Impacts de la variation du nombre de traits discriminants sur la catégorisation des documents

Dominic Forest avec la collaboration de Astrid van Hoeydonck,
Danny Létourneau et Martin Bélanger

Université de Montréal – École de bibliothéconomie et des sciences de l'information
C.P. 6128, Succursale Centre-Ville, Montréal, Québec, Canada, H3C 3J7
dominic.forest@umontreal.ca

Résumé – Abstract

Cet article fait état des résultats générés par deux démarches de fouille de textes. La première démarche a été réalisée afin d'assister l'identification du caractère objectif ou subjectif d'un corpus d'articles de journaux. La seconde démarche a été appliquée afin d'assister l'identification du parti politique d'un parlementaire. Dans les deux cas, les démarches ont été réalisées en utilisant l'algorithme des k plus proches voisins (*k-nearest neighbor*). Cet article porte sur la dimension informationnelle de la démarche de fouille de textes. Il présente l'impact de la variation du nombre de traits discriminants sur les résultats de la catégorisation automatique des documents.

This paper presents the results of two text mining processes. The first process aimed at assisting the identification of the objective or subjective characteristic of a corpus of newspaper articles. The second process aimed at assisting the identification of the political party of a corpus of political interventions. Both processes were accomplished using the k-nearest neighbor algorithm. This paper focuses on the informational dimensions of text mining processes. It reports on the impact of the variation of features on automatic text categorization results.

Mots-Clés – Keywords

Catégorisation automatique de documents, traits discriminants, variation, application.

Automatic text classification, feature selection, variation, application.

1 Introduction

Depuis une dizaine d'années, le domaine de la fouille de textes s'avère un territoire de recherche des plus actifs. Ainsi, de nombreuses initiatives de recherche ont tenté de développer des applications informatiques permettant d'extraire des patrons d'informations caractéristiques de documents, d'en identifier le contenu thématique ou même d'en extraire certaines informations précises. Les applications développées intègrent des concepts et des techniques provenant de plusieurs disciplines académiques bien établies. Ainsi, les récents développements dans le domaine de la fouille de textes reposent très souvent sur des concepts et des traitements éprouvés issus des domaines de l'intelligence

artificielle et de l'apprentissage machines, de la linguistique informatique, des sciences de l'information, etc.

Plusieurs facteurs ont motivé les recherches dans ce domaine. Parmi ces facteurs, on retrouve au premier plan le nombre croissant de documents disponibles en format numérique. Nous disposons désormais de très volumineux corpus de documents textuels en format numérique, dont l'exploitation ne peut être réalisée sans avoir recours à des techniques évoluées de traitement de l'information. Ainsi, les conséquences des initiatives de numérisation ont des répercussions directes sur le développement d'applications visant à assister la recherche, l'analyse, la structuration, la gestion et la diffusion de l'information présente dans les documents textuels.

Les développements technologiques dans le domaine de la fouille de textes ont pris la forme de logiciels propriétaires permettant de combiner différentes opérations de fouille afin d'effectuer des traitements plus ou moins complexes sur des corpus de documents textuels non structurés. Parmi les principaux logiciels de fouille de textes commerciaux disponibles, on retrouve entre autres *Text Analyst* (Megaputer), *Text Miner* (SAS), *PASW Modeler* (SPSS). Plus récemment, plusieurs efforts ont été consacrés au développement de plates-formes de fouille de données modulaires et flexibles offrant aux utilisateurs la possibilité de combiner différents modules afin de générer rapidement des chaînes de traitement plus adaptées à leurs besoins (*RapidMiner*, *Weka*, etc.). Ces plates-formes ont d'abord été conçues pour traiter des données structurées. Cependant, elles sont souvent accompagnées de modules supplémentaires (prétraitement, filtrage linguistique, conversion numérique, etc.) leur permettant de traiter efficacement des documents textuels non structurés.

Les développements technologiques dans le domaine de la fouille de textes ont été notables au cours des dernières années, tant en ce qui a trait aux algorithmes de traitement, qu'aux performances et aux interfaces utilisateurs. Cependant, malgré ces avancées technologiques, nous constatons que certaines difficultés demeurent en ce qui a trait à l'utilisation des techniques de fouille de textes. Au-delà des particularités et des difficultés propres à chaque corpus, deux difficultés font toujours obstacle à l'utilisation des outils de fouille de textes. La première de ces difficultés est d'ordre méthodologique. Dans de nombreux cas, les utilisateurs ont une idée relativement précise des informations qu'ils souhaitent extraire à partir des données textuelles dont ils disposent. Cependant, la démarche grâce à laquelle ils pourraient extraire optimalement les informations recherchées est trop souvent inconnue. Il existe certes un ensemble d'opérations potentiellement utiles afin d'atteindre l'objectif souhaité, mais la méthodologie à déployer pour atteindre l'objectif n'a pas été identifiée. Dans plusieurs projets de fouille de textes, les questions suivantes demeurent souvent sans réponse satisfaisante : quels sont les algorithmes à appliquer afin d'assister le plus efficacement la réalisation d'une tâche spécifique ? Par exemple, quelle serait la démarche méthodologique optimale à mettre en œuvre si l'on souhaite assister l'extraction d'opinions à partir de large corpus bilingues ? Est-il utile d'appliquer un algorithme de lemmatisation sur les données ? Si oui, est-il préférable de le faire après ou avant la suppression des mots fonctionnels ? À quelle étape doit être appliqué l'algorithme d'extraction des entités nommées ? Est-il nécessaire de procéder préalablement à une opération de marquage morphosyntaxique des données linguistiques initiales ? Voilà autant de questions d'ordre technique et méthodologique auxquelles il est actuellement très difficile de fournir des réponses éclairées, *a fortiori* lorsqu'elles sont posées dans des contextes applicatifs bien spécifiques. Ainsi, nous constatons que la dimension méthodologique de la fouille de textes n'a pas été une préoccupation importante dans ce domaine. Nous disposons de techniques de fouille efficaces, mais nous n'en connaissons malheureusement trop peu sur les modalités d'application des différents algorithmes afin d'assister efficacement les tâches auxquelles nous sommes confrontées.

