Gestion et visualisation des résultats d'une requête

Nicolas Bonnel, Alexandre Cotarmanac'H, Annie Morin

Gestion et visualisation des résultats d'une requête

Nicolas Bonnel*,**, Alexandre Cotarmanac'h*, Annie Morin**

*France Telecom R&D, 4 rue Clos Courtel
35512 Cesson Sévigné Cedex - France
{nicolas.bonnel,alexandre.cotarmanach}@francetelecom.com
**IRISA, Rennes - France

Résumé. Lors de recherches sur le Web, l'utilisateur est souvent confronté à un grand nombre de résultats affichés sous la forme d'une succession de listes ordonnées selon un rang. Constatant les limites de cette approche, nous proposons à travers notre prototype SmartWeb d'explorer de nouvelles organisations et présentations des résultats d'une recherche, ainsi que de nouveaux types d'interactions afin de rendre leur exploration plus intuitive et efficace. L'idée consiste à organiser les résultats selon leur "sens" en utilisant les cartes auto-organisatrices, et à les visualiser dans une scène 3D afin d'augmenter l'espace de représentation.

Mots-clés. Métaphores 3D de visualisation, Cartes auto-organisatrices de Kohonen, Interfaces adaptatives, Gestion de documents.

1 Introduction

Avec l'augmentation constante des données disponibles sur le World Wide Web, il devient de plus en plus difficile d'extraire l'information pertinente pour une recherche donnée. Les moteurs de recherche, qui sont un moyen de représentation du Web pour les utilisateurs, retournent un nombre si important de résultats qu'il est nécessaire de chercher de nouvelles méthodes de gestion de ces résultats. Ces méthodes doivent être plus adaptées grâce à : une organisation plus pertinente des résultats, une interface de visualisation plus riche et une navigation intuitive dans l'espace des résultats.

L'objectif des travaux et du prototype présentés dans cet article est d'offrir à l'utilisateur une interface de recherche qui lui permette de trouver rapidement les informations pertinentes. On se place dans le cadre d'une recherche sur le Web où les documents sont les pages Web retournées par une requête. Seule la partie textuelle de ces documents est utilisée ce qui permet d'avoir une représentation vectorielle (vecteurs de mots) des pages. Les deux points essentiels pour atteindre notre objectif sont une bonne classification des données et une visualisation efficace. Concernant ces deux aspects, on s'oriente vers une méthode de classification automatique non supervisée (les cartes auto-organisatrices) et une visualisation en trois dimensions. Le choix d'une visualisation 3D s'explique par la volonté d'augmenter l'espace de visualisation, et de permettre d'afficher un grand nombre de résultats qui n'est pas limité par la taille de l'écran mais par la perception de l'utilisateur. Cependant, bien que la 3D se rapproche de l'esprit humain d'un point de vue cognitif, elle complique la navigation et l'interaction. Enfin, la visualisation des données étant dépendante de certains critères,

Actes du Troisième Atelier Visualisation et Extraction de Connaissances (associé aux Cinquièmes Journées Francophones "Extraction et Gestion des Connaissances" - EGC'05), pages 37-47, Paris, France, 18 janvier 2005.

le prototype dispose de plusieurs interfaces qui s'adaptent en fonction du contexte.

Ce travail s'intéresse à l'organisation non supervisée des documents, à la représentation graphique des résultats ainsi qu'à l'interaction avec l'utilisateur. Dans la section suivante, un bref état de l'art sur la visualisation est proposé. Ensuite, on traite la partie organisation des documents (section 3) avant de présenter le prototype SmartWeb en section 4. Enfin la dernière section permet de faire le bilan.

2 Etat de l'art

Il existe de nombreux travaux sur la visualisation des résultats d'une recherche. Sans essayer d'en faire l'inventaire, on propose un bref aperçu de ces systèmes de visualisation qui peuvent se différencier par le nombre de dimensions qu'ils utilisent. On ne s'intéressera ici qu'aux visualisations 2D et 3D.

