
Diversité hiérarchique et utilisation d'arbres de concepts pour la recherche d'images

Christian Kuoman^{*,**} — Sabrina Tollari^{*} — Marcin Detyniecki^{*}

** UPMC Univ Paris 06 - UMR 7606 LIP6, 4 place Jussieu 75252 PARIS cedex 05
{Christian.Kuoman-Mamani,Sabrina.Tollari,Marcin.Detyniecki}@lip6.fr*

*** Xilopix, 2 rue de Nancy, 88000 Epinal, France*

RÉSUMÉ. La recherche d'images est de plus en plus efficace, mais les résultats similaires ont tendance à se regrouper. Dans cet article, nous montrons comment améliorer la diversité des résultats en prenant en compte la nature intrinsèquement hiérarchique de la diversité. Afin d'exploiter les différents niveaux de granularité de la diversité, nous utilisons une approche basée sur une classification ascendante hiérarchique (CAH). De plus, nous introduisons une nouvelle approche qui exploite une arborescence de concepts. Pour éviter le conflit entre pertinence et diversité, les différentes approches sont comparées sur un corpus dédié annoté manuellement où tous les documents sont pertinents, et sur un corpus de l'état de l'art (ImageClef photo 2008). Les résultats montrent que l'approche hiérarchique améliore la diversité des résultats.

ABSTRACT. Current search engines return relevant, but often similar results. In this paper we propose not only to show how to increase the diversity, but also how to address the hierarchical nature of the diversity. To address this challenge, we propose innovative image ordering strategies based on an agglomerative hierarchical classification (AHC). Moreover, we introduce a novel approach for exploiting richer description resources, such as "concept trees". The different approaches are compared on a highly relevant and manually annotated benchmark: the Xilopix benchmark; and on the, more general but less adapted, ImageClef2008 benchmark. Any of the proposed approaches increase the diversity (CR20) compared to search engine's standard output and outperform an average random shuffling (baseline). Furthermore, hierarchical ordering increases the diversity in all cases, in particular compared to a flat ordering.

MOTS-CLÉS : diversité, classification ascendante hiérarchique, recherche d'images, concepts

KEYWORDS: diversity, hierarchical ascendant classification, image retrieval, concept trees

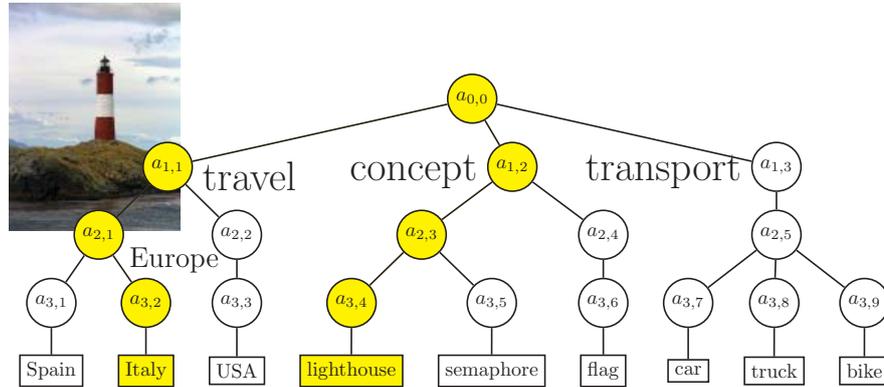


Figure 1. Représentation graphique d'un exemple d'arborescence de concepts

1. Introduction

Actuellement, les moteurs de recherche d'images retournent des résultats pertinents. Cependant, dans la plupart des cas, les résultats similaires ont tendance à se regrouper entre eux. Par exemple, si nous effectuons la requête "phare" dans un moteur de recherche en ligne, les premiers résultats fournis seront quasiment tous des images correspondant à des phares dans la mer par une belle journée ensoleillée. Un utilisateur devra parcourir de nombreux résultats afin de trouver des phares de voiture. De plus, même en ce qui concerne les phares (au sens système de signalisation), il existe différents niveaux de catégories de phares : les phares maritimes, les phares aéronautiques... qui se trouvent de plus dans différents types de scènes : de jour, de nuit, en pleine tempête... Pour éviter à l'utilisateur de parcourir de nombreux résultats dans la liste pour trouver le type d'images qu'il recherche, l'idéal serait d'obtenir une liste d'images qui soient les plus pertinentes et diverses à la fois.

Généralement, les images sont décrites par un vecteur de descripteurs visuels, cependant, parfois, des sources d'informations plus riches sont disponibles. Ainsi notre travail se déroule dans le contexte réel d'une entreprise qui a développé un moteur de recherche en ligne et qui dispose d'images classées dans une arborescence de concepts (thésaurus). Cette arborescence de concepts contient plusieurs sous-arbres thématiques (appelés univers). La figure 1 donne un exemple de description d'une image dans l'arborescence des concepts. L'image de cette figure est associée aux concepts : 'Italy' dans l'univers 'Travel' et 'lighthouse' dans l'univers 'Concept'.

De nombreux travaux ont déjà été effectués dans le domaine de la diversité mais peu de travaux se sont intéressés à la diversité dans le cas d'images décrites dans une arborescence de concepts. De plus, nous proposons de résoudre le problème de la diversité en prenant en compte particulièrement certains aspects : (i) la diversité possède différents niveaux de granularité, (ii) la diversité doit être adaptée à la diversité

intrinsèque de la requête et donc aux images retrouvées, (iii) la méthode qui produit la diversité doit pouvoir être adaptée à la requête de l'utilisateur et donc doit pouvoir être exécutée "en ligne" en un temps minimal. Pour une requête donnée, il existe plusieurs granularités dans la diversité des résultats. Par exemple, si la requête est "avocat". Un premier niveau de diversité permet de séparer les images représentant le fruit avocat, des images représentant l'homme de loi. Un deuxième niveau permet de séparer les images représentant l'homme de loi en plusieurs catégories : une seule personne, groupe de personnes, photographies prises en intérieur, en extérieur... Pour prendre en compte les niveaux de granularité, nous avons identifié que la classification ascendante hiérarchique (CAH) est une approche adaptée, car elle permet d'obtenir une organisation hiérarchique des images¹, et nous montrons qu'elle est également adaptée pour une utilisation "en ligne".

