SpecificSearch : Un outil de recommandation automatique pour la veille d'information sur le web

Christophe Brouard[1], Christian Pomot[2]

[1]*Université Grenoble Alpes, LIG UMR 5217/équipe AMA, Grenoble, France*
Christophe.Brouard@imag.fr

[2]*Société Com&Net, Télespace Vercors, 38250 Villard-de-Lans, France*
cpomot@com-et-net.com

RÉSUMÉ. *Les systèmes de recommandation automatique par le contenu ont pour principale fonction de proposer à un utilisateur des informations susceptibles de l'intéresser sur la base des retours de pertinence qu'il a pu donner antérieurement sur d'autres informations. Différents algorithmes d'apprentissage automatique ont été intégrés à des systèmes de recherche d'information pour proposer des solutions permettant de réaliser cette tâche. Ces solutions n'ont cependant pas débouché sur des systèmes de recommandation automatique pour le web accessibles à tous. Il existe bien des agrégateurs de flux RSS permettant de recueillir de l'information sur le web mais les systèmes intégrant un apprentissage en sont encore à leurs balbutiements. Nous présentons ici les fonctionnalités et l'architecture d'une application web nommée SpecificSearch accessible en ligne qui se présente comme un agrégateur de flux RSS intégrant un apprentissage (http://www.specific-search.com). Une première évaluation permet de montrer la réalisabilité et l'utilité d'un tel système.*

ABSTRACT. *The main goal of a content-based recommender system is to propose to a user new documents which are likely to have some interest for him considering feedbacks he gave for other documents. Different machine learning algorithms have been integrated to information retrieval systems in order to cope with this task. However, these systems have not become as popular on the web as the well-known search engines. Otherwise, the RSS feed aggregators allow to gather information on the web but these systems do not integrate machine learning in order to improve the quality of the recommendations with the users' feedbacks. Here, we present the functionnalities and the architecture of a web application available online called SpecificSearch (http://www.specific-search.com) which is an RSS aggregator integrating a machine learning algorithm. A first evaluation shows how a such tool can be implemented and how it can be useful.*

MOTS-CLÉS : *recommandation automatique, veille d'information sur le web, filtrage adaptatif, flux RSS.*

KEYWORDS: *recommender system, web content monitoring, adaptive filtering, RSS feed aggregator*

Congrès INFORSID, 34ème édition, Grenoble, 31 mai – 3 juin 2016

1. Introduction

Les moteurs de recherche (Google, Bing, Yahoo, pour citer les plus utilisés[1]) constituent actuellement les outils incontournables pour l'accès à l'information sur le web. Ces outils de recherche permettent à l'utilisateur de saisir quelques mots clefs pour exprimer son besoin d'information et fournissent en retour une liste de pages web. Même si les moteurs de recherche intègrent une forme de personnalisation basée notamment sur les clics des utilisateurs sur les résultats proposés, cette adaptation est implicite et assez diffuse. Il n'est pas possible, pour l'utilisateur d'indiquer explicitement qu'un résultat correspond ou pas à un besoin d'information particulier. L'interaction entre l'utilisateur et ces outils est d'abord conçue comme un simple échange requête-réponse. Rien n'empêche l'utilisateur de reformuler sa requête en tenant compte de la première liste de résultats retournée par le moteur de recherche, mais ce travail de recherche et de composition de mots clefs lui revient. Si cette situation peut être admissible dans le cas d'une recherche ponctuelle, elle l'est beaucoup moins lorsque que le besoin d'information est récurrent comme dans le cas de la veille d'information où l'utilisateur souhaite être informé en permanence des nouveautés relatives à un sujet qui l'intéresse. En effet, dans ce cas, n'ayant pas la possibilité de capitaliser tout le travail réalisé lors des précédentes interactions, l'utilisateur répète les mêmes actions à chaque nouvelle interaction et le besoin de nouveaux outils adaptés et permettant une veille d'information efficace se fait alors ressentir. Les technologies et algorithmes permettant le développement de tels outils existent. Cependant, le développement d'outils dédiés à la veille d'information, en mesure d'apprendre les préférences de l'utilisateur et accessibles à tous comme le sont les moteurs de recherche en est encore à ses balbutiements et les fonctionnalités, l'architecture et le mode d'interaction avec l'utilisateur de tels outils restent à inventer.

L'objet de cet article est la présentation de SpecificSearch, un outil de veille d'information sur le web qui permet la capitalisation du travail de recherche et en particulier celui d'expression du besoin d'information. Il repose sur une interface homme-machine facilitant l'interaction et sur un apprentissage automatique permettant de garder une trace utile de celle-ci. La suite de l'article a la structure suivante : dans un premier temps, nous mettons en relief le fossé qui existe entre les besoins liés à une tâche de veille d'information et les outils classiques de recherche que l'on peut trouver sur le web. Dans un deuxième temps, nous décrivons des systèmes existants (agrégateurs de flux RSS et systèmes de filtrage adaptatif) qui apportent des solutions partielles par rapport aux besoins énoncés. Nous présentons ensuite SpecificSearch un outil empruntant aux différents systèmes décrits précédemment pour proposer une réponse la plus complète possible. Nous terminons en décrivant une première évaluation, la mise en place de l'outil et ses perspectives d'évolution.

---

[1] selon le site https://www.netmarketshare.com/search-engine-market-share.aspx?qprid=4&qpcustomd=0

2. *La nécessité d'outils novateurs pour la veille d'information le web*

2.1. *Introduction à la notion de veille d'information*

La veille d'information englobe différents types de veille (la veille technologique, économique, réglementaire, etc...). Sans entrer dans le détail de cette typologie, Serge Cacaly (2008) définit la veille d'information ou veille informationnelle comme un « processus continu et dynamique faisant l'objet d'une mise à disposition personnalisée et périodique de données ou d'informations, traitées selon une finalité propre au destinataire, faisant appel à une expertise en rapport avec le sujet ou la nature de l'information collectée ». Cette définition fait notamment ressortir deux aspects importants de l'activité sur lesquels nous reviendrons par la suite. D'abord, le processus est continu, il ne correspond pas à un besoin ponctuel mais s'étend sur une certaine durée. Ensuite, il est lié à l'expertise de la personne réalisant la veille et suppose l'existence d'un savoir important permettant de distinguer dans un important volume d'informations celles qui sont pertinentes par rapport à un objectif ou une problématique donnée.