Parmi les visualisations 2D, on distingue essentiellement les graphes et les cartes. Le méta-moteur Kartoo[1] est un exemple de graphe, auquel on reproche cependant l'absence de vue d'ensemble des résultats. Concernant les cartes, on peut citer le projet WEBSOM (Kohonen et al. 2000). De plus, les cartes permettent souvent de tirer profit de l'aspect cognitif, du fait que l'utilisateur possède des connaissances *a priori*. C'est pourquoi on retrouve souvent des métaphores géographiques (Skupin et Fabrikant 2003) telles que Map.net[2]. Cependant, dans les deux cas, on se retrouve face à un problème de lisibilité avec l'augmentation des résultats et de la complexité des relations.

Lorsque l'on passe à des visualisations 3D, on augmente l'espace de visualisation, ce qui permet d'afficher des graphes complexes de façon plus lisible et de remplacer les cartes par des univers 3D : des paysages comme dans VxInsight (Boyack et al. 2002) ou encore des villes (Rossi et Varga 1999). Par contre, l'ajout d'une dimension rend la navigation indispensable et surtout plus compliquée. Enfin, la méthode AVE (Wiza et al. 2004) et son système Periscope sont les travaux les plus proches de ceux présentés dans cet article. Nous avons en commun l'usage du langage X-VRML, d'interfaces mixtes (scène 3D et interface 2D), ou encore l'usage de plusieurs métaphores de visualisations qui répondent à des objectifs différents. Cependant notre approche prend plus en compte le problème de l'organisation des données d'un point de vue "sémantique".

3 Organisation non supervisée des documents

Dans cette section, on s'intéresse à la classification des résultats d'une requête. La méthode proposée dans cette partie sera ensuite utilisée dans le module d'organisation des données de notre prototype. Dans notre cadre applicatif, il nous paraît essentiel que le processus d'organisation soit non supervisé et entièrement automatique. On

[1] www.kartoo.com
[2] http ://maps.map.net

s'intéresse alors plus particulièrement aux cartes auto-organisatrices de Kohonen (Kohonen 1995). Ce choix a été motivé par certaines propriétés de cette méthode : il s'agit d'une méthode de *clustering* qui organise les documents sur une carte de dimension pré-définie, ce qui garantit une bonne utilisation de l'espace lors de la visualisation. De plus, l'organisation ainsi obtenue possède une notion de voisinage. En effet, deux documents voisins sur la carte correspondent à deux documents ayant des vecteurs de mots proches. Les cartes auto-organisatrices permettent également d'avoir différents niveaux de hiérarchie ou encore une taille de carte dynamique (Dittenbach et al. 2000), ce qui peut se révéler intéressant dans notre cadre applicatif. On va donc adapter cet algorithme pour qu'il organise le plus efficacement possible les informations textuelles. On ne détaille pas l'algorithme très connu des cartes auto-organisatrices de Kohonen (Kohonen 1995) mais uniquement les points essentiels de notre version des cartes auto-organisatrices.

- **distance.** L'algorithme des cartes auto-organisatrices de Kohonen utilise généralement la distance euclidienne. On décide ici de mettre ce choix en concurrence avec d'autres distances comme la pondération de type *tf.idf* (Sparck Jones 1972) ou encore la distance du χ^2. Dans les deux cas, il s'agit d'une pondération particulière de la distance euclidienne permettant d'augmenter ou de diminuer l'importance de certains mots. Si on note $\mathbf{x_j}$ un document du corpus et $\mathbf{m_i}$ un neurone de la carte, on peut alors définir la distance entre un document et un neurone de la façon suivante (la première expression concerne l'approche *tf.idf* et la seconde celle du χ^2) :

$$d^2_{tf.idf}(\mathbf{x_j}, \mathbf{m_i}) = \sum_k (\log(\frac{D}{n_k}))^2 (\mathbf{x_j}_k - \mathbf{m_i}_k)^2 \quad , \quad d^2_{\chi^2}(\mathbf{x_j}, \mathbf{m_i}) = \sum_k \frac{1}{f_{.k}} (\mathbf{x_j}_k - \mathbf{m_i}_k)^2 \tag{1}$$

  où :
  - k est un indice associé à un mot du corpus
  - n_k est le nombre de documents du corpus contenant le mot associé à l'indice k
  - D est le nombre de documents du corpus
  - $\sum_k \mathbf{x_j}_k = 1$ et $f_{.k} = \sum_j \mathbf{x_j}_k$

  Des tests réalisés sur de petits corpus ont révélé de meilleurs résultats pour la distance du χ^2. En effet, il semble que cette distance permette de mieux différencier les groupes (par rapport à la pondération *tf.idf* ou la distance euclidienne). Travaillant sur de corpus de taille raisonnable dans notre prototype, nous avons donc favoriser le choix de la distance du χ^2. Cependant des vérifications doivent encore être réalisées pour confirmer ce choix qui permet de rendre l'algorithme plus efficace dans le cadre de données textuelles. Nos tests sur la dissimilarité de Kullback-Leibler (non présentée ici) doivent aussi être approfondis afin de tirer des conclusions. On rappelle que cette dissimilarité permet de calculer la distance entre deux documents en utilisant uniquement les mots appartenant à l'intersection des deux documents.

