Evaluation de la pertinence dans un système de recommandation sémantique de nouvelles économiques

Résumé. De nos jours dans les secteurs commerciaux et financiers la veille électronique d'articles économiques est cruciale. Maintenir une veille efficace implique de cibler les articles à consulter, car la charge d'information est importante. Pour répondre à cette problématique, nous proposons un système de recommandation d'articles novateurs, car il s'appuie sur l'intégration d'une description sémantique des items et des profils basés sur une modélisation ontologique des connaissances. Nous appuyons notre système de recommandation sur un modèle vectoriel intrinsèquement efficace que nous avons perfectionné pour pallier la confusion native de ces modèles entre les notions de similarité et de pertinence qui ne permet pas de prendre en compte les effets de la différence dans la précision des descriptions sémantiques des profils et articles, sur la perception de la pertinence pour l'utilisateur. Nous présentons donc dans cet article une nouvelle méthode d'évaluation de la pertinence adaptée au modèle vectoriel.

1 Introduction

Afin de rester en phase avec les tendances actuelles du marché, le processus de prise de décision dans le domaine économique nécessite la centralisation et l'apport de grandes quantités d'informations. Pour cela, les hommes d'affaires, les entrepreneurs et les vendeurs doivent parfaitement connaitre leur environnement. Cela signifie qu'il faut maintenir une veille économique constante facilitent l'identification des perspectives d'affaires, permettant de décrocher de nouveaux contrats. Cette veille incontournable est cependant complexe à assurer, c'est pourquoi nous proposons un outil efficace de recommandation d'articles régionaux d'actualités économiques, utilisant des représentations sémantiques communes des connaissances afin de décrire les besoins de l'utilisateur et les informations permettant d'y répondre. La surcharge d'informations est un problème bien connu, dans le domaine de recherche d'information et des systèmes de recommandation. L'efficassitéde notre system repose sur son adéquation aux besoins des utilisateurs, ainsi avons-nous mené une enquête auprès des clients afin de définir les critères qui pourraient permettre la personnalisation du contenu de la revue. Les résultats de l'enquête ainsi que la connaissance des experts du domaine ont permis de mettre en avant les trois critères principaux suivants, les Thèmes (principaux évènements économiques traités dans l'article), les Secteurs économiques dont traitent l'article et les Localisations. Cette étude

nous permet de baser de manière adéquate notre système de recommandation sur le contenu des articles économiques. Pour cela des méthodes de Traitement Automatisé du Langage Naturel, TALN, ainsi qu'un travail d'indexation manuel sont utilisées afin d'affecter au mieux les trois vocabulaires contrôlés décrivant ces critères principaux d'indexation. Ces critères sont communs à la description des profils utilisateur ainsi que des articles, ainsi nous pouvons rapprocher ces deux indexations selon des meusrures de similarité et de pertinence.

Cet article est organisé de la façon suivante. La section 2 présente l'état de l'art. Dans la section 3, nous présentons les définitions. La section 4 présente notre approche, c'est à dire les différentes méthodes de comparaison mise en place. La section 5 propose une évaluation des algorithmes, et enfin, la section 6 propose une discussion autour des résultats avant la conclusion du travail et l'ouverture sur des travaux à venir.

2 Etat de l'art

Deux types principaux de systèmes de recommandation sont distingués : les systèmes dits de filtrage collaboratif et les systèmes de recommandation basés sur le contenu. La synthèse de K. Rao Rao et Talwar (2008) propose une comparaison générale des principaux avantages et inconvénients de chacun des deux types de systèmes de recommandation. Les avantages des systèmes de recommandation basés sur le contenu dans le cadre de la recommandation d'articles d'actualité sont également développés dans Liu et al. (2010).

Notre besoin de recommandation rapide d'articles économiques, chaque jour nouveaux, exclut la solution de systèmes à base de filtrage collaboratif, car ces systèmes nécessitent qu'un nombre suffisant d'utilisateurs aient lu chacun des articles avant d'être capables de les recommander ils peinent donc à recommander de nouveaux items. En outre, nous devons être en mesure de recommander des profils d'utilisateurs très particuliers, car certains besFoins clients peuvent être uniques. Cela n'est pas possible avec un système de filtrage collaboratif car la recommandation des items se fait en fonction des appétences de personnes ayant un profil similaire. Nous nous sommes donc tournés vers les systèmes basés sur le contenu, plus adaptés à nos besoins.