Dans la section 2, nous dressons un rapide état de l'art des différents types de méthodes de diversité. Ensuite, dans la section 3, nous montrons comment utiliser la CAH dans le cas d'images décrites par des ensembles de chemins dans une arborescence afin de créer des clusters hiérarchisés. Puis, dans la section 4, nous présentons une méthode pour réordonner les résultats en prenant en compte la hiérarchie des clusters obtenue par CAH. Dans la section 5, nous montrons expérimentalement sur 2 corpus différents que l'utilisation de notre méthode de diversité basée sur un réordonnement utilisant les clusters hiérarchisés est une méthode efficace.

2. État de l'art des méthodes de diversité pour la recherche d'images

Quand un utilisateur pose une requête, ce qui l'intéresse c'est d'avoir des documents qui soient certes tous pertinents, mais aussi qui soient les plus dissimilaires les uns des autres. Les modèles traditionnels de recherche d'information supposent que la pertinence d'un document est indépendante de la pertinence des autres documents. Cependant, pour une requête donnée, l'utilité d'un document peut dépendre des documents que l'utilisateur a déjà vu (Zhai *et al.*, 2003), car si il a déjà vu un document d'un certain type, il pourra être plus intéressé par un document d'un autre type. Il existe une manière simple de produire de la diversité en permutant aléatoirement les résultats. Certains travaux proposent également d'utiliser la date de prise de vue de l'image ou bien sa géolocalisation (Maisonasse *et al.*, 2009, Lestari Paramita *et al.*, 2009). Ce type d'informations semblent permettre une certaine diversité, mais n'est pas toujours disponible. La plupart des méthodes de diversité sont exécutées après la requête : une recherche traditionnelle est effectuée, une liste ordonnée de résultats est obtenue, puis dans une phase de post-traitement, les résultats sont réordonnés afin de produire la diversité. Il existe aussi des méthodes (Zwol *et al.*, 2008) qui intègrent le processus de diversification lors de la phase de recherche. Par exemple, en créant un partitionnement de la base d'images.

1. La hiérarchie de clusters d'images fournie par la CAH ne doit pas être confondue avec l'arborescence des concepts du thésaurus.

C. Kuoman, S. Tollari, M. Detyniecki

Plusieurs travaux (Deselaers *et al.*, 2009, Halvey *et al.*, 2009, Ah-Pine *et al.*, 2010) proposent d'utiliser des algorithmes gloutons : à chaque itération, l'image recherchée est celle qui est la plus dissimilaire aux images sélectionnées. D'autres travaux (Ah-Pine *et al.*, 2010, Tollari *et al.*, 2009) proposent de fusionner plusieurs listes ordonnées de résultats. Dans chaque liste, l'élément ayant le rang le plus faible est sélectionné, puis le second élément de chaque liste, et ainsi de suite, mais en évitant les doublons. Les listes ordonnées peuvent provenir de recherches effectuées sur des modalités différentes (texte, image, texte-image...) ou de méthodes différentes (clustering, permutation...). D'autres types de méthodes de diversité sont basées sur du clustering. Dans (Tollari *et al.*, 2009), le clustering est effectué par un partitionnement de l'espace visuel, et non pas des données. (Zhao *et al.*, 2009) propose une méthode basée sur un clustering avec propagation d'affinité. Dans (Inoue *et al.*, 2008), un clustering hiérarchique est effectué sur des vecteurs construits à partir de concepts visuels ; comparé à un kmeans, le clustering hiérarchique donne de meilleurs résultats. Ce travail est proche du notre parce qu'il utilise un clustering hiérarchique pour produire de la diversité, mais il diffère, car il n'utilise ni arborescence de concepts, ni descripteurs visuels. De plus, les méthodes de réordonnement utilisées sont totalement différentes.

Un des problèmes souvent rencontrés par les méthodes de diversité est qu'en augmentant la diversité, la pertinence des résultats a tendance à diminuer (Ziegler *et al.*, 2005, Tollari *et al.*, 2009). C'est pourquoi certains travaux (Deselaers *et al.*, 2009) proposent d'effectuer une optimisation conjointe avec l'objectif d'obtenir un résultat qui soit pertinent et divers à la fois. Cependant, dans le cas où la diversité augmente, une petite perte de précision peut être acceptable, car elle peut correspondre à la suppression d'une image pertinente, mais qui appartient à l'une des catégories déjà représentées, et qui n'apporte donc qu'une information redondante pour l'utilisateur. Afin d'éviter le conflit entre pertinence et diversité, dans le cadre de notre étude, nous nous plaçons dans le cas où toutes les images sont pertinentes.

3. Classification ascendante hiérarchique pour l'arborescence de concepts

Dans le cas où chaque image est décrite par un seul concept appartenant à une arborescence de concepts, il peut être assez simple de construire une hiérarchie des images en utilisant la hiérarchie de l'arborescence des concepts. Il suffit de mettre dans le même cluster les images qui appartiennent à la même feuille (méthode que nous appelons *XiloTree*). Cependant, dans la plupart des cas, les images sont décrites par plusieurs concepts. Une approche consistant à classer les images en fonction de l'arborescence des concepts n'est donc pas applicable.

La Classification Ascendante Hiérarchique (CAH) (Lance *et al.*, 1967) est une méthode de clustering qui permet d'obtenir une hiérarchie de clusters d'images. Cette caractéristique nous offre plusieurs niveaux de diversité. Ainsi, cette méthode permet à l'utilisateur de parcourir en profondeur la hiérarchie en fonction du niveau de granularité qu'il souhaite obtenir. L'algorithme classique de la CAH se déroule ainsi : (i) initialiser chaque cluster par une image, (ii) calculer la distance entre toutes les paires

de clusters, (iii) agréger les deux clusters qui ont la plus petite distance, (iv) répéter (ii) et (iii) jusqu'à ce que tous les clusters soient regroupés dans un cluster. Une fois la hiérarchie construite, une technique de découpage peut être appliquée pour obtenir des clusters. Nous allons dans la suite décrire les différentes techniques que nous avons utilisées pour adapter la CAH dans le cas d'images décrites par des chemins dans l'arborescence de concepts.