- **étiquetage.** La méthode LabelSOM (Rauber 1999)(Rauber et Merkl 2001) est utilisée pour étiqueter les neurones. Cette méthode repose sur le calcul, pour

chaque neurone, d'un vecteur $\mathbf{q_i}$ représentant l'erreur de quantification. La k *ième* composante du vecteur d'erreur $\mathbf{q_i}$ est définie par l'expression suivante :

$$\mathbf{q_i}_k = \sum_{\mathbf{x_j} \in C_i} \sqrt{(\mathbf{m_i}_k - \mathbf{x_j}_k)^2} \quad (2)$$

où C_i est l'ensemble des documents associés au neurone i. Les mots sélectionnés (étiquette) pour représenter un neurone sont alors ceux qui correspondent à une faible erreur de quantification ($\mathbf{q_i}_k$ proche de 0) mais qui correspondent aussi à une valeur du vecteur de poids (représentant le neurone) supérieure à un certain seuil τ. Le dernier critère est indispensable afin de ne pas sélectionner l'absence d'un mot comme un critère valide pour étiqueter un neurone.

– **version déterministe.** L'algorithme a pour objectif de classifier les résultats d'une requête, la contrainte d'avoir une version déterministe de l'algorithme paraît donc logique. En effet, deux requêtes identiques sur le même corpus doivent aboutir à des résultats identiques. Pour cela, on utilise la version batch (ou hors-ligne) de l'algorithme ainsi qu'une initialisation fixe des neurones par les n premières données.

– **temps d'exécution.** Notre prototype doit fonctionner en temps réel, d'où une attention particulière au temps de calcul. L'algorithme des cartes auto-organisatrices est de façon générale assez coûteux en temps de calcul. Cela est dû à la sélection du neurone gagnant pour chaque donnée de l'ensemble d'apprentissage et pour chaque itération (et il faut en général un certain nombre d'itérations avant de converger). De plus la complexité de l'algorithme augmente linéairement avec les données de l'ensemble d'apprentissage et de façon quadratique avec la taille de la carte. Bien que cet algorithme ne semble *a priori* pas adapté à notre cas, son utilisation est concevable si l'on fait les hypothèses suivantes : les données d'apprentissage sont peu nombreuses (sélection des "meilleures" réponses ou échantillonnage) et la taille de la carte est raisonnable du fait qu'elle dépende du nombre de données à classer. Si ces hypothèses s'avèrent insuffisantes pour obtenir une exécution "quasi temps réel", alors on peut également envisager un pré-calcul de la carte pour une requête donnée. Une autre alternative, détaillée dans le point suivant, est de rendre l'algorithme progressif.

– **progressivité et interactivité.** Une idée intéressante est de rendre progressif et interactif l'algorithme des cartes auto-organisatrices. L'aspect progressif consiste à fournir à l'utilisateur des résultats intermédiaires à différentes étapes du calcul. Pour cela, on doit cependant attendre que l'ordre général de la carte ne soit plus modifié pour proposer à l'utilisateur une première version des résultats. En effet, cette contrainte permet de s'assurer que l'affichage des résultats suivants ne modifie pas radicalement l'ordre de la carte. Elle permet donc aussi d'éviter à l'utilisateur de produire de nouveaux efforts pour s'approprier une "nouvelle" carte. Le deuxième point consiste à rendre l'algorithme interactif en donnant à l'utilisateur la possibilité d'identifier des documents pertinents ou non pertinents

au cours de l'algorithme. L'objectif est ici de voir dans quelles mesures on peut prendre en compte ces choix de l'utilisateur au cours de l'algorithme sans calculer une nouvelle carte. Une idée serait alors d'augmenter l'importance (et donc la pondération) des éléments pertinents et à l'inverse de diminuer l'importance des éléments non pertinents (ou de les supprimer). Cet aspect progressif et interactif de l'algorithme de classification correspond, d'un point de vue interface, à un affichage dynamique des résultats avec une possibilité d'intervention de l'utilisateur. Les idées développées dans ce dernier point représentent les directions de nos travaux à venir sur l'algorithme de classification.

