, but its battle with Merck & Co. in the high stakes painkiller drug wars is far from over.

Merck knows that being first to market with a new class of drugs is no guarantee of long-term success.

Merck, which is awaiting FDA approval for its Vioxx drug, another cox-2 pill, hopes to get a less stern warning label.

Vioxx, the highly anticipated new pain pill from Merck & Co. Inc. that promises to be easier on the stomach than aspirin, works well in relieving menstrual and post-surgical pain, according to two company studies unveiled today at a medical meeting.

Merck held the lead in cholesterol lowering drugs with its Zocor drug, but Warner Lambert in the last two years has taken the market lead with its Lipitor pill.

“We really feel the drug is performing as expected”.

FDA advisory committee meets April 20 to review Merck's application for Vioxx.

Annexe 10. Diagramme de l'interface pour l'obtention des phrases du résumé

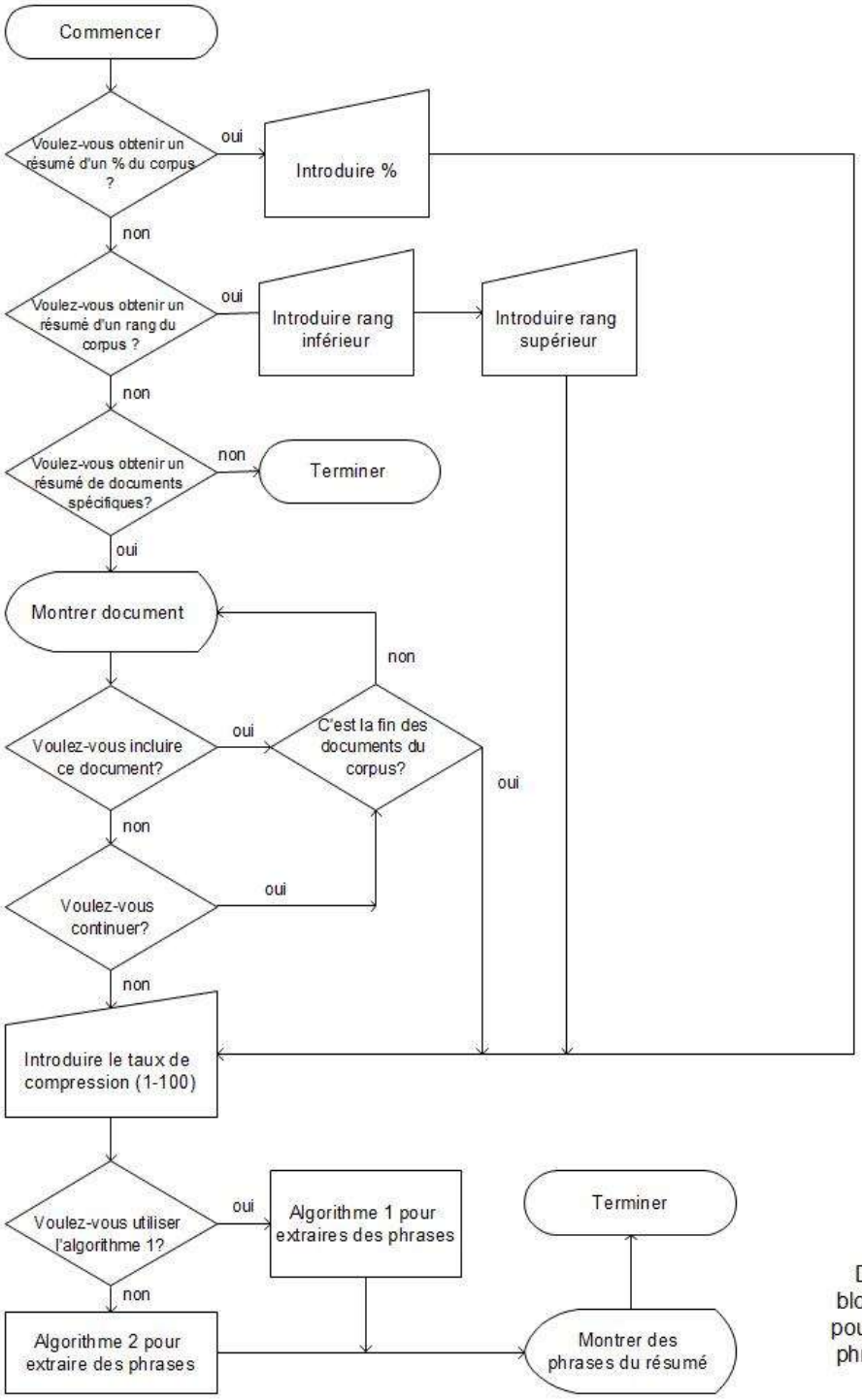


Diagramme en blocs de l'interface pour l'obtention des phrases du résumé