La seconde difficulté est étroitement liée à la première. Elle concerne les paramètres à spécifier au niveau de chaque étape du processus général de fouille de textes. À l'instar de la dimension méthodologique de la fouille de texte, les paramètres à spécifier pour chaque algorithme sont inévitablement dépendants de l'objectif à atteindre, des particularités inhérentes aux documents à traiter et du contexte dans lequel le traitement est réalisé. Cependant, malgré la particularité de chaque

contexte applicatif, peu d'informations sont connues concernant les paramètres optimaux à spécifier pour chaque étape du processus de fouille. Par exemple, peu d'informations pratiques sont disponibles concernant les paramètres de prétraitement et de filtrage à mettre en œuvre dans les premières étapes du processus. Quels sont les paramètres qu'il serait souhaitable de mettre en œuvre pour effectuer un filtrage statistique des données linguistiques extraites du corpus que l'on souhaite analyser ? En dessous de quel seuil de fréquence les mots retenus comme traits discriminants doivent-ils être supprimés afin d'accroitre les performances d'un algorithme de classification automatique ? Voilà autant de questions auxquelles il est actuellement difficile de répondre. Dans le domaine de la fouille de données (que les données soient de nature textuelle ou non), la majorité des opérations que l'on exécute impliquent que l'on spécifie un ou plusieurs paramètres. Il en va ainsi tant au début du processus (au moment du filtrage des données) qu'à l'étape finale de validation. Souvent, trop peu d'informations rigoureuses sont connues concernant les paramètres optimaux à mettre en œuvre afin de réaliser des tâches en contexte spécifique.

2 Objectifs

Dans cet article, nous présentons la démarche que nous avons employée, ainsi que les résultats que nous avons obtenus lors de notre participation à l'édition 2009 du DEfi Fouille de Textes (DEFT'09). En outre, nous présentons nos travaux sous l'angle du second problème que nous avons identifié concernant l'utilisation des processus de fouille de textes en contexte applicatif réel. Plus spécifiquement, l'objectif de notre démarche dans ce projet consiste à identifier, dans des contextes expérimentaux précis, l'impact de la fluctuation du nombre de traits discriminants retenus pour représenter des documents qui sont soumis à une opération de catégorisation automatique.

Dans le cadre de DEFT'09, nous avons exécuté deux tâches qui nous ont permis d'explorer l'impact de la variation du nombre de traits discriminants sur les résultats de la catégorisation automatique. La première de ces tâches consiste à prédire le caractère objectif ou subjectif d'un corpus d'articles de journaux. L'identification du caractère objectif ou subjectif d'un article de journal est une opération importante dans le domaine de la fouille de textes. En effet, elle est à la base de plusieurs applications complexes de traitement de l'information, parmi lesquelles figure au premier plan l'identification automatique d'opinions (*sentiment analysis*) (Lui, 2007).

La seconde tâche consiste à identifier le parti politique d'un parlementaire. Cette tâche est comparable à celle qui consiste à prédire l'auteur d'un document (*authorship attribution*). Plutôt que d'identifier l'auteur d'un document, l'objectif de cette seconde tâche consiste à identifier automatiquement le parti politique auquel se rattache d'auteur d'une intervention politique. L'identification d'auteur a fait l'objet de nombreux travaux dans le domaine de la fouille de textes. Comme l'a clairement souligné (Juola, 2008), l'identification d'auteur est une tâche complexe qui fait intervenir plusieurs dimensions dans la description des données textuelles. Malgré les récents développements dans ce domaine, cette tâche pose encore de nombreux problèmes, tant théoriques que pratiques.

3 Méthodologie

La démarche méthodologique que nous avons employée pour accomplir les deux tâches est inspirée de celle que l'on retrouve au cœur de nombreux projets de fouille de données. Cette démarche est principalement de nature numérique. Pour des raisons théoriques et pratiques, nous avons volontairement réduit au minimum le nombre d'opérations faisant intervenir des dimensions linguistiques. La démarche repose sur le modèle vectoriel pour le traitement des documents (Salton, 1988 ; Memmi, 2000). Elle est composée de cinq principales étapes.

La première étape de tout processus de fouille de données réside dans le développement ou la constitution d'un corpus de documents. Il est essentiel que le corpus de documents soit constitué en tenant compte des objectifs à atteindre par le processus de fouille. À l'étape de constitution du corpus, quatre grandes familles de caractéristiques doivent être évaluées et prises en considération. La constitution d'un corpus à des fins de fouille de textes implique certains choix en ce qui concerne les caractéristiques 1) générales (provenance, taille, date de création, etc.), 2) technologiques (support, format, etc.), 3) informationnelles (thématiques et sujets abordés) et 4) linguistiques (langue, genre, registres, etc.) des documents. Dans le cadre de DEFT'09, la constitution du corpus a été entièrement prise en charge par le comité organisateur, avec le concours de l'Agence pour l'Évaluation et la Distribution des Ressources Linguistique (ELDA). Nous n'avons apporté aucune modification aux documents qui nous ont été fournis par le comité organisateur.