Des mesures ont été effectuées avec différentes distances et différents échantillonnages. Afin de rendre les résultats comparables, les paramètres de l'algorithme ont été fixés (parmi ces paramètres : carte de taille 8×8, corpus de 8570 documents représentés par des vecteurs en 164 dimensions). Les résultats ne sont pas détaillés dans cet article, mais ils permettent d'envisager l'utilisation de la distance du χ^2 et la possibilité d'un échantillonnage (si le corpus est suffisamment important). L'évaluation de ces résultats s'est réalisée sur certains critères classiques tels que l'erreur de quantification ou encore la variance intra-groupe. Cependant d'autres critères doivent être envisagés pour prendre en compte l'organisation des groupes les uns par rapport aux autres (Lesot et al. 2003).

4 Prototype SmartWeb

SmartWeb est un prototype qui s'apparente à un moteur de recherche classique du point de vue requête et base de données. Par contre l'objectif est de fournir à l'utilisateur les meilleures organisation et visualisation possibles des résultats de sa requête, sans solliciter une quelconque intervention de sa part dans le processus. On va d'abord présenter l'architecture générale du prototype, puis on s'intéressera à l'utilisation du prototype avec une métaphore de visualisation particulière : la ville. Enfin, on évoquera les performances du prototype dans le cas d'une métaphore de visualisation en 3D.

4.1 Architecture

La figure 1 représente l'architecture simplifiée du prototype. D'un côté, on a la partie serveur qui contient la base de données ainsi que les différents modèles d'interfaces. De l'autre côté se trouve la partie client qui est simplement composée de deux éléments ancrés dans une page HTML : une applet Java destinée à recevoir la partie 2D de l'interface et un navigateur VRML qui sert à l'affichage de la scène 3D. Cela a l'avantage de permettre un accès facile au prototype étant donné que le client a uniquement besoin de pouvoir exécuter une applet Java et de disposer d'un navigateur VRML.

Une représentation conceptuelle du prototype est également proposée sur la figure 2. On voit alors le cheminement de la requête et des résultats entre les différents modules. Les points essentiels sont les parties organisation, visualisation et interaction. La base de données indexe des documents de type pages Web. Les descripteurs stockés dans la

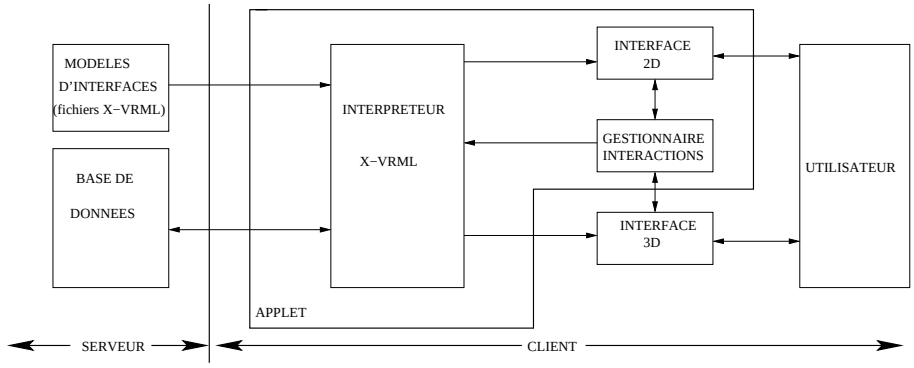

FIG. 1 – Architecture simplifiée du prototype SmartWeb.

base sont ceux couramment utilisés par les moteurs de recherche : URL, titre, résumé, vecteur de mots... Les requêtes de l'utilisateur sont interprétées puis envoyées à la base de données. Elles peuvent être simples ou booléennes. Les documents, retournés par la base de données en réponse à la requête, passent successivement par deux modules essentiels : les modules d'organisation et de visualisation. Ces deux modules sont détaillés dans les paragraphes suivants. Il est intéressant de noter une certaine indépendance entre l'organisation des données et les métaphores choisies pour la visualisation.