Il existe de nombreux systèmes de recommandation qui fonctionnent sans utilisation de connaissances supplémentaires Liu et al. (2010), Billsus et Pazzani (1999) et Resnick et al. (1994) mais IJntema et al. (2010) ont montré que l'utilisation de connaissances extérieures peut améliorer la recommandation. On parle alors de systèmes de recommandation basés sur la sémantique. Les systèmes de recommandation basés sur la sémantique constituent un cas particulier des systèmes basés sur le contenu. Ils utilisent des connaissances lexicales Getahun et al. (2009), comme WordNet Fellbaum (1998), ou alors des connaissances de domaine Middleton et al. (2004), voire une combinaison des deux IJntema et al. (2010) dans l'objectif d'améliorer les performances du système. Les ontologies utilisées par ces systèmes existent déjà ou sont créées à la main, et maintenues. Contrairement à ces systèmes, notre base de connaissances du domaine est utilisée comme un index, les articles et des profils y sont définis.

Le modèle vectoriel Salton (1971) est une méthode qui permet de comparer deux vecteurs dans un espace vectoriel. Beaucoup de systèmes de recommandation basés sur le contenu l'utilisent IJntema et al. (2010) Middleton et al. (2004) Getahun et al. (2009) Billsus et Pazzani (1999) Ahn et al. (2007) lors de la réalisation des tâches de comparaison (que ce soit entre items, ou entre item et profil). Cette méthode d'algèbre linaire présente deux principaux avan-

tages : 1) fournir un résultat non binaire, permettant donc d'ordonner les résultats des systèmes de recommandation, 2) permettre des calculs rapides et une bonne résistance à la montée en charge.

Des méthodes utilisées en recherche d'informations peuvent être utilisées afin de prendre en compte cette connaissance tout en utilisant une modélisation vectorielle. L'approche proposée par Voorhees (1994) utilise la base de connaissance lexicale WordNet Fellbaum (1998) afin de permettre une meilleure gestion de l'hétérogénéité du langage naturelet donc une amelioration de la compréhension des besoins de l'utilisateur. L'idée est d'ajouter de l'information aux requêtes des utilisateurs (*expansion de requêtes*). Cette méthode permet d'augmenter le rappel et donc les performances globales du système.

Nous avons transposé cette méthode aux systèmes de recommandation. Nous la nommerons *expansion profil* dans la suite de l'article. Middleton et al. (2004) utilise cette méthode sans la nommer. IJntema et al. (2010) y a également recours, mais contrairement à Middleton, il utilise d'autres relations de l'ontologies que is_a pour étendre le profil de l'utilisateur. Contrairement aux méthodes précédentes, dans notre approche, nous distinguons les notions de similarité et de pertinence, généralement confondues dans les systèmes utilisant une modélisation vectorielle. Définir la pertinence comme une similarité ne permet pas de prendre en compte les différents degrés de spécificité dans la description du besoin des utilisateurs. Cette description est pourtant rendu possible par l'utilisation de connaissances externes au système. Nous proposons donc une mesure d'évaluation de la pertinence utilisant les notions de similarité, mais prenant en compte la perception de la pertinence par l'utilisateur.

3 Définitions

Dans cette section nous introduisons 1) les notions d'ontologie et de base de données qui nous servent à gérer sémantiquement les connaissances métier et lexicale dans le système ainsi qu'a décrire les articles et profils. 2) les notions de similarité et de pertinence ainsi que de modèle vectoriel que nous utilisons lors de la comparaison des dites descriptions sémantiques.

3.1 Ontologie et Base de connaissances

Dans cet article nous utilisons la modélisation de base de connaissances définie par Ehrig et al. (2004).

Soit $O = (C, T, \leq_C, \leq_T, R, A, \sigma_A, \sigma_R, \leq_R, \leq_A)$ une ontologie avec C, T, R, A des ensembles disjoints de concepts, types de données, relations et attributs, $\leq_C, \leq_T, \leq_R, \leq_A$ les hiérarchies de classes, type de données, relation et attributs, et σ_A, σ_R des fonctions qui produisent une signature pour chaque $\sigma_A : A \to C \times T$ attribut et $\sigma_R : R \to C \times C$ relation.

Soit $K = (C, T, R, A, I, V, i_C, i_T, i_R, i_A)$ une base de connaissances avec C, T, R, A, I, V des ensembles disjoints de concepts, type de données, relations, attributs, instance et valeurs de données. i_C la fonction d'instanciation des classes $i_C : C \to 2^I$. i_T est la fonction d'instanciation des types de données $i_T : T \to 2^V$. i_R est la fonction d'instanciation des relations $i_R : R \to 2^{I \times I}$. i_A est la fonction d'instanciation des attributs $i_A : A \to 2^{I \times V}$.

