Modélisation de documents combinant texte et image : application à la catégorisation et à la recherche d'information multimédia

Christophe Moulin

École Doctorale ED488, Sciences, Ingénierie, Santé
Faculté des Sciences et Techniques de l'Université Jean-Monnet de Saint-Étienne
Laboratoire Hubert-Curien, UMR CNRS 5516

Modélisation de documents combinant texte et image : application à la catégorisation et à la recherche d'information multimédia

Thèse en vue de l'obtention du diplôme de
DOCTEUR DE L'UNIVERSITÉ DE SAINT-ÉTIENNE
mention INFORMATIQUE

Christophe MOULIN

Cette thèse a été soutenue le 22 juin 2011
devant le jury constitué de :

Annie	MORIN	présidente
Patrick	GROS	rapporteur
Philippe	MULHEM	rapporteur
Matthieu	CORD	examinateur
Christine	LARGERON	directrice
Christophe	DUCOTTET	codirecteur
Cécile	BARAT	encadrante
Mathias	GÉRY	encadrant

Remerciements

La reconnaissance silencieuse ne sert à personne (Gladys Bronwyn Stern) ; je tiens donc à remercier tous ceux et celles qui d'une façon ou d'une autre m'ont aidé à réaliser ce travail de thèse.

Je souhaite adresser mes premiers remerciements à mes quatre encadrants : j'aimerais tout d'abord remercier ma directrice, Christine Largeron, pour m'avoir proposé ce sujet de thèse ainsi que Christophe Ducottet pour l'avoir codirigée. Merci également à Cécile Barat et Mathias Géry pour m'avoir co-encadré pendant ces quatre années. Travailler avec quatre personnes n'a pas été toujours simple, mais cette thèse n'en a été que plus enrichissante tant au niveau scientifique qu'au niveau relationnel. Je les remercie donc pour tout cela ainsi que pour m'avoir offert un cadre de travail très agréable.

Je voudrais ensuite remercier les membres du jury et plus particulièrement Patrick Gros et Philippe Muhlem qui ont accepté de rapporter mon travail ainsi que pour leurs remarques et leurs commentaires sur mon manuscrit. Merci enfin à Annie Morin d'avoir présidé le jury et Mathieu Cord pour son rôle d'examinateur.

J'aimerais remercier tous mes collègues et les personnes du laboratoire qui ont contribué de près ou de loin au bon déroulement de ce travail : Adriana, Amélie, Anne-Laure, Aurélien, Baptiste, Catherine, Chahrazed, Claude, Colin, Dalila, David, Élisa, Éric, Fabien, Fabrice, Florent, Florian, François, Franck, Hazaël, Jean, Jeanine, Jean-Christophe, Jean-Philippe, Julien, Léo, Marc, Mattias, Nathalie, Patricia, Patrick, Pierre, Philippe, Richard, Sabri, Stéphanie, Tung et tous ceux que j'oublie.

Et parce que j'ai peur de me faire taper, je n'oublie évidemment pas Émilie et Laurent avec qui j'ai débuté et partagé un bureau où le travail et la bonne humeur cohabitaient. Même si nous n'avons pas réussi à gagner le gros lot, merci à vous deux pour tous les bons moments passés ensemble.

J'aurais sûrement encore d'autres questions, ou d'autres raccourcis à te demander, mais merci Frédéric pour tout : awk, bash, CJ, diff, emacs, firefox, GR, hexdump, isketch, J, KK, lpr, μT, NC, oldschool, Python, QT, RL, SQ, tubulo, unsort, Victor, WeChall, xkcd, λ, zcat... Merci également Thierry pout tous tes conseils et tes scripts. Je remercie également aragorn, frodon, gandalf, gimli, legolas, magohamoth, peregrin, sam et titeuf sans qui mes expérimentations tourneraient sûrement encore.

Merci enfin à toutes les personnes que je n'ai pas nommé mais qui étaient là pour me soutenir toutes ces années. Je pense à ma famille, mes parents, Maryline, Samuel, le tout petit Noé et tous mes amis : Amale et Valérie qui ont toujours répondu à mes appels, parfois très longs, quand j'en avais besoin ; Jean-Vincent qui a été plus

qu'un simple partenaire de jeu télévisé ; Mathieu, Nathalie, Romain et Vincent pour les week-ends d'évasion en espérant aller un jour tous ensemble dans ce célèbre parc d'attraction ; Anthony, Carole, Cécile, David, Nicolas, Qi, Rémi, tous ceux de la Tour pour les cinés, les quiz, les balades et les soirées ; les parachutistes de l'ASPL avec qui j'espère m'envoyer en l'air encore longtemps et parce qu'on est *tellement bien entre nous*. Merci Benoît d'avoir été compréhensif et de m'avoir supporté jusqu'au bout.

```
3__ ..4
_4   1.
32:-y11
_1   3_
5.. ..4
```

Table des matières

Introduction

L'augmentation croissante des capacités de production, de stockage et de diffusion des documents multimédias rend l'accès à l'information utile de plus en plus difficile. Dès lors que le nombre de documents qui composent une collection dépasse la centaine, il est fastidieux de rechercher manuellement un document particulier ou un sous-ensemble de cette collection. La mise en place d'outils automatiques d'organisation et de recherche de documents est donc indispensable.

Le développement des nouvelles technologies a également entraîné une diversification de ces documents. Ces derniers peuvent comporter du texte, des images, du son, des vidéos ou une combinaison de ces différents éléments. Afin d'organiser et de rechercher au mieux ces documents, les outils doivent prendre en compte cette diversité. Cette thèse s'intéresse à la représentation de documents multimédias permettant d'exploiter les différentes informations contenues dans les documents et plus particulièrement les informations textuelle et visuelle.

Après avoir introduit le contexte général dans lequel se situe cette recherche, nous détaillerons les problématiques auxquelles nous nous sommes intéressés. Enfin, nous présenterons les objectifs et l'organisation de ce mémoire.

Contexte du travail : accès à l'information multimédia

Avec le développement des terminaux mobiles et embarqués, les utilisateurs sont de plus en plus assistés par des outils qui tentent d'exploiter le maximum d'information disponible pour répondre à leurs besoins. Un utilisateur qui souhaite par exemple réaliser un achat, peut se voir proposer une liste de produits en fonction de ses préférences et de son historique de commandes. Ces systèmes de recommandation sont très présents sur les sites marchands et exploitent tous les achats effectués par l'ensemble des utilisateurs du système. Avec l'augmentation des plateformes de réseaux sociaux, les relations entre les différents utilisateurs sont également très utilisées pour conseiller ces derniers. Toutes ces informations sont très hétérogènes et rendent difficile la mise en place de systèmes permettant de répondre à tous les besoins de tous les utilisateurs.

Dans la suite, nous nous intéresserons plus particulièrement aux situations où un utilisateur a un besoin particulier d'information. Pour satisfaire ce besoin, il dispose d'une collection de documents composés d'images, de texte ou d'une combinaison des deux. Pour l'assister et lui éviter d'étudier chaque document de la collection, il peut

utiliser des outils automatiques de catégorisation et de recherche d'information. Ces outils permettent d'organiser les documents de la collection comme l'illustre la figure 1. Dans certains cas, l'utilisateur peut exprimer son besoin sous la forme d'une requête composée de quelques mots ou d'une ou plusieurs images. Les systèmes de recherche d'information (SRI) ont alors pour objectif de fournir une liste triée de documents sensés répondre au besoin de l'utilisateur. Dans d'autres cas, l'utilisateur ne peut pas exprimer son besoin par une requête et il préférera rechercher dans un sous-ensemble de documents de la collection. Ce sous-ensemble sera formé de quelques documents ayant par exemple un thème commun dans lesquels l'utilisateur sera susceptible de trouver la réponse à son besoin. C'est la catégorisation de documents qui, à l'aide d'un classifieur, regroupe les documents similaires en catégories.

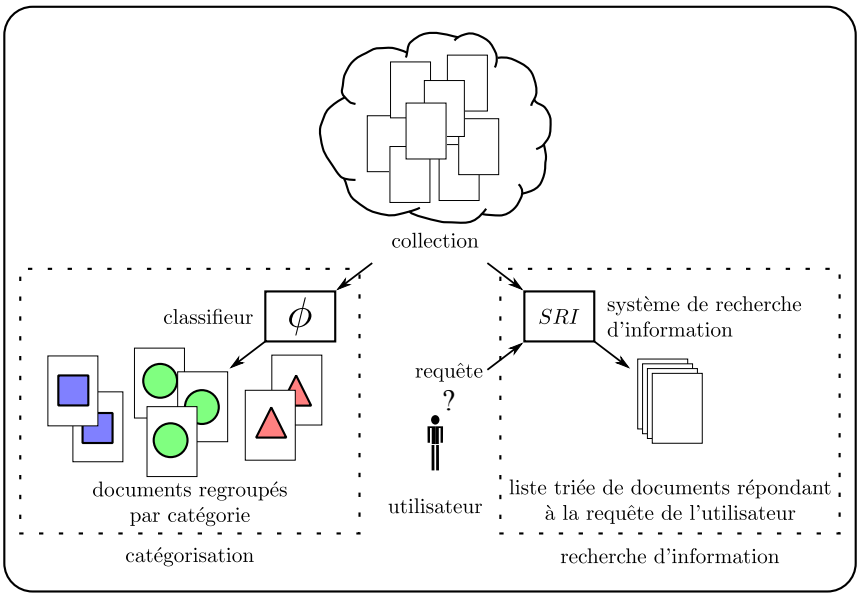


FIG. 1 – Présentation de la catégorisation et de la recherche d'information pour répondre au besoin d'un utilisateur.

Quel que soit le contexte de travail, nous utiliserons une représentation des documents par sac de mots. Si l'ordre des mots dans un document permet d'analyser et de comprendre en détail son contenu, il n'est pas forcément nécessaire pour extraire son sujet principal. En effet, quelques mots clés présents dans le document suffisent souvent à saisir le sujet de ce dernier. Pour des documents textuels, l'approche par sac de mots correspond à une représentation sous la forme d'un ensemble non ordonné des mots extraits du texte. L'ensemble des mots possibles forme alors un vocabulaire de mots textuels. Cette idée a ensuite été étendue à la représentation des images. Contrairement aux documents textuels où le vocabulaire peut être construit directement à partir des mots présents dans les documents, l'approche par sac de mots pour les images nécessite une étape préalable de création d'un vocabulaire de mots visuels à partir de caractéristiques locales extraites des images.

Problématiques

La représentation des documents en sacs de mots nécessite la création d'un vocabulaire spécifique pour chaque modalité, texte et image. En fonction de la modalité considérée et du contexte de travail, des problèmes se posent sur :

- la réduction de la taille du vocabulaire ;
- l'extraction et la pondération des mots visuels ;
- la combinaison des informations textuelle et visuelle ;
- la sélection des catégories à associer à un document.

Le vocabulaire textuel se construit le plus simplement en utilisant les mots apparaissant dans les documents de la collection. Cependant, même pour un nombre réduit de documents, le vocabulaire ainsi obtenu peut être de très grande taille. Principalement pour des raisons d'efficacité, mais aussi de performance des algorithmes, il est intéressant de réduire la taille de ce vocabulaire. Dans un contexte de catégorisation, nous nous sommes demandés comment réduire efficacement la taille du vocabulaire en fonction de la distribution des mots qui apparaissent dans des documents appartenant aux mêmes catégories.

À la différence du vocabulaire textuel, les mots visuels ne sont pas aussi bien définis. Trois étapes principales peuvent être distinguées pour créer un vocabulaire visuel : la détection de points d'intérêt, la description de caractéristiques locales autour de ces points d'intérêt et leur quantification. Nous avons étudié les différents problèmes liés à la création d'un vocabulaire visuel : comment détecter efficacement les points d'intérêts dans les images ? Quelles caractéristiques sont à calculer pour extraire le plus d'information utile possible ? Comment créer les mots visuels à partir de ces descriptions locales ? Combien de mots faut-il choisir pour décrire les images ? Comment utiliser ces mots pour représenter les images ?

Les vocabulaires ainsi créés pour chaque modalité permettent de représenter les documents de la collection. Cette représentation est ensuite utilisée pour classer les documents en catégories ou pour les ordonner en fonction de leur pertinence pour une requête posée par un utilisateur.

Nous nous sommes demandés comment combiner les modalités textuelles et visuelles, dans une tâche de recherche d'information multimédia. Dans ce contexte, l'utilisateur exprime son besoin sous la forme d'une requête composée de quelques mots textuels ou d'une ou quelques images. La première approche que nous avons envisagée, consiste à n'utiliser qu'une seule partie, textuelle ou visuelle, de la requête. Notre but a ensuite été de développer une seconde approche exploitant les modalités conjointement. Pour cela, nous avons considéré un système de recherche d'information qui combine linéairement les résultats obtenus par le système pour chaque modalité. Plusieurs questions peuvent alors se poser : est-il possible d'améliorer les résultats en exploitant les différentes informations ? Combien de modalités pouvons-nous combiner ? Quel poids doit être accordé à chaque type d'information (textuelle et visuelle) ?

Enfin dans le contexte le plus simple de la catégorisation de documents, les documents ne sont associés qu'à une seule catégorie. Pour réaliser cette catégorisation, des algorithmes issus du domaine de l'apprentissage automatique sont utilisés pour générer un classifieur qui pour un nouveau document à classer, retourne la catégorie qui est la plus probable. S'il existe plusieurs catégories, nous parlerons de catégorisation multiclasse. Dans certaines applications, dites de catégorisation multilabel, plusieurs catégories peuvent être associées à un même document. Dans ce contexte, nous souhai-

tons exploiter les résultats des algorithmes de catégorisation multiclasse : le problème est alors de sélectionner le nombre de catégories à conserver pour un nouveau document en fonction de la pertinence des catégories retournées par le classifieur.

Objectifs

Le principal objectif de notre travail est de proposer un modèle pour représenter les documents multimédias. Ce modèle doit pouvoir être utilisé pour des documents qui comportent une ou plusieurs images, du texte ou les deux. Il doit être en mesure d'exploiter toutes les informations textuelles et visuelles disponibles et de les combiner pour améliorer les résultats dans des contextes de catégorisation de documents et de recherche d'information.

Afin de valider ce modèle, nos recherches ont été évaluées sur des collections classiques, mais également en participant à des compétitions internationales comme ImageCLEF et INEX XMLMining [Moulin *et al.*, 2008, Moulin *et al.*, 2009, Géry *et al.*, 2009, Largeron *et al.*, 2010].

Organisation du mémoire

Le premier chapitre, consacré à l'état de l'art, introduit tout d'abord les tâches de catégorisation et de recherche d'information ainsi que les mesures d'évaluation associées. Les approches permettant de représenter et de fusionner les informations textuelle et visuelle sont ensuite présentées.

Le deuxième chapitre porte sur la représentation des documents textuels. Dans le contexte de la catégorisation, nous introduisons un nouveau critère mettant en avant les mots les plus représentatifs des catégories dans le but de réduire le vocabulaire textuel [Largeron *et al.*, 2011]. Nous proposons également une nouvelle méthode de sélection du nombre de catégories à associer aux documents dans le cadre multilabel [Largeron et Moulin, 2010].

Le troisième chapitre s'intéresse à la représentation des images et s'inspire des approches classiquement utilisées avec des données textuelles. Les images sont ainsi représentées à l'aide d'un modèle basé sur les sacs de mots visuels. Les différentes étapes de création du vocabulaire visuel sont analysées et une étude sur la pondération des mots visuels ainsi que sur la fusion de différents descripteurs est réalisée dans le cadre de la catégorisation d'images [Moulin *et al.*, 2010a].

Le quatrième chapitre se place dans le contexte de la recherche d'information multimédia et étudie l'apport de l'information visuelle en combinant linéairement les résultats obtenus séparément sur chaque modalité. Différentes approches consistant à apprendre les paramètres de combinaison ont été considérées, soit en effectuant une recherche exhaustive de la valeur optimale des paramètres de combinaison, soit en le calculant de façon analytique [Lemaître *et al.*, 2009, Moulin *et al.*, 2010b, Moulin *et al.*, 2010c].

Chapitre 1

Représentation de documents multimédia

Avec le développement des nouvelles technologies et de l'internet, la recherche s'est intéressée au problème de l'accès à l'information. L'information étudiée dans la suite correspond à des documents multimédias susceptibles de contenir du texte et des images. Ce chapitre est consacré à l'état de l'art de la représentation de ces documents multimédias.

Le problème de recherche de documents dans une collection ainsi que les approches adoptés pour résoudre ce problème seront tout d'abord introduits. Les modèles classiques utilisés pour représenter les données textuelles, puis les images seront ensuite présentés. Enfin, les différentes possibilités permettant de fusionner ces deux types d'information multimédia seront étudiées.

1.1 Positionnement du problème

Deux approches principales peuvent être utilisées pour répondre au problème de la recherche de documents dans une collection : la catégorisation de documents et la recherche d'information. La première consiste à réduire le problème de la taille de la collection en classant en sous-catégories les documents similaires pour n'avoir à chercher que dans un sous-ensemble restreint de documents plutôt que dans la collection complète [Sebastiani, 2002]. Cette approche est par exemple exploitée par les annuaires ou les portails de l'internet qui proposent une liste de sites classés hiérarchiquement en différentes catégories [Dumais et Chen, 2000, Adami *et al.*, 2003]. Les plus connus de ces portails sont Google directory [1], Yahoo directory [2] et Open directory project [3].

La seconde approche, la recherche d'information, consiste à chercher une liste de documents pertinents pour une requête donnée dans l'ensemble des documents de la collection préalablement indexés [van Rijsbergen, 1979, Manning *et al.*, 2008]. L'indexation est une phase qui consiste à associer des termes à des documents ; cela permet en formulant une requête composée de termes de l'index, de retrouver plus facilement les documents. Pour les documents textuels, l'indexation exploite directement les mots extraits des documents. Pour les images, cette indexation n'est pas immédiate et peut

[1] `http://www.google.com/dirhp`
[2] `http://dir.yahoo.com`
[3] `http://www.dmoz.org`

s'effectuer par exemple à l'aide de mots clés associés aux images [Barnard *et al.*, 2003]. Ce procédé d'annotation peut être effectué manuellement ou en utilisant des méthodes de catégorisation [Hanbury, 2008]. Cette approche visant à rechercher des documents indexés est utilisée par les moteurs de recherche comme Google[1], Yahoo[2], Exalead[3].

La démarche générale sous-jacente à ces deux tâches est tout d'abord introduite puis est suivie d'une présentation détaillée de la recherche d'information et de la catégorisation. Un dernier paragraphe revient sur l'importance de la représentation des documents.

1.1.1 Présentation de la démarche générale

Quelle que soit la tâche considérée (recherche d'information ou catégorisation de documents), la démarche générale est la même : dans un premier temps, les documents sont décrits à l'aide d'un modèle de représentation qui permet ainsi de les manipuler plus facilement. Ensuite, des outils de comparaison basés sur des mesures de similarité sont utilisés pour mettre en correspondance les documents. Enfin, la pertinence des résultats issus de différentes méthodes est évaluée à l'aide de plusieurs critères.

Pour la recherche d'information et la catégorisation de documents, l'évaluation des résultats se base sur deux principaux critères qui sont la précision (P) et le rappel (R). La précision mesure la proportion des documents pertinents retrouvés parmi les documents retournés alors que le rappel mesure la proportion de documents pertinents retrouvés parmi les documents à retrouver. La figure 1.1 illustre graphiquement ces deux critères.

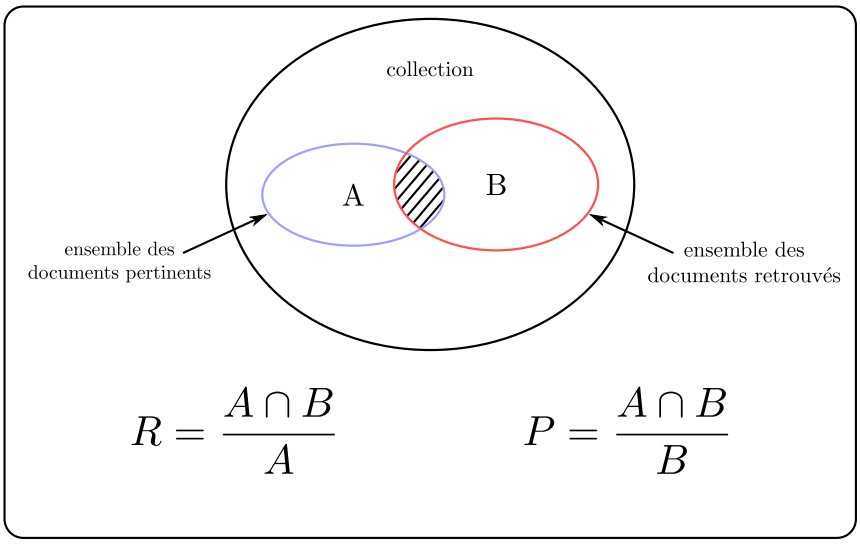


FIG. 1.1 – Illustration des critères de précision et de rappel.

De façon générale, il faut donc proposer des systèmes qui retournent le plus de résultats pertinents, c'est-à-dire ayant un rappel élevé, tout en limitant le nombre d'erreurs, autrement dit ayant une précision élevée.

[1] `http://www.google.com`
[2] `http://www.yahoo.com`
[3] `http://www.exalead.fr`

Un ensemble de documents $\mathcal{D} = \{d_1, \ldots, d_i, \ldots, d_{|\mathcal{D}|}\}$, est appelé collection ou corpus. Le nombre de documents de la collection ou sa taille correspond au cardinal de l'ensemble $\mathcal{D}$ et sera noté $|\mathcal{D}|$. Les documents considérés dans la suite peuvent être composés d'images ou de texte. Quand la nature du document porte à confusion, l'exposant T sera utilisé pour représenter l'information textuelle et l'exposant V pour l'information visuelle.

1.1.2 Recherche d'information

Pour rechercher des documents dans une collection donnée, un utilisateur peut exprimer sous forme de requêtes ses besoins. Pour un ensemble de requêtes $\mathcal{Q} = \{q_1, \ldots, q_k, \ldots, q_{|\mathcal{Q}|}\}$ fournies par un ou plusieurs utilisateurs , le but d'un système de recherche d'information est de retourner pour chaque requête q_k, une liste de documents de $\mathcal{D}$ triée par pertinence. Une requête peut correspondre à du texte, des images ou une combinaison des deux. La partie textuelle d'une requête est généralement formée de quelques mots [O'Keefe et Trotman, 2003, Kamps *et al.*, 2005].

1.1.2.1 Principes de la recherche d'information

Les systèmes de recherche d'information ont été introduits en bibliothéconomie afin d'améliorer les techniques de gestion et d'organisation des bibliothèques [Cleverdon, 1991]. Pour trouver une page particulière dans un livre, l'utilisation d'un index est indispensable. Cette idée a été étendue dans les bibliothèques en utilisant un index permettant de retrouver les livres plus facilement.

L'utilisation des ordinateurs et de l'informatique en général a permis la mise en place d'outils facilitant le traitement de l'information et la création automatique des index. La recherche d'information est devenu un domaine très actif ces dernières années ; L'arrivée de l'internet a nécessité la mise en place d'outils beaucoup plus performants pour traiter des quantités très importantes d'information [Kobayashi et Takeda, 2000].

Un système de recherche d'information possède deux parties principales illustrées par la figure 1.2. La première concerne l'indexation des documents alors que la seconde correspond à la recherche elle-même.

1.1.2.2 Indexation

L'indexation a pour but de bien représenter les documents de la collection afin d'accéder rapidement et efficacement à leur contenu. Lire tous les livres d'une bibliothèque permet de trouver ceux qui contiennent une information particulière. Mais cette solution n'est pas viable et il est nécessaire d'utiliser une indexation pour retrouver plus simplement les livres intéressants pour la recherche envisagée. Une indexation simple, mais néanmoins efficace, est l'utilisation d'un index inversé. À chaque terme est associée la liste des documents qui contiennent ce terme à l'image d'un index présent à la fin d'un livre qui associe à chaque mot clé les pages correspondantes. Pour un terme et un document donnés, des informations complémentaires peuvent être ajoutées selon les besoins comme le nombre d'apparitions du terme dans le document. Ces informations sont ensuite utilisées pour représenter les documents de la collection.

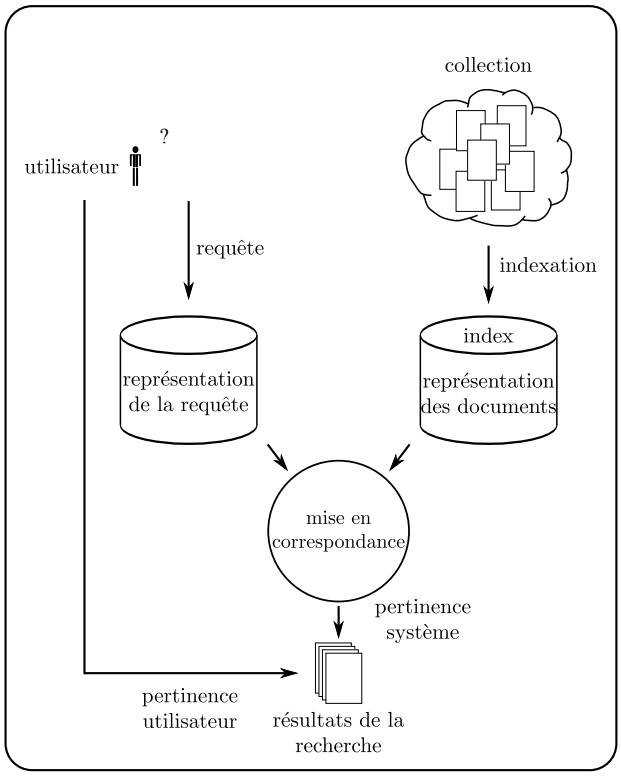


FIG. 1.2 – Représentation d'un système de recherche d'information.

1.1.2.3 Recherche

La recherche correspond à la phase du système qui produit une liste ordonnée de documents susceptibles de répondre à une requête posée par un utilisateur. À partir d'une requête q_k, un score est attribué à chaque document d_i de la collection, noté $score(d_i, q_k)$. Il évalue la pertinence entre la représentation du document d_i et celle de la requête q_k grâce à une fonction de mise en correspondance. Il est ensuite utilisé pour trier l'ensemble des documents de $\mathcal{D}$ par pertinence. Cela correspond alors à la pertinence système.

Dans le but d'obtenir de meilleurs résultats, le système peut modifier la requête initiale fournie par l'utilisateur. Deux approches principales peuvent être utilisées, soit globalement en modifiant la requête de l'utilisateur avant d'effectuer une nouvelle recherche, soit localement en demandant à l'utilisateur d'évaluer les résultats retournés par le système à partir de la requête initiale.

L'approche globale inclut l'extension de la requête fournie par l'utilisateur en utilisant un thésaurus ou en la corrigeant. Le recours à un thésaurus est très pratiqué quand les applications sont limitées à un domaine très spécifique comme le thésaurus médical MeSH (*Medical Subject Headings*) [1]. Dans un contexte général, le thésaurus le plus utilisé en langue anglaise est Wordnet[2], un thésaurus construit manuellement [Miller *et al.*, 1990]. D'autres approches construisent automatiquement le thésaurus à partir des documents de la collection [Schütze, 1998]. La requête initiale peut également contenir des erreurs qu'il est possible de corriger en cherchant les mots les plus proches

[1] `http://www.nlm.nih.gov/mesh/meshrels.html`
[2] `http://wordnet.princeton.edu`

en terme de distance d'édition ou de voisinage du mot [Kukich, 1992]. La distance d'édition entre deux mots correspond au nombre minimal d'insertion, de suppression ou de substitution qu'il faut effectuer pour passer d'un mot à l'autre tandis que le voisinage des mots est généralement déterminé grâce aux n-grammes. Ces techniques sont largement utilisées par les moteurs de recherche. Par exemple, en cherchant *flur bleue*, le moteur de recherche Google corrige la requête et retourne les résultats pour la requête *fleur bleue*. Il propose également d'étendre cette requête en cherchant *être fleur bleue*, *fleur bleue des alpes* ou *fleur bleue paroles*, *fleur bleue* étant une chanson de Charles Trenet.

Contrairement à l'approche globale, l'approche locale traite dans un premier temps la requête initiale proposée par l'utilisateur et lui demande ensuite d'évaluer la pertinence d'un certain nombre de documents retournés. Cela correspond alors à la pertinence utilisateur. Le jugement de pertinence qu'il porte sur ces premiers documents est ensuite utilisé pour modifier la requête en cherchant à distinguer les mots qui sont présents dans les documents pertinents. L'algorithme Rocchio est le plus connu pour exploiter ce retour de pertinence utilisateur et modifier la requête en conséquence [Salton et Buckley, 1990, Joachims, 1997, Moschitti, 2003].

1.1.2.4 Évaluation

L'évaluation d'un système de recherche d'information s'effectue généralement sur une collection test à l'aide d'un ensemble de requêtes $\mathcal{Q}$ pour lesquelles les documents pertinents sont connus pour chaque requête. Pour une requête q_k, le sous-ensemble $\mathcal{D}_k = \{d_{k,1}, \ldots, d_{k,i}, \ldots, d_{k,|\mathcal{D}_k|}\}$ de $\mathcal{D}$ correspond à l'ensemble des documents qui sont pertinents pour cette requête. Le résultat retourné par un système de recherche d'information pour la requête q_k est une liste L_k ordonnée de documents considérés pertinents et triés grâce au score obtenu par la fonction de mise en correspondance. Pour la requête q_k, $|L_k|$ noté également N_k, correspond au nombre de documents de L_k. Le rang r correspond au r^e document retrouvé par le système parmi les N_k documents ; r est donc compris entre 1 et N_k. Il existe plusieurs critères qui permettent d'évaluer les systèmes de recherche d'information [Kamps *et al.*, 2008, Manning *et al.*, 2008], les principaux cités précédemment étant la précision et le rappel, et des extensions de ces derniers comme la précision moyenne et la précision interpolée.

Précision et rappel

La précision $P_k(N)$ correspond à la proportion de documents pertinents retrouvés pour la requête q_k parmi les N premiers documents de L_k. Le rappel $R_k(N)$ correspond au rapport entre les documents pertinents pour la requête q_k figurant dans les N premiers documents et le nombre de documents pertinents à retrouver pour cette requête. $P_k(N)$ et $R_k(N)$ s'obtiennent par :

$$P_k(N) = \frac{\sum_{r=1}^{N} \mathrm{rel}_k(r)}{N} \qquad R_k(N) = \frac{\sum_{r=1}^{N} \mathrm{rel}_k(r)}{|\mathcal{D}_k|} \tag{1.1}$$

où $\mathrm{rel}_k(r)$ est une fonction binaire de pertinence qui vaut 1 si le document placé au rang r dans la liste L_k est pertinent pour la requête q_k, c'est-à-dire s'il appartient à $\mathcal{D}_k$, et 0 sinon. Les mesures de précision et de rappel calculées sur l'ensemble des résultats retournés ne sont pas indépendantes. En général, augmenter l'une des deux mesures fait diminuer l'autre. Par conséquent, pour se rendre compte de l'efficacité

d'un système, il convient fréquemment de calculer la courbe de précision-rappel. Cette courbe s'obtient en parcourant la liste des documents retournés par le système du plus pertinent au moins pertinent et en calculant pour chaque rang la précision et le rappel correspondant aux coordonnées du point à placer sur la courbe. L'allure générale de cette courbe est présentée par la figure 1.3.

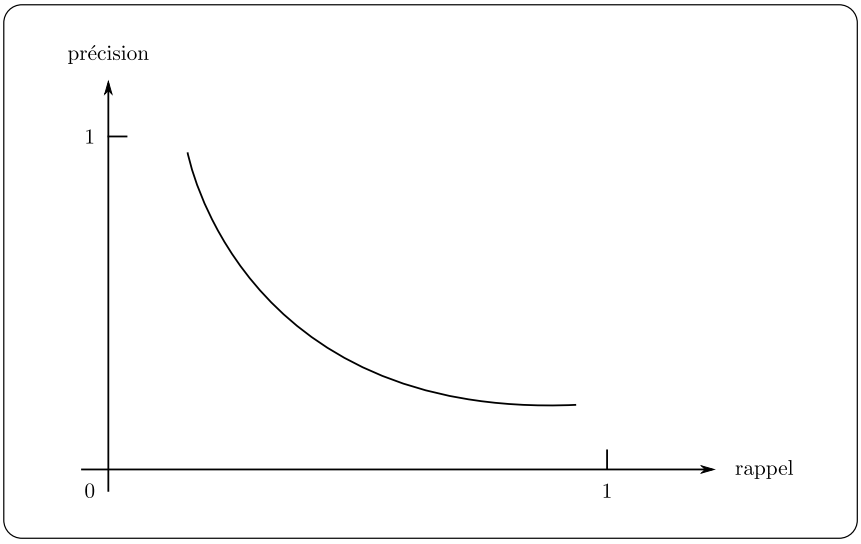


FIG. 1.3 – Allure générale d'une courbe précision-rappel.

Précision moyenne

Une première extension de ces critères correspondant à la précision moyenne AP_k est également utilisée pour évaluer les performances d'un système de recherche d'information. Elle se calcule par :

$$AP_k = \frac{\sum_{r=1}^{N_k}(P_k(r).\mathrm{rel}_k(r))}{|\mathcal{D}_k|} \tag{1.2}$$

Les critères de précision et de rappel sont calculés sur un ensemble de documents retournés par un système. Lorsque ce système retourne une liste triée de documents, le critère de précision moyenne peut être utilisé car c'est un critère global qui prend en compte l'ordre dans lequel les documents ont été retrouvés.

Précision interpolée

La précision $iP_k[x]$ correspond à la précision à un point de rappel donné x et se calcule par :

$$iP_k[x] = \begin{cases} \max_{1 \leq r \leq N_k}(P_k(r)|R_k(r) \geq x) & \text{si } x \leq R_k(N_k) \\ 0 & \text{sinon} \end{cases} \tag{1.3}$$

iP_k est généralement donné pour un rappel de $0,1$ et représente donc la précision lorsque 10% des documents pertinents à retrouver ont été retrouvés.

Évaluation pour un jeu de requêtes

Pour évaluer globalement un système de recherche d'information les moyennes des critères de précision moyenne (MAP), de précision pour les N premiers documents ($P@N$) et de précision interpolée à un point de rappel donné x sont calculées sur l'ensemble des requêtes $\mathcal{Q}$ par :

$$MAP = \frac{\sum_{k=1}^{|\mathcal{Q}|} AP_k}{|\mathcal{Q}|} \quad P@N = \frac{\sum_{k=1}^{|\mathcal{Q}|} P_k(N)}{|\mathcal{Q}|} \quad iP[x] = \frac{\sum_{k=1}^{|\mathcal{Q}|} iP_k[x]}{|\mathcal{Q}|} \tag{1.4}$$

1.1.3 Catégorisation de documents

Le problème de la recherche d'un document dans une collection peut être simplifié en utilisant des méthodes de catégorisation de documents. La catégorisation de documents fait appel à des méthodes issues du domaine de l'apprentissage automatique. Dans la pratique, un algorithme d'apprentissage exploite des observations extraites d'une population particulière, appelé échantillon d'apprentissage, pour produire un modèle. Dans le cadre de l'apprentissage non supervisé, ou classification, l'objectif est de constituer un modèle qui regroupe les observations semblables entre elles. La catégorisation de documents considérée dans la suite s'inscrit dans le cadre de l'apprentissage supervisé ou classement. L'objectif est alors de produire un modèle également appelé classifieur, noté ϕ, qui pour une nouvelle observation, correspondant ici à un nouveau document, prédit une étiquette qui doit correspondre à la catégorie associée au document, appelé label. Le principe général de la catégorisation est illustré par la figure 1.4. L'ensemble des catégories ou classes qui peuvent être associées à un document est représenté par $\mathcal{C} = \{c_1, \ldots, c_k, \ldots, c_{|\mathcal{C}|}\}$. Les différentes catégorisations qui existent, à savoir les catégorisations binaire, multiclasse et multilabel, sont présentées avant d'introduire les classifieurs classiques et les critères qui permettent d'évaluer les résultats d'un classement.

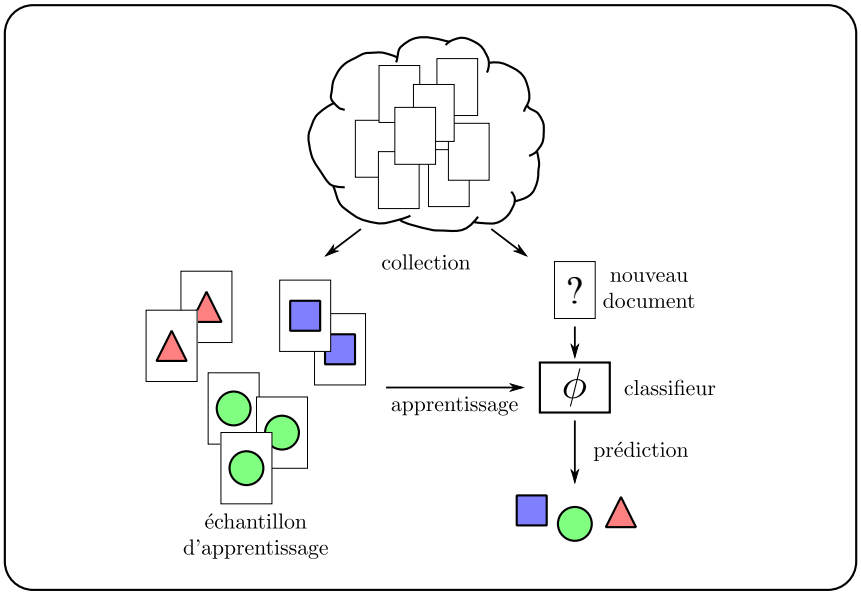


FIG. 1.4 – Représentation générale de la catégorisation de documents.

1.1.3.1 Catégorisations binaire, multiclasse et multilabel

La catégorisation de documents fait partie des premiers principes d'organisation utilisés à l'origine dans les bibliothèques pour retrouver plus facilement les ouvrages. Dès 1627, le septième point énoncé par Gabriel Naudé souligne l'importance *de l'ordre et de la disposition que doivent garder les livres dans une Bibliothèque : car il n'y a point de doute que sans icelle toute nostre recherche seroit vaine et nostre labeur sans fruict, puis que les livres ne sont mis et réservez en cet endroit que pour en tirer service aux occasions qui se présentent. Ce que toutesfois il est impossible de faire s'ils ne sont rangez et disposez suivant leurs diverses matières, ou en telle autre façon qu'on les puisse trouver facilement et à point nommé.* [Naudé, 1627]. Il existe différentes classifications bibliographiques, comme la classification décimale de Dewey[1] ou la classification décimale universelle[2] qui répartissent les ouvrages en dix classes, chaque classe étant découpée en dix divisions elles-mêmes découpées en dix sous-divisions. Un ouvrage est ensuite classé en lui assignant une catégorie composée d'au moins 3 chiffres correspondant à sa classe, sa division et sous-division. Pour un livre de cuisine, la classe associée est 6 (Techniques), la division est 4 (Vie domestique) et la sous-division est 1 (Alimentation). D'autres chiffres séparés par un point sont ensuite ajoutés pour préciser la catégorie du livre. Pour le livre de cuisine, la catégorie finale associée est 641.5.

Pour retrouver facilement un livre dans une bibliothèque, les classes sont définies de telle sorte qu'un livre ne puisse pas appartenir à plus d'une classe. Ceci correspond à la catégorisation multiclasse. Dans d'autres contextes, il peut arriver qu'un document possèdent plusieurs labels ce qui correspond alors, à la catégorisation multilabel.

Dans la suite, pour un document d_i, $L(d_i)$ correspondra à l'ensemble des labels qui lui sont associés et $\hat{L}(d_i)$ l'ensemble des étiquettes qui ont été affectées à d_i par un classifieur. Les labels et les étiquettes correspondent à des catégories ou classes de $\mathcal{C}$, mais les labels représentent les classes qui sont effectivement associées à un document, alors que les étiquettes correspondent aux classes qui ont été affectées à un document par un classifieur. Le but de la catégorisation est alors d'obtenir pour chaque document d_i que $L(d_i)$ soit égal à $\hat{L}(d_i)$. Quelle que soit la catégorisation considérée, le nombre de classes possibles est toujours supérieur ou égal à deux ($|\mathcal{C}| \geq 2$). En effet, quand une seule classe c_1 est considérée, l'ensemble des catégories $\mathcal{C}$ possède deux classes qui sont égales à la classe c_1 et à son complémentaire $\bar{c_1}$. Lorsque l'ensemble des catégories $\mathcal{C}$ est composé de strictement deux classes ($|\mathcal{C}| = 2$) et que les documents ne possèdent qu'un seul label ($L(d_i) = 1$)), il s'agit d'une catégorisation binaire. En revanche, si le nombre de classes possibles est strictement supérieur à deux ($|\mathcal{C}| > 2$) et le nombre de labels associés aux documents reste égal à un ($L(d_i) = 1$)), il s'agit d'une catégorisation multiclasse. Enfin quel que soit le nombre de classes possibles ($|\mathcal{C}| \geq 2$), la catégorisation multilabel considère des documents pour lesquels plusieurs labels peuvent être associés aux documents ($L(d_i) \geq 1$)). Un résumé de ces différentes catégorisations possibles est donné par la table 1.1.

1.1.3.2 Algorithmes de catégorisation

Dans le contexte de la catégorisation supervisée, un échantillon de base correspond à un ensemble de documents pour lesquels la représentation et la ou les classes associées à ces documents sont connues. Dans la pratique, cet échantillon de base est partitionné

[1] `http://www.oclc.org/dewey`
[2] `http://www.udcc.org/guide.htm`

TAB. 1.1 – Catégorisations binaire, multiclasse et multilabel.

Types de catégorisation	$\|L(d_i)\| = 1$	$\|L(d_i)\| \geq 1$
$\|\mathcal{C}\| = 2$	binaire	multilabel
$\|\mathcal{C}\| > 2$	multiclasse	

en deux afin de constituer un échantillon d'apprentissage $\mathcal{D}_A$ et un échantillon de test $\mathcal{D}_T$. Le but est alors de construire un classifieur à l'aide de l'échantillon d'apprentissage $\mathcal{D}_A$. Ce classifieur permettra ensuite de prédire les classes à affecter à n'importe quel autre document. L'évaluation du classifieur ainsi construit se fait à l'aide de l'échantillon de test $\mathcal{D}_T$. Dans certains cas, l'évaluation peut aussi être faite en pratiquant une validation croisée (ou *cross-validation*) ; c'est-à-dire en partitionnant l'échantillon de base en plusieurs échantillons puis en itérant le processus d'apprentissage sur l'ensemble des éléments à l'exception de ceux figurant dans un des échantillons qui servira à l'évaluation. Cette validation est notamment utilisée lorsque la taille de la collection est limitée et permet alors de mieux estimer les taux de précision et de rappel [Efron, 1983]. Pour illustrer cette tâche de catégorisation, les k plus proches voisins, le classifieur naïf bayésien et les machines à vecteurs de support sont présentés dans un contexte de catégorisation binaire ou multiclasse.

k plus proches voisins

Le principe des k plus proches voisins (ou *kppv*) consiste à classer pour un nouveau document d_i, la liste des k documents de l'échantillon d'apprentissage les plus proches selon une distance choisie [Hinneburg *et al.*, 2000]. Le document d_i est alors associé à la catégorie majoritairement représentée parmi ces k plus proches voisins.

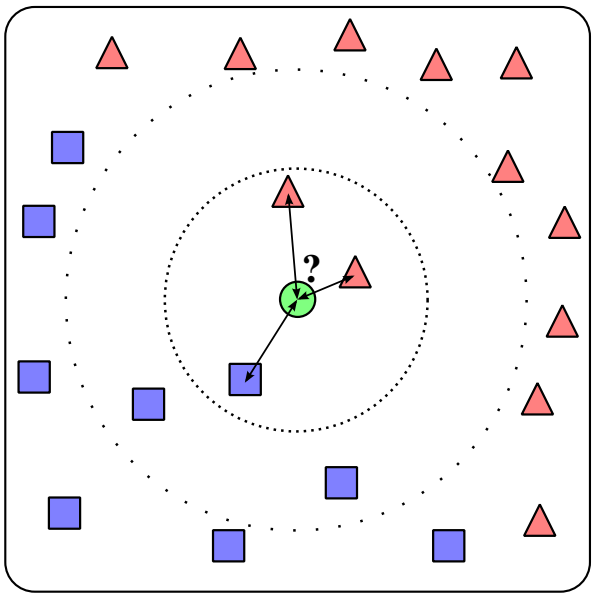


FIG. 1.5 – Représentation de l'algorithme de k plus proche voisins.

La figure 1.5 illustre graphiquement cet algorithme. Le but est de trouver la catégorie

à affecter au document *rond vert* entre les catégories *carré bleu* et *triangle rouge*. Sur cette figure, en utilisant la distance euclidienne et les trois plus proches voisins, le document *rond vert* est affecté à la catégorie *triangle rouge*. En revanche, si les cinq documents les plus proches sont pris en compte, la catégorie à associer au document *rond vert* est *carré bleu*.

Dans sa version la plus simple, cet algorithme offre l'avantage d'être simple à appréhender et à implémenter. Il nécessite cependant le choix du nombre k de voisins à considérer et de la métrique, généralement la distance euclidienne, à utiliser pour calculer la distance entre les différents éléments. Que ce soit en catégorisation de documents textuels ou d'images, cet algorithme est robuste et donne de bons résultats [Han *et al.*, 2001, Szummer et Picard, 1998]. La complexité de l'algorithme des k plus proches voisins est en revanche très élevée car pour chaque nouveau document, les distances avec tous les éléments de l'échantillon d'apprentissage doivent être calculées. Des approches pour réduire cette complexité ont été proposées en utilisant par exemple des structures arborescentes pour représenter les éléments de l'échantillon d'apprentissage [Bentley, 1975].

Approche naïve bayésienne

L'approche naïve bayésienne permet d'effectuer un classement probabiliste basé sur le théorème de Bayes. Pour classer un document d_i, il faut calculer pour chaque classe c_k, la probabilité $P(c_k|d_i)$ d'appartenir à la classe c_k sachant la représentation du document d_i. Cette probabilité peut s'obtenir par :

$$P(c_k|d_i) = \frac{P(d_i|c_k).P(c_k)}{P(d_i)} \tag{1.5}$$

où $P(c_k)$ est la probabilité qu'un document quelconque appartienne à la classe c_k, $P(d_i)$ est la probabilité associée au document d_i et $P(d_i|c_k)$ est la probabilité d'avoir le document d_i sachant que la classe c_k est considérée.

L'étiquette associée à d_i correspond alors à la catégorie c_k pour laquelle la probabilité $P(c_k|d_i)$ est la plus élevée. Étant donné que $P(d_i)$ est constante quelle que soit la classe c_k, $P(c_k|d_i)$ peut être estimée par le produit $P(d_i|c_k).P(c_k)$. En effet, l'ordre des probabilités pour les différentes catégories n'est pas modifié par la suppression de la constante $P(d_i)$.

$P(d_i|c_k).P(c_k)$ est ensuite calculé en faisant l'hypothèse que les mots qui composent le document d_i apparaissent de façon indépendante dans le document :

$$P(c_k|d_i) = P(c_k) \prod_{t_j \in d_i} P(t_j|c_k) \tag{1.6}$$

où $P(t_j|c_k)$ est la probabilité conditionnelle d'apparition du terme t_j dans la classe c_k.

C'est l'hypothèse de l'apparition indépendante des termes qui explique le caractère naïf de la méthode. Cette hypothèse n'est évidemment pas vérifiée pour des documents textuels, car par exemple, après le mot *apprentissage*, la probabilité d'apparition du mot *automatique* n'est pas la même que celle du mot *fleur*. Bien que dans la plupart des cas, l'apparition indépendante des termes ne soit pas vérifiée, l'approche naïve bayésienne permet d'obtenir de très bons résultats que ce soit pour des documents textuels [Friedman *et al.*, 1997] ou des images [Vailaya *et al.*, 2001]. Il convient de remarquer que des adaptations pour prendre en compte la dépendance des termes ont

été proposées sans pour autant améliorer significativement les résultats [Domingos et Pazzani, 1997].

Machines à vecteurs de support

Les machines à vecteurs de support ou séparateurs à vaste marge (SVM) regroupent un ensemble de méthodes originellement définies pour résoudre des problèmes de classement à deux classes [Boser *et al.*, 1992]. Le principe des machines à vecteurs de support, illustré par la figure 1.6, consiste à séparer les éléments de chacune des classes par un hyperplan de façon à maximiser la distance minimale qui existe entre les éléments et l'hyperplan. Cette distance, appelée marge, permet de choisir l'hyperplan qui optimise la séparation des deux classes. Comme le montre la figure 1.6, il existe une infinité d'hyperplans en pointillés verts qui permettent de séparer les classes. En revanche, il n'existe qu'un seul hyperplan en tirets rouges qui maximise la marge. Les éléments les plus proches de l'hyperplan sont appelés vecteurs supports et sont entourés en bleu. Cette méthode est efficace si les classes sont linéairement séparables. Lorsque ce n'est pas le cas, il est possible d'utiliser des marges douces qui autorisent certaines erreurs en ajoutant des pénalités en fonction de la distance à la marge [Cortes et Vapnik, 1995].

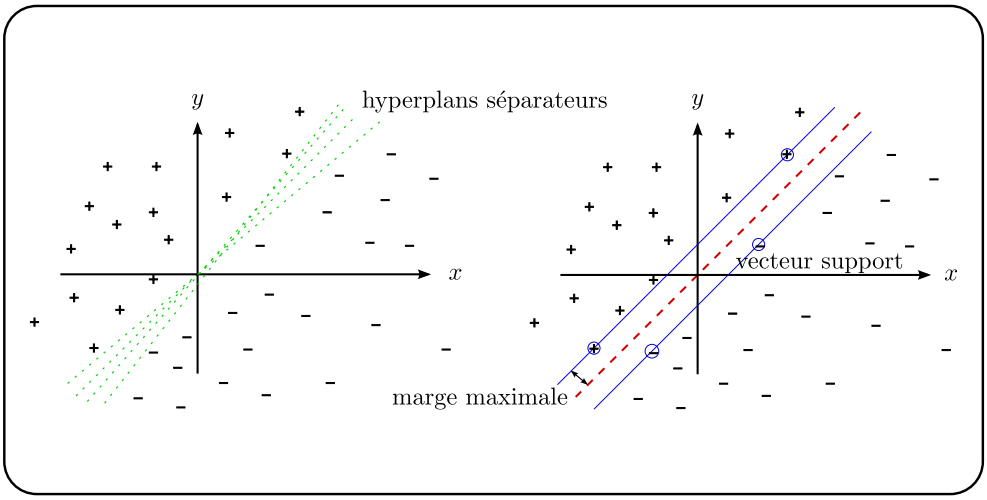


FIG. 1.6 – Exemple d'un problème de classement à deux classes linéairement séparable.

Dans la pratique, l'hypothèse de séparabilité linéaire n'étant en général pas vérifiée, une projection des éléments dans un espace de dimension plus grande permet de les séparer linéairement plus facilement. Cette transformation se fait à l'aide de fonctions, appelées fonctions noyaux, qui sous certaines conditions, permettent le changement d'espace sans connaître explicitement la transformation à appliquer. Ces méthodes de plus en plus utilisées ont des fondements théoriques solides et donnent de très bons résultats sur les documents textuels [Abe, 2010, Burges, 1998, Joachims, 1998].

1.1.3.3 Évaluation

Le résultat d'un classement s'évalue en calculant le taux de documents bien classés ou l'exactitude sur l'échantillon de test composé de documents pour lesquels les labels sont connus [Kazawa *et al.*, 2005]. Ceci revient à compter le nombre de documents qui ont été correctement classés par rapport au nombre de documents à classer. En

catégorisation binaire et multiclasse, un document est correctement classé si l'étiquette prédite par le classifieur correspond au label du document. Dans le contexte multilabel, un document est bien classé si et seulement si toutes les étiquettes ont été correctement affectées par le processus de classement. Le critère d'exactitude (C_{exact}) est alors défini par :

$$C_{exact} = \frac{|\{d_i \in \mathcal{D}_T | L(d_i) = \hat{L}(d_i)\}|}{|\mathcal{D}_T|} \tag{1.7}$$

Pour évaluer les catégorisations binaire et multiclasse, le taux de bien classés est généralement le seul critère utilisé. Dans le contexte multilabel, il apparaît clairement que ce critère est très contraignant et ne permet pas de prendre en compte les correspondances partielles où seulement une partie des étiquettes est correcte. Deux autres critères basés sur la F-mesure [Yang et Liu, 1999], une mesure qui correspond à la moyenne harmonique de la précision (P) et du rappel (R), sont alors considérés pour évaluer les résultats d'un classement. Ces critères correspondent à la moyenne micro et macro de la F-mesure.

$$F - mesure = \frac{2PR}{P+R} \tag{1.8}$$

Ils peuvent être définis à partir d'une table de contingence telle que la table 1.2 dans laquelle $\bar{c_k}$ correspond aux catégories de $\mathcal{C}$ qui ne sont pas c_k ($c_k = \mathcal{C} \setminus \{c_k\}$). Cette table est construite à partir de l'échantillon de test $\mathcal{D}_T$ où tp_k (les vrais positifs, de l'anglais *true positive*) correspond au nombre de documents qui appartiennent à la catégorie c_k et qui ont été correctement classés, tn_k (les vrais négatifs, de l'anglais *true negative*) représente le nombre de documents qui n'appartiennent pas à la catégorie c_k et qui ont été correctement classés comme n'appartenant pas à cette catégorie, fp_k (les faux positifs, de l'anglais *false positive*) est le nombre de documents n'appartenant pas à la catégorie c_k et qui ont été incorrectement classés comme appartenant à la catégorie c_k, fn_k (les faux négatifs, de l'anglais *false negative*) correspond au nombre de documents qui appartiennent à la catégorie c_k mais qui n'ont pas été correctement classés.

TAB. 1.2 – Table de contingence définie pour la catégorie c_k.

		Classes à prédire	
		c_k	$\bar{c_k}$
Classes prédites	c_k	tp_k	fp_k
	$\bar{c_k}$	fn_k	tn_k

- $tp_k = |\{d_i \in \mathcal{D}_T | c_k \in L(d_i),\ c_k \in \hat{L}(d_i)\}|$
- $fp_k = |\{d_i \in \mathcal{D}_T | c_k \notin L(d_i),\ c_k \in \hat{L}(d_i)\}|$
- $fn_k = |\{d_i \in \mathcal{D}_T | c_k \in L(d_i),\ c_k \notin \hat{L}(d_i)\}|$
- $tn_k = |\{d_i \in \mathcal{D}_T | c_k \notin L(d_i),\ c_k \notin \hat{L}(d_i)\}|$

Les moyennes micro de la précision, du rappel et de la F-mesure sont ensuite obtenues en considérant toutes les catégories. La précision P et le rappel R sont alors définis par :

$$P = \frac{\sum_{k=1}^{|\mathcal{C}|} tp_k}{\sum_{k=1}^{|\mathcal{C}|}(tp_k + fp_k)} \qquad R = \frac{\sum_{k=1}^{|\mathcal{C}|} tp_k}{\sum_{k=1}^{|\mathcal{C}|}(tp_k + fn_k)} \tag{1.9}$$

La moyenne micro de la F-mesure est ensuite calculée par :

$$C_{micro} = \frac{2PR}{P+R} \tag{1.10}$$

Les moyennes macro de la précision, du rappel et de la F-mesure sont calculées en faisant la moyenne des scores calculés pour chaque catégorie. Les précision (P_k) et rappel (R_k) sont définis pour la catégorie c_k par :

$$P_k = \frac{tp_k}{tp_k + fp_k} \qquad R_k = \frac{tp_k}{tp_k + fn_k} \tag{1.11}$$

La moyenne macro de la F-mesure est obtenue par :

$$C_{macro} = \frac{1}{|\mathcal{C}|} \sum_{k=1}^{|\mathcal{C}|} \frac{2P_k R_k}{P_k + R_k} \tag{1.12}$$

Contrairement à la moyenne micro de la F-mesure, la moyenne macro favorise les catégories qui sont rares, c'est-à-dire les catégories pour lesquelles il y a peu de documents qui appartiennent à cette catégorie. Les trois critères C_{exact}, C_{micro} et C_{macro} sont utilisés pour évaluer les performances d'un classement multilabel alors que seul le critère C_{exact} qui correspond au taux de bien classés est généralement utilisé pour un classement binaire ou multiclasse.