3.1. Similarité entre deux images

Dans notre cas, pour calculer la similarité entre deux images, nous devons calculer la similarité entre deux ensembles de chemins de l'arborescence de concept. Pour chaque univers j , une image i est décrite par un chemin : $A_i^j = (a_{0,0}, a_1^j(i), \dots, a_p^j(i))$ où $a_1^j(i)$ est le noeud racine de l'univers j et $a_p^j(i)$ est la feuille du chemin de l'image i dans l'univers j . Par exemple, pour l'image de la figure 1, le chemin dans l'univers 'Travel' est : $(a_{0,0}, a_{1,1}, a_{2,1}, a_{3,2})$. Dans notre benchmark, pour chaque image, il y a au maximum un seul chemin possible pour un univers donné, mais les images sont décrites dans différents univers. Les images sont donc associées à un ensemble de chemins dans l'arborescence de concepts : $A_i = \{A_i^1, A_i^2, \dots, A_i^j, \dots\}$. Afin de calculer la similarité entre deux chemins, nous comparons deux mesures classiques : la similarité de Wu-Palmer (Wu *et al.*, 1994) et la similarité de Lin (Lin *et al.*, 1998). Puis, nous proposons une généralisation de ces mesures de similarités entre deux chemins dans le cas où nous avons deux ensembles de chemins.

3.1.1. Similarité de Wu-Palmer

La similarité de Wu-Palmer (Wu *et al.*, 1994) mesure la similarité entre deux concepts d'une même arborescence. Dans notre cas, les concepts qui nous intéressent sont les feuilles de l'arbre, nous calculons la similarité entre le chemin $A_{i_1}^j$ de l'image i_1 et le chemin $A_{i_2}^j$ de l'image i_2 par la formule suivante :

$$sim_{WP}(A_{i_1}^j, A_{i_2}^j) = \frac{2 \times z}{p(i_1) + p(i_2)}$$

où $p(i_1)$ (resp. $p(i_2)$) est la profondeur de l'arborescence de l'image i_1 (resp. l'image i_2) et z est la profondeur du noeud commun le plus profond.

3.1.2. Similarité de Lin

La similarité de Lin (Lin *et al.*, 1998) est une autre mesure de la similarité entre deux concepts dans une arborescence. Elle a la particularité de prendre en compte le nombre d'images qui sont associées à un noeud par rapport au nombre total d'images. La similarité entre le chemin $A_{i_1}^j$ de l'image i_1 et le chemin $A_{i_2}^j$ de l'image i_2 est :

$$sim_{LIN}(A_{i_1}^j, A_{i_2}^j) = \frac{2 \times \log P(a_z^j(i_1))}{\log P(a_{p(i_1)}^j(i_1)) + \log P(a_{p(i_2)}^j(i_2))}$$

où z est la profondeur du noeud commun le plus profond et $P(a_{p(i_1)}^j(i_1))$ (resp. $P(a_{p(i_2)}^j(i_2))$) est la probabilité calculée en fonction du nombre d'occurrences du noeud $a_{p(i_1)}^j(i_1)$ (resp. $a_{p(i_2)}^j(i_2)$) de profondeur $p(i_1)$ (resp. $p(i_2)$) dans l'univers j par rapport au nombre total d'images.

3.1.3. Généralisation au cas de la dissimilarité entre deux ensembles de chemins

Nous proposons une généralisation des mesures de similarités entre deux chemins dans le cas où nous avons deux ensembles de chemins. Nous faisons cependant l'hypothèse que, pour un univers donné, chaque image est décrite au plus par un seul concept. De plus, pour avoir un modèle identique dans le cas des descripteurs visuels (où la distance euclidienne est utilisée), et dans le cas des descripteurs basés sur l'arborescence de concepts, nous souhaitons obtenir une dissimilarité plutôt qu'une similarité. C'est pourquoi au lieu d'utiliser sim_{WP} , nous utilisons $\delta_{WP} = 1 - sim_{WP}$, et de même au lieu de sim_{LIN} , nous utilisons $\delta_{LIN} = 1 - sim_{LIN}$. La dissimilarité δ_G entre l'ensemble de chemins A_{i_1} de l'image i_1 et l'ensemble de chemins A_{i_2} de l'image i_2 est définie par la formule suivante :

$$\delta_G(A_{i_1}, A_{i_2}) = \frac{1}{|J(i_1) \cup J(i_2)|} \sum_{j \in J(i_1) \cup J(i_2)} \delta(A_{i_1}^j, A_{i_2}^j)$$

où $J(i_1)$ (resp. $J(i_2)$) est l'ensemble des univers de l'image i_1 (resp. i_2) et $\delta(A_{i_1}^j, A_{i_2}^j)$ est une dissimilarité entre le chemin de l'image i_1 et le chemin de l'image i_2 dans l'univers j , dans ce travail, nous utilisons soit $\delta_{WP}(A_{i_1}^j, A_{i_2}^j)$ soit $\delta_{LIN}(A_{i_1}^j, A_{i_2}^j)$.

Par exemple, dans le cas de l'arborescence de la figure 1, nous avons une image i_1 associée aux concepts 'Italy' dans l'univers 1 ('Travel') et 'lighthouse' dans l'univers 2 ('Concept'), c'est-à-dire : $A_{i_1} = (A_{i_1}^1, A_{i_1}^2)$. Prenons une autre image i_2 associée aux concepts 'Spain' dans l'univers 1 ('Travel') et 'truck' dans l'univers 3 ('Transport'), c'est-à-dire : $A_{i_2} = (A_{i_2}^1, A_{i_2}^3)$. Pour ces deux ensembles des chemins, le calcul de la dissimilarité généralisée en utilisant Wu-Palmer est :

$$\begin{aligned} \delta_G(A_{i_1}, A_{i_2}) &= \frac{1}{3}(\delta(A_{i_1}^1, A_{i_2}^1) + \delta(A_{i_1}^2, A_{i_2}^2) + \delta(A_{i_1}^3, A_{i_2}^3)) \\ &= \frac{1}{3}(1 - sim_{WP}(A_{i_1}^1, A_{i_2}^1) + 1 + 1) = 1 - \frac{1}{3} \frac{2 \times 2}{3+3} = 0.778 \end{aligned}$$

Cette mesure suppose qu'une image décrite par un seul mot est plus proche d'une image décrite par un seul mot du même univers, que d'une image décrite par un mot du même univers et d'autres mots en provenance d'autres univers. Par exemple, une image décrite par 'avocat' sera plus proche d'une image décrite par 'magistrat' que d'une image décrite par 'avocat' et 'robe', car les magistrats et les avocats appartiennent au même univers 'Justice', tandis que 'robe' appartient à un autre univers.