3.2 Similarité

$Similarite(x, y) : I \times I \rightarrow [0, 1]$ est une fonction qui permet d'évaluer le degré de similarité entre deux objets x et y, dans notre cas x est un article et y un profil. Cette fonction doit satisfaire certaines propriétés 1) La Positivité $\forall x, y \in I\ Similarite(x, y) \geq 0$; 2) La Réflexivité $\forall x, y \in I\ Similarite(x, x) = 0$ et 3) La Symétrie,$\forall x, y \in I\ Similarite(x, y) = Similarite(y, x)$. Nous avons choisi de conserver l'analogie classique avec les mesures de distances et donc de conserver l'axiome de symétrie dans notre définition (Richter, 1993), car nous utilisions des algorithmes bien connus de comparaisons de vecteurs qui sont symétriques comme la similarité Cosinus, la similarité Jaccard et la distance Euclidienne.

3.3 Pertinence

$Pertinence(x, y) : I \times I \rightarrow [0, 1]$ est une fonction qui permet de mesurer le degré de pertinence d'un article x vis-à-vis d'un profil y. Cette mesure de pertinence doit aussi respecter les propriétés de positivité et réflexivité.

La pertinence est une notion provenant des sciences de l'information, largement utilisée dans le domaine de la recherche d'informations et des systèmes de recommandation. Dans notre cas, la pertinence n'est pas binaire, un article peut plus ou moins correspondre au besoin d'informations d'un utilisateur, c'est pourquoi nous utilisons le modèle vectoriel pour l'estimer. Contrairement aux approches classiques confondant les notions de similarité et de pertinence Salton (1971), nous les distinguons. La mesure de pertinence proposée dont les caractéristiques sont présentées ici, est non-symétrique, car nous considérons que comparer un profil et un article n'est pas la même chose que de comparer deux articles, ou deux profils, et que donc, les spécificités de chacun doivent être prises en compte.

3.4 Approche vectorielle

Le modèle vectoriel, en anglais, Vector Space Model, VSM Salton (1971) est utilisé dans notre prototype, car il permet de mesurer le degré de correspondance entre un article et un profil et donc de fournir une réponse ordonnée en fonction du degré de correspondance. Cette méthode d'algèbre linaire permet de très bonnes performances tant en temps de calcul qu'en gestion de la montée en charge. Les articles et les profils sont représentés par des vecteurs dans un espace vectoriel. Chaque dimension de l'espace est une instance potentielle de critères utilisés pour décrire les articles et profils. Plusieurs méthodes peuvent être utilisées pour comparer les vecteurs, la plus commune est la similarité Cosinus, mais il existe aussi la similarité Jaccard ou la distance euclidienne pour ne citer que les principales. La similarité Cosinus entre deux vecteurs $\overrightarrow{a}$ et $\overrightarrow{p}$ est defini comme le Cosinus de l'ange Θ entre les deux vecteurs, elle peut etre défini par :

$$Similarite(\overrightarrow{a}, \overrightarrow{p}) = cos\Theta = \frac{\overrightarrow{a}.\overrightarrow{p}}{|\overrightarrow{a}|.|\overrightarrow{p}|} = \frac{\sum_{x=1}^{t} i_{a,x} * i_{p,x}}{\sqrt{\sum_{x=1}^{t} i_{a,x}^2} * \sqrt{\sum_{x=1}^{t} i_{p,x}^2}}$$

Avec C l'ensemble de tous les concepts et C' l'ensemble des concepts définis comme critère descriptif des articles et profils, c'est-à-dire critère d'indexation, tel que $C \in C'$, I l'ensemble de toutes instances des concepts de C et I' l'ensemble de toutes les instances des concepts

de C. Nous définissons un article comme un vecteur d'instances de critères d'indexation tels que $\overrightarrow{a} =< i_1, i_2, ..., i_n >$ avec $i_x \in I'$ et $x \in [0; |I'|]$. Un profil se définit de façon similaire $\overrightarrow{p} =< i_1, i_2, ..., i_m >$ avec $i_x \in I'$ et $x \in [0; |I'|]$.

4 Approche

Une ontologie modélisant les connaissances du domaine a été définie et peuplée avec l'aide d'experts afin d'y appuyer les descriptions sémantiques des profils et articles. Dans un système classique de recommandation basé sur le contenu, deux tâches principales sont distinguées. La première est l'indexation qui consiste à créer une représentation des besoins des utilisateurs et du contenu des articles. La qualité de l'analyse du contenu est alors importante pour la qualité de l'indexation. Ainsi, notre système est semi-supervisé par des experts, afin d'éviter autant que possible des erreurs d'indexation. La deuxième tâche est la comparaison des modélisations d'articles et de profils afin de recommander au mieux les articles aux utilisateurs.

4.1 Indexation

La partie concernant l'indexation s'appuie sur des ontologies afin de pouvoir indexer d'une part les articles et d'autre part les profils des utilisateurs. Nous allons détailler, dans cette partie, ces trois éléments.