1.1.4 Importance de la représentation

Les tâches considérées dans cette thèse ainsi que les méthodes d'évaluation associées ont été présentées sans tenir compte de la représentation des documents. Cependant, pour se convaincre de l'importance de la représentation de l'information, il suffit de penser au problème qui consiste à rechercher dans un dictionnaire la définition d'un mot dont l'orthographe est connue. Si ce problème est facile à résoudre, le problème inverse qui consiste à rechercher les mots du dictionnaire correspondant à une définition donnée, est beaucoup plus difficile. Lors de la recherche d'un mot dans un dictionnaire, il est évident que le fait que les mots soient indexés dans l'ordre alphabétique rend la recherche rapide et facile. En revanche, pour la recherche inverse, l'ordre alphabétique des mots n'est plus d'aucune utilité. Ce simple exemple montre que la représentation de l'information est essentielle pour pouvoir accéder facilement et efficacement à l'information utile. Dans la suite, les documents multimédias considérés sont composés d'une partie texte, d'une image ou des deux. Les différentes représentations pour chacune de ces modalités, ainsi que les méthodes qui permettent de les fusionner seront présentées.

1.2 Représentation des données textuelles

Les documents textuels sont l'un des supports les plus utilisés pour communiquer ou transmettre de l'information. Du livre au document XML en passant par le télégramme et les courriers électroniques, les données textuelles font depuis longtemps partie de notre quotidien. Elles peuvent se présenter sous différentes formes et il est possible de

les représenter en considérant toute l'information sur la structure du texte ou en ne considérant que le contenu textuel. Plusieurs modèles ont été employés en recherche d'information et l'utilisation d'un index pour représenter les documents est la méthode la plus courante.

1.2.1 Qu'est-ce qu'un document texte ?

Du simple texte, dit plat, lorsqu'il n'a ni mise en page ni formatage particulier, au texte structuré avec l'utilisation de balises comme dans le format XML, la représentation d'un document textuel est très importante afin de conserver le plus d'information utile pour retrouver les documents tout en étant le plus concis possible.

1.2.1.1 Du document structuré au texte plat

Quelle que soit le document textuel considéré, il peut être représenté selon différents points de vue [Fourel, 1998], comme illustré par la figure 1.7 [Fuhr, 2003]. La vue de contenu, ou sémantique, caractérise l'information textuelle. Elle correspond à l'information la plus souvent recherchée par un utilisateur. C'est ce qui représente le fond du document. La vue logique concerne l'organisation du document généralement sous la forme d'une arborescence et peut s'utiliser pour la mise en forme, l'écriture en gras ou en italique, mais également pour les métadonnées associées au document. Pour un ouvrage, cela peut correspondre au nom de l'auteur, à la date de parution, au titre et aux découpages en chapitres, paragraphes. Les informations associées à la vue logique permettent d'effectuer des recherches beaucoup plus précises, mais ne sont pas systématiquement existantes. La vue de présentation concerne les informations liées à la mise en page de l'information textuelle et au découpage du texte avec la position des entêtes, l'alignement des paragraphes, etc. Ces informations peuvent, par exemple, aider à identifier la structure logique.

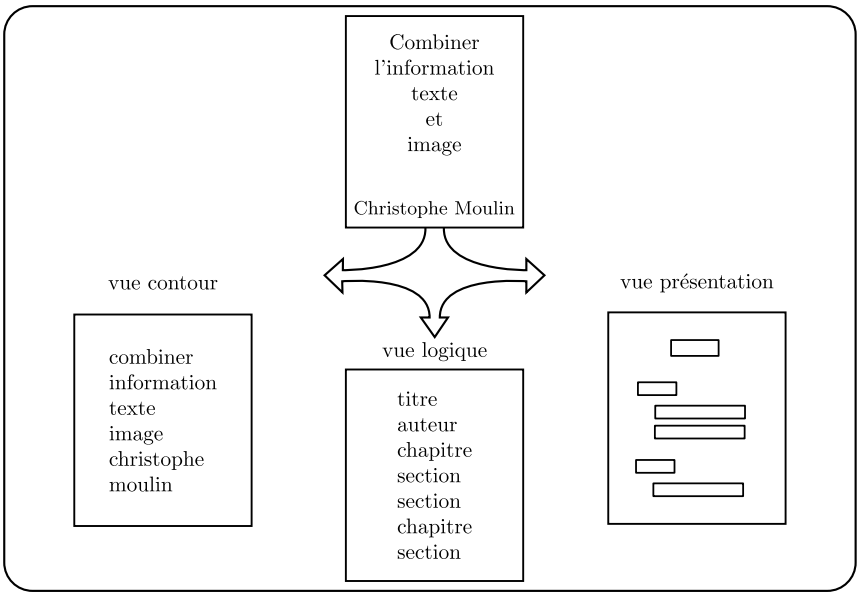


FIG. 1.7 – Différentes structures associées à un document texte.

Le besoin d'un utilisateur peut s'exprimer sous la forme d'une requête. Lorsque

ce dernier recherche une information particulière, il formule sa requête et attend une réponse rapide du système de recherche d'information. Des choix concernant les différentes informations des documents à conserver sont donc à faire afin de pouvoir répondre aux différents besoins des utilisateurs dans des temps raisonnables. De plus en plus, la structure des documents est utilisée, notamment avec le développement des formats comme XML, pour améliorer les résultats d'une recherche en ne retournant à l'utilisateur qu'une sous partie du document original [Géry *et al.*, 2009, Lalmas, 2009] ou en exploitant les liens entre les documents [Verbyst et Mulhem, 2009].

Les documents textuels considérés dans la suite sont les documents qui contiennent uniquement du texte plat. Quand le texte est formaté ou structuré, les informations associées comme la mise en forme ou les métadonnées sont simplement supprimées pour ne considérer que du texte plat.

1.2.1.2 Spécificité des documents texte

Si un document textuel particulier est généralement monolingue, il se peut que pour deux documents issus d'une même collection, les langues utilisées soient différentes. Ce problème de collection multilingue peut se résoudre en utilisant une approche par dictionnaire qui consiste à traduire tous les documents dans une même langue avant l'indexation. La requête fournie par l'utilisateur devra également être traduite avant d'effectuer la recherche [Hull et Grefenstette, 1996]. Les spécificités des langues sont également à prendre en compte. Les idéogrammes de la langue chinoise ne peuvent pas être manipulés comme les mots issus de langues indo-européennes. Dans la suite, la langue anglaise sera principalement utilisée. Pour représenter l'information textuelle, plusieurs problèmes liés à la langue sont à considérer. Le problème de la polysémie correspond au fait que plusieurs définitions peuvent correspondre pour un même mot ; c'est le cas par exemple du mot *blanc* qui peut désigner, une couleur, un vin, une espace typographique, etc. Ce problème est proche de celui de l'homonymie pour lequel deux mots peuvent s'écrire (homographe) ou se prononcer (homophone) de la même façon, mais avoir des sens différents ; le mot *est* peut désigner à la fois le verbe *être* conjugué à la troisième personne du singulier de l'indicatif présent et la direction opposée à l'Ouest. Ces deux problèmes sont proches mais néanmoins distincts. Contrairement au problème de polysémie, deux mots homonymes possèdent deux entrées dans un dictionnaire. Enfin le problème de la synonymie concerne deux mots distincts qui ont le même sens ; les mots *souvent* et *fréquemment* sont considérés comme synonymes.

1.2.1.3 Représentation des documents texte

Les différentes informations qu'il est possible d'extraire d'un texte plat dépendent de l'analyse envisagée comme le montre la figure 1.8.

La plus simple des analyses est l'analyse lexicale qui consiste à découper le texte en une suite de mots. Le découpage se fait généralement grâce aux caractères de ponctuation et aux espaces. Les mots qui ressortent de ce découpage peuvent alors être utilisés séparément (sac de mots) [Lewis, 1998], sous forme de séquence (n-grammes) [Cavnar et Trenkle, 1994] ou regroupés en concepts [Voss *et al.*, 1999].

L'analyse grammaticale associe à chaque mot une étiquette correspondant à sa partie du discours (nom, adjectif, verbe, etc.). En utilisant cette analyse plus fine que l'analyse lexicale, il est par exemple plus facile de distinguer les homographes [Brill, 1992, Brants, 2000].

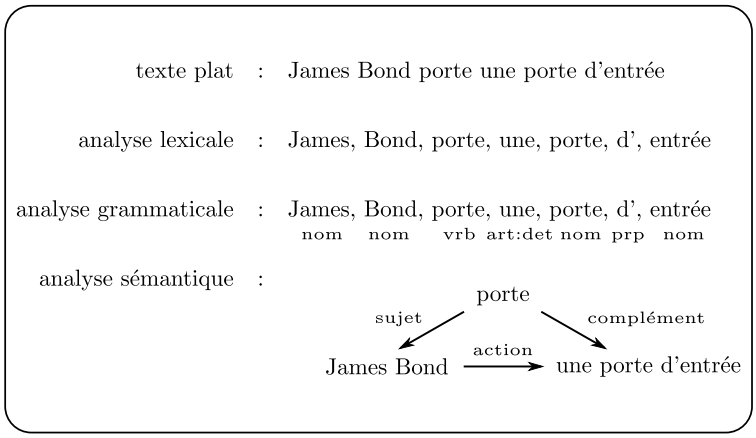


FIG. 1.8 – Les différentes analyses possibles d'un texte plat.

Enfin, l'analyse sémantique s'attache à comprendre le sens des phrases pour comprendre le texte. Des approches existent pour traiter des documents structurés ou semi-structurés, mais il n'existe aucune méthode qui permette de faire cette analyse efficacement en partant de texte plat [Shah *et al.*, 2002].

Dans la suite, l'accent sera mis sur la représentation en sac de mots. Cette représentation ne permet pas de reconstituer le texte original du document puisque l'ordre des mots dans le document est perdu. Bien que cette représentation semble très simpliste, elle a largement fait ses preuves ce qui explique qu'elle reste l'une des plus utilisée pour la représentation des documents textuels [Salton *et al.*, 1975, Lewis, 1998].

1.2.2 Modèle de représentation par sac de mots

La représentation en sac de mots repose sur un ensemble $T = \{t_1, \ldots, t_j, \ldots, t_{|T|}\}$ de mots ou termes formant le vocabulaire adapté permettant de représenter le contenu d'un document. Ce vocabulaire est généralement construit à partir des mots qui apparaissent dans les documents de la collection D. Le nombre de mots $|T|$ qui composent ce vocabulaire correspond à sa taille (ou dimension) et peut être très élevé même pour un faible nombre de documents. La représentation des documents dans de très grandes dimensions entraîne des problèmes lorsqu'il faut calculer des distances entre les documents. En effet, le rapport entre la distance maximale et la distance minimale en très grande dimension tend vers un : cet effet est connu comme le fléau ou la malédiction de la dimension [Indyk et Motwani, 1998]. Il est alors intéressant de chercher à réduire la taille du vocabulaire [Lewis, 1992b, Sebastiani, 2002]. Pour un mot t_j du vocabulaire T et un document d_i de la collection $\mathcal{D}$, $w_{i,j}$ correspond au poids du mot t_j dans le document d_i.

Il existe trois grandes familles de modèles principalement issus des études réalisées en recherche d'information qui exploitent un tel sac de mots : les modèles booléens, vectoriels et probabilistes. Le modèle booléen sera tout d'abord présenté ; il est le plus simple et s'appuie sur la théorie des ensembles. Le modèle vectoriel, basé sur une intuition géométrique, sera ensuite introduit. Enfin le modèle probabiliste qui repose sur la théorie des probabilités sera expliqué. Ces modèles seront présentés dans un contexte de recherche d'information en précisant pour chacun les poids $w_{i,j}$ classiquement employés pour la représentation des documents et des requêtes ainsi que la fonction de mise en

correspondance utilisée pour juger de la pertinence d'un document d_i par rapport à une requête q_k. Ces modèles seront illustrés à l'aide de plusieurs articles extraits de l'encyclopédie Wikipedia (Australie, James Bond, Motus (jeu télévisé), Natation, Origami, Parachute et Roller) en utilisant un vocabulaire limité à quelques mots (base, épreuve, papillon, pliage, porte et sport).

1.2.2.1 Modèles booléens

Les modèles booléens se servent du vocabulaire T pour représenter les documents sous forme d'ensembles. Avec le modèle standard, les documents sont caractérisés par la présence ou l'absence de chaque terme t_j dans leur contenu. En utilisant le formalisme de l'algèbre de Boole [Boole, 1854], un document d_i est représenté par un vecteur comportant autant de composantes qu'il y a de termes dans T. Le poids $w_{i,j}$ du terme t_j dans le document d_i vaut 1 si le terme t_j apparaît dans le document d_i, 0 sinon.

Une requête peut se construire grâce aux trois opérateurs logiques (**et** : $\wedge$, **ou** : $\vee$, **non** : $\neg$). Le langage des requêtes est très expressif et permet d'effectuer des recherches très précises.

La mise en correspondance s'effectue ensuite à l'aide des opérations d'union, d'intersection et de différence entre les ensembles de résultats associés à chaque terme de la requête. Un exemple de l'utilisation de ce modèle est illustré par la figure 1.9.

La table 1.3 représente un ensemble de termes et de documents extraits de Wikipedia où un élément (d_i, t_j) vaut 1 si le terme t_j apparaît dans le document d_i, et 0 sinon.

Pour rechercher un document, il suffit de considérer les vecteurs associés aux termes de la requête et d'effecteur le calcul en utilisant les propriétés de l'algèbre de Boole.

Pour représenter un terme d'une requête, il faut regarder dans la table 1.3 la colonne qui lui est associée. Le mot *épreuve* est ainsi représenté par *1011000*.

Pour la requête : *épreuve* **ou** *papillon* **ou** *sport*, il suffit d'effectuer l'opération booléenne **ou** entre les vecteurs représentatifs de *épreuve*, *papillon* et *sport* :

$$1011000 \vee 0101001 \vee 1101011 = 1111011$$

Les documents qui répondent à la requête sont donc : *Australie*, *James Bond*, *Motus*, *Natation*, *Parachute* et *Roller*.

Pour la requête : *épreuve* **et** *papillon* **et** *sport* :

$$1011000 \wedge 0101001 \wedge 1101011 = 0001000$$

Le document qui correspond à la requête est donc le document *Natation*.

FIG. 1.9 – Exemple de l'utilisation d'un modèle booléen.

Le principal avantage de ce modèle est qu'il est simple à comprendre par l'utilisateur. Il est très efficace dans le cadre de collections spécifiques où des spécialistes connaissent les termes exacts pour formuler les requêtes. Cette efficacité n'est malheureusement plus vérifiée pour des collections généralistes où la formulation des requêtes peut se révéler longue et fastidieuse. Un autre inconvénient est l'impossibilité de retourner

TAB. 1.3 – Matrice document-terme où un élément (d_i, t_j) vaut 1 si le terme t_j apparaît dans le document d_i, et 0 sinon.

Documents \ Termes	base	épreuve	papillon	pliage	porte	sport
Australie	1	1	0	0	1	1
James Bond	0	0	1	0	1	1
Motus	1	1	0	0	0	0
Natation	0	1	1	0	1	1
Origami	1	0	0	1	1	0
Parachute	1	0	0	1	1	1
Roller	0	0	1	0	0	1

des documents qui répondent partiellement à la requête. Enfin, l'utilisation d'un score binaire de la pertinence des documents ne permet pas de les ordonner.

Des extensions de cette approche comme le modèle booléen étendu [Salton *et al.*, 1983] et les modèles basées sur la logique floue permettent de corriger certains de ces inconvénients. Les deux principaux représentants utilisant la logique floue sont le modèle MMM (Mixed Min and Max) et le model de Paice [Fox et Sharan, 1986, Lee et Fox, 1988, Mercier et Beigbeder, 2006].

1.2.2.2 Modèles vectoriels

Le modèle vectoriel se base sur une intuition géométrique et représente les documents sous forme de vecteurs dans l'espace des termes du vocabulaire [Salton *et al.*, 1975]. Le document d_i est alors décrit par le vecteur $\vec{d_i} = (w_{i,1}, \ldots, w_{i,j}, \ldots, w_{i,|T|})$. Dans sa version la plus simple, le poids $w_{i,j}$ correspond au nombre d'occurrences du terme t_j dans le document d_i [Garcia, 2006].

Une requête q_k est également représentée sous la forme d'un vecteur de la même façon qu'un document : $\vec{q_k} = (w_{k,1}, \ldots, w_{k,j}, \ldots, w_{k,|T|})$ où le poids $w_{k,j}$ est égal à 1.

Pour calculer la pertinence d'une requête avec un document, il faut choisir une mesure de similarité. La plus utilisée est la distance du cosinus qui est définie pour une requête q_k et un document d_i par :

$$score(d_i, q_k) = \cos\alpha = \frac{\vec{d_i} \cdot \vec{q_k}}{||\vec{d_i}||\ ||\vec{q_k}||} \tag{1.13}$$

où $\vec{d_i} \cdot \vec{q_k}$ représente le produit scalaire entre $\vec{d_i}$ et $\vec{q_k}$ et où $||\vec{d_i}||$ et $||\vec{q_k}||$ représentent les normes des vecteurs $\vec{d_i}$ et $\vec{q_k}$. D'autres mesures de similarité peuvent être utilisées comme la distance du χ^2 ou la distance de Kullback-Leibler [Rajman et Lebart, 1998].

Une illustration du calcul de la distance du cosinus est donnée par la figure 1.10 en utilisant les valeurs de la table 1.4 pour les mots *sport* et *papillon* et les documents *James Bond*, *Natation* et *Roller*.

Ce modèle qui utilise une approche basée sur l'algèbre linéaire offre l'avantage d'être simple. Il n'impose pas une pondération binaire des termes et permet de retourner une liste de documents triée par pertinence. Enfin, il considère également les documents qui ne répondent que partiellement à la requête. La similarité calculée pour des documents

TAB. 1.4 – Matrice document-terme où un élément (d_i, t_j) correspond au nombre d'occurrences du terme t_j dans le document d_i.

Termes / Documents	base	épreuve	papillon	pliage	porte	sport
Australie	7	1	0	0	1	20
James Bond	0	0	1	0	5	2
Motus	1	1	0	0	0	0
Natation	0	6	6	0	1	8
Origami	10	0	0	11	1	0
Parachute	1	0	0	1	1	4
Roller	0	0	1	0	0	4

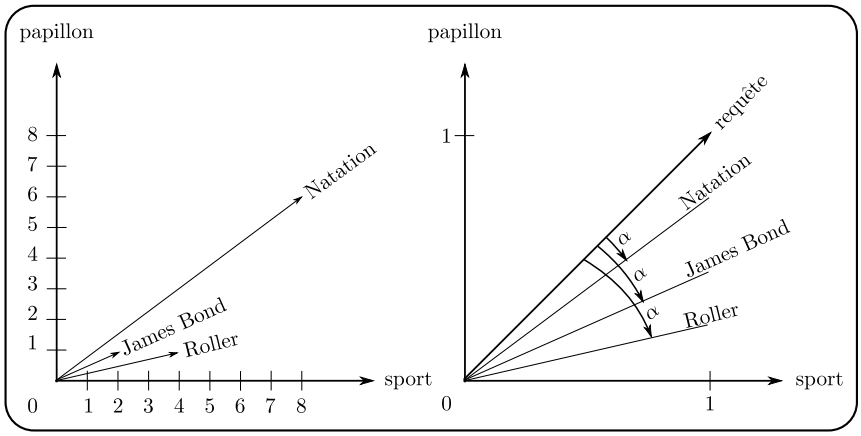


FIG. 1.10 – Représentation de la distance cosinus.

Pour rechercher les documents pertinents, il suffit de calculer la distance cosinus entre la requête et chaque document comme illustré dans la figure 1.10.

Pour la requête : *épreuve papillon sport*, les résultats retournés dans l'ordre de la distance cosinus décroissante sont :

documents	distance cosinus
Natation	0,990
James Bond	0,775
Roller	0,700
Australie	0,605
Parachute	0,577
Motus	0
Origami	0

FIG. 1.11 – Exemple de l'utilisation d'un modèle vectoriel.

de grande taille est faible car le produit scalaire est petit alors que la norme du vecteur est grande. De ce fait les grands documents sont moins bien représentés. Dans ce modèle, les termes de l'index sont supposés indépendants ce qui se traduit par une orthogonalité des vecteurs des termes du vocabulaire. Un autre modèle algébrique, connu sous le nom de modèle vectoriel généralisé, propose une représentation où les vecteurs des termes ne sont plus orthogonaux deux à deux [Wong *et al.*, 1985].

1.2.2.3 Modèles probabilistes

Dans les modèles probabilistes, la pertinence d'un document par rapport à une requête est vue comme une probabilité. Le but est de savoir si pour un document d_i et une requête q_k, la probabilité que le document d_i soit pertinent pour la requête q_k est supérieure à la probabilité que le document d_i ne soit pas pertinent pour la requête q_k. Le score qui permet de classer les documents par pertinence est calculé par :

$$score(d_i, q_k) = \frac{P(pert|\vec{d_i})}{P(\overline{pert}|\vec{d_i})} \tag{1.14}$$

où $P(pert|\vec{d_i})$ (respectivement $P(\overline{pert}|\vec{d_i})$) est la probabilité d'avoir une information pertinente (respectivement non pertinente) sachant $\vec{d_i}$.

Le plus simple des scores de pertinence entre un document d_i et une requête q_k peut être obtenu par [Spärck Jones *et al.*, 2000] :

$$score(d_i, q_k) = -\sum_{t_j \in q_k} \ln\left(\frac{|\{d_i|t_j \in d_i\}|}{|\mathcal{D}|}\right) \tag{1.15}$$

où $|\{d_i|t_j \in d_i\}|$ correspond au nombre de documents de $\mathcal{D}$ qui contient t_j.

Pour trier les documents par ordre de pertinence, il faut calculer le score de pertinence pour chaque document.

Pour la requête : *épreuve papillon sport*, les résultats retournés dans l'ordre des scores de pertinence décroissants sont :

documents	score
Natation	0.882
Australie	0.514
James Bond	0.514
Roller	0.514
Motus	0.368
Parachute	0.146
Origami	0

FIG. 1.12 – Exemple de l'utilisation d'un modèle probabiliste.

Les modèles de langage relèvent aussi d'une approche probabiliste suivant un autre point de vue qui consiste à estimer la probabilité que la requête ait été inférée par le document [Ponte et Croft, 1998, Song et Croft, 1999]. L'approche originelle considère les documents comme des sacs de mots mais des extensions prennent en compte l'ordre de ces mots et la structure des documents [Chen et Goodman, 1999, Mulhem et Chevallet,

2010]. Ces modèles sont utilisés dans d'autres contextes, comme la reconnaissance de la parole, la traduction, l'étiquetage grammatical et reposent sur d'importants fondements théoriques [Zhai, 2008].

1.2.3 Pondération tf.idf

L'analyse empirique de la fréquence des mots dans un texte a donné lieu à la loi de Zipf. Cette loi affirme que la fréquence d'un mot est inversement proportionnelle à son rang dans l'ordre des fréquences des mots en échelle logarithmique, c'est-à-dire que si le mot le plus courant apparaît 20 000 fois dans les documents de la collection, le dixième mot le plus fréquent apparaît alors 2 000 fois, le centième 200 fois le millième 20 fois etc. Pour illustrer la loi de Zipf, la fréquence des mots a été calculée sur ce chapitre et est illustrée par la figure 1.13. Le mot le plus courant, *de*, apparaît 949 fois, le dixième mot, *est*, apparaît 171 fois et le centième mot, *plusieurs*, apparaît 20 fois.

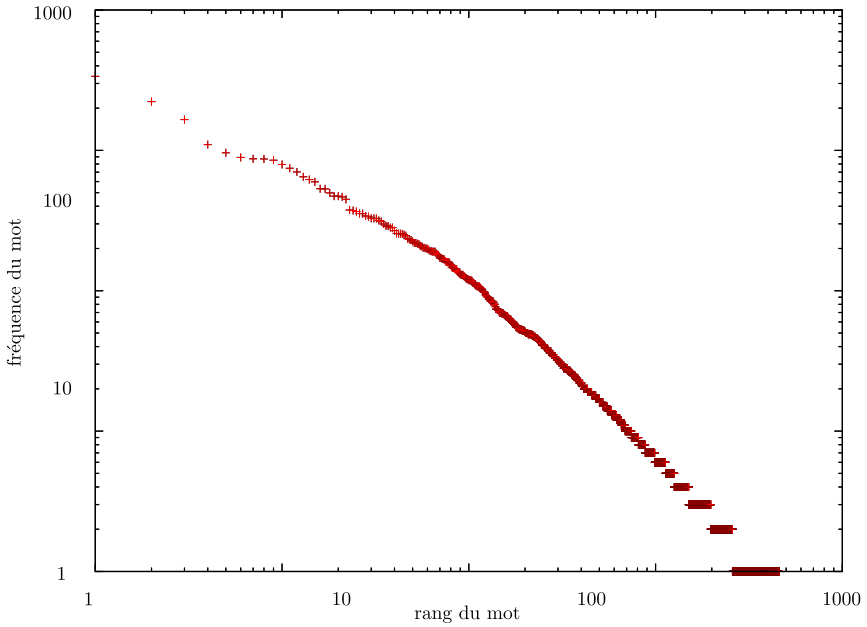


FIG. 1.13 – Représentation de la loi de Zipf avec la fréquence des mots calculée sur ce chapitre.

1.2.3.1 Principe

La pondération tf.idf, utilisée pour calculer le poids $w_{i,j}$ du terme t_j dans le document d_i, repose sur l'observation empirique illustrée par la loi de Zipf. Cette pondération considère d'une part la fréquence du terme $tf_{i,j}$ qui met en valeur la fréquence relative du terme t_j dans le document d_i et d'autre part la fréquence inverse de document idf_j mesurant l'importance du terme t_j sur l'ensemble de la collection $\mathcal{D}$. Pour mesurer l'importance d'un terme dans un document, la fréquence de ce terme est utilisée pour favoriser les termes qui sont représentatifs du document, c'est-à-dire qui apparaissent souvent dans ce dernier. Cependant, les termes qui apparaissent fréquemment dans de nombreux documents ne sont pas discriminants. La fréquence inverse de document est de ce fait utilisée pour caractériser cette discriminance.

Dans sa version la plus simple, la fréquence $tf_{i,j}$ du terme t_j dans le document d_i est définie par [Salton *et al.*, 1983] :

$$tf_{i,j} = \frac{n_{i,j}}{|d_i|} \tag{1.16}$$

où $n_{i,j}$ est le nombre d'occurrences du terme t_j dans le document d_i et $|d_i|$ correspond à la taille du document d_i : $|d_i| = \sum_j n_{i,j}$. Plus le terme t_j est fréquent dans le document d_i, plus $tf_{i,j}$ est élevé.

De même, la fréquence inverse du document idf_j du terme t_j est définie par [Salton *et al.*, 1983] :

$$idf_j = \ln\left(\frac{|\mathcal{D}|}{|\{d_i : t_j \in d_i\}|}\right) \tag{1.17}$$

où $|\mathcal{D}|$ désigne le nombre de documents dans le corpus et $|\{d_i : t_j \in d_i\}|$ le nombre de documents dans lesquels le terme t_j apparaît au moins une fois. Plus le terme t_j est rare dans le corpus, plus le coefficient idf_j est élevé.

Le poids $w_{i,j}$ du terme t_j dans le document d_i est ensuite obtenu en combinant les deux critères précédents :

$$w_{i,j} = tf_{i,j}.idf_j \tag{1.18}$$

Ce poids aura une valeur d'autant plus élevée que le terme t_j apparaît fréquemment dans le document d_i, mais peu dans les autres documents de la collection $\mathcal{D}$.

1.2.3.2 Pondération Okapi

D'autres mesures de la fréquence inverse du document ont été proposées en se basant sur des modèles probabilistes qui utilisent la notion de pertinence d'un document pour une requête. C'est le cas, par exemple, de l'approche introduite par Robertson et Spärck Jones qui utilise la notion de pertinence d'un document pour calculer idf_j [Robertson et Spärck Jones, 1976].

Dans l'exemple ci-dessous, la table de contingence 1.5 considère la distribution des documents pour un terme t_j et une requête q_k donnés :

- $N = |\mathcal{D}|$ est le nombre de documents dans la collection ;
- $R = |\mathcal{D}_k|$ est le nombre de documents pertinents pour la requête q_k ;
- $n = |\{d_i | t_j \in d_i, d_i \in \mathcal{D}\}|$ est le nombre de documents contenant t_j ;
- $r = |\{d_i | t_j \in d_i, d_i \in \mathcal{D}_k\}|$ le nombre de documents pertinents pour la requête q_k contenant t_j.

TAB. 1.5 – Table de contingence.

		Documents pertinents pour q_k	Documents non pertinents pour q_k	
		+	-	
Documents contenant t_j	+	r	n-r	n
Documents ne contenant pas t_j	-	R-r	N-n-R+r	N-n
		R	N-R	N

Différentes formules de pondération prenant en compte la pertinence des documents peuvent être extraites de cette table de contingence [Robertson et Spärck Jones, 1976] :

$$w_1 = \ln \frac{(\frac{r}{R})}{(\frac{n}{N})} \tag{1.19}$$

$$w_2 = \ln \frac{(\frac{r}{R})}{(\frac{n-r}{N-R})} \tag{1.20}$$

$$w_3 = \ln \frac{(\frac{r}{R-r})}{(\frac{n}{N-n})} \tag{1.21}$$

$$w_4 = \ln \frac{(\frac{r}{R-r})}{(\frac{n-r}{N-n-R+r})} \tag{1.22}$$

où les poids w_1, w_2, w_3 et w_4 mesurent la fréquence inverse du document pour un terme particulier. w_1 représente le rapport entre la proportion de documents pertinents qui contiennent le terme t_k et la proportion de documents sur la collection complète qui contiennent le terme t_k alors que w_2 représente le rapport entre la proportion de documents pertinents qui contiennent le terme t_k et la proportion de documents non pertinents qui contiennent le terme t_k. w_3 représente le rapport entre la chance que le terme t_k soit dans les documents pertinents et la chance que le terme soit dans les documents de la collection. La chance que le terme t_k soit dans les documents pertinents correspond à la proportion entre le nombre de documents pertinents et le nombre de documents non pertinents dans lesquels le terme apparaît. Enfin, w_4 représente le rapport entre la chance que le terme t_k soit dans les documents pertinents et la chance que le terme soit dans les documents non pertinents de la collection. Les poids w_1 et w_2 utilisent des proportions alors que w_3 et w_4 des chances. De plus, les poids w_1 et w_3 comparent la distribution de la pertinence des documents sur la collection alors que les poids w_2 et w_4 comparent cette pertinence par rapport à la distribution des documents non pertinents.

L'estimation de ces poids est souvent réalisée en pratiquant un lissage. Pour ce faire, les valeurs indiquées dans la table 1.5 sont remplacées par celles de la table 1.6 pour le calcul des formules de 1.19 à 1.22.

TAB. 1.6 – Table de contingence.

		Documents pertinents pour q_k	Documents non pertinents pour q_k	
		+	-	
Documents contenant t_j	+	r+0,5	n-r+0,5	n+1
Document ne contenant pas t_j	-	R-r+0,5	N-n-R+r+0,5	N-n+1
		R+1	N-R+1	N+2

L'une des pondérations les plus connues issue de ce modèle est la pondération Okapi BM25 [Robertson *et al.*, 1994]. Cette pondération se base sur le poids w_4 et est calculée

par :

$$idf_j = \ln \frac{(\frac{r+0.5}{R-r+0.5})}{(\frac{n-r+0.5}{N-n-R+r+0.5})} \tag{1.23}$$

$$\tag{1.24}$$

La pertinence des documents n'étant pas nécessairement connue, les valeurs de r et R sont généralement fixées à 0.

$$idf_j = \ln \left(\frac{N-n+0.5}{n+0.5} \right) \tag{1.25}$$

$$idf_j = \ln \left(\frac{|\mathcal{D}| - |\{d_i | t_j \in d_i\}| + 0.5}{|\{d_i | t_j \in d_i\}| + 0.5} \right) \tag{1.26}$$

Dans la pondération Okapi BM25 [Robertson *et al.*, 1994], la fréquence $tf_{i,j}$ du terme t_j dans le document d_i prend en compte la taille du document par rapport à la taille moyenne des documents de la collection et est calculée par :

$$tf_{i,j} = \frac{(k_1+1).n_{i,j}}{n_{i,j} + k_1(1-b+b\frac{|d_i|}{d_{avg}})} \tag{1.27}$$

où $n_{i,j}$ est le nombre d'occurrences du terme t_j dans le document d_i, $|d_i|$ représente la taille du document d_i, $d_{avg} = \frac{\sum_i |d_i|}{|\mathcal{D}|}$ est la taille moyenne des documents de la collection $\mathcal{D}$ et k_1 et b sont des paramètres du système. b permet de prendre plus ou moins en compte l'écart entre la taille du document d_i et la taille moyenne des documents de $\mathcal{D}$ et k_1 d'atténuer la fréquence du terme. Les valeurs par défaut de ces paramètres sont de $0,75$ (respectivement $2,0$) pour b (respectivement pour k_1).

Différentes représentations des données textuelles sont envisageables, avec pour chacune ses avantages et ses inconvénients qui dépendent généralement du compromis à faire entre l'efficacité et la simplicité du modèle. Dans la suite, le choix s'est porté sur le modèle vectoriel qui est classiquement utilisé pour les documents textuels représentés en sacs de mots et qui dans de nombreuses études a fait les preuves de son efficacité.

1.3 Représentation des images

Avec le développement des nouvelles technologies numériques, les images sont un support de plus en plus utilisé. Des milliards de photos[1] sont accessibles sur le site flickr[2], un site de partage de photos. Retrouver une image particulière dans une collection peut s'avérer très compliqué. En plus de la difficulté à représenter le contenu d'une image, il n'est pas simple pour un utilisateur d'expliciter ses besoins. Les images sont très riches en information comme l'illustre l'adage populaire, « un bon croquis vaut mieux qu'un long discours. ». Il n'est pas toujours possible d'extraire l'information sémantique contenue dans une image, alors que c'est justement ce que recherche l'utilisateur.

Les images et leurs spécificités présentées, les différentes approches utilisées pour représenter leur contenu visuel seront introduites avant de détailler celle qui décrit les images à l'aide de sacs de mots visuels.

[1] `http://blog.flickr.net/en/2010/09/19/5000000000/`

[2] `http://www.flickr.com`

1.3.1 Qu'est-ce qu'une image ?

D'après le dictionnaire, une image est une représentation de quelqu'un ou de quelque chose. Une statue peut être considérée comme une image en trois dimensions. Dans la suite, seules les images numériques en deux dimensions seront traitées. La notion d'image numérique définie, les différentes approches pour traiter ces images ainsi que leurs spécificités seront ensuite présentées.

1.3.1.1 Définition d'une image numérique

Une image numérique correspond à une image enregistrée sur un support numérique, comme un disque dur ou une clé USB. Techniquement elle correspond à une suite de bits représentée par les valeurs binaires 0 et 1. Deux grandes familles d'images numériques peuvent être distinguées : les images vectorielles et les images matricielles. Seules les images matricielles qui permettent de représenter tout type d'image seront considérées dans la suite. Une image matricielle est une image représentée par une matrice de pixels. La taille d'une image correspond au nombre de pixels dans la matrice. Comme le montre la figure 1.14[1], un pixel correspond à l'élément minimal qui compose une image.

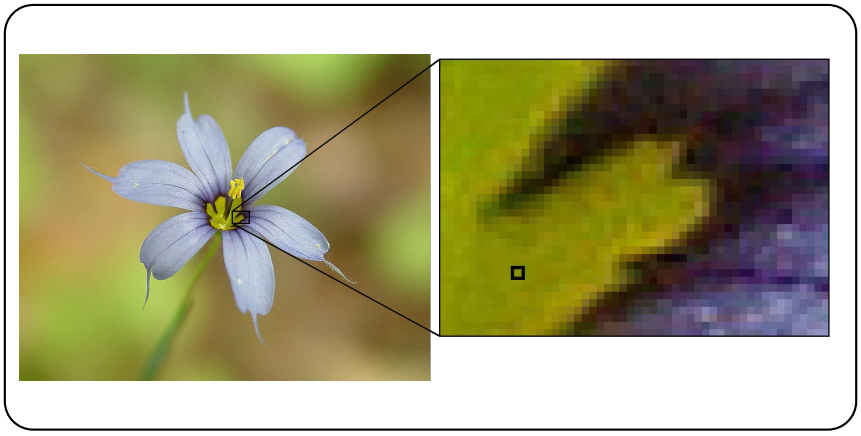


FIG. 1.14 – Représentation d'un pixel pour une image matricielle.

À chaque pixel est associée une couleur. Cette couleur peut correspondre à une valeur particulière ou à un index indiquant une couleur dans une palette. Une palette à deux couleurs peut par exemple servir à représenter une image en noir et blanc. Les images en noir et blanc sont généralement confondues avec les images en niveau de gris qui sont elles, représentées à l'aide d'une palette de 256 couleurs grises différentes. Les palettes pour représenter les images étaient très utilisées au début de l'internet car elles permettaient de transférer rapidement les images, le nombre de couleurs étant limité. Généralement, les palettes de couleurs sont composées de 16 ou 256 couleurs. En utilisant la fleur de la figure 1.14, différentes images utilisant une palette peuvent être obtenues comme présenté par la figure 1.15. Les technologies évoluent et les images sont maintenant représentées en utilisant toute la palette des couleurs visibles par un œil humain. Ces couleurs dites vraies peuvent être représentées par différents codages,

[1] source de la fleur : `http://commons.wikimedia.org/wiki/File:Maysmallpurpleflower.jpg`

le plus connu étant le codage RVB (Rouge, Vert, Bleu) ou RGB en anglais (Red, Green et Blue) dont les trois couleurs primaires en synthèse additive sont le rouge, le vert et le bleu [Süsstrunk *et al.*, 1999]. Chaque composante correspond à une valeur comprise entre 0 et 255. Il est ainsi possible de créer plus de 16 millions de couleurs différentes ($256^3 = 16\ 777\ 216$). Ce système de couleur n'est pas forcément adapté à la vision humaine et d'autres codages ont été proposés comme le codage TSL (Teinte Saturation et Lumière) ou HSL en anglais (Hue, Saturation et Lightness) qui représente une couleur en trois composantes correspondant à la teinte, la saturation et la lumière [Ford et Roberts, 1998]. Il existe également des codages, comme le codage CIE Lab, utilisant la luminance et la chromacité pour décrire la couleur [Zhang *et al.*, 1997].

image en noir et blanc	image en niveau de gris	image palette de 16 couleurs	image palette de 256 couleurs

FIG. 1.15 – Représentation de différentes images utilisant une palette.

1.3.1.2 Traitement des images

Le traitement d'image regroupe l'ensemble des techniques classiquement utilisées pour manipuler les images dans le but d'améliorer ou d'extraire leur contenu. Pour comprendre ces techniques, il faut s'intéresser aux conditions d'acquisition et de numérisation des images. Le principe général de l'acquisition et du traitement d'une image est présenté par la figure 1.16.

Une fois l'acquisition d'une image réalisée, la première étape du traitement vise à prétraiter cette dernière pour éliminer par exemple le bruit. La deuxième étape concerne le traitement de l'image comme par exemple la segmentation ayant pour but de regrouper en régions des pixels ayant des critères communs prédéfinis. Cette étape peut par exemple aider à séparer les objets du fond de l'image. Enfin la dernière étape effectue des mesures sur les régions pour détecter et retrouver les objets de l'image.

1.3.1.3 Spécificité des images

Contrairement aux documents textuels où chaque mot peut être interprété, les caractéristiques qu'il est possible d'extraire d'une image ne s'appréhendent pas aussi facilement.

Ainsi, détecter et reconnaître automatiquement la présence d'un objet, comme une fleur, sur une image n'est pas un problème simple. Malgré les nombreuses recherches menées en vision artificielle, aucune solution générale n'a été proposée pour résoudre ce problème.

La principale difficulté est liée à la variabilité importante des images. Une image numérique enregistrée lors de l'observation d'une scène par exemple sera très différente si les conditions d'acquisition, comme le point de vue, l'éclairage, les réglages de l'objectif, le capteur, etc. sont légèrement modifiés.

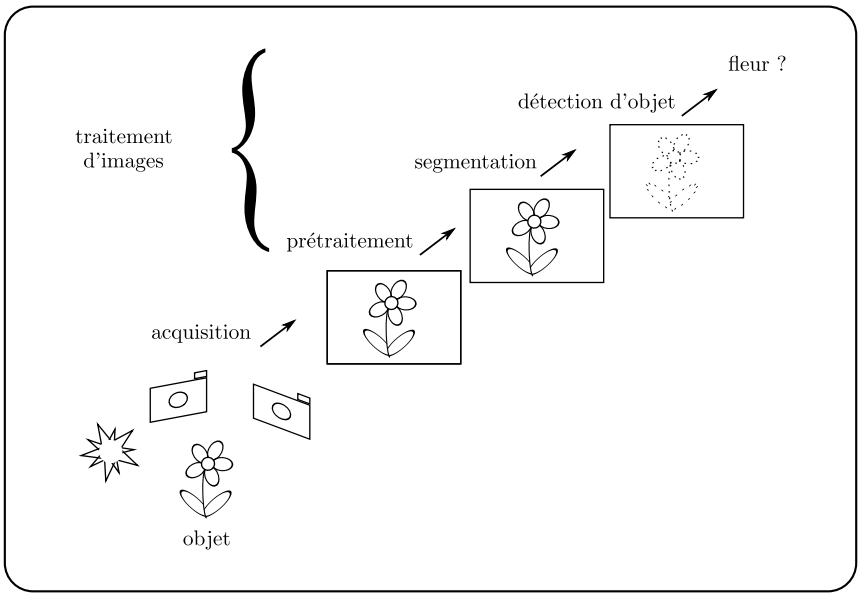


FIG. 1.16 – Acquisition et traitement d'images.

Le changement des conditions d'acquisition peut être représenté par différentes transformations affines comme l'illustre la figure 1.17. Dans un cas réel, ces tranformations peuvent aussi être tridimentionnelles.

FIG. 1.17 – Différentes transformations géométriques simulées par ordinateur.

Ainsi, les rotations, les translations ou les changements d'échelle peuvent correspondre à des points de vue différents d'un même objet. Les problèmes liés à l'éclairage de l'objet sont caractérisés par des changements de luminosité et de contraste comme l'illustre la figure 1.18

Pour contourner ces problèmes, les chercheurs se sont attachés à étudier des caractéristiques robustes aux différentes transformations que peut subir une image. De part la nature numérique des pixels, ces caractéristiques sont assez éloignées des caractéristiques sémantiques qui peuvent se trouver dans les mots d'un texte ; l'écart entre la description de bas niveau de l'image et ce qu'elle représente est un problème connu comme le fossé sémantique (*semantic gap*) [Smeulders *et al.*, 2000, Zhao et Grosky, 2002].

Plusieurs types de caractéristiques peuvent être distinguées. Les caractéristiques globales telles que les histogrammes de couleurs, les caractéristiques locales telles que les contours [Canny, 1986, Marr et Hildreth, 1980] ou encore les points caractéristiques [Harris et Stephens, 1988, Lowe, 1999].

FIG. 1.18 – Différentes transformations colorimétriques simulées par ordinateur.

1.3.1.4 Représentation des images

Les systèmes de recherche d'information pour les documents textuels étant très efficaces, les premières approches exploitant les images ont consisté à se servir le plus possible d'information textuelle [Goodrum, 2000]. Pour représenter une image, le texte qui l'entoure ainsi que son nom de fichier peuvent alors être utilisés [Torjmen *et al.*, 2009]. Cependant, les noms des fichiers ne sont pas toujours porteurs de sens, et les images ne sont pas forcément accompagnées d'un texte descriptif. Des alternatives consistent à annoter les images par des mots textuels. Ce procédé d'annotation peut être réalisé de façon manuelle [Russell *et al.*, 2008] mais il est très fastidieux. De plus pour retrouver les images annotées, il faut que les mots donnés par l'utilisateur qui recherche des images soient les mêmes que ceux qui ont servi à l'annotation, ce qui n'est pas évident si les deux utilisateurs sont différents [Furnas *et al.*, 1987]. L'annotation automatique a donc ensuite été proposée pour pallier ces problèmes. Cette annotation s'effectue généralement en associant des mots extraits d'un dictionnaire prédéfini à des zones détectées dans les images [Barnard *et al.*, 2003]. Pour une zone horizontale située en haut de l'image où la couleur dominante est le bleu, le mot ciel sera par exemple associé à l'image.

L'utilisation du texte est cependant limitée car elle ne permet pas de représenter toutes les spécificités des images et n'est pas nécessairement disponible pour une image quelconque. Des méthodes ne se servant que de l'information contenue dans l'image, sans utiliser de texte, se sont également développées. Les premières méthodes exploitent une représentation globale de l'image, à l'aide par exemple d'un histogramme [Boughorbel *et al.*, 2002]. Les histogrammes correspondent à une quantification de différentes caractéristiques, comme la couleur, la texture ou la forme. La distance calculée entre deux histogrammes permet ensuite de juger si deux images se ressemblent. Bien que cette représentation globale offre l'avantage d'être concise et rapide à calculer, deux histogrammes identiques peuvent correspondre à deux images très différentes et deux images qui se ressemblent ne conduisent pas forcément à deux histogrammes proches [Pass et Zabih, 1996].

La représentation des images s'est donc ensuite focalisée sur la description de parties de l'image correspondant à des régions issues d'une segmentation [Suematsu *et al.*, 2002]. Ces régions d'intérêt sont formées d'un ensemble de pixels contigus et dans l'idéal correspondent à des objets dans l'image. Les histogrammes précédemment calculés sur l'image globale sont alors calculés pour chaque région dans le but d'obtenir des descriptions plus précises. Dans des applications comme la détection d'objets, les régions de taille plus ou moins importantes sont souvent sujettes aux problèmes d'occultation et les approches basées sur la détection de points d'intérêts se sont développées [Lowe, 1999]. Les points d'intérêt sont des points correspondant à des caractéristiques locales

particulières de la matrice de pixel (par exemple les coins) et qui sont robustes à certaines transformations de l'image (par exemple les transformations affines). Ils sont souvent associés à un voisinage dans lequel on peut calculer un descripteur local. Une mise en correspondance des descriptions permet de détecter et de rechercher des objets particuliers dans des images [Lowe, 1999]. Ces méthodes sont très efficaces pour faire de la correspondance entre deux objets identiques, mais elles ne sont pas utilisables directement dans le cadre d'une recherche de catégories d'objets. De plus en plus, des nouvelles approches inspirées des modèles de sacs de mots textuels se développent. Ces approches utilisent un vocabulaire visuel créé à partir de différentes descriptions et représentent les images sous forme de sacs de mots visuels [Csurka *et al.*, 2004].

1.3.2 Représentation locale des images

Les représentations locales se sont révélées plus performantes notamment pour la classification ou la recherche d'image par le contenu. Parmi celles-ci, on distingue les approches basées sur la segmentation [Beucher et Lantuejoul, 1979, Shi et Malik, 2000], et celles basées sur la détection de points ou de régions d'intérêt [Matas *et al.*, 2002, Lowe, 2004]. Dans la suite, seules les méthodes décrivant le voisinage de points d'intérêt seront considérées. Ces méthodes se décomposent en deux principales étapes. La première est une étape de détection de points caractéristiques. Le but est de choisir des points sur l'image au voisinage desquels sera calculée une description locale. Ces points doivent être capables de capturer l'information visuelle présente dans l'image. Ils peuvent être obtenus soit par un détecteur spécifique, soit par un échantillonnage régulier ou aléatoire. La plupart du temps, on associe une région d'intérêt à chacun de ces points. La taille et la forme de la région peut être déterminée automatiquement ou fixée par l'utilisateur. La deuxième étape est une étape de description locale. Il s'agit de calculer des paramètres associés à la région entourant chaque point caractéristique. Ces paramètres traduisent une information locale de forme, de texture ou de couleur associée au point. Après une présentation des approches pour la détection de points d'intérêts, les méthodes pour décrire leur voisinage seront présentées.

1.3.2.1 Détection de points d'intérêt

La détection des points d'intérêts peut ainsi s'effectuer de trois façons différentes, en utilisant soit un algorithme de détection de points d'intérêt, soit une grille régulière, soit une sélection aléatoire de points. La première approche inspirée des premiers détecteurs de points d'intérêt consiste à chercher des points intersections. Ces détecteurs sont généralement associés à la détection de coins et correspondent à des jonctions en L, en T, en X, en Y comme illustré par la figure 1.19.

Différentes méthodes algorithmiques peuvent être considérées pour détecter des points d'intérêt. La première des approches consiste dans un premier temps à détecter les contours de l'image à l'aide par exemple de méthodes de filtrage, comme les filtres de Prewitt, de Sobel ou de Canny illustrés par la figure 1.20 [Canny, 1986, Koschan, 1995]. Les contours obtenus sont ensuite parcourus pour détecter les points d'intersection et les points où la courbure est maximale.

La deuxième approche utilise des modèles théoriques de points d'intérêt [Deriche et Giraudon, 1990]. Un modèle correspond à un motif qui est ensuite recherché dans les images. Chacune des correspondances trouvées correspond alors à la détection du point d'intérêt considéré. Cette approche ne permet de détecter qu'un nombre limité

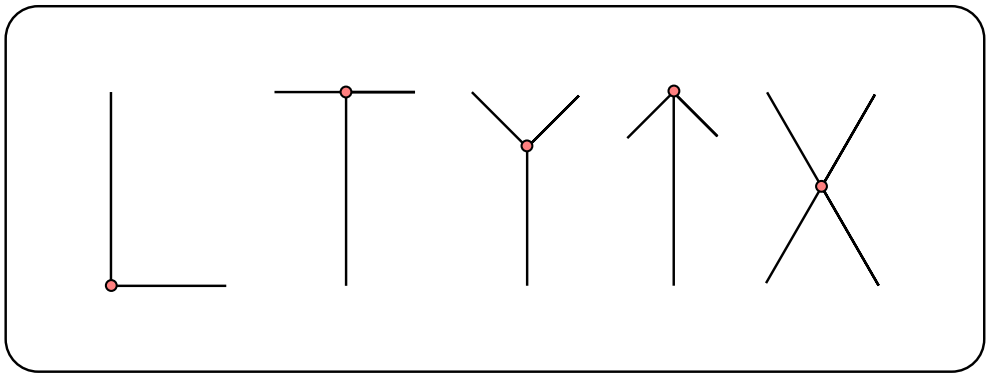


FIG. 1.19 – Différentes jonctions

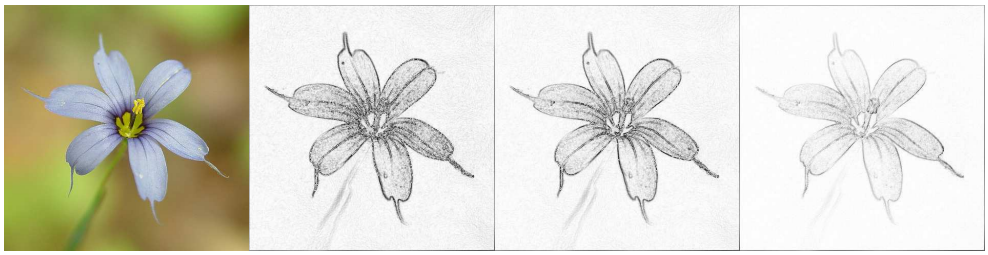


FIG. 1.20 – Différents filtres (Sobel, Prewitt et Canny) pour la détection de contour.

de types de points d'intérêt comme les coins car il n'existe pas de modèle théorique générique pour définir l'ensemble des points d'intérêt.

La troisième approche calcule les points d'intérêts sur les images converties en niveau de gris. La valeur des pixels correspond alors à une fonction d'intensité utilisée pour détecter les changements brutaux de l'intensité. Les premiers détecteurs proposés utilisent directement cette fonction d'intensité pour détecter les points d'intérêts [Moravec, 1977, Harris et Stephens, 1988, Smith et Brady, 1997, Trajkovic et Hedley, 1998]. La détection de points d'intérêt est illustrée par la figure 1.21.

FIG. 1.21 – Détection des points d'intérêts basée sur la recherche des coins dans l'image. De gauche à droite, les méthodes sont la détection de Moravec [Moravec, 1977], de Harris et Stephens [Harris et Stephens, 1988] et de Trajkovic [Trajkovic et Hedley, 1998].

D'autres approches cherchent les points d'intérêts stables après modification de la fonction d'intensité à l'aide d'un filtre, comme le filtre gaussien. Un espace d'échelle est

obtenu en utilisant sur la même image des filtres de différentes intensités. Les points d'intérêt qui se retrouvent sur les différentes échelles ainsi créées sont jugés stables. Le détecteur le plus connu basé sur cette approche est le détecteur SIFT [Lowe, 1999]. D'autres méthodes utilisent également le laplacien (LoG) ou la matrice hessienne pour détecter des points d'intérêt (DoH) [Mikolajczyk et Schmid, 2004]. Contrairement aux approches précédentes, ces approches permettent généralement de définir également une région caractéristique entourant le point. La forme de cette région est souvent circulaire ou elliptique et ses paramètres sont déterminés à l'aide d'une approche multi-échelle [Lowe, 1999].

Dans un contexte plus général où le but est de représenter les images pour diverses applications, les méthodes qui considèrent un sous-ensemble de pixels choisis régulièrement dans toute l'image, peuvent également être utilisées. Contrairement aux méthodes algorithmiques qui essayent de détecter des points spécifiques, les méthodes régulières ont pour but de représenter le maximum d'information en recouvrant toute l'image [Li et Perona, 2005, Vogel et Schiele, 2002]. La taille de la région est dans ce cas définie en relation avec la période d'échantillonnage des points éventuellement en adoptant des approches multi-échelles.

D'autres méthodes utilisent un sous-ensemble de pixels de l'image choisis de façon aléatoire. Une fois les pixels sélectionnés, une zone entourant ces pixels est définie pour pouvoir associer une description aux points d'intérêt [Vidal-Naquet et Ullman, 2003, Maree *et al.*, 2005]. La figure 1.22 illustre une sélection aléatoire des zones caractéristiques et un découpage régulier pour une taille de région égale à la période d'échantillonnage des points.