D'un point de vue plus analytique, on a coutume de distinguer plusieurs étapes dans une activité de veille. Selon, Elisabeth Noël (2008), on peut distinguer quatre étapes : le ciblage, la recherche, l'analyse et la diffusion. Le ciblage correspond à la définition précise du sujet d'intérêt et des objectifs de la veille ainsi qu'à la sélection des sources d'information. La recherche correspond au recueil d'informations pertinentes par rapport au sujet défini. L'analyse correspond à l'exploitation des informations recueillies afin d'en extraire du sens (cette étape fera intervenir des connaissances autres que celles présentes dans les informations recueillies). La diffusion correspond essentiellement à la préparation de l'information sous une forme intelligible et à sa transmission aux personnes susceptibles d'être intéressées.

2.2. *Ce qui manque aux traditionnels moteurs de recherche*

Les moteurs de recherche sont conçus pour des besoins d'information ponctuels. Ils permettent bien de trouver des sources d'informations et des informations pertinentes mais bien qu'il soit aussi possible de sauvegarder les sources d'informations (URLs de pages web) ou les informations (contenus des pages web) au moyen des favoris du navigateur, ces fonctions de sauvegardes restent rudimentaires. On notera que la sauvegarde de l'URL comme favori est une sauvegarde de la source d'information et non de l'information elle-même car si le contenu de la page change, l'information est perdue. L'information ne peut être réellement sauvegardée qu'en enregistrant tout simplement la page au moyen du navigateur ou d'un outil d'aspiration de site. Il y a aussi un problème de granularité car la sauvegarde de la totalité de la page web n'est pas adaptée si l'information pertinente se situe au milieu d'une page contenant beaucoup d'informations. De plus, les résultats non pertinents lus par l'utilisateur ne sont pas non plus marqués

comme ayant déjà été lus et l'utilisateur relira peut-être la page web avant de se souvenir qu'il l'avait déjà lue et conclu à son inadéquation par rapport à sa recherche. De même, tout le travail de formulation de la requête consistant à trouver les bons mots clés par reformulations successives en fonctions des résultats retournés par le moteur de recherche n'est pas non plus capitalisé. Enfin, les moteurs de recherche n'offrent pas d'outil d'annotation pour commenter les résultats et transmettre l'information et son commentaire à d'autres utilisateurs.

Par ailleurs, les moteurs de recherche sont des outils dits « PULL ». Ils nécessitent une action de l'utilisateur pour aller chercher l'information. Cela signifie que l'utilisateur devra régulièrement se connecter aux différents moteurs ou outils de recherche sur les sites pertinents par rapport à son sujet de veille pour aller chercher l'information. Cette vérification de l'utilisateur peut se révéler très fastidieuse et inutile si aucune information intéressante n'est apparue sur le site qu'il surveille. Les outils dits « PUSH » sont mieux adaptés à une activité de veille, dans ce cas, l'utilisateur utilise un seul outil qui récupère automatiquement les informations à partir des différentes sources et les lui présente.

3. *Les outils et technologies existantes*

3.1. *Flux RSS et agrégateurs de flux*

Les flux (ou fils) RSS sont de simples fichiers XML respectant une structure XML prédéfinie[2]. Au même titre que des pages HTML, ce fichier est servi par le serveur web associé au site et récupéré au moyen d'une requête HTTP. Ce fichier peut être récupéré et affiché par n'importe quel navigateur (cf Fig.1). Il est composé d'une suite d'items (par exemple les différentes nouvelles liées à l'actualité) chaque item étant lui-même composé d'un titre, d'une description et d'un éventuel hyperlien renvoyant sur une page web entrant dans le détail de la nouvelle. Il est souvent mis à jour pour contenir uniquement les informations les plus récentes (de nouvelles informations sont ajoutées, d'anciennes sont supprimées).

De nombreux sites proposent des résumés de l'actualité de leur contenu sous forme de flux RSS. Si le flux RSS pour une page web que l'on souhaite surveiller n'existe pas on peut le créer à partir d'outils existants[3]. Il suffit d'indiquer la façon dont il faut analyser le fichier HTML correspondant à la page à surveiller pour que l'outil génère un fichier RSS et une URL pour le récupérer. Cela fonctionne à condition que le fichier HTML dispose d'une structure et que cette structure ne change pas en permanence. D'autres outils dédiés de même nature mais pour des types de sites particuliers ayant tous la même structure existent aussi. Il est par exemple possible de transformer les messages d'un mur facebook ou les tweets d'un

[2] la spécification du format est ici : http://www.rssboard.org/rss-specification
[3] comme http://feed43.com/, http://www.page2rss.com/, ...

compte twitter en items d'un flux RSS[4]. A l'origine Facebook et Twitter donnaient accès à un flux RSS associé à chaque compte. Mais il existe une tendance pour certains sites, pour maximiser leur trafic, à forcer l'utilisateur à consulter les informations qu'il délivre directement sur le site en ne proposant plus la création automatique de flux RSS qui pourraient être consultés à partir d'outils externes. Cette tendance est corroborée par certaines expérimentations montrant que dans certains cas, il est préférable pour un site de se passer de flux RSS pour maximiser son trafic (Dan, 2012). Il restera néanmoins toujours possible de passer par des outils externes. Potentiellement, beaucoup de contenu web peut donc être transformé en flux RSS. L'un des intérêts des flux RSS est de pouvoir découper la page web en unités de contenu et donc d'avoir un niveau de granularité d'information plus faible et même réglable si on passe par un outil générant le flux à partir de la page.