La seconde étape de la démarche a consisté à extraire, à filtrer et à normaliser le lexique du corpus. L'opération de filtrage du lexique est composée traditionnellement de plusieurs sous-opérations. La première d'entre elles consiste à supprimer certains mots non pertinents pour l'analyse. Le filtrage du lexique peut être effectué à l'aide de plusieurs techniques, certaines étant de nature linguistique, d'autres de nature statistique. Une première opération a pour but de supprimer l'ensemble des mots fonctionnels présents dans le texte. Ce processus est réalisé en retirant les termes figurant dans une liste prédéfinie de mots fonctionnels. Il est aussi souhaitable d'appliquer certains filtres statistiques au lexique du corpus afin d'en éliminer les unités qui, tout en ne figurant pas dans la liste des mots fonctionnels, ne sont pas pertinentes pour l'analyse. La pertinence des termes est très étroitement associée à leur potentiel discriminant. Ainsi, il importe de supprimer les mots dont la fréquence est supérieure ou inférieure à certains seuils (souvent déterminés empiriquement). Dans un dernier temps, il est d'usage d'appliquer un processus de généralisation sensible aux variantes sémantiques et syntaxiques présentes dans le corpus. Il importe alors d'appliquer au lexique du corpus une opération de lemmatisation. L'opération de lemmatisation est réalisée généralement d'abord en effectuant un marquage morphosyntaxique des différents lexèmes à analyser, ensuite en comparant ceux-ci à un dictionnaire. Ce processus permet de dégager une liste de lemmes propres à une langue donnée. S'il est impossible de lemmatiser les données à traiter (par manque de ressources linguistiques, par exemple), il est souhaitable de recourir à un processus d'amputation des terminaisons (*stemming*), lequel génère une liste de *stems* (racines).

La troisième étape de la démarche consiste à convertir le corpus initial dans un format pouvant être traité par les algorithmes de fouille. Cette opération est réalisée en structurant les documents du corpus en une matrice de vecteurs dans laquelle chaque document (ou segment de document) est représenté par l'absence ou la présence, binaire ou pondérée, de chaque unité lexicale retenue à l'étape précédente.

C'est à la quatrième étape de la démarche que sont réalisées les opérations permettant plus spécifiquement d'extraire et de structurer les informations présentes dans le corpus. Dans une perspective de fouille de textes, la majorité des opérations d'extraction et de structuration des informations sont réalisées en utilisant des algorithmes développés dans les domaines de l'intelligence artificielle et de l'apprentissage machine.

Les tâches de catégorisation automatique – qui consistent à attribuer une ou plusieurs catégories à chaque document d'un corpus – sont traditionnellement accomplies en utilisant des algorithmes d'apprentissage supervisés. La principale particularité des techniques supervisées réside dans leur capacité à projeter certaines caractéristiques des documents préalablement connues et apprises par le système sur un ensemble de documents pour lesquels les mêmes caractéristiques ne sont pas encore connues. En vertu de cette particularité, les techniques supervisées impliquent donc d'abord une phase d'apprentissage (réalisée sur un corpus d'apprentissage) et, ensuite, une phase de test (ou d'application) lors que laquelle l'apprentissage effectué par le système est projeté sur de nouveaux documents (en contexte de test ou d'application concrète). Dans le cadre de DEFT'09, nous avons d'abord exploré deux algorithmes de classification : l'algorithme des k plus proches voisins (Manning

et Schütze, 1999) et un classifieur bayésien naïf (Manning et Schütze, 1999). Lors de la phase de test, seul l'algorithme des k plus proches voisins a été retenu.

La cinquième étape de la démarche réside dans l'interprétation, l'évaluation et l'intégration des résultats générés par les algorithmes de fouille de textes. Les opérations d'interprétation et d'évaluation sont des plus complexes, car elles sont dépendantes de plusieurs facteurs extrinsèques au processus de traitement des documents textuels. Les algorithmes supervisés peuvent être évalués selon les mesures classiques de rappel et de précision.

Par ailleurs, l'interprétation des résultats des algorithmes de fouille ne peut être dictée par aucun cadre théorique qui ferait abstraction du contexte dans lequel l'opération de fouille est réalisée. Finalement, les résultats doivent normalement faire l'objet d'un processus d'intégration à l'intérieur d'une application finale plus complexe dans laquelle le processus de fouille ne constitue d'une étape bien précise. Les applications finales intégrant des processus de fouille de textes sont de plus en plus nombreuses et variées. Parmi celles-ci, on trouve entre autres les applications de veille scientifique, de gestion électronique des documents et de recherche d'informations. La figure 1, inspirée de Fayyad *et al.* (1996), présente les principales étapes de la méthodologie générique de fouille de textes.

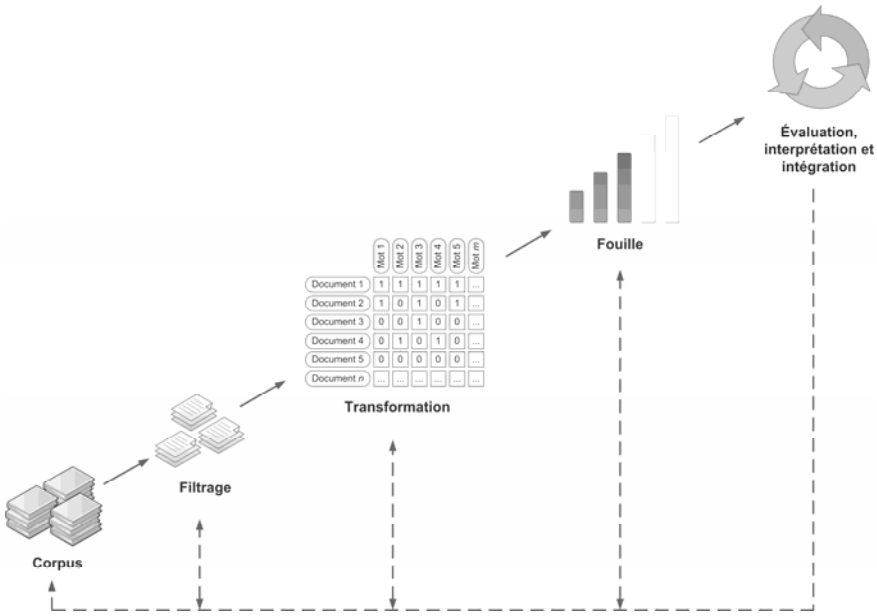

Figure 1. La démarche méthodologique (Forest, 2009).