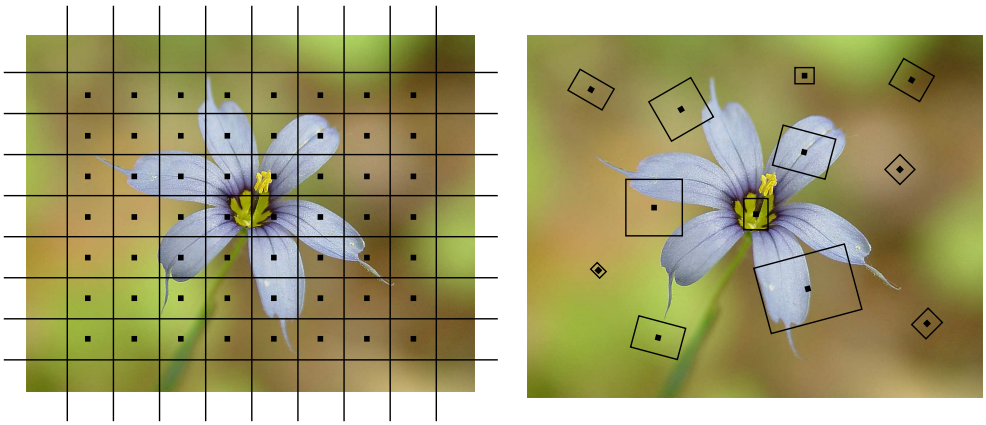


FIG. 1.22 – Détection de points d'intérêt sur une grille régulière ou de façon aléatoire.

1.3.2.2 Description de caractéristiques

À chaque zone caractéristique est associé un vecteur descriptif qui permet d'ajouter des informations complémentaires sur, par exemple, la couleur, la texture et la forme. Ces informations peuvent correspondre aux mêmes descriptions utilisées pour la représentation globale des images, mais elles sont calculées sur un sous-ensemble de pixels plutôt que sur l'image entière.

Les informations sur la couleur sont les plus simples à exploiter. Il est par exemple possible d'affecter pour une zone considérée, la couleur dominante, la couleur moyenne,

la proportion de l'ensemble des couleurs [Swain et Ballard, 1991, Tollari et Glotin, 2006]. L'utilisation de la couleur permet de différencier les scènes d'intérieurs et d'extérieurs, mais aussi de détecter les zones de végétation, le ciel, la mer etc. L'espace RVB n'étant pas très bien adapté à la perception humaine, d'autres espaces de couleur comme TSL peuvent être utilisés [Schettini *et al.*, 2001].

Les textures sont également très utilisées pour détecter les zones où des motifs se répètent. L'herbe, le sable, les vagues, les grilles sont des motifs facilement identifiables. Les méthodes les plus utilisées pour décrire la texture utilisent des filtres. Ces filtres sont appliqués aux images et permettent d'identifier des motifs particuliers. Les filtres de Gabor sont les plus connus et les plus efficaces [Chen *et al.*, 2004, Howarth et Rüger, 2004].

Lorsque les zones caractéristiques sont le résultat d'une segmentation ou d'une détection de régions d'intérêt, la forme des ces zones peut être décrite par des descripteurs de formes comme la transformée de Fourier-Mellin [Derrode et Ghorbel, 2001]. Les formes circulaires permettent par exemple de détecter les yeux d'un visage [Toennies *et al.*, 2002].

Le descripteur sans doute le plus utilisé est le descripteur SIFT qui pour une zone caractéristique donnée associe 128 valeurs correspondant à un histogramme d'orientation du gradient dans huit directions pour 16 fenêtres autour du pixel comme illustré par la figure 1.23 [Lowe, 1999].

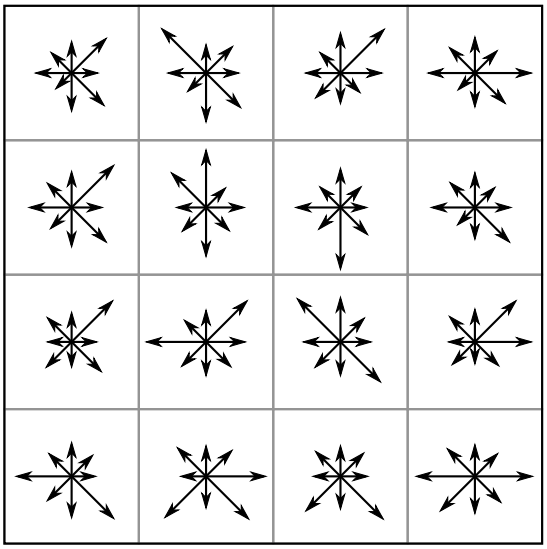


FIG. 1.23 – Illustration du descripteur sift.

1.3.3 Représentation des images à l'aide d'un sac de mots visuels

La recherche d'objets dans les images et la mise en correspondance d'images s'effectuent généralement en utilisant directement les caractéristiques locales calculées sur les images [Lowe, 2004]. Dans des contextes plus généraux comme la catégorisation d'images ou la recherche d'information, ces caractéristiques locales sont regroupées à l'aide d'un algorithme de classification non-supervisée pour former un vocabulaire de mots visuels comme l'illustre la figure 1.24 [Sivic et Zisserman, 2003, Jurie et Triggs,

2005, Yang *et al.*, 2007]. L'approche la plus utilisée correspond à l'algorithme des nuées dynamique ou k-means [MacQueen, 1967, Diday, 1971]. L'étape de classification des motifs peut être vue comme une étape d'échantillonnage adaptatif de l'espace des caractéristiques. Cela permet donc de réduire la dimension de cet espace et de calculer ensuite des histogrammes d'occurrence de mots visuels. Un mot visuel s'interprète comme une classe de motifs présents fréquemment sur les images.

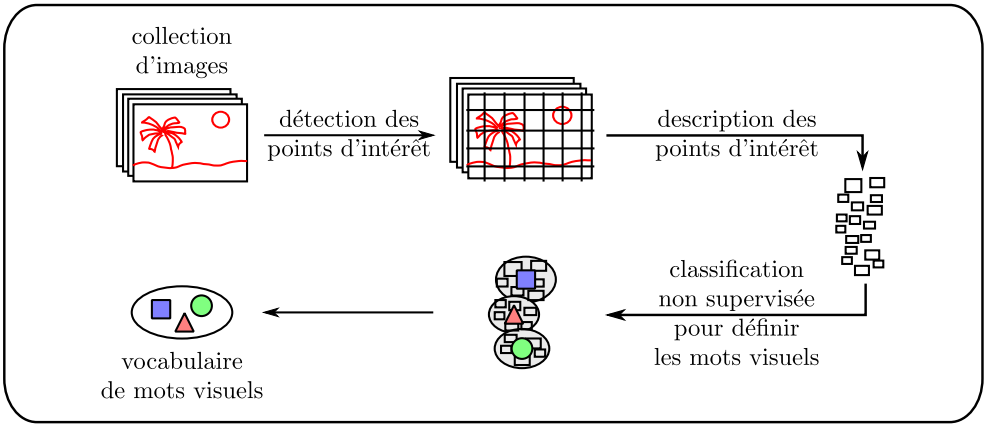


FIG. 1.24 – Création d'un vocabulaire de mots visuels.

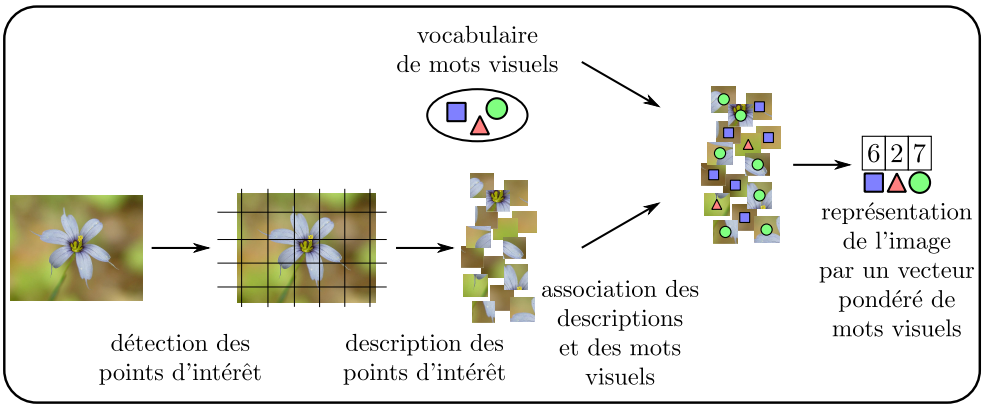


FIG. 1.25 – Représentation d'une image à l'aide d'un modèle de sac de mots visuels.

Le vocabulaire de mots visuels est ensuite utilisé pour représenter une image comme le montre la figure 1.25. Les régions de l'image sont associées à des mots visuels en fonction de leurs descriptions. L'image peut alors se représenter à l'aide d'un vecteur pondéré de mots visuels. La représentation à l'aide d'un modèle de sac de mots visuels dépend de nombreux paramètres comme le choix de la détection et de la description des points d'intérêts [Mikolajczyk et Schmid, 2004], le choix de l'algorithme de classification [Jurie et Triggs, 2005], la taille du vocabulaire visuel, le nombre de mots visuels à calculer par image ainsi que leur normalisation [Nowak *et al.*, 2006]. Dans la suite, ces paramètres feront l'objet d'une étude plus précise.

Des études récentes en catégorisation d'images ont montré que la détection régulière des points d'intérêts donne les meilleurs résultats [Jurie et Triggs, 2005] surtout

lorsque la taille du vocabulaire est importante [Nowak *et al.*, 2006]. Comme pour la représentation des documents textuels, les mots visuels sont pondérés pour chaque image. L'occurrence des mots visuels est utilisée, mais le pouvoir discriminant de ces derniers est aussi pris en compte [Yang *et al.*, 2007]. Certaines approches utilisent une pondération binaire des mots visuels en considérant l'apparition ou la non apparition des mots dans les images [Nowak *et al.*, 2006]. Une sélection de ces mots visuels à considérer pour la représentation est parfois effectuée en ne conservant que ceux qui maximisent l'information mutuelle entre un mot et une classe [Nowak *et al.*, 2006]. Un des enjeux actuels concernant la description de l'information visuelle est de prendre en compte les informations spatiales. Plusieurs approches ont été proposées dans ce domaine en se basant par exemple sur un découpage régulier de l'image, à partir d'une décomposition pyramidale ou en construisant des phrases visuelles [Lazebnik *et al.*, 2006, Cao *et al.*, 2010, Albatal *et al.*, 2010].

Le choix de la représentation des images est difficile, principalement à cause de l'existence du fossé sémantique entre la description et l'interprétation de ces images. L'approche qui est de plus en plus utilisée, décrit les images en sacs de mots tout comme le sont les documents textuels. Ces informations textuelles et visuelles ont été présentées séparément, mais pour décrire efficacement les documents multimédias, il est indispensable de les combiner.

1.4 Combinaison multimodale

De par leur nature, les documents multimédias peuvent être représentés en exploitant les différents types d'information, textuel et visuel, qu'ils contiennent. Différents cadres théoriques peuvent être utilisés pour traiter le problème de la fusion de données : les approches bayésiennes reposent sur la théorie des probabilités pour estimer les imperfections de l'information considérée ; la théorie des croyances proposée par Dempster en 1967 [Dempster, 1967] repose sur la notion de preuves et utilise les fonctions de croyance et le raisonnement plausible ; la théorie des possibilités s'appuie sur la théorie des ensembles flous développée dans les années 60 [Zadeh, 1965] et permet de représenter les imprécisions et les incertitudes de l'information. Dans la suite, ces cadres théoriques ne seront pas étudiés et les méthodes de fusion décrites seront celles qui fusionnent les représentations des documents, c'est-à-dire les fusions précoces, et celles qui permettent de combiner les résultats d'un classement ou d'une recherche, c'est-à-dire les fusions tardives [Snoek *et al.*, 2005].

1.4.1 Fusion précoce

La fusion précoce, parfois appelée fusion de caractéristiques ou encore fusion bas niveau, a pour but de représenter un document multimédia par un vecteur unique qui englobe toutes les modalités comme l'illustre la figure 1.26. La plus simple des fusions précoces correspond à une simple concaténation des vecteurs de représentation des différentes modalités [Zhou et Huang, 2002, Magalhaes et Rüger, 2007]. Les dimensions des vecteurs représentatifs n'étant pas les mêmes pour différentes modalités, il est parfois nécessaire de les normaliser avant d'effectuer la mise en correspondance [Sclaroff *et al.*, 1999]. Les deux principales normalisations correspondent à la normalisation linéaire et à la normalisation gaussienne. La transformation linéaire permet de normaliser la liste

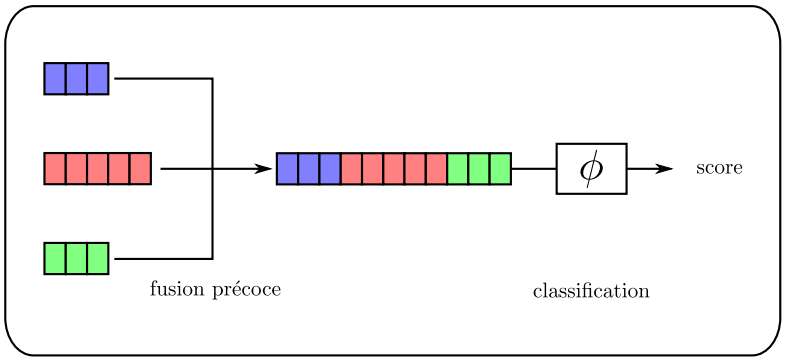


FIG. 1.26 – Illustration d'une fusion précoce.

$X = (x_1, \ldots, x_i, \ldots, x_{|X|})$ en une liste $X' = (x'_1, \ldots, x'_i, \ldots, x'_{|X|})$ où chaque valeur x'_i, comprise dans l'intervalle $[a, b]$, est obtenue par :

$$x'_i = b.\frac{x_i - min(X)}{max(X) - min(X)} + a \tag{1.28}$$

où $min(X)$ et $max(X)$ correspondent respectivement aux valeurs minimales et maximales de la liste X. En supposant que X suive une distribution gaussienne d'espérance μ_X et d'écart type σ_X, la transformation gaussienne permet de transformer X en une liste X' qui suit une distribution centrée, c'est-à-dire que la moyenne $\mu_{X'}$ de X' vaut 0, et réduite, c'est-à-dire que l'écart-type $\sigma_{X'}$ de X' vaut 1. La valeur x'_i est calculée par :

$$x'_i = \frac{x_i - \mu_X}{\sigma_X} \tag{1.29}$$

où μ_X et σ_X correspondent respectivement à la moyenne et l'écart-type de la liste X.

La concaténation de plusieurs vecteurs peut conduire à un vecteur unique de très grande dimension. Des méthodes pour réduire la dimensionalité du vecteur concaténé peuvent être utilisées dans le but d'optimiser les caractéristiques en considérant les problèmes de redondance ou de complémentarité. L'analyse en composantes principales et l'analyse en composantes indépendantes correspondent aux deux approches les plus utilisées [Wu *et al.*, 2004]. Sur le même principe, l'analyse sémantique latente introduite initialement pour les documents textuels a pour but de construire des concepts liés aux documents et aux termes du vocabulaire [Deerwester *et al.*, 1990]. Elle a été appliquée sur des vecteurs concaténant des informations textuelles et visuelles [Pham *et al.*, 2008, Zhao et Grosky, 2002].

1.4.2 Fusion tardive

Contrairement aux méthodes précoces, les fusions tardives, de classifieurs ou de scores, sont des méthodes de haut niveau qui s'appliquent soit aux classifieurs soit aux scores comme illustré par la figure 1.27. En fonction de l'application considérée, ces fusions correspondent soit à la combinaison des sorties de classifieurs, soit à la fusion des scores obtenus par plusieurs système de recherche d'information.

Quelle que soit l'application considérée, le principal avantage de la fusion tardive est la possibilité d'utiliser des approches spécifiques adaptées à chaque modalité.

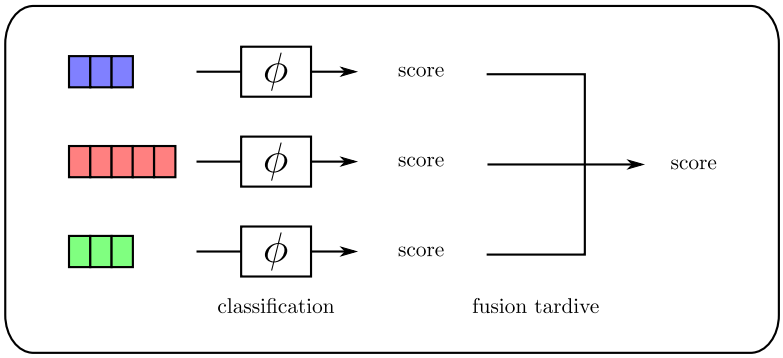


FIG. 1.27 – Illustration d'une fusion tardive.

En catégorisation, pour illustrer la combinaison de plusieurs classifieurs, le problème des courses de chevaux sera utilisé [Freund et Schapire, 1999]. Dans ce problème, le but est de créer un classifieur capable de prédire le cheval gagnant de la prochaine course. Un turfiste expert rencontrera des difficultés s'il doit définir toutes les hypothèses pour prédire quel cheval à le plus de chance de gagner. En revanche, pour les courses qui se sont déjà déroulées, ce dernier peut trouver des critères qui expliquent le classement final. Ainsi, une hypothèse peut être, par exemple, de miser sur le cheval qui a gagné le plus de courses sur les dix dernières. En utilisant une autre course, il pourra compléter ces hypothèses en remarquant qu'un cheval particulier est meilleur sur les courses de courtes distances. Ces hypothèses prises séparément n'offrent pas une chance de réussite élevée car elles sont simples et ne prennent pas assez de paramètres en compte. Cependant, elles semblent raisonnables par rapport à des hypothèses qui consisteraient à choisir aléatoirement les chevaux. Une fois la liste des hypothèses établie, il faudra trouver la meilleure façon de les combiner pour optimiser les chances de trouver le cheval gagnant de la prochaine course. Dans ce contexte, il faut donc répondre à deux problèmes : le premier est de trouver de quelle façon il faut présenter les courses à l'expert pour que les hypothèses générées soient le plus efficace possible ; le deuxième est de savoir comment combiner les différentes hypothèses pour créer un unique classifieur qui soit le plus efficace possible.

Pour répondre à ces deux problèmes, il existe deux grandes méthodes principales qui sont le bagging [Breiman, 1996] et le boosting [Freund et Schapire, 1995]. Le bagging pour boostrap aggregating, est une méthode qui répond au premier problème de présentation des courses à l'expert par un tirage aléatoire avec remise. Le but est de générer des ensembles d'apprentissage les plus diversifiés possibles. Le bagging utilise ensuite une combinaison par vote non pondéré pour répondre au deuxième problème de la combinaison des classifieurs générés. La méthode du boosting traite les deux problèmes différemment. La génération des différents ensembles d'apprentissage est réalisée par une approche déterministe. La distribution des exemples de l'échantillon d'apprentissage est modifiée pour mettre en avant les exemples difficiles à apprendre. Ces exemples correspondent à ceux pour lesquels les classifieurs générés n'ont pas encore permis de les classer correctement. La combinaison finale correspond à un vote pondéré pour mettre en avant les classifieurs les plus efficaces.

Pour répondre au deuxième problème de la combinaison de classifieur, les méthodes les plus simples correspondent aux méthodes par vote [Bahler et Navarro, 2000]. Le vote majoritaire attribue la classe qui a été le plus souvent retournée par les classifieurs. Le

vote majoritaire pondéré correspond au même principe que le vote majoritaire, mais en attribuant des poids différents aux classifieurs. Ainsi la classe associée à un document est celle qui a obtenu le plus grand poids pondéré. Il est également possible d'affecter une classe si et seulement si tous les classifieurs s'accordent à retourner la même classe. Dans ce cas, des documents peuvent ne pas être affectés à des classes si les résultats des classifieurs ne sont pas cohérents.

Une autre solution consiste, pour un document donné, à utiliser les résultats de plusieurs classifieurs comme une nouvelle représentation de ce document. Cette représentation est ensuite utilisée par un dernier classifieur qui retourne la classe du document en fonction de cette nouvelle représentation. Cette méthode fait référence à la méthode dite du stacking [Wolpert, 1992] utilisée, par exemple, pour différencier des images prises en intérieur ou en extérieur [Szummer et Picard, 1998].

En recherche d'information, le but est de fusionner la liste ordonnée des documents obtenus par un ou plusieurs systèmes de recherche d'information utilisant une ou plusieurs modalités. De ce fait, les méthodes de fusion utilisées en catégorisation ne sont pas toutes applicables à la recherche d'information, et inversement.

Les méthodes les plus utilisées se rapprochent des combinaisons par vote en catégorisation [Fox et Shaw, 1994]. Parmi elles, la plus simple (CombSUM) consiste à ajouter pour un document ces scores obtenus par les différents systèmes de recherche d'information. Il est également possible de calculer le score moyen (CombANZ) ou de prendre en compte le nombre de systèmes qui retournent le document (CombMNZ). Cette dernière méthode de combinaison a fait l'objet de plusieurs études montrant qu'une amélioration pouvait être envisagée lorsque les systèmes retournent un ensemble commun de documents pertinents [Lee, 1997], mais surtout si des documents pertinents étaient retournés en premier [Beitzel *et al.*, 2003]. Ainsi le but de ces méthodes est de prendre en compte l'union et l'intersection des résultats [Kompaore et Mothe, 2008].

Plus généralement, ce sont les combinaisons linéaires qui sont le plus utilisées pour combiner les résultats de plusieurs systèmes [Bartell *et al.*, 1994, Vogt et Cottrell, 1999]. Ces combinaisons sont efficaces quand le nombre de modalités à fusionner est raisonnable [Yan et Hauptmann, 2003]. D'autres approches non linéaires utilisent des méthodes à noyaux pour combiner les différentes informations [Lanckriet *et al.*, 2004]. Ces méthodes ont été appliquées pour des documents vidéos dans le cadre de la compétition TRECVID [Ayache *et al.*, 2007].

1.5 Positionnement du travail

Notre but est de proposer un modèle flexible combinant les informations textuelles et visuelles des documents multimédias. Dans ce chapitre, nous nous sommes intéressés à la représentation des documents multimédias composés de texte et d'images. Historiquement et principalement pour des raisons techniques, nous avons vu que les recherches ont d'abord été effectuées sur les documents textuels.

Dans la suite, nous nous intéresserons uniquement au texte plat sans prendre en compte la structure éventuellement associée à ces documents textuels. Pour la représentation de ces documents, nous utiliserons le modèle vectoriel qui les décrit sous forme de sacs de mots à l'aide d'un vocabulaire de mots textuels issus des documents. La taille de ce vocabulaire peut être très importante et nous verrons dans le chapitre 2 que dans un contexte de catégorisation multiclasse de documents, il est possible de le réduire considérablement sans dégrader les résultats du classement. Ce problème de classe-

ment utilise des algorithmes issus du domaine de l'apprentissage automatique, mais ces derniers ne sont généralement pas adaptés au problème de catégorisation multilabel. Nous étudierons ensuite dans le même chapitre les différentes possibilités d'utilisation de ces algorithmes dans un contexte de catégorisation multilabel et nous introduirons une nouvelle méthode permettant de sélectionner le nombre d'étiquettes à associer à un document.

La représentation des images à l'aide de sac de mots est de plus en plus utilisée et les travaux actuels s'intéressent à la prise en compte de l'information spatiale dans les images [Cao *et al.*, 2010]. Représenter les images par des sacs de mots n'est cependant pas aussi direct que pour les documents textuels et dépend d'un nombre important de paramètres. Dans le chapitre 3, nous étudierons, dans un contexte de catégorisation d'images, la création de vocabulaires visuels en considérant différentes détections et descriptions locales des images. Nous comparerons différentes pondérations des mots visuels inspirées principalement du modèle tf.idf utilisé pour les documents textuels. Les différents vocabulaires obtenus seront ensuite combinés pour étudier l'apport de plusieurs informations visuelles.

Dans un contexte de recherche d'information de documents multimedias, nous nous intéresserons dans le chapitre 4 à la combinaison linéaire des résultats obtenus par les différentes modalités et étudierons l'apport de l'information visuelle. Nous montrerons qu'il est possible d'apprendre efficacement, à partir d'un échantillon d'apprentissage, le poids à accorder à chaque type d'information (textuelle et visuelle) en effectuant une recherche exhaustive. Quand le nombre de paramètres à apprendre augmente, cette approche coûteuse sera cependant limitée. Nous introduirons alors une nouvelle méthode, basée sur l'analyse discriminante, permettant d'apprendre ces poids.

Chapitre 2

Représentation de l'information textuelle

Avant de considérer des documents multimédias, nous allons dans un premier temps nous focaliser sur des documents composés uniquement de texte. Dans l'état de l'art, nous avons présenté différents modèles de représentation des documents textuels. Le modèle vectoriel, considéré dans la suite, représente un document par un sac de mots. Cette représentation nécessite la création d'un vocabulaire de mots généralement obtenu à partir de tous les mots qui apparaissent dans les documents étudiés. Même pour un nombre limité de documents, la taille du vocabulaire généré peut être très importante.

L'objectif de ce chapitre est double. D'une part, il s'agit d'étudier le problème de la réduction de la taille du vocabulaire dans un contexte de catégorisation multiclasse. Nous introduisons un nouveau critère qui permet de sélectionner les termes les plus pertinents pour réduire la taille du vocabulaire. D'autre part, il s'agit d'étendre le problème de la catégorisation au cas multilabel où un document peut être associé à plusieurs catégories. Nous proposons une alternative aux méthodes classiques de sélection du nombre de catégories à affecter à un document.

2.1 Réduction du vocabulaire

Le vocabulaire composé des mots des documents peut avoir une taille très importante même pour une collection réduite. Or tous ces mots ne sont pas forcément utiles et informatifs. De plus, la malédiction de la dimension peut poser des problèmes de représentation lorsque la taille du vocabulaire est très importante. Il est alors intéressant de chercher à réduire la taille du vocabulaire. Différentes études ont montré que sur des collections de documents relativement courts, comme Reuters-21578[1], réduire la taille du vocabulaire de 90% ne dégradait pas les résultats [Yang et Pedersen, 1997, Sebastiani, 2002, Fragoudis *et al.*, 2005]. Notre but est d'introduire un nouveau critère pour sélectionner les mots qui permettent de différencier les classes les unes des autres. Après avoir présenté les différentes approches pour réduire le vocabulaire, nous les comparons à notre critère sur des collections plus grandes avec des documents de tailles conséquentes. Ceci nous permettra de confronter nos résultats avec ceux des études précédentes obtenus sur les collections classiques de documents courts.

[1] `http://www.daviddlewis.com/resources/testcollections/reuters21578/readme.txt`

2.1.1 Différentes approches pour réduire le vocabulaire

Dans un contexte généraliste, le vocabulaire n'est pas limité à des mots textuels, mais à des caractéristiques. L'extraction et la sélection de caractéristiques sont deux approches principales qui peuvent être distinguées pour résoudre le problème de la réduction de la taille du vocabulaire [Lewis, 1992a, Fodor, 2002, Cunningham, 2007]. L'extraction de caractéristiques consiste à combiner les caractéristiques du vocabulaire pour en créer de nouvelles en moindre nombre. Pour des documents textuels, cela peut correspondre à la substitution des mots synonymes par un unique mot. La sélection de caractéristiques vise à supprimer du vocabulaire les caractéristiques jugées inutiles ce qui correspond par exemple, pour du texte, à supprimer les mots vides tels que les articles.

2.1.1.1 Extraction de caractéristiques

L'extraction de caractéristiques vise à simplifier la quantité d'information nécessaire pour représenter les documents. Les caractéristiques de sortie correspondent à une combinaison de celles d'entrée. Dans un contexte de documents textuels, le but est alors de maximiser l'efficacité du vocabulaire en essayant de réduire les problèmes de polysémie, d'homonymie, de synonymie dont souffre le vocabulaire d'origine.

L'analyse en composantes principales (ACP ou PCA de l'anglais : *Principal Component Analysis*) est la plus célèbre des méthodes d'extraction de caractéristiques. Le but de cette méthode est de représenter des caractéristiques possiblement corrélées par un nombre plus petit de caractéristiques, les composantes principales, non corrélées. D'un point de vue géométrique, cette méthode peut s'interprêter comme la représentation des données dans un nouvel espace géométrique qui possède des axes d'inertie maximale. Elle peut également se comprendre d'un point de vue statistique par la recherche des composantes principales qui expliquent au mieux la variance des données. Pour un nombre de composantes principales choisi, le sous espace obtenu avec l'analyse en composantes principales correspond à la transformation linéaire optimale qui possède la plus grande variance des données. Des généralisations de cette méthode utilisant des transformations non linéaires ont été proposées, comme l'analyse en composantes indépendantes qui cherche des caractéristiques indépendantes et pas seulement non corrélées [Tang *et al.*, 2005]. Basée sur le même principe que l'analyse en composante principale, l'analyse sémantique latente (LSA, de l'anglais : *Latent Semantic Analysis*) est une méthode qui a pour but de générer un vocabulaire de concepts en regroupant les mots qui apparaissent dans des contextes similaires. Cette méthode utilise une décomposition en valeurs singulières sur la matrice des occurrences des mots dans les documents. Il faut ensuite fixer un paramètre pour définir le nombre de concepts à calculer et qui représentera la taille du vocabulaire de sortie. Ce paramètre permettra de sélectionner les plus grandes valeurs singulières pour approcher la matrice des occurrences d'origine. Les concepts ainsi créés forment une combinaison linéaire de mots du vocabulaire d'origine. Un des inconvénients de l'analyse sémantique latente est de ne pas capturer la polysémie des mots. De plus, les concepts générés sont souvent difficilement interprétables [Deerwester *et al.*, 1990].

L'analyse en composantes principales à noyaux projette de façon non linéaire les données dans un espace de très grande dimension avant d'effectuer l'analyse en composantes principales [Schölkopf *et al.*, 1998].

L'analyse discriminante linéaire correspond à la version supervisée de l'analyse en

composantes principales où le but est de minimiser la distance entre les caractéristiques d'une même classe et de maximiser la distance entre les classes [Fisher, 1936, Saporta, 2006].

Enfin l'analyse en composantes curvilignes est une transformation non linéaire qui essaye de préserver localement les distances entre les observations après la projection dans le nouvel espace des caractéristiques [Demartines et Hérault, 1995].

Le principal problème des méthodes d'extraction de caractéristiques réside en la difficulté d'interprétation des nouvelles caractéristiques créées. Dans la suite, nous ne considérerons donc que les méthodes de réduction du vocabulaire qui permettent de sélectionner les caractéristiques pertinentes du vocabulaire plutôt que de créer de nouvelles caractéristiques en nombre limité.

2.1.1.2 Sélection de caractéristiques

Dans le contexte de la catégorisation de documents, la sélection de caractéristiques correspond à la mise en relief des caractéristiques du vocabulaire qui permettent d'obtenir un bon classement. Dans ce cadre, la sélection des mots pertinents se fait à l'aide d'un indicateur qui calcule l'importance d'un terme t_j par rapport à une classe c_k. L'indicateur permet alors de trier les mots du vocabulaire par pertinence et la réduction du vocabulaire s'effectue en ne conservant qu'un certain nombre de mots.

L'un des indicateurs les plus connus est le χ^2, originellement issu du test du χ^2 introduit par Pearson [Pearson, 1900], qui permet d'évaluer l'indépendance entre un terme t_j et une classe c_k :

$$\chi^2(t_j, c_k) = \frac{|\mathcal{D}|.[P(t_j, c_k).P(\bar{t_j}, \bar{c_k}) - P(\bar{t_j}, c_k).P(t_j, \bar{c_k})]^2}{P(t_j).P(\bar{t_j}) - P(c_k).P(\bar{c_k})} \tag{2.1}$$

où

- $P(t_j)$ est la probabilité de voir apparaître le terme t_j dans la collection $\mathcal{D}$;
- $P(c_k)$ est la probabilité de voir apparaître la classe c_k dans la collection $\mathcal{D}$;
- $P(\bar{c_k}) = 1 - P(c_k)$;
- $P(t_j, c_k)$ est la probabilité qu'un document appartienne à la classe c_k et contienne le terme t_j ;
- $P(t_j, \bar{c_k})$ est la probabilité qu'un document n'appartienne pas à la classe c_k et contienne le terme t_j ;
- $P(\bar{t_j}, c_k)$ est la probabilité qu'un document appartienne à la classe c_k et qu'il ne contienne pas le terme t_j ;
- $P(\bar{t_j}, \bar{c_k})$ est la probabilité qu'un document n'appartienne pas à la classe c_k et ne contienne pas le terme t_j.

Ce critère prend une valeur nulle si t_j et c_k sont indépendants. Au contraire, le terme t_j est considéré comme d'autant plus représentatif de la classe c_k que $\chi^2(t_j, c_k)$ est élevé. La puissance au numérateur tend à équilibrer l'impact des probabilités indiquant une corrélation positive entre t_j et c_k ($P(t_j, c_k)$ et $P(\bar{t_j}, \bar{c_k})$) et celles correspondant à une corrélation négative ($P(t_j, \bar{c_k})$ et $P(\bar{t_j}, c_k)$) [Ng *et al.*, 1997]. Une extension du χ^2, GSS, a été proposé en supprimant la taille de la collection $|\mathcal{D}|$ constante pour tous les termes ainsi que les probabilités du dénominateur qui tendent à favoriser aussi bien les termes les plus rares que les classes les plus petites [Galavotti *et al.*, 2000].

$$GSS(t_j, c_k) = P(t_j, c_k).P(\bar{t_j}, \bar{c_k}) - P(\bar{t_j}, c_k).P(t_j, \bar{c_k}) \tag{2.2}$$

Le gain d'information $IG(t_j, c_k)$, la divergence de Kullback-Leibler ou l'entropie relative sont d'autres critères utilisés pour sélectionner un sous ensemble de descripteurs initiaux discriminants [Caropreso *et al.*, 2001] :

$$IG(t_j, c_k) = P(t_j, c_k) \ln(\frac{P(t_j, c_k)}{P(t_j)P(c_k)}) + P(\bar{t_j}, c_k) \ln(\frac{P(\bar{t_j}, c_k)}{P(\bar{t_j})P(c_k)}) \tag{2.3}$$

Seuls les termes pour lesquels le gain d'information est le plus important sont considérés comme caractéristiques de la classe.

Dans un contexte de catégorisation, les termes qui apparaissent peu souvent dans les documents d'apprentissage de la collection ont un pouvoir prédictif limité et ne doivent pas être conservés pour représenter les documents en vue de leur classification. Pour un terme t_j donné, cette hypothèse se traduit par un faible nombre de documents dans lesquels le terme t_j apparaît. La fréquence du terme t_j peut être calculée de manière globale sur la collection par $DFG(t_j)$, ou de façon locale pour chaque classe c_k avec $DFL(t_j, c_k)$ de la façon suivante :

$$DFG(t_j) = P(t_j) \qquad DFL(t_j, c_k) = P(t_j|c_k) \tag{2.4}$$

où $P(t_j|c_k)$ correspond à la probabilité que le terme t_j apparaisse sachant que les documents considérés appartiennent à la classe c_k.

Dans la pratique, cette technique est souvent utilisée de façon systématique en éliminant tous les termes apparaissant dans au plus 1, 2 ou 3 documents. Elle est souvent combinée à d'autres méthodes de sélection de descripteurs [Dumais *et al.*, 1998, Li et Jain, 1998, Wiener *et al.*, 1995].

D'autres critères peuvent également être utilisés comme l'information mutuelle (MI) [Novoviccaronová *et al.*, 2004], le coefficient de Ng-Goh-Low (NGL) [Ng *et al.*, 1997], le rapport des chances (Odds Ratio (OR) en anglais) [Mladenic, 1998]. Ces critères ne seront pas présentés dans la partie expérimentale car il n'ont pas conduit à de bons résultats à l'image de certaines études qui soulignent la non robustesse de certains de ces critères [Sebastiani, 2002, Uchyigit et Ma, 2008]. Nous pouvons souligner que le critère MI mesure la dépendance statistique de deux variables. Les mauvais résultats obtenus par ce critère s'explique principalement par le fait qu'il met en avant les mots qui sont rares. La suppression de ces mots à l'aide du critère DFG permet de résoudre en partie ce problème [Uchyigit et Ma, 2008]. Le critère NGL est inspiré du χ^2 et retourne une valeur positive s'il existe une forte correlation entre le mot t_j et la classe c_k alors qu'une valeur négative sera obtenue s'il existe une forte correlation entre t_j et $\bar{c_k}$. Il donne de meilleurs résultats que le χ^2 si seuls les termes pour lesquels les valeurs sont positives, sont conservés [Ng *et al.*, 1997]. Enfin le critère OR a été proposé pour sélectionner les mots à conserver dans le cadre du retour de pertinence. Il est sensé donner de meilleurs résultats que les critères MI et IG mais comme il favorise les mots qui sont corrélés avec une classe particulière, si un mot n'apparaît que très rarement dans une classe mais jamais dans les autres, il obtiendra un score élevé pour cette classe [Mladenic, 1998]. Il favorise donc, comme le critère MI, les mots qui sont rares.

2.1.2 Proposition d'un nouveau critère de sélection : CCDE

Dans le contexte de la catégorisation de documents, un mot qui n'apparaît que dans un seul document ne permettra pas de faire un classement efficace et peut dans ce cas être supprimé du vocabulaire utilisé pour représenter les documents. De plus, un

terme qui est réparti dans tous les documents de toutes les classes n'est également pas informatif pour le classement puisqu'il ne permet pas de caractériser une classe. Partant de ces hypothèses, nous avons défini un critère permettant de calculer l'importance d'un terme t_j pour une classe c_k afin de ne conserver que les termes pertinents qui apparaissent relativement souvent et qui sont concentrés dans les documents d'une même classe.

En s'inspirant du concept *tf.idf* utilisé pour représenter l'importance d'un terme t_j dans un document d_i, nous proposons de calculer un critère de sélection qui permet de calculer l'importance d'un terme t_j pour une classe c_k. Nous introduisons dans un premier temps, $CCD(t_j, c_k)$ qui comme $tf_{i,j}$, calcule l'importance du terme t_j dans la classe c_k. Ensuite, nous utilisons l'entropie du terme t_j pour mettre en avant la discriminance du terme t_j par rapport aux classes sur la collection, à l'image de idf_j qui mesure l'importance du terme t_j sur l'ensemble des documents de la collection $\mathcal{D}$, .

2.1.2.1 Importance d'un terme pour une classe

Pour calculer l'importance du terme t_j par rapport à une classe c_k, nous introduisons le critère de différence de couverture de classe $CCD(t_j, c_k)$(*Category Coverage Difference*) qui mesure l'importance du terme t_j pour la classe c_k. Ce critère utilise la probabilité $P(t_j|c_k)$ que le terme t_j apparaisse dans un document sachant que ce document appartient à la classe c_k :

$$P(t_j|c_k) = \frac{|\{d_i \in c_k | t_j \in d_i\}|}{|\{d_i \in c_k\}|} \tag{2.5}$$

Pour qu'un terme t_j soit important pour une classe c_k, il faut que le terme t_j apparaisse dans beaucoup de documents de la classe c_k et donc que $P(t_j|c_k)$ soit élevée. Cette information est nécessaire, mais ne suffit pas à caractériser l'importance du terme t_j par rapport à la classe c_k puisque cette hypothèse est vérifiée pour les mots qui sont très fréquents comme les mots vides. De ce fait, il faut également que $P(t_j|c_k)$ soit plus élevée que $P(t_j|\bar{c_k})$ qui représente la probabilité que le terme t_j apparaisse dans un document sachant que ce document n'appartient pas à la classe c_k. En prenant en compte ces deux hypothèses, le critère $CCD(t_j, c_k)$ qui mesure l'importance du terme t_j pour la classe c_k est obtenu en calculant la différence entre $P(t_j|c_k)$ et $P(t_j|\bar{c_k})$:

$$CCD(t_j, c_k) = P(t_j|c_k) - P(t_j|\bar{c_k}) \tag{2.6}$$

Le rapport entre $P(t_j|c_k)$ et $P(t_j|\bar{c_k})$ aurait pu être utilisé mais la différence a été privilégiée pour éviter les cas particuliers où la probabilité $P(t_j|\bar{c_k})$ est égale à 0. Pour que la valeur $CCD(t_j, c_k)$ soit maximale, il faut que $P(t_j|c_k)$ soit égale à 1, c'est-à-dire que le terme t_j apparaîsse dans tous les documents de la classe c_k et que $P(t_j|\bar{c_k})$ soit égale à 0, c'est-à-dire que le terme t_j n'apparaîsse pas dans les documents qui n'appartiennent pas à la classe c_k ; plus simplement, le critère $CCD(t_j, c_k)$ est égal à 1 si le terme t_j apparaît dans tous les documents qui appartiennent à la classe c_k et uniquement dans ceux-là. La valeur de $CCD(t_j, c_k)$ est au contraire minimale lorsque le terme t_j apparaît dans tous les documents qui n'appartiennent pas à la classe c_k et uniquement ceux-là ; dans ce cas, la valeur de $CCD(t_j, c_k)$ est égale à -1.

2.1.2.2 Représentativité du terme sur la collection

Un terme est représentatif pour une collection si son apparition est concentrée dans une seule classe. Soit $n_{j,k}$ le nombre d'apparitions du terme t_j dans la classe c_k. La

fréquence du terme t_j dans la classe c_k est définie par :

$$tf_j^k = \frac{n_{j,k}}{\sum_{k=1}^{|\mathcal{C}|} n_{j,k}} \tag{2.7}$$

Pour calculer la représentativité du terme t_j dans la collection, l'entropie de Shannon $E(t_j)$ du terme t_j est définie par [Shannon, 1948] :

$$E(t_j) = -\sum_{k=1}^{|\mathcal{C}|} tf_j^k \times \ln_2(tf_j^k) \tag{2.8}$$

Cette entropie est minimale, égale à 0, si le terme t_j apparaît dans une seule classe. Au contraire, elle est maximale, si le terme t_j est réparti équitablement dans toutes les classes. Cette entropie maximale E_{max} correspond donc à :

$$E_{max} = -\sum_{k=1}^{|\mathcal{C}|} \frac{1}{|\mathcal{C}|} \times \ln_2(\frac{1}{|\mathcal{C}|}) = \ln_2(|\mathcal{C}|) \tag{2.9}$$

2.1.2.3 Critère de sélection CCDE

De même que la pondération $tf.idf$, le critère de sélection $CCDE(t_j, c_k)$ calcule l'importance du terme t_j pour la classe c_k. Il est obtenu par :

$$CCDE(t_j, c_k) = CCD(t_j, c_k) \times \frac{E_{max} - E(t_j)}{E_{max}} \tag{2.10}$$

où l'information de discriminance du terme t_j est réalisée par une normalisation de $E(t_j)$ par rapport à E_{max}. Ce critère retourne donc des scores élevés pour les termes qui sont discriminants pour une classe donnée et qui sont principalement concentrés dans cette classe.

2.1.3 Expérimentations

Le critère de sélection $CCDE$ que nous avons proposé a été évalué et comparé à d'autres critères sur la collection XML Mining 2008 (XML08) [Géry *et al.*, 2009, Largeron et Moulin, 2010, Largeron *et al.*, 2011]. Cette collection sera tout d'abord décrite. Le protocole expérimental et les résultats seront ensuite présentés.

2.1.3.1 Présentation de la collection XML08

La collection XML08 est extraite du corpus XML Wikipedia proposé par Denoyer et Gallinari [Denoyer et Gallinari, 2006]. Cette collection est utilisée dans le cadre de la compétition INEX 2008[1]. Plus de détails sur la collection et notre participation à cette compétition sont donnés dans l'Annexe A.1. La collection XML08 est composée de 114 366 documents répartis dans 15 catégories représentant chacune un sujet particulier comme le sport, le tourisme, les États-Unis, etc. Un résumé des caractéristiques de la collection est donné dans la table 2.1.

[1] `http://www.inex.otago.ac.nz/`

TAB. 2.1 – Brève description de la collections XML09.

	XML08
Nombre de documents	114 366
Longueur moyenne des documents	423,44
Longueur moyenne des documents (terme unique)	117,46
Taille du vocabulaire original	652 876
Taille du vocabulaire prétraité	77 706

2.1.3.2 Protocole expérimental

Chaque document de la collection est affecté à une unique catégorie qui peut être différente d'un document à l'autre. L'échantillon d'apprentissage $\mathcal{D}_A$ est composé de 10% des documents de la collection, soit 11437 documents. Les 102929 documents restants sont utilisés pour l'échantillon de test $\mathcal{D}_T$. Pour effectuer notre étude comparative, nous avons représenté à l'aide d'un vecteur pondéré $tf.idf$ les documents de la collection. L'étude a ensuite été réalisée en comparant les résultats de la classification obtenus pour différentes tailles de vocabulaire en utilisant plusieurs critères de sélection.

Prétraitement de la collection XML08

La taille du vocabulaire original de la collection XML08 est de 652 876 mots. Avant d'effectuer la classification, la taille du vocabulaire a été réduite par différentes approches classiques en traitement de texte. La lemmatisation de Porter a été appliquée réduisant ainsi le nombre de mots à 560 209 [Porter, 1980]. D'autres termes non discriminants et succeptibles de dégrader la catégorisation ont été enlevés comme les nombres, les mots composés de moins de trois caractères, les mots apparaissant moins de trois fois dans la collection ainsi que ceux figurant dans tous les documents. La taille du vocabulaire obtenu est réduite à 161 609 mots sur l'ensemble des documents de la collection $\mathcal{D}$ et 77 706 mots sur les documents de $\mathcal{D}_A$. Chaque document d_i est ensuite représenté par un vecteur de poids $w_{i,j}$ calculé pour chaque terme t_j par :

$$w_{i,j} = \frac{n_{i,j}}{|d_i|} \times \ln \frac{|\mathcal{D}|}{|\{d_i : t_j \in d_i\}|} \tag{2.11}$$

qui correspond à la représentation classique du modèle $tf.idf$ où $n_{i,j}$ correspond au nombre de fois que le terme t_j apparaît dans le document d_i, $|d_i|$ est la taille du document d_i, $|\{d_i : t_j \in d_i\}|$ représente le nombre de documents dans lesquels le terme t_j apparaît et $|D|$ correspond au nombre de documents dans la collection D.

Utilisation d'un critère de sélection pour réduire la taille du vocabulaire

Soit $CS(t_j, c_k)$, un critère de sélection quelconque. Le but est d'utiliser $CS(t_j, c_k)$ pour réduire la taille du vocabulaire T en un vocabulaire T' de taille inférieure ($|T'| < |T|$) permettant de représenter les documents. Deux approches peuvent être envisagées pour créer ce nouveau vocabulaire [Sebastiani, 2002].

L'approche globale consiste à déterminer pour chaque terme t_j le meilleur score $CS(t_j, c_k)$ obtenu pour les différentes classes de $\mathcal{C}$. Cette première approche ne peut se faire que si les scores $CS(t_j, c_k)$ sont comparables entre les différentes classes comme par exemple les critères DFL, χ^2 et IG. Un paramètre n est ensuite utilisé pour sélectionner les n termes qui ont obtenu les scores $CS(t_j, c_k)$ les plus élevés. Le paramètre n permet alors de réduire la taille du vocabulaire T' à n mots, c'est-à-dire $|T'| = n$.

La seconde approche locale, utilisée dans la suite, n'impose pas une comparabilité des scores $CS(t_j, c_k)$ entre les classes. Pour une classe c_k donnée, les valeurs $CS(t_j, c_k)$ sont triées par ordre décroissant pour tous les mots du vocabulaire T. Après avoir fixé un nombre n de termes à sélectionner pour chaque classe, nous conservons l'ensemble des mots correspondant aux n premiers termes qui ont obtenu les valeurs $CS(t_j, c_k)$ les plus élevées. Le vocabulaire T' est ensuite construit en faisant l'union de ces termes pour toutes les classes de $\mathcal{C}$. La taille du vocabulaire ne peut donc pas être fixé préalablement, car un même terme peut être sélectionné dans plusieurs classes, alors qu'il n'apparaîtra qu'une seule fois dans le vocabulaire final T'. Les différentes expérimentations ont été réalisées en considérant les valeurs de n variant de 10 à 5000 mots sélectionnés par classe. L'algorithme 1 détaille l'approche utilisée pour la création de T'.

Données : n, $CS(t_j, c_k), j = 1 \dots |T|, k = 1 \dots |\mathcal{C}|$
Résultat : T'
$T' = \emptyset$;
pour $k \leftarrow 1$ ***à*** $|\mathcal{C}|$ **faire**
  $Ttmp = sort(CS(t_j, c_k) : j = 1 \dots |T|)$;
  pour $i \leftarrow 1$ ***à*** n **faire**
    $T' = T' \cup \{Ttmp[i]\}$;
  fin
fin
retourner T'

Algorithme 1: Création du vocabulaire réduit T' à partir des scores $CS(t_j, c_k)$ obtenu par un critère de sélection pour un nombre n de termes à conserver par classe.

Comparaison avec les différents critéres de sélection

Les différents critères de sélection peuvent se calculer en construisant une table de contingence pour le terme t_j et la classe c_k illustée en table 2.2. Dans cette table, on note :

- A, le nombre de documents de la collection appartenant à la classe c_k et contenant le terme t_j ;
- B, le nombre de documents de la collection n'appartenant pas à la classe c_k et contenant le terme t_j ;
- C, le nombre de documents de la collection appartenant à la classe c_k et ne contenant pas le terme t_j ;
- D, le nombre de documents de la collection n'appartenant pas à la classe c_k et ne contenant pas le terme t_j.

Le nombre de documents correspond à $A + B + C + D$ et sera noté N avec $N = |\mathcal{D}_A|$.

Différents critères ont été comparés avec CCDE mais seuls GSS, IG, χ^2, DFG et DFL qui ont donné les meilleurs résultats seront présentés. La table 2.3 regroupe l'ensemble

TAB. 2.2 – Table de contingence.

	c_k	$\bar{c_k}$
t_j	A	B
$\bar{t_j}$	C	D

des critères utilisés pour la comparaison et leur expression en fonction des termes de la table de contingence 2.2.

TAB. 2.3 – Table du calcul des critères en utilisant la table de contingence.

critère	formule
$CCDE(t_j, c_k)$	$= \begin{cases} CCD(t_j, c_k) \times \frac{E_{max} - E(t_j)}{E_{max}} \\ \frac{AD-BC}{(A+C)(B+D)} \times \frac{E_{max} - E(t_j)}{E_{max}} \end{cases}$
$DFG(t_j)$	$= \begin{cases} P(t_j) \\ \frac{A+B}{N} \end{cases}$
$DFL(t_j, c_k)$	$= \begin{cases} P(t_j \mid c_k) \\ \frac{A}{A+C} \end{cases}$
$IG(t_j, c_k)$	$= \begin{cases} P(t_j, c_k) \ln(\frac{P(t_j, c_k)}{P(t_j)P(c_k)}) + P(\bar{t_j}, c_k) \ln(\frac{P(t_j, c_k)}{P(\bar{t_j})P(c_k)}) \\ -\frac{A+C}{N} \ln(\frac{A+C}{N}) + \frac{A}{N} \ln(\frac{A}{A+B}) + \frac{C}{N} \ln(\frac{C}{C+D}) \end{cases}$
$\chi^2(t_j, c_k)$	$= \begin{cases} \frac{N.[P(t_j, c_k).P(\bar{t_j}, \bar{c_k}) - P(\bar{t_j}, c_k).P(t_j, \bar{c_k})]^2}{P(t_j).P(t_j) - P(c_k).P(\bar{c_k})} \\ \frac{N.(AD-BC)^2}{(A+B)(A+C)(B+D)(C+D)} \end{cases}$
$GSS(t_j, c_k)$	$= \begin{cases} P(t_j, c_k).P(\bar{t_j}, \bar{c_k}) - P(\bar{t_j}, c_k).P(t_j, \bar{c_k}) \\ \frac{AD-BC}{N^2} \end{cases}$

Évaluation

L'évaluation des critères de sélection se fait en comparant pour différentes tailles de vocabulaire le taux de bien classés obtenu après avoir effectué la classification sur le vocabulaire réduit T'. Le taux de bien classés correspond au pourcentage de documents pour lesquels les étiquettes associées aux documents sont correctes. Le taux de réduction permettant de passer de l'index initial T à l'index réduit T' est également pris en compte. Ce taux de réduction est égal à : $(1 - \frac{|T'|}{|T|})$. Le but est d'avoir un taux de réduction élevé sans dégrader le taux de bien classés. Les scores sont ensuite comparés aux résultats obtenus à partir du vocabulaire de référence T en calculant la perte par rapport à ce résultat de référence.

Classification

Pour chaque expérimentation, la catégorisation est réalisée en utilisant la représentation sous forme de vecteurs de poids des documents calculés à partir du vocabulaire T'. L'algorithme de classification utilisée est basé sur des machines à vecteurs de support (SVM) effectuée grâce au programme liblinear [Fan *et al.*, 2008].

2.1.3.3 Résultats

Pour le vocabulaire initial T, le taux de bien classés obtenu est de 78,79%. Ce résultat servira de *référence* pour comparer les résultats obtenus avec les vocabulaires réduits.

TAB. 2.4 – Taux de réduction et diminution du taux de bien classés par rapport au résultat de référence

critère	taille index	taux de bien classés	taux de réduction	perte par rapport au résultat de *référence*
référence	77706	78,79%	0%	0%
$CCDE$	5495	77,86%	92,93%	1,18%
GSS	5571	77,21%	92,83%	2,01%
DFL	4486	76,85%	94,23%	2,46%
IG	6929	77,90%	91,08%	1,13%
DFG	7879	77,14%	89,86%	2,09%

La table 2.4 représente le taux de réduction et la diminution du taux de bien classés associé par rapport au résultat de référence obtenu pour le vocabulaire T original. Le critère χ^2 n'est pas représenté car les résultats sont nettement moins bons que pour les autres critères. Une des limites de l'approche locale, utilisée pour réduire le vocabulaire, est qu'elle ne permet pas de comparer les résultats pour une taille de vocabulaire fixée puisque les tailles d'index obtenus ne sont pas les mêmes pour les différents critères. Ainsi, il n'est pas possible de comparer les résultats pour un taux de réduction identique entre les différents critères de sélection. De ce fait, nous avons considéré un taux de réduction proche de 90% pour tous les critères. La table 2.4 montre que de très bons résultats de classification peuvent être obtenus après une très forte sélection des termes. Ainsi, quel que soit le critère de sélection présenté dans la table, une réduction de l'index de 90% n'entraine qu'une diminution de 2% du taux de bien classés. Notons que pour une diminution du taux de bien classés de seulement 1%, le critère $CCDE$ permet une réduction de la taille du vocabulaire de 93%. Ces résultats corroborent les conclusions des précédentes études obtenues sur des collections de documents courts [Yang et Pedersen, 1997].