3.2. Critère d'agrégation : *RootFusion*

Il existe de nombreux critères d'agrégation. Citons par exemple le lien simple, le lien complet, le lien moyen... Le critère centroïde est un critère d'agrégation classique qui calcule la distance entre deux clusters en calculant la distance entre les barycentres des clusters. Dans le cas où les images sont décrites par un ensemble de chemins dans une arborescence, les critères classiques ne peuvent pas être utilisés, c'est pourquoi en nous inspirant du critère centroïde, nous proposons une nouvelle méthode d'agrégation appelée *RootFusion* adaptée à ce cas. A l'initialisation de la CAH, chaque cluster est initialisé par une seule image : le représentant du cluster C_{i_1} est donc l'ensemble de chemins A_{i_1} de l'image i_1 . Ensuite, le système calcule la dissimilarité δ_G entre tous les couples de clusters et agrège les deux clusters dont les représentants sont les plus similaires. Les deux représentants sont alors agrégés en gardant leurs noeuds communs afin de former le représentant du nouveau cluster. Par exemple, pour l'arborescence de la figure 1, l'image i_1 est associé aux concepts : 'Italy' et 'lighthouse'. Si nous considérons une autre image i_3 associée aux concepts : 'Spain', 'semaphore' et 'car', alors le nouveau centroïde obtenu par la méthode *RootFusion* est l'ensemble de chemins : $\{(a_{0,0}, a_{1,1}, a_{2,1}), (a_{0,0}, a_{1,2}, a_{2,3})\}$. Ce critère est intéressant car le nouveau représentant de chaque cluster est facile à calculer et ne nécessite que peu de temps de calcul, il est donc adapté pour une utilisation en ligne.

4. Diversité par réordonnement des images

Dans notre travail, nous considérons le cas où toutes les images sont pertinentes par rapport à la requête. L'ordre dans lequel les images ont été retrouvées n'est donc pas significatif. Notre problème est donc de trouver un ordre des images qui améliore la diversité. En utilisant une CAH, les images similaires ont été regroupées hiérarchiquement. Nous devons maintenant trouver à quel endroit couper la hiérarchie de la CAH afin d'obtenir des clusters qui augmentent la diversité, puis nous devons ordonner de manière pertinente les clusters obtenus, et finalement ordonner les images en fonction de la hiérarchie des clusters et de l'ordre des clusters.

4.1. Coupure du dendrogramme

Une fois que le dendrogramme a été créé par la CAH, nous devons déterminer à quel niveau couper horizontalement le dendrogramme afin d'obtenir des clusters adaptés à notre problème de diversité. Le problème de la coupure du dendrogramme est un problème qui n'a pas de réponse unique. Un choix simple est de couper le dendrogramme de manière à obtenir un nombre fixé de clusters (méthode appelée *Fixe*). Nous pouvons également adapter le nombre de clusters pour chaque requête si nous connaissons le nombre de sous-thèmes attendus pour chaque requête (méthode appelée *Adapt*). Dans la réalité, on ne connaît pas le nombre de sous-thèmes pour une requête donnée, mais cette méthode *Adapt* est intéressante comme élément de

C. Kuoman, S. Tollari, M. Detyniecki

comparaison. Pour connaître le nombre de sous-thèmes, nous utilisons la vérité terrain. (Milligan *et al.*, 1985) propose une technique souvent utilisée qui consiste à calculer, à chaque itération de la CAH, la différence $\Delta(n, n + 1)$ entre la valeur de la distance d'agrégation à l'étape n et celle à l'étape $n + 1$, et à couper le dendrogramme après l'étape n qui a la plus grande différence (méthode appelée *Traditionnel*).

4.2. *Priorité des clusters*

Une fois que nous avons découpé le dendrogramme, nous obtenons une liste non-ordonnée de clusters. Nous devons alors déterminer la priorité des clusters, c'est-à-dire dans quel ordre les clusters seront utilisés. Lorsqu'une requête est effectuée, les résultats sont souvent ordonnés en fonction de leur similarité à la requête (Inoue *et al.*, 2008). Une manière simple d'ordonner les clusters est de prendre en premier le cluster qui contient l'image de rang 1, puis en deuxième le cluster qui contient l'image restante de rang le plus fort et ainsi de suite (priorité que nous appelons *Rank*). Cependant, lorsque beaucoup d'images sont très pertinentes, cet ordre n'est pas toujours très significatif. Par exemple, si la requête est le mot "phare", toutes les images contenant un phare seront très pertinentes, mais il est difficile de dire qu'une image contenant un phare est plus pertinente qu'une autre image contenant un phare.

Afin de trouver un ordre de priorité des clusters, nous proposons de prendre en compte le nombre d'images que contient chaque cluster. Si les clusters sont ordonnés du plus grand nombre d'images au plus petit nombre d'images (priorité *Décroissante*), les premiers résultats seront des images que l'utilisateur s'attend à trouver pour sa requête ; dans le cas contraire (priorité *Croissante*), les premiers résultats montreront en premier des images originales. Par exemple, si nous ordonnons les clusters de manière décroissante, nous obtenons pour la requête "phare" d'abord des images de phares maritimes, puis les images de phares de voiture. Inversement, si nous ordonnons les clusters de manière croissante, nous obtenons d'abord les images de phares de voiture, puis les phares maritimes.

4.3. *Réordonnements plat et hiérarchique*

Afin d'utiliser les différents niveaux de granularité fournis par la CAH, nous proposons une nouvelle méthode de réordonnement des images appelée *HierO* est nous la comparons avec un réordonnement plat qui est une méthode plus classique pour ordonner les images.

Dans le cas du réordonnement plat (méthode *FlatO*), les images sont réordonnées en prenant dans l'ordre des clusters une image de chaque cluster. Par exemple, pour l'image de la figure 2, les images du cluster 1 sont liés au thème 'jaguar animal' et les images du cluster 2 sont liés au thème 'jaguar voiture'. Le résultat final, en utilisant le réordonnement plat, est un mélange d'images entre le thème 'jaguar animal' et le thème 'jaguar voiture' (voir résultats (a), (c) et (e) dans la figure).