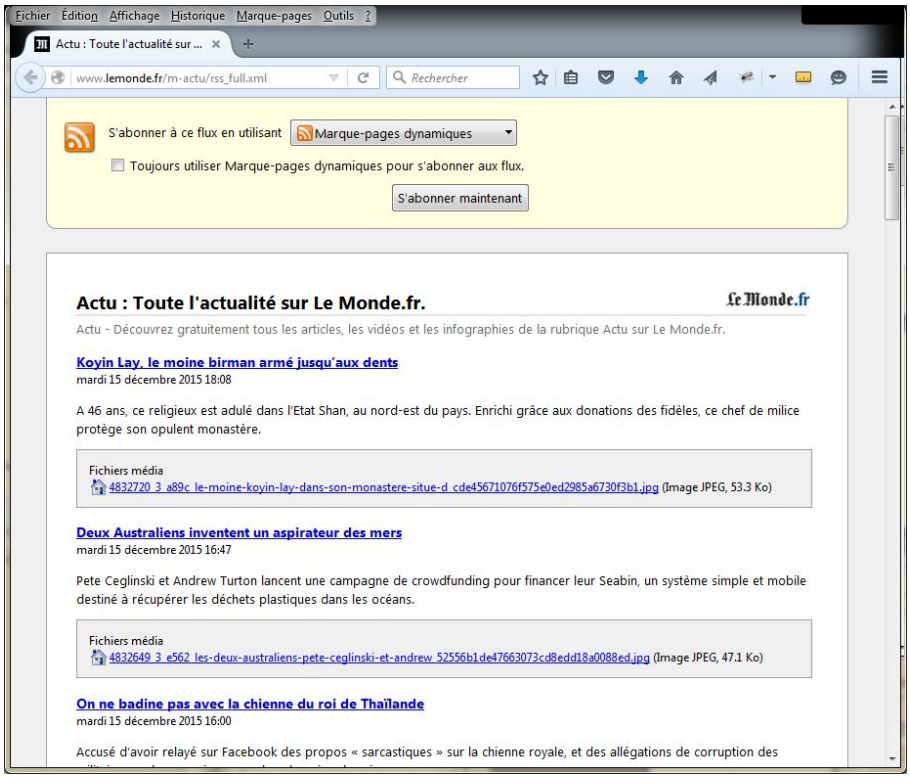

Figure 1. *Affichage du flux d'URL : http://www.lemonde.fr/m-actu/rss_full.xml dans un navigateur*

[4] http://twitrss.me/ et http://www.wallflux.com/fr/

Bien que ces fichiers RSS puissent s'afficher dans des navigateurs, des outils dont l'interface conviviale permet l'abonnement, l'affichage et le rafraîchissement automatique de plusieurs flux existent. Ces outils s'appellent des agrégateurs de flux. Il en existe de nombreux. Il peut s'agir d'agrégateurs en ligne (une application web sur laquelle l'utilisateur se crée un compte) comme netvibes[5] ou feedly[6] ou bien d'applications s'exécutant en local sur le poste de l'utilisateur comme par exemple rssowl[7] ou thunderbird (l'outil de messagerie).

En plus de l'intérêt des flux RSS qu'ils manipulent, les agrégateurs de flux offrent, comparativement aux moteurs de recherche, différents avantages pour la veille d'information. D'abord, il s'agit d'outils « PUSH ». L'utilisateur n'a plus besoin d'aller chercher l'information sur les différents sites ou moteurs de recherche, les différents fichiers XML sont récupérés automatiquement. Par ailleurs, certains agrégateurs permettent aussi de stocker les différents résultats recueillis.

Même si certains agrégateurs permettent le filtrage par mots clés, cette fonctionnalité reste assez rudimentaire. Il est en effet difficile d'identifier tous les bons mots clés. Il est aussi impossible de pondérer ces mots et le filtrage est binaire : soit l'information contient les mots clés et elle est sélectionnée soit elle ne les contient pas et elle est ignorée. Or cette fonction de filtrage est souvent utile car même s'il existe des flux dont le sujet est souvent très focalisé et que ces flux contiennent des informations qui peuvent potentiellement toutes intéresser l'utilisateur, il existe aussi des cas où la recherche de l'utilisateur est tellement précise qu'aucun flux ne correspond totalement à la recherche ou des cas où certains flux ne délivrent qu'occasionnellement de l'information pertinente. De façon plus globale, comme nous l'avons évoqué précédemment un sujet ou un objectif de veille d'information est souvent complexe et liée aux connaissances de l'expert et ne peut se résumer à un filtrage binaire sur de simples mots clés.

3.2. *Les systèmes de filtrage adaptatif*

De nombreux systèmes de filtrage (ou recommandation) automatique (Bodilla et al, 2013) basés sur les retours de pertinence des utilisateurs ont été conçus. Parmi ces systèmes, on peut distinguer, les systèmes de filtrage collaboratif et les systèmes de filtrage basés sur le contenu. Dans le cas du filtrage collaboratif, les retours de pertinence d'un utilisateur pour une information sont utilisés pour évaluer la pertinence de cette information pour les autres utilisateurs. Nous nous focalisons ici sur le cas du filtrage basé sur le contenu où l'on considère un utilisateur isolé. Le principe de ces systèmes est le suivant : à partir d'un ensemble d'exemples

[5] http://www.netvibes.com/fr
[6] https://feedly.com
[7] http://www.rssowl.org/

d'informations pertinentes et non pertinentes pour un utilisateur et un sujet de veille donné, la tâche consiste à évaluer la pertinence d'un nouveau document. Ces systèmes s'appuient sur des systèmes de classification automatique (Joachims, 1998) ou sur des systèmes de recherche d'information (Robertson et Walker, 2001), (Wu et al, 2001) qui intègrent une méthode de mise à jour du poids des termes comme la méthode Rocchio (Rocchio, 1971). Ils sont en mesure d'améliorer la qualité de la sélection au fur et à mesure des retours de pertinence des utilisateurs. Les conférences TREC 2000, 2001 et 2002 (Robertson et Hull, 2000), (Robertson et Soboroff, 2001&2002) ou TREC KBA[8] ont été l'occasion de comparer les performances de différents systèmes de filtrage sur différents corpus de textes et de se confronter à différents problèmes liés à cette tâche notamment à celui de la définition d'un seuil de sélection (Arampatzis, 2001) et à celui de « démarrage à froid » (Blerina, 2014) car en effet, la tâche est particulièrement difficile lorsque peu d'exemples de documents pertinents sont donnés au départ.