La figure 2.1 présente les taux de bien classés obtenus en fonction de la taille du vocabulaire T' réduit. Le critère CCDE, que nous avons introduit, fournit les meilleurs résultats aussi bien pour des valeurs faibles de n que pour des valeurs élevées. L'efficacité de ce critère apparaît surtout pour les plus forts taux de sélection puisqu'il permet d'obtenir plus de 75% de bien classés pour une taille d'index de l'ordre de 2000 termes. Viennent ensuite, par ordre de performance décroissante, les critères GSS, IG, CC et DFL. Les critères DFG et χ^2 s'avèrent les moins efficaces, y compris pour ce dernier avec des taux de sélection très faibles. Globalement, ces résultats confirment les études comparatives antérieures [Caropreso *et al.*, 2001, Ng *et al.*, 1997, Sebastiani, 2002, How et Kiong, 2005] qui ont déjà souligné l'efficacité des critères IG et GSS pour la catégorisation de textes plats de taille réduite. Toutefois, il convient de signaler le mauvais score obtenu par le χ^2 qui d'après [Sebastiani, 2002] devrait fournir des résultats com-

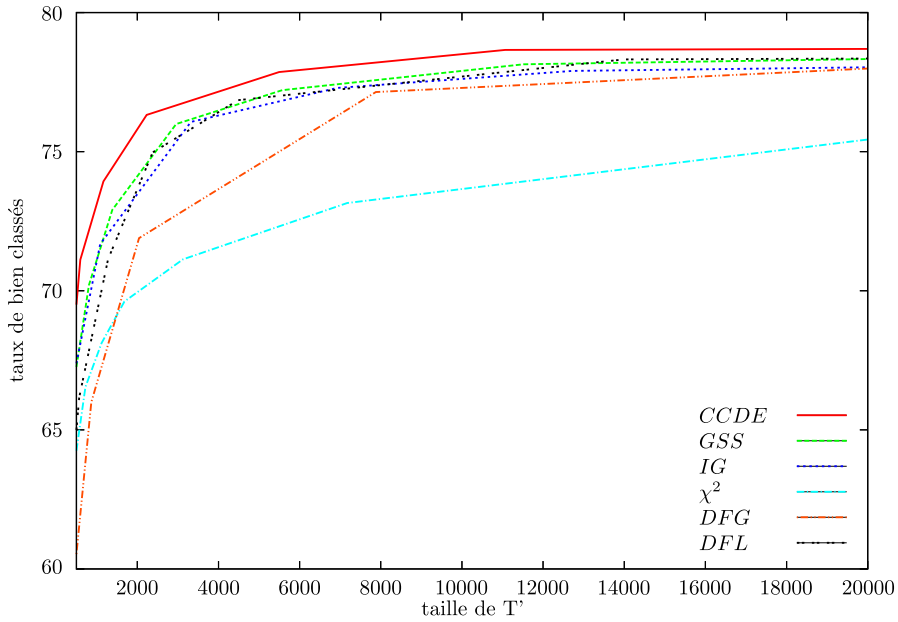


FIG. 2.1 – Taux de bien classés en fonction de la taille de l'index pour différents critères

parables à ceux obtenus par IG. Ainsi, cette étude comparative confirme l'importance de la sélection des descripteurs pour la catégorisation de documents textuels déjà vérifiée pour le traitement de textes plats et met en évidence l'intérêt de l'utilisation de la fréquence des termes par le biais de l'entropie.

2.2 Problème de la catégorisation multilabel

Dans la partie précédente, nous avons étudié l'information textuelle dans le contexte de la catégorisation multiclasse où les documents appartiennent à une seule catégorie. Ce contexte est inspiré du classement réalisé dans les bibliothèques où les livres ne doivent être associés qu'à une seule catégorie pour pouvoir être retrouvés facilement. Cependant, il n'est pas rare de voir plusieurs catégories affectées, par exemple, à un article Wikipédia. En effet, l'article sur la classification automatique possède par exemple trois catégories : analyse des données, philosophie de la connaissance, apprentissage automatique.

Cette partie s'intéresse au problème multilabel qui concerne de plus en plus de documents. Il existe différentes méthodes qui permettent de répondre au problème de la catégorisation multilabel avec d'une part, celles qui adaptent les algorithmes de classification et d'autre part celles qui transforment le problème multilabel en un ou plusieurs problèmes multiclasses [Tsoumakas et Katakis, 2007].

Dans la suite, nous ne considérons pas les algorithmes adaptés aux problèmes de catégorisation multilabel. Nous pouvons néanmoins citer certains algorithmes classiques ayant fait l'objet d'adaptation au problème multilabel : $C4.5$ [Clare et King, 2001], Adaboost avec Adaboost.MH et Adaboost.MR [Schapire et Singer, 2000] ainsi que les k plus proches voisins [Luo et Zincir-Heywood, 2005, Zhang et Zhou, 2005].

Pour exploiter les algorithmes de classification multiclasse, il faut préalablement transformer le problème multilabel. Dans un premier temps, nous expliquons comment

transformer ce problème en un ou plusieurs problèmes multiclasses. Nous nous focalisons ensuite sur les transformations qui ne répondent que partiellement au problème multilabel et qui nécessitent la sélection du nombre de catégories à conserver après avoir effectué une catégorisation multiclasse. Nous introduisons une nouvelle méthode de sélection que nous confrontons aux méthodes classiques sur différentes collections.

2.2.1 Transformation des problèmes multilabels

Il existe différentes méthodes qui transforment les données d'un problème multilabel en un ou plusieurs problèmes multiclasses. Pour illustrer ces méthodes, considérons la table 2.5 qui présente quatre documents, concernant le parachute, la natation, l'origami et l'Australie, classés selon quatre catégories qui correspondent à la notion du pliage, à la géographie, à l'eau et au sport.

TAB. 2.5 – Exemple d'une catégorisation multilabel de documents

Documents \ Catégories	Pliage	Géographie	Eau	Sport
Australie		×	×	
Natation			×	×
Origami	×			
Parachute	×			×

2.2.1.1 Sélection des catégories ou des documents

Les deux approches les plus simples pour transformer le problème multilabel en un problème multiclasse consistent soit à ne conserver qu'une catégorie choisie aléatoirement parmi celles affectées aux documents comme illustré par la table 2.6, soit à ne conserver que les documents qui ne sont associés qu'à une seule catégorie comme le montre la table 2.7.

TAB. 2.6 – Transformation du problème de la catégorisation multilabel en choisissant aléatoirement une seule catégorie par document.

Documents \ Catégories	Pliage	Géographie	Eau	Sport
Australie			×	
Natation			×	
Origami	×			
Parachute				×

Nous pouvons remarquer que dans l'exemple de la sélection aléatoire des catégories illustré par la table 2.6, la catégorie géographie ne pourra pas être apprise. De même en ne conservant que les documents qui possèdent une seule catégorie, les catégories géographie, eau et sport ne pourront pas être apprises comme le montre la table 2.7.

TAB. 2.7 – Transformation du problème de la catégorisation multilabel en ne conservant que les documents qui sont affectés à une seule catégorie.

Documents \ Catégories	Pliage	Géographie	Eau	Sport
Origami	×			

Dans les deux cas, le fait qu'un document appartienne à plusieurs catégories est complètement perdu et ne peut donc pas être une solution entièrement satisfaisante. Ces méthodes ne seront donc pas considérées dans la suite.

2.2.1.2 Approche binaire

L'une des approches les plus utilisées pour appréhender le problème de catégorisation multilabel consiste à apprendre séparément chaque classe en utilisant autant de classifieurs binaires qu'il y a de classes. La transformation des données est illustrée dans la table 2.8.

TAB. 2.8 – Transformation du problème de la catégorisation multilabel en utilisant autant de classifieurs binaires qu'il y a de classes. Les catégories *Eau*, *Géographie*, *Pliage* et *Sport* sont représentées respectivement par les tables 2.8(a), 2.8(b), 2.8(c), 2.8(d)

(a)

	Eau	$\overline{Eau}$
Australie	×	
Natation	×	
Origami		×
Parachute		×

(b)

	Géographie	$\overline{Géographie}$
Australie	×	
Natation		×
Origami		×
Parachute		×

(c)

	Pliage	$\overline{Pliage}$
Australie		×
Natation		×
Origami	×	
Parachute	×	

(d)

	Sport	$\overline{Sport}$
Australie		×
Natation	×	
Origami		×
Parachute	×	

Plus formellement, étant donné un document d_i et $L(d_i)$ l'ensemble des labels correspondant aux catégories du document d_i, pour chaque classe c_k de $\mathcal{C}$, le but est de savoir si d_i appartient ou non à c_k, c'est-à-dire si $c_k \in L(d_i)$ ou $c_k \notin L(d_i)$. Pour la classe c_k considérée, l'ensemble $\mathcal{C}_k = \{c_k, \bar{c_k}\}$ est défini comme le nouvel ensemble des classes auxquels les documents de la collection $\mathcal{D}$ appartiennent.

Pour chaque document d_i de l'échantillon d'apprentissage $\mathcal{D}_A$, l'ensemble $L(d_i)$ des labels affectés au document d_i est modifié par $L_k(d_i)$ où $L_k(d_i)$ est égal au singleton

$\{c_k\}$ si $c_k \in L(d_i)$ et au singleton $\{\bar{c_k}\}$ sinon. Pour un document d_i de $\mathcal{D}$, le classifieur binaire préalablement appris sur $\mathcal{D}$ retourne deux scores $\phi_k(\vec{d_i}, c_k)$ et $\phi_k(\vec{d_i}, \bar{c_k})$. Le document d_i de $\mathcal{D}$ sera ensuite affecté à la classe c_k (c'est-à-dire $\hat{L}_k(d_i) = \{c_k\}$) si $\phi_k(\vec{d_i}, c_k) > \phi_k(\vec{d_i}, \bar{c_k})$:

$$\hat{L}_k(d_i) = \begin{cases} \{c_k\} & \text{si } \phi_k(\vec{d_i}, c_k) > \phi_k(\vec{d_i}, \bar{c_k}) \\ \emptyset & \text{sinon} \end{cases} \tag{2.12}$$

Une catégorisation binaire est alors réalisée pour toutes les classes c_k de $\mathcal{C}$ et l'ensemble des classes assignées à un document d_i de $\mathcal{D}_T$ correspond alors à l'union des $\hat{L}_k(d_i)$:

$$\hat{L}(d_i) = \bigcup_k \hat{L}_k(d_i) \tag{2.13}$$

2.2.1.3 Combinaison des catégories

Le principal problème de l'approche binaire est qu'elle ne considère pas les relations qui peuvent exister entre les différentes catégories qui apparaissent souvent ensemble. Ceci est particulièrement gênant dans le cadre de la catégorisation de documents où les catégories sont souvent organisées en hiérarchie. Pour en tenir compte, une autre méthode consiste à créer de nouvelles catégories correspondant à une combinaison des catégories qui apparaissent conjointement comme le montre la table 2.9.

TAB. 2.9 – Transformation du problème de la catégorisation multilabel en utilisant l'apparition conjointe des catégories.

Documents \ Catégories	Pliage	Pliage ∧ Sport	Eau ∧ Sport	Eau ∧ Géographie
Australie				×
Natation			×	
Origami	×			
Parachute		×		

Cette méthode répond correctement au problème de la catégorisation multilabel, mais dans la pratique, elle peut difficilement être mise en œuvre. Ainsi pour un problème à dix classes, le nombre de nouvelles classes créées peut atteindre 1 024 et les classes ainsi créées ont de grandes chances de n'avoir que très peu de représentants, ce qui rend l'apprentissage difficile.

2.2.1.4 Duplication des documents

La dernière approche consiste à dupliquer les documents en autant d'exemplaires qu'ils ont de catégories associées. Pour chaque duplication, une seule catégorie est conservée et n'est plus utilisée pour les duplications suivantes. La table 2.10 présente le résultat obtenu après la duplication des documents.

TAB. 2.10 – Transformation du problème de la catégorisation multilabel en dupliquant chaque document.

Documents \ Catégories	Pliage	Géographie	Eau	Sport
Australie		×		
Australie			×	
Natation			×	
Natation				×
Origami	×			
Parachute	×			
Parachute				×

La combinaison des catégories des données et la transformation binaire sont les seules transformations qui permettent de répondre parfaitement au problème de catégorisation multilabel en utilisant simplement des classifieurs binaires et multiclasses. La première n'est, dans la plupart des cas, pas réalisable et la seconde ne tient pas compte des relations entre les catégories en plus d'être coûteuse en terme de complexité. En effet, un modèle doit être appris pour chaque catégorie. La dernière transformation, consistant à dupliquer les documents, semble être la plus intéressante, mais elle ne permet pas de répondre directement au problème de catégorisation multilabel à l'aide d'un classifieur multiclasse.

2.2.2 Méthodes de sélection du nombre de catégories

Dans le contexte de la catégorisation multilabel, une ou plusieurs catégories peuvent être affectées à un document. Après avoir utilisé la méthode de dupliquation des documents présentée précédemment, nous utilisons un classifieur multiclasse pour classer un nouveau document. Cependant, ce dernier ne pourra lui affecter qu'une seule classe correspondant à la plus probable. Pour répondre au problème de catégorisation multilabel, nous utilisons un classifieur multiclasse qui pour chaque document d_i de $\mathcal{D}$ et chaque classe c_k de $\mathcal{C}$ retourne un score $\phi(\vec{d_i}, c_k)$. Quand un classifieur ϕ utilisé dans le contexte d'une catégorisation multiclasse retourne pour un document d_i, une liste de scores $\phi(\vec{d_i}, c_k), k = 1, \ldots, |\mathcal{C}|$, il est possible d'utiliser ce classifieur pour une catégorisation multilabel. Pour cela, il existe différentes stratégies qui permettent de sélectionner le nombre de catégories à affecter à d_i en fonction des scores retournés par le classifieur. Les stratégies les plus classiques sont $RCut$, $PCut$ et $SCut$. Ces stratégies ont déjà fait l'objet de plusieurs évaluations sur des collections comme Reuters-21578, OHSUMED-233445, MeSH et HEP [Yang, 2001, Ráez et López, 2006]. En plus d'être pour la plupart des petites collections, ces dernières sont composées de textes relativement courts. À notre connaissance, ces critères n'ont pas encore été évalués sur de grandes collections. Dans la suite, nous comparons ces stratégies avec notre critère sur des collections importantes possédant des documents de taille conséquente.

2.2.2.1 Stratégie RCut

La plus simple des stratégies consiste à affecter à chaque document un même nombre r de catégories et correspond à l'algorithme 2. Pour un document d_i donné, les scores $\phi(\vec{d_i}, c_k), k = 1, \ldots, |\mathcal{C}|$ sont triés et seules les r premières catégories sont affectées à d_i.

Dans le cas où r est égal à 1, cela revient à considérer le problème comme une catégorisation multiclasse. Il est possible d'utiliser cette stratégie pour classer un document quelconque de $\mathcal{D}$: elle est orientée document.

Données : $r, Tab[k] = \phi(\overrightarrow{d_i}, c_k) : k = 1 \ldots |\mathcal{C}|$
Résultat : $\hat{L}(d_i)$
$\hat{L}(d_i) = \emptyset$;
$S = sort(Tab)$;
$rcut = S[r]$;
pour $k \leftarrow 1$ **à** $|\mathcal{C}|$ **faire**
  si $Tab[k] \geq rcut$ **alors**
    $\hat{L}(d_i) = \hat{L}(d_i) \cup c_k$;
  fin
fin
retourner $\hat{L}(d_i)$

Algorithme 2: Présentation de la stratégie RCut.

Des améliorations de cette stratégie ont été proposées avec RTCut pour gérer les conflits qui peuvent se poser quand certains scores $\phi(\vec{d_i}, c_k)$ sont égaux pour différentes catégories et qu'il n'est pas possible de choisir lesquelles sont à affecter à d_i [Yang, 2001]. Cependant dans un contexte multilabel, les documents ne sont pas sensés avoir le même nombre de catégories ; *RCut* et *RTCut* ne sont donc pas vraiment adaptées au problème multilabel.

2.2.2.2 Stratégie PCut

Par rapport à la stratégie RCut orientée document, PCut est une stratégie orientée catégorie [Lewis, 1992a, Lewis et Ringuette, 1994]. Le but est d'affecter la classe c_k à un nombre n_k de documents calculé en fonction du pourcentage de documents de l'échantillon d'apprentissage $\mathcal{D}_A$ qui appartiennent effectivement à la classe c_k. Les scores $\phi(\vec{d_i}, c_k), i = 1, \ldots, |\mathcal{D}|$ pour une classe c_k donnée sont triés et les n_k premiers documents sont affectés à la classe c_k. La valeur de n_k est donnée par [Yang, 2001] :

$$n_k = P(c_k) * x * |\mathcal{C}| \tag{2.14}$$

où $P(c_k)$ est la probabilité qu'un document quelconque appartienne à la classe c_k et x est un paramètre de la stratégie PCut. Généralement ce dernier est calculé pour optimiser globalement la catégorisation sur un échantillon de validation $\mathcal{D}_V$. En faisant l'hypothèse que la distribution des catégories entre les échantillons $\mathcal{D}_A$ et $\mathcal{D}_T$ est similaire, le paramètre x peut être fixé à $\frac{|\mathcal{D}_T|}{|\mathcal{C}|}$.

Notons que cette stratégie n'est pas utilisable si le but est de classer les documents séparément puisque le classement s'effectue sur l'ensemble des documents de l'échantillon de test. De plus, les performances de cette stratégie dépendent fortement de

l'égalité des distributions des classes entre l'échantillon d'apprentissage et de test. Si ces distributions ne sont pas identiques, la méthode ne donne pas de bons résultats.

2.2.2.3 Stratégie SCut

Contrairement aux stratégies RCut et PCut qui nécessitent un seul paramètre chacun, la stratégie SCut calcule un seuil par catégorie. Pour une catégorie c_k donnée, différentes valeurs sont testées et celle qui optimise le classement, pour cette catégorie sur l'ensemble des documents d'un échantillon de validation, est utilisée comme seuil. La stratégie SCut calcule ces paramètres de façon locale et ne garantit pas de bonnes performances globales car elle présente un risque de surapprentissage [Yang, 2001].

Les différentes stratégies pour sélectionner le nombre de catégories à conserver pour un document possèdent plusieurs problèmes. La stratégie RCut souffre de son incapacité à affecter un nombre différent de catégories pour différents documents. Lorsque le nombre de catégories à associer aux documents est relativement constant, cette stratégie peut se révéler efficace, mais elle ne pourra pas produire de bons résultats si le nombre de catégories par document diffère fortement. Le principal problème de la stratégie PCut est qu'elle est orientée catégorie et qu'elle ne permet donc pas de sélectionner les catégories à associer à un seul document en particulier. Cette contrainte empêche l'utilisation de cette méthode pour beaucoup d'applications, comme la détection des messages indésirables, par exemple, puisque les messages à analyser arrivent les uns après les autres. De plus, cette stratégie nécessite le calcul du paramètre x et suppose que la proportion des catégories reste équivalente entre les échantillons d'apprentissage et de test. Enfin la catégorie SCut calcule un seuil pour chaque catégorie. Ce calcul s'effectue localement et peut poser des problèmes de surapprentissage. Pour palier les limites des méthodes classiques, nous avons proposé *MCut*, une méthode de sélection de catégories qui est orienté document.

2.2.3 Nouvelle méthode de sélection du nombre de catégories : MCut

Notre but est de proposer *MCut* (Maximum Cut), une méthode alternative aux méthodes classiques de sélection du nombre de catégories qui soit orienté document, c'est-à-dire utilisable pour un document particulier et qui ne nécessite pas l'apprentissage de paramètres sur un échantillon d'apprentissage. L'idée de la stratégie MCut est de ne conserver que les catégories qui ont obtenu des scores particulièrement élevés par rapport à l'ensemble des scores de toutes les catégories. Le principe sous-jacent à *MCut* est de calculer la plus importante différence entre deux scores successifs et d'utiliser le score intermédiaire comme seuil. Il ne suffit ensuite que de garder les catégories pour lesquelles le score est plus élevé que le seuil.

Plus formellement, pour un document d_i, l'ensemble des scores $(\phi(\overrightarrow{d_i}, c_k) : k = 1 \ldots \mathcal{C})$ pour toutes les catégories est trié par ordre décroissant. La liste triée obtenue est notée $S = (s(l), l = 1 \ldots \mathcal{C})$ avec $s(l) = \phi(\overrightarrow{d_i}, c_k)$ si $\phi(\overrightarrow{d_i}, c_k)$ est le l^e plus grand score. La plus grande différence entre deux scores successifs est ensuite calculée. Le seuil *mcut* correspond à la moyenne des deux scores successifs qui ont permis d'obtenir ce saut maximum. Nous calculons l'indice m qui permet d'obtenir la plus grande différence de scores par :

$$m|(s(m) - s(m+1)) = Max\{(s(l) - s(l+1)) : l = 1, \ldots, |\mathcal{C}| - 1\} \tag{2.15}$$

Le seuil $mcut$ est ensuite obtenue par :

$$mcut = \frac{s(m) + s(m+1)}{2} \tag{2.16}$$

Les catégories associées au document d_i correspondent aux catégories c_k dont le score $\phi(\overrightarrow{d_i}, c_k)$ est supérieur à $mcut$:

$$\hat{L}(d_i) = \{c_k \in \mathcal{C} | \phi(\overrightarrow{d_i}, c_k) > mcut\} \tag{2.17}$$

Données : $Tab[k] = \phi(\overrightarrow{d_i}, c_k) : k = 1 \ldots |\mathcal{C}|$
Résultat : $\hat{L}(d_i)$
$\hat{L}(d_i) = \emptyset$;
$S = sort(Tab)$;
$Sdiff[ind] = (S[ind] - S[ind+1]) : ind = 1 \ldots |\mathcal{C}| - 1$;
$m = index(max(Sdiff))$;
$mcut = \frac{S[m]+S[m+1]}{2}$;
pour $k \leftarrow 1$ **to** $|\mathcal{C}|$ **faire**
  si $Tab[k] > mcut$ **alors**
    $\hat{L}(d_i) = \hat{L}(d_i) \cup c_k$;
  fin
fin
retourner $\hat{L}(d_i)$

Algorithme 3: Algorithme de la stratégie $MCut$.

La stratégie $MCut$ est présentée par l'Algorithme 3 et est illustrée graphiquement par la Figure 2.2. Cette figure montre la liste triée S obtenue pour deux documents (d_1 sur la gauche et d_2 sur la droite). La stratégie $RCut_1$ (respectivement $RCut_2$) fixe le paramètre r à 1 (respectivement 2). De ce fait, pour le document d_1, une seule catégorie est associée à d_1 avec les stratégies $RCut_1$ et $MCut$, et les deux premières catégories pour $RCut_2$. L'exemple du document d_2 montre que contrairement à d_1, la stratégie $MCut$ associe trois catégories à d_2 alors que les stratégies $RCut_1$ et $RCut_2$ affecte respectivement une et deux catégories à d_2.

La stratégie $MCut$ offre l'avantage d'être une stratégie locale qui associe un nombre de catégories qui n'est pas nécessairement le même pour tous les documents de $\mathcal{D}$. De plus il n'y a pas besoin d'apprendre un paramètre ou de calculer de seuil en utilisant un échantillon d'apprentissage. Comme le calcul du nombre de catégories à affecter à un document d_i ne dépend pas des autres scores obtenus pour les documents de $\mathcal{D}$, cette stratégie est particulièrement bien adaptée aux applications qui ont besoin de traiter les documents les uns après les autres comme le filtrage de messages électroniques indésirables. Enfin la stratégie $MCut$ ne fait aucune hypothèse sur la distribution des catégories entre $\mathcal{D}_A$ et $\mathcal{D}$.

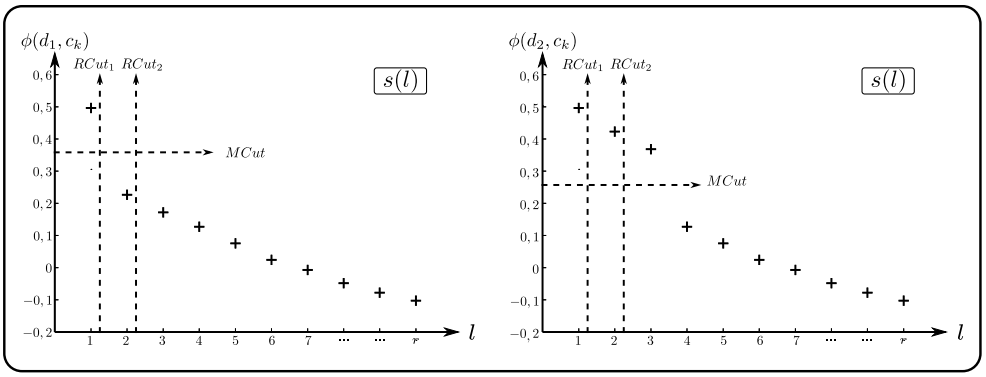


FIG. 2.2 – Illustration comparant les deux stratégies *RCut* et *Mcut*.

2.2.4 Expérimentations

La stratégie de sélection du nombre de catégories *MCut* a été évaluée en la comparant à d'autres méthodes de sélection sur deux collections : XML Mining 2009 (XML09) et Reuters Corpus Volume 1 (RCV1). Après une description de ces deux collections et du protocole expérimental utilisé, les résultats seront présentés.

2.2.4.1 Présentation des collections XML09 et RCV1

Les deux collections XML09 et RCV1 utilisées sont brièvement présentées dans la table 2.11.

TAB. 2.11 – Brève description des collections XML Mining 2009 et RCV1.

	XML09	RCV1
Nombre de documents	54 632	804 414
Longueur moyenne des documents	2 103,47	123,9
Longueur moyenne des documents (terme unique)	535,86	75,73
Taille de l'index	295 721	47 220

XML09

La première collection XML09 est extraite du corpus XML Wikipedia proposé par Denoyer et Gallinari [Denoyer et Gallinari, 2006]. Elle est utilisée dans le cadre de la compétition INEX 2009[1] [Nayak *et al.*, 2010] et présentée dans la table 2.12. Une présentation de notre participation à cette compétition se trouve en Annexe A.2 [Largeron *et al.*, 2010]. La collection XML09 est composée de 54 632 documents classés à l'aide de 39 catégories correpondant chacune à un sujet précis comme le baseball, la politique, la guerre civile américaine, la seconde guerre mondiale, etc. Chaque document

[1] `http://www.inex.otago.ac.nz/`

est affecté à au moins une catégorie. Les documents de la collection sont relativement longs avec une moyenne de 2 100 mots. L'échantillon d'apprentissage $\mathcal{D}_A$ représente environ 20% de la collection totale soit 10 978 documents. La moyenne du nombre de catégories affectées par document est de 1,45 avec 83% de documents de la collection qui ne possèdent qu'une seule catégorie.

TAB. 2.12 – Description des échantillons d'apprentissage ($\mathcal{D}_A$) et de test ($\mathcal{D}_T$) pour la collection XML Mining 2009.

	XML09	
	$\mathcal{D}_A$	$\mathcal{D}_T$
Nombre de documents	10 978	43 654
Numbre de catégories	39	39
Nombre de documents avec une seule catégorie	9 053	36 042
Moyenne du nombre de catégories par document	1,46	1,45
Variance du nombre de catégories par document	3,15	2,96

RCV1

La seconde collection RCV1 correspond à une collection très utilisée en catégorisation de texte [Lewis *et al.*, 2004]. RCV1 est une archive d'environ 800 000 articles provenant de l'agence de presse Reuters. Ce sont des documents plutôt courts avec en moyenne 120 mots par document qui ont été annotés manuellement. Ils sont repartis en trois jeux de données correspondant au sujet général de l'article *(topics)*, à la région géographique, au parti économique ou politique auquel il est associé *(regions)* et au type d'affaires dont traite l'article *(industries)*. Le premier jeu *topics* utilise un ensemble de 103 catégories pour classer les documents. 3,18 catégories de *topics* sont affectées en moyenne aux documents de la collection. Dans le deuxième jeu *regions*, les documents sont classés en utilisant 296 catégories. Les documents possèdent en moyenne un peu plus d'une catégorie de *regions*. Enfin, le dernier jeu *industries* qui utilise 350 catégories ne sera pas considéré dans la suite car plus de la moitié des documents ne sont affectés à aucune catégorie. La table 2.13 présente les deux jeux *topics* et *regions* de la collection RCV1. Les documents de l'échantillon d'apprentissage sont les mêmes dans les deux jeux et correspondent à moins de 3% des documents de la collection totale. Une particularité de cette collection est l'absence de certaines catégories dans l'échantillon d'apprentissage qui ne pourront de ce fait pas être affectées aux documents lors de la phase de test.

TAB. 2.13 – Description des échantillons d'apprentissage ($\mathcal{D}_A$) et de test ($\mathcal{D}_T$) pour la collection RCV1.

	RCV1 (*topics*)		RCV1 (*regions*)	
	$\mathcal{D}_A$	$\mathcal{D}_T$	$\mathcal{D}_A$	$\mathcal{D}_T$
Nombre de documents	23 149	781 265	23 149	781 265
Numbre de catégories	101	103	228	296
Nombre de documents avec une seule catégorie	734	23 871	18 638	616 762
Moyenne du nombre de catégories par document	3,18	3,24	1,28	1,32
Variance du nombre de catégories par document	1,84	1,98	0,53	0,65

2.2.4.2 Protocole expérimental

Les documents des différentes collections sont représentés sous forme de vecteurs de poids tf.idf avant d'effectuer la classification.

Représentation des documents de la collection XML09

La collection XML Mining 2009 est composée de documents XML extraits de l'encyclopédie Wikipedia. Pour représenter ces documents sous forme de vecteurs de poids tf.idf, nous avons dans un premier temps construit le vocabulaire puis nous avons calculé les poids $w_{i,j}$ associés pour chaque terme t_j au document d_i.

La structure XML n'étant pas considérée dans la suite, toutes les balises ont été supprimées pour ne garder que le texte plat. Sans aucun prétraitement, la taille du vocabulaire original de la collection XML Mining 2009 s'élève à 1 136 737. Dans le but de réduire la taille de ce vocabulaire, une lemmatisation à l'aide de l'algorithme de Porter a été effectuée [Porter, 1980]. Un nombre important de termes considérés comme non pertinents ont également été supprimés : les mots qui contiennent des chiffres, les mots qui font moins de trois caractères, les mots qui apparaissent moins de trois fois et les termes qui apparaissent dans tous les documents. Après réduction du vocabulaire, le nombre de termes considérés dans le vocabulaire est de 295 721.

Pour calculer le poids $w_{i,j}$ du terme t_j dans le document d_i, nous avons utilisé les formules classiques du $tf.idf$ présentées dans l'équation 2.11.

Représentation des documents de la collection RCV1

Les documents originaux de la collection RCV1 ne sont pas libres de droit. Pour utiliser cette collection, une représentation sous forme de sac de mots est proposée par Lewis avec une pondération tf.idf [Lewis *et al.*, 2004]. La construction des vecteurs de représentation des documents s'est déroulée après avoir transformé le texte en minuscule, construit le vocabulaire et pondéré les mots du vocabulaire pour chaque document

[Lewis *et al.*, 2004].

La construction du vocabulaire a été effectuée après avoir supprimé tous les signes de ponctuaction. Les mots composés de chiffres ainsi que les mots considérés vides, comme dans le projet SMART de Salton, ont été supprimés. Cette liste comporte 571 mots de la langue anglaise qui sont pour la plupart des articles et des adverbes. Une lemmatisation basée sur l'algorithme de Porter a ensuite été appliquée aux mots restants [Porter, 1980]. Le vocabulaire est finalement composé de 47 220 mots.

La pondération des termes correspond à une approche basée sur le principe du tf.idf. Le poids $w_{i,j}$ du terme t_j dans le document d_i est calculé par :

$$w_{i,j} = (1 + \ln(n_{i,j})) \times \ln \frac{|D_A|}{|\{d_i : t_j \in d_i\}|} \tag{2.18}$$

où $|D_A|$ correspond au nombre de documents utilisés pour calculer la partie idf du poids car Lewis ne s'est servi que de l'échantillon d'apprentissage pour calculer la partie idf du poids $w_{i,j}$.

Catégorisation multilabel

Pour répondre au problème de catégorisation multilabel, plusieurs méthodes ont été appliquées aux deux collections XML09 et RCV1. La première a consisté à utiliser l'approche binaire qui vise à apprendre un classifieur par catégorie. Ensuite, nous avons considéré les autres stratégies de sélection du nombre de catégories à affecter aux document, à savoir *RCut* et *PCut* ainsi notre stratégie *MCut*. La stratégie *RCut* a été utilisée en faisant varier le paramètre r de 1 et 4. Pour *PCut* nous avons posé l'hypothèse que les distributions des catégories étaient constantes entre les échantillons $\mathcal{D}_A$ et $\mathcal{D}_T$ et avons fixé le paramètre x à $\frac{|\mathcal{D}_T|}{|\mathcal{C}|}$. Pour chaque collection, la catégorisation a ensuite été réalisée en utilisant la représentation sous forme de vecteurs de poids des documents. L'algorithme de classification utilisé est basé sur des machines à vecteurs de support (SVM). Le logiciel utilisé pour la classification est liblinear [Fan *et al.*, 2008][1]. Il permet de classer les documents en utilisant un noyau linéaire. Pour chaque méthode, nous avons ensuite utilisé trois critères classiques, C_{Exact}, C_{Micro} et C_{Macro} décrits dans la partie 1.1.3.3, pour évaluer le résultat du classement.

2.2.4.3 Résultats

Après avoir présenté les résultats obtenus pour la collection XML09 et pour la collection RCV1, nous ferons un bilan général concernant l'efficacité des différentes stratégies de sélection du nombre de catégories.

XML09

La table 2.14 montre l'ensemble des résultats obtenus pour la collection XML09. Comme nous pouvons le voir, la méthode binaire obtient les plus mauvais résultats par rapport aux stratégies *MCut*, $RCut_1$ et *PCut*. Ces résultats peuvent se justifier par la nature des documents catégorisés. En effet, les catégories utilisées pour classer les documents extraits de Wikipedia peuvent se superposer, comme les catégories, guerre, première guerre mondiale, guerre civile américaine. L'hypothèse d'indépendance des catégories n'étant pas vérifiée, les mauvais résultats obtenus par la méthode binaire

[1] `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

TAB. 2.14 – Résultats pour la collection XML09.

	XML Mining 2009		
	Exact	Micro	Macro
Binaire	0,330	0,402	0,324
MCut	0,524	0,600	0,560
$RCut_1$	0,594	0,597	0,536
$RCut_2$	0,037	0,535	0,515
$RCut_3$	0,006	0,460	0,454
$RCut_4$	0,002	0,401	0,398
PCut	0,438	0,544	0,513

ne sont pas surprenants. La stratégie $RCut_1$ qui consiste à ne retenir que la première catégorie, comme s'il s'agissait d'un problème de catégorisation multiclasse, obtient le meilleur résultat pour le critère C_{Exact} avec une valeur de 0,594. Ce critère permet de mettre en avant les stratégies qui affectent uniquement les bonnes catégories aux documents. Comme le montre la table 2.12, 82% des documents de la collection ne possèdent qu'une seule catégorie. De ce fait, la stratégie $RCut_1$ est favorisée. La stratégie $MCut$ obtient un résultat légèrement plus faible de 0,524 pour le critère C_{Exact}. En revanche, elle permet d'améliorer les deux autres critères de 0,597 à 0,600 (respectivement de 0,536 à 0,560) pour la moyenne micro C_{Micro} (respectivement macro C_{Macro}) de la F-mesure par rapport à la stratégie $RCut_1$. Quant à la stratégie $PCut$, les résultats sont meilleurs par rapport à l'approche binaire, mais restent moins bons que les stratégies $RCut_1$ et $MCut$. Sans surprise, les résultats des stratégies $RCut_2$, $RCut_3$ et $RCut_4$ ne sont pas bons car la table 2.14 indique que le nombre moyen de catégories par document dans l'échantillon d'apprentissage est légèrement inférieur à 1,5.

RCV1 (*topics*)

TAB. 2.15 – Résultats pour la collection RCV1 (*topics*).

	RCV1 (*topics*)		
	Exact	Micro	Macro
Binary	0,509	0,799	0,494
MCut	0,443	0,627	0,306
$RCut_1$	0,030	0,444	0,179
$RCut_2$	0,193	0,671	0,378
$RCut_3$	0,325	0,750	0,482
$RCut_4$	0,041	0,710	0,491
PCut	0,378	0,655	0,355

Les résultats obtenus pour le jeu *topics* de la collection RCV1 sont présentés dans la table 2.15. Contrairement aux résultats obtenus avec la collection XML09, la méthode binaire obtient les meilleurs résultats quel que soit le critère considéré. Par rapport à

ces résultats, les stratégies de sélection du nombre de catégories obtiennent globalement de moins bons résultats. Le nombre moyen de catégories par document d'après la table 2.13 étant de 3,2, la stratégie $RCut_3$ obtient globalement les meilleurs résultats par rapport aux stratégies $RCut_1$, $RCut_2$ et $RCut_4$, à l'exception du critère C_{Macro} qui est légèrement meilleur pour $RCut_4$. Si $PCut$ obtient de meilleurs résultats pour le critère C_{Exact} avec 0,378 par rapport au 0,325 de $RCut_3$, C_{Micro} et C_{Macro} sont moins bons. La stratégie $MCut$ obtient également de moins bons résultats pour les critère C_{Micro} et C_{Macro}, mais elle permet de retrouver plus souvent que les autres stratégies les bonnes catégories avec un critère C_{Exact} de 0,443.

RCV1 (*regions*)

TAB. 2.16 – Résultats pour la collection RCV1 (*regions*).

	RCV1 (*regions*)		
	Exact	Micro	Macro
Binary	0,746	0,835	0,407
MCut	0,765	0,670	0,289
$RCut_1$	0,756	0,823	0,433
$RCut_2$	0,101	0,683	0,490
$RCut_3$	0,017	0,554	0,464
$RCut_4$	0,003	0,461	0,423
PCut	0,722	0,774	0,363

La table 2.16 présente les résultats obtenus pour le jeu *regions* de la collection RCV1. Contrairement aux deux dernières expérimentations, aucune approche ne se démarque véritablement. Parmi les différentes stratégies $RCut$, $RCut_1$ obtient globalement les meilleurs résultats. Ceci n'est pas étonnant étant donnée la table 2.13 qui montre qu'environ 1,3 catégories sont affectées en moyenne aux document. La stratégie $RCut_2$ est la meilleure approche si nous considérons le critère C_{Macro}, alors qu'il s'agit de l'approche binaire pour le critère C_{Micro}. Enfin la stratégie $MCut$ obtient les meilleurs résultats pour le critère C_{Exact}. La stratégie $PCut$ obtient une nouvelle fois des résultats corrects.

Conclusion sur les différentes stratégies

Si dans certains cas l'approche binaire permet d'obtenir de très bons résultats, l'utilisation de stratégies de sélection du nombre de catégories s'avèrent également très efficace. La stratégie $RCut$ est l'une des plus simples à mettre en œuvre et donne de très bons résultats quand le nombre de catégories à affecter aux documents est globalement le même pour tous les documents. Dans ce cas, le paramètre peut se calculer facilement. Le principal problème de la stratégie $PCut$ est l'impossibilité de traiter les documents séparément. Elle obtient de bons résultats quand les distributions des catégories entre l'échantillon d'apprentissage et le reste de la collection sont semblables. Comme le montre la table 2.17, la différence entre les distributions des catégories n'étaient pas importante dans nos jeux de données, ce qui était en faveur de cette stratégie. Finalement la stratégie $MCut$ est une bonne alternative aux autres stratégies classiques. En

plus d'être simple à mettre en œuvre, elle offre l'avantage, contrairement aux autres méthodes, de ne pas nécessiter l'estimation d'un paramètre. Elle n'affecte pas nécessairement le même nombre de catégorie aux différents documents et peut s'utiliser pour les documents pris séparément. Cette stratégie dépend du pouvoir discriminant du classifieur : plus ce dernier peut apprendre et différentier les classes et meilleurs seront les résultats.

TAB. 2.17 – Moyenne du pourcentage de la différence de la proportion des catégories entre l'échantillon d'apprentissage et celui de test.

Collection	Moyenne
XML09	0,157
RCV1 (*topics*)	0,190
RCV1 (*regions*)	0,698

Dans ce chapitre sur la représentation de l'information textuelle, les objectifs étaient doubles :
- réduire la taille du vocabulaire dans une tâche de catégorisation multiclasse ;
- sélectionner le nombre de catégories à associer à un document dans le contexte de la catégorisation multilabel.

Deux critères $CCDE$ et $MCut$ ont été proposés et confrontés à des critères classiques.

Dans un premier temps, nous avons montré qu'il était possible de réduire la taille du vocabulaire de plus de 90% en ne dégradant les résultats que de 1% grâce au critère $CCDE$ dans le contexte d'une catégorisation multiclasse sur la collection XML Mining 2008 [Largeron *et al.*, 2011, Largeron et Moulin, 2010]. L'étude réalisée a permis de confirmer l'importance de la réduction du vocabulaire dans ce contexte et corrobore les études précédentes menées sur des collections de documents de texte court comme Reuters-21578.

Dans le contexte de la catégorisation multilabel, notre but a été de proposer un critère permettant de sélectionner le nombre de catégories à affecter à un nouveau document sans avoir besoin de fixer préablement des paramètres. Dans le cadre de notre participation à XML Mining 2009, notre critère $MCut$ nous a permis d'améliorer les résultats obtenus à partir des sélections classiques $RCut$, $PCut$ ou par rapport à l'approche binaire et d'obtenir en moyenne les meilleurs résultats [Géry *et al.*, 2009]. Nous avons ensuite utilisé ce critère sur une autre collection $RCV1$ classiquement employée dans le contexte de la catégorisation multilabel de documents textuels et montré qu'il permettait d'obtenir de très bons résultats.

Notre objectif étant de proposer une solution permettant de combiner les informations textuelle et visuelle contenues dans les documents multimédias, nous allons maintenant nous intéresser à l'adaptation du modèle sac de mots aux images ; notamment, nous nous interrogerons sur les approches de pondérations à utiliser pour les mots visuels et la combinaison de différents vocabulaires visuels.

Chapitre 3

Représentation des images par sacs de mots visuels pondérés tf.idf

Dans le chapitre précédent, nous nous sommes intéressés à la représentation de l'information textuelle sous forme de sacs de mots. Le succès de cette approche a largement inspiré les récents travaux en indexation, recherche et classification d'objets ou d'images. Les bonnes performances de la représentation des images en sacs de mots visuels et l'analogie avec la représentation du texte ont guidé notre choix vers cette méthode de représentation de l'information visuelle des documents multimédias.

Dans ce chapitre, nous introduisons la méthodologie adoptée dans notre travail pour représenter les images en sacs de mots visuels dans un contexte de catégorisation. Nous détaillons les différentes étapes permettant la création d'un vocabulaire visuel en justifiant les choix des techniques et des paramètres utilisés pour chaque étape. Parmi eux, nous pouvons citer le choix de la méthode de sélection des points d'intérêts, du nombre de points sélectionnés, des descriptions de ces points, de la méthode de quantification et du nombre de mots visuels à créer.

Ensuite, nous proposons deux pistes d'amélioration : la pondération $tf.idf$ des mots visuels et la combinaison de différents vocabulaires visuels. L'utilisation des mots visuels pour représenter les images en sacs de mots nécessite de pondérer ces derniers pour prendre en compte leurs représentativité et discriminance. Ainsi, nous étudions l'influence de la pondération des mots visuels dans les images en comparant plusieurs pondérations classiquement inspirées des approches utilisées pour représenter les documents textuels. Comme pour les documents textuels, les mots visuels utilisés pour représenter les images peuvent être pondérés de tel sorte que ceux qui apparaissent souvent dans une image et peu dans le reste des images de la collection soient favorisés. Différentes descriptions peuvent être utilisées pour générer des vocabulaires visuels. Nous considérons la fusion des mots visuels issus de plusieurs vocabulaires pour améliorer les résultats de la catégorisation et nous la comparons aux approches classiques de fusions précoce et tardive.

3.1 Présentation des différents paramètres

Dans cette partie, nous décrivons les méthodes que nous utilisons pour créer différents vocabulaires ainsi que les différentes pondérations permettant de pondérer les mots visuels dans les images. Pour une image d_i donnée, nous noterons respectivement la largeur et la hauteur de cette image par m et n.

3.1.1 Création d'un vocabulaire visuel

Nous décrirons les différentes étapes de la construction d'un vocabulaire visuel : la détection des points d'intérêt, la description de ces derniers et la quantification de ces descriptions.

3.1.1.1 Détection des points d'intérêt

Comme nous l'avons présenté dans la partie 1.3.1.2, le but de cette étape est de sélectionner les points au voisinage desquels nous calculons un descripteur. Elle peut s'effectuer par une détection préalable de points d'intérêts à l'aide d'algorithmes spécifiques [Harris et Stephens, 1988, Lowe, 1999]. Plusieurs études montrent qu'un simple échantillonnage régulier des points ou un échantillonnage multi-échelle sont très efficaces et conduisent à de meilleurs résultats dans le contexte de la catégorisation d'images [Jurie et Triggs, 2005, Nowak *et al.*, 2006]. Dans la suite, nos choix se sont portés sur l'échantillonnage dense et multi-échelle.

Échantillonnage dense

L'échantillonnage dense, ou la détection régulière, consiste à échantillonner les points régulièrement et à définir une région rectangulaire de taille fixe autour des points. Nous devons donc choisir la période d'échantillonnage et la taille des régions. Dans nos expérimentations, nous avons choisi des régions de même taille que la période d'échantillonnage, de telle sorte qu'il n'y ait pas de recouvrement entre les imagettes ainsi créées. Chaque image est découpée en imagettes rectangulaires proportionnelles à $m \times n$, comme illustré par la figure 3.1.

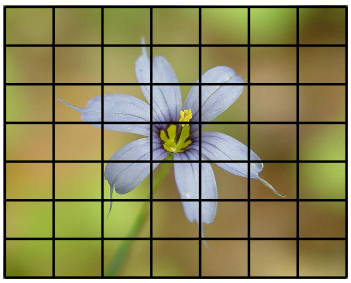


FIG. 3.1 – Détection régulière des images.

La détection régulière ne dépend alors que d'un seul paramètre a permettant de faire varier le nombre d'imagettes obtenues par image. Pour une image donnée, a^2 correspond alors au nombre d'imagettes détectées dans l'image. Chaque imagette possède donc une largeur de $\frac{m}{a}$ pixels et une hauteur de $\frac{n}{a}$ pixels. La grille utilisée pour ce découpage offre l'avantage d'être simple à réaliser et permet de recouvrir l'image dans sa globalité. Elle

permet également d'être invariante à la taille de l'image, mais ne l'est pas en revanche pour les translations et autres changements de point de vue.

Détection multi-échelle

Contrairement à la détection régulière, l'approche multi-échelle considère des régions de différentes tailles échantillonnées à différentes périodes [Nowak *et al.*, 2006]. La taille et la période d'échantillonnage de ces régions dépend de l'échelle considérée. On commence par fixer la taille de la région et la période d'échantillonnage à la plus petite échelle. Pour les autres échelles, ces deux paramètres sont multipliés par le coefficient de progression des échelles afin de conserver un taux de recouvrement constant entre les régions pour les différentes échelles. Le plus courant est d'adopter une progression des échelles en puissance de 2 et de multiplier également la taille des régions et leur période d'échantillonnage par 2. Dans nos expérimentations, nous avons choisi une progression en facteur 2 et à l'échelle 1, des régions de 12×12 pixels échantillonnées tous les pixels. La taille des régions à l'échelle 1 est un paramètre qui a été fixé pour pouvoir calculer la description $sift$. L'échelle maximale dépend de la taille de l'image et correspond à l'imagette dont le côté ne dépasse ni m, ni n. Ainsi, le nombre de régions générées est de plus en plus faible quand le niveau d'échelle augmente. Pour une image donnée, le nombre de régions générées, en considérant toutes les échelles possibles, est très important. Nous avons choisi de sélectionner un nombre limité de régions choisies aléatoirement parmi toutes les régions possibles, favorisant ainsi les régions issues des petites échelles. La taille de la région à l'échelle 1 étant fixe, cette détection ne nécessite également qu'un seul paramètre correspondant au nombre de régions choisies aléatoirement dans l'image.

3.1.1.2 Description

Après la détection des points et de leur voisinage dans une image donnée, différentes descriptions peuvent être utilisées pour les représenter. Il existe un très grand nombre de descripteurs exploitant différentes informations, comme la couleur [Boughorbel *et al.*, 2002, Schettini *et al.*, 2001, Swain et Ballard, 1991, van de Sande *et al.*, 2008], la forme [Zhang et Lu, 2004, Ferrari *et al.*, 2008], la texture [Manjunath *et al.*, 2002, Lowe, 2004] ou plusieurs d'entres elles, comme le descripteur MPEG-7 [Salembier et Smith, 2002, Spyrou *et al.*, 2005]. Nous nous sommes focalisés sur deux descriptions complémentaires, la première basée principalement sur la couleur (*mstd*) et la seconde sur la texture et la forme (*sift*).

mstd

La description *mstd* est composée de six dimensions correspondant à la moyenne et à l'écart-type de la luminance, du rouge et du vert normalisés calculés sur l'ensemble des pixels de la région. Pour un pixel donné, la luminance, le rouge et le vert normalisés s'obtiennent respectivement par :

- $\frac{R+G+B}{3*255}$
- $\frac{R}{R+G+B}$
- $\frac{V}{R+G+B}$

où R, G et B correspondent respectivement aux trois composantes rouge, vert et bleu du pixel. Cette description a l'avantage d'être très compacte et simple à calculer. Elle

met en avant l'information couleur d'une imagette considérée.

sift

L'un des descripteurs les plus connus est *sift* utilisé dans différentes applications comme la recherche et le suivi d'objets [Lowe, 2004, Zhou *et al.*, 2008], l'alignement d'images [Szeliski, 2006], la reconnaissance de visages [Bicego *et al.*, 2006], etc. Ce descripteur est basé sur le calcul d'histogrammes d'orientation du gradient. La région d'intérêt est découpée en une grille 4×4 pour capturer l'information sur la position et un histogramme est calculé par cellule. L'orientation du gradient est quant à elle quantifiée en 8 directions. Comme nous l'avons vu dans la partie 1.3.2.2, le descripteur *sift* possède 128 dimensions.

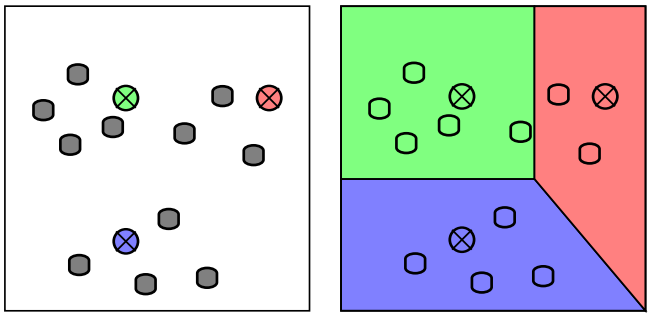
3.1.1.3 Quantification

La quantification consiste à classer les vecteurs descripteurs afin de créer un vocabulaire de mots visuels. Le nombre de vecteurs en entrée peut être très important et l'algorithme de classification doit de ce fait être très efficace. Comme nous l'avons vu précédemment, la plus utilisée des approches utilise l'algorithme des nuées dynamique ou k-means [MacQueen, 1967, Diday, 1971].

L'algorithme des k-means a pour but de trouver un nombre limité de centres, égal au plus à k, qui minimisent la distance de chaque vecteur descripteur à son centre représentatif le plus proche. Cet algorithme est illustré par la figure 3.2. Dans sa version la plus simple, les k centres sont choisis initialement aléatoirement (figure 3.2(a)) avec $k = 3$. À chaque itération, les descriptions les plus proches de ces centres sont assignées aux centres (figure 3.2(b)) et les centres sont mis à jour en calculant le nouveau barycentre des descriptions associées (figure 3.2(c)). L'algorithme se termine lorsque les centres se stabilisent (figure 3.2(d)).

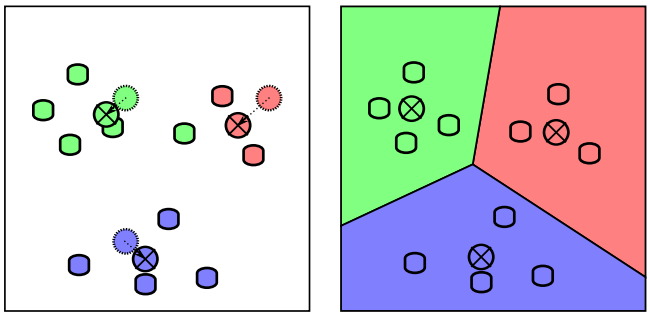
Cet algorithme possède plusieurs inconvénients [Jurie et Triggs, 2005] : il nécessite de connaître le nombre de classes ; il n'est pas robuste car son résultat dépend du choix initial des centres ; il favorise les régions denses où les vecteurs sont concentrés ; de plus, certaines classes peuvent ne pas être identifiées. Une solution pour résoudre les problèmes de l'initialisation des centres et de la favorisation des régions denses consiste à employer les méthodes agglomératives qui rassemblent en premier les descriptions les plus proches [Agarwal *et al.*, 2004, Leibe et Schiele, 2006]. Cependant, ces méthodes ne sont pas adaptées lorsque le nombre de descriptions est important du fait de leur complexité algorithmique forte. Pour résoudre ce problème de complexité, une autre solution consiste à construire les centres incrémentalement [Jurie et Triggs, 2005] en remplaçant chaque description par un représentant médian (*mean-shift*) [Comaniciu et Meer, 2002] situé dans un certain rayon. Le nombre de centres n'est donc plus à déterminer : il est directement lié à la valeur du rayon.

Dans la suite, nous utiliserons l'algorithme k-means qui reste le plus utilisé et le plus simple pour la création de vocabulaires visuels [Leung et Malik, 2001, Sivic et Zisserman, 2003, Nister et Stewenius, 2006, Philbin *et al.*, 2007, Tirilly *et al.*, 2008]. Le résultat obtenu par l'algorithme k-means est le vocabulaire de mots visuels $V = \{v_1, \ldots, v_j, \ldots, v_{|V|}\}$ où chaque v_j correspond à un des k centres. Dans ce cas, le nombre de mots visuels $|V|$ correspond au paramètre k de l'algorithme.



(a) Initialisation des k centres pour $k = 3$.

(b) Affectation des éléments au centre le plus proche.



(c) Calcul des nouveaux centres correspondant aux barycentres des éléments.

(d) Les étapes 3.2(b) et 3.2(c) sont répétées jusqu'à la convergence des centres.

FIG. 3.2 – Algorithme du k-means.

3.1.2 Pondération

Comme pour la représentation d'un texte, les mots visuels sont pondérés pour chaque image. Pour un mot visuel v_j et une image d_i, le poids $w_{i,j}$ est calculé de telle sorte que ce poids est d'autant plus élevé que le mot visuel v_j est représentatif et discriminant pour l'image d_i. La pondération peut être calculée à partir de la fréquence du terme v_j dans le document d_i, de la fréquence du terme v_j dans la collection d'images $\mathcal{D}$, ou à partir de ces deux informations. Dans sa version la plus simple, la pondération est effectuée en considérant le nombre d'apparitions $n_{i,j}$ du terme v_j dans l'image d_i. Cette pondération sera notée dans la suite tf_{raw} avec $tf_{raw}(v_j, d_i) = n_{i,j}$.

3.2 Modèle adapté à la catégorisation d'images

Nous venons de préciser nos choix permettant de construire un modèle classique de sacs de mots visuels. Nous allons maintenant étudier l'influence de différents paramètres dans un contexte de catégorisation, afin de définir les paramétrages qui serviront de référence par la suite pour la collection SIMPLIcity considérée.