4 Corpus

4.1 Corpus 1

Le corpus associé à la tâche de prédiction du caractère objectif ou subjectif a été divisé en respectant un ratio 2/3 – 1/3 classique dans les domaines de la recherche d'informations et du traitement automatique des langues (TAL). Les deux tiers du corpus ont été exploités à des fins d'apprentissage, alors que le tiers restant a été utilisé à des fins de test.

Le sous-corpus d'apprentissage est composé de 25 176 documents français. Il est composé de 12 511 590 mots (occurrences) et de 75 302 formes. Le lexique de ce sous-corpus a été lemmatisé. Nous en avons aussi supprimé les mots fonctionnels. Finalement, les hapax, ainsi que les formes présentes dans

plus de 25% des documents ont été supprimés. À la suite de ces prétraitements, le lexique restant était composé de 51 864 mots. Par la suite, nous avons utilisé une mesure de Chi2 pour identifier des sous-ensembles de mots fortement discriminants qui ont servi à décrire numériquement les documents.

Le sous-corpus de test est composé de 16 788 documents français. Il est composé de 8 499 931 mots (occurrences) et de 110 297 formes. Afin de catégoriser automatiquement les documents du corpus de test, les documents ont préalablement été convertis numériquement en utilisant les mots retenus à l'étape d'apprentissage.

4.2 Corpus 2

Le corpus associé à la tâche de prédiction du parti politique a été divisé en respectant aussi un ratio 2/3 – 1/3. Les deux tiers du corpus ont été exploités à des fins d'apprentissage, alors que le tiers restant a été utilisé à des fins de test.

Le sous-corpus d'apprentissage est composé de 19 370 documents français. Il est composé de 7 208 721 mots (occurrences) et de 53 531 formes. Le lexique de ce sous-corpus a été lemmatisé. Nous en avons aussi supprimé les mots fonctionnels. Finalement, les hapax, ainsi que les formes présentes dans plus de 50% des documents ont été supprimés. À la suite de ces prétraitements, le lexique restant était composé de 21 698 mots. Par la suite, nous avons encore une fois utilisé une mesure de Chi2 pour identifier des sous-ensembles de mots fortement discriminants qui ont servi à décrire numériquement les documents.

Le sous-corpus de test est composé de 12 914 documents français. Il est composé de 4 799 665 mots (occurrences) et de 46 242 formes. Afin de catégoriser automatiquement les documents du corpus de test, les documents ont préalablement été convertis numériquement en utilisant les mots retenus à l'étape d'apprentissage.

5 Résultats

Cette section présente les résultats que nous avons obtenus pour les deux tâches que nous avons réalisées. Nous présentons d'abord les résultats obtenus lors de la phase d'apprentissage, puis lors de la phase de test. Tel que nous l'avons mentionné précédemment, nous avons fait varier plusieurs paramètres lors de la phase d'apprentissage. Les résultats que nous présentons font état de ces variations. Les résultats obtenus lors de la phase d'apprentissage sont quantifiés en utilisant les mesures de rappel et de précision. Afin d'évaluer les performances du processus de catégorisation au moment de la phase d'apprentissage, une méthode d'échantillonnage croisée a été appliquée (*10-fold cross-validation*). L'ensemble des expérimentations menées dans ce projet a été réalisé avec le logiciel *WordStat* (Provalis Research).

En ce qui concerne les résultats de la phase de test, nous présentons les résultats que nous avons obtenus selon trois configurations expérimentales. Ces résultats ont été calculés par le comité organisateur de DEFT'09.

5.1 Tâche 1. Prédiction du caractère objectif ou subjectif

5.1.1 Phase d'apprentissage

Expérimentations d'apprentissage utilisant les paramètres suivants (figures 2 et 3) :

- Nombre de traits discriminants : entre 10 000 et 50 000 mots discriminants (avec un incrément de 10 000)
- Méthodes de représentation des traits : fréquence et fréquence pondérée par IDF
- Algorithmes employés : k plus proches voisins (k=1 et k=5) et classifieur bayésien naïf

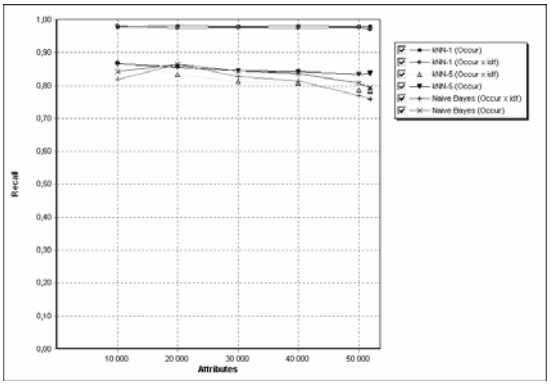

Figure 2. Tâche 1 – performances (rappel) du premier ensemble d'expérimentations d'apprentissage.

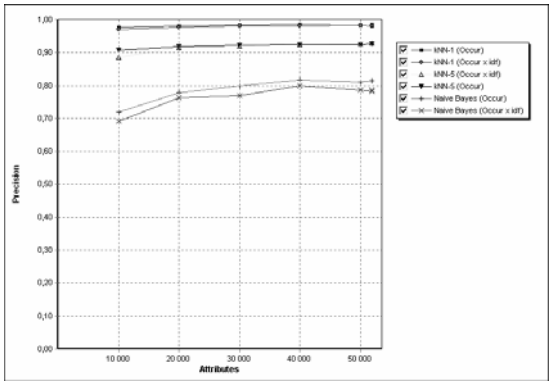

Figure 3. Tâche 1 – performances (précision) du premier ensemble d'expérimentations d'apprentissage.