Module d'organisation Il n'existe probablement pas de méthode fournissant la meilleure organisation possible dans tous les cas. C'est pourquoi, ce module permet de mettre en œuvre différents types d'organisation et de sélectionner le plus adapté à chaque cas. L'organisation présentée ici est celle des cartes auto-organisatrices détaillées dans la section précédente. Cette méthode a l'avantage de respecter la proximité "sémantique" des données en se basant uniquement sur la distribution des mots. Elle permet aussi d'avoir un niveau d'abstraction si on se place au niveau des neurones. Ce niveau d'abstraction ne se base ni sur des ontologies, ni sur un étiquetage des documents. Mais on peut envisager un recoupement des étiquettes des neurones avec une ontologie ou encore l'utilisation de données sémantiques sur les documents lors de l'algorithme. Notre approche reste donc assez ouverte et assez proche de certaines approches sémantiques telles que les *Topic Maps* (Le Grand et Soto 2002). En effet, les cartes auto-organisatrices permettent de naviguer à différents niveaux d'abstraction tout comme les *Topic Maps* (les *topics* sont représentés ici par les neurones). On retrouve alors des similarités dans les méthodes de visualisation de ces deux approches, telles que les visualisations cognitives en 2D (cartes) ou en 3D (villes).

Module de visualisation Le module de visualisation reçoit les résultats organisés et un modèle d'interface (sélectionné par l'utilisateur ou automatiquement). Il fournit en sortie des données pour l'interface 2D et pour la scène 3D. Le modèle, qui définit la métaphore de visualisation utilisée ainsi que les interactions, est sélectionné dans la

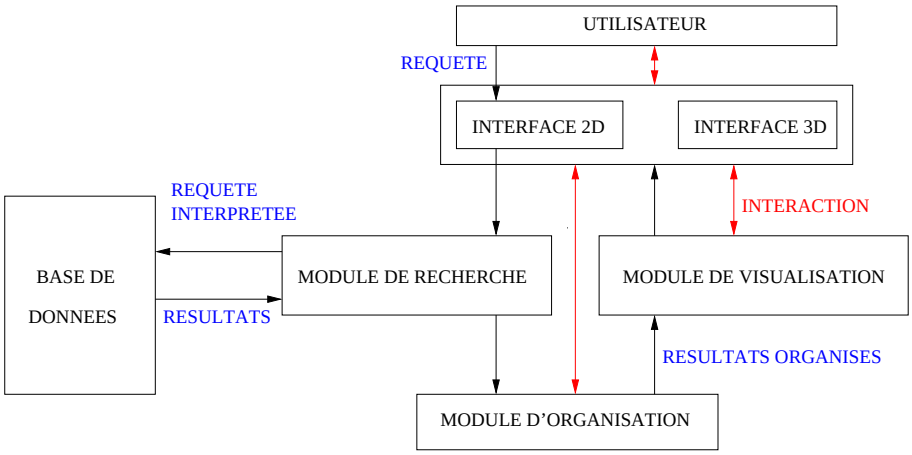

FIG. 2 – Schéma conceptuel du prototype SmartWeb.

liste des modèles d'interface du prototype. Cependant, il est possible pour l'utilisateur de personnaliser l'interface de visualisation et les interactions en créant son propre modèle. Les modèles sont exprimés en X-VRML[3] qui s'apparente à un méta-langage de plus haut niveau que le VRML permettant d'ajouter de nombreuses fonctionnalités (l'interrogation d'une base de données ou encore les itérations). Lors d'une requête, le modèle sera interprété afin de produire dynamiquement un fichier VRML qui correspond à l'interface 3D. Le module de visualisation est aussi adaptatif. En effet, le choix du modèle peut se faire en fonction de certains critères tels que le type et le nombre de résultats, l'objectif de la recherche, la catégorie de l'utilisateur.

4.2 Utilisation de SmartWeb

Définition d'une métaphore Une métaphore est la réalisation d'une association entre des paramètres graphiques de la présentation et des informations sur les documents indexés.