3.2.1 Présentation de la collection

La collection SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries) est une base d'environ 200 000 images extraites de la collection COREL [Wang

FIG. 3.3 – Exemples extraits de la collection SIMPLIcity.

et al., 2000]. Wang propose en téléchargement un sous-ensemble de 1 000 images réparties dans un ensemble de dix catégories [1] : bâtiments, bus, chevaux, dinosaures, éléphants, fleurs, montagnes, nourriture, peuple africain et plages. Chaque catégorie est composée de 100 images. Les images ont toute une taille de 384×256 pixels. Cette collection a fait l'objet de plusieurs études pour évaluer différentes méthodes de classification d'images. Les résultats sont compris entre 70 et 86% d'images bien classées [Ros *et al.*, 2006, Mouret *et al.*, 2009].

3.2.2 Modèle et protocole expérimental

Le but de cette expérimentation est d'évaluer l'influence du nombre de mots choisi pour décrire les images, de comparer deux descripteurs *mstd* et $sift$ et d'étudier l'influence de la taille du vocabulaire. Dans cette première expérimentation, une détection multi-échelle est utilisée. Nous nous sommes servis de l'algorithme fourni par Vedaldi [Vedaldi et Fulkerson, 2010][2] pour calculer le descripteur *sift*. La quantification de ces descriptions est réalisée grâce à l'algorithme k-means utilisé dans sa version parallélisée [Bisgin, 2007][3]. Nous avons étudié deux tailles de vocabulaire différentes, à savoir 1000 et 5000 mots visuels correspondant au paramètre k de l'algorithme. La pondération utilisée pour pondérer les mots visuels dans les images correspond à tf_{raw}. Enfin, l'algorithme de classification utilisé est basé sur des machines à vecteurs de support à noyaux linéaires (SVM) réalisées grâce au programme libsvm [Chang et Lin, 2001][4] en effectuant une validation croisée d'ordre dix sur des données normalisées.

3.2.3 Résultats

Les résultats obtenus sont présentés par la figure 3.4 qui montre le pourcentage d'images bien classées pour un nombre de mots visuels par image variant de 500 à 5000. Cette figure met en évidence l'influence du descripteur et de la taille du vocabulaire.

Influence du descripteur

Comme le montre la figure 3.4, les descripteurs utilisés, *mstd* et *sift*, ont des comportements très différents en fonction du nombre de mots visuels utilisé pour décrire les images. Quelle que soit la taille du vocabulaire, 1000 ou 5000, le descripteur *mstd* offre

[1] `http://wang14.ist.psu.edu/cgi-bin/zwang/regionsearch_show.cgi`
[2] `http://www.vlfeat.org/~vedaldi/code/siftpp.html`
[3] `http://users.eecs.northwestern.edu/~wkliao/Kmeans/`
[4] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

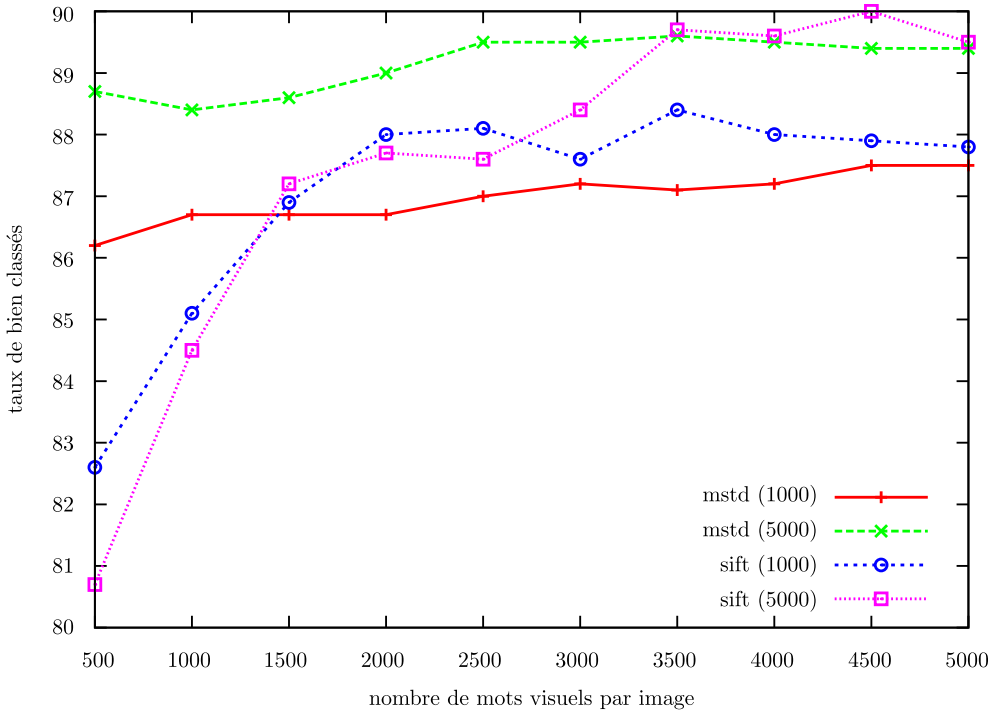


FIG. 3.4 – Taux de bien classés pour un nombre de mots visuels par image, compris entre 500 et 5000, normalisés par tf_{raw} en utilisant différents descripteurs (*mstd* et *sift*) et tailles de vocabulaire (1000 et 5000).

des résultats relativement constants autour de 87% de bien classés pour le vocabulaire de 1000 mots visuels représenté par la courbe rouge (+) et 89% pour celui de 5000 avec la courbe verte (×). Les résultats obtenus avec le descripteur *sift* augmentent avec le nombre de mots visuels utilisés pour représenter les images. Ces derniers se stabilisent à partir d'environ 3000 mots visuels par image pour atteindre un taux de bien classés d'environ 88% pour un vocabulaire de 1000 mots visuels et d'environ 89,5% pour 5000 mots représentés respectivement par la courbe bleu (∘) et la courbe rose (□). Pour un nombre limité de mots visuels, le descripteur *mstd* est plus efficace que *sift*, ce qui n'est plus toujours vérifié quand ce nombre augmente.

Importance de la taille du vocabulaire

Les résultats illustrés par la figure 3.4 montrent que globalement les vocabulaires de 5000 mots sont plus efficaces que ceux de 1000 mots que ce soit pour le descripteur *mstd* ou pour *sift*. Pour le descripteur *mstd*, l'amélioration est d'environ 2%. Pour le descripteur *sift* et un nombre de mots par image inférieur à 3000, la taille du vocabulaire n'influe pas significativement sur les résultats. En revanche, pour un nombre de mots par image supérieur à 3000, le vocabulaire de 5000 mots visuels est plus efficace que celui de 1000 avec un gain d'environ 2%.

Le modèle de référence évalué sur la collection SIMPLIcity permet d'obtenir de très bons résultats avec environ 89% de taux de bien classés par rapport à ceux des études précédentes qui étaient autour de 70% et 86%. Le comportement des descripteurs *mstd* et $sift$ diffèrent en fonction du nombre de mots visuels par image avec une relative

stabilité constatée pour *mstd* et une augmentation des performances pour $sift$. Quel que soit le descripteur considéré, les vocabulaires de 5000 mots obtiennent toujours de meilleurs résultats que ceux de 1000 mots. Dans ces expérimentations, nous avons choisi une pondération simple qui ne considère que le nombre d'apparitions des mots dans les images. Les résultats de référence vont maintenant servir à l'étude de cette pondération.

3.3 Pondération $tf.idf$ pour les images

Dans le domaine du texte, c'est une représentation par sac de mots avec une pondération $tf.idf$ qui conduit souvent aux meilleurs performances. La pondération $tf.idf$ Okapi sert généralement de référence.

Dans le domaine image, si l'influence de nombreux facteurs ont été étudiés pour chaque étape de construction du vocabulaire, nous avons constaté que le thème de la pondération des mots visuels était moins fouillé [Yang *et al.*, 2007, Tirilly *et al.*, 2009]. Nous proposons ici une étude sur la pondération des mots visuels.

Dans cette partie, nous nous intéresserons à la pondération des mots visuels pour représenter les images. Tout d'abord, nous présenterons différentes pondérations inspirées du modèle $tf.idf$ utilisé pour les documents textuels, puis nous les étudierons dans un contexte de catégorisation d'images.

3.3.1 Pondérations

Un certain nombre d'études exploitent des pondérations inspirées principalement des approches utilisées pour les documents textuels dans des contextes de recherche d'images [Nguyen *et al.*, 2009, Tirilly, 2010]. Ces pondérations sont basées sur la fréquence des termes qui apparaissent dans les images, la fréquence documentaire ou une combinaison des deux [Wookey et Geller, 2004, Lan *et al.*, 2005].

3.3.1.1 Pondération basée sur la fréquence d'un terme

La fréquence d'apparition du terme est l'une des représentations les plus classiques utilisées pour les documents textuels. Elle consiste à diviser le nombre d'apparitions du mot v_j dans le document d_i par la longueur du document. Cela revient donc à diviser tf_{raw} par une constante. Dans la suite, cette pondération ne sera donc pas considérée seule :

$$tf(v_j, d_i) = \frac{n_{i,j}}{\sum_j n_{i,j}} \tag{3.1}$$

Au lieu de diviser par la taille du document qui est constante pour les images, une autre approche consiste à diviser le nombre d'apparitions du mot visuel v_j par le nombre d'apparitions du mot le plus fréquent dans l'image d_i. Cette pondération sera notée tf_{max} :

$$tf_{max}(v_j, d_i) = \frac{n_{i,j}}{max_j(n_{i,j})} \tag{3.2}$$

La pondération tf_{ln} donne moins d'importance aux mots qui apparaissent fréquemment dans l'image en effectuant une transformation logarithmique du nombre d'apparitions du mot v_j dans l'image d_i :

$$tf_{ln}(v_j, d_i) = \ln(1 + n_{i,j}) \tag{3.3}$$

Enfin la pondération tf_{okapi} implémentée dans *lemur* correspond à une version légèrement modifiée du tf défini précédemment dans la partie 1.2.3.2

$$tf_{okapi}(v_j, d_i) = \frac{k_1.n_{i,j}}{n_{i,j} + k_1(1 - b + b\frac{|d_i|}{d_{avg}})} \tag{3.4}$$

Elle permet de donner plus ou moins d'importance, grâce aux paramètres b et k_1, à la longueur moyenne des documents de la collection et à la fréquence du terme qui apparaît dans l'image. Nous avons choisi de sélectionner un nombre égal de mots visuels par image ce qui correspond $|d_i|$ égal à d_{avg}. La pondération tf_{okapi} peut alors se simplifier par

$$tf_{okapi}(v_j, d_i) = \frac{n_{i,j}}{n_{i,j} + 1} \tag{3.5}$$

avec k_1 fixé à 1.

3.3.1.2 Pondération basée sur la fréquence documentaire

La fréquence inverse du document est calculée dans sa version la plus simple pour les images de la même façon que pour les documents textuels :

$$idf(v_j) = \ln(\frac{|\mathcal{D}|}{|\mathcal{D}_j|}) \tag{3.6}$$

où $\mathcal{D}_j$ correspond à l'ensemble des images qui contiennent au moins une fois le mots v_j.

Deux autres approches basées sur les pondérations w_2 et w_4, présentées précédemment dans la partie 1.2.3.2, peuvent aussi être utilisées :

$$idf_{w_2}(v_j) = \ln(\frac{|\mathcal{D}| + 1}{|D_j| + 0,5}) \tag{3.7}$$

$$idf_{w_4}(v_j) = \ln(\frac{|\mathcal{D}| - |D_j| + 0,5}{|D_j| + 0,5}) \tag{3.8}$$

3.3.1.3 Pondération $tf.idf$

La pondération $tf.idf$ combine les pondérations basées sur la fréquence des mots dans les images et la fréquence documentaire comme présenté précédemment dans la partie 1.2.3 pour les documents textuels. Différentes combinaisons peuvent être utilisées. La pondération classique $tf.idf$ combine les tf et idf classiques. Les deux autres pondérations $tfidf_{ln}$ et $tfidf_{w_2}$ combinent respectivement tf_{ln} avec idf et tf_{okapi} avec idf_{w_2} :

$$tfidf(v_j, d_i) = tf(v_j, d_i)idf(v_j) \tag{3.9}$$

$$tfidf_{ln}(v_j, d_i) = tf_{ln}(v_j, d_i)idf(v_j) \tag{3.10}$$

$$tfidf_{w_2}(v_j, d_i) = tf_{okapi}(v_j, d_i)idf_{w_2}(v_j) \tag{3.11}$$

3.3.1.4 Pondérations binaires

D'autres approches ne considèrent qu'une pondération binaire des mots apparaissant dans les images : d'une part celle qui prend en compte l'apparition ou la non apparition des mots ; d'autre part celle qui ne considère que les mots qui apparaissent suffisamment dans les images [Nowak *et al.*, 2006].

Pondération binaire simple

La pondération binaire simple ne prend en compte que la présence ou l'absence des mots, comme le modèle booléen présenté dans la partie 1.2.2.1 :

$$bin(v_j, d_i) = \begin{cases} 1 & \text{si } n_{i,j} > 0 \\ 0 & \text{sinon} \end{cases} \tag{3.12}$$

Pondération binaire automatique

La pondération binaire automatique compare le nombre d'apparitions des mots avec un seuil qui maximise un critère, par exemple l'information mutuelle moyenne entre le mot considéré et les classes, calculé grâce à un échantillon d'apprentissage :

$$bin_{MI}(v_j, d_i) = \begin{cases} 1 & \text{si } n_{i,j} > seuil_j \\ 0 & \text{sinon} \end{cases} \tag{3.13}$$

où $seuil_j$ peut être égal à $MI(v_j, c_k)$, le seuil maximisant l'information apportée par la présence du mot v_j en utilisant son information mutuelle moyenne calculée pour le mot v_j et une classe c_k par :

$$MI(v_j, c_k) = P(v_j, c_k) \ln \frac{P(v_j, c_k)}{P(v_j)P(c_k)} + P(\bar{v_j}, c_k) \ln \frac{P(\bar{v_j}, c_k)}{P(\bar{v_j})P(c_k)} \tag{3.14}$$

avec les différentes probabilités calculées comme dans le chapitre 2.

L'ensemble des pondérations utilisées dans la suite sont résumées dans la table 3.1.

3.3.2 Expérimentation

Le but de cette étude est d'évaluer l'influence de différentes pondérations utilisées pour la pondération des mots visuels dans le modèle de sacs de mots appliqué à la catégorisation d'images. Nous comparerons les résultats obtenus avec les pondérations inspirées des documents textuels avec les approches proposées classiquement [Nowak *et al.*, 2006].

3.3.2.1 Protocole expérimental

Le protocole expérimental pour l'étude des différentes pondérations est le même que celui utilisé pour obtenir les résultats de référence. Comme nous l'avons vu précédemment, les résultats obtenus sont globalement meilleurs pour le vocabulaire de 5000 mots visuels. Dans la suite, nous n'étudierons donc que les différentes pondérations pour des vocabulaires de 5000 mots visuels calculés avec les descripteurs *mstd* et *sift* en faisant varier le nombre de mots par image entre 500 et 5000 mots. Pour terminer ces expérimentations, nous comparerons les résultats obtenus par les pondérations inspirées des approches utilisées pour les documents textuels avec les pondérations binaires [Nowak *et al.*, 2006].

TAB. 3.1 – Différentes pondérations pour représenter un mot v_j dans un document d_i.

tf	
$tf_{raw}(v_j, d_i)$	$n_{i,j}$
$tf(v_j, d_i)$	$\frac{n_{i,j}}{\sum_j n_{i,j}}$
$tf_{max}(v_j, d_i)$	$\frac{n_{i,j}}{max_j(n_{i,j})}$
$tf_{ln}(v_j, d_i)$	$\ln(1 + n_{i,j})$
$tf_{okapi}(v_j, d_i)$	$\frac{n_{i,j}}{n_{i,j}+1}$
idf	
$idf(v_j)$	$\ln(\frac{\lvert\mathcal{D}\rvert}{\lvert\mathcal{D}_j\rvert})$
$idf_{w_4}(v_j)$	$\ln(\frac{\lvert\mathcal{D}\rvert-\lvert D_j\rvert+0,5}{\lvert D_j\rvert+0,5})$
$idf_{w_2}(v_j)$	$\ln(\frac{\lvert\mathcal{D}\rvert+1}{\lvert D_j\rvert+0,5})$
$tf.idf$	
$tfidf(v_j, d_i)$	$tf(v_j, d_i)idf(v_j)$
$tfidf_{ln}(v_j, d_i)$	$tf_{ln}(v_j, d_i)idf(v_j)$
$tfidf_{w_2}(v_j, d_i)$	$tf_{okapi}(v_j, d_i)idf_{w_2}(v_j)$
$binaire$	
$bin(v_j, d_i)$	$\begin{cases} 1 & \text{si } n_{i,j} > 0 \\ 0 & \text{sinon} \end{cases}$
$bin_{MI}(v_j, d_i)$	$\begin{cases} 1 & \text{si } n_{i,j} > seuil_j \\ 0 & \text{sinon} \end{cases}$

3.3.2.2 Résultats

La figure 3.5 montre le taux de bien classés obtenus pour la catégorisation des images représentées par différentes pondérations basées sur la fréquence des termes dans les images. Ces résultats sont comparés à ceux de référence obtenus précédemment en ne considérant que le nombre d'apparitions des mots dans les images (tf_{raw}). Pour les résultats obtenus à l'aide du descripteur $mstd$, les pondérations tf_{max}, tf_{ln} et tf_{okapi} permettent toutes d'améliorer les résultats de tf_{raw} de 89% à environ 91% quel que soit le nombre de mots visuels choisis dans les images. Cette amélioration fait passer les résultats de ce descripteur au dessus de ceux du descripteur $sift$. Les résultats sont globalement légèrement supérieurs pour la pondération tf_{okapi}. En ce qui concerne les résultats obtenus avec la description $sift$, l'amélioration par rapport à tf_{raw} est moins significative. Les résultats sont même légèrement moins bons pour la pondération tf_{max} qui pour des raisons de performance et de complexité n'ont été calculés que pour un nombre de mots par image supérieur à 3500. Les pondérations tf_{okapi} et tf_{ln} sont légèrement meilleures que tf_{raw} avec un avantage pour tf_{ln} qui permet d'atteindre un taux de 90% de bien classés.

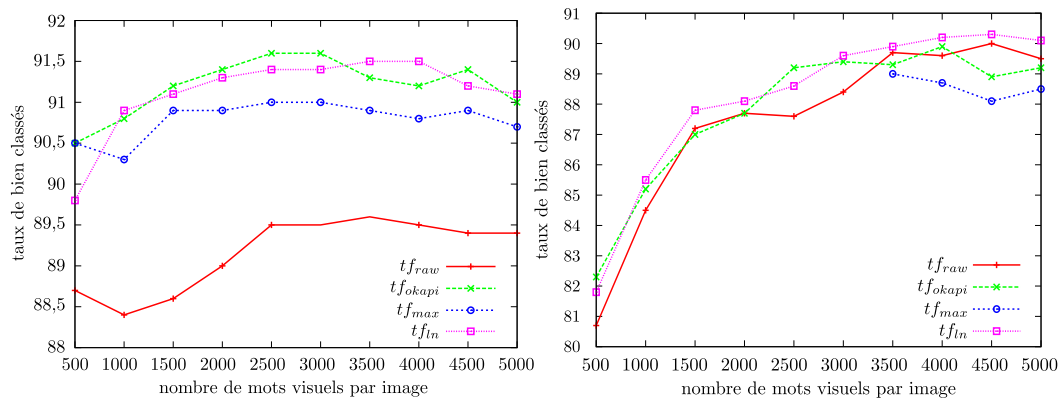


FIG. 3.5 – Résultats comparant les différentes pondérations utilisant la fréquence des termes pour des vocabulaires de 5000 mots basés sur le descripteur mstd (à gauche) et sift (à droite).

Les résultats de la figure 3.6 sont obtenus en considérant la représentation des images à l'aide de la fréquence documentaire calculée par idf, idf_{w_4} et idf_{w_2}. Cette représentation est en fait obtenue en considérant une pondération $tf.idf$ des mots visuels où le tf correspond à la pondération binaire bin. Les résultats sont comparés à ceux obtenus par tf_{raw}. Si pour la description $mstd$, l'utilisation de la fréquence documentaire semble donner de meilleurs résultats que l'utilisation de la fréquence des mots, il convient de rappeler que ces deux pondérations n'exploitent pas la même information et qu'il est donc préférable de ne comparer entres elles que celles basées sur l'inverse de la fréquence du document. Sur cette figure, nous remarquons que le comportement est de nouveau différent selon le descripteur considéré. Contrairement au descripteur $sift$, l'utilisation des formules idf_{w_4} et idf_{w_2} améliore les résultats de l'idf classique pour le descripteur $mstd$. Quel que soit le descripteur, la pondération idf_{w_2} est globalement plus efficace que idf_{w_4}.

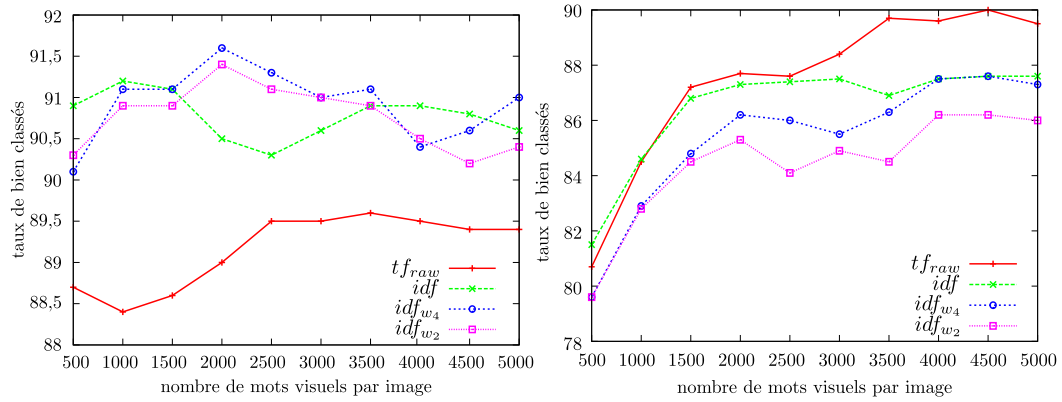


FIG. 3.6 – Résultats comparant les différentes pondérations utilisant la fréquence documentaire pour des vocabulaires de 5000 mots basés sur le descripteur $mstd$ (à gauche) et $sift$ (à droite).

La figure 3.7 présente les résultats obtenus en combinant les pondérations tf et idf. Comme précédemment, nous pouvons voir sur cette figure que les résultats diffèrent selon le descripteur. En effet, les résultats liés au descripteur $mstd$ sont plus hétéro-

gènes que ceux obtenus à partir du descripteur $sift$. Pour les résultats obtenus avec le descripteur $mstd$, l'utilisation des pondérations $tfidf_{ln}$ et $tfidf_{w_2}$ permet d'obtenir de meilleurs résultats que la pondération classique $tfidf$. Parmi elles, la pondération $tfidf_{w_2}$ donne les meilleurs résultats avec des écarts allant jusqu'à 1% de taux de bien classés en plus obtenus pour 3000 mots par image. Les résultats pour le descripteur $sift$ sont globalement plus stables avec un léger avantage pour la pondération $tfidf_{ln}$. Le même comportement s'observe pour les pondérations précédentes avec une nette amélioration des performances lorsque le nombre de mots choisis par image augmente de 500 à 3500. Quel que soit le descripteur considéré, même si les résultats ne sont pas dégradés par l'utilisation d'une pondération $tf.idf$, ils ne sont pas significativement meilleurs. Dans la suite nous avons donc privilégié la pondération $tfidf_{w_2}$ qui obtient de bons résultats pour les deux descripteurs $mstd$ et $sift$.

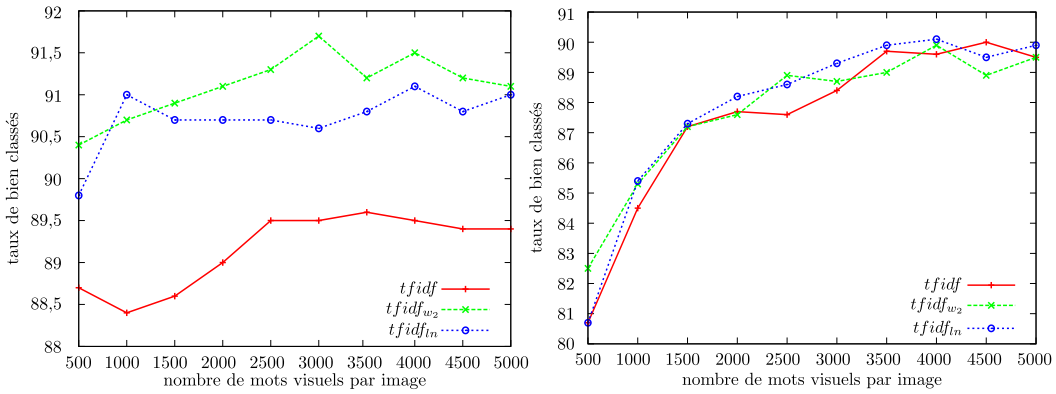


FIG. 3.7 – Résultats comparant les différentes pondérations tf.idf pour des vocabulaires de 5000 mots basés sur le descripteur $mstd$ (à gauche) et $sift$ (à droite).

Les derniers résultats de la figure 3.8 sont obtenus en comparant la pondération $tfidf_{w_2}$ choisi précédemment et les pondérations binaires simple et automatique [Nowak *et al.*, 2006]. Quel que soit le descripteur choisi, la pondération binaire automatique améliore systématiquement les résultats obtenus avec la pondération binaire simple. Pour le descripteur $mstd$, la pondération $tfidf_{w_2}$ améliore les résultats par rapport à la pondération binaire automatique avec environ 1% de bien classés supplémentaire à partir de 2000 mots choisis par image. Ce constat est différent pour le descripteur $sift$ où les résultats obtenus avec la pondération binaire automatique sont légèrement meilleurs à partir de 3000 mots par image. En considérant les deux descripteurs, la pondération $tfidf_{w_2}$ obtient globalement les meilleurs résultats.

Comme nous venons de le voir, la pondération des mots visuels est une étape importante pour représenter efficacement les images. L'influence de cette pondération diffère entre les descripteurs $mstd$ et $sift$ et globalement la pondération $tfidf_{w_2}$ semble donner les meilleurs résultats sur la collection spécifique SIMPLIcity. Ces deux descripteurs considèrent des informations visuelles complémentaires et nous allons maintenant étudier leur utilisation conjointe dans le but d'améliorer encore les résultats.

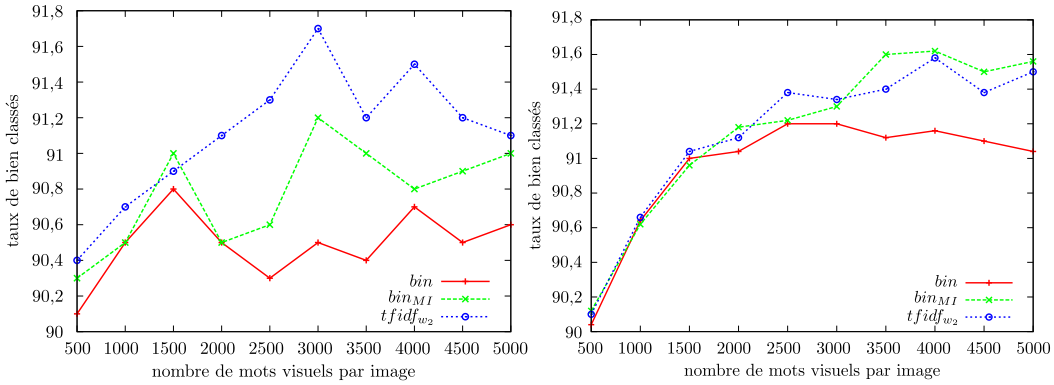


FIG. 3.8 – Résultats comparant la pondération $tfidf_{w_2}$ et les approches binaires pour des vocabulaires de 5000 mots basés sur le descripteur mstd (à gauche) et sift (à droite).

3.4 Fusion de descripteurs visuels

Dans les expérimentations vues jusqu'ici, nous avons considéré en parallèle les descripteurs *mstd* et $sift$ et comparé les résultats obtenus pour chacune de ces modalités. Or, ces descripteurs renseignent chacun sur un aspect de l'image : la couleur et la texture. La combinaison de ces informations en vue d'améliorer les résultats de classification se pose. Nous nous sommes donc intéressés aux méthodes de fusion à appliquer pour encore améliorer les résultats de classification.

Les fusions précoces et tardives sont des fusions classiquement utilisées quel que soit le modèle de représentation des images choisis. Grâce au modèle sac de mots visuels, nous pouvons utiliser une fusion intermédiaire correspondant à la fusion des sacs de mots visuels. Après avoir introduit ces différentes fusions, nous montrerons les résultats obtenus dans un contexte de catégorisation avec la collection SIMPLIcity [Moulin *et al.*, 2010a].

3.4.1 Présentation des différentes fusions

Comme nous l'avons vu précédemment dans le chapitre 1, il est possible d'utiliser, dans le contexte de la catégorisation d'images, les combinaisons classiques : d'une part la fusion précoce qui intervient avant la création du vocabulaire visuel, d'autre part, la fusion tardive qui s'effectue après la tâche de catégorisation [Snoek *et al.*, 2005].

3.4.1.1 Fusion précoce

La plus simple des fusions précoces correspond à une simple concaténation des vecteurs descripteurs d'une imagette. Dans notre cas, les descriptions *mstd* et $sift$ sont calculées pour une région et sont concaténées pour former un vecteur de $128 + 6 = 134$ dimensions. Le vocabulaire $V_{mstd,sift}$ est ensuite construit à partir des descriptions obtenues pour toutes les imagettes considérées.

3.4.1.2 Fusion tardive

La fusion tardive s'effectue à partir des résultats de la catégorisation obtenus pour différents vocabulaires issus de différents descripteurs. En utilisant deux vocabulaires

V_{mstd} et V_{sift}, nous avons un ensemble de scores $\phi^{V_{mstd}}(\overrightarrow{d_i}, c_k)$ et $\phi^{V_{sift}}(\overrightarrow{d_i}, c_k)$ pour $c_k \in \mathcal{C}$. La classe affectée à un document d_i est celle qui a obtenu le score $\phi(\overrightarrow{d_i}, c_k)$ le plus élevé pour $c_k \in \mathcal{C}$ sur l'ensemble des descripteurs choisis. Ainsi, pour le document d_i, la classe associée correspond à celle qui a obtenu le score le plus élevé parmi les scores obtenus pour tous les vocabulaires considérés :

$$\hat{L}(d_i) = \{c_k\} = \begin{cases} c_k = c_{k_{max}}^{V_{mstd}} & \text{si } \phi^{V_{mstd}}(\overrightarrow{d_i}, c_{k_{max}}^{V_{mstd}}) > \phi^{V_{sift}}(\overrightarrow{d_i}, c_{k_{max}}^{V_{sift}}) \\ c_k = c_{k_{max}}^{V_{sift}} & \text{sinon} \end{cases} \tag{3.15}$$

où $c_{k_{max}}^{V_{mstd}}$ et $c_{k_{max}}^{V_{sift}}$ correspondent aux étiquettes affectées à l'image d_i en considérant respectivement le descripteur $mstd$ et $sift$:

$$c_{k_{max}}^{V_{mstd}} \quad | \quad \phi^{V_{mstd}}(\overrightarrow{d_i}, c_{k_{max}}^{V_{mstd}}) = max(\phi^{V_{mstd}}(\overrightarrow{d_i}, c_k), c_k \in \mathcal{C}) \tag{3.16}$$

$$c_{k_{max}}^{V_{sift}} \quad | \quad \phi^{V_{sift}}(\overrightarrow{d_i}, c_{k_{max}}^{V_{sift}}) = max(\phi^{V_{sift}}(\overrightarrow{d_i}, c_k), c_k \in \mathcal{C}) \tag{3.17}$$

3.4.1.3 Fusion des sacs de mots visuels

Une dernière fusion envisagée correspond à la fusion des sacs de mots visuels [Spyrou *et al.*, 2005]. Cette fusion correspond à une concaténation des vecteurs qui représentent les images à partir des vocabulaires V_{mstd} et V_{sift}. Ceci revient donc à représenter les images en utilisant le vocabulaire $V_{mstd \cup sift}$ formé de l'union de V_{mstd} et V_{sift} ($V_{mstd} \cup V_{sift}$). Cette fusion peut être considérée comme une fusion précoce car la fusion s'effectue avant la catégorisation, mais elle peut également correspondre à une fusion tardive par rapport à la tâche qui consiste à créer le vocabulaire.

3.4.2 Expérimentations

Le but de cette partie est d'étudier l'apport de la fusion des sacs de mots visuels. Dans un premier temps, nous comparerons les résultats obtenus à partir des vocabulaires utilisés séparément avec ceux qui fusionnent les sacs de mots visuels, puis nous comparerons les différentes fusions présentées précédemment entre elles.

3.4.2.1 Protocole expérimental

Le protocole expérimental utilisé dans ces expérimentations correspond à celui présenté pour l'obtention des résultats de référence. Deux tailles de vocabulaire (1000 et 5000 mots visuels) seront calculés pour les descripteurs $mstd$ et $sift$. La pondération $tfidf_{w_2}$ sera utilisée pour représenter les images en faisant varier le nombre de mots par image de 500 à 5000. Contrairement aux expérimentations précédentes, la validation est effectuée suivant la méthode du *leave one out*. Ce procédé effectue l'apprentissage sur l'ensemble des images de la collection sauf une puis classe l'image n'ayant pas servie. Le traitement est itéré pour l'ensemble des images de la collection. Cette validation n'a pas été réalisée pour les expérimentations précédentes car elle nécessite beaucoup plus de temps dans la mesure où chaque image requiert un nouvel apprentissage. Contrairement aux expérimentations de référence, l'algorithme de classification utilisé est celui de la librairie liblinear [Fan *et al.*, 2008][1].

[1] `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

3.4.2.2 Résultats

Apport de la fusion des sacs de mots visuels

La figure 3.9 montre les résultats obtenus pour chaque descripteur *mstd* et $sift$ utilisé séparément ainsi que ceux obtenus avec la fusion des sacs de mots. Comme nous l'avons vu dans les expérimentations précédentes, le descripteur *mstd* permet d'obtenir de meilleurs résultats que $sift$ de même que l'utilisation d'un vocabulaire de 5000 mots par rapport à 1000. Les résultats diffèrent légèrement mais sont comparables aux expérimentations de référence bien qu'ils n'utilisent pas la même approche (*leave one out*) et le même algorithme (*liblinear*). Pour les vocabulaires pris séparément, les résultats associés au descripteur *mstd* correspondent à un peu plus de 90% d'images bien classées alors qu'ils sont légèrement supérieurs à 86% pour le descripteur $sift$. Comme nous le montre la figure 3.9, la fusion des sacs de mots visuels améliore significativement les résultats de la catégorisation. En effet, pour le vocabulaire de 1000 mots visuels, les résultats atteignent presque 95% de bien classés et ce taux est dépassé pour un vocabulaire de 5000 mots. L'utilisation de plusieurs vocabulaires visuels qui exploitent différentes information visuelles (couleur, texture) permet d'améliorer les résultats qui n'exploitent qu'une seule information.

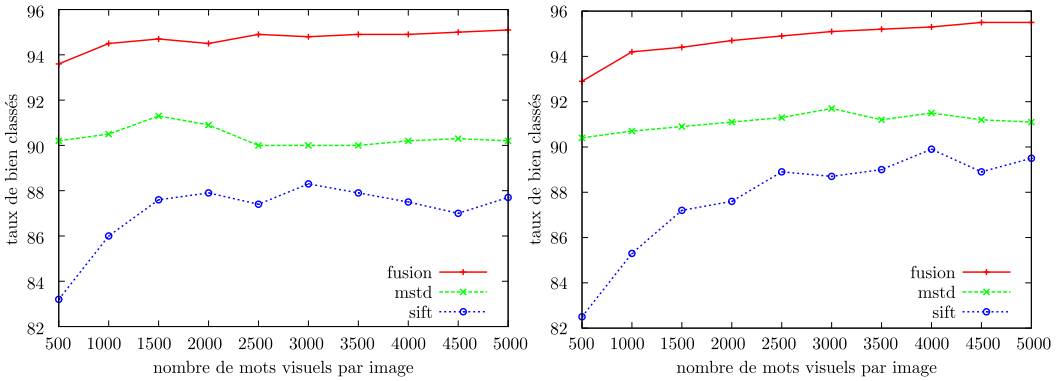


FIG. 3.9 – Résultats comparant l'utilisation de la fusion par rapport aux résultats des modalités séparés pour des vocabulaires de 1000 mots (à gauche) et 5000 mots (à droite).

Comparaison des différentes fusions

Les résultats des comparaisons des différentes méthodes de fusion sont présentés par la figure 3.10. Comme le montre cette figure, les moins bons résultats sont obtenus par la fusion précoce qui concatène les descriptions *mstd* et $sift$. Ces résultats se rapprochent fortement de ceux obtenus par le descripteur $sift$. Ceci s'explique par la nature du nouveau descripteur qui contient plus de 95% d'information liée au descripteur $sift$ ($\frac{128*100}{128+6}$). Les résultats associés à la fusion tardive améliorent légèrement ceux obtenus en utilisant les descripteurs séparément mais restent moins bons que ceux résultant de la fusion des sacs de mots visuels.

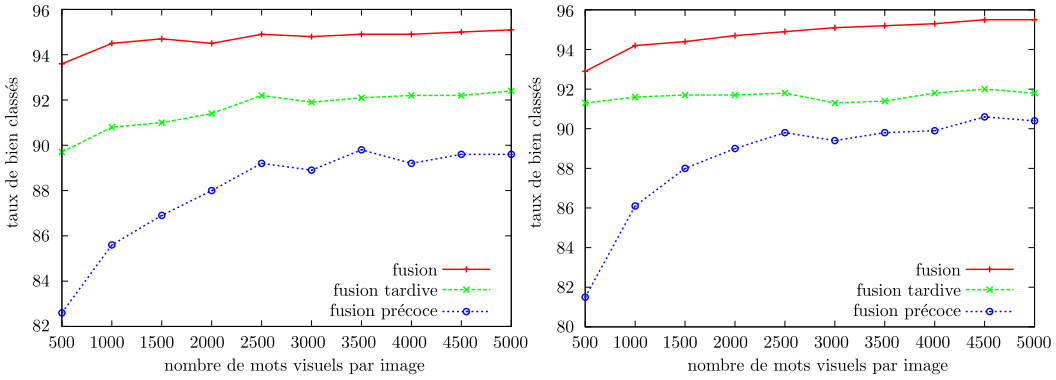


FIG. 3.10 – Résultats comparant les différentes fusions correspondant à la fusion précoce, tardive et à la fusion de sacs de mots pour des vocabulaires de 1000 mots (à gauche) et 5000 mots (à droite).

Dans ce chapitre, nous nous sommes intéressés à la représentation des images en sacs de mots dans un contexte de catégorisation d'images. Nous avons adopté un modèle de référence permettant de caractériser les images à partir d'une détection régulière multi-échelle et d'un vocabulaire visuel donné. L'influence des paramètres intervenant dans cette représentation, comme le nombre de points retenus par image, le descripteur utilisé, la taille du vocabulaire a été étudiée à partir de la collection SIMPLIcity fournissant des résultats de référence pour deux descripteurs $mstd$ et $sift$.

Deux axes d'amélioration de ces résultats ont ensuite été proposés :
– l'utilisation d'un schéma de pondération type $tf.idf$;
– la fusion de plusieurs vocabulaires visuels.

Les résultats de référence obtenus étaient d'environ 89% et 90% pour un vocabulaire visuel de 5000 mots. Nous avons montré que la représentation de l'information visuelle à l'aide de vecteurs pondérés $tf.idf$ permettait d'améliorer les résultats ne considérant que le nombre d'occurrences des mots. Nous avons ainsi étudié différentes pondérations et la pondération $tfidf_{w_2}$ a permis d'obtenir globalement les meilleurs résultats avec environ 1% d'amélioration des résultats de référence.

Enfin, la complémentarité des descripteurs $mstd$ et $sift$ nous a conduit à étudier leur utilisation conjointe. Grâce à la combinaison des sacs de mots visuels, nous avons pu améliorer les résultats de la catégorisation d'images d'environ 6% par rapport aux résultats de référence pour atteindre un taux de bien classés d'environ 95% sur la collection SimpliCITY [Moulin *et al.*, 2010a].

Nous disposons à présent d'outils performants pour représenter les informations textuelle et visuelle sous forme de sac de mots. Soulignons que ces études ont été menées à partir d'une seule collection (SIMPLIcity) et de deux descripteurs ($mstd$ et $sift$). Ainsi les résultats obtenus ne sont donc pas nécessairement généralisables pour tout type d'images et tout descripteur. Dans le chapitre suivant, nous allons mettre ces outils au service de la recherche d'information de documents multimédias.

Chapitre 4

Combinaison des informations textuelle et visuelle

Dans les deux chapitres précédents, nous avons considéré des documents composés d'une seule modalité, soit textuelle, soit visuelle. Dans ce chapitre, nous souhaitons étudier des documents multimédias combinant ces deux informations. De par la nature de ces documents, il est intéressant d'exploiter conjointement les informations textuelles et visuelles contenues dans les documents pour répondre plus efficacement aux besoins des utilisateurs. Nous avons présenté dans les chapitres 2 et 3, comment exploiter ces différentes modalités dans un contexte de catégorisation. Nous avons défini en utilisant une approche par sac de mots, différents vocabulaires adaptés à chaque type d'information. Les documents multimédias peuvent être représentés en ne considérant qu'une seule de ces modalités, mais également en les fusionnant pour tirer parti de toute l'information disponible.

Comme nous l'avons vu dans la partie 1.4.2, la plus simple des approches pour fusionner différentes informations, consiste à combiner linéairement les résultats d'une recherche. Cette approche, dite tardive, pose cependant le problème du poids à accorder à chaque modalité, autrement dit de la valeur à fixer pour les paramètres du modèle.

L'objectif de ce chapitre est d'étudier l'apport respectif de chaque type d'information, textuelle et visuelle et de calculer automatiquement le paramètre de combinaison pour améliorer les résultats d'une recherche n'exploitant qu'une seule modalité. Dans un premier temps, nous proposerons un modèle adapté au contexte de la recherche d'information en présentant la méthode de combinaison linéaire. Ensuite nous introduirons une première approche permettant d'apprendre les poids à associer à chaque modalité en effectuant une recherche exhaustive de la valeur du paramètre et nous étudierons en détail l'influence de ce paramètre, autrement dit de l'impact du poids accorder à chaque type d'information visuelle ou textuelle sur la qualité des résultats du système de recherche d'information. La recherche exhaustive étant une approche coûteuse, elle est difficilement réalisable quand le nombre de modalités utilisées pour décrire les documents augmente. Nous proposerons donc une approche analytique permettant d'apprendre automatiquement les poids à accorder à chaque modalité.

4.1 Présentation du modèle

Le contexte de la recherche d'information étant différent de celui de la catégorisation de documents considérée jusqu'à présent, nous allons dans un premier temps présenter l'architecture globale du système de recherche d'information multimédia. Nous préciserons ensuite le modèle de représentation des informations textuelle et visuelle utilisé ainsi que la méthode permettant de combiner ces informations. Nous présenterons enfin la collection de documents multimédias et les paramètres que nous avons utilisés pour réaliser nos différentes expérimentations.

4.1.1 Architecture globale du système

L'architecture globale du système de recherche d'information multimédia que nous proposons est présentée sur la figure 4.1. Cette architecture comporte plusieurs modules correspondant à l'indexation, au calcul des scores de pertinence et à la combinaison des scores obtenus pour chaque type d'information. Le premier module est consacré à l'indexation des documents de la collection et des requêtes des utilisateurs qui peuvent être composées à la fois de mots textuels et d'une ou plusieurs images. Les contenus textuels et visuels de chaque document sont représentés sous forme de vecteurs pondérés.

Le second module calcule, pour une requête particulière, un score par modalité pour chaque document de la collection. Ce score est d'autant plus élevé que le contenu du document, relativement à la modalité considérée, correspond à celui de la requête.

Le dernier module permet de combiner linéairement les scores obtenus pour chaque modalité dans le but d'identifier les documents qui répondent le mieux à la requête fournie par l'utilisateur.

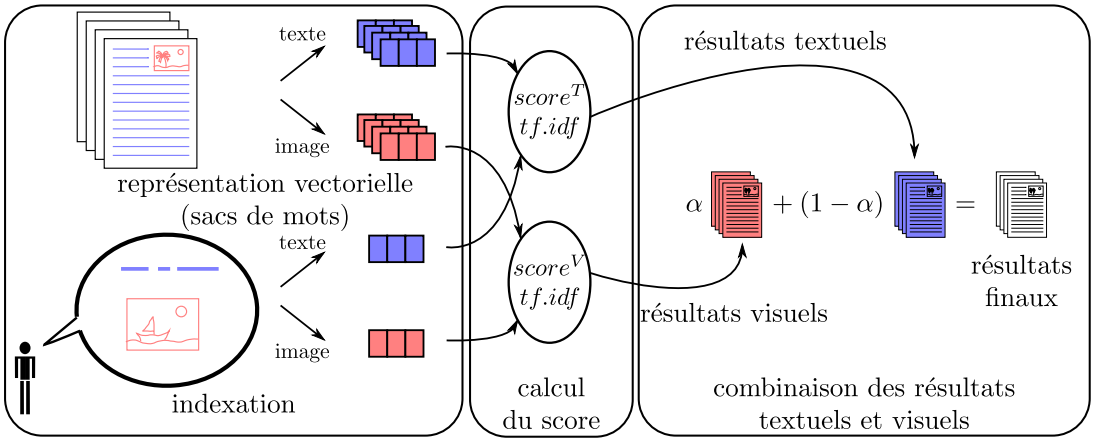


FIG. 4.1 – Architecture globale du modèle de recherche d'information multimodale.

4.1.2 Modèle de représentation textuelle et visuelle

Dans la suite, la collection $\mathcal{D}$ sera composée de documents multimédias contenant une image et du texte. Le vocabulaire textuel $T = \{t_1, \ldots, t_j, \ldots, t_{|T|}\}$ est composé des mots présents dans les documents de la collection $\mathcal{D}$ alors que le vocabulaire visuel $V = \{v_1, \ldots, v_j, \ldots, v_{|V|}\}$ est construit à partir des images de $\mathcal{D}$. Chaque document d_i de $\mathcal{D}$ est ensuite représenté sous la forme de deux vecteurs de poids

$\vec{d_i^T} = (w_{i,1}^T, \ldots, w_{i,j}^T, \ldots, w_{i,|T|}^T)$ et $\vec{d_i^V} = (w_{i,1}^V, \ldots, w_{i,j}^V, \ldots, w_{i,|V|}^V)$ où les poids $w_{i,j}^T$ et $w_{i,j}^V$ sont calculés suivant le modèle tf.idf. Les formules de calcul du poids pour l'information textuelle et pour l'information visuelle sont les mêmes. Dans cette formule, la fréquence $tf_{i,j}$ d'un terme $t_j \in T$ (ou $v_j \in V$) dans un document d_i est définie par :

$$tf_{i,j} = \frac{k_1.n_{i,j}}{n_{i,j} + k_1(1 - b + b\frac{|d_i|}{d_{avg}})} \tag{4.1}$$

où k_1 et b sont deux constantes, $n_{i,j}$ correspond au nombre d'occurrences du terme t_j (ou v_j) dans le document d_i, $|d_i| = \sum_j n_{i,j}$ est la taille du document et d_{avg} est la taille moyenne des documents de $\mathcal{D}$.

La fréquence inverse de document idf_j du terme $t_j \in T$ (ou $v_j \in V$) est définie par :

$$idf_j = \frac{|\mathcal{D}| + 1}{df_j + 0,5} \tag{4.2}$$

où $|\mathcal{D}|$ correspond au nombre de documents dans la collection et df_j au nombre de documents de $\mathcal{D}$ dans lesquels le terme t_j ou v_j apparaît au moins une fois. Le poids $w_{i,j}$ est ensuite obtenu en multipliant $tf_{i,j}$ et idf_j :

$$w_{i,j} = tf_{i,j} \times idf_j \tag{4.3}$$

Nous avons choisi ces formules tf et idf définies à l'origine pour les modalités textuelles, car ce sont celles qui donnent les meilleurs résultats en catégorisation d'images, comme nous l'avons vu dans la partie 3.3.

Une requête q_k, composée de mots textuels q_k^T ou visuels q_k^V, est considérée comme un document et peut également être représentée sous la forme d'un vecteur de poids. Étant donnée une requête q_k, l'objectif est de retrouver une liste de documents jugés pertinents vis à vis de q_k. Pour ce faire deux scores $score^T(q_k^T, d_i^T)$ et $score^V(q_k^V, d_i^V)$ sont calculés respectivement pour la modalité textuelle de la requête par :

$$score^T(q_k^T, d_i^T) = \sum_{t_j \in q_k^T} tf_{i,j} idf_j tf_{k,j} idf_j \tag{4.4}$$

et pour sa modalité visuelle par :

$$score^V(q_k^V, d_i^V) = \sum_{v_j \in q_k^V} tf_{i,j} idf_j tf_{k,j} idf_j \tag{4.5}$$

4.1.3 Combinaison linéaire

Pour exploiter conjointement les informations textuelles et visuelles des documents, nous utilisons une fusion tardive qui consiste à combiner les scores de façon linéaire. Le score global $score(q_k, d_i)$ d'un document d_i pour une requête q_k donnée est donc obtenu par :

$$score_\alpha(q_k, d_i) = \alpha score^V(q_k^V, d_i^V) + (1 - \alpha) score^T(q_k^T, d_i^T) \tag{4.6}$$

Comme nous l'avons vu dans le chapitre 3, il est possible de définir plusieurs vocabulaires visuels. Si nous notons U l'ensemble des vocabulaires visuels permettant de représenter les images, le score global de la combinaison linéaire peut se calculer par :

$$score(q_k, d_i) = (\sum_{u \in U} \alpha_u score^u(q_k^u, d_i^u)) + (1 - \sum_{u \in U} \alpha_u) score^T(q_k^T, d_i^T) \tag{4.7}$$

où α_u correspond au poids accordé à la modalité visuelle $u \in U$. Pour la modalité textuelle, nous ne considérons dans la suite qu'un seul vocabulaire T, mais il serait possible d'en utiliser plusieurs.

4.1.4 Application du système à la collection ImageCLEF

La catégorisation est une tâche simple à évaluer si nous disposons d'un échantillon test composé de documents dont les classes sont connues. En effet, il suffit d'appliquer l'algorithme de classification et de vérifier si les classes ont été correctement trouvées. Pour la recherche d'information, le problème est différent car un même document peut être pertinent pour une requête et non pertinent pour une autre. L'évaluation des documents est donc faite requête par requête. Bien évidemment, quand une collection possède plusieurs milliers ou millions de documents, la recherche des documents qui sont pertinents pour une requête donnée ne peut pas être effectuée manuellement par une seule personne. Pour pouvoir évaluer les systèmes de recherche d'information, des compétitions sont généralement organisées dans le but de tester différents modèles proposés par les participants comme ImageCLEF[1], ou TrecVid[2] [Tsikrika et Kludas, 2008, Tsikrika et Kludas, 2009, Over *et al.*, 2010]. L'évaluation des documents est ensuite faite de façon collaborative par les participants qui n'ont qu'un sous-ensemble des documents à juger.

Afin d'évaluer notre modèle dans le contexte de la recherche d'information, nous avons participé aux compétitions ImageCLEF 2008 et ImageCLEF 2009. Après avoir présenté la collection ImageCLEF, nous détaillerons les paramètres de notre système.

4.1.4.1 Présentation de la collection ImageCLEF

La collection ImageCLEF est une collection multimédia composée de 151 519 documents XML extraits de l'encyclopédie Wikipedia. Ces documents comprennent une image accompagnée d'un texte. Les images ont des tailles très hétérogènes et sont au format JPEG ou PNG. Elles peuvent représenter aussi bien des photos, que des dessins ou des peintures comme illustré par la figure 4.2. Le texte relativement court décrit généralement l'image, mais il peut également contenir des informations relatives à l'utilisateur qui a fourni l'image ou les droits d'utilisation de cette dernière. Les principales caractéristiques de la collection utilisée dans le cadre de la compétition ImageCLEF 2008 et 2009 sont présentées dans la Table 4.1.

TAB. 4.1 – Collection ImageCLEF 2008 et 2009.

	2008	2009
Nombre de documents	151 519	
Nombre moyen de mots textuels par document	33	
Nombre de requêtes	75	45
Nombre moyen d'images par requête	1,97	1,84
Nombre moyen de mots textuels par requête	2,64	2,93

[1] `http://www.imageclef.org`
[2] `http://trecvid.nist.gov`

FIG. 4.2 – Images extraites de la collection ImageCLEF.

Pour les deux éditions ImageCLEF 2008 et ImageCLEF 2009, les 151 519 documents utilisés sont les mêmes, mais l'ensemble des requêtes est différent.

Édition 2008

L'édition 2008 est composée d'un ensemble de 75 requêtes composées d'un texte court, d'une ou plusieurs images et d'un ou plusieurs concepts. Dans la suite, les concepts des requêtes ne seront pas pris en compte. Toutes ces requêtes possèdent au moins une partie textuelle composée en moyenne de 2,64 mots textuels mais 32 d'entre elles n'ont pas d'images requêtes associées. Cet ensemble de requêtes et les documents pertinents fournis par les organisateurs sont utilisés comme un échantillon d'apprentissage pour calculer les paramètres de fusion de notre modèle.

Afin de ne pas pénaliser les requêtes qui ne sont composées que d'une partie textuelle et d'uniformiser l'échantillon d'apprentissage, nous avons décidé de choisir pour l'ensemble des 75 requêtes, deux images requêtes. Ces deux images correspondent aux deux premières images pertinentes retournées après une recherche sur la collection effectuée à partir de la partie textuelle de la requête originale. Dans un système de recherche d'information, ce procédé s'apparente à un retour de pertinence utilisateur.

Édition 2009

En 2009, les organisateurs de la compétition ont décidé de fournir pour chacune des 45 requêtes au moins une image requête. En moyenne, ces requêtes sont composées de 2,93 mots textuels et de 1,84 images. Cet ensemble de requêtes est utilisé comme échantillon de test pour évaluer le système de recherche d'information.

4.1.4.2 Paramétrage du modèle général

Les expérimentations ont été effectuées en utilisant différents modèles de représentation des informations textuelle et visuelle des documents.

Création du vocabulaire textuel

Comme nous l'avons vu dans le chapitre 2, il peut être important de réduire la taille du vocabulaire. Contrairement au contexte de catégorisation, il n'est pas possible d'effectuer une sélection des termes pertinents par rapport à une requête étant donné qu'il n'est pas possible de connaître toutes les requêtes à l'avance. De plus contrairement au contexte de la catégorisation, la suppression d'un mot extrêmement rare est problématique et n'est pas recommandé. En effet, si ce mot rare appartient à la requête d'un utilisateur, il ne faut surtout pas l'avoir supprimé lors de la phase d'indexation. Pour créer le vocabulaire, nous n'utilisons donc aucun anti-dictionnaire ; seule une lemmatisation des mots est réalisée à l'aide de l'algorithme de Porter [Porter, 1980]. Le vocabulaire textuel, qui comporte au final 196 954 mots, sera noté T.