Expérimentations d'apprentissage utilisant les paramètres suivants (figures 4 et 5) :

- Nombre de traits discriminants : entre 1 000 et 10 000 mots discriminants (avec un incrément de 1 000)
- Méthodes de représentation des traits : fréquence et fréquence pondérée par IDF
- Algorithmes employés : k plus proches voisins (k=1 et k=5) et classifieur bayésien naïf

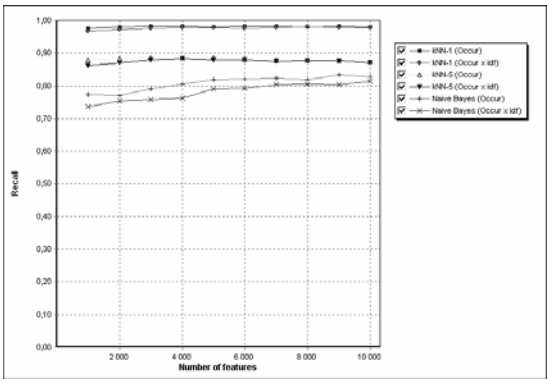

Figure 4. Tâche 1 – performances (rappel) du deuxième ensemble d'expérimentations d'apprentissage.

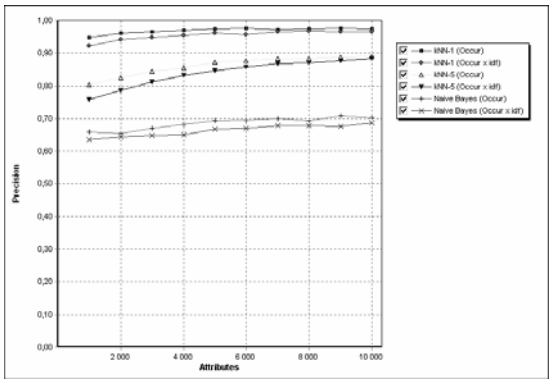

Figure 5. Tâche 1 – performances (précision) du deuxième ensemble d'expérimentations d'apprentissage.

Expérimentations d'apprentissage utilisant les paramètres suivants (figures 6 et 7) :

- Nombre de traits discriminants : entre 100 et 3 000 mots discriminants (avec un incrément de 100)
- Méthodes de représentation des traits : fréquence et fréquence pondérée par IDF
- Algorithmes employés : k plus proches voisins (k=1 et k=5) et classifieur bayésien naïf

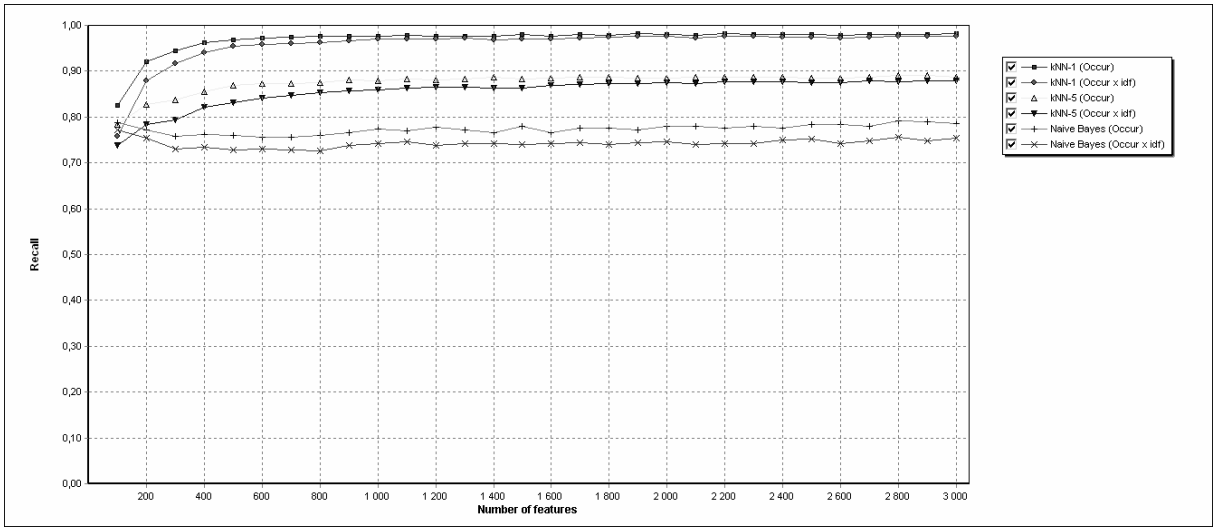

Figure 6. Tâche 1 – performances (rappel) du troisième ensemble d'expérimentations d'apprentissage.

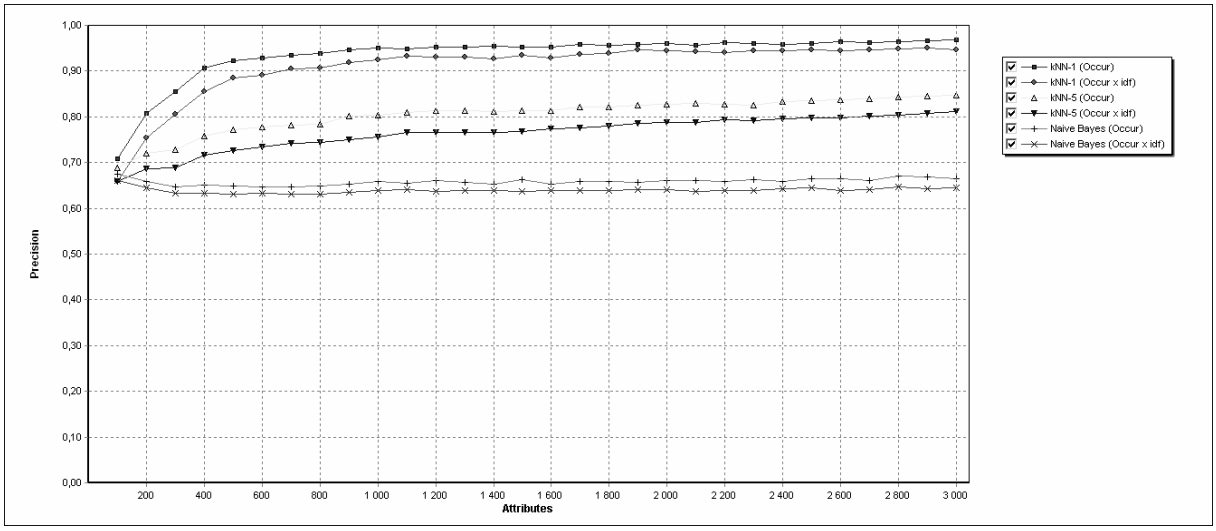

Figure 7. Tâche 1 – performances (précision) du troisième ensemble d'expérimentations d'apprentissage.