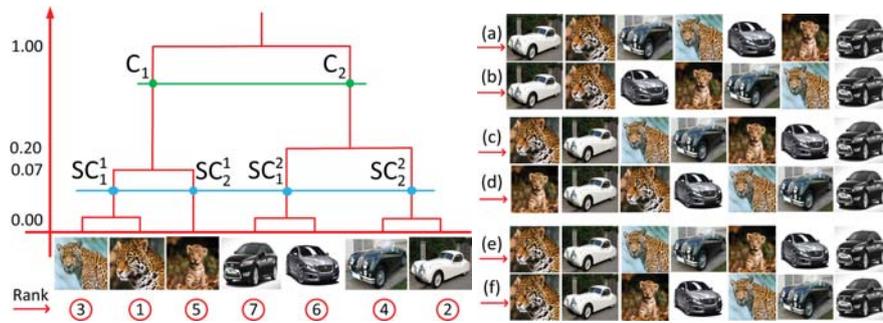


Figure 2. Dendrogramme de la CAH et résultats de réordonnancement. Les images du cluster 1 sont liées au thème ‘jaguar animal’ (et au sous-thème ‘bébé jaguar’ et ‘jaguar adulte’) et les images du cluster 2 sont liées au thème ‘jaguar voiture’ (et au sous-thème ‘voiture actuelle’ et ‘voiture ancienne’). Les résultats fournis par la méthode de réordonnancement plat avec priorités des clusters (a) Décroissante, (c) Croissante ou (e) Rank sont moins diversifiés que les résultats fournis par le réordonnancement hiérarchique avec priorités (b) Décroissante, (d) Croissante ou (f) Rank

Dans le cas du réordonnancement hiérarchique (méthode *HierO*), les images sont réordonnées en prenant dans l’ordre des clusters et en alternant les sous-clusters, une image de chaque cluster. Les sous-clusters sont obtenus en appliquant une deuxième fois la même technique de coupure de la hiérarchie. Par exemple, pour la technique de coupure *Fixe15/25*, la hiérarchie est coupée une première fois afin d’obtenir 15 clusters, puis la hiérarchie est coupée une deuxième fois afin d’obtenir 25 clusters. La méthode *HierO Fixe15/25* n’est pas équivalente à *FlatO Fixe15* (15 clusters) ni à *FlatO Fixe25* (25 clusters), car dans le cas de *HierO Fixe15/25* les 15 clusters sont triés une première fois selon le critère de priorité des clusters choisis, puis lors de la deuxième coupure, les sous-clusters sont retriés mais seulement à l’intérieur des premiers clusters. Pour produire le résultat final, les images sont triées dans chaque cluster selon leur rang, puis on ajoute au résultat final la première image de chaque cluster, puis au deuxième passage, on choisit la seconde image dans un sous-cluster différent que celui de la première image. Par exemple, en utilisant le réordonnancement hiérarchique, le résultat est une alternance d’images entre les sous-thèmes du thème ‘jaguar animal’ et les sous-thèmes du thème ‘jaguar voiture’ (voir résultats (b), (d) et (f) dans la figure). Si on compare les images réordonnées, nous remarquons que les quatre premières images des résultats (b), (d) et (f) appartiennent bien à quatre sous-thèmes différents, contrairement aux résultats (a), (c) et (e) où les quatre premières images contiennent seulement deux sous-thèmes. Il y a donc bien une plus grande diversité des sous-thèmes, avec le réordonnancement hiérarchique qu’avec le réordonnancement plat.

Requête	nbClusters	Requête	nbClusters	Requête	nbClusters
rouge	16	composition	16	transport	12
blanc	12	commonwealth	12	bleu	12
fond	12	tradition	12	animaux	12
ancien	11	île	11	mer	11
marie	11	grande	11	aquatique	10
bois	10	voiture	9	ciel	9
corde	7	fête	6	fer	6

Tableau 1. Les 21 requêtes du benchmark Xilopix (*nbClusters* est le nombre de clusters pour chaque requête)

5. Expérimentations

5.1. Mesure de la diversité

La mesure de diversité utilisée dans cet article est le cluster rappel (CR) citée dans (Arni *et al.*, 2008, Clarke *et al.*, 2008, Zhai *et al.*, 2003). Pour une requête donnée, la vérité terrain donne le nombre de clusters différents pour cette requête. Le cluster rappel à n documents consiste à calculer le nombre de clusters différents retrouvés dans les n premiers résultats par rapport au nombre total de clusters différents pour une requête donnée :

$$CR(n) = \frac{\text{nombre de clusters retrouvés dans les } n \text{ premiers résultats}}{\text{nombre de clusters}}$$

Nous utilisons comme mesure d'évaluation le CR(20), noté par la suite CR20, car c'est celle qui est la plus souvent utilisée, et elle permet une comparaison directe avec les autres travaux. De plus, elle correspond approximativement au nombre d'images que peut contenir une page Web de résultats de recherche d'images.

5.2. Benchmark Xilopix

Le benchmark Xilopix a été réalisé dans le cadre d'un partenariat avec une entreprise de recherches d'images sur le Web. Les images et leurs annotations sont mises à disposition par l'entreprise. Chaque image est décrite par un histogramme de couleurs dans l'espace HSV. Le texte associé aux images consiste généralement en quelques mots-clés choisis manuellement. Chaque image est associée manuellement à une ou plusieurs feuilles de l'arborescence de concepts. Ce benchmark est donc le reflet d'une application réaliste. La vérité terrain de ce benchmark a été construite en partenariat avec les documentalistes de l'entreprise selon le protocole suivant. Les documentalistes ont choisis 21 requêtes qui offrent le plus de diversité parmi les plus demandées par les utilisateurs dans le moteur de recherche. Pour chacune des 21 requêtes, les

100 premiers résultats retournés par le moteur de recherche ont été récupérés. Ces résultats sont tous pertinents, car les images sont annotées à la main. Par contre, les résultats ne sont pas diversifiés. Après avoir obtenu les 21 requêtes, les documentalistes ont classé les images résultats afin de construire la vérité terrain. Par exemple, pour la requête “voiture”, les documentalistes ont analysé et classé les images résultats de cette requête en 9 clusters (voiture dans la ville, voiture de foire, voiture ancienne, voiture neuve, voiture crachée, voiture artistique, volant de voiture, voiture d’enfants, voiture du train). Le tableau 1 donne la liste des 21 requêtes. Pour les 21 requêtes, il y a en moyenne 10.9 clusters (minimum 6 clusters, maximum 16 clusters) par requête. En moyenne sur les images du corpus, il y a 1.75 concepts par image (maximum 3 concepts), 13 univers différents sont utilisés et la profondeur moyenne de l’arbre de concept est de 6.54.