Création des vocabulaires visuels

Dans le chapitre 3, différents vocabulaires ont été étudiés. Les images manipulées en recherche d'information sont très hétérogènes et leur représentation a été obtenue par un découpage en grille régulière. Parmi les détecteurs décrits dans la partie 3.1.1.1, nous avons privilégié le découpage régulier au découpage multi-échelle, le nombre d'images de la collection étant très supérieur à celui de la collection utilisée en catégorisation. Les grilles régulières sont obtenues suivant un découpage en 16×16 régions dont la taille est égale à un seizième de la taille de l'image. Chaque région a ensuite été décrite à l'aide des descripteurs $mstd$ et $sift$ présentés dans la partie 3.1.1.2. Le vocabulaire pour la description $mstd$ (respectivement $sift$) sera noté V_{mstd} (respectivement V_{sift}). La taille de chaque vocabulaire a été fixée empiriquement à 10 000 mots visuels.

Paramétrage du système de recherche d'information

Les valeurs des paramètres de notre système de recherche d'information correspondent à celles utilisées par défaut dans le logiciel Lemur [Zhai, 2001]. Ainsi, la valeur du paramètre k_1 de l'équation 4.1 correspondant à la fonction de pondération d'un terme dans un document ou une requête est fixée à 1. Étant donné que $|d_k|$ et d_{avg} ne sont pas définis pour une requête q_k, $tf_{k,j}$ est calculé en fixant le paramètre b à 0. Pour un document d_i, ce paramètre est fixé à 0,5.

4.2 Approche empirique globale

Le modèle présenté dans un contexte de recherche d'information, décrit les documents multimédias comme des sacs de mots représentés par des vecteurs pondérés tf.idf. Pour prendre en compte le caractère multimédia des documents, le paramètre α permet de pondérer l'information visuelle par rapport à l'information textuelle et de combiner linéairement les scores obtenus pour chaque modalité :

$$score_\alpha(q_k, d_i) = \alpha score^V(q_k^V, d_i^V) + (1-\alpha) score^T(q_k^T, d_i^T) \tag{4.8}$$

Dans cette partie, nous souhaitons évaluer l'apport de l'information visuelle dans une tâche de recherche d'information. Ceci nous a conduit à étudier le poids à accorder à α qui mesure cet apport sur les performances du système.

4.2.1 Mesures d'évaluation

Plusieurs mesures (MAP, $P10$ et $iP[0,1]$) présentées dans la partie 1.1.2.4 ont été utilisées pour évaluer les performances de notre système de recherche d'information. Nous notons MAP_α (respectivement $P10_\alpha$ et $iP[0,1]_\alpha$), la valeur du MAP (respectivement de $P10$ et $iP[0,1]$) obtenue en utilisant α comme paramètre de combinaison. Le MAP considère la précision moyenne des résultats et correspond au critère utilisé dans le cadre de la compétition ImageCLEF pour classer les participants. Nous utilisons également $P10$ qui mesure la précision sur les dix premiers documents retournés par le système et $iP[0,1]$ qui correspond à la précision au point de rappel $0,1$.

4.2.2 Protocole expérimental

Afin d'évaluer l'influence du paramètre α, nous avons effectué différentes expérimentations. La première a consisté à produire des résultats qui serviront de référence. Les suivantes ont consisté à rechercher de façon exhaustive la valeur du paramètre α permettant d'optimiser une mesure d'évaluation ; le but étant d'étudier la stabilité du paramètre de combinaison α par rapport aux différentes mesures d'évaluation en considérant d'abord le MAP, puis $P10$ et $iP[0,1]$.

4.2.2.1 Résultats de référence

Avant d'étudier la combinaison de plusieurs modalités, les résultats n'en exploitant qu'une seule modalité ont été calculés pour les utiliser comme référence. Pour ce faire, nous ne considérerons que la partie textuelle T ou la partie visuelle représentée à l'aide d'un seul descripteur V_{mstd} ou V_{sift} des requêtes de l'édition 2009 d'ImageCLEF. Pour une requête q_k et pour chaque document d_i de $\mathcal{D}$, le $score^T(q_k^T, d_i^T)$ est utilisé pour classer les documents par ordre de pertinence pour le vocabulaire T et les scores $score^{V_{mstd}}(q_k^{V_{mstd}}, d_i^{V_{mstd}})$ et $score^{V_{sift}}(q_k^{V_{sift}}, d_i^{V_{sift}})$ pour les vocabulaires V_{mstd} et V_{sift}.

4.2.2.2 Étude du paramètre α

L'approche globale consiste à utiliser pour toutes les requêtes la même valeur pour le paramètre de combinaison α, notée α_g. Afin de combiner les informations textuelles et visuelles, nous proposons de calculer la valeur du paramètre α_g qui optimise le critère MAP sur une collection d'apprentissage formée d'un ensemble de requêtes et de documents pertinents associés à chaque requête. Une fois calculée, cette valeur α_g peut être utilisée pour traiter une nouvelle requête. Ainsi la valeur globale du paramètre α_g est définie par :

$$\alpha_g = \underset{\alpha \in [0,1]}{\arg\max}\ MAP_\alpha \tag{4.9}$$

La recherche de α_g est effectuée à partir des requêtes de la collection 2008 et les documents pertinents associés à chaque requête. Le paramètre de combinaison obtenu est noté α_g^{2008}. Le calcul de α_g^{2008} est réalisé par une recherche exhaustive des valeurs possibles de α comprises entre 0 et 1 avec un pas de 0,001. La même méthode peut être réalisée sur la collection 2009 pour calculer α_g^{2009} qui permet d'obtenir les meilleurs résultats en combinant l'information textuelle et l'information visuelle sur les requêtes de 2009. Dans le cadre de la compétition, cette approche n'est évidemment pas envisageable, car les documents pertinents ne sont pas connus au préalable. La valeur α_g^{2009} obtenue correspond donc à une valeur idéale qui peut être comparée à α_g^{2008} pour

vérifier la stabilité de la valeur du paramètre de combinaison et analyser la pertinence de l'apprentissage du paramètre α_g de combinaison.

4.2.2.3 Stabilité du paramètre α par rapport aux mesures d'évaluation

Le but de ces expérimentations complémentaires est de vérifier si l'importance de l'information visuelle par rapport à l'information textuelle, quantifiée par la valeur de α, reste la même selon que nous privilégions une recherche exhaustive retournant un nombre important de résultats, pas nécessairement tous pertinents, ou au contraire une recherche précise renvoyant moins de résultats, mais avec moins d'erreurs. Pour ce faire, nous nous proposons de comparer les valeurs optimales de α_g^{2008}, apprises à partir de l'ensemble des requêtes de la collection 2008, avec les valeurs idéales α_g^{2009}, calculées sur l'ensemble des requêtes de la collection 2009, pour différents critères d'évaluation à savoir les mesures $P10$ et $iP[0,1]$ qui mettent l'accent sur la précision contrairement au MAP qui tient également compte du rappel.

4.2.3 Résultats

Après avoir présenté les résultats des différentes expérimentations, nous reviendrons sur la façon de calculer le paramètre de combinaison α_g et sur les limites d'une telle approche.

4.2.3.1 Résultats de référence

La table 4.2 regroupe les valeurs des mesures MAP, $P10$ et $iP[0,1]$ obtenues à partir des vocabulaires T, V_{mstd} et V_{sift} utilisés séparément. D'après les organisateurs de la compétition, 1 622 documents pertinents sont à retrouver pour l'ensemble des 45 requêtes de la collection 2009.

TAB. 4.2 – MAP, $P10$, $iP[0,1]$ et nombre de documents pertinents retrouvés sur la collection ImageCLEF 2009 en exploitant les vocabulaires T, V_{mstd} et V_{sift} séparément.

Expérimentations	MAP	$P10$	$iP[0,1]$	Nombre de documents retournés	Nombre de documents pertinents retrouvés
T	**0,1667**	**0,2733**	**0,3929**	35 611	1 192
V_{mstd}	0,0060	0,0200	0,0187	45 000	113
V_{sift}	0,0085	0,0178	0,0160	45 000	187

Les résultats obtenus sur la collection ImageCLEF 2009 montrent clairement que la modalité textuelle permet de retrouver principalement les documents pertinents (1 192 sur 1 622) et que les modalités visuelles $mstd$ et $sift$ sont beaucoup moins efficaces. En effet, seuls 113 documents et 187 documents ont pu être retrouvés en utilisant respectivement les descripteurs $mstd$ et $sift$. Concernant les critères MAP, $P10$ et $iP[0,1]$, ceux-ci sont, sans réelle surprise, très faibles pour les deux descripteurs visuels. Dans la suite, pour comparer l'apport de l'information visuelle, nous utiliserons donc

la valeur du critère MAP (respectivement $P10$ et $iP[0,1]$) de $0,1667$ (respectivement $0,2733$ et $0,3929$) obtenu par la modalité T.

4.2.3.2 Étude du paramètre α

Pour simplifier la présentation des résultats, seuls ceux combinant l'information textuelle et l'information visuelle décrite avec le descripteur $sift$ seront présentés. Pour les autres résultats qui conduisent aux mêmes conclusions, le lecteur est invité à lire l'annexe B.2.

TAB. 4.3 – Résultats combinant l'information textuelle et visuelle sur la collection ImageCLEF 2009 en utilisant la mesure MAP.

Expérimentations	MAP	Gain par rapport au texte seul T
T	0,1667	
T et V_{sift} ($\alpha_g^{2008} : 0,084$)	0,1903	+14,16%
T et V_{sift} ($\alpha_g^{2009} : 0,085$)	0,1905	+14,28%

La table 4.3 montre les résultats obtenus pour le critère MAP sur l'ensemble des requêtes de la collection 2009 en combinant les vocabulaires T et V_{sift} pour le paramètre de combinaison optimal α_g^{2008} appris sur les requêtes de la collection 2008 (courbe MAP 2008 sur la figure 4.3) et le paramètre de combinaison idéal α_g^{2009} calculé sur les requêtes de 2009 (courbe MAP 2009 sur la figure 4.3). Comme nous pouvons le constater sur cette table, la combinaison des informations textuelles et visuelles permet une amélioration par rapport aux résultats textuels de référence de plus de 14%. Par rapport à la valeur idéale du MAP obtenue avec α_g^{2009} qui est de $0,1905$, la valeur du MAP de $0,1903$ calculée avec α_g^{2008} montre que le calcul de la valeur du paramètre de combinaison est envisageable sur une collection d'apprentissage. En effet, les valeurs des paramètres α_g^{2008} et α_g^{2009} sont très proches : $0,084$ et $0,085$.

La figure 4.3 montre l'évolution des valeurs des MAP sur l'ensemble des requêtes des éditions 2008 et 2009 en fonction des valeurs de α. Sur cette figure, nous pouvons voir que l'augmentation du paramètre α permet d'améliorer fortement les résultats, notamment pour des valeurs α proches de $0,1$. Ces résultats sont cependant fortement dégradés lorsque la valeur de α devient supérieur à $0,1$. La prise en compte de l'information visuelle permet donc d'améliorer les résultats d'une recherche par rapport à l'utilisation de l'information textuelle seule et le poids à lui accorder peut se calculer à partir d'un ensemble de requêtes.

4.2.3.3 Stabilité du paramètre α par rapport aux mesures d'évaluation

Par rapport à la mesure MAP présentée par la figure 4.3, les mesures d'évaluations plus spécifiques, orientées précision comme $P10$ et $iP[0,1]$ semblent moins stables comme le montrent les figures 4.4 et 4.5 où les courbes $P10$ 2008 et $P10$ 2009 ont été obtenues respectivement à partir des requêtes de la collection 2008 et celles de 2009. Il existe une différence entre les valeurs du paramètre calculées à partir des collections 2008 et 2009 qui peut s'expliquer par le nombre de requêtes entre ces deux collections

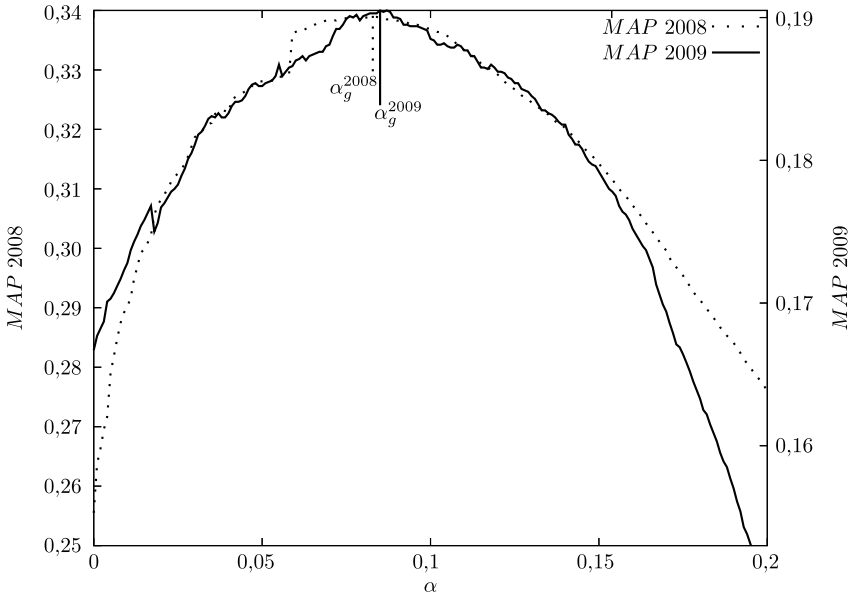


FIG. 4.3 – Variation de la mesure MAP en fonction du paramètre α_g pour 2008 et 2009.

qui est de 75 pour 2008 et 45 pour 2009. Rappelons que les mesures $P10$ et $iP[0,1]$ sont des moyennes alors que la mesure MAP est une moyenne de moyenne.

Pour les mesures $P10$ et $iP[0,1]$, les valeurs optimales du paramètre α apprises sur 2008 ($P10$: $\alpha_g^{2008} = 0,140$; $iP[0,1]$: $\alpha_g^{2008} = 0,108$) diffèrent des valeurs idéales de α calculées pour 2009 ($P10$: $\alpha_g^{2009} = 0,095$; $iP[0,1]$: $\alpha_g^{2009} = 0,078$). Si la différence entre les valeurs de α_g^{2008} et α_g^{2009} peut sembler grande, dans les deux cas, l'amélioration des mesures reste importante. Que le paramètre de combinaison soit calculé en considérant la collection 2008 ou celle de 2009, les tables 4.4 et 4.5 montrent que la combinaison des informations textuelles et visuelles à l'aide de α_g^{2008} et α_g^{2009} améliore les résultats de référence basés uniquement sur l'information textuelle. En effet, en utilisant le paramètre α_g^{2008} appris sur 2008, le gain est de 19,54% pour la mesure $P10$ et de 9,49% pour $iP[0,1]$ par rapport aux résultats obtenus avec l'information textuelle seule. Ces résultats sont bons par rapport à ceux obtenus avec le paramètre idéal α_g^{2009} qui conduit à une amélioration de 20,34% pour $P10$ et de 13,67% pour $iP[0,1]$.

TAB. 4.4 – Résultats sur la collection ImageCLEF 2009 en utilisant la mesure $P10$.

Expérimentations	$P10$	Gain par rapport au texte seul
T	0,2733	
T et V_{sift} ($\alpha_g^{2008} : 0,140$)	0,3267	+19,54%
T et V_{sift} ($\alpha_g^{2009} : 0,095$)	**0,3289**	+20,34%

Il est difficile d'interpréter les valeurs des paramètres de combinaison obtenus pour les différentes mesures d'évaluation, mais la contribution visuelle paraît plus importante pour la mesure $P10$ que pour les mesures MAP et $iP[0,1]$. L'apport de l'information

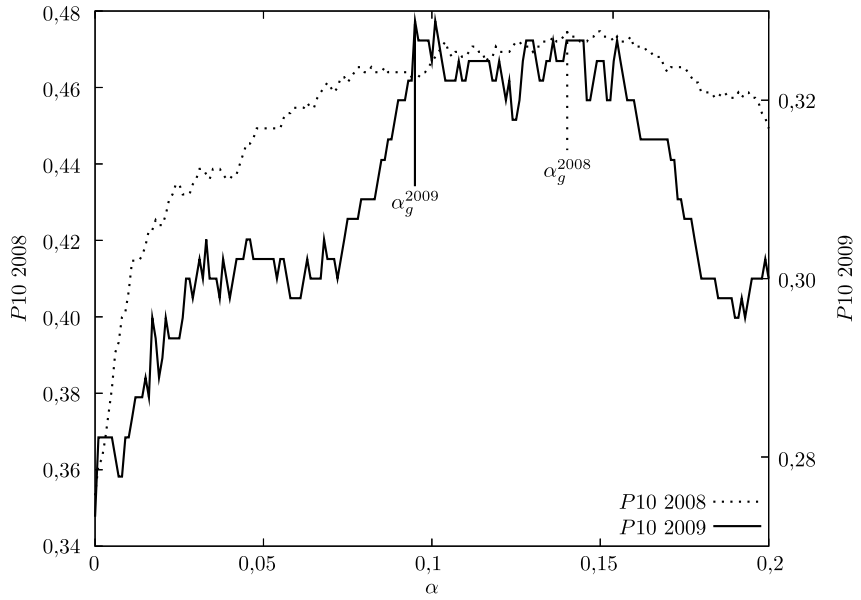


FIG. 4.4 – Variation de la mesure $P10$ en fonction du paramètre α_g pour 2008 et 2009.

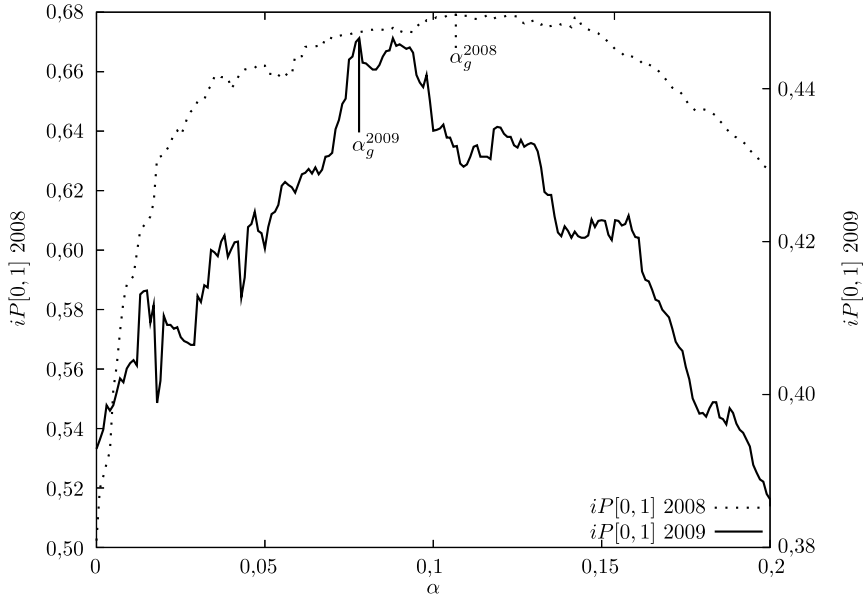


FIG. 4.5 – Variation de la mesure $iP[0,1]$ en fonction du paramètre α_g pour 2008 et 2009.

TAB. 4.5 – Résultats sur la collection ImageCLEF 2009 en utilisant la mesure $iP[0,1]$.

Expérimentations	$iP[0,1]$	Gain par rapport au texte seul
T	0,3929	
T et V_{sift} ($\alpha_g^{2008} : 0,108$)	0,4302	+9,49%
T et V_{sift} ($\alpha_g^{2009} : 0,078$)	**0,4466**	+13,67%

visuelle semble contribuer à l'augmentation de la précision au détriment du rappel.

4.3 Étude avancée de l'utilisation du paramètre de fusion α

Nous venons de voir qu'il était possible d'améliorer les résultats en combinant les informations textuelles et visuelles par une approche globale. Cependant, certaines requêtes semblent avoir un profil plutôt textuel comme les *gens avec des chiens* ou les *musiciens de rue*, alors que d'autres ont un profil plutôt visuel, comme les *fruits rouges* ou les *arcs en ciel*. Compte tenu de ces spécificités, on peut se demander s'il ne faudrait pas adapter le modèle pour chaque type de requêtes ; ce qui en pratique reviendrait à choisir la valeur de α pour chaque requête.

Nous nous proposons donc d'étudier l'influence du paramètre α pour chaque requête en vérifiant si la valeur idéale du paramètre de combinaison calculée sur la collection 2009 est approximativement la même pour toutes les requêtes ou au contraire s'il existe différents types de requêtes. Cette approche locale n'est cependant pas réalisable facilement, car elle nécessite de connaître préalablement le type de la requête ; ce qui reste un problème ouvert. Néanmoins, la comparaison entre les résultats de l'approche globale et ceux de l'approche locale, permettra de se rendre compte si des améliorations seraient envisageables s'il était possible de spécifier le type des requêtes (plutôt visuel ou plutôt textuel) par exemple en les catégorisant au préalable.

4.3.1 Protocole expérimental

Après avoir présenté l'approche locale, nous expliquerons la méthode utilisée pour la comparer à l'approche globale de la partie 4.2.

4.3.1.1 Optimisation du paramètre α par rapport à une requête

L'approche locale vise à calculer une valeur α_k du paramètre de combinaison correspondant à la valeur idéale de α pour la requête q_k. En calculant ensuite la moyenne et l'écart type des différentes valeurs de α_k, nous pourrons conclure sur la variation du paramètre α en fonction des requêtes et sur l'intérêt de mettre en place des méthodes pour estimer une valeur α_k à utiliser pour chaque nouvelle requête soumise par l'utilisateur.

L'étude prendra en compte les trois principales mesure d'évaluation, MAP, $P10$ et $iP[0,1]$ en calculant, de façon exhaustive, les différentes valeurs α_k qui optimisent ces mesures.

4.3.1.2 Comparaison entre une approche globale et une approche locale

Dans l'approche globale décrite dans la partie 4.2, une valeur α_g est calculée et est utilisée ensuite pour toutes les requêtes. La valeur idéale α_g^{2009} est apprise sur les requêtes de la collection 2009 en optimisant une mesure particulière.

Dans l'approche locale, on souhaite optimiser les mesures AP_k, $P_k(10)$ et $iP_k[0,1]$ définies dans la partie 1.1.2.4 pour chaque requête q_k de l'ensemble des requêtes Q. Ainsi, il existe une valeur idéale α_{k_l} qui optimise la mesure AP_k pour la requête q_k. Nous calculons ensuite MAP_{α_l} en faisant la moyenne des valeurs AP_k optimisées :

$$MAP_{\alpha_l} = \frac{\sum_{k=1}^{|Q|} AP_k|\alpha_k = \alpha_{k_l}}{|Q|} \tag{4.10}$$

Il est alors possible de calculer la moyenne (μ_{α_l}) et l'écart-type (σ_{α_l}) des différentes valeurs idéales α_k :

$$\mu_{\alpha_l} = \frac{\sum_{k=1}^{|Q|} \alpha_k}{|Q|} \tag{4.11}$$

$$\sigma_{\alpha_l} = \sqrt{\frac{\sum_{k=1}^{|Q|} (\alpha_k - \mu_{\alpha_l})^2}{|Q|}} \tag{4.12}$$

La même étude peut être réalisée pour les mesures $P_k(10)$ et $iP_k[0,1]$.

4.3.2 Résultats

Dans l'hypothèse où les valeurs α_k peuvent être calculées pour chaque requête, l'approche locale offre une marge de progression importante avec une amélioration potentielle de 29,99% (respectivement 52,87%, 39,14%) pour la mesure MAP (respectivement $P10$, $iP[0,1]$), comme le montre la table 4.6. Cependant, la mise en place de cette approche paraît difficile. D'après les valeurs de moyennes μ_{α_l} et des écarts-types σ_{α_l} observées pour les 3 mesures d'évaluation, nous pouvons voir que les valeurs de $\alpha_{k_l}, k \in |Q|$ sont principalement comprises entre $0,017$ et $0,143$ (respectivement entre $-0,003$ et $0,113$ et entre $0,011$ et $0,155$) lorsque le critère MAP (respectivement $P10$ et $iP[0,1]$) est optimisé. Sachant que toutes les valeurs de $\alpha_{k_l}, k \in |Q|$ sont comprises entre 0 et $0,2$, il existe une forte disparité des valeurs de $\alpha_{k_l}, k \in |Q|$ en fonction des requêtes.

L'information visuelle ne permet pas à elle seule d'obtenir de bons résultats. En revanche, en la combinant à l'information textuelle, l'apport peut être relativement important en fonction de la mesure d'évaluation considérée. Dans l'idéal, il faudrait pouvoir effectuer une catégorisation sur les requêtes afin de calculer un paramètre de combinaison spécifique à chaque classe de requêtes. Même si cette approche locale n'est pas réalisable, nous avons montré dans la partie 4.2 que la recherche du paramètre de combinaison est au moins réalisable de façon globale et permet d'améliorer les résultats obtenus à partir d'une seule modalité. En revanche, cette recherche exhaustive est possible pour combiner deux modalités, mais se complexifie quand le nombre de modalités augmente. En effet, pour deux modalités et un pas de $0,001$, la phase d'apprentissage nécessite pour chaque requête 1000 nouveaux calculs des scores des 150 000 documents

TAB. 4.6 – Résultats obtenus après optimisation de α_k pour chaque requête.

	Expérimentations		Gain par rapport texte seul T	μ_{α_l}	σ_{α_l}
MAP	T	0,1667			
	T+V_{sift} avec $\alpha_g^{2009} : 0,085$)	0,1905	+14,28%		
	T+V_{sift} avec $\alpha_{k_l}, k \in \|Q\|$	**0,2167**	+29,99%	0,080	0,063
$P10$	T	0,2733			
	T+V_{sift} avec $\alpha_g^{2009} : 0,095$	0,3289	+20,34%		
	T+V_{sift} avec $\alpha_{k_l}, k \in \|Q\|$	**0,4178**	+52,87%	0,055	0,058
$iP[0,1]$	T	0,3929			
	T+V_{sift} avec $\alpha_g^{2009} : 0,078$	0,4466	+13,67%		
	T+V_{sift} avec $\alpha_{k_l}, k \in \|Q\|$	**0,5467**	+39,14%	0,083	0,072

de la collection. De façon générale, pour un nombre n de modalités, $|D|$ de documents et un pas $p \in]0,1[$, le nombre de nouveaux scores à calculer pour une requête est de :

$$|D| \times \left(\frac{1}{p}\right)^{(n-1)} \tag{4.13}$$

La complexité de la recherche exhaustive étant exponentielle en fonction du nombre de modalités, cette approche ne peut pas être entièrement satisfaisante. Par contre, l'étude expérimentale menée précédemment confirme son intérêt pour améliorer les résultats de référence. Dès lors, nous avons chercher à calculer plus efficacement la valeur du paramètre de combinaison et la section suivante présente l'approche analytique que nous proposons.

4.4 Approche analytique

Dans cette partie, nous proposons de calculer les paramètres de fusion à l'aide d'une méthode issue du domaine de l'apprentissage automatique. En apprentissage supervisé, les données d'apprentissage correspondent à un ensemble de couples (entrée, sortie). L'objectif est alors d'élaborer un modèle qui à partir des données d'entrée permet, pour un nouvel élément, de prédire la sortie. Différentes méthodes peuvent alors être utilisées pour réaliser l'apprentissage en fonction de la nature des données. Celles-ci peuvent être numériques (données quantitatives) ou non (données qualitatives). Si les données en entrée et en sortie sont quantitatives, le problème peut être résolu à l'aide de méthodes de régression [Saporta, 2006]. En revanche, si elles sont qualitatives, on peut avoir recours à des arbres de décision [Quinlan, 1996]. Dans le cas où les données en entrée sont quantitatives et en sortie qualitatives, différentes méthodes peuvent être envisagées, comme par exemple les SVM [Cortes et Vapnik, 1995]. En recherche d'information, les données en entrée, correspondant à un vecteur de poids tf.idf, sont quantitatives. Les données de sortie, quant à elles, correspondent dans l'idéal à une liste triée de documents (données quantitatives). Cependant, cette information n'est généralement disponible que sous la forme qualitative : le document est pertinent ou non. De ce fait, on perd la notion d'ordre de pertinence.

Pour traiter le problème, nous l'assimilons à un problème de catégorisation binaire où pour chaque requête, un document est soit pertinent soit non pertinent. La combi-

naison des scores textuels et visuels que nous étudions étant linéaire, notre but consiste à déterminer les différents paramètres de combinaison grâce à l'analyse discriminante [Mahalanobis, 1936, Lebart et Fénelon, 1971, Saporta, 2006]. Après avoir expliqué cette approche dans un contexte général puis dans le cas particulier à deux classes qui nous concerne, nous présenterons les expérimentations réalisées et les résultats obtenus.

4.4.1 Présentation de l'analyse discriminante

L'analyse discriminante permet de trouver la combinaison linéaire qui permet de décrire et prédire l'appartenance à des catégories prédéfinies. Dans sa version paramétrique, cette technique suppose que les catégories sont multinormales : c'est l'hypothèse de multinormalité, et que les variances associées aux catégories sont semblables : c'est l'hypothèse d'homoscédasticité. Ces hypothèses sont très fortes et contraignantes, mais l'analyse discriminante reste une méthode très populaire car même si ces hypothèses ne sont pas vérifiées, elle est robuste et conduit à de très bons résultats [Bouveyron *et al.*, 2005]. C'est la raison pour laquelle elle a été employée dans des applications très variées : en économie, elle sert par exemple à prédire la faillite d'une banque ou d'une entreprise [Altman, 1968, Bardos et Zhu, 1997] ; en médecine, elle permet de distinguer les cellules cancéreuses de celles qui sont saines [Wallace *et al.*, 1997].

Étant donnée une requête q, $s_{i,j}$ désignera le score obtenu pour le document d_i et la requête q en considérant la modalité j. Le nombre de modalités n'est pas limité, mais pour simplifier la présentation de l'analyse discriminante, nous ne considérons qu'une seule modalité textuelle et une seule modalité visuelle : $j \in \{T, V\}$, la généralisation à plus de deux modalités étant évidente.

Soit $\mathcal{S}$ l'ensemble des scores $s_{i,j}$ obtenu pour toutes les modalités j et tous les documents d_i de $\mathcal{D}$:

$$S = \{s_{i,j}, i \in \{1, \ldots, |D|\}, j \in \{T, V\}\}$$

Sur l'ensemble des documents de la collection, le score moyen $\overline{s_j}$ obtenu pour la modalité j est défini par :

$$\overline{s_j} = \frac{1}{|\mathcal{D}|} \sum_{d_i \in \mathcal{D}} s_{i,j}$$

Le but est de chercher une combinaison linéaire des scores visuels et textuels permettant de séparer le mieux possible, pour la requête q, les documents pertinents de ceux qui ne le sont pas. Étant donnée une variable u associée à cette combinaison linéaire ; les termes u_j de u correspondent aux coefficients α_j de notre modèle. Pour le document d_i, nous notons $u(i)$ la valeur de la combinaison linéaire calculée avec la variable u :

$$u(i) = \sum_j \alpha_j (s_{i,j} - \overline{s_j})$$

La variable u étant centrée, la variance de cette variable calculée sur $\mathcal{D}$ est égale à :

$$\begin{aligned} V(u) &= \frac{1}{|\mathcal{D}|} \sum_{d_i \in \mathcal{D}} u(i)^2 \\ &= \frac{1}{|\mathcal{D}|} \sum_{d_i \in \mathcal{D}} \sum_{j \in \{T,V\}} \sum_{j' \in \{T,V\}} \alpha_j \alpha_{j'} (s_{i,j} - \overline{s_j})(s_{i,j'} - \overline{s_{j'}}) \end{aligned}$$

En posant $t_{jj'} = \frac{1}{|\mathcal{D}|}(s_{i,j} - \overline{s_j})(s_{i,j'} - \overline{s_{j'}})$, on obtient :

$$\begin{aligned} V(u) &= \sum_{j \in \{T,V\}} \sum_{j' \in \{T,V\}} \alpha_j \alpha_{j'} t_{jj'} \\ &= {}^t u T u \end{aligned}$$

où T est la matrice de variance-covariance des deux modalités de terme général $t_{j,j'}$, u est la matrice de terme général α_j correspondant aux coefficient de la combinaison linéaire et ${}^t u$ est la transposée de u. La variance de u peut se décomposer d'après le théorème de Huygens en variance intra-classe W (*within*) et en variance inter-classe B (*between*).

$${}^t u T u = {}^t u W u + {}^t u B u$$

La combinaison linéaire qui permettra de séparer au mieux les documents pertinents des non pertinents par rapport à la requête q correspond à celle qui maximisera la variance entre les classes et qui minimisera la variance à l'intérieur des classes. La variance totale étant constante, cette combinaison linéaire u doit maximiser la fonction $f(u)$:

$$f(u) = \frac{{}^t u B u}{{}^t u T u}$$

Comme la fonction $f(u)$ est homogène de degré 0 en u, c'est-à-dire invariante si on multiplie u par un scalaire quelconque, maximiser $f(u)$ revient à maximiser ${}^t u B u$ sous la contrainte ${}^t u T u = 1$.

Ce problème d'optimisation peut être résolu en faisant appel à la méthode des multiplicateurs de Lagrange [Lagrange, 1853] et en calculant la dérivée $\nabla\phi(u)$ de la fonction $\phi(u)$ définie par :

$$\phi(u) = {}^t u B u - \lambda({}^t u T u - 1)$$

soit

$$\nabla\phi(u) = 2Bu - 2\lambda T u$$

Le maximum est atteint lorsque la dérivée du lagrangien par rapport à u s'annule :

$$\begin{aligned} & \nabla\phi(u) = 0 \\ \Leftrightarrow \quad & 2Bu - 2\lambda T u = 0 \\ \Leftrightarrow \quad & Bu = \lambda T u \end{aligned} \tag{4.14}$$

En général, la matrice de variance-covariance T étant inversible, nous avons :

$$T^{-1} B u = \lambda u$$

En multipliant 4.14 par ${}^t u$, on obtient comme ${}^t u T u = 1$

$${}^t u B u = \lambda$$

Ainsi λ correspond à la plus grande valeur propre de $T^{-1}B$ et u est le vecteur propre de $T^{-1}B$ relatif à λ.

4.4.2 Cas d'un problème à deux classes

En assimilant le problème à un problème de catégorisation, il est possible de répartir les documents en deux classes R et $\overline{R}$ selon qu'ils sont pertinents ou non pertinents pour la requête q. Dans la suite, $\mathcal{D}_R$ (respectivement $\mathcal{D}_{\overline{R}}$) désignera l'ensemble des documents de $\mathcal{D}$ qui sont pertinents (respectivement non pertinents) pour la requête q :

$$\begin{aligned} \mathcal{D}_R &= \{d_i \in \mathcal{D} | d_i \text{ pertinent pour } q\} \\ \mathcal{D}_{\overline{R}} &= \{d_i \in \mathcal{D} | d_i \text{ non pertinent pour } q\} \end{aligned}$$

Ainsi nous avons : $|D| = |D_R| + |D_{\overline{R}}|$. Il est alors possible de calculer la moyenne pour chaque modalité $j \in \{T, V\}$ dans chacune des classes :

$$\begin{aligned} \overline{s_{R,j}} &= \frac{1}{|\mathcal{D}_R|} \sum_{d_i \in \mathcal{D}_R} s_{i,j} \\ \overline{s_{\overline{R},j}} &= \frac{1}{|\mathcal{D}_{\overline{R}}|} \sum_{d_i \in \mathcal{D}_{\overline{R}}} s_{i,j} \end{aligned}$$

En remarquant que le terme général de la matrice de la variance inter-classe B est :

$$b_{j,j'} = \frac{|\mathcal{D}_R|}{|\mathcal{D}|}(\overline{s_{R,j}} - \overline{s_j})(\overline{s_{R,j'}} - \overline{s_{j'}}) + \frac{|\mathcal{D}_{\overline{R}}|}{|\mathcal{D}|}(\overline{s_{\overline{R},j}} - \overline{s_j})(\overline{s_{\overline{R},j'}} - \overline{s_{j'}}) \tag{4.15}$$

avec

$$\overline{s_j} = \frac{|\mathcal{D}_R|}{|\mathcal{D}|}\overline{s_{R,j}} + \frac{|\mathcal{D}_{\overline{R}}|}{|\mathcal{D}|}\overline{s_{\overline{R},j}}$$

et en reportant $\overline{s_j}$ dans 4.15, nous obtenons :

$$b_{j,j'} = \frac{|\mathcal{D}_R||\mathcal{D}_{\overline{R}}|}{|\mathcal{D}|^2}(\overline{s_{R,j}} - \overline{s_{\overline{R},j}})(\overline{s_{R,j'}} - \overline{s_{\overline{R},j'}})$$

La matrice B est symétrique et peut être calculée comme le produit d'une matrice colonne c par sa transposée :

$$B = c^t c \tag{4.16}$$

où c désigne la matrice de terme général $c_j = \frac{\sqrt{|\mathcal{D}_R||\mathcal{D}_{\overline{R}}|}}{|\mathcal{D}|}(\overline{s_{R,j}} - \overline{s_{\overline{R},j}})$, $j \in \{T, V\}$.

En reportant 4.16 dans 4.14, on a :

$$\begin{aligned} & c^t c u = \lambda T u \\ \Leftrightarrow \quad & T^{-1} c^t c u = \lambda u \end{aligned}$$

En multipliant par ${}^t c$, on obtient :

$$({}^t c T^{-1} c)^t c u = \lambda {}^t c u$$

avec $({}^t c T^{-1} c) = \lambda$ qui est une valeur propre de $T^{-1}B$.

Le vecteur propre u correspondant aux paramètres α_j de la combinaison linéaire qui sépare au mieux les documents pertinents des non pertinents, peut donc se calculer simplement par $T^{-1}c$.

4.4.3 Protocole expérimental

Pour évaluer l'approche présentée précédemment, nous nous sommes à nouveau appuyés sur la base de documents des collections ImageCLEF 2008 et 2009. Cependant, ne disposant plus des résultats intermédiaires nécessaires à l'évaluation, nous avons dû recalculer les descripteurs et les vocabulaires visuels. Comme certaines étapes de ces calculs utilisent des tirages aléatoires, notamment pour l'algorithme des k-means, les vocabulaires visuels V_{mstd} et V_{sift} obtenus ne sont pas strictement les mêmes que dans les évaluations des parties précédentes. Pour permettre l'étude comparative, nous avons également recalculé les scores de référence sur les modalités séparées ainsi que les paramètres de combinaison obtenus par la recherche exhaustive.

Tout d'abord, nous présenterons les résultats qui serviront de référence, puis les résultats obtenus en utilisant l'analyse discriminante pour apprendre les paramètres de combinaison. Toutes ces expérimentations seront présentées en considérant le critère MAP utilisé comme référence par les organisateurs des compétitions.

4.4.3.1 Résultats de référence

Tout comme pour les expérimentations précédentes, les résultats de référence correspondent à ceux obtenus en considérant chaque modalité séparément. Nous présenterons également les résultats obtenus par la recherche exhaustive en combinant deux modalités. Ainsi, nous calculerons grâce à l'approche globale, les paramètres α_g^{2008} (respectivement α_g^{2009}) sur la collection 2008 (respectivement 2009) qui combinent l'information textuelle et une information visuelle parmi V_{mstd} et V_{sift}.

4.4.3.2 Apprentissage grâce à l'analyse discriminante

Pour apprendre les paramètres de combinaison à partir de l'analyse discriminante, nous utiliserons comme échantillon d'apprentissage les requêtes de la collection 2008 et les documents pertinents associés. Comme pour la recherche exhaustive, nous calculerons les paramètres de combinaison entre la modalité textuelle T et une des deux modalités visuelles V_{mstd} et V_{sift}. À partir des scores textuels et visuels de l'échantillon d'apprentissage, nous pouvons calculer les coefficients de la combinaison linéaire grâce à l'analyse discriminante. Pour intégrer ces coefficients dans notre modèle de représentation de documents, ils ont été normalisés de façon à ce que leur somme soit égale à 1. Ainsi, le coefficient normalisé associé à α_j pour la modalité $j \in \{T, V\}$ est égal à :

$$\frac{\alpha_j}{\sum_j \alpha_j} \tag{4.17}$$

Cette normalisation correspondant à la division par la somme des coefficients calculés, est constante et n'influence donc pas l'ordre des documents retrouvés par le système. La comparaison avec les résultats de référence permettra de conclure sur l'efficacité de la méthode analytique utilisée pour l'apprentissage par rapport à la recherche exhaustive.

4.4.3.3 Combinaison de toutes les modalités

L'approche analytique permet d'utiliser un nombre plus important de modalités pour la combinaison. À partir des trois vocabulaires T, V_{mstd} et V_{sift}, nous apprendrons les paramètres de combinaison pour chaque modalité sur la collection 2008. Après avoir

normalisés ces coefficients, nous les utiliserons sur les requêtes de la collection 2009. Nous pourrons ainsi comparer les résultats avec ceux obtenus précédemment.

4.4.4 Résultats

Tout d'abord, nous présenterons les résultats de référence obtenus pour chaque modalité appris sur 2008 et 2009 en effectuant une recherche exhaustive. Ensuite, nous les comparerons à ceux obtenus grâce à l'analyse discriminante permettant de combiner les différentes modalités.

4.4.4.1 Nouveaux résultats de référence

Les nouveaux résultats de référence obtenus sur la collection 2009 sont présentés dans la table 4.7. Rappelons que cette édition comporte 45 requêtes et que 1 622 documents sont jugés pertinents sur l'ensemble de la collection.

TAB. 4.7 – Résultats de référence obtenus sur la collection ImageCLEF 2009 en considérant la mesure MAP.

Expérimentations	MAP	Nombre de documents retournés	Nombre de documents pertinents retrouvés
T	0,1661	35 617	1 190
V_{mstd}	0,0071	45 000	117
V_{sift}	0,0083	45 000	175
T+V_{mstd} : $\alpha_g^{2008} : 0,034$	0,1791	45 000	1195
T+V_{sift} : $\alpha_g^{2008} : 0,077$	0,1813	45 000	1213
T+V_{mstd} : $\alpha_g^{2009} : 0,026$	0,1815	45 000	1211
T+V_{sift} : $\alpha_g^{2009} : 0,084$	0,1822	45 000	1208

Les résultats confirment ceux obtenus précédemment. La modalité textuelle permet de retrouver principalement les documents pertinents (1 190 sur 1 622) et un MAP de $0,1661$. Les modalités visuelles $mstd$ et $sift$ sont beaucoup moins efficaces et ne permettent de retrouver que 117 documents pour $mstd$ et 175 documents pour $sift$. Les résultats obtenus après la combinaison des informations textuelles et visuelles montrent que la recherche exhaustive sur la collection 2008 de la valeur du paramètre de combinaison, permet d'obtenir de très bons résultats relativement aux résultats idéaux calculés sur 2009. En effet, grâce à l'approche globale, les résultats appris sur 2008 permettent d'approcher les résultats optimaux calculés sur 2009 avec pour le descripteur $mstd$ un MAP de 0,1791 pour un MAP optimal de 0,1815 et pour le descripteur $sift$, un MAP de 0,1813 pour un MAP optimal de 0,1822. Tous ces résultats seront utilisés pour la comparaison des résultats de l'analyse discriminante.

4.4.4.2 Apprentissage grâce à l'analyse discriminante

L'ensemble des coefficients calculés pour deux modalités est présenté par la table 4.8. les coefficients de combinaison α diffèrent légèrement de ceux calculés avec la recherche

exhaustive avec une valeur de $0,019$ (respectivement $0,059$) pour la modalité $mstd$ (respectivement $sift$) par rapport à une valeur α_g^{2008} apprise de $0,034$ (respectivement $0,077$).

TAB. 4.8 – Coefficients de combinaison obtenus avec l'analyse discriminante en combinant une modalité textuelle et une modalité visuelle.

Modalités	Coefficient α	Coefficient normalisé α
T	0,068102	0,981270
V_{mstd}	0,001300	0,018730
T	0,073285	0,940902
V_{sift}	0,004603	0,059098

Les résultats obtenus pour les requêtes de 2009 en utilisant les paramètres de la combinaison linéaire calculés grâce à l'analyse discriminante sur les requêtes de 2008 (table 4.8) sont présentés dans la table 4.9.

TAB. 4.9 – Résultats obtenus sur la collection 2009 en utilisant les coefficients définis analytiquement pour deux modalités à partir des requêtes de 2008.

Expérimentations	MAP	Nombre de documents retournés	Nombre de documents pertinents retrouvés
T	0,1661	35 617	1 190
T et $V_{mstd} : \alpha_{V_{mstd}} : 0,018730$	0,1801	45 000	1 218
T et $V_{sift} : \alpha_{V_{sift}} : 0,059098$	0,1795	45 000	1 219

Bien que les valeurs des paramètres de combinaison diffèrent de celles de référence calculées, ces valeurs obtenues à partir de l'analyse discriminante permettent d'améliorer significativement le MAP. En effet, comme nous pouvons le voir sur cette table, la combinaison de la modalité textuelle à celle d'un descripteur visuel permet d'augmenter le MAP de $0,1661$ à $0,1801$ pour le descripteur $mstd$ et à $0,1795$ pour le descripteur $sift$. En plus de retrouver quelques documents supplémentaires, 28 avec $mstd$ et 29 avec $sift$, l'augmentation du MAP, par rapport à celui n'utilisant que la modalité T, indique que les documents pertinents retrouvés sont mieux classés. Par rapport aux résultats obtenus en déterminant le paramètre α_g^{2008} par une recherche exhaustive sur les valeurs possibles (table 4.7), les performances sont comparables étant donné que l'analyse discriminante permet d'obtenir un MAP de $0,1801$ pour le descripteur $mstd$ au lieu de $0,1791$, et un MAP de $0,1795$ au lieu de $0,1813$ pour le descripteur $sift$. Globalement les résultats obtenus sont très bons par rapport aux résultats idéaux calculés sur 2009 qui ont un MAP de $0,1815$ pour le descripteur $mstd$ et de $0,1822$ pour le descripteur $sift$. Notons que l'analyse discriminante permet de retrouver toujours plus de documents pertinents sur l'ensemble des résultats ; ce qui est plutôt positif étant donné que c'est le critère qui est considéré pour effectuer l'apprentissage.

La significativité de ces résultats a été contrôlée à l'aide de tests statistiques. Le test de Student apparié unilatéral a été réalisé sur les précisions moyennes des 45 requêtes de la collection 2009. Ce test montre que la probabilité critique (*p value*) est de $0,000470$ (respectivement $0,023290$) pour l'expérimentation combinant la modalité textuelle et *mstd* (respectivement $sift$) [Saporta, 2006], ce qui conduit en prenant un risque de 5% à refuser l'hypothèse d'égalité des moyennes. Les améliorations des résultats combinant une modalité textuelle et une modalité visuelle peuvent donc être considérées comme significatives. Les détails concernant le test de Student sont présentés dans l'annexe C.

4.4.4.3 Combinaison de toutes les modalités

Contrairement à l'approche qui consiste à rechercher exhaustivement les valeurs des paramètres de combinaison, l'analyse discriminante peut s'appliquer à un plus grand nombre de modalités. Ainsi, la table 4.10 montre les coefficients obtenus par l'analyse discriminante en considérant les trois modalités T, V_{mstd} et V_{sift}.

TAB. 4.10 – Coefficients de combinaison obtenus avec l'analyse discriminante pour les trois modalités T, V_{mstd} et V_{sift}.

Modalités	Coefficient α	Coefficient normalisé α
T	0,079162	0,941711
V_{mstd}	0,001163	0,013837
V_{sift}	0,003737	0,044451

La mesure MAP obtenue en combinant les différentes modalités T, V_{mstd} et V_{sift}, est de $0,1875$ et le nombre de documents pertinents retrouvés est de 1 235 ce qui correspond à 45 nouveaux documents. Ce résultat est bien meilleur que ceux obtenus à l'aide de deux descripteurs, même en considérant les paramètres idéaux. L'analyse discriminante est donc une méthode efficace qui permet d'améliorer les résultats en utilisant les trois modalités. Par rapport aux résultats obtenus uniquement avec l'information textuelle, l'amélioration est de nouveau significative avec une probabilité critique de $0,000995$ (annexe C).

TAB. 4.11 – Résultats sur la collection 2009 en utilisant les coefficients définis analytiquement à partir des trois modalités T, V_{mstd} et V_{sift}.

Expérimentations	MAP	Nombre de documents retournés	Nombre de documents pertinents retrouvés
T	0,1661	35 617	1 190
T, $V_{mstd} : \alpha_{V_{mstd}} : 0,013837$ et $V_{sift} : \alpha_{V_{sift}} : 0,044451$	0,1875	45 000	1 235

Dans ce chapitre, nous avons défini un modèle de représentation de documents multimédias permettant de combiner linéairement les informations textuelle et visuelle contenues dans les documents.

Ce modèle a été validé dans le cadre de la recherche d'information en participant aux compétitions ImageCLEF 2008 et 2009 [Moulin *et al.*, 2008, Moulin *et al.*, 2009]. À partir de nos premières expérimentations, nous avons pu constater que la modalité visuelle apportait de l'information utile. En effet, en ajoutant cette dernière à la modalité textuelle, il nous a été possible d'améliorer les résultats obtenus uniquement avec l'information textuelle. De plus, à partir d'un ensemble de requêtes utilisé comme une base d'apprentissage, nous avons montré qu'en effectuant une recherche exhaustive sur les valeurs du paramètre de combinaison, il était possible de trouver une valeur globale permettant de combiner efficacement les informations textuelle et visuelle. Pour la mesure MAP, servant de référence au classement à la compétition, nous avons ainsi amélioré les résultats obtenus à partir de l'information textuelle seule de 14%.

Nous avons cherché dans un second temps à savoir si l'utilisation d'un paramètre de combinaison spécifique à chaque requête permettrait d'améliorer la recherche. En supposant qu'il est possible de catégoriser les requêtes à l'avance, et ainsi de paramétrer l'importance des modalités textuelle et visuelle, nous avons montré qu'il était possible, pour la mesure MAP, d'améliorer les résultats textuels de 30%. Par rapport à l'amélioration de 14% obtenue par l'approche globale, connaître au préalable la classe d'une requête permet d'améliorer fortement les résultats.

Comme il semble difficile de préalablement catégoriser les requêtes, nous nous sommes focalisés sur la première approche. Cependant, la complexité de la recherche exhaustive est exponentielle en fonction du nombre de modalités et pose des problèmes de temps de calcul. De ce fait, nous avons proposé une approche analytique permettant d'apprendre les paramètres de combinaison quand ce nombre augmentait. À partir de l'analyse discriminante, nous avons montré expérimentalement qu'il était possible d'améliorer significativement les résultats textuels en combinant deux et trois modalités différentes.

Conclusion et perspectives

La représentation des documents multimédias composés d'images, de texte ou d'une combinaison des deux est un problème essentiel pour le traitement automatique de l'information multimédia. En effet, pour répondre aux besoins des utilisateurs, il est nécessaire de mettre en place des modèles capables de représenter et de combiner les informations textuelle et visuelle contenues dans les documents. Dans cette thèse, nous avons étendu le modèle vectoriel qui permet de représenter efficacement et simplement les documents en sacs de mots, pour combiner les différentes informations dans deux tâches : la catégorisation et la recherche d'information multimédia. Après avoir résumé nos contributions, nous proposons différentes perspectives.

Contributions

Dans le contexte de la catégorisation de documents textuels, nous nous sommes intéressés au problème de la réduction de la taille du vocabulaire [Largeron et Moulin, 2010, Largeron *et al.*, 2011]. Dans ce contexte, nous avons introduit un nouveau critère $CCDE$, inspiré de l'approche $tf.idf$, permettant de sélectionner les mots qui sont les plus discriminants pour les catégories. D'une part, ce critère met en avant les mots qui apparaissent souvent dans les documents d'une catégorie, à l'image de la fréquence d'un terme dans un document. D'autre part, il considère également les mots qui sont principalement présents dans une seule catégorie, comme la fréquence inverse d'un terme dans une collection. Si dans certains cas la réduction du vocabulaire peut conduire à de meilleures performances [Géry *et al.*, 2009], elle permet surtout une réduction importante de la taille du vocabulaire sans dégrader les résultats de la classification. Nous avons comparé $CCDE$ à des critères classiques de sélection et montré qu'il conduisait à une réduction plus importante que les autres critères tout en produisant de meilleurs résultats.

Ce cadre a ensuite été étendu au cas multilabel où un document peut être associé à plusieurs catégories. Nous avons introduit un nouveau critère $MCut$ qui est orienté document et peut donc être utilisé pour un document particulier. Ce critère ne conserve que les catégories qui ont obtenu des scores bien plus élevés que ceux des autres catégories. Il a fait l'objet d'une première évaluation avec la participation à INEX XMLMining 2009 où il a permis d'obtenir en moyenne les meilleurs résultats sur l'ensemble des cri-

tères d'évaluation considérés par la compétition [Largeron *et al.*, 2010]. Nous l'avons également comparé à d'autres méthodes classiques de sélection de catégories ainsi qu'à l'approche binaire. Nous avons poursuivi notre étude sur la collection RCV1, ce qui a permis de confirmer nos précédents résultats et de montrer que *MCut* est une bonne alternative aux critères classiques de sélection de catégories. Il est en effet simple à mettre en place et ne nécessite aucun paramétrage contrairement aux autres critères.

Après avoir considéré des documents textuels, nous avons étudié la représentation des images et la création de différents vocabulaires visuels. Nous avons étudié deux approches pour détecter les régions à décrire dans les images qui sont l'échantillonnage dense et la détection multi-échelle. Ces deux approches sont simples à mettre en place et conduisent à de très bons résultats. Le seul paramètre pour chaque approche est le nombre de régions à détecter. Pour décrire ces régions, nous avons considéré deux descripteurs. L'information sur la couleur est capturée par le descripteur *mstd*. Nous l'avons évalué en participant à la compétition ImageCLEF 2008 [Moulin *et al.*, 2008]. Les informations de texture et de forme sont quant à elles représentées grâce au descripteur classique $sift$. Dans le but de mettre en avant, pour une image donnée, les mots visuels qui apparaissent souvent dans cette image et peu dans les autres, nous avons étudié différentes pondérations inspirées des approches $tf.idf$ utilisées pour les documents textuels. Cette étude a été réalisée grâce à la collection SimpliCITY dans le contexte d'une catégorisation d'images [Moulin *et al.*, 2010a]. Nous avons ainsi montré que la pondération $tf.idf_{w_2}$ permettait d'améliorer les résultats de la catégorisation. À partir de différents vocabulaires visuels, il est possible d'exploiter différentes informations comme la couleur, la texture, la forme, etc. Pour avoir une meilleure représentation des images, nous nous sommes intéressés à la combinaison de ces informations et avons montré qu'en combinant les mots visuels au niveau des vocabulaires, il était possible d'améliorer les résultats obtenus à partir d'une seule modalité visuelle. De plus, cette combinaison a fourni de meilleurs résultats que les approches classiques de fusion précoce ou tardive.