Malgré ces difficultés, les diverses expérimentations ont montré que ces systèmes sont en mesure d'apprendre à sélectionner les informations pertinentes de façon beaucoup plus fiable qu'un simple filtrage binaire basé sur des mots clés. Dans ces systèmes, de très nombreux mots clés ajoutés automatiquement avec différentes pondérations calculées automatiquement sont pris en compte et les pondérations sont combinées de diverses façons pour calculer un score global. Ces bons résultats ont conduit à quelques initiatives récentes pour une application de ce type de système à la veille d'information sur le web (Katakis et al., 2009), (Nanas et al., 2010), (Paliouras et al., 2008). Actuellement, à notre connaissance, deux applications web pour la veille d'information intègrent un apprentissage automatique : il s'agit de Prismatic[9] et Noowit[10]. Néanmoins, avec ces outils, soit le choix des sources d'information reste très limité, soit les temps de réponse et l'ergonomie rendent leurs utilisations difficiles et semblent indiquer que ces outils ne sont encore pas totalement aboutis.

4. Présentation de SpecificSearch

Considérant l'existence de systèmes capables d'apprendre à filtrer l'information sur la base de retours de pertinence et aussi celle d'un format de description de l'information sur le web (RSS) et considérant aussi l'essor de technologies permettant le développement d'interfaces web suffisament sophistiquées (comme les librairies JavaScript), il semble que les ingrédients nécessaires au développement de moteurs de recherche accessibles à tous et dédiés à la veille d'information pour le web soient maintenant présents. En nous appuyant sur le système de filtrage Echo

[8] http://trec-kba.org/
[9] http://getprismatic.com/
[10] http://www.noowit.com/

(Brouard, 2012), nous avons conçu et développé une application web permettant la veille d'information sur le web en s'appuyant sur les flux RSS. Nous décrivons dans cette partie les fonctionnalités, l'interface et l'architecture de cette application.

4.1. *Les différentes étapes dans l'utilisation de l'outil*

4.1.1. *Création d'une recherche*

Contrairement à la plupart des agrégateurs, SpecificSearch est centré « recherche » et non « flux ». L'utilisateur ne commence donc pas par s'abonner à des flux mais par créer une recherche, c'est-à-dire par définir un sujet d'intérêt (étape « ciblage » de la veille).

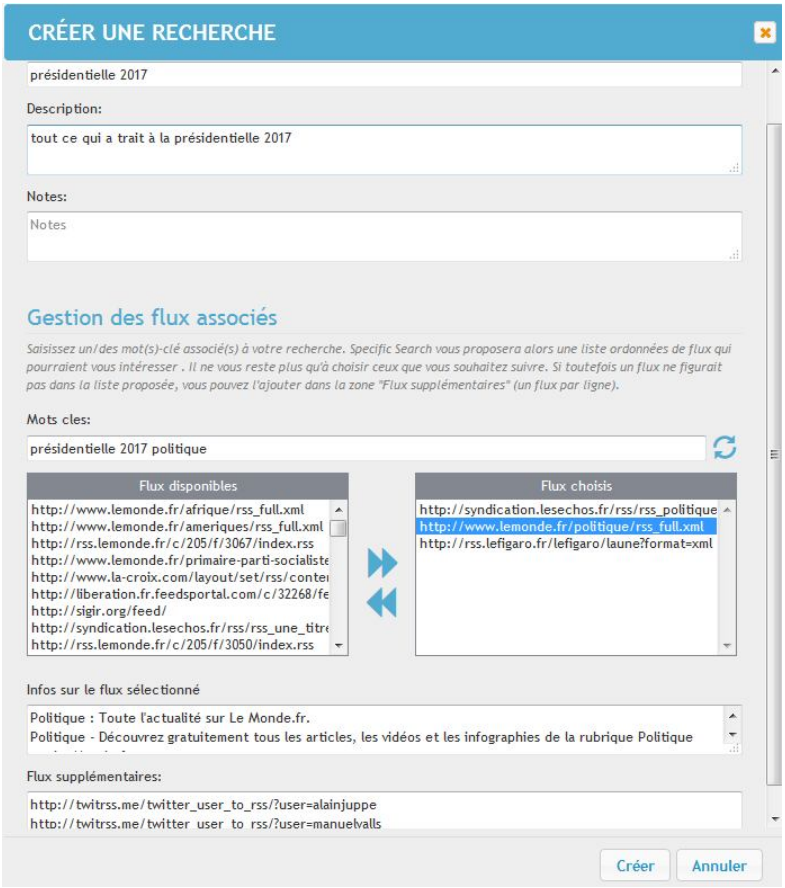

Figure 2. *Boîte de dialogue permettant la création d'une recherche (un sujet de veille), avec recherche de flux par mots clés. Il est aussi possible d'ajouter directement les URL s des flux dans le champ « flux supplémentaires ».*

Supposons que le sujet de la veille concerne tout ce qui a trait à la l'éléction présidentielle française de 2017. L'utilisateur commence par ouvrir la fenêtre de dialogue de création de recherche. Une fois cette fenêtre ouverte (cf Fig.2), il commence par nommer sa recherche puis ensuite sélectionne des sources d'informations à surveiller. A ce sujet, l'une des explications au succès très relatif des flux RSS semble être la difficulté pour un large public à sélectionner une source. En effet, il faut d'abord trouver un flux correspondant à la recherche (les annuaires de flux ne sont pas faciles à trouver, peu nombreux et très incomplets bien qu'en voie d'amélioration notamment avec feedly ou Instant RSS Search[11] basé sur une api Google). Puis il faut aussi souvent copier-coller une URL dans son outil. Afin d'éviter cette recherche et ce copier-coller d'URL pas toujours maîtrisé, SpecificSearch permet la recherche de flux par mots clés. L'utilisateur va par exemple taper les mots clés « présidentielle 2017 politique » dans le champ mots clés. De nombreux flux lui sont alors proposés. Il choisit parmi ces flux ceux qui lui semblent les plus pertinents et/ou ajoute manuellement des URL s de flux.