5.3. *Benchmark ImageClef2008*

Le second benchmark est le benchmark de ImageClef 2008 Photo Retrieval (Arni *et al.*, 2008). Il contient 20k images et 39 requêtes. Chaque image est décrite par du texte. Chaque requête est composée de texte et d’images, ainsi que par un critère imposé de diversité. Pour chaque requête, la vérité terrain indique les images pertinentes ainsi que les clusters de diversité auxquelles appartiennent ces images pertinentes. Par exemple, la première requête demande de retourner des églises qui ont plus de deux tours ; le critère de diversité pour cette requête est le critère “ville”, c’est-à-dire que ce que l’on cherche à obtenir est que chaque image dans les premiers résultats doit avoir été prise dans une ville différente.

Pour éviter le conflit entre pertinence et diversité, nous avons récupéré les résultats ordonnés fournis dans (Tollari *et al.*, 2009) appelés “runs idéaux”. Un “run” est l’ensemble des 39 résultats ordonnés. Ces runs idéaux ont été construits à l’aide de la vérité terrain. Pour chaque requête, seules les images pertinentes pour la requête sont considérées (run idéal numéro 1). Puis une permutation aléatoire de ces résultats pertinents est répétée 9 fois pour chaque requête afin d’obtenir en tout 10 runs idéaux. Pour ces 10 runs idéaux, la précision à 20 documents est proche de 1 (car tous les documents sont pertinents), mais par contre le CR20 varie de 0.683 à 0.807 avec une moyenne de 0.750. Il y a en moyenne 7.9 clusters (minimum 2 clusters, maximum 23 clusters, écart-type de 5.0) et 62 images pertinentes (minimum 18 images, maximum 185, écart-type de 33.8) par requête. Pour ce benchmark, nous n’utilisons ni texte, ni concepts, mais seulement pour chaque image un histogramme de couleurs HSV.

5.4. *Résultats*

Pour les méthodes basées sur la couleur, nous utilisons la distance euclidienne et le critère centroïde afin de calculer la CAH. Pour les méthodes basées sur l’arborescence de concepts, nous utilisons le critère RootFusion.

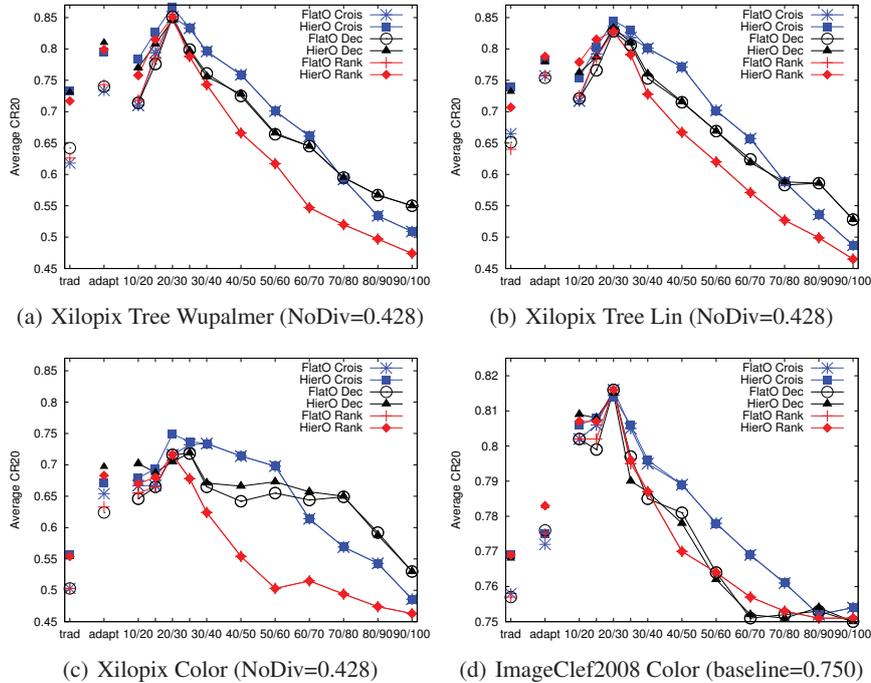


Figure 3. Comparaison des CR20 moyens pour les réordonnements FlatO et HierO, avec priorité des clusters Croissante, Décroissante ou Rank sur le benchmark Xilopix en utilisant (a) l'arborescence des concepts et la similarité Wu-Palmer, (b) l'arborescence des concepts et la similarité de Lin, et (c) les descripteurs de couleurs, et (d) sur le benchmark ImageClef2008 en utilisant les descripteurs de couleurs

La figure 3(a) donne les CR20 moyens obtenus sur le benchmark Xilopix en utilisant comme descripteurs l'arborescence de concepts et la dissimilarité généralisée utilisant Wu-Palmer. Nous remarquons que la technique de coupe *Adapt* est plus efficace que la technique *Traditionnelle*. Nous notons également que les courbes croissent de *Fixe10/20* à *Fixe20/30*, puis décroissent de *Fixe20/30* jusqu'à *Fixe90/100*. Les résultats optimaux sont obtenus à *Fixe20/30*. Nous avons également calculé le CR10 et le CR30 pour ces mêmes expériences et nous avons noté que les résultats optimaux sont obtenus aux alentours de *Fixe10/20* pour le CR10 et *Fixe30/40* pour le CR30. Il semblerait donc que le bon objectif d'optimisation n'est pas le nombre de vrai classes comme dans la méthode *Adapt*, mais le nombre d'images à afficher, par exemple 20 images pour le CR20.