4.1.1 Base de connaissances

La base de connaissances utilisée pour ce système est composée de plusieurs ontologies selon les principes des ontologies modulaires d'Aquin et al. (2009). Une ontologie de niveau supérieur est utilisée pour gérer l'information partagée par tous les domaines d'application. Des concepts de haut niveau sont définis pour gérer des localisations, des informations géo-spatiales, la temporalité, les événements, etc… Une ontologie de domaine est utilisée pour gérer les connaissances spécifiques au domaine économique. D'autres ontologies sont utilisées pour gérer les articles, les profils et les ressources lexicales (utilisées lors de l'analyse des articles par un gazetteer [1]).

4.1.2 Indexation

Afin d'effectuer la recommandation d'articles aux clients, le système a besoin d'une représentation du contenu de chaque article, ainsi que de la représentation des besoins de chaque client. La base de connaissances est utilisée comme index, ce qui permet de baser les descriptions des articles et profils sur un référentiel commun. Ils sont alors représentés par des instances dans la base de connaissances.

Indexation d'articles. Cette indexation permet une représentation compréhensible par la machine du contenu de chaque article afin de les comparer avec les profils. Les informations non structurées contenues dans les articles sont analysées. Deux types d'information peuvent

1. Processus qui permet la détection de termes contenus dans un dictionnaire

être distingués : les informations *explicites* (par exemple les lieux, les personnes, les organisations, etc) et les informations *implicites* (par exemple, le thème de chaque article ou les secteurs économiques concernés). La plate-forme GATE Cunningham (2002) est utilisée pour l'analyse des articles et l'extraction des informations. Les résultats de l'analyse sont vérifiés à la main, corrigés et validés, par les redacteurs. Pour chaque article analysé, une instance du concept article ainsi que des instances de la relation « isAbout » sont créées dans la base de connaissance afin de faire le lien entre l'article et les critères permettant de le qualifier. Ces relations forment une représentation sémantique compréhensible par la machine des articles.

4.1.3 Indexation des profils

Les vendeurs, en charge de la compréhension des besoins de chaque client, proposent une période d'essai gratuite qui permet à un expert de créer un premier profil à chacun des clients. Cela permet d'éviter le problème du démarrage à froid, commun aux systèmes de recommandation basés sur le contenu Rao et Talwar (2008).
Le processus d'indexation des profils est le même que celui des articles. Une instance de profil est donc créée dans la base de connaissances pour chaque profil établi par les experts. Les relations «isInterestedIn » sont créées entre l'instance de profil et les instances de critères, ce qui permet de créer une représentation sémantique des besoins et intérêts de chaque utilisateur compréhensible par la machine.

4.2 Recommandation

La tâche de recommandation est basée sur la comparaison entre le profil et les articles disponibles selon leur index commun défini par la base de connaissances. Les méthodes classiques de recherche d'information ou de recommandation utilisant une modélisation vectorielle déduisent directement la pertinence de la mesure de similarité entre le vecteur représentant le profil et celui représentant l'article. La base théorique, est que le profil peut être considéré comme un article idéal, donc plus un article est similaire à cet article idéal (profil) plus il est pertinent, plus il correspond aux intérêts et besoins de l'utilisateur. Nous présentons dans cette partie notre modèle vectoriel puis les mesures de pertinence d'articles pour des profils utilisateurs.

4.2.1 Le modèle vectoriel, sans l'apport des connaissances

Il nous est possible d'utiliser le modèle vectoriel avec notre modélisation des articles et profils de la facon suivante :

$$SimilariteF(\overrightarrow{a}, \overrightarrow{p}) = \frac{\sum \omega_c Similarite_c(\overrightarrow{a_c}, \overrightarrow{p_c})}{\sum \omega_c}$$

$Similarite_c(\overrightarrow{a}, \overrightarrow{p})$ étant la mesure de similarité entre le profil $\overrightarrow{p}$ et l'article $\overrightarrow{a}$ pour le critère spécifique c, tel que $c \in \{Themes, Secteur, Location\}$. Nous utiliserons lors de l'évaluation alternativement les mesures, Cosinus, Jaccard et Euclide en tant que mesure de similarité $Similarite_c(\overrightarrow{a}, \overrightarrow{p})$. ω_c est le coefficient de pondération défini pour le critère c. $\forall i_{x,c} \in I'_c$, $\overrightarrow{p_c} =< i_{1,c}, i_{2,c}, ..., i_{n,c} >$ et $\overrightarrow{a_c} =< i_{1,c}, i_{2,c}, ..., i_{m,c} >$ avec n et $m \in [0; |I'_c|]$. Un ou plusieurs concepts peuvent être définis pour chaque critère. Pour le critère localisation

par exemple, nous avons les concepts, pays région, département et ville. Ainsi donc dans la liste des instances disponibles pour la description des articles et profils sur chaque critère il est possible d'utiliser toutes les instances de tous ces concepts. Notons, qu'ici nous ne tenons pas compte des connaissances externes lors de la création des vecteurs. Ainsi si un article concerne Dijon et qu'un profil s'intéresse à la Côte d'Or, le système ne sachant pas que Dijon est en Côte d'Or, la similarité mesurée à l'aide de cette méthode sera nulle et donc l'article considéré comme non pertinent pour l'utilisateur. La methode de creation des vecteurs permettant la prise en compte des connaissances externe sera l'objet de la section suivante.