Enfin, nous nous sommes intéressés à la représentation des documents multimédias composés d'une image et de texte, dans le contexte de la recherche d'information. En exploitant les résultats que nous avons obtenus précédemment, nous avons construit un vocabulaire spécifique pour chaque modalité. Nous avons évalué notre modèle sur une collection de documents extraits de Wikipedia en participant aux compétitions ImageCLEF 2008 et 2009 [Moulin *et al.*, 2008, Moulin *et al.*, 2009]. Nous nous sommes intéressés à la combinaison linéaire des résultats obtenus séparément à partir des informations textuelle et visuelle. Nous avons montré qu'il était possible d'apprendre le paramètre de combinaison en effectuant une recherche exhaustive à partir des requêtes et des documents pertinents associés sur une collection d'apprentissage [Moulin *et al.*, 2010b, Moulin *et al.*, 2010c]. En effet, en combinant l'information textuelle et une modalité visuelle, nous avons pu améliorer les résultats obtenus à partir de l'information textuelle seule. Cependant, cette recherche étant exponentielle en fonction du nombre de modalités, il n'est donc pas envisageable de l'appliquer quand le nombre de modalités augmente.

De ce fait, nous avons mis en place une approche analytique permettant d'apprendre les paramètres de combinaison pour un plus grand nombre de modalités. L'apprentissage est réalisé par analyse discriminante en considérant la recherche d'information

comme un problème de catégorisation où le but est de séparer les documents pertinents des non pertinents pour un ensemble de requêtes d'apprentissage. Nous avons montré expérimentalement qu'en utilisant l'information textuelle et les informations visuelles obtenues à partir des descripteurs $mstd$ et $sift$, il est possible d'améliorer significativement les résultats de la recherche sur un nouvel ensemble de requêtes.

Perspectives

Deux tâches ont été considérées dans cette thèse : la catégorisation et la recherche d'information. Une première perspective serait d'étendre notre modèle de combinaison des modalités au contexte de la catégorisation de documents multimédias. Dans ce cas, les paramètres de combinaison des vocabulaires pourraient être appris à l'aide de l'analyse discriminante sur un échantillon d'apprentissage, en tenant compte de toutes les classes. Ceci pose cependant le problème de l'évaluation d'un tel modèle à partir d'une collection. En effet, dans le cadre de la catégorisation de documents multimédias, nous n'avons pas trouvé de collections de documents composés d'images et de texte, à l'exception des collections où le texte correspond à des mots clés associés aux images ; ces mots clés ne constituent pas une information textuelle suffisamment riche et réaliste. Une première idée serait de construire une collection à partir des documents de la collection ImageCLEF et de les classer selon les différentes requêtes. Ainsi, chaque requête correspondrait à une catégorie. Une deuxième idée pourrait être de se servir des catégories associées aux articles de Wikipedia pour créer une nouvelle collection à l'image des collections INEX XMLMining 2008 et 2009 [Denoyer et Gallinari, 2006]. Enfin, une dernière solution consisterait à utiliser des documents vidéos comme ceux présents dans la compétition TRECVID et d'en extraire les informations textuelle et visuelle [Over *et al.*, 2010] : la partie textuelle pourrait correspondre, par exemple, à la reconnaissance vocale effectuée sur la bande son.

Une seconde perspective s'intéresse à la définition et la création d'autres descripteurs textuels et visuels. Le vocabulaire textuel que nous avons considéré est construit à partir des mots extraits des documents. Nous pourrions également en définir un autre en créant des concepts obtenus, par exemple, avec l'analyse sémantique latente [Deerwester *et al.*, 1990]. L'information textuelle des documents pourrait alors être représentée à l'aide de plusieurs vocabulaires textuels.

Une piste à exploiter pour la création du vocabulaire visuel concerne la détection des points d'intérêt et leur description. Tout d'abord, nous pourrions favoriser certaines parties fixes, comme le centre, le haut de l'image, etc. pour tenir compte de l'information spatiale [Lazebnik *et al.*, 2006]. En plus des descriptions $mstd$ et $sift$, il serait possible d'en envisager d'autres comme ceux du standard $MPEG7$ [Salembier et Smith, 2002] ou des descripteurs mieux adaptés à la prise en compte de la couleur [Song *et al.*, 2009]. Comme les mots visuels construits sont difficilement interprétables, il serait intéressant d'effectuer une étude similaire à celle réalisée pour la partie textuelle. Nous pourrions par exemple étudier l'influence de la réduction du vocabulaire visuel dans le contexte de la catégorisation d'images.

Une autre perspective intéressante pourrait être de considérer l'information structurelle. Pour les documents textuels au format XML, cette information est facilement accessible et peut être exploitée pour ajuster, par exemple, le poids $tf.idf$ des mots

textuels dans les documents [Géry *et al.*, 2009]. La représentation des images en sacs de mots ne tient pas compte de la structure, mais différentes approches essayent d'exploiter cette information. Il est par exemple possible de représenter les images grâce à des arbres [Boyer *et al.*, 2007] ou des graphes [Samuel *et al.*, 2010]. Ces informations structurelles pourraient par exemple permettre de définir de nouvelles pondérations des mots.

Dans le contexte de la recherche d'information, nos recherches se sont intéressées à la combinaison linéaire des résultats obtenus pour chaque modalité. Plutôt que de chercher à combiner les résultats des recherches, de nouvelles méthodes d'apprentissage automatique visent à apprendre à classer les documents en fonction de leur rang (*learning to rank*) [Burges *et al.*, 2005]. Une première étape consiste, à partir d'un échantillon d'apprentissage composé de différents résultats retournés par le système, à créer un modèle capable de réordonner plus efficacement les documents. Pour une nouvelle requête, le système retourne une liste de documents qui est ensuite réorganisée à l'aide du modèle appris précédemment. Dans le but d'évaluer et d'étudier la fusion des différentes modalités, il serait intéressant de les comparer sur une même collection.

Enfin, les documents que nous avons considérés dans notre recherche sont composés d'images ou de texte. Cependant, de façon générale, les documents multimédias peuvent contenir d'autres modalités. En effet, les fichiers audios et vidéos fournissent par exemple des informations auditives et temporelles. Il existe notamment des descripteurs spatio-temporels [Laptev et Lindeberg, 2003, Dollar *et al.*, 2005] qui pourraient servir à la création de vocabulaires spécifiques. À plus long terme, une perspective serait alors d'étendre le modèle proposé pour pouvoir exploiter les vidéos.

Annexes

Annexe A

Présentation des collections XML Mining 2008 et 2009

XML Mining s'inscrit dans le cadre de la compétition internationale INEX[1] (*INitiative for the Evaluation of XML retrieval*). XML Mining regroupe deux tâches qui sont la classification non-supervisée et la catégorisation de documents XML. Notre participation concerne la catégorisation de documents. Cette tâche est organisée par Ludovic Denoyer et Patrick Gallinari de l'université de Paris 6[2]. Nous avons participé aux éditions 2008 et 2009 de cette compétition [Géry *et al.*, 2009, Largeron *et al.*, 2010].

A.1 XML Mining 2008

Présentation de la collection XML Mining 2008

La collection XML Mining 2008 est composée de 114 366 documents XML extraits de Wikipedia à classer parmi 15 catégories [Denoyer et Gallinari, 2006, Denoyer et Gallinari, 2009]. Il s'agit d'une catégorisation multiclasse où chaque document appartient à une seule catégorie. Les 15 catégories associées au document sont : *reference, social institutions, sociology, sports, fiction, united states, categories by nationality, europe, tourism, politics by region, urban geography, americas, art genres, demographics, human behavior*. La collection fournit des informations sur les liens entre les différents documents XML, mais nous ne les avons pas utilisés dans la suite.

Notre participation à XML Mining 2008

Le but de notre participation à cette compétition était d'une part d'obtenir des résultats de référence pour de futures recherches qui pourraient considérer la structure des documents et d'autre part d'évaluer l'influence d'un critère de sélection permettant de réduire la taille du vocabulaire [Géry *et al.*, 2009].

[1] `http://www.inex.otago.ac.nz`
[2] `http://xmlmining.lip6.fr/`

TAB. A.1 – Taille des différents vocabulaires.

vocabulaire	taille du vocabulaire
T	77697
CC_{100}	1 051
CC_{10000}	75 181
CCE_{100}	909
CCE_{10000}	77 580

Critère de sélection

Pour cette participation, nous avons défini deux critères CC et CCE.

$CC(t_j, c_k)$ correspond à la couverture de classe pour le terme t_j dans la classe c_k et se définit par :

$$CC(t_j, c_k) = \frac{P(t_j|c_k)^2}{\sum_k P(t_j|c_k)} \tag{A.1}$$

où $P(t_j|c_k)$ correspond à la probabilité d'apparition du terme t_j en ne considérant que les documents qui appartiennent à la classe c_k.

$CCE(t_j, c_k)$ correspond à la couverture de classe en prenant en compte l'entropie du terme t_j. $CCE(t_j, c_k)$ est défini par :

$$CCE(t_j, c_k) = \alpha.P(t_j|c_k) + (1-\alpha).(\frac{E_{max} - E(t_j)}{E_{max}}) \tag{A.2}$$

où α est un paramètre qui permet de donner plus ou moins d'importance à l'entropie du terme t_j par rapport à la probabilité d'apparition du terme t_j dans la classe c_k et E_{max} et $E(t_j)$ correspondent respectivement à l'entropie maximale et l'entropie associée au mot t_j comme nous l'avons défini dans la partie 2.1.2.2.

Ces critères sont ensuite utilisés pour réduire le vocabulaire T obtenu après le traitement classique de création du vocabulaire, à savoir, la lemmatisation de Porter et la suppression des mots vides. La réduction s'effectue à l'aide d'une approche locale où pour chaque critère, les 100 et les 10 000 premiers mots qui obtiennent les scores les plus élevés pour chaque classe sont conservés. Les vocabulaires obtenus sont notés respectivement CC_{100} et CC_{10000} pour le critère CC et CCE_{100} et CCE_{10000} pour le critère CCE.

Soumissions

Nous avons réalisé 5 soumissions. La table A.1 résume la taille des vocabulaires obtenus pour chaque soumission. Le vocabulaire T correspond à la soumission qui servira de référence. Les autres vocabulaires ont pour but d'étudier l'influence de la réduction à l'aide des deux critères CC et CCE précédemment définis. Chaque catégorisation a été réalisée à l'aide de liblinear [Fan *et al.*, 2008].

Résultats

La table A.2 regroupe l'ensemble des résultats de tous les participants. Nos résultats correspondent à l'équipe : LaHC. Cette table A.2 montre que notre résultat de référence

TAB. A.2 – Présentation de l'ensemble des résultats de toutes les équipes pour XML Mining 2008.

rang	équipe	soumission	taux	documents
1	LaHC	expe_5.tf_idf_CC_10000.txt	78,76%	102 929
2	LaHC	expe_3.tf_idf_CCE_10000.txt	78,74%	102 929
3	LaHC	expe_1.tf_idf_TA.txt	78,73%	102 929
4	Vries	classification_text_and_links.txt	78,49%	102 929
5	boris	boris_inex.tfidf.sim.037.it3.txt	73,79%	102 929
6	boris	boris_inex.tfidf1.sim.0.38.3.txt	73,47%	102 929
7	boris	boris_inex.tfidf.sim.034.it2.txt	73,09%	102 929
8	LaHC	expe_4.tf_idf_CC_100.txt	72,30%	102 929
9	kaptein	kaptein_2008NBscoresv02.txt	69,80%	102 929
10	kaptein	kaptein_2008run.txt	69,78%	102 929
11	romero	romero_naive_bayes_links.txt	68,13%	102 929
12	LaHC	expe_2.tf_idf_CCE_100.txt	67,70%	102 929
13	romero	romero_naive_bayes.txt	67,67%	102 929
14	Vries	classification_links_only.txt	62,32%	102 929
15	Vries	classification_text_only.txt	24,44%	92 647

obtient le 3e meilleur résultat avec un taux de bien classé de 78,73%. Pour un score de référence, ce dernier est très élevé puisqu'il est plus élevé que les résultats obtenus par les autres participants. La réduction de la taille du vocabulaire nous a cependant permis d'augmenter légèrement le taux de bien classés avec 78,76% et 78,74% pour respectivement les deux critères CC et CCE en considérant 10 000 mots par classe.

Une sélection des 100 premiers mots par classe entraîne une réduction importante de la taille du vocabulaire de référence. En effet, pour les critères CC et CCE cela correspond respectivement à une réduction de 74% avec 1 051 mots dans le vocabulaire et de 85% avec 909 mots. Bien que la réduction de la taille du vocabulaire soit importante, le taux de bien classés reste correct avec un pourcentage de 72,30% pour CC et 67,70% pour CCE.

A.2 XML Mining 2009

Présentation de la collection XML Mining 2009

La collection XML Mining 2009 est composée de 54 889 documents XML extraits de Wikipedia [Denoyer et Gallinari, 2006, Nayak *et al.*, 2010]. Chaque document appartient à au moins une catégorie, mais plusieurs labels peuvent être associés aux documents. Cette catégorisation multilabel possède un ensemble de 39 catégories correspondant chacune à un portail de Wikipedia : *Portal :American Civil War, Portal :Anarchism, Portal :Architecture, Portal :Astronomy, Portal :Aviation, Portal :Baseball Portal :Bible, Portal :Biography, Portal :Business and economics, Portal :Catholicism Portal :Chemistry, Portal :Chess, Portal :Christianity, Portal :Comics, Portal :Cricket Portal :Food, Portal :Formula One, Portal :Geography, Portal :History, Portal :Horror Portal :Literature, Portal :Medicine, Portal :Music, Portal :Nautical, Portal :Pharmacy*

and Pharmacology, Portal :Philosophy, Portal :Physics, Portal :Politics Portal :Pornography, Portal :Religion, Portal :Saints, Portal :Science, Portal :Space Portal :Trains, Portal :Tropical cyclones, Portal :Video games, Portal :War Portal :Weather, Portal :World War I. L'échantillon d'apprentissage est composé de 20% de la collection, soit 11 028 documents. Parmi ces documents 9 809 ne sont associés qu'à une seule classe. Comme pour l'édition précédente, les liens entre les différents documents XML n'ont pas été utilisés dans la suite.

Notre participation à XML Mining 2009

Le but de notre participation à l'édition 2009 a été d'une part de tester la réduction du vocabulaire en utilisant un critère de sélection, et d'autre part de proposer une nouvelle méthode de sélection du nombre de catégories à associer à un nouveau document.

Critère de sélection

Le critère de sélection utilisé pour XML Mining 2009 correspond au critère CCD. Il correspond à la différence de couverture de classe (CC) utilisé pour l'édition 2008 de XML Mining 2008 :

$$CCD(t_j, c_k) = CC(t_j, c_k) - CC(t_j, \bar{c_k}) \tag{A.3}$$

où $CC(t_j, \bar{c_k})$ est défini par :

$$CC(t_j, \bar{c_k}) = \frac{P(t_j|\bar{c_k})^2}{\sum_k P(t_j|c_k)} \tag{A.4}$$

avec $P(t_j|\bar{c_k}) = \frac{|\{d_i \notin c_k | t_j \in d_i\}|}{|\mathcal{D}| - |\{d_i \in c_k\}|}$

$CCD(t_j, c_k)$ peut alors se simplifier par :

$$\begin{aligned} CCD(t_j, c_k) &= \frac{P(t_j|c_k)^2 - P(t_j|\bar{c_k})^2}{P(t_j|c_k) + P(t_j|\bar{c_k})} && \text{(A.5)} \\ &= P(t_j|c_k) - P(t_j|\bar{c_k}) && \text{(A.6)} \end{aligned}$$

Sélection du nombre de catégories

Les méthodes pour sélectionner le nombre de catégories à associer à un document correspondent aux méthodes $MCut$, $PCut$ et $RCut$ définies dans les parties 2.2.3, 2.2.2.2 et 2.2.2.1. Les méthodes $PCut$ et $RCut$ sont utilisées pour les comparer à notre méthode $MCut$.

Soumissions

Pour notre participation, 8 soumissions résumées dans la table A.3 ont été effectuées. Lors de l'évaluation, l'ordre des étiquettes associées aux documents était pris en compte. Nos soumissions correspondent pour chaque document à un ensemble d'étiquettes. Un ensemble peut correspondre à un seul document, *singleton*, à un ensemble *trié*, ou *non trié*.

TAB. A.3 – Présentation des 8 soumissions à XML Mining 2009.

soumission	méthode	voc.	stratégie de sélection	ensemble des étiquettes
LaHC_1_baseline	*multiclasse*	T	-	*singleton*
LaHC_2_bin	*binaire*	T	-	*non trié*
LaHC_3_bin_lk	*binaire*	T_k	-	*non trié*
LaHC_4_bin_lk_1000	*binaire*	$T_{k_{1000}}$	-	*non trié*
LaHC_5_max	*multiclasse*	T	$MCut$	*trié*
LaHC_6_pcut	*multiclasse*	T	$PCut$	*trié*
LaHC_7_rcut_1	*multiclasse*	T	$RCut_1$	*trié*
LaHC_8_rcut_2	*multiclasse*	T	$RCut_2$	*trié*

TAB. A.4 – Présentation des différents vocabulaires utilisés.

vocabulaire		définition
T	=	$\{t_j \in d_i \| d_i \in D\}$
T_k	=	$\{t_j \in d_i \| d_i \in D \wedge d_i \in c_k\}$
$T_{k_{1000}}$	=	$\{t_j \in T_k \wedge CCD_j^k \geqslant CCD_{1000}^k\}$

La première soumission correspond à notre référence et considère le problème comme de la catégorisation multiclasse (*multiclasse*). Une seule étiquette est associée aux documents (*singleton*). Les soumissions 2, 3 et 4 utilisent l'approche binaire pour considérer le problème multilabel comme expliqué dans la partie 2.2.1.2. La différence entre ces soumissions correspond au vocabulaire utilisé pour représenter les documents. Les vocabulaires pour les soumissions 2, 3 et 4 correspondent respectivement à T, T_k et $T_{k_{1000}}$ comme expliqué par la table A.4. L'approche binaire ne permet pas de retourner un ensemble trié des classes (*non trié*). Les quatre dernières soumissions utilisent des méthodes de sélection du nombre d'étiquettes à partir des résultats d'une classification *multiclasse*. Les soumissions 5, 6, 7 et 8 correspondent respectivement à l'utilisation des méthodes $MCut$, $PCut$, $RCut_1$ et $RCut_2$. L'ordre des classes retourné par le classifieur est conservé (*trié*).

Résultats

Tous les résultats de toutes les équipes sont regroupés dans la table A.5. Les critères classiques ACC, ROC et PRF correspondent respectivement à la précision, l'aire sous la courbe ROC [Fawcett, 2006] et à la F-mesure. Pour ces critères, les valeurs micro et macro sont calculées. Le critère MAP est quant à lui utilisé pour évaluer la pertinence de l'ordre des étiquettes retournées. Pour ordonner les soumissions, nous avons calculé un dernier critère correspondant à la moyenne de tous les critères considérés par la compétition.

La table A.5 montre que les résultats de notre soumission de référence sont plutôt bons par rapport à ceux obtenus par les autres participants. Comme il n'y a qu'une seule catégorie affectée par document, les critères PRF, ROC et MAP ne sont pas très bons. En revanche, le nombre d'erreurs réalisées est faible et explique le bon résultat

TAB. A.5 – Présentation de l'ensemble des résultats de toutes les équipes pour XML Mining 2009 triés par la moyenne sur l'ensemble des critères.

équipe	soumission	macro ACC	micro ACC	macro ROC	micro ROC	macro PRF	micro PRF	MAP	moyenne
LaHC	LaHC_5_max	0,968	0,952	0,936	0,934	0,549	0,578	0,788	0,820
LaHC	LaHC_7_rcut_1	0,974	0,962	0,938	0,935	0,531	0,564	0,788	0,817
LaHC	LaHC_6_pcut	0,973	0,961	0,927	0,925	0,548	0,563	0,748	0,816
LaHC	LaHC_8_rcut_2	0,959	0,933	0,903	0,906	0,515	0,528	0,788	0,791
xerox	nxQ.3.merge.tfidf	0,975	0,964	0,753	0,767	0,579	0,605	0,678	0,774
xerox	netxQ.4.plus.tfidf	0,974	0,963	0,748	0,765	0,571	0,600	0,679	0,770
xerox	nxQ.4.merge	0,974	0,963	0,748	0,765	0,571	0,600	0,679	0,770
peking	3	0,963	0,948	0,842	0,850	0,480	0,519	0,702	0,767
peking	2	0,963	0,948	0,842	0,850	0,480	0,518	0,702	0,767
peking	1	0,962	0,947	0,842	0,850	0,478	0,516	0,702	0,766
granada	nb_with_links_sub	0,952	0,934	0,802	0,820	0,500	0,530	0,642	0,756
granada	nb_sub	0,951	0,933	0,803	0,820	0,496	0,527	0,641	0,755
LaHC	LaHC_1_baseline	0,974	0,962	0,721	0,743	0,531	0,564	0,685	0,749
granada	orgate_with_links_sub	0,848	0,819	0,928	0,927	0,316	0,360	0,725	0,700
LaHC	LaHC_3_bin_Ik	0,967	0,950	0,619	0,629	0,334	0,355	0,407	0,642
granada	orgate_sub	0,754	0,678	0,925	0,922	0,253	0,263	0,730	0,632
LaHC	LaHC_2_bin	0,971	0,958	0,600	0,613	0,289	0,323	0,393	0,626
LaHC	LaHC_4_bin_Ik_1000	0,965	0,947	0,585	0,596	0,252	0,279	0,330	0,604
wollongon	bpts2.f1.r3	0,913	0,892	0,625	0,619	0,192	0,218	0,138	0,576
wollongon	bptsext.f1a.r3	0,131	0,160	0,558	0,561	0,072	0,103	0,100	0,264
wollongon	bptsext.f1.r3	0,038	0,055	0,632	0,623	0,071	0,102	0,208	0,253
wollongon	bpts2.f1a.r3	0,038	0,055	0,598	0,599	0,071	0,102	0,125	0,244
wollongon	bptsext.map.r3	0,137	0,141	0,506	0,513	0,065	0,096	0,192	0,243
wollongon	bpts2.map.r3	0,115	0,123	0,511	0,510	0,070	0,101	0,129	0,238

obtenu pour le critère ACC.

Les résultats des soumissions 2, 3 et 4 qui ont pour but d'évaluer l'utilisation d'un vocabulaire spécifique en effectuant une catégorisation binaire, sont tous inférieurs au résultat de référence. Parmi ces soumissions, nous pouvons remarquer que l'utilisation d'un vocabulaire adapté par catégorie offre de meilleurs résultats que l'utilisation d'un vocabulaire global (soumission 2 et 3), mais que la réduction du vocabulaire à l'aide du critère CCD n'a pas permis d'améliorer les résultats (soumission 4).

Les méthodes de sélection du nombre de catégories correspondant aux soumissions $5, 6, 7$ et 8 ont permis d'obtenir en moyenne les meilleurs résultats de la compétition. Le critère ACC favorisant les méthodes qui font peu d'erreur, la stratégie $RCut_1$ obtient le meilleur résultat. La soumission 8 correspondant à la méthode $RCut_2$ obtient en moyenne les moins bons résultats car elle associe à chaque document un nombre de deux catégories ce qui n'est pas du tout réaliste étant donnée la moyenne du nombre de catégories par document qui est de 1,46. Les résultats de la stratégie $PCut$ sont globalement moins bons que ceux de $RCut_1$ mais meilleurs que ceux de $RCut_2$. En moyenne, la stratégie $MCut$ de la soumission 5 obtient les meilleurs résultats. Elle permet en effet d'obtenir les meilleurs résultats pour la moyenne micro et macro de la mesure F1 (PRF) en ne dégradant que légèrement ceux du critère ROC (macro : $0,936$, micro $0,934$ par rapport à macro : $0,938$, micro $0,935$ de $RCut$).

Annexe B

Présentation des collections ImageCLEF 2008 et 2009

ImageCLEF [1] est une compétition annuelle qui regroupe plusieurs challenges comme l'annotation et la recherche d'images médicales, l'annotation et la recherche de photos, la recherche de documents multimédias etc. Nous avons participé aux éditions 2008 et 2009 [Moulin *et al.*, 2008, Moulin *et al.*, 2009] de cette compétition qui s'inscrit dans le cadre de la campagne CLEF (*Cross Language Evaluation Forum*) [2]. En ce qui concerne nos différentes participations, nous nous sommes intéressés au problème de la recherche de documents multimédias dans une collection de 151 519 documents.

B.1 ImageCLEF 2008

Pour notre première participation à la compétition ImageCLEF 2008, nous avons proposé un modèle simple qui permet de représenter les modalités textuelles et visuelles à l'aide de sacs de mots [Moulin *et al.*, 2008]. Après avoir détaillé la collection ImageCLEF 2008, nous présentons les paramètres de notre modèle ainsi que les soumissions effectuées lors de cette édition 2008.

Présentation de la compétition ImageCLEF 2008

La collection ImageCLEF 2008 fait suite aux compétitions INEX Multimedia de 2006 et 2007 [Tsikrika et Kludas, 2008]. Les documents sont extraits de Wikipedia et possèdent tous une image avec un texte plus ou moins long qui décrit l'image mais qui peut également fournir des informations sur les droits qui lui sont associés ou sur la personne qui a fourni l'image.

Le but de cette compétition est de retourner pour un ensemble de requêtes, la liste des documents de la collection qui répondent le mieux à chaque requête. Ces requêtes sont composées d'un ou plusieurs mots textuels et peuvent également possédées des images ou des concepts. Les concepts que nous n'avons pas considéré dans la suite, sont extraits des 101 concepts de la collection MediaMill [Snoek *et al.*, 2006]. En moyenne

[1] `http://www.imageclef.org`
[2] `http://www.clef-campaign.org`

les requêtes possèdent 2,64 mots et sur l'ensemble des 75 requêtes, 43 possèdent au moins une image.

Notre participation à ImageCLEF 2008

Pour notre première participation à ImageCLEF, nous avons proposé un modèle qui ne pondérait pas les différentes modalités. Nous avons effectué 6 soumissions qui exploitent l'information textuelle et/ou l'information visuelle. Après avoir présenté le modèle général et les soumissions effectuées, nous confronterons les résultats obtenus à ceux des autres participants.

Modèle utilisé

Le modèle général utilisé est présenté par la figure B.1. Il correspond à notre tout premier modèle et diffère donc légèrement de celui présenté dans ce manuscrit. Il correspond à un modèle basé sur une approche par sac de mots. Les modalités textuelles et visuelles sont dans un premier temps représentées séparément à l'aide de sac de mots. Nous avons ensuite utilisé un modèle vectoriel pour représenter les documents à l'aide d'un vecteur pondéré de mots textuels et visuels.

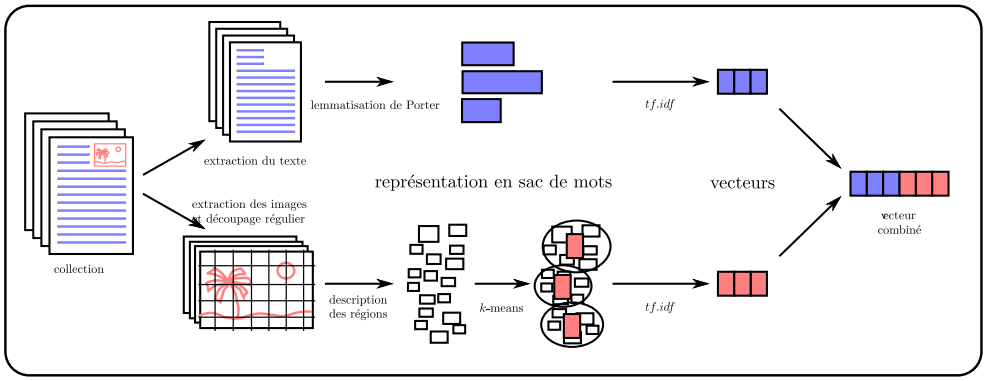


FIG. B.1 – Présentation du modèle

Pour la modalité textuelle, le vocabulaire $T = \{t_1, \ldots, t_j, \ldots, t_{|T|}\}$ est obtenu à partir des mots textuels présents dans les documents, lemmatisés à l'aide de l'algorithme de Porter. Le document d_i est ensuite représenté par un vecteur de poids $\vec{d_i^T} = (w_{i,1}^T, \ldots, w_{i,j}^T, \ldots, w_{i,|T|}^T)$ où les poids $w_{i,j}^T = tf_{i,j}.idf_j$ sont calculés avec la pondération tf.idf okapi où

$$tf_{i,j} = \frac{(k_1+1).n_{i,j}}{n_{i,j} + k_1(1 - b + b\frac{|d_i|}{d_{avg}})} \tag{B.1}$$

et

$$idf_j = \ln \frac{|\mathcal{D}| + df_j + 0,5}{df_j + 0,5} \tag{B.2}$$

avec $n_{i,j}$ le nombre d'occurrence du terme t_j dans le document d_i, $|d_i| = \sum_j n_{i,j}$ la taille du document, d_{avg} est la taille moyenne des documents de $\mathcal{D}$, df_j le nombre de documents de $\mathcal{D}$ dans lequel le terme t_j apparaît et b et k_1 deux constantes.

TAB. B.1 – Calcul des scores et leurs paramètres.

score	paramètre $tf_{i,j}$	paramètre $tf_{k,j}$
$score(q_k, d_i) = \sum_{t_j \in q_k} tf_{i,j} idf_j tf_{k,j}$	$k_1 = 1.2$ $b = 0.75$	$k_1 = 7$ $b = 0$

Les images sont également représentées à l'aide d'un sac de mots à partir de notre vocabulaire V défini grâce au descripteur $mstd$ calculé sur une grille régulière 16×16 comme présenté précédemment. La description est ensuite obtenue en calculant la moyenne et l'écart-type des valeurs $\frac{R}{R+G+B}$, $\frac{G}{R+G+B}$ et $\frac{R+G+B}{3*255}$ où R, G et B sont les valeurs des composants rouge, vert et bleu des imagettes. Le document d_i est alors représenté à l'aide d'un vecteur de poids $\vec{d_i^V} = (w_{i,1}^V, \ldots, w_{i,j}^V, \ldots, w_{i,|V|}^V)$ où les poids sont calculés de la même façon que la modalité textuelle.

Les scores textuels et visuels finaux pour une requête q_k donnée sont ensuite calculés par

$$score^T(q_k^T, d_i^T) = \sum_{t_j \in q_k^T} tf_{i,j} idf_j tf_{k,j} \quad \text{(B.3)}$$

$$score^V(q_k^V, d_i^V) = \sum_{v_j \in q_k^V} tf_{i,j} idf_j tf_{k,j} \quad \text{(B.4)}$$

où q_k^T et q_k^V correspondent respectivement aux mots textuels et visuels de la requête q_k. Les mots de la requête q_k sont pondérés par $tf_{k,j}$. La table B.1 résume le calcul du score et les valeurs des constantes b et k_1.

Soumissions

À partir du modèle présenté, nous avons effectué un ensemble de 6 soumissions résumées dans la figure B.2 et la table B.2.

Les 6 soumissions sont dénommées entre LaHC_run01 et LaHC_run06. Nous avons effectué deux types de soumission, automatique ($auto$) et manuel (man). Notre but est d'étudier le choix des mots visuels obtenus avec le descripteur $mstd$, l'utilisation conjointe des modalités textuelles et visuelles et l'apport de l'utilisation de l'information visuelle.

Notre première soumission (LaHC_run01) correspond à la référence et utilise uniquement l'information textuelle. Les résultats obtenus à partir de cette soumission sont notés R_1. Cette soumission est automatique et n'utilise ni retour de pertinence, ni d'extension de requête.

Toutes les autres soumissions utilisent les deux informations textuelles et visuelles des documents. Elles sont obtenues à partir de deux requêtes successives : la première (Q_1) utilise uniquement l'information textuelle et correspond à la soumission de référence (LaHC_run01) alors que la seconde (Q_2) exploite soit l'information visuelle seule, soit les deux informations textuelles et visuelles. Les requêtes ne possédant pas toute une image de référence, nous avons construit l'information visuelle à partir des résultats de référence (R_1) de façon manuelle ou automatique. Les soumissions automatiques

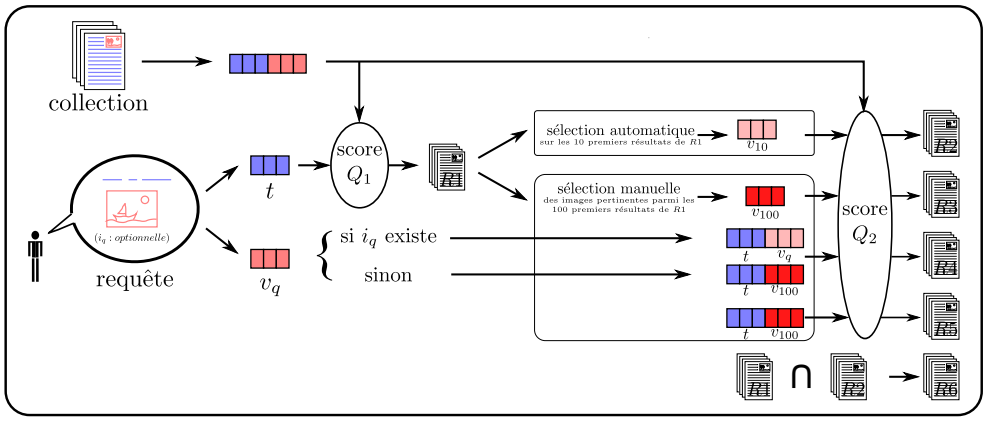


FIG. B.2 – Présentation des soumissions : run 01 est le run textuel qui fourni les résultats de référence $R1$. Les runs 02 à 06 consistent en deux requêtes successives : la première requête Q_1 correspond à la requête textuelle alors que la seconde Q_2 est une requête visuelle ou textuelle et visuelle. Les mots visuels sont sélectionnés à partir des images requêtes, ou à partir d'une sélection automatique ou manuelle des images issues des résultats de référence.

utilisent les mots visuels présents dans les dix premiers documents retrouvés par la soumission de référence (v_{10}) ; l'idée étant que ces premiers documents sont pertinents. La construction de la requête visuelle s'est également faite manuellement en sélectionnant les images pertinentes choisies manuellement parmi les 100 premiers documents retournés par la requête de référence (v_{100}). Il n'y a pas de limite dans le nombre d'images pertinentes sélectionnées. Que ce soit automatiquement ou manuellement, la seconde requête (Q_2) est créée à partir des mots visuels extraits des images sélectionnées à l'exception de la soumission LaHC_run04 qui utilise les images fournies par l'utilisateur (i_q) quand elles existent.

Les soumissions LaHC_run02 et LaHC_run06 sont automatiques. La soumission LaHC_run02 utilise uniquement l'information visuelle pour la seconde requête et les résultats sont notés R_2. La soumission LaHC_run06 correspond à l'intersection des résultats obtenus par la soumission de référence et ceux obtenus automatiquement avec la soumission LaHC_run02. Ces résultats seront notés R_6. Cette soumission est intéressante pour étudier l'apport de l'information visuelle car elle permet de mettre en avant les documents qui n'ont pu être retrouvés qu'avec cette information visuelle.

Les soumissions LaHC_run03, LaHC_run04 et LaHC_run05 sont manuelles. La seconde requête de la soumission LaHC_run03 n'utilise que l'information visuelle des images sélectionnées manuellement alors que les autres soumissions LaHC_run04 et LaHC_run05 utilisent également l'information textuelle. LaHC_run05 considère la requête textuelle initiale et les images sélectionnées manuellement (t+v_{100}). La soumission LaHC_run04 se différencie de la précédente pour les requêtes où une image requête (i_q) est fournie. Dans ce cas, la requête finale est composée de la requête textuelle initiale ainsi que des mots visuels extraits des images requêtes.

TAB. B.2 – Présentation des expérimentations : run 01 est l'expérimentation de référence n'exploitant que l'information textuelle. run 02 à run 06 consistent de deux requêtes successives : la première Q_1 correspond à la requête textuelle alors que la seconde Q_2 est visuelle ou bien textuelle et visuelle. Le mots visuels sont sélectionnées soit automatiquement, soit manuellement.

nom soumission	Q_1	type de la soumission	usage de Q_1	Q_2	résultats
LaHC_run01	t	*auto*	-	-	R_1
LaHC_run02	t	*auto*	v_{10}	v_{10}	R_2
LaHC_run03	t	*man*	v_{100}	v_{100}	R_3
LaHC_run04	t	*auto* *man*	- v_{100}	$t+\begin{cases} v_q & \text{si } i_q \text{ existe} \\ v_{100} & \text{sinon} \end{cases}$	R_4
LaHC_run05	t	*man*	v_{100}	$t+v_{100}$	R_5
LaHC_run06	-	*auto*	-	-	$R_6 = R_1 \cap R_2$

avec
- t : requête textuelle seule ;
- R_i : résultats de la soumission run_i ;
- i_q : requête image ;
- v_{10} : sélection automatique des 10 premiers résultats de R_1 ;
- v_{100} : sélection manuelle des documents pertinents parmi les 100 premiers résultats de R_1 ;
- v_q : mots visuels extraits des requêtes images.

Résultats

Pour l'édition 2008, un ensemble de 77 soumissions on été proposées avec 12 participants. Les résultats sont disponibles sur l'internet `http://www.imageclef.org/2008/wikimm-results` et sont résumés par les tables B.4 et B.3.

Recherche textuelle. Notre résultat de référence est comparé à ceux des autres participants grâce à la table B.3. Seuls les résultats qui exploitent l'information textuelle sont présentés car beaucoup de participants ne se sont pas servis de l'information visuelle et les méthodes utilisées sont très différentes et ne permettent pas une comparaison directe des résultats qui utilisent l'information visuelle. Notre soumission LaHC_run01 correspond à notre meilleur résultat et se classe 22e par rapport aux 77 soumissions. En ne considérant que les soumissions textuelles, sans extension de requête ni retour de pertinence, notre équipe se situe en 3e position (table B.3). Cette soumission obtient donc de très bons résultats en retrouvant 3 467 documents sur les 5 593 à retrouver.

Recherche visuelle. Notre plus mauvais résultat est obtenu par la soumission automatique LaHC_run02 qui exploite uniquement l'information visuelle alors que notre soumission automatique LaHC_run03 arrive en 2e position. Les mauvais résultats de la soumission LaHC_run02 ne sont pas vraiment étonnant car cette expérimentation utilise les dix premiers documents retrouvés par notre résultat de référence. Comme nous pouvons le voir dans la table B.4, la précision pour les dix premiers document (P@10) est de 0,3680 ce qui signifie que seulement 37% des images utilisées pour la requête automatique de la soumission LaHC_run02 sont pertinentes. La soumission

TAB. B.3 – Résultats textuels de référence de l'ensemble des participants.

Rang	Participant	Run	MAP	P@10
11	sztaki	bp_acad_textonly_qe	0,2546	0,3720
13	cwi	cwi_lm_txt	0,2528	0,3427
22	curien	LaHC_run01	0,2453	0,3680
29	ualicante	IRn	0,2178	0,3200
30	chemnitz	cut-txt-a	0,2166	0,3440
44	imperial	SimpleText	0,1918	0,3240
48	irit	SigRunText	0,1652	0,2880
50	upeking	zhou1	0,1525	0,2573
52	ugeneva	unige_text_baseline	0,1440	0,2053
56	upmc-lip6	TFUSION_TFIDF_LM	0,1193	0,2160
70	utoulon	LSIS_TXT_method1	0,0399	0,0467

TAB. B.4 – Résumé de nos résultats.

Rang	Run	MAP	P@10	Nombre de documents retrouvés	Nombre de documents pertinents retrouvés
22	LaHC_run01	0,2453	0,3680	54638	3467
57	LaHC_run03	0,1174	0,2613	74986	1004
58	LaHC_run05	0,1161	0,2600	74986	987
61	LaHC_run06	0,1067	0,3280	1741	429
65	LaHC_run04	0,0760	0,1813	74986	822
69	LaHC_run02	0,0577	0,1613	74989	643

LaHC_run03 permet de valider l'utilisation de notre descripteur visuel mstd. En effet, le nombre d'images retrouvées est bien plus important que le nombre d'images sélectionnées manuellement. Par exemple pour la requête *fleur bleue*, le nombre d'images sélectionnée manuellement est de 9. Ces 9 images permettent de retrouver 42 images pertinentes sur les 71 à retrouver. En considérant les deux soumissions LaHC_run02 et LaHC_run03, nous arrivons à retrouver 1 222 images uniquement avec l'information visuelle.

Amélioration grâce à l'information visuelle. Même si l'information visuelle conduit à des résultats significativement moins bons que l'information textuelle, il ne faut pas la négliger. En effet, la comparaison des résultats obtenus par les soumissions LaHC_run01, LaHC_run02 et LaHC_run06 montre que 214 nouveaux documents ont pu être retrouver grâce à l'information visuelle seule. En effet, comme le montre la table B.4, 643 documents pertinents sont retrouvés avec la soumission LaHC_run02 et 429 avec la soumission LaHC_run06. Comme cette dernière soumission correspond à l'intersection entre les résultats de référence et les résultats de la soumission LaHC_run02, cela signifie que $643 - 429 = 214$ documents ont été retrouvés uniquement grâce à l'information visuelle.

Combinaison des informations textuelle et visuelle. La combinaison des dif-

férentes informations ne nous a pas permis d'améliorer les résultats de référence comme nous l'espérions. Si nous avions pu effectivement retrouvés l'ensemble des documents pertinents obtenus grâce aux soumissions LaHC_run01 et LaHC_run03, nous aurions pu trouver 3 818 documents pertinents. Par rapport aux soumissions LaHC_run03 et LaHC_run04, nous pouvons conclure que l'ajout de deux ou trois mots textuels aux différents mots visuels dans la requête ne permet pas d'améliorer les résultats. 92% des documents sont communs à ces deux soumissions ce qui confirme la trop grande importance accorder aux mots visuels.

L'utilisation d'une seule image requête. Les soumissions LaHC_run04 et LaHC_run05 montrent que l'utilisation d'une seule image requête semble donner de moins bons résultats que lorsque plusieurs images sont sélectionnées. En effet, les résultats obtenus par la soumission LaHC_run04 sont systématiquement moins bons que ceux de la soumissions LaHC_run05.

B.2 ImageCLEF 2009

Pour notre seconde participation, notre but a été de reprendre notre modèle proposé dans l'édition 2008 et de le compléter en ajoutant des informations textuelles extraites des articles originaux de Wikipedia, des informations visuelles en calculant un second descripteur et enfin en combinant linéairement ces différentes informations. Après avoir présenté les nouveautés de l'édition 2009, nous détaillerons notre participation.

Présentation de la collection ImageCLEF 2009

Pour l'édition 2009, les documents de la collection sont les mêmes que ceux de 2008. La différence avec l'édition précédente concerne les requêtes. En 2009, 45 requêtes sont proposées chacune accompagnée au moins d'une image requête. En moyenne, les requêtes sont composées de 2,93 mots textuels et de 1,84 images.

Notre participation à ImageCLEF 2009

Pour notre participation à ImageCLEF 2009, nous avons effectué un ensemble de 13 soumissions. Nous présenterons dans un premier temps le modèle utilisé pour cette édition et dans un second temps nous détaillerons les soumissions que nous avons effectué ainsi que les résultats obtenus.

Modèle utilisé

Le modèle utilisé pour l'édition 2009 est principalement issu des expérimentations réalisées pour l'édition 2008.

Concernant l'information textuelle, nous avons dans un premier temps utilisé la même approche que l'édition 2008 à savoir, le texte des utilisateurs de Wikipedia qui ont fourni les images (*metadata*). Comme nous l'avons vu précédemment, ce texte peut ne pas être approprié et nous avons donc exploité de l'information textuelle supplémentaire en extrayant le texte original des articles de Wikipedia utilisant les images. Pour ce faire, nous utilisons un paramètre qui permet de sélectionner un certain nombre de caractères autour de l'image. Dans la suite, nous avons ainsi choisi 50 et 100 caractères autour des images (50 ou 100 *car*). Le texte ainsi extrait est ensuite ajouté au texte d'origine (*metadata*).

Pour cette édition, nous avons utilisé différents vocabulaires visuels. Le premier *mstd* correspond à celui de l'édition précédente. *mstd* est obtenu après un découpage régulier des images 16×16. La description est ensuite obtenue en calculant la moyenne et l'écart-type des valeurs $\frac{R}{R+G+B}$, $\frac{G}{R+G+B}$ et $\frac{R+G+B}{3*255}$ où R, G et B sont les valeurs des composants rouge, vert et bleu des imagettes. Les deux autres vocabulaire utilisent le descripteur $SIFT$ pour différents découpages. $sift_1$ est calculé après avoir détecté préalablement des régions d'intérêt avec le détecteur $MSER$ alors que $sift_2$ utilise le même découpage régulier que *mstd*. Pour ces trois descriptions, nous avons utilisé l'algorithme k-means pour obtenir des vocabulaires de 10 000 mots visuels.

Contrairement à l'édition précédente, nous avons effectué deux calculs pour les scores de pertinence. Le premier $score^1$ utilise les mêmes pondérations que pour l'édition 2008 et est obtenu par

$$score^1(q_k, d_i) = \sum_{t_j \in q_k} tf_{i,j} idf_j tf_{k,j} \tag{B.5}$$

où $tf_{i,j}$ et $tf_{k,j}$ sont calculés par

$$tf_{i,j} = \frac{(k_1 + 1).n_{i,j}}{n_{i,j} + k_1(1 - b + b\frac{|d_i|}{d_{avg}})} \tag{B.6}$$

et idf_j par

$$idf_j = \ln \frac{|\mathcal{D}| + df_j + 0,5}{df_j + 0,5} \tag{B.7}$$

avec $n_{i,j}$ le nombre d'occurrence du terme t_j dans le document d_i, $|d_i| = \sum_j n_{i,j}$ la taille du document, d_{avg} est la taille moyenne des documents de $\mathcal{D}$, df_j le nombre de documents de $\mathcal{D}$ dans lequel le terme t_j apparaît et b et k_1 deux constantes.

Le second $score^2$ est obtenu par

$$score^2(q_k, d_i) = \sum_{t_j \in q_k} tf_{i,j} idf_j tf_{k,j} idf_j \tag{B.8}$$

avec la même formule du tf que le $score^1$, mais avec idf_j calculé par

$$idf_j = \frac{|\mathcal{D}| + 1}{df_j + 0,5} \tag{B.9}$$

afin d'éviter des valeurs d'*idf* négatives. Un résumé de ces deux scores est présenté par la table B.5.

TAB. B.5 – Calcul des scores et leurs paramètres.

score	paramètres pour $tf_{i,j}$	paramètres pour $tf_{k,j}$
$score^1(q_k, d_i) = \sum_{t_j \in q_k} tf_{i,j} idf_j tf_{k,j}$	$k_1 = 1.2$ $b = 0.75$	$k_1 = 7$ $b = 0$
$score^2(q_k, d_i) = \sum_{t_j \in q_k} tf_{i,j} idf_j tf_{k,j} idf_j$	$k_1 = 1$ $b = 0.5$	$k_1 = 1$ $b = 0$

Afin d'utiliser toute l'information disponible, nous avons utilisé deux méthodes pour fusionner les résultats obtenus par les modalités séparées. La première correspond simplement à l'intersection des résultats obtenus par l'information textuelle et ceux obtenus avec l'information visuelle (IN).

La seconde illustrée par la figure B.3 effectue une combinaison linéaire des scores calculés à partir des informations textuelles et visuelles (CL) :

$$score_\alpha(q_k, d_i) = \alpha score^V(q_k^V, d_i^V) + (1-\alpha) score^T(q_k^T, d_i^T) \tag{B.10}$$

Le paramètre α permet de donner plus ou moins d'importance à l'information visuelle. Pour calculer ce paramètre, nous avons ainsi utilisé les requêtes de la collection 2008 comme échantillon d'apprentissage.

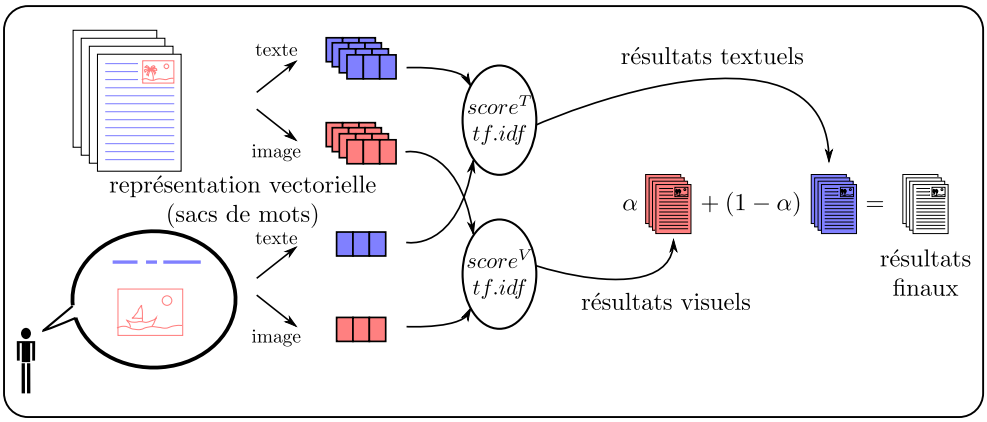


FIG. B.3 – Présentation de la combinaison linéaire des résultats textuels et visuels

Soumissions

Les soumissions que nous avons effectué pour l'édition 2009 d'ImageCLEF sont automatiques et résumées dans la table B.6. Nous avons proposé deux soumissions de référence ($LaHC\_1$ et $LaHC\_2$). La première correspond à la même référence proposé pour l'édition 2008 où seule l'information textuelle est utilisée avec le calcul du $score^1$. La seconde utilise le $score^2$ sera ensuite utilisé pour toutes les autres soumissions.

Deux autres soumissions exploitant uniquement l'information textuelle sont proposées avec $LaHC\_9$ et $LaHC\_10$ qui considèrent également le texte englobant avec respectivement 100 et 50 caractères autour des images des articles originaux de Wikipedia (100 *car* et 50 *car*).

Toutes les autres soumissions combinent les informations textuelles et visuelles. Les soumissions $LaHC\_3$, $LaHC\_4$ et $LaHC\_8$ sont obtenues après avoir effectué l'intersection des résultats entre ceux de la soumission de référence $LaHC\_2$, et ceux obtenus pour l'information visuelle avec respectivement les méthodes $mstd$, $sift_1$ et $sift_2$. Les dernières soumissions $LaHC\_5$, $LaHC\_6$, $LaHC\_7$ et $LaHC\_13$ correspondent à la combinaison linéaire des résultats de référence calculés grâce à l'information textuelle ($LaHC\_2$) et les résultats obtenus à partir de l'information visuelle ($mstd$, $sift_1$ et $sift_2$). $LaHC\_5$ et $LaHC\_6$ différent par leur paramètre de combinaison où celui de $LaHC\_5$ n'a pas été appris sur l'ensemble de la collection 2008 mais sur un sous-

TAB. B.6 – Paramètres de nos soumissions.

run	score	texte	image	combinaison
$LaHC\_1$	$score^1$	*metadata*	-	-
$LaHC\_2$	$score^2$	*metadata*	-	-
$LaHC\_3$	$score^2$	*metadata*	*mstd*	*intersection* (IN)
$LaHC\_4$	$score^2$	*metadata*	$sift_1$	*intersection* (IN)
$LaHC\_5$	$score^2$	*metadata*	*mstd*	α= 0.015 (CL)
$LaHC\_6$	$score^2$	*metadata*	*mstd*	α= 0.025 (CL)
$LaHC\_7$	$score^2$	*metadata*	$sift_1$	α= 0.012 (CL)
$LaHC\_8$	$score^2$	*metadata*	$sift_2$	*intersection* (IN)
$LaHC\_9$	$score^2$	100 *car*	-	-
$LaHC\_10$	$score^2$	50 *car*	-	-
$LaHC\_11$	$score^2$	100 *car*	*mstd*	α= 0.025 (CL)
$LaHC\_12$	$score^2$	50 *car*	*mstd*	α= 0.025 (CL)
$LaHC\_13$	$score^2$	*metadata*	$sift_2$	α= 0.084 (CL)

ensemble. Enfin les soumissions $LaHC\_11$ et $LaHC\_12$ combinent les résultats des soumissions $LaHC\_9$ et $LaHC\_10$ avec les résultats obtenus par le descripteur *mstd*.

Résultats

Les résultats de la compétition sont disponibles sur l'internet[1]. Un résumé des meilleures soumissions de l'ensemble des 8 participants est présenté par la table B.7. Comme le montre cette table, notre meilleure soumission nous classe deuxième sur l'ensemble des participants. La table B.8 regroupe tous les résultats de nos soumissions.

TAB. B.7 – Meilleures soumissions de chaque participant.

rang	participant	soumission	MAP
1	deuceng	deuwiki2009_205	0.2397
5	lach	$LaHC\_11$	0.2178
7	cea	cealateblock	0.2051
17	ualicante	Alicante-MMLCA	0.1878
26	dcu	DCUTFIDF-DBpediaMetadata-QE	0.1752
29	sztaki	bp_acad_txt4_min_txtimg	0.1699
41	sinai	sinai_NTWn_T	0.1566
55	iiit_hyd	iiithr1	0.0186

Recherche textuelle. Notre meilleur résultat n'utilisant que le texte est celui qui exploite l'information textuelle supplémentaire extraite des articles originaux pour une fenêtre de 100 caractères ($LaHC\_9$). Le MAP obtenu est alors de $0,1890$ ce qui correspond à une augmentation de 13% par rapport à la soumission de référence $LaHC\_2$ qui a un MAP de $0,1667$. L'ajout de texte ne permet pas vraiment de

[1] `http://www.imageclef.org/2009/wikiMM-results`

retrouver de nouveaux documents, 13 pour 100 caractères ($LaHC\_9$) et 6 pour 50 caractères ($LaHC\_10$). L'amélioration du MAP indique que l'ajout de texte extrait des articles originaux de Wikipedia permet de mieux classer les documents. Ceci est conforté par la meilleure qualité des cinq premiers documents retournés qui augmente de 33% avec $P@5$ qui passe de 0,2978 à 0,3956 comme le montre la table B.8

TAB. B.8 – Présentation de nos résultats.

rang	run	fusion	texte	image	MAP	P@5	Nombre de documents retournés	Nombre de documents pertinents retrouvés
5	$LaHC\_11$	CL	100 *car*	*mstd*	0,2178	0,3956	44993	1213
6	$LaHC\_12$	CL	50 *car*	*mstd*	0,2148	0,3956	44993	1218
14	$LaHC\_13$	CL	*metadata*	$sift_2$	0,1903	0,3333	44993	1212
15	$LaHC\_9$	-	100 *car*	-	0,1890	0,3600	38004	1205
16	$LaHC\_10$	-	50 *car*	-	0,1880	0,3422	37041	1198
20	$LaHC\_6$	CL	*metadata*	*mstd*	0,1845	0,3067	44993	1208
21	$LaHC\_7$	CL	*metadata*	$sift_1$	0,1807	0,3511	44995	1200
24	$LaHC\_5$	CL	*metadata*	*mstd*	0,1792	0,2978	44993	1213
33	$LaHC\_2$	-	*metadata*	-	0,1667	0,2978	35611	1192
44	$LaHC\_1$	-	*metadata*	-	0,1432	0,2622	35611	1164
52	$LaHC\_8$	IN	*metadata*	$sift_2$	0,0365	0,1867	619	142
53	$LaHC\_3$	IN	*metadata*	*mstd*	0,0338	0,2089	574	76
54	$LaHC\_4$	IN	*metadata*	$sift_1$	0,0321	0,1556	637	120

Intersection des recherches textuelles et visuelles. Les trois moins bons résultats sont obtenus par les soumissions qui consistaient à ne conserver que les résultats obtenus par le texte et par un descripteur visuel. Ce constat est vrai si nous considérons le critère MAP, mais si nous considérons la précision globale, ces trois soumissions obtiennent les meilleures performances avec environ un document pertinent sur six retrouvés. Si le but de l'utilisateur est de retrouver principalement des documents pertinents, ces soumissions répondent le mieux au problème.