4.1.2. *Saisie des retours de pertinence et stockage des informations*

Une fois la recherche validée, les informations provenant des différents flux choisis sont affichées dans un onglet « Nouveaux résultats » (cf Fig.3).

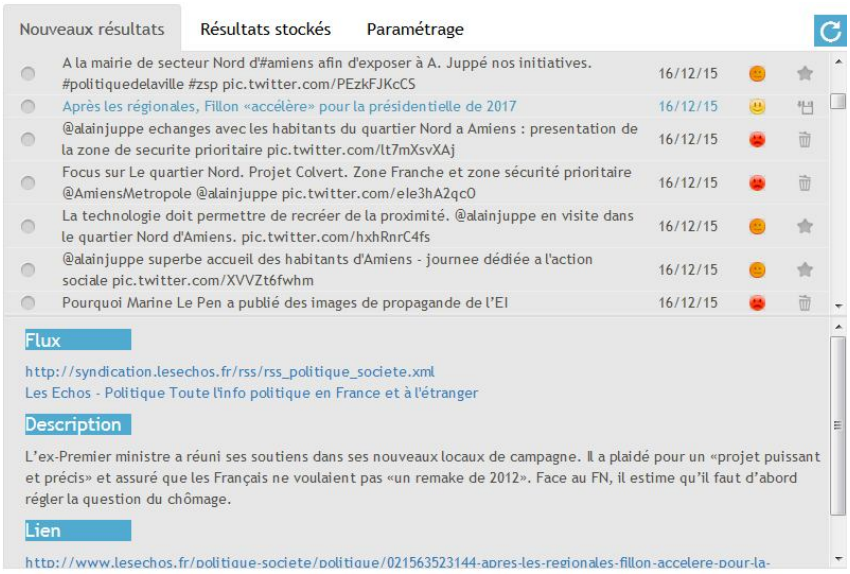

Figure 3. *Les nouveaux résultats sont affichés. L'utilisateur indique au moyen d'une émoticône si une nouvelle est pertinente ou pas. Indépendemment de son retour de pertinence, il peut aussi choisir de sauvegarder ou de supprimer une nouvelle.*

[11] http://ctrlq.org/rss/

L'utilisateur va alors commencer à apprendre au système à sélectionner les informations pertinentes en indiquant pour certaines nouvelles, au moyen d'une émoticône si elles sont pertinentes ou pas. Il peut aussi choisir de sauvegarder certaines nouvelles (elles seront alors déplacées dans l'onglet « Résultats stockés ») ou de les supprimer (elles seront alors déplacées dans l'onglet « Résultats supprimés » qui n'apparaît pas par défaut).

4.1.3. *Recalcul des scores de pertinence et visualisation des résultats*

Une fois les retours de pertinence donnés, l'utilisateur peut demander le calcul des scores de pertinence pour toutes les nouvelles non jugées. Une fois le calcul effectué par Echo (cf section 4.2.2), les nouvelles apparaissent alors classées par ordre de pertinence décroissante (cf Fig. 4).

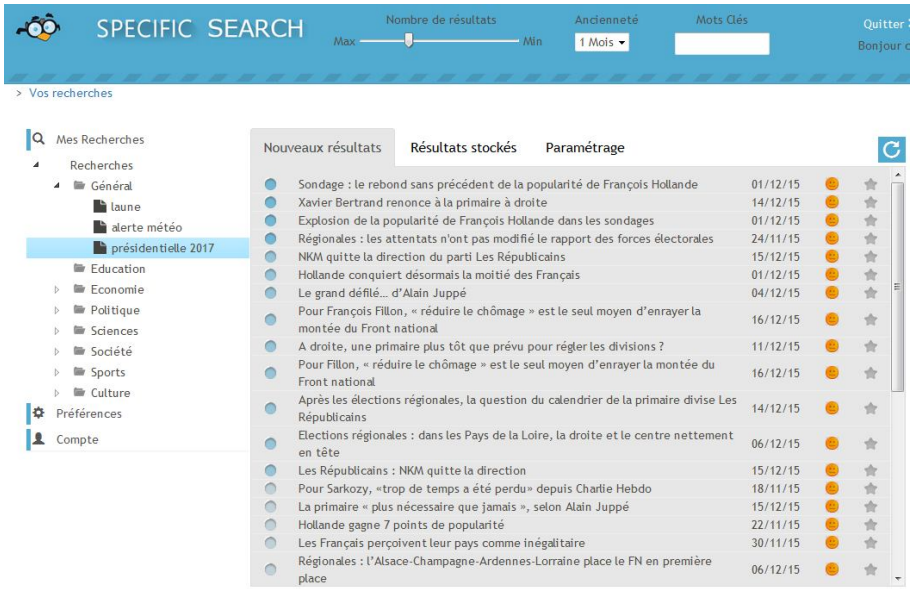

Figure 4. *Une fois le calcul des scores et des probabilités de pertinence effectués, les nouvelles sont affichées dans l'ordre des scores de pertinence décroissant. L'intensité de la coloration de la puce qui précède le titre de la nouvelle indique sa probabilité de pertinence. Il est aussi possible de filtrer les nouvelles sur cette probabilité, sur leur ancienneté ou encore sur de simples mots clés. Globalement, on trouve à gauche de l'interface, l'arborescence des recherches (sujets de veille) et au centre les informations propres à la recherche sélectionnée.*

Une probabilité de pertinence est indiquée au moyen de l'intensité de la coloration de la puce qui précède le titre de la nouvelle. L'utilisateur a la possibilité de n'afficher que les nouvelles dont la probabilité dépasse un certain seuil. Il peut aussi choisir l'ancienneté des nouvelles affichées et filtrer sur des mots clés. Sur la

base de ce nouvel affichage, l'utilisateur peut donner de nouveaux retours de pertinence et demander par un simple clic le recalcul des scores et un nouvel affichage et itérer ce processus autant de fois qu'il le souhaite.