4.2.2 Le modèle vectoriel avec utilisation des connaissances

Dans notre approche, nous utilisons un vecteur différent pour les instances de chaque critère. Par exemple pour le critère secteur économique, nous pourrions avoir un vecteur de profil $\overrightarrow{p_s} =< Transport, Informatique, Electronique >$. Cela nous donne la possibilité de pondérer l'importance des critères.
Par ailleurs, nous nous basons sur les apports de IJntema et al. (2010) précisés dans l'état de l'art et nous choisissons de donner à l'*expansion profil* la forme d'ajout d'instances au vecteur de description. Par exemple, si le profil de l'utilisateur U montre un intérêt pour l'entreprise Co et que dans la base de connaissances une relation symétrique « is_aSubsidiaryOf » est instanciée avec une autre société Co', il est possible d'ajouter cette société à son profil.

Pour comparer les instances d'articles et de profils nous les modélisons à l'aide de vecteurs contenant les instances de concepts définis comme étant des critères d'indexation ayant une relation avec l'instance d'article ou de profil (via les relations, « isAbout » ou « isInterestedIn »). Voorhees (1994) montre que toutes les dimensions sont orthogonales dans le modèle vectoriel et qu'ainsi, tous les éléments de chaque vecteur sont considérés comme indépendants. Ce qui n'est pas le cas de son lexique, ni pour les instances utilisées dans notre système. Dans sa méthode, les relations méronomiques [2] et synonymiques définies dans WorldNet sont ajoutées au vecteur du lexique en relation avec les mots qu'il contient déjà. Cependant, la similarité entre un profil intéressé par la Bourgogne par exemple et un article sur Dijon sera très faible avec cette méthode, alors qu'elle devrait être relativement forte. Afin de pallier ce problème, nous étendons les articles en plus des profils. De plus afin de limiter la taille du vecteur, nous ajoutons les ancêtres de l'instance sélectionnée et non les descendants (par des relations méronomiques). Notre méthode est donc analogue aux méthodes de recherche de l'ancêtre commun dans un graphe pour évaluer la distance sémantique entre deux nœuds.

Par ailleurs, nous voulons gérer les distinctions de précision entre un profil et un article. Le fait d'avoir un article plus précis qu'un profil, n'a pas les mêmes répercussions sur la perception de la pertinence que le fait d'avoir un profil plus précis qu'un article et donc cela doit être pris en compte lors de l'évaluation de la pertinence.

4.2.3 Prise en compte de la précision des descriptions

Dans cette partie, nous apportons la distinction entre *pertinence* et *similarité*. Nous nous concentrons ici sur la gestion de la différence de précision entre la définition des profils et des

2. Désigne une sous partie, par exemple *toit* est un méronyme de *maison*.

articles et son influence sur la mesure pertinence. Par exemple, si un article traite de Dijon et un profil montre un intérêt pour la Bourgogne, la pertinence doit être plus élevée que dans le cas inverse. Afin de résoudre ce problème, nous utilisons un vecteur intermédiaire pour chaque critère. Le sous-vecteur $\overrightarrow{s_c}$ est composé des instances communes entre le vecteur de l'article $\overrightarrow{a_c}$ et celui du profil $\overrightarrow{p_c}$. Dans la hiérarchie des concepts, les concepts les plus généraux englobent d'autres concepts, plus spécifiques, il en va de même dans la hiérarchie d'instances. Les instances hautes dans la hiérarchie sont moins précises que les instances basses. Si un article traite d'instances de bas niveau et qu'un profil s'intéresse à des instances de haut niveau (de la même branche) la pertinence de l'article pour le profil doit être plus élevée que dans le cas contraire. Car si l'article est plus général que le profil alors il y a une perte de précision par rapport au besoin de l'utilisateur et qui doit être répercuté par une perte de pertinence.

$$Pertinence_c(\overrightarrow{a_c}, \overrightarrow{p_c}) = \frac{\omega'_{1,c} \times Similarite_c(\overrightarrow{a_c}, \overrightarrow{s_c}) + \omega'_{2,c} \times Similarite_c(\overrightarrow{p_c}, \overrightarrow{s_c})}{\omega'_{1,c} + \omega'_{2,c}}$$

Avec S_c le sous-ensemble commun d'éléments de l'ensemble d'instances en relation à la fois avec le profil $I'_{p,c}$ et l'article $I'_{a,c}$; $S_c = I'_{p,c} \cap I'_{a,c}$. $\forall i_{x,c} \in S_c$ le vecteur $\overrightarrow{s_c}$ est composé des éléments de l'ensemble S_c ; $\overrightarrow{s_c} =< i_{1,c}, i_{2,c}, ..., i_{t,c} >$.