Durant la phase d'apprentissage, les meilleures performances ont été obtenues en utilisant 40 000 mots discriminants représentés uniquement par leur fréquence. Ces mots ont été employés pour décrire les documents qui ont été soumis à l'algorithme des k plus proches voisins avec le paramètre k=2. Ainsi, selon en utilisant cet algorithme avec ce paramètre, nous avons obtenu un taux de rappel de 98.09% et un taux de précision de 98.46%. En contrepartie, les pires performances ont été obtenues en utilisant 800 mots discriminants représentés par leur fréquence pondérée qui ont été soumis au classifieur bayésien naïf. Ainsi, selon en utilisant cet algorithme avec ce paramètre, nous avons obtenu un taux de rappel de 63% et un taux de précision de 72.59%.

5.1.2 Phase de test

Lors de la phase de test, nous avons mené trois exécutions en ne faisant varier que le nombre de traits discriminants. En effet, nous avons constaté lors de la phase d'apprentissage que les meilleurs résultats ont toujours été obtenus sans pondérer la fréquence des mots et en utilisant l'algorithme des k plus proches voisin avec le paramètre k=1. Les résultats que nous avons obtenus lors de ces exécutions sont les suivants :

Exécution de test 1 utilisant les paramètres suivants :

- Nombre de traits discriminants : 6 000
- Méthodes de représentation des traits : fréquence
- Algorithmes employés : k plus proches voisins (k=1)

Performances globales : rappel = 77.80%, précision = 73.80%, f-mesure = 75.70%

Performances spécifiques (catégorie *Objectif*) : rappel = 88.40%, précision = 92.80%

Performances spécifiques (catégorie *Subjectif*) : rappel = 67.30%, précision = 54.70%

Exécution de test 2 utilisant les paramètres suivants :

- Nombre de traits discriminants : 20 000
- Méthodes de représentation des traits : fréquence
- Algorithmes employés : k plus proches voisins (k=1)

Performances globales : rappel = 77.90%, précision = 77.60%, f-mesure = 77.80%

Performances spécifiques (catégorie *Objectif*) : rappel = 92.20%, précisions = 92.40%

Performances spécifiques (catégorie *Subjectif*) : rappel = 63.60%, précision = 62.90%

Exécution de test 3 utilisant les paramètres suivants :

- Nombre de traits discriminants : 40 000
- Méthodes de représentation des traits : fréquence
- Algorithmes employés : k plus proches voisins (k=1)

Performances globales : rappel = 77.30%, précision = 79.00%, f-mesure = 78.10%

Performances spécifiques (catégorie *Objectif*) : rappel = 93.40%, précision = 92.00%

Performances spécifiques (catégorie *Subjectif*) : rappel = 61.20%, précision = 65.90%

5.2 Tâche 2. Prédiction du parti politique

5.2.1 Phase d'apprentissage

Expérimentations d'apprentissage utilisant les paramètres suivants (figures 8 et 9) :

- Nombre de traits discriminants : entre 5 000 et 20 000 mots discriminants (avec un incrément de 5 000)
- Méthodes de représentation des traits : fréquence et fréquence pondérée par IDF
- Algorithmes employés : k plus proches voisins (k=1 et k=5) et classifieur bayésien naïf

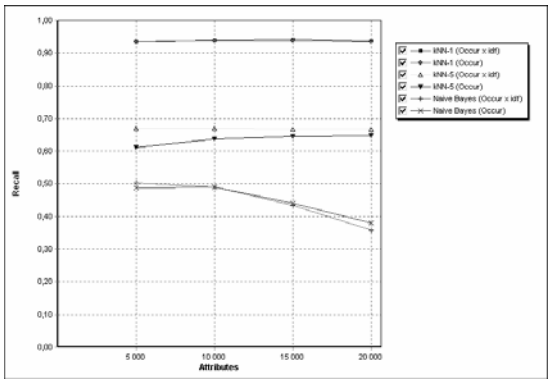

Figure 8. Tâche 2 – performances (rappel) du premier ensemble d'expérimentations d'apprentissage.

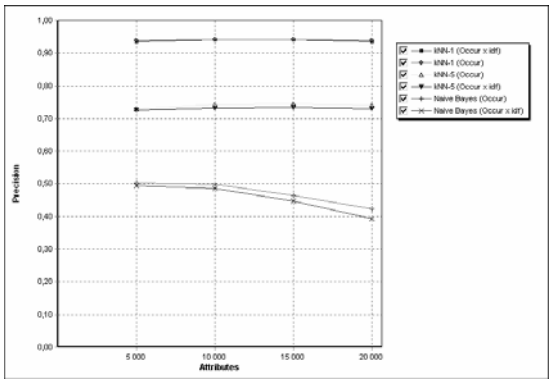

Figure 9. Tâche 2 – performances (précision) du premier ensemble d'expérimentations d'apprentissage.