4.2. *L'architecture de l'outil*

4.2.1. *Architecture générale*

SpecificSearch est une application web orientée client. Elle s'appuie sur un framework MVC PHP et sur la librairie JavaScript JQuery. Une représentation schématique de son architecture est donnée dans la figure ci-dessous (cf Fig.5).

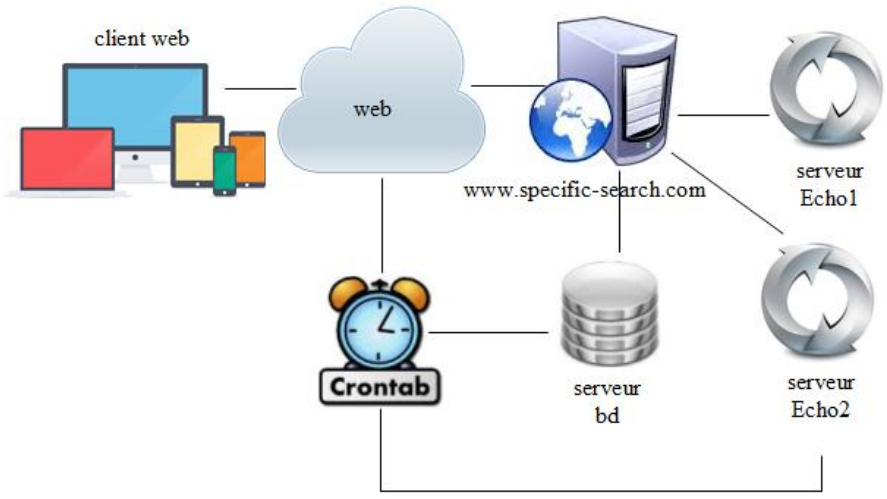

Figure 5. *Architecture de l'application web SpecificSearch. Le navigateur, le serveur web et le serveur de base de données sont les ingrédients classiques d'une application web. Par ailleurs, deux serveurs Echo dialoguent avec le serveur web. Echo1 calcule les scores de pertinence des nouvelles étant donnés les retours de pertinence de l'utilisateur. Echo2 calcule les scores de pertinence de flux pour des mots clés. Des scripts exécutés régulièrement et automatiquement récupèrent les nouvelles des différents flux sauvegardés dans la base de données et mettent à jour l'index des flux de Echo2.*

Les traitements qui ne nécessitent pas de données du serveur sont exécutés par le navigateur sur le poste client. Les traitements qui nécessitent l'intervention du serveur sont appelés au moyen de requêtes AJAX. Certains traitements comme la récupération des nouvelles sont effectués hors ligne auprès des différents serveurs web servant les différents flux RSS. Pour les calculs de pertinence, le serveur web fait appel au système Echo qui a été réécrit comme un serveur. Des scores de

pertinence sont calculées pour les nouvelles sur la base des retours de pertinence donnés mais aussi pour les flux dans le contexte de la recherche de flux par mots clés. Dans les deux cas, il s'agit bien du système Echo qui est à l'œuvre, mais deux serveurs Echo différents tenant compte d'un besoin d'interaction différent ont dû être mis en place. Ces deux types d'interaction qui constituent l'originalité de cette architecture sont détaillés dans la suite.

4.2.2. *Le système Echo*

Echo (Brouard, 2012) est un système de sélection d'information pertinente. Il peut être appliqué à différents types de problèmes, comme la recherche d'information (sélection d'un document à partir d'une requête), la classification supervisée (sélection d'une classe à partir d'un document) ou encore l'extension de requête (sélection d'un terme à partir d'un ensemble de termes). Il est basé sur une formalisation de la notion de pertinence qui combine les notions de spécificité et d'exhaustivité qui sont au cœur des modèles de pertinence en recherche d'information (Brouard, 2004). Echo peut être décrit comme un système de construction et d'exploitation de réseau associatif s'appuyant sur des mécanismes neuronaux simples. La construction du réseau s'appuie sur la règle de Hebb liant deux informations survenant simultanément (deux termes cooccurant dans le même document par exemple), son exploitation s'appuie sur une méthode de propagation et une mesure d'écho c'est-à-dire la mesure d'une quantité d'activation rétro-propagée vers les sources d'activation. Le système a été appliqué avec succès à différents problèmes dont celui de la classification de textes qui nous intéresse plus particulièrement dans le cadre de cette application. Il obtient sur cette tâche des performances légèrement inférieure aux SVM mais supérieures à la méthode des k plus proches voisins (meilleure méthode hors SVM) sur le corpus de référence Reuters-21578 (Brouard, 2012). Echo peut par ailleurs être appliqué à de grandes quantités de données. Ainsi Echo a été appliqué lors du deuxième challenge international "Large Scale Hierarchical Text Classification"[12] et dans le cadre de la compétition Kaggle[13] à la classification automatique de plusieurs millions de documents dans plusieurs centaines de milliers de classes. Les résultats obtenus par Echo dans le cadre de ces compétitions (1^{er}/17 et $9^{ème}$/115) ont montré son efficacité. Echo a par ailleurs l'avantage d'être totalement incrémental (il est inutile de reconstruire tout le réseau lorsque des exemples sont ajoutés ou supprimés). Enfin, les connexions dans le réseau pouvant s'interpréter comme des règles « SI ... ALORS », ses choix de classification sont facilement explicables à l'utilisateur.