La méthode *HierO* obtient des résultats nettement meilleurs que la méthode *FlatO*, en particulier pour les techniques *Adapt*, *Traditionnelle*, *Fixe10/20* et *Fixe15/25*, où

Méthode	CR20	Temps
NoDiv	0.428 (ref.)	-
Aléatoire	0.673 (+57%)	-
XiloTree Croissant	0.793 (+85%)	-
XiloTree Décroissant	0.800 (+87%)	-
XiloTree Rank	0.751 (+75%)	-
FlatO Couleur Croissant Fixe30/40	0.734 (+71%)	1.20
HierO Couleur Croissant Fixe20/30	0.749 (+75%)	1.21
FlatO Arborescence Croissant Fixe20/30	0.851 (+99%)	0.90
HierO Arborescence Croissant Fixe20/30	0.867 (+103%)	0.90

Tableau 2. Synthèse des principaux résultats sur le benchmark Xilopix. Le temps est le temps moyen de réordonnement par requête (en seconde)

les courbes *HierO* sont nettement supérieures aux courbes *FlatO* correspondantes. Pour les techniques *Fixe20/30* à *Fixe90/100*, les méthodes *FlatO* et *HierO* donnent quasiment les mêmes résultats. La diversité qui est un problème hiérarchique est donc mieux appréhendée par une approche hiérarchique que par une approche plate.

En ce qui concerne la priorité des clusters, les scores montrent que la priorité *Croissante* donne globalement de meilleurs résultats que les priorités *Décroissante* et *Rank*. Rechercher les images originales (celles qui se trouvent dans les clusters contenant peu d'images) et les placer en premier dans la liste de résultats semble donc être une piste intéressante. La priorité *Rank* ne donne pas de bons résultats ici car dans notre travail le rang n'est pas significatif, mais sur un benchmark où le rang est important, cette méthode devrait donner de bons résultats.

La comparaison de la figure 3(a) avec la figure 3(b) obtenue à partir de la dissimilarité généralisée utilisant la dissimilarité de Lin, montre que globalement la similarité de Lin permet d'obtenir des résultats assez similaires en terme de comportement, mais légèrement inférieurs à ceux obtenus à l'aide de Wu-Palmer.

Si nous comparons les figures 3(a) et 3(c), nous remarquons que les méthodes basées sur la couleur donnent des résultats nettement inférieurs aux méthodes basées sur l'arborescence. Cependant, les courbes obtenues en utilisant la couleur ont des comportements assez similaires à ceux obtenus à partir de l'arborescence de concept. L'arborescence de concept est donc une source d'information riche qui permet d'obtenir de la diversité, tandis que la couleur n'est pas très adaptée pour ce qui est de trouver des clusters sémantiques. Néanmoins, si on compare les courbes de couleurs par rapport à la baseline sans diversité (NoDiv), qui obtient un CR20 de 0.428, on remarque que la couleur améliore nettement les résultats.

Le tableau 2 résume les principaux résultats obtenus sur le benchmark Xilopix. La méthode NoDiv correspond aux résultats retournés par le moteur de recherche

C. Kuoman, S. Tollari, M. Detyniecki

d'images de Xilopix. Tous les résultats sont pertinents, la précision est donc de 1. Cependant, le CR20 de NoDiv est seulement de 0.428, ce qui signifie que les résultats similaires ont tendances à se regrouper. En effet, les images étant annotées avec des mots-clés, beaucoup d'images ont les mêmes indexations, et sont donc ordonnées par rapport à leur date d'indexation. Or généralement, les images indexées au même moment ont des mots-clés similaires.

Une méthode simple pour obtenir de la diversité est de permuter aléatoirement les résultats (méthode *Aléatoire*). L'aléatoire permet ici d'augmenter le CR20 de 57%.

Pour pouvoir mesurer l'intérêt de la CAH par rapport à une utilisation directe de l'arborescence de concepts, nous utilisons la méthode *XiloTree* qui construit les clusters à partir des feuilles de l'arborescence de concepts. Pour cette méthode, nous ne pouvons prendre en compte que le premier concept qui annote chaque image. *XiloTree* permet d'augmenter le CR20 jusqu'à 0.8. Ces résultats sont supérieurs aux résultats utilisant la CAH sur les descripteurs de couleurs. Mais, nous voyons l'intérêt d'utiliser une CAH, car les résultats obtenus avec les méthodes *HierO* et *FlatO* sur l'arborescence de concepts sont nettement supérieurs à ceux de la méthode *XiloTree*.

Au niveau des temps de calcul, les méthodes utilisant l'arborescence sont plus rapides à calculer que les méthodes utilisant les descripteurs de couleurs, car la dissimilarité généralisée est très rapide à calculer. Le temps moyen de calcul pour une requête en lançant la CAH sur les 100 premières images est de l'ordre de la seconde, ce qui est un temps raisonnable pour une application en ligne, surtout que nous n'avons pas cherché à optimiser les calculs.

Afin de comparer nos travaux à ceux de l'état de l'art, nous effectuons maintenant nos expériences sur le benchmark ImageClef2008. Rappelons que ce benchmark n'est pas vraiment adapté, car il ne possède pas d'arborescence de concepts, nous ne pouvons donc qu'utiliser les descripteurs de couleurs. De plus, le CR20 moyen sur les 10 runs idéaux est de 0.75, ce qui est déjà assez élevé, car une permutation aléatoire a été réalisée sur ces runs, il est donc difficile d'augmenter la diversité sur ces 10 runs.

La figure 3(d) montre des comportements assez similaires à ceux obtenus sur le benchmark Xilopix. Comme précédemment les résultats de *HierO* sont nettement supérieurs aux résultats de *FlatO* pour les techniques de coupure *Adapt*, *Traditionnel*, *Fixe10/20* et *Fixe15/25*, et similaires pour les autres.

Le tableau 3 compare nos résultats avec ceux de l'état de l'art utilisant seulement l'information visuelle et résumés dans (Tollari *et al.*, 2009). La méthode DIVALEA permute aléatoirement les 40 premiers résultats. La méthode VISKMEANS est une méthode de partitionnement des données, tandis que la méthode DIVVISU est basée sur un partitionnement de l'espace visuel. En moyenne des CR20 sur les 10 runs idéaux, nos méthodes *FlatO* et *HierO* améliorent la diversité de +9% par rapport aux 10 runs idéaux, de +4% par rapport à la méthode DIVVISU, et de +7% par rapport aux méthodes VISKMEANS et DIVALEA.