Amélioration grâce à l'information visuelle. Comme nous pouvons le voir sur la table B.8, les soumissions qui combinent linéairement les informations textuelles et visuelles obtiennent systématiquement de meilleurs résultats que ceux n'exploitant qu'une seule modalité. La meilleure amélioration de 14% est obtenue pour la soumission $LaHC\_11$ qui utilise le descripteur $sift$ après avoir effectué un découpage régulier. Globalement $sift_2$ ($MAP : 0,1903$) est meilleur que *mstd* ($MAP : 0,1845$) qui est meilleur que $sift_1$ ($MAP : 0,1807$). Comme pour le texte englobant, même si l'information visuelle permet de trouver plus de documents pertinents, elle permet avant tout de mieux les classer.

Texte englobant et information visuelle. La figure B.4 illustre les améliorations obtenues pour les soumissions exploitant le texte englobant, le descripteur *mstd* et la combinaison des deux. Comme nous pouvons le voir sur la figure, la combinaison du texte englobant avec le descripteur *mstd* permet d'améliorer le critère MAP de 30% de 0,1667 à 0,2178. D'après la table B.8, le descripteur *mstd* n'est pas le meilleur de

nos descripteurs visuel qui est le descripteur $sift_2$ et nous pouvons de ce fait espérer obtenir encore de meilleurs résultats en combinant ce descripteur au texte englobant.

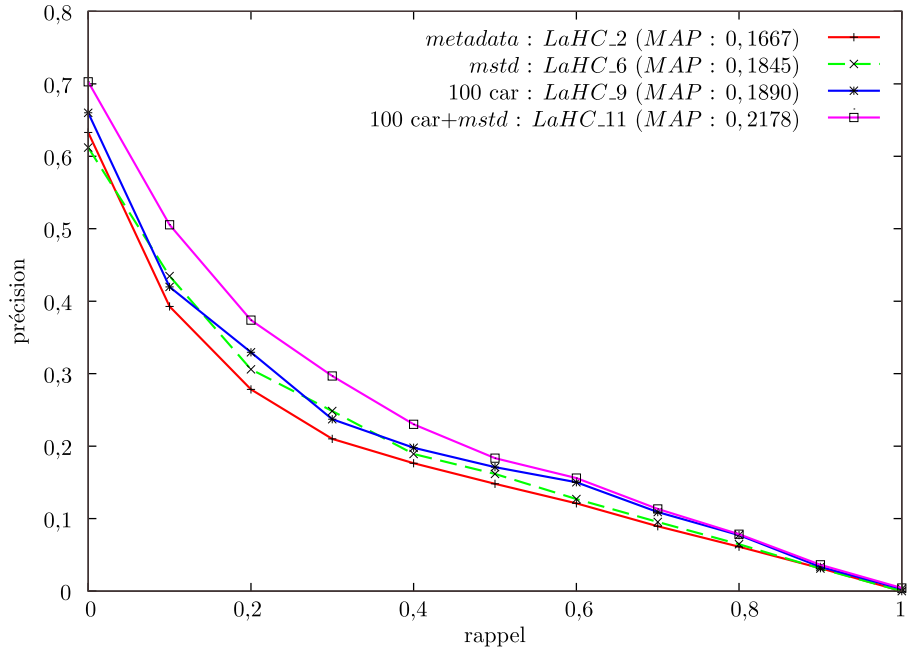


FIG. B.4 – Comparaison de différentes expérimentations par rapport à celle de référence n'exploitant que l'information textuelle.

Annexe C

Test de Student

Pour un même ensemble de requêtes Q, deux expérimentations ont été réalisées en utilisant les informations textuelles et visuelles pour la première, en apprenant le paramètre de combinaison à l'aide de l'analyse discriminante, et uniquement l'information textuelle pour la seconde. Le test de Student apparié unilatéral a permis de comparer les moyennes MAP obtenues pour ces deux expérimentations. Le but est alors de tester si les moyennes ont varié entre les deux expérimentations de façon significative.

$\mathcal{H}_0$ correspond à l'hypothèse où les moyennes sont égales entre les deux expérimentations. Tester $\mathcal{H}_0$ consiste à former les différences $AP_k^1 - AP_k^1 = d_k$ où AP_k^1 (respectivement AP_k^2) correspond à la précision moyenne de la première (respectivement deuxième) expérimentation et à faire le test de Student sur la moyenne des d_k [Student, 1908, Saporta, 2006] :

$$t = \frac{\mu_d}{\sigma_d}\sqrt{|Q| - 1} \tag{C.1}$$

où μ_d correspond à la moyenne des différences d_k et σ_d à l'écart-type. t est la valeur du test de Student à $|Q| - 1$ degrés de liberté. À partir de t et de $|Q| - 1$, il est possible de calculer la probabilité critique p (p *value*) correspondant à la probabilité de commettre une erreur en rejetant l'hypothèse $\mathcal{H}_0$.

La table C.1 regroupe l'ensemble des précisions moyennes pour la modalité textuelle seule T, et les combinaisons grâce à l'approche analytique pour les modalités combinant T et V_{mstd}, T et V_{sift} puis T, V_{mstd} et V_{sift} obtenues sur l'ensemble des 45 requêtes de la collection 2009 numérotées de 76 à 120.

En comparant l'expérimentation exploitant l'information textuelle seule et les expérimentations combinant les différentes modalités, les valeurs de p obtenues correspondent respectivement à $0,0004701$, $0,0232912$ et $0,0009946$ pour les combinaisons T et V_{mstd}, T et V_{sift} puis T, V_{mstd} et V_{sift} ce qui conduit en prenant un risque de 5% à refuser l'hypothèse $\mathcal{H}_0$. Les améliorations sont donc significatives.

TAB. C.1 – Précisions moyennes AP_k obtenues pour différentes expérimentations exploitant l'information textuelle seule T ou combinant cette dernière aux informations visuelles V_{mstd} et V_{sift} sur les 45 requêtes de la collection ImageCLEF 2009.

q_k	AP_k de T	AP_k de T+V_{mstd}	AP_k de T+V_{sift}	AP_k de T+V_{mstd} + V_{sift}
76	0,1451	0,1776	0,1697	0,2104
77	0,2375	0,3320	0,2995	0,3305
78	0,0922	0,0719	0,0825	0,0837
79	0,1638	0,1708	0,1839	0,1869
80	0,4684	0,5247	0,4689	0,5034
81	0,1013	0,1281	0,0895	0,1200
82	0,0767	0,0779	0,0801	0,0806
83	0,1342	0,1296	0,1145	0,1164
84	0,0609	0,0574	0,0615	0,0623
85	0,1352	0,1193	0,1326	0,1147
86	0,1170	0,1493	0,1438	0,1519
87	0,0813	0,0951	0,0904	0,1039
88	0,3543	0,3870	0,3983	0,3960
89	0,0646	0,0635	0,0545	0,0635
90	0,1961	0,2356	0,2215	0,2499
91	0,2160	0,2160	0,1865	0,1865
92	0,1055	0,1355	0,1323	0,1293
93	0,3058	0,3296	0,2557	0,3114
94	0,2467	0,2165	0,2144	0,2111
95	0,1306	0,1592	0,1126	0,1314
96	0,1191	0,1167	0,1171	0,1163
97	0,0393	0,0338	0,0321	0,0325
98	0,0086	0,0107	0,0098	0,0112
99	0,1706	0,1927	0,2059	0,2128
100	0,0746	0,0730	0,0725	0,0703
101	0,1032	0,0909	0,1262	0,1143
102	0,1767	0,1725	0,0865	0,0863
103	0,3293	0,3612	0,3645	0,3742
104	0,0912	0,1122	0,0797	0,0830
105	0,0494	0,0522	0,0510	0,0532
106	0,1918	0,2370	0,2644	0,3200
107	0,1465	0,2174	0,1802	0,2192
108	0,4315	0,4287	0,4131	0,4130
109	0,0985	0,1028	0,0988	0,1010
110	0,2097	0,2063	0,2684	0,2475
111	0,1707	0,1605	0,1929	0,1804
112	0,4908	0,4927	0,5131	0,5103
113	0,1280	0,1458	0,1405	0,1510
114	0,2228	0,2432	0,3028	0,2993
115	0,0797	0,0840	0,2369	0,2104
116	0,1667	0,1462	0,1908	0,1716
117	0,0404	0,0507	0,0566	0,0613
118	0,2298	0,2663	0,2750	0,2983
119	0,1181	0,1300	0,1521	0,1515
120	0,1551	0,1990	0,1541	0,2054
MAP :	0,1661	0,1801	0,1795	0,1875

Bibliographie

[Abe, 2010] Abe, S. (2010). *Support vector machines for pattern classification.* Springer-Verlag New York Inc.

[Adami *et al.*, 2003] Adami, G., Avesani, P. et Sona, D. (2003). Clustering documents in a web directory. *Dans WIDM'03 : 5th ACM international workshop on Web Information and Data Management*, pages 66–73.

[Agarwal *et al.*, 2004] Agarwal, S., Awan, A. et Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11): pages 1475–1490.

[Albatal *et al.*, 2010] Albatal, R., Mulhem, P. et Chiaramella, Y. (2010). Phrases Visuelles pour l'annotation automatique d'images. *Dans CORIA'10 : 7e COnférence en Recherche d'Information et Applications*, pages 3–18.

[Altman, 1968] Altman, E. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23(4): pages 589–609.

[Ayache *et al.*, 2007] Ayache, S., Quénot, G. et Gensel, J. (2007). Classifier fusion for SVM-based multimedia semantic indexing. *Dans ECIR'07 : 29th European Conference on Information Retrieval*, pages 494–504.

[Bahler et Navarro, 2000] Bahler, D. et Navarro, L. (2000). Methods for combining heterogeneous sets of classifiers. *Dans AAAI'00 : 17th national conference on artificial intelligence, workshop on new research problems for machine learning.*

[Bardos et Zhu, 1997] Bardos, M. et Zhu, W. (1997). Comparaison de l'analyse discriminante linéaire et de réseaux de neurones. Application à la détection de défaillances d'entreprises. *Revue de Statistique Appliquée*, 45(4): pages 65–92.

[Barnard *et al.*, 2003] Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D. et Jordan, M. (2003). Matching words and pictures. *The Journal of Machine Learning Research*, 3: pages 1107–1135.

[Bartell *et al.*, 1994] Bartell, B., Cottrell, G. et Belew, R. (1994). Automatic combination of multiple ranked retrieval systems. *Dans SIGIR'94 : 17th annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 173–181.

[Beitzel *et al.*, 2003] Beitzel, S., Frieder, O., Jensen, E., Grossman, D., Chowdhury, A. et Goharian, N. (2003). Disproving the fusion hypothesis : an analysis of

data fusion via effective information retrieval strategies. *Dans SAC'03 : 18th ACM Symposium on Applied Computing*, pages 823–827.

[Bentley, 1975] Bentley, J. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9): pages 509–517.

[Beucher et Lantuejoul, 1979] Beucher, S. et Lantuejoul, C. (1979). Use of watersheds in contour detection. *Dans ICIP'79 : IEEE International Conference on Image Processing : Real-time Edge and Motion Detection/Estimation.*

[Bicego *et al.*, 2006] Bicego, M., Lagorio, A., Grosso, E. et Tistarelli, M. (2006). On the use of SIFT features for face authentication. *Dans CVPRW'06 : IEEE conference on Computer Vision and Pattern Recognition Workshop*, pages 35–41.

[Bisgin, 2007] Bisgin, H. (2007). Parallel clustering algorithms with application to climatology. Mémoire de master, Informatics Institute, Istanbul Technical University, Turkey.

[Boole, 1854] Boole, G. (1854). *An investigation of the laws of thought.* Project Gutenberg. `http://www.gutenberg.org/ebooks/15114`.

[Boser *et al.*, 1992] Boser, B., Guyon, I. et Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Dans COLT'92 : 5th annual ACM workshop on COmputational Learning Theory*, pages 144–152.

[Boughorbel *et al.*, 2002] Boughorbel, S., Boujemaa, N. et Vertan, C. (2002). Histogram-based color signatures for image indexing. *Dans IPMU'02 : Information Processing and Management of Uncertainty in knowledge-based systems*, pages 977–984.

[Bouveyron *et al.*, 2005] Bouveyron, C., Girard, S. et Schmid, C. (2005). Analyse discriminante de haute dimension. Rapport technique 5470, INRIA.

[Boyer *et al.*, 2007] Boyer, L., Habrard, A. et Sebban, M. (2007). Learning metrics between tree structured data : application to image recognition. *Dans ECML'07 : 18th European Conference on Machine Learning*, pages 54–66.

[Brants, 2000] Brants, T. (2000). TnT : a statistical part-of-speech tagger. *Dans ANLP'00 : 6th ACL conference on Applied Natural Language Processing*, pages 224–231.

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2): pages 123–140.

[Brill, 1992] Brill, E. (1992). A simple rule-based part of speech tagger. *Dans ANLP'92 : 3rd ACL conference on Applied Natural Language Processing*, pages 152–155.

[Burges, 1998] Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2: pages 121–167.

[Burges *et al.*, 2005] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N. et Hullender, G. (2005). Learning to rank using gradient descent. *Dans ICML'05 : 22nd ACM International Conference on Machine learning*, pages 89–96.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *Readings in Computer Vision : Issues, Problems, Principles, and Paradigms*, 184: pages 679–698.

[Cao *et al.*, 2010] Cao, Y., Wang, C., Li, Z., Zhang, L. et Zhang, L. (2010). Spatial-bag-of-features. *Dans CVPR'10 : 23rd IEEE conference on Computer Vision and Pattern Recognition.*

[Caropreso *et al.*, 2001] Caropreso, M., Matwin, S. et Sebastiani, F. (2001). A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. *Dans Text Databases and Document Management : Theory and Practice*, pages 78–102. Idea Group Publishing.

[Cavnar et Trenkle, 1994] Cavnar, W. et Trenkle, J. (1994). N-gram-based text categorization. *Dans SDAIR'94 : 3rd annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.

[Chang et Lin, 2001] Chang, C.-C. et Lin, C.-J. (2001). LIBSVM : a library for support vector machines. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Chen *et al.*, 2004] Chen, L., Lu, G. et Zhang, D. (2004). Effects of different gabor filter parameters on image retrieval by texture. *Dans MMM'04 : 10th international MultiMedia Modelling conference*, pages 273–278.

[Chen et Goodman, 1999] Chen, S. et Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4): pages 359–393.

[Clare et King, 2001] Clare, A. et King, R. (2001). Knowledge discovery in multi-label phenotype data. *Dans PKDD'01 : 5th european conference on Principles of Data mining and Knowledge Discovery*, pages 42–53.

[Cleverdon, 1991] Cleverdon, C. (1991). The significance of the Cranfield tests on index languages. *Dans SIGIR'91 : 14th annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 3–12.

[Comaniciu et Meer, 2002] Comaniciu, D. et Meer, P. (2002). Mean shift : a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5): pages 603–619.

[Cortes et Vapnik, 1995] Cortes, C. et Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3): pages 273–297.

[Csurka *et al.*, 2004] Csurka, G., Dance, C., Fan, L., Willamowski, J. et Bray, C. (2004). Visual categorization with bags of keypoints. *Dans ECCV'04 : 8th European Conference on Computer Vision : workshop on Statistical Learning in Computer Vision*, pages 59–74.

[Cunningham, 2007] Cunningham, P. (2007). Dimension reduction. Rapport technique UCD-CSI-2007-7, University of Dublin.

[Deerwester *et al.*, 1990] Deerwester, S., Dumais, S., Landauer, T., Furnas, G. et Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6): pages 391–407.

[Demartines et Hérault, 1995] Demartines, P. et Hérault, J. (1995). CCA : curvilinear component analysis. *Dans GRETSI'95 : 15e colloque du Groupe d'Etudes du Traitement du Signal et des Images.*

[Dempster, 1967] Dempster, A. (1967). Upper and lower probabilities generated by a random closed interval. *The Annals of Mathematical Statistics*, 39(3): pages 957–966.

[Denoyer et Gallinari, 2006] Denoyer, L. et Gallinari, P. (2006). The Wikipedia XML corpus. *ACM Special Interest Group on Information Retrieval Forum*, 40(1): pages 64–69.

[Denoyer et Gallinari, 2009] Denoyer, L. et Gallinari, P. (2009). Overview of the INEX 2008 XML mining track. *Dans INEX'09 : 8th international workshop of the INitiative for the Evaluation of XML retrieval*, pages 401–411.

[Deriche et Giraudon, 1990] Deriche, R. et Giraudon, G. (1990). Accurate corner detection : An analytical study. *Dans ICCV'90 : 3rd International Conference on Computer Vision*, pages 66–70.

[Derrode et Ghorbel, 2001] Derrode, S. et Ghorbel, F. (2001). Robust and efficient Fourier-Mellin transform approximations for gray-level image reconstruction and complete invariant description. *Computer Vision and Image Understanding*, 83(1): pages 57–78.

[Diday, 1971] Diday, E. (1971). Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques. *Revue de Statistique Appliquée*, 19(2): pages 19–33.

[Dollar *et al.*, 2005] Dollar, P., Rabaud, V., Cottrell, G. et Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. *Dans VSPETS'05 : 2nd joint IEEE international workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72.

[Domingos et Pazzani, 1997] Domingos, P. et Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2): pages 103–130.

[Dumais et Chen, 2000] Dumais, S. et Chen, H. (2000). Hierarchical classification of Web content. *Dans SIGIR'00 : 23rd annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 256–263.

[Dumais *et al.*, 1998] Dumais, S., Platt, J., Heckerman, D. et Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. *Dans CIKM'98 : 7th international Conference on Information and Knowledge Management*, pages 148–155.

[Efron, 1983] Efron, B. (1983). Estimating the error rate of a prediction rule : improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331.

[Fan *et al.*, 2008] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. et Lin, C.-J. (2008). LIBLINEAR : A library for large linear classification. *Journal of Machine Learning Research*, 9: pages 1871–1874.

[Fawcett, 2006] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8): pages 861–874.

[Ferrari *et al.*, 2008] Ferrari, V., Fevrier, L., Jurie, F. et Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1): pages 36–51.

[Fisher, 1936] Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7: pages 179–188.

[Fodor, 2002] Fodor, I. (2002). A survey of dimension reduction techniques. Rapport technique UCRL-ID-148494, Lawrence Livermore National Laboratory.

[Ford et Roberts, 1998] Ford, A. et Roberts, A. (1998). Colour space conversions.

[Fourel, 1998] Fourel, F. (1998). *Modélisation, indexation et recherche de documents structurés*. Thèse de doctorat, Université Joseph Fourier.

[Fox et Sharan, 1986] Fox, E. et Sharan, S. (1986). A comparison of two methods for soft boolean operator interpretation in information retrieval. Rapport technique TR-86-01, Virginia Polytechnic Institute & State University.

[Fox et Shaw, 1994] Fox, E. et Shaw, J. (1994). Combination of multiple searches. *Dans TREC-3 : 3rd Text REtrieval Conference*, pages 243–252.

[Fragoudis *et al.*, 2005] Fragoudis, D., Meretakis, D. et Likothanassis, S. (2005). Best terms : an efficient feature-selection algorithm for text categorization. *Knowledge and Information Systems*, 8(1): pages 16–33.

[Freund et Schapire, 1995] Freund, Y. et Schapire, R. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *Dans COLT'95 : 2nd european conference on COmputational Learning Theory*, pages 23–37.

[Freund et Schapire, 1999] Freund, Y. et Schapire, R. (1999). A short introduction to boosting. *Japonese Society for Artificial Intelligence*, 14(5): pages 771–780.

[Friedman *et al.*, 1997] Friedman, N., Geiger, D. et Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29: pages 131–163.

[Fuhr, 2003] Fuhr, N. (2003). Information retrieval. Lecture Notes in Summer Semester.

[Furnas *et al.*, 1987] Furnas, G., Landauer, T., Gomez, L. et Dumais, S. (1987). The vocabulary problem in human-system communication. *Communication ACM*, 30(11): pages 964–971.

[Galavotti *et al.*, 2000] Galavotti, L., Sebastiani, F. et Simi, M. (2000). Experiments on the use of feature selection and negative evidence in automated text categorization. *Dans ECDL'00 : 4th European Conference on research and advanced technology for Digital Libraries*, pages 59–68.

[Garcia, 2006] Garcia, E. (2006). The term count model. `http://www.miislita.com/term-vector/term-vector-2.html`.

[Géry *et al.*, 2009] Géry, M., Largeron, C. et Moulin, C. (2009). UJM at INEX 2008 XML mining track. *Dans INEX'09 : 8th international workshop of the INitiative for the Evaluation of XML retrieval*, pages 446–452.

[Géry *et al.*, 2009] Géry, M., Largeron, C. et Thollard, F. (2009). Impact précoce du poids des balises pour la recherche d'information ciblée. *Dans CORIA'09 : 6e COnférence en Recherche d'Information et Applications*, pages 333–348.

[Goodrum, 2000] Goodrum, A. (2000). Image information retrieval : An overview of current research. *Informing Science*, 3(2): pages 63–66.

[Han *et al.*, 2001] Han, E.-H., Karypis, G. et Kumar, V. (2001). Text categorization using weight adjusted k-nearest neighbor classification. *Dans PAKDD'01 : 5th Pacific-Asia conference on Knowledge Discovery and Data mining*, pages 53–65.

[Hanbury, 2008] Hanbury, A. (2008). A survey of methods for image annotation. *Journal of Visual Languages and Computing*, 19(5): pages 617–627.

[Harris et Stephens, 1988] Harris, C. et Stephens, M. (1988). A combined corner and edge detector. *Dans Alvey vision conference*, pages 147–151.

[Hinneburg *et al.*, 2000] Hinneburg, E., Aggarwal, C., Keim, D. et Hinneburg, A. (2000). What is the nearest neighbor in high dimensional spaces ? *Dans VLDB'00 : 26th international conference on Very Large Data Bases*, pages 506–515.

[How et Kiong, 2005] How, B. et Kiong, W. (2005). An examination of feature selection frameworks in text categorization. *Dans AIRS'05 : 2nd Asia Information Retrieval Symposium*, pages 558–564.

[Howarth et Rüger, 2004] Howarth, P. et Rüger, S. (2004). Evaluation of texture features for content-based image retrieval. *Dans CIVR'04 : 3rd ACM international Conference on Image and Video Retrieval*, pages 326–334.

[Hull et Grefenstette, 1996] Hull, D. et Grefenstette, G. (1996). Querying across languages : a dictionary-based approach to multilingual information retrieval. *Dans SIGIR'96 : 19th annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 49–57.

[Indyk et Motwani, 1998] Indyk, P. et Motwani, R. (1998). Approximate nearest neighbors : towards removing the curse of dimensionality. *Dans STOC'98 : 30th annual ACM Symposium on Theory of COmputing*, pages 604–613.

[Joachims, 1997] Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *Dans ICML'97 : 14th International Conference on Machine Learning*, pages 143–151.

[Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines : learning with many relevant features. *Dans ECML'98 : 10th European Conference on Machine Learning*, pages 137–142.

[Jurie et Triggs, 2005] Jurie, F. et Triggs, B. (2005). Creating efficient codebooks for visual recognition. *Dans ICCV'05 : 10th IEEE International Conference on Computer Vision*, pages 604–610.

[Kamps *et al.*, 2005] Kamps, J., Marx, M., de Rijke, M. et Sigurbjörnsson, B. (2005). Structured queries in XML retrieval. *Dans CIKM'05 : 14th ACM international Conference on Information and Knowledge Management*, pages 2–11.

[Kamps *et al.*, 2008] Kamps, J., Pehcevski, J., Kazai, G., Lalmas, M. et Robertson, S. (2008). INEX 2007 evaluation measures. *Dans INEX'07 : 6th international workshop of the INitiative for the Evaluation of XML retrieval*, pages 24–33.

[Kazawa *et al.*, 2005] Kazawa, H., Izumitani, T., Taira, H. et Maeda, E. (2005). Maximal margin labeling for multi-topic text categorization. *Dans NIPS'05 : advances in Neural Information Processing Systems 17*, pages 649–656.

[Kobayashi et Takeda, 2000] Kobayashi, M. et Takeda, K. (2000). Information retrieval on the web. *ACM Computing Surveys*, 32(2): pages 144–173.

[Kompaore et Mothe, 2008] Kompaore, N. D. et Mothe, J. (2008). Fusion de résultats en recherche d'information : mesure de l'impact de l'union et de l'intersection de résultats. *Dans CIDE'08 : Conférence Internationale sur le Document Électronique.*

[Koschan, 1995] Koschan, A. (1995). A comparative study on color edge detection. *Dans ACCV'95 : 2nd Asian Conference on Computer Vision*, pages 574–578.

[Kukich, 1992] Kukich, K. (1992). Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4): pages 377–439.

[Lagrange, 1853] Lagrange, J. (1853). *Mécanique analytique*. Mallet-Bachelier.

[Lalmas, 2009] Lalmas, M. (2009). XML information retrieval. Encyclopedia of Library and Information Sciences.

[Lan *et al.*, 2005] Lan, M., Sung, S., Low, H. et Tan, C. (2005). A comparative study on term weighting schemes for text categorization. *Dans IJCNN'05 : IEEE International Joint Conference on Neural Networks*, volume 1, pages 546–551.

[Lanckriet *et al.*, 2004] Lanckriet, G., Deng, M., Cristianini, N., Jordan, M. et Noble, W. (2004). Kernel-based data fusion and its application to protein function prediction in yeast. *Dans PSB'04 : Pacific Symposium on Biocomputing*, pages 300–311.

[Laptev et Lindeberg, 2003] Laptev, I. et Lindeberg, T. (2003). Space-time Interest Points. *Dans ICCV'03 : 9th IEEE International Conference on Computer Vision*, pages 432–439.

[Largeron et Moulin, 2010] Largeron, C. et Moulin, C. (2010). Sélection par entropie de descripteurs textuels pour la catégorisation de documents XML. *Dans EGC'10 : 10e conférence internationale francophone sur l'Extraction et la Gestion des Connaissances*, pages 645–646.

[Largeron *et al.*, 2010] Largeron, C., Moulin, C. et Géry, M. (2010). UJM at INEX 2009 XML mining track. *Dans INEX'09 : 8th international workshop of the INitiative for the Evaluation of XML retrieval*, pages 426–433.

[Largeron *et al.*, 2011] Largeron, C., Moulin, C. et Géry, M. (2011). Entropy based feature selection for text categorization. *Dans SAC'11 : 26th Symposium On Applied Computing*, pages 924–928.

[Lazebnik *et al.*, 2006] Lazebnik, S., Schmid, C. et Ponce, J. (2006). Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories. *Dans CVPR'06 : IEEE computer society conference on Computer Vision and Pattern Recognition*, pages 2169–2178.

[Lebart et Fénelon, 1971] Lebart, L. et Fénelon, J.-P. (1971). *Statistique et informatique appliquées*. Dunod.

[Lee, 1997] Lee, J. H. (1997). Analyses of multiple evidence combination. *Dans SIGIR'97 : 20th annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 267–276.

[Lee et Fox, 1988] Lee, W. et Fox, E. (1988). Experimental comparison of schemes for interpreting Boolean queries. Rapport technique TR-88-27, Virginia Polytechnic Institute & State University Blacksburg.

[Leibe et Schiele, 2006] Leibe, B. et Schiele, B. (2006). Interleaving object categorization and segmentation. *Dans Cognitive Vision Systems*, pages 145–161.

[Lemaître *et al.*, 2009] Lemaître, C., Moulin, C., Barat, C. et Ducottet, C. (2009). Combinaison d'information visuelle et textuelle pour la recherche d'information multimédia. *Dans GRETSI'09 : 22e colloque du Groupe d'Etudes du Traitement du Signal et des Images*.

[Leung et Malik, 2001] Leung, T. et Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1): pages 29–44.

[Lewis, 1992a] Lewis, D. (1992a). An evaluation of phrasal and clustered representations on a text categorization task. *Dans SIGIR'92 : 15th annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 37–50.

[Lewis, 1992b] Lewis, D. (1992b). Feature selection and feature extraction for text categorization. *Dans Speech and Natural Language Workshop*, pages 212–217.

[Lewis, 1998] Lewis, D. (1998). Naive (Bayes) at Forty : The Independence Assumption in Information Retrieval. *Dans ECML'98 : 10th European Conference on Machine Learning*, pages 4–15.

[Lewis et Ringuette, 1994] Lewis, D. et Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. *Dans SDAIR'94 : 3rd annual Symposium on Document Analysis and Information Retrieval*, pages 81–93.

[Lewis *et al.*, 2004] Lewis, D., Yang, Y., Rose, T. et Li, F. (2004). RCV1 : A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5: pages 361–397.

[Li et Perona, 2005] Li, F.-F. et Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. *Dans CVPR'05 : IEEE computer society conference on Computer Vision and Pattern Recognition*, pages 524–531.

[Li et Jain, 1998] Li, Y. et Jain, A. (1998). Classification of text documents. *The Computer Journal*, 41: pages 537–546.

[Lowe, 1999] Lowe, D. (1999). Object recognition from local scale-invariant features. *Dans ICCV'99 : 7th International Conference on Computer Vision*, pages 1150–1157.

[Lowe, 2004] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2): pages 91–110.

[Luo et Zincir-Heywood, 2005] Luo, X. et Zincir-Heywood, N. (2005). Evaluation of two systems on multi-class multi-label document classification. *Dans ISMIS'05 : 15th International Syposium on Methodologies for Intelligent Systems*, pages 161–169.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Dans 5th Berkeley symposium on mathematical statistics and probability*, pages 281–297.

[Magalhaes et Rüger, 2007] Magalhaes, J. et Rüger, S. (2007). Information-theoretic semantic multimedia indexing. *Dans CIVR'07 : 6th ACM international Conference on Image and Video Retrieval*, pages 619–626.

[Mahalanobis, 1936] Mahalanobis, P. (1936). On the generalised distance in statistics. *Dans PNAS'36 : Proceedings National Institute of Science*, pages 49–55.

[Manjunath *et al.*, 2002] Manjunath, B., Ohm, J., Vasudevan, V. et Yamada, A. (2002). Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6): pages 703–715.

[Manning *et al.*, 2008] Manning, C., Raghavan, P. et Schtze, H. (2008). *Introduction to information retrieval.* Cambridge University Press.

[Maree *et al.*, 2005] Maree, R., Geurts, P., Piater, J. et Wehenkel, L. (2005). Random subwindows for robust image classification. *Dans CVPR'05 : IEEE computer society conference on Computer Vision and Pattern Recognition*, pages 34–40.

[Marr et Hildreth, 1980] Marr, D. et Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167): pages 187–217.

[Matas *et al.*, 2002] Matas, J., Chum, O., Martin, U. et Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. *Dans BMVC'02 : 13th British Machine Vision Conference*, pages 384–393.

[Mercier et Beigbeder, 2006] Mercier, A. et Beigbeder, M. (2006). Fuzzy term proximity with boolean queries at 2006 TREC terabyte task. *Dans TREC'06 : 15th Text REtrieval Conference.*

[Mikolajczyk et Schmid, 2004] Mikolajczyk, K. et Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1): pages 63–86.

[Miller *et al.*, 1990] Miller, G., Beckwith, R., Fellbaum, C., Gross, D. et Miller, K. (1990). Introduction to WordNet : An on-line lexical database. *International Journal of Lexicography*, 3: pages 235–244.

[Mladenic, 1998] Mladenic, D. (1998). Feature subset selection in text-learning. *Dans ECML'98 : 10th European Conference on Machine Learning*, pages 95–100.

[Moravec, 1977] Moravec, H. (1977). Towards Automatic Visual Obstacle Avoidance. *Dans IJCAI'77 : 5th International Joint Conference on Artifical Intelligence*, pages 584–584.

[Moschitti, 2003] Moschitti, A. (2003). A study on optimal parameter tuning for Rocchio text classifier. *Dans ECIR'03 : 25th European Conference on Information Retrieval*, pages 546–547.

[Moulin *et al.*, 2010a] Moulin, C., Barat, C. et Ducottet, C. (2010a). Fusion of tf.idf weighted bag of visual features for image classification. *Dans CBMI'10 : workshop on Content Based Multimedia Indexing*, pages 124–129.

[Moulin *et al.*, 2008] Moulin, C., Barat, C., Géry, M., Ducottet, C. et Largeron, C. (2008). UJM at ImageCLEFwiki 2008. *Dans CLEF'08 : 9th workshop of the Cross-Language Evaluation Forum*, pages 779–786.

[Moulin *et al.*, 2009] Moulin, C., Barat, C., Lemaître, C., Géry, M., Ducottet, C. et Largeron, C. (2009). Combining text/image in WikipediaMM task 2009. *Dans CLEF'09 : 10th workshop of the Cross-Language Evaluation Forum*, pages 164–171.

[Moulin *et al.*, 2010b] Moulin, C., Largeron, C. et Géry, M. (2010b). Impact de l'information visuelle pour la Recherche d'Images par le contenu et le contexte. *Dans CORIA'10 : 7e COnférence en Recherche d'Information et Applications*, pages 179–193.

[Moulin *et al.*, 2010c] Moulin, C., Largeron, C. et Géry, M. (2010c). Impact of visual information on text and content based image retrieval. *Dans S+SSPR'10 :13th international workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 159–169.

[Mouret *et al.*, 2009] Mouret, M., Solnon, C. et Wolf, C. (2009). Classification of images based on hidden markov models. *Dans CBMI'09 : workshop on Content Based Multimedia Indexing*, pages 169–174.

[Mulhem et Chevallet, 2010] Mulhem, P. et Chevallet, J.-P. (2010). Modèle de langue par type de doxel pour l'indexation de documents structurés. *Dans CORIA'10 : 7e COnférence en Recherche d'Information et Applications*, pages 361–372.

[Naudé, 1627] Naudé, G. (1627). *Advis pour dresser une bibliothèque*. Isidore Liseux (1876).

[Nayak *et al.*, 2010] Nayak, R., De Vries, C., Kutty, S., Geva, S., Denoyer, L. et Gallinari, P. (2010). Overview of the INEX 2009 XML mining track : Clustering and classification of XML documents. *Dans INEX'09 : 8th international workshop of the INitiative for the Evaluation of XML retrieval*, pages 366–378.

[Ng *et al.*, 1997] Ng, H. T., Goh, W. B. et Low, K. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. *Dans SIGIR'97 : 20th annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 67–73.

[Nguyen *et al.*, 2009] Nguyen, N., Ogier, J.-M., Tabbone, S. et Boucher, A. (2009). Text retrieval relevance feedback techniques for bag of words model in CBIR. *Dans ICMLPR'09 : International Conference on Machine Learning and Pattern Recognition*.

[Nister et Stewenius, 2006] Nister, D. et Stewenius, H. (2006). Scalable recognition with a vocabulary tree. *Dans CVPR'06 : IEEE computer society conference on Computer Vision and Pattern Recognition*, pages 2161–2168.

[Novoviccaronová *et al.*, 2004] Novoviccaronová, J., Malík, A. et Pudil, P. (2004). Feature selection using improved mutual information for text classification. *Dans S+SSPR'04 :10th international workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 1010–1017.

[Nowak *et al.*, 2006] Nowak, E., Jurie, F. et Triggs, B. (2006). Sampling strategies for bag-of-features image classification. *Dans ECCV'06 : 9th European Conference on Computer Vision : workshop on Statistical Learning in Computer Vision*, pages 490–503.

[O'Keefe et Trotman, 2003] O'Keefe, R. et Trotman, A. (2003). The simplest query language that could possibly work. *Dans INEX'03 : 2nd international workshop of the INitiative for the Evaluation of XML retrieval*, pages 167–174.

[Over *et al.*, 2010] Over, P., Awad, G., Fiscus, J., Michel, M., Smeaton, A. et Kraaij, W. (2010). TRECVID 2009 - goals, tasks, data, evaluation mechanisms and metrics. *Dans TRECVID'09 : TREC VIDeo retrieval evaluation*.

[Pass et Zabih, 1996] Pass, G. et Zabih, R. (1996). Histogram refinement for content-based image retrieval. *Dans WACV'96 : 3rd IEEE Workshop on Applications of Computer Vision*, pages 96–102.

[Pearson, 1900] Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series 5*, 50(302): pages 157–175.

[Pham *et al.*, 2008] Pham, T.-T., Chevallet, J.-P. et Lim, J.-H. (2008). Fusion de multi-modalités et réduction par sémantique latente. *Dans CORIA'08 : 5e COnférence en Recherche d'Information et Applications*, pages 39–53.

[Philbin *et al.*, 2007] Philbin, J., Chum, O., Isard, M., Sivic, J. et Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. *Dans CVPR'07 : IEEE conference on Computer Vision and Pattern Recognition*, pages 1–8.

[Ponte et Croft, 1998] Ponte, J. et Croft, B. (1998). A language modeling approach to information retrieval. *Dans SIGIR'98 : 21th annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 275–281.

[Porter, 1980] Porter, M. (1980). An algorithm for suffix stripping. *Program : electronic library and information systems*, 14(3): pages 130–137.

[Quinlan, 1996] Quinlan, J. (1996). Bagging, boosting, and C4.5. *Dans AAAI'96 : 13th national conference on artificial intelligence, workshop on new research problems for machine learning*, pages 725–730.

[Ráez et López, 2006] Ráez, A. et López, L. (2006). Selection strategies for multi-label text categorization. *Dans Advances in Natural Language Processing*, pages 585–592.

[Rajman et Lebart, 1998] Rajman, M. et Lebart, L. (1998). Similarités pour données textuelles. *Dans JADT'98 : 4th international conference on statistical analysis of textual sata*, pages 545–555.

[Robertson et Spärck Jones, 1976] Robertson, S. et Spärck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3): pages 129–146.

[Robertson *et al.*, 1994] Robertson, S., Walker, S., Hancock-Beaulieu, M., Gull, A. et Lau, M. (1994). Okapi at TREC-3. *Dans TREC-3 : 3rd Text REtrieval Conference*, pages 21–30.

[Ros *et al.*, 2006] Ros, J., Laurent, C. et Lefebvre, G. (2006). A cascade of unsupervised and supervised neural networks for natural image classification. *Dans CIVR'06 : 5th ACM international Conference on Image and Video Retrieval*, pages 92–101.

[Russell *et al.*, 2008] Russell, B., Torralba, A., Murphy, K. et Freeman, W. (2008). LabelMe : A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1): pages 157–173.

[Salembier et Smith, 2002] Salembier, P. et Smith, J. (2002). MPEG-7 multimedia description schemes. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6): pages 748–759.

[Salton et Buckley, 1990] Salton, G. et Buckley, C. (1990). Improving retrieval performance by relevance feedback. *American Society for Information Science*, 41(4): pages 288–297.

[Salton *et al.*, 1983] Salton, G., Fox, E. et Wu, H. (1983). Extended boolean information retrieval. *Communications of the ACM*, 26(11): pages 1022–1036.

[Salton *et al.*, 1975] Salton, G., Wong, A. et Yang, C. (1975). A vector space model for automatic indexing. *Communications*, 18(11): pages 613–620.

[Samuel *et al.*, 2010] Samuel, E., de La Higuera, C. et Janodet, J.-C. (2010). Extracting plan graphs from images. *Dans S+SSPR'10 :13th international workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 233–243.

[Saporta, 2006] Saporta, G. (2006). *Probabilités, analyses des données et statistique.* Éditions Technip, 2e édition révisée et augmentée édition.

[Schapire et Singer, 2000] Schapire, R. et Singer, Y. (2000). BoosTexter : A boosting-based system for text categorization. *Machine Learning*, 39(2): pages 135–168.

[Schettini *et al.*, 2001] Schettini, R., Ciocca, G. et Zuffi, S. (2001). A survey of methods for colour image indexing and retrieval in image databases. *Dans In Color Imaging Science : Exploiting Digital*, pages 183–211.

[Schölkopf *et al.*, 1998] Schölkopf, B., Smola, A. et Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5): pages 1299–1319.

[Schütze, 1998] Schütze, H. (1998). Automatic word sense discrimination. *Computational linguistics*, 24(1): pages 97–123.

[Sclaroff *et al.*, 1999] Sclaroff, S., Cascia, M., Sethi, S. et Taycher, L. (1999). Unifying textual and visual cues for content-based image retrieval on the world wide web. *Computer Vision and Image Understanding*, 75(1): pages 86–98.

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34: pages 1–47.

[Shah *et al.*, 2002] Shah, U., Finin, T., Joshi, A., Cost, S. et Matfield, J. (2002). Information retrieval on the semantic web. *Dans CIKM'02 : 11th international Conference on Information and Knowledge Management*, pages 461–468.

[Shannon, 1948] Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27: pages 379–423 and 623–656.

[Shi et Malik, 2000] Shi, J. et Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 22(8): pages 888–905.

[Sivic et Zisserman, 2003] Sivic, J. et Zisserman, A. (2003). Video Google : A text retrieval approach to object matching in videos. *Dans ICCV'03 : 8th International Conference on Computer Vision*, pages 1470–1477.

[Smeulders *et al.*, 2000] Smeulders, A., Worring, M., Santini, S., Gupta, A. et Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12): pages 1349–1380.

[Smith et Brady, 1997] Smith, S. et Brady, J. (1997). SUSAN - A new approach to low level image processing. *International journal of computer vision*, 23(1): pages 45–78.

[Snoek *et al.*, 2005] Snoek, C., Worring, M. et Smeulders, A. (2005). Early versus late fusion in semantic video analysis. *Dans MM'05 : 13th annual ACM international conference on MultiMedia*, pages 399–402.

[Snoek *et al.*, 2006] Snoek, C., Worring, M., van Gemert, J., Geusebroek, J.-M. et Smeulders, A. (2006). The challenge problem for automated detection of 101 semantic concepts in multimedia. *Dans MM'06 : 14th annual ACM international conference on MultiMedia*, pages 421–430.

[Song et Croft, 1999] Song, F. et Croft, B. (1999). A general language model for information retrieval. *Dans CIKM'99 : 8th international Conference on Information and Knowledge Management*, pages 316–321.

[Song *et al.*, 2009] Song, X., Muselet, D. et Trémeau, A. (2009). Compact local color descriptor based on rank correlations. *Dans ICCV'09 : International Conference on Computer Vision workshops*, pages 1878–1884.

[Spärck Jones *et al.*, 2000] Spärck Jones, K., Walker, S. et Robertson, S. (2000). A probabilistic model of information retrieval : development and comparative experiments. *Information Processing and Management*, 36(6): pages 779–808.

[Spyrou *et al.*, 2005] Spyrou, E., Borgne, H. L., Mailis, T., Cooke, E., Avrithis, Y. et Connor, N. (2005). Fusing MPEG-7 visual descriptors for image classification. *Dans ICANN'05 : International Conference on Artificial Neural Networks : formal models and their applications*, pages 847–852.

[Student, 1908] Student, B. (1908). The probable error of a mean. *Biometrika*, 6: pages 1–25.

[Suematsu *et al.*, 2002] Suematsu, N., Ishida, Y., Hayashi, A. et Kanbara, T. (2002). Region-based image retrieval using wavelet transform. *Dans VI'02 : 15th international conference on Vision Interface*, pages 167–173.

[Süsstrunk *et al.*, 1999] Süsstrunk, S., Buckley, R. et Swen, S. (1999). Standard RGB color spaces. *Dans CIC'99 : 7th Color Imaging Conference : Color Science and Engineering Systems*, pages 127–134.

[Swain et Ballard, 1991] Swain, M. et Ballard, D. (1991). Color indexing. *International Journal of Computer Vision*, 7: pages 11–32.

[Szeliski, 2006] Szeliski, R. (2006). Image alignment and stitching : A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1): pages 1–104.

[Szummer et Picard, 1998] Szummer, M. et Picard, R. (1998). Indoor-outdoor image classification. *Dans CAIVD'98 : international workshop on Content-based Access of Image and Video Databases*, pages 42–51.

[Tang *et al.*, 2005] Tang, B., Shepherd, M., Milios, E. et Heywood, M. (2005). Comparing and combining dimension reduction techniques for efficient text clustering. *Dans International workshop on feature selection for data mining : Interfacing machine learning and statistics*, pages 17–26.

[Tirilly, 2010] Tirilly, P. (2010). *Traitement automatique des langues pour l'indexation d'images*. Thèse de doctorat, Université de Rennes 1, Rennes, France.

[Tirilly *et al.*, 2008] Tirilly, P., Claveau, V. et Gros, P. (2008). Language modeling for bag-of-visual words image categorization. *Dans CIVR'08 : international conference on Content-based Image and Video Retrieval*, pages 249–258.

[Tirilly *et al.*, 2009] Tirilly, P., Claveau, V. et Gros, P. (2009). A review of weighting schemes for bag of visual words image retrieval. Rapport technique PI-1927, Université de Rennes I.

[Toennies *et al.*, 2002] Toennies, K., Behrens, F. et Aurnhammer, M. (2002). Feasibility of hough-transform-based iris localisation for real-time-application. *Dans ICPR'02 : 16th International Conference on Pattern Recognition*, pages 1053–1056.

[Tollari et Glotin, 2006] Tollari, S. et Glotin, H. (2006). WISTI : a simple efficient textuo-visual web image retrieval model - Specifications and Benchmarks. *Dans CIVR'06 : 5th ACM international Conference on Image and Video Retrieval : workshop ImagEVALxq*.

[Torjmen *et al.*, 2009] Torjmen, M., Pinel-Sauvagnat, K. et Boughanem, M. (2009). Some experiments on the WikipediaMM 2008 task : Evaluating the impact of image names in context-based retrieval. *Dans CLEF'09 : 10th workshop of the Cross-Language Evaluation Forum*, pages 756–762.

[Trajkovic et Hedley, 1998] Trajkovic, M. et Hedley, M. (1998). Fast corner detection. *Image and Vision Computing*, 16(2): pages 75–87.

[Tsikrika et Kludas, 2008] Tsikrika, T. et Kludas, J. (2008). Overview of the wikipediaMM task at ImageCLEF 2008. *Dans CLEF'08 : 9th workshop of the Cross-Language Evaluation Forum.*

[Tsikrika et Kludas, 2009] Tsikrika, T. et Kludas, J. (2009). Overview of the wikipediaMM task at ImageCLEF 2009. *Dans CLEF'09 : 10th workshop of the Cross-Language Evaluation Forum.*

[Tsoumakas et Katakis, 2007] Tsoumakas, G. et Katakis, I. (2007). Multi-label classification : an overview. *International Journal of Data Warehousing and Mining*, 3(3): pages 1–13.

[Uchyigit et Ma, 2008] Uchyigit, G. et Ma, M. (2008). *Personalization techniques and recommender systems*, volume 70. World Scientific Pub Co Inc.

[Vailaya *et al.*, 2001] Vailaya, A., Figueiredo, M., Jain, A. et Zhang, H.-J. (2001). Image classification for content-based indexing. *IEEE Transactions on Image Processing*, 10: pages 117–130.

[van de Sande *et al.*, 2008] van de Sande, K. E., Gevers, T. et Snoek, C. (2008). A comparison of color features for visual concept classification. *Dans CIVR'08 : 7th ACM international Conference on Image and Video Retrieval*, pages 141–150.

[van Rijsbergen, 1979] van Rijsbergen, C. K. (1979). *Information retrieval.* Butterworth-Heinemann ; 2nd edition.

[Vedaldi et Fulkerson, 2010] Vedaldi, A. et Fulkerson, B. (2010). VLFeat – An open and portable library of computer vision algorithms. *Dans MM'2010 : 18th annual ACM international conference on MultiMedia*, pages 1469–1472.

[Verbyst et Mulhem, 2009] Verbyst, D. et Mulhem, P. (2009). Using collectionlinks and documents as Context for INEX 2008. *Dans INEX'08 : 7th international workshop of the INitiative for the Evaluation of XML retrieval.*

[Vidal-Naquet et Ullman, 2003] Vidal-Naquet, M. et Ullman, S. (2003). Object recognition with informative features and linear classification. *Dans ICCV'03 : 9th IEEE International Conference on Computer Vision*, pages 281–288.

[Vogel et Schiele, 2002] Vogel, J. et Schiele, B. (2002). On performance characterization and optimization for image retrieval. *Dans ECCV'02 : 7th European Conference on Computer Vision*, pages 49–63.

[Vogt et Cottrell, 1999] Vogt, C. et Cottrell, G. (1999). Fusion via a linear combination of scores. *Information Retrieval*, 1(3): pages 151–173.

[Voss *et al.*, 1999] Voss, A., Nakata, K. et Juhnke, M. (1999). Concept indexing. *Dans GROUP'99 : international ACM SIGGROUP conference on supporting group work*, pages 1–10.

[Wallace *et al.*, 1997] Wallace, J., Raaphorst, G., Somorjai, R., Ng, C., Fung Fung, M., Senterman, M. et Smith, I. (1997). Classification of 1H MR spectra of biopsies from untreated and recurrent ovarian cancer using linear discriminant analysis. *Magnetic Resonance in Medicine*, 38(4): pages 569–576.

[Wang *et al.*, 2000] Wang, J., Li, J. et Wiederhold, G. (2000). SIMPLIcity : Semantics-sensitive integrated matching for picture libraries. *Dans VISUAL'00 : 4th international conference on advances in visual information systems*, pages 360–371.

[Wiener *et al.*, 1995] Wiener, E., Pedersen, J. et Weigend, A. (1995). A neural network approach to topic spotting. *Dans SDAIR'95 : 4th annual Symposium on Document Analysis and Information Retrieval*, pages 317–332.

[Wolpert, 1992] Wolpert, D. (1992). Stacked generalization. *Neural networks*, 5(2): pages 241–259.

[Wong *et al.*, 1985] Wong, S., Ziarko, W. et Wong, P. (1985). Generalized vector spaces model in information retrieval. *Dans SIGIR'85 : 8th annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 18–25.

[Wookey et Geller, 2004] Wookey, L. et Geller, J. (2004). Semantic hierarchical abstraction of web site structures for web searchers. *Journal of Research and Practice in Information Technology*, 36(1): pages 23–34.

[Wu *et al.*, 2004] Wu, Y., Chang, E., Chang, K. et Smith, J. (2004). Optimal multimodal fusion for multimedia data analysis. *Dans MM'04 : 12th annual ACM international conference on MultiMedia*, pages 572–579.

[Yan et Hauptmann, 2003] Yan, R. et Hauptmann, A. (2003). The combination limit in multimedia retrieval. *Dans MM'03 : 11th ACM international conference on MultiMedia*, pages 339–342.

[Yang *et al.*, 2007] Yang, J., Jiang, Y.-G., Hauptmann, A. et Ngo, C.-W. (2007). Evaluating bag-of-visual-words representations in scene classification. *Dans MIR'07 : international workshop on Multimedia Information Retrieval*, pages 197–206.

[Yang, 2001] Yang, Y. (2001). A study of thresholding strategies for text categorization. *Dans SIGIR'01 : 24th annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 137–145.

[Yang et Liu, 1999] Yang, Y. et Liu, X. (1999). A re-examination of text categorization methods. *Dans SIGIR'99 : 22rd annual international ACM Special Interest Group on Information Retrieval conference on research and development in information retrieval*, pages 42–49.

[Yang et Pedersen, 1997] Yang, Y. et Pedersen, J. (1997). A comparative study on feature selection in text categorization. *Dans ICML'97 : 14th International Conference on Machine Learning*, pages 412–420.

[Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3): pages 338–353.

[Zhai, 2001] Zhai, C. (2001). Notes on the Lemur TFIDF model. Rapport technique, Carnegie Mellon University.

[Zhai, 2008] Zhai, C. (2008). Statistical language models for information retrieval a critical review. *Foundations and Trends in Information Retrieval*, 2: pages 137–213.

[Zhang et Lu, 2004] Zhang, D. et Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, 37(1): pages 1–19.

[Zhang et Zhou, 2005] Zhang, M.-L. et Zhou, Z.-H. (2005). A k-nearest neighbor based algorithm for multi-label classification. *Dans GrC'2005 : 1st IEEE international conference on Granular Computing*, pages 718–721.

[Zhang *et al.*, 1997] Zhang, X., Farrell, J. et Wandell, B. (1997). Applications of a spatial extension to CIELAB. *Dans 9th annual symposium on electronic imaging*, pages 154–157.

[Zhao et Grosky, 2002] Zhao, R. et Grosky, W. (2002). Narrowing the semantic gap - improved text-based web document retrieval using visual features. *IEEE Transactions on Multimedia*, 4: pages 189–200.

[Zhou *et al.*, 2008] Zhou, H., Yuan, Y. et Shi, C. (2008). Object tracking using SIFT features and mean shift. *Computer vision and image understanding*, 113(3): pages 345–352.

[Zhou et Huang, 2002] Zhou, X. et Huang, T. (2002). Unifying keywords and visual contents in image retrieval. *IEEE MultiMedia*, 9(2): pages 23–33.

Résumé

L'exploitation des documents multimédias pose des problèmes de représentation des informations textuelles et visuelles contenues dans ces documents. Notre but est de proposer un modèle permettant de représenter chacune de ces informations et de les combiner en vue de deux tâches : la catégorisation et la recherche d'information.

Ce modèle représente les documents sous forme de sacs de mots nécessitant la création de vocabulaires spécifiques. Le vocabulaire textuel, généralement de très grande taille, est constitué des mots apparaissant dans les documents. Le vocabulaire visuel est quant à lui construit en extrayant des caractéristiques de bas niveau des images. Nous étudions les différentes étapes de sa création et la pondération tf.idf des mots visuels dans les images, inspirée des approches classiquement utilisées pour les mots textuels.

Dans le contexte de la catégorisation de documents textuels, nous introduisons un critère qui sélectionne les mots les plus discriminants pour les catégories afin de réduire la taille du vocabulaire sans dégrader les résultats du classement. Nous présentons aussi dans le cadre multilabel, une méthode permettant de sélectionner les différentes catégories à associer à un document.

En recherche d'information, nous proposons une approche analytique par apprentissage pour combiner linéairement les résultats issus des informations textuelles et visuelles, permettant d'améliorer significativement la recherche. Notre modèle est validé pour ces différentes tâches en participant à des compétitions internationales telles que XML Mining et ImageCLEF et sur des collections de taille conséquente.

Abstract

Exploiting multimedia documents leads to representation problems of the textual and visual information within documents. Our goal is to propose a model to represent these both information and to combine them for two tasks : categorization and information retrieval.

This model represents documents as bags of words, which requires to define adapted vocabularies. The textual vocabulary, usually very large, corresponds to the words of documents while the visual one is created by extracting low-level features from images. We study the different steps of its creation and the tf.idf weighting of visual words in images usually used for textual words.

In the context of the text categorization, we introduce a criterion to select the most discriminative words for categories in order to reduce the vocabulary size without degrading the results of classification. We also present in the multilabel context, a method that lets us to select the number of categories which must be associated with a document.

In multimedia information retrieval, we propose an analytical approach based on machine learning techniques to linearly combine the results from textual and visual information which significantly improves research results. Our model has shown its efficiency on different collections of important size and was evaluated in several international competitions such as XML Mining and ImageCLEF.