[12] http://lshtc.iit.demokritos.gr/
[13] https://www.kaggle.com/c/lshtc

4.2.3. *L'API du système de calcul des scores des informations*

Une première forme du serveur Echo (Echo1) est utilisée pour calculer les scores de pertinence lors d'une demande de mise à jour de l'utilisateur. Dans cette forme, un seul type de requête est autorisé : le serveur prend en entrée une chaîne de caractères décrivant toutes les nouvelles. Cette chaine contient, pour chaque nouvelle, l'identifiant de la nouvelle, les mots clés qu'elle contient et le jugement de pertinence associé (pertinent, non pertinent ou pas de jugement). Le serveur retourne une chaîne de caractères correspondant aux scores et probabilités de pertinence de toutes les nouvelles auxquelles aucun jugement de pertinence n'a été associé. La probabilité de pertinence est calculée sur la base des distributions des scores calculés par Echo des documents pertinents et non pertinents (Brouard, 2012). Dès lors qu'un nouveau jugement de pertinence est donné tous les scores doivent être recalculés. Les volumes de données échangés dans ce cas restent relativement restreints (quelques centaines, voire quelques milliers de nouvelles tout au plus) car basés sur les retours de pertinence de l'utilisateur qui sont donnés manuellement.

4.2.4. *L'API du système de sélection des sources d'information*

Une autre forme du serveur Echo (Echo2) a dû être mise en place pour répondre au problème de la recherche de flux par mots clés. Contrairement à Echo1 où tout est reconstruit à chaque fois, Echo2 garde un index en mémoire associant mots clés et identifiants de flux. Deux types de requêtes sont possibles : les requêtes relatives à l'ajout de nouvelles dans l'index et des requêtes relatives au calcul de scores de pertinence de flux pour des mots clés particuliers. L'index est construit sur la base de toutes les nouvelles contenues dans la base de données (actuellement plusieurs centaines de milliers et potentiellement plusieurs millions) et est mis à jour régulièrement et incrémentalement avec les nouvelles informations recueillies. Les volumes de données échangés lors de la construction de l'index sont donc très importants. Ils sont par contre très réduits lors des requêtes de calcul de scores (quelques mots clés en entrée et une centaine d'identifiants de flux en retour).

5. Première évaluation

Bien que les performances du système Echo aient déjà été testées largement par ailleurs, nous avons réalisé une première évaluation de l'outil afin de vérifier que le système s'appliquait bien à des flux d'actualités (textes de taille réduite), répondait suffisament vite quand on l'intègrait à une application web (car l'apprentissage est réalisé en ligne à chaque demande de mise à jour de l'utilisateur) et ne nécessitait pas trop de retours de pertinences pour fournir des résultats intéressants en compraison d'un simple système de recherche par mots clefs. En particulier nous avons étudié l'évolution de la qualité des réponses avec les retours de pertinence et nous les avons comparés avec un simple filtrage sur mot clef. Cinq sujets de veille

ont été définis (portant respectivement sur la crise des migrants, les suites des attentats du 13 novembre, le football, la présidentielle de 2017 et le monde de l'éducation). Pendant un mois, les nouvelles de la une du journal « Le Monde », ce qui représente environ mille nouvelles, ont été étiquetées comme pertinentes ou non vis-à-vis de ces différents sujets. Puis des retours positifs et négatifs sur les premiers documents retournés par le système ont été simulés. Différentes quantités de retours, respectivement 5/10 (5 positifs, 10 négatifs), 10/20 et 15/30 ont été testées pour les 15 premiers jours. En fonction de ces retours, la précision à respectivement 5, 10, 15 documents retournés et la R-précision (c'est-à-dire la précision lorsque le nombre de documents retournés est égal au nombre de documents pertinents) a été calculée sur les 15 derniers jours. De façon à montrer que la tâche n'était pas triviale, les résultats du système ont été comparés avec un simple filtrage par mot clef (en choisissant celui donnant les meilleurs résultats, respectivement « migrant », « déchéance », « foot », « primaire » et « université »).

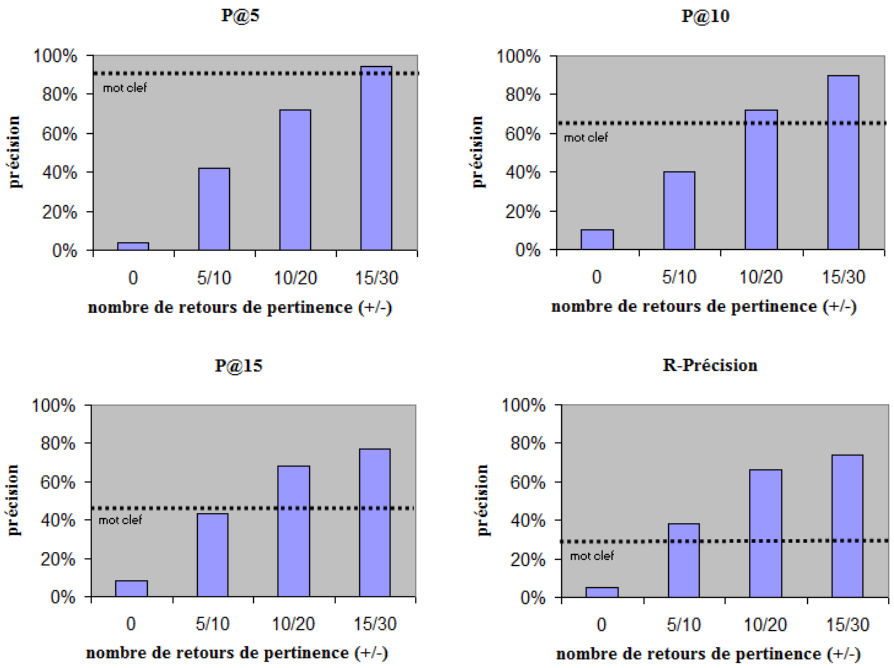

Figure 6. *Dans le cas des précisions à 15 et de la R-précision, dès lors qu'au moins 10 retours de pertinence positifs et 20 retours négatifs sont donnés par l'utilisateur, les résultats sont très nettement en faveur du système basé sur l'apprentissage automatique en comparaison d'un simple système de filtrage par mots clefs. Dans le cas des précisions à 5 et à 10, l'utilisateur doit donner plus de retours de pertinence pour obtenir de meilleur résultats.*