Avec cette méthode, il est possible de pondérer de plusieurs façons la différence de précision entre profils et articles, afin de la gérer différemment. Dans notre cas nous utilisons $\omega'_{1,c} = 1$ et $\omega'_{2,c} = 4$ car nous considérons que la perte de précision du profil par rapport à l'article ne doit pas influencer plus de 20% du résultat. De plus, la perte de précision de l'article par rapport au profil doit influencer fortement le résultat, ici 80%. Il est toutefois possible de modifier ces valeurs, et il est aussi possible de les gérer de façon différente selon le critère.

$$PertinenceF(\overrightarrow{a}, \overrightarrow{p}) = \frac{\sum \omega_c * Pertinence_c(\overrightarrow{a_c}, \overrightarrow{p_c})}{\sum \omega_c}$$

La pertinence finale $PertinenceF(\overrightarrow{a}, \overrightarrow{p})$ est la somme des mesures de pertinence pour chacun des critères, éventuellement pondérée. Cette mesure est utilisée dans notre prototype pour trier les résultats (articles) proposés à l'utilisateur en fonction de son profil.

Avec cette méthode, la valeur de pertinence pour le cas A.2 (cf. figure 1) est plus élevée que dans le cas A.1, car dans le cas A.2, a et p ont des ancêtres communs plus généraux que dans le cas A.2 (c'est le problème développé dans la section 4.2). En outre, les cas B1 et B2 illustrent le problème de la précision. Avec notre méthode asymétrique, la valeur de similarité entre a et p dans le cas B.1 est plus élevée que dans le cas B.2. Les besoins des utilisateurs sont plus spécifiques que les informations de l'article (sur ce critère) sont très générales par rapport aux besoins de l'utilisateur. Donc, l'article est moins pertinent pour l'utilisateur.

5 Expérimentations

Nous voulons comparer la méthode de création de vecteurs et son influence sur les résultats de pertinence.Pour la comparaison entre les vecteurs, nous avons utilisé des méthodes classiques pour le modèle vectoriel, à savoir la similarité Cosinus, la similarité de Jaccard et la distance euclidienne.

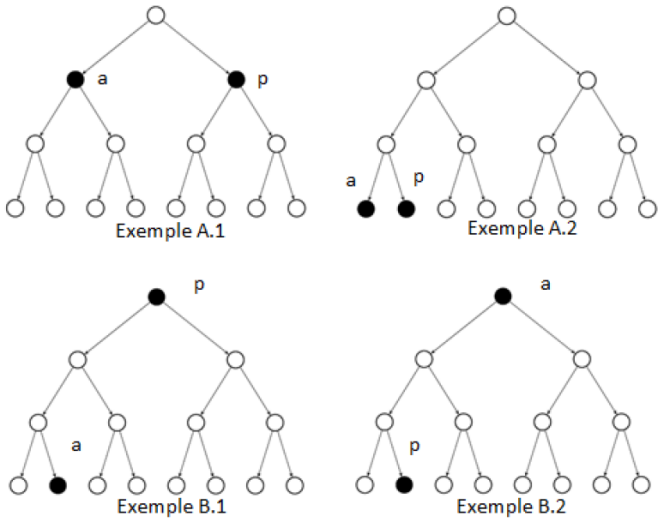

FIG. 1 – *Exemple de Profils et d'articles pour un critère donné*

Nous avons comparé trois algorithmes de recommandation, de deux manières différentes (évaluation binaire et l'évaluation graduée). Le premier algorithme est le modèle vectoriel classique sans extension des vecteurs nous le nommons méthode *C*. Le second est notre méthode modifiée par une expansion des deux vecteurs articles et profils par l'ajout d'instances venant du haut de la hiérarchie que nous appelons méthode *B*. Le troisième est une nouvelle méthode de mesure de pertinence nous la nommons méthode A. Cette méthode utilise un sous vecteur commun et permet de gérer la différence de précision entre les profils et articles. Pour ces trois méthodes, nous avons utilisé les algorithmes Jaccard, Euclide et Cosinus pour la comparaison des vecteurs dans un espace vectoriel.