Le prototype possède plusieurs métaphores de visualisation telles que : une scène 3D sous forme de ville ou encore un graphique 2D avec une perspective isométrique (permettant d'avoir un troisième axe). On s'intéresse ici uniquement à une métaphore de visualisation : la ville. Le choix de cette métaphore vient essentiellement de son aspect cognitif et du fait que cette métaphore est propice à un environnement 3D. On va donc chercher à utiliser efficacement les trois dimensions afin de montrer l'apport des visualisations 3D en terme de place. Une première version de cette métaphore a été développée et soumise à des tests utilisateurs. Suite aux résultats des tests, nous avons fait évoluer la métaphore de la ville. La figure 3 donne un aperçu de la nouvelle métaphore dont les explications peuvent être classées en quatre catégories : les

[3]X-VRML est un langage basé sur XML et développé par France Telecom R&D

FIG. 3 – Prototype SmartWeb avec une métaphore de visualisation de type ville.

primitives, l'organisation, la visualisation et la navigation. Cette nouvelle version fera à nouveau l'objet de tests utilisateurs.

Primitives Chaque bâtiment de la ville représente une page Web et les bâtiments sont regroupés par quartiers (marquage au sol). La "pertinence" des pages est représentée par la taille des bâtiments, ce qui permet de distinguer assez rapidement les pages les mieux classées par rapport à ce critère.

Organisation Les quartiers sont placés sur le sol selon une grille 2D qui permet de réaliser une classification des résultats de la recherche. Cette classification, réalisée par le module d'organisation, est issue de l'application de l'algorithme des cartes auto-organisatrices. En effet, la grille 2D correspond à celle utilisée dans l'algorithme. Cette organisation possède deux propriétés intéressantes : l'unicité des pages Web (ou bâtiments) dans la ville et la notion de voisinage "sémantique" entre les différentes pages et entre les différents quartiers. Ainsi, les pages d'un même quartier sont similaires et deux quartiers voisins correspondent à deux thèmes aussi voisins que possible.

Visualisation Le choix d'une interface 3D a séduit les utilisateurs. Globalement, la visualisation 3D ne pose pas de problème car elle correspond à notre mode de vision naturel. La texture d'un bâtiment représente le contenu du document, ce qui permet d'avoir rapidement un aperçu des résultats lorsque l'on se promène dans l'environne-

Actes du Troisième Atelier Visualisation et Extraction de Connaissances (associé aux Cinquièmes Journées Francophones "Extraction et Gestion des Connaissances" - EGC'05), pages 37-47, Paris, France, 18 janvier 2005.

FIG. 4 – Métaphore de visualisation de type galerie pour les images.

ment 3D. L'affichage contextuel d'une page est accompagné de l'affichage de quatre pages voisines et donc proches de la page choisie. A partir de cette métaphore, on peut aussi interagir avec chacun des bâtiments afin de visualiser l'ensemble des images contenues dans un document via une autre métaphore 3D appelée galerie (figure 4). Cette dernière s'avère utile pour une recherche d'images et a été très appréciée.

Navigation Les tests utilisateurs révèlent que l'inconvénient majeur concerne la navigation dans la ville qui ne semble pas être triviale. Les principaux problèmes de navigation inhérents à la 3D sont liés : à une utilisation moins familière de la souris (celle proposée par les navigateurs VRML), à la sensibilité de la souris lors des déplacements dans la scène (translations et rotations), à la perte de repères dans l'environnement 3D ou encore aux difficultés d'utilisation des différents points de vue (avec le problème des occultations). On a alors décidé de simplifier certains déplacements vers des endroits stratégiques en rendant le plan 2D de la scène interactif. Cependant d'autres solutions doivent encore être trouvées afin de rapprocher la navigation à quelque chose de plus familier pour l'utilisateur. Par contre, la métaphore de type galerie a l'avantage d'avoir une navigation plus contrainte (déplacement selon un axe et angles de rotation latéraux limités), ce qui évite de se perdre dans l'environnement 3D. C'est pourquoi elle a posé moins de difficultés de navigation.