Expérimentations d'apprentissage utilisant les paramètres suivants (figures 10 et 11) :

- Nombre de traits discriminants : entre 1 000 et 10 000 mots discriminants (avec un incrément de 1 000)
- Méthodes de représentation des traits : fréquence et fréquence pondérée par IDF
- Algorithmes employés : k plus proches voisins (k=1, k=2, k=3, k=4 et k=5)

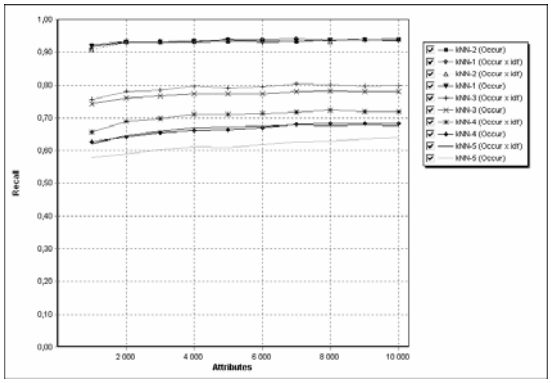

Figure 10. Tâche 2 – performances (rappel) du deuxième ensemble d'expérimentations d'apprentissage.

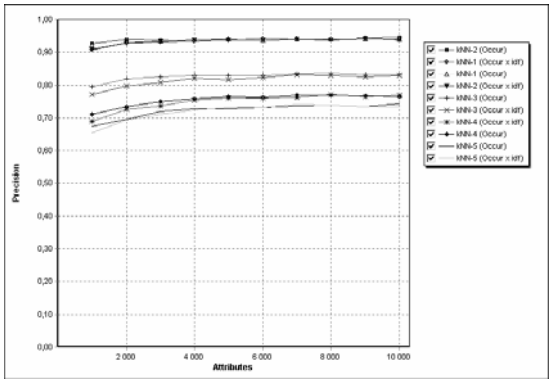

Figure 11. Tâche 2 – performances (précision) du deuxième ensemble d'expérimentations d'apprentissage.

Expérimentations d'apprentissage utilisant les paramètres suivants (figures 12 et 13) :

- Nombre de traits discriminants : entre 100 et 2 000 mots discriminants (avec un incrément de 1 00)
- Méthodes de représentation des traits : fréquence et fréquence pondérée par IDF
- Algorithmes employés : k plus proches voisins (k=1, k=2, k=3, k=4 et k=5)

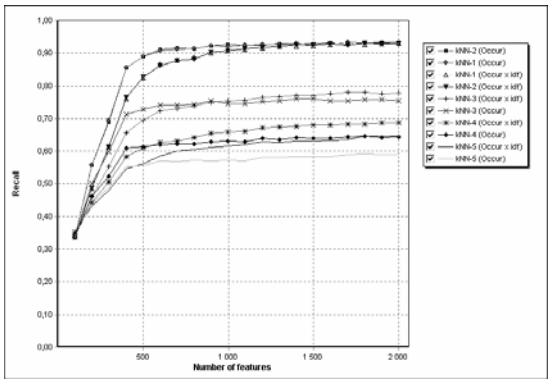

Figure 12. Tâche 2 – performances (rappel) du troisième ensemble d'expérimentations d'apprentissage.

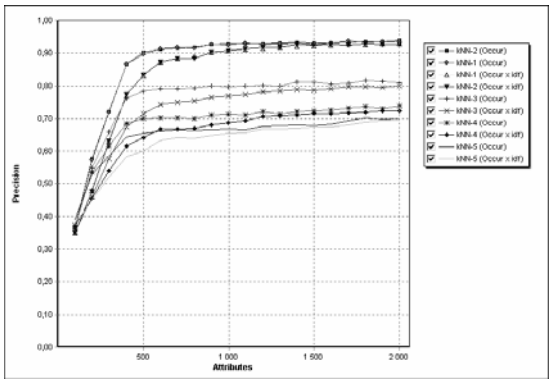

Figure 13. Tâche 2 – performances (précision) du troisième ensemble d'expérimentations d'apprentissage.

Durant la phase d'apprentissage de la tâche 3, les meilleures performances ont été obtenues en utilisant 10 000 mots discriminants représentés uniquement par leur fréquence. Ces mots ont été employés pour décrire les documents qui ont été soumis à l'algorithme des k plus proches voisins avec le paramètre k=2. Ainsi, selon en utilisant cet algorithme avec ce paramètre, nous avons obtenu un taux de rappel de 94.12% et un taux de précision de 94.53%. En contrepartie, les pires performances ont été obtenues en utilisant 100 mots discriminants représentés par leur fréquence. Ainsi, selon en utilisant cet algorithme avec ce paramètre, nous avons obtenu un taux de rappel moyen de 33.57% et un taux de précision moyen de 36.34%.

5.2.2 Phase de test

Lors de la phase de test, nous avons mené trois exécutions en faisant varier le nombre et la pondération des traits discriminants, ainsi que la valeur du paramètre *k*. Les résultats que nous avons obtenus lors de ces exécutions sont les suivants :

Exécution de test 1 utilisant les paramètres suivants :

- Nombre de traits discriminants : 5 000
- Méthodes de représentation des traits : fréquence pondérée par IDF
- Algorithmes employés : k plus proches voisins (k=1)

Performances globales : rappel = 32.20%, précision = 31.90%, f-mesure = 32.00%

Performances spécifiques (catégorie *ELDR*) : rappel = 18.90%, précision = 21.00%

Performances spécifiques (catégorie *GUE-NGL*) : rappel = 39.30%, précision = 34.50%

Performances spécifiques (catégorie *PPE-DE*) : rappel = 43.70%, précision = 44.70%

Performances spécifiques (catégorie *PSE*) : rappel = 36.00%, précision = 36.50%

Performances spécifiques (catégorie *VERT-ALE*) : rappel = 23.30%, précision = 22.60%

Exécution de test 2 utilisant les paramètres suivants :