5.1 Evaluation binaire

Pour évaluer la recommandation binaire un ensemble de 10 profils et 70 articles a été choisi. Une sélection (faite à la main) des articles pertinents a été réalisée pour chaque profil par des experts. Puis, nous avons comparé les résultats de la sélection par un expert aux résultats des algorithmes de recommandation. Pour effectuer une évaluation binaire, nous avons besoin de résultats binaires à partir d'algorithmes de recommandation. Or ils fournissent des articles de façon triée à l'aide d'une valeur de pertinence (entre 0 et 1). Donc, nous avons défini un seuil au-delà duquel un item est recommandé et en dessous duquel il ne l'est pas. Le seuil de 0,5 a été choisi pour l'évaluation de la pertinence binaire. La mesure d'évaluation de la recommandation binaire classique est le calcul de la précision et du rappel. Précisions, rappel et F-mesure : sont des fonctions qui produisent un résultat entre 0 et 1.

L'optimisation de la précision [3] et du rappel sont au cœur des problèmes dans les systèmes

3. Dans cette section, le mot *précision* fait uniquement référence à la mesure de la précision de la recommandation, dans le sens précision / rappel.

de recommandations. Afin de considérer les deux, (Lewis et Gale, 1994) ont proposé une mesure simple : la F-mesure. La F-mesure est une combinaison pondérée de précision et de rappel qui produisent des scores allant de 0 à 1.

Algorithmes	Précision	Rappel	F1-mesure	Tau de Kendall	Rho de Spearman
COSINUS A	0.856	0.971	**0.910**	**0.836**	**0.898**
COSINUS B	0.916	0.453	0.607	0.830	0.894
COSINUS C	0.883	0.181	0.301	0.713	0.694
JACCARD A	**0.928**	0.588	0.720	0.836	0.896
JACCARD B	0.883	0.150	0.256	0.819	0.886
JACCARD C	0.883	0.150	0.256	0.712	0.693
EUCLIDE A	0.566	0.971	0.715	0.728	0.817
EUCLIDE B	0.396	**0.985**	0.565	0.649	0.734
EUCLIDE C	0.549	0.495	0.521	0.549	0.615

TAB. 1 – *Résultat des mesures d'évaluation binaires et des mesures de corrélation de rang pour chaque algorithme.*

Les résultats de l'évaluation de la recommandation binaire présentés dans la table 1 montrent que pour chaque méthode de comparaison (Jaccard, Cosinus, Euclide) lorsque les vecteurs sont étendus vers le haut avec notre méthode B, les résultats sont au moins aussi bons que les vecteurs classiques C pour la F1-mesure. Le constat est identique pour la comparaison entre les vecteurs étendus B et notre méthode avec un sous-vecteur commun A. Le procédé A a d'au moins aussi bons résultats que la méthode B en ce qui concerne la F1-mesure. Notre méthode A utilisant la similarité cosinus donne les meilleurs résultats, avec une bonne précision et un bon rappel, et surtout le meilleur équilibre entre les deux. Nous pouvons observer une perte de précision entre les méthodes B et A lorsque la méthode de comparaison de vecteurs utilisée est la similarité Jaccard. Ce problème de perte de précision a déjà été expliqué par Voorhees (1994) avec sa propre méthode d'expansion du vecteur. En effet, l'expansion des vecteurs vise l'amélioration du rappel et comme le montrent les résultats, cela peut avoir un coût en précision.

5.2 Évaluation de l'ordre

Nous évaluons la pertinence de l'ordre proposé à partir du même jeu de données. Un classement à la main des articles en fonction de leur pertinence par rapport à chacun des profils a été réalisé par des experts. Puis, ce classement a été comparé avec les résultats des algorithmes de recommandation. Pour évaluer la pertinence graduée, nous utilisons les deux mesures de corrélation linéaire de rang les plus populaires : le rho de Spearman et le tau de Kendall. Ces deux métriques produisent des scores allant de -1 à 1. 0 étant l'absence de similitude, 1 la similitude complète et -1 l'inverse.

L'évaluation graduée présentée par la table 1 montre des résultats équivalents à l'évaluation binaire. B est plus performant que C et notre méthode A donne les meilleurs résultats, et ce pour toute méthode de comparaison des vecteurs (Jaccard, Cosinus, Euclide). Avec les résultats des deux méthodes d'évaluation, nous pouvons conclure que notre seconde méthode (A) améliore la F1-mesure et de donne le meilleur classement d'articles. Cette méthode de calcul pertinence fournit les meilleurs résultats de recommandation quand elle est utilisée avec la comparaison

de vecteurs, cosinus. Cependant, nous tenons à noter que la méthode vectorielle peut impliquer des effets de bord lorsque le volume de données devient conséquent.