Il apparaît que la précision augmente bien avec le nombre de retours de pertinence donnés et devient vite supérieure à un filtrage par mot clef (cf Fig. 6). Plus précisément, il apparaît que le filtrage par mot clef ne permet pas de retrouver de nombreuses nouvelles pourtant pertinentes mais qui ne contiennent pas le mot clef (une nouvelle décrivant le résultat d'une équipe de football connue pourra par exemple ne pas contenir le mot « foot ») et que des nouvelles qui contiennent le mot clef peuvent ne pas être pertinentes (une nouvelle contenant le mot « primaire » peut concerner les primaires américaines et non la politique française). L'apprentissage automatique permet de retenir un plus grand ensemble de mots, de pondérer l'importance respective de chacun de ces mots et de combiner ces pondérations de façon optimale. Par ailleurs, les temps liés au calcul des scores et à la réponse de Echo1 se sont révélés inférieurs à la seconde et ceci même en considérant plusieurs centaines de documents exemples et tests. Ces résultats permettent de conclure que l'apprentissage en ligne qui incluait la préparation des données à envoyer au serveur Echo1, leur traitement par Echo1, la récupération des résultats et leur rangement dans la base de données est donc réalisable. Pour Echo2, la construction du réseau qui prend moins d'une minute pour plusieurs centaines de milliers de nouvelles est réalisée hors ligne.

6. Conclusion

Une interface homme-machine intégrant des fonctionnalités permettant de faciliter l'activité de veille sur le web a été proposée. Une architecture client-serveur intégrant un système d'apprentissage automatique a été conçue. Bien que la première évaluation soit relativement limitée et bien que la réalisation d'expérimentations en conditions réelles avec différents utilisateurs véritablement engagés dans un processus de veille sur une durée dépassant le mois semble incontournable, on peut considérer que le résultat obtenu montre déjà la faisabilité d'un tel outil (temps de réponse et qualité des résultats).

L'outil a été mis en ligne (http://www.specific-search.com) à la fin de l'année 2015. N'importe quel internaute peut maintenant se créer un compte et utiliser l'outil, laisser des commentaires, des souhaits d'évolution ou signaler des bugs. SpecificSearch est actuellement en bêta-test et nous comptons sur des retours nombreux et pertinents pour l'améliorer. Il est plus que probable que l'outil évoluera encore. Certaines évolutions concerneront notamment la mise en place d'outils d'annotation relatifs à l'étape de diffusion de la veille d'information (l'outil se limitant pour l'instant aux étapes de ciblage et de recherche), d'autres concerneront l'explication de la sélection d'une nouvelle ou encore l'auto-évaluation du système dans sa capacité à sélectionner l'information pertinente. Par ailleurs, nous sommes conscients que si le succès est au rendez-vous et que les données dans l'index deviennent plus volumineuses, l'infrastructure actuelle risque de ne pas être adaptée. Une version distribuée d'Echo est à l'étude.

7. Bibliographie

Arampatzis A., Beney J., Koster C.H.A, van der Weide T.P., Incrementality, Half-life, and Threshold Optimization for Adaptive Document Filtering. *Proceedings of the TextRetrieval Conference (TREC9),* NIST Special Publication, p.589-600, 2001.

Blerina L., Kostas K., Stathes H.,« Facing the cold start problem in recommender systems», *Expert Systems with Applications*, 41, p. 2063-2073, 2014.

Bobadilla J., Ortega F., Hernando A., Gutierrez R. « Recommender systems survey», *Knowledge-Based Systems*, 46, p. 109-132, 2013.

Brouard C., « Document Classification by Computing an Echo in a Very Simple Neural Network», *ICTAI*, 735-741, 2012.

Brouard C., Nie J.Y., « Relevance as Resonance: a New Theoretical Perspective and a Practical Utilization in Information Filtering », *Information Processing and Management*, 40, p. 1-19, 2004.

Cacaly S., Le Coadic Y-F, Pomart P-D., Sutter E., Dictionnaire de l'information (3e édition), Paris, A. Colin, 95 p., 2008.

Dan M., « Use of RSS feeds to push online content to users», *Decision Support Systems*, 54, p. 740-749, 2012.

Joachims T. « Text categorization with support vector machines:Learning with many relevant features », In Proceeding of ECML-98, 1998.

Katakis I., Tsoumakas G., Banos E., Bassiliades N., Vlahavas I.« An adaptive personalized news dissemination system», *Journal of Intel. Inform. Systems*, 32(2), p. 191-212, 2009.

Nanas N., Manolis V., Elias H.,« Personalised news and scientific literature aggregation», *Inform. Process. and Management*, 46, p. 268-283, 2010.

Noël E., « Veille et nouveaux outils d'information ». In DINET Jérôme, Usages, usagers et compétences informationnelles au XXIème siècle. Paris, Hermès; p. 257-284, 2008.

Paliouras G., Mouzakidis A., Moustakas V., Skourlas C., « PNS: A personalized news aggregator on the web », *Intelligent Interactive Systems in Knowledge-based Environments*, 104, p. 175-197, 2008.

Robertson S., Hull D. « The TREC-9 filtering track final report », *Proceedings of the TextRetrieval Conference (TREC9),* NIST Special Publication, 2001.

Robertson S., Walker S. « Microsoft Cambridge at TREC-9: Filtering Track», *Proceedings of the TextRetrieval Conference (TREC9),* p. 361-368, 2001.

Robertson S., Soboroff I. « The TREC-2002 filtering track report », *Proceedings of the TextRetrieval Conference (TREC11),* NIST Special Publication, 2003.

Rocchio J.J., « Relevance Feedback in Information Retrieval », *In The Smart Retrieval System, G. Salton (Ed.), Prentice* Hall, p. 313-323, 1971.

Wu L., Luang X., Guo Y., Zhang Y. « FDU at TREC-9:CLIR, Filtering and QA Tasks », In *Proceedings of the TextRetrieval Conference (TREC9),* p. 189-202, 2001.