- Nombre de traits discriminants : 10 000
- Méthodes de représentation des traits : fréquence
- Algorithmes employés : k plus proches voisins (k=2)

Performances globales : rappel = 33.20%, précision = 34.60%, f-mesure = 33.90%

Performances spécifiques (catégorie *ELDR*) : rappel = 23.10%, précision = 23.60%

Performances spécifiques (catégorie *GUE-NGL*) : rappel = 33.20%, précision = 42.20%

Performances spécifiques (catégorie *PPE-DE*) : rappel = 49.80%, précision = 45.20%

Performances spécifiques (catégorie *PSE*) : rappel = 39.40%, précision = 37.00%

Performances spécifiques (catégorie *VERT-ALE*) : rappel = 20.70%, précision = 25.20%

Exécution de test 3 utilisant les paramètres suivants :

- Nombre de traits discriminants : 15 000
- Méthodes de représentation des traits : fréquence pondérée par IDF

- Algorithmes employés : k plus proches voisins (k=1)

Performances globales : rappel = 33.30%, précision = 33.50%, f-mesure = 33.40%

Performances spécifiques (catégorie *ELDR*) : rappel = 20.20%, précision = 20.50%

Performances spécifiques (catégorie *GUE-NGL*) : rappel = 37.60%, précision = 38.40%

Performances spécifiques (catégorie *PPE-DE*) : rappel = 46.20%, précision = 46.20%

Performances spécifiques (catégorie *PSE*) : rappel = 38.30%, précision = 36.90%

Performances spécifiques (catégorie *VERT-ALE*) : rappel = 24.30%, précision = 25.50%

6 Discussions

Les résultats que nous avons obtenus lors de la phase d'apprentissage étaient des plus prometteurs (plus de 98% de rappel et de précision pour la tâche 1 et plus de 94% de rappel et de précision pour la tâche 3). Lors de la phase de test, les meilleurs résultats obtenus pour la tâche 1 sont de 77.30% au niveau du rappel et de 79.00% au niveau de la précision. Dans le cadre de la tâche 3, les meilleurs résultats obtenus sont bien en deçà des performances que nous étions en mesure d'obtenir lors de la phase d'apprentissage. Ainsi, pour la tâche 3, les meilleurs résultats obtenus sont de 33,30% au niveau du rappel et de 33,50% au niveau de la précision.

Les performances finales sont plutôt décevantes, surtout lorsque nous les comparons aux performances élevées obtenues lors de la phase d'apprentissage. Il est difficile d'identifier les causes exactes de ces performances finales. Il est cependant raisonnable de proposer que les modèles de catégorisation mis à l'épreuve lors de la phase de test sont caractérisés par un surapprentissage (*overfitting*). Ainsi, les modèles employés (surtout lors de la phase de test de la tâche 3) se sont avérés trop étroitement liés aux données initiales, peu généralisables et difficilement applicables aux données de test. Nous sommes d'avis que les performances observées découlent en partie des choix effectués lors du filtrage du lexique du corpus. Ainsi, il est probable que certains des mots qui ont été retenus pour décrire les documents en raison de leur présence élevée dans une des catégories (au moment de l'apprentissage) se sont avérés être beaucoup moins discriminants dans les données de test.

Une comparaison plus approfondie de la distribution des mots retenus dans les modèles de catégorisation à l'intérieur des données d'apprentissage et des données de test pourrait nous permettre de mieux comprendre pourquoi les performances initiales n'ont pu être reproduites sur les données de test. Nous sommes d'avis que cette piste d'explication pourrait être plus riche qu'une explication qui reposerait principalement sur l'algorithme de catégorisation employé.

En ce qui concerne les paramètres à spécifier lors de tâches de catégorisation automatique de documents textuels, les expérimentations que nous avons menées nous indiquent, dans un premier temps, que les performances des algorithmes sont en partie liées aux nombres de traits discriminants retenus pour décrire les documents à traiter. Ainsi, on constate une corrélation les performances et le nombre de traits discriminants employés. Plus le nombre de traits discriminants employés est élevé, meilleurs sont les performances du système. Au départ, cette amélioration est très prononcée. Elle devient plus subtile à partir de quelques milliers de traits discriminants (mais elle est néanmoins toujours présente). Ce phénomène a d'abord été observé sur les données d'apprentissage, puis, dans une moindre ampleur, sur les données de test.

Au niveau des algorithmes de catégorisation, l'algorithme des k plus proches voisins a toujours généré de meilleurs résultats que le classifieur bayésien naïf. Nos expériences nous portent donc à croire que

le classifieur bayésien naïf est peu efficace pour des tâches de catégorisation des documents textuels, lesquelles font inévitablement intervenir un nombre élevé de traits discriminants. En ce qui concerne l'algorithme des k plus proches voisins, nous constatons qu'il est optimal lorsque la valeur du paramètre *k* (paramètre spécifiant le nombre de « voisins » dans l'espace vectoriel auquel les éléments à catégoriser sont comparés) est très faible (idéalement 1 ou 2).

Références

Fayyad, U., G. Piatetsky-Shapiro et P. Smyth (1996), From data mining to knowledge discovery in databases, *AI Magazine*, vol. 1, pp. 37-54.

Forest, D. (2009, sous presse), Vers une nouvelle génération d'outils d'analyse et de recherche d'informations, *Documentation et bibliothèque*.

Juola, P. (2008), *Authorship attribution*. Now Publishers Inc.

Liu, B. (2007), *Web data mining: exploring hyperlinks, contents, and usage data*, London, Springer.

Manning, C. D. et H. Schütze (1999), *Foundations of statistical natural language processing*. Cambridge (Mass.): MIT Press.

Memmi, D. (2000), *Le modèle vectoriel pour le traitement de documents*. Grenoble, Cahiers Leibniz, no 2000-14.

Salton, G. (1989), *Automatic Text Processing*. Reading (Mass.), Addison-Wesley.