4.3 Performances

Avec le choix d'une métaphore de visualisation en 3D, se pose le problème des performances de la 3D (notamment avec un volume important de données). On a

Requêtes	Nombre de résultats	Temps d'attente (en s)	Taille (Ko)
matisse	50	6	277
picasso	173	12	658
toile	423	34	1551
artistique	565	54	2043
artistique OR toile	982	88	3415

TAB. 1 – Mesures des performances liées à la 3D pour différentes requêtes.

donc choisi 5 requêtes possédant un nombre différent de résultats. Ces requêtes ont été effectuées sur une base de données indexant des pages web sur l'art. Le tableau 1 présente pour ces 5 requêtes : le nombre de résultats issus de la base, le temps d'attente (non optimisé) pour l'utilisateur et la taille du fichier VRML (contenant la scène 3D). Le temps d'attente de l'utilisateur comprend essentiellement le temps de génération et de chargement (affichage) de la scène 3D. On précise aussi que la génération du fichier VRML nécessite préalablement d'interpréter le fichier X-VRML et de classer les données. Ces mesures ont été effectuées sur une machine standard jouant à la fois le rôle de serveur et de client. Pour ces mesures, nous avons choisi une organisation des données peu coûteuse, à savoir un simple placement selon 2 axes (noms de domaine et mots-clés). Ce choix permet d'avoir un temps de calcul raisonnable pour la classification et ainsi de ne pas trop influencer le temps d'attente mesuré. Cependant, un algorithme de classification plus coûteux tel que les cartes auto-organisatrices de Kohonen, augmente le temps de calcul de la classification mais pas celui de la construction de l'interface (génération et affichage de la scène 3D).

5 Conclusion

Cet article présente une méthode d'organisation et de visualisation 3D des résultats issus d'une requête. L'organisation repose sur une carte auto-organisatrice particulière de par la distance utilisée (distance du χ^2). Cependant nos tests doivent encore être enrichis pour valider ce choix. La visualisation 3D proposée ici est une représentation sous forme de ville. Ce type de métaphore s'avère très efficace pour représenter des données organisées. Un aspect essentiel du prototype est le couplage entre la scène 3D et l'interface 2D. On rappelle également que l'interface graphique est générée dynamiquement. Elle est aussi interactive et adaptative.

L'organisation des résultats peut encore être améliorée en rendant l'algorithme des cartes auto-organisatrices progressif et interactif, ou encore en y intégrant une classification ascendante hiérarchique. Ces ajouts nécessitent cependant une réflexion sur les modifications à apporter à la métaphore de visualisation et notamment aux liens entre les différents niveaux hiérarchiques. L'idée est de dégager les différents sens sémantiques de la requête. Deux autres objectifs sont la diminution du temps d'exécution et la simplification de la navigation dans une scène 3D.

Références

Boyack K., Wylie B. et Davidson G. (2002), Domain visualization using vxinsight for science and technology management.

Dittenbach M., Merkl D. et Rauber A. (2000), Using growing hierarchical self-organizing map, Proceedings of European Symposium on Artificial Neural Networks, pp 7-12.

Kohonen T. (1995), Self-Organizing Maps, Springer.

Kohonen T., Kaski S., Lagus K., Salojärvi J., Honkela J., Paatero V. et Saarela A. (2000), Self Organization of a Massive Document Collection, IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, 11(3) 574-585.

Le Grand B. et Soto M. (2002), Visualisation of the Semantic Web : Topic Maps Visualisation, Proceedings of the 6th Int. Conf. on Information Visualization.

Lesot M.-J., d'Alché-Buc F. et Siolas G. (2003), Evaluation des cartes auto-organisatrices et de leur variante à noyaux, Actes de la conférence CAp.

Rauber A. et Merkl D. (2001), Automatic Labeling of Self-Organizing Maps for Information Retrieval, JSRIS, 10(10) 23-45.

Rossi A.M. et Varga M. (1999), Visualization of Massive Retrieved Newsfeeds in Interactive 3D, Proceedings of Int. Conf. on Information Visualization, pp 12-17.

Sparck Jones K. (1972), A statistical interpretation of term specificity and its application in retrieval, Journal of Documentation, 28(1) 11-20.

Skupin A. et Fabrikant S. (2003), Spatialization Methods : A Cartographic Research Agenda for Non-Geographic Information Visualization, Cartography and Geographic Information Science, 30(2) 99-119.

Wiza W., Walczak K. et Cellary W. (2004), Periscope - A System for Adaptive 3D Visualization of Search Results, Proceedings of the Int. Conf. on 3D Web technology, pp 29-40.

Actes du Troisième Atelier Visualisation et Extraction de Connaissances (associé aux Cinquièmes Journées Francophones "Extraction et Gestion des Connaissances" - EGC'05), pages 37-47, Paris, France, 18 janvier 2005.