Méthode	CR20 moyen	Ecart-type moyen
NoDiv (run numéro 1)	0.683	-
10 runs idéaux (Aléatoire)	0.750 (ref.)	0.032
DIVVISU	0.787 (+5%)	0.025
VISKMEANS	0.767 (+2%)	0.034
DIVALEA	0.764 (+2%)	0.022
FlatO Couleur Rank Fixe20/30	0.816 (+9%)	0.016
HierO Couleur Rank Fixe20/30	0.816 (+9%)	0.016

Tableau 3. Comparaison des méthodes de diversité sur ImageClef2008. CR20 moyen : moyenne de la moyenne des 39 CR20 de chacun des 10 runs idéaux

6. Conclusion et perspectives

La diversité est un problème que nous pensons intrinsèquement hiérarchique. En effet, pour une requête donnée, les résultats peuvent être regroupés par thème, puis par sous-thème et ainsi de suite. C'est pourquoi nous avons choisi d'utiliser une CAH afin de tirer profit de sa structure hiérarchique. Dans cet article, nous nous sommes concentrés sur deux principaux problèmes. Premièrement, comment exploiter une arborescence de concepts afin de regrouper des images. Nous montrons que nous pouvons facilement généraliser la similarité de Wu-Palmer afin de mesurer la dissimilarité entre deux ensembles de chemins dans l'arborescence. Les résultats montrent qu'il est préférable d'utiliser l'arborescence de concepts plutôt que les descripteurs visuels dans une perspective de diversité. Deuxièmement, comment exploiter la hiérarchie de clusters obtenue par CAH afin de produire une plus grande diversité. D'une part, nous montrons que l'utilisation de la CAH permet d'améliorer les résultats de diversité par rapport à une méthode utilisant directement la structure de l'arborescence de concepts (*XiloTree*). D'autre part, nous constatons qu'une approche hiérarchique *HierO* donne systématiquement de meilleurs résultats qu'une méthode analogue non-hiérarchique. Enfin, la plupart des méthodes de diversité n'utilisent que le rang pour choisir les représentants des clusters, mais dans ce travail nous montrons que prendre en compte le nombre d'images présentes dans le cluster est une information utile pour produire la diversité. Dans nos expériences, notre approche surpasse les modèles de l'état de l'art, mais les benchmarks publiquement disponibles ne sont pas adaptés pour mesurer l'efficacité des méthodes proposées. C'est pourquoi nous avons développé un benchmark incluant des images décrites par une arborescence de concepts. Du fait du temps nécessaire pour réaliser la vérité terrain pour chaque requête, la plupart des benchmarks contiennent beaucoup de documents, mais peu de requêtes. Pour évaluer la diversité, il serait important de créer un benchmark qui privilégie le nombre de requêtes. En ce qui concerne l'avenir de notre méthode, une piste intéressante serait de combiner l'arborescence de concept et les descripteurs visuels durant la construction de la CAH.

C. Kuoman, S. Tollari, M. Detyniecki

7. Bibliographie

- Ah-Pine J., Clinchant S., Csurka G., Perronnin F., Renders J.-M., « Leveraging image, text and cross-media similarities for diversity-focused multimedia retrieval », *ImageCLEF*, 2010.
- Arni T., Clough P., Sanderson M., Grubinger M., « Overview of the ImageCLEFphoto 2008 Photographic Retrieval Task », *Evaluating Systems for Multilingual and Multimodal Information Access*, LNCS 5706, 2008.
- Clarke C., Kolla M., Cornack G., Vechtomova O., Ashkan A., Buttcher S., MacKinnon I., « Novelty and diversity in information retrieval evaluation », *ACM SIGIR*, p. 20-24, 2008.
- Deselaers T., Gass T., Dreuw P., Ney H., « Jointly Optimising Relevance and Diversity in Image Retrieval », *ACM Conference on Image and Video Retrieval*, p. 08-10, 2009.
- Halvey M., Punitha P., Hannah D., Villa R., Hopfgartner F., Goyal A., Jose J. M., « Diversity, Assortment, Dissimilarity, Variety : A Study of Diversity Measures Using Low Level Features for Video Retrieval », *European Conference on Information Retrieval (ECIR)*, 2009.
- Inoue M., Grover P., « Query Types and Visual Concept-Based Post-retrieval Clustering », *Evaluating Systems for Multilingual and Multimodal Information Access*, 2008.
- Lance G., Williams W., « A general theory of classificatory sorting strategies : I. Hierarchical systems », *Computer Journal*, vol. 9, p. 373-380, 1967.
- Lestari Paramita M., Tang J., Sanderson M., « Generic and Spatial Approaches to Image Search Results Diversification », *European Conference on Information Retrieval (ECIR)*, 2009.
- Lin, Dekang, « An information-theoretic definition of similarity », *International Conference on Machine Learning (ICML)*, p. 296-304, 1998.
- Maisonnasse L., Mulhem P., Gaussier E., Chevallet J.-P., « LIG at ImageCLEF 2008 », *Evaluating Systems for Multilingual and Multimodal Information Access*, p. 704-711, 2009.
- Milligan G., Cooper M., « An examination of procedures for determining the number of clusters in a data set », *Psychometrika*, vol. 50, p. 159-179, 1985.
- Tollari S., Mulhem P., Ferecatu M., Glotin H., Detyniecki M., Gallinari P., Sahbi H., Zhao Z., « A comparative study of diversity methods for hybrid text and image retrieval approaches », *Evaluating Systems for Multilingual and Multimodal Information Access*, 2009.
- Wu Z., Palmer M., « Verbs semantics and lexical selection », *32nd annual meeting on Association for Computational Linguistics*, p. 133-138, 1994.
- Zhai C. X., Cohen W. W., Lafferty J., « Beyond independent relevance : methods and evaluation metrics for subtopic retrieval », *ACM SIGIR*, p. 10-17, 2003.
- Zhao Z.-Q., Glotin H., « Diversifying Image Retrieval with Affinity-Propagation Clustering on Visual Manifolds », *IEEE Multimedia*, vol. 16, n° 4, p. 34-43, 2009.
- Ziegler C., McNee S., Konstan J., Lausen G., « Improving recommendation lists through topic diversification », *WWW*, p. 22-32, 2005.
- Zwol R. V., Murdock V., Pueyo L. G., Ramirez G., « Diversifying image search with user generated content », *ACM MIR*, p. 67-74, 2008.