6 Conclusion

Dans cet article, nous avons présenté l'adaptation d'un système basé sur le modèle vectoriel de recommandation à notre méthode spécifique d'indexation qui définit sémantiquement les articles et les profils dans une base de connaissances par l'intermédiaire des relations avec les connaissances du domaine prédéfinies dans celle-ci. Nous avons présenté notre approche qui répond aux manques de l'état de l'art sur les points de gestion du degré de precision du besoin et de comparaison de ces degrés de précision du besoin avec l'offre. Nous avons exposé la tâche spécifique de comparaison et d'évaluation de la pertinence. Enfin, nous avons évalué nos algorithmes en utilisant à la fois une méthode d'évaluation binaire et un de corrélation de rang afin de montrer les apports de notre approche.

Nous avons évalué notre approche selon deux méthodes qui démontrent que notre seconde méthode (avec un sous-vecteur commun) donne le meilleur classement d'articles, notamment quand elle est utilisée avec la comparaison de vecteurs cosinus. Nous projetons une évaluation sur jeu de données plus vaste et prenant également en compte les comportements de l'utilisateur lors de l'utilisation de l'outil de recommandation. De plus, la base de connaissances étant en cours d'enrichissement, les prochaines évaluations devraient être encore meilleures.

Par ailleurs, nous souhaitons améliorer la recommandation en nous appuyant sur des algorithmes provenant de l'étude des graphes afin de calculer la pertinence entre les profils et les articles. L'information que nous utilisons est déjà structurée dans l'ontologie (qui transmet bien mieux la sémantique que le modèle vectoriel), il semble donc inutile de restructurer l'information sous forme de vecteurs pour effectuer des comparaisons entre les articles et les profils à moins que cela n'apporte un réel gain en temps de calcul. Certaines approches permettant de comparer des instances dans une base de connaissances existent déjà Albertoni et Martino (2006), Ehrig et al. (2004).

Références

Ahn, J.-w., P. Brusilovsky, J. Grady, D. He, et S. Y. Syn (2007). Open user profiles for adaptive news systems : help or harm ? pp. 11. ACM Press.

Albertoni, R. et M. D. Martino (2006). Semantic similarity of ontology instances tailored on the application context. In *Lecture Notes in Computer Science Volume 4275*, pp. 1020–1038. Springer.

Billsus, D. et M. J. Pazzani (1999). A personal news agent that talks, learns and explains. pp. 268–275. ACM Press.

Cunningham, H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities 36*(2), 223–254.

d'Aquin, M., A. Schlicht, H. Stuckenschmidt, et M. Sabou (2009). Criteria and evaluation for ontology modularization techniques. In *Modular Ontologies*, Volume 5445, pp. 67–89. Springer Berlin Heidelberg.

Ehrig, M., P. Haase, M. Hefke, et N. Stojanovic (2004). Similarity for ontologies - a comprehensive framework. In *In Workshop Enterprise Modelling and Ontology : Ingredients for Interoperability, at PAKM 2004*.

Fellbaum, C. (1998). *WordNet : an electronic lexical database*. Cambridge, Mass : MIT Press.

Getahun, F., J. Tekli, R. Chbeir, M. Viviani, et K. Yetongnon (2009). Relating RSS News/Items. In *Web Engineering*, Number 5648 in Lecture Notes in Computer Science, pp. 442–452. Springer Berlin Heidelberg.

IJntema, W., F. Goossen, F. Frasincar, et F. Hogenboom (2010). Ontology-based news recommendation. pp. 1. ACM Press.

Lewis, D. D. et W. A. Gale (1994). A sequential algorithm for training text classifiers. In *SIGIR '94*, pp. 3–12. Springer London.

Liu, J., P. Dolan, et E. R. Pedersen (2010). Personalized news recommendation based on click behavior. pp. 31. ACM Press.

Middleton, S. E., N. R. Shadbolt, et D. C. De Roure (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems 22*(1), 54–88.

Rao, K. et V. Talwar (2008). Application domain and functional classification of recommender Systems—A survey. *DESIDOC Journal of Library and Information Technology 28*(3), 17–35.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, et J. Riedl (1994). GroupLens : an open architecture for collaborative filtering of netnews. pp. 175–186. ACM Press.

Richter, M. M. (1993). Classification and learning of similarity measures. In *Information and Classification*, Studies in Classification, Data Analysis and Knowledge Organization, pp. 323–334. Springer Berlin Heidelberg.

Salton, G. (1971). The SMART retrieval system - experiments in automatic document processing.

Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *SIGIR '94*, pp. 61–69. Springer London.

Summary

Today in the commercial and financial sectors, staying informed about economic news is crucial and involves targeting good articles to read, because the huge amount of information. To address this problem, we propose an innovative article recommendation system, based on the integration of a semantic description of articles and on a knowledge ontological model. We support our recommendation system on an intrinsically efficient vector model that we have perfected to overcome the confusion existing in models between the concepts of similarity and relevancy that does not take into account the effects of the difference in the accuracy of the semantic descriptions precision between profiles and articles, on the perceived relevancy to the user. We present in this paper a new evaluation of the relevancy adapted to vector model.