Retweeter ou ne pas retweeter :

Le dilemme des portails de diffusion d'information temps-réel

Thomas Palmer, Gilles Hubert, Karen Pinel-Sauvagnat

IRIT UMR 5505 CNRS, Université Paul Sabatier – Toulouse 3
{Thomas.Palmer, Gilles.Hubert, Karen.Sauvagnat}@irit.fr

RÉSUMÉ. *L'étude des caractéristiques contextuelles a été largement traitée en Recherche d'Information (RI), mais les applications concrètes sur de vrais flux de données ne sont pas très répandues. Dans cet article, notre problématique concerne la décision automatique de retweeter un message. En considérant le centre d'intérêt d'un utilisateur, nous proposons un modèle pour effectuer un filtrage automatique en temps-réel du flux Twitter en utilisant de multiples caractéristiques contextuelles. Le modèle sépare l'aspect contextuel du contenu du message en lui-même, tout en conservant une très grande vitesse d'exécution. Des expérimentations ont été réalisées sur la collection TREC Microblog 2015. Les résultats montrent que l'intégration de caractéristiques de contexte a un impact positif sur l'efficacité du filtrage sans pénaliser son efficience.*

ABSTRACT. *The study of contextual features has been widely discussed in Information Retrieval (IR), but concrete applications on real data streams are not common. In this paper, we aim at doing retweet recommandation. Considering a user interest, we introduce a model to perform real-time online filtering of the Twitter stream using several contextual features. The model separates content and contextual aspects, achieving a very high velocity. Experiments were performed on the TREC Microblog 2015 collection. Results show that integrating contextual features has a positive impact on the effectiveness of the filtering without penalizing efficiency.*

MOTS-CLÉS : *RI Contextuelle, Filtrage temps-réel, Microblogs.*

KEYWORDS: *Contextual IR, Real-Time Filtering, Microblogs*

1. Introduction

Les plateformes de microblogging, initialement conçues comme des outils de communication, ont vu leur utilisation évoluer dans de nouvelles directions, telle que la collecte d'information en temps-réel. Twitter est l'exemple type de ce genre de plateformes avec 313 millions d'utilisateurs uniques et actifs par mois et plus de 500 millions de messages (tweets) par jour[1]. Une grande partie des utilisateurs de Twitter, que nous appellerons « auditeurs » dans cet article, se contentent de lire les tweets d'autres utilisateurs sans jamais poster eux-mêmes de message. Twitter est alors considéré comme une source d'information temps-réel et les auditeurs scannent continuellement le flux de tweets postés par les personnes auxquelles ils sont abonnés. Leur but est de trouver des informations à la fois nouvelles (qu'ils n'ont jamais lues auparavant), récentes (qui viennent de sortir) et précises (qui les concernent).

Dans la plupart des cas, les utilisateurs actifs (les personnes suivies) postent des messages qui peuvent être regroupés en différents thèmes précis et distincts. Les auditeurs ont leurs propres centres d'intérêts et ils doivent identifier quelles sont les bonnes personnes à suivre, pour ensuite filtrer les messages pertinents issus des différents flux d'information auxquels ils sont abonnés. Certains comptes spécifiques essayent d'aider ces auditeurs en se concentrant sur un sujet en particulier. Leur but est de retweeter (c'est-à-dire relayer un message posté par un autre utilisateur sans aucune modification) le plus de contenu possible sur ce sujet spécifique. (Zhao et Tajima, 2014) les appellent des « comptes portails ». Ces comptes sont à l'heure actuelle administrés manuellement par une ou plusieurs personnes. Un administrateur de compte portail est confronté à plusieurs challenges : (i) il doit sélectionner un nombre approprié de tweets par unité de temps (comme une journée) afin d'éviter de surcharger ses followers[2] avec trop de messages, (ii) il doit également le faire le plus rapidement possible, car être le premier à apporter l'information est essentiel (Takemura et Tajima, 2012), et enfin par dessus tout (iii) il ne doit pas laisser passer d'information cruciale.

L'aspect le plus important ici, au-delà de l'habituel critère de pertinence des résultats, est la vitesse de retransmission (c'est-à-dire le temps écoulé entre la première émission d'une information nouvelle et son relais par le compte portail). En effet, la valeur d'une information va décroître au fur et à mesure que le temps passe avant que l'auditeur ne la lise. Une situation encore plus problématique survient si l'auditeur en question lit cette même information à partir d'une autre source. Il peut dans ce cas interrompre son abonnement au portail. Donner la sensation d'être « le premier à savoir » aux auditeurs est l'atout majeur de ce type de systèmes. Toutefois il existe un risque de surcharge de l'auditeur en étant trop rapide à retransmettre de trop nombreux messages. Supprimer un message une fois qu'il est relayé étant impossible, les comptes portails doivent impérativement trouver un équilibre entre élire les meilleurs messages candidats dans un intervalle de temps donné et la vitesse de retransmission.

1. `http://about.twitter.com/fr/company`, `http://www.internetlivestats.com/`, 2016

2. Dans le jargon Twitter, les followers sont les personnes qui sont abonnées au compte.

Notre question de recherche est d'automatiser le fonctionnement de ces comptes portails, c'est-à-dire de concevoir un outil de recommandation de retweet automatique en temps-réel. Nous proposons un modèle qui suggère automatiquement un ensemble restreint de tweets pertinents provenant d'un flux de données selon un centre d'intérêt donné. Ce modèle repose sur un ensemble de caractéristiques de contexte associées à chacun des tweets, combiné à un traitement du contenu du message en lui-même. De plus, le modèle est conçu pour respecter des contraintes de temps de traitement et ainsi retransmettre les messages sélectionnés dans les plus brefs délais.

Cet article est organisé comme suit. La section 2 formalise notre problématique. Les travaux de la littérature ainsi que leurs différences avec notre approche sont présentés dans la section 3. Notre modèle est décrit dans la section 4. La section 5 présente les expérimentations menées sur la collection TREC Microblog 2015 ainsi qu'une analyse des résultats obtenus. Pour finir, la partie 6 conclut cet article et présente les améliorations envisagées.

2. Formalisation du problème

Étant donné le flux Twitter $\mathcal{T}$ de tweets t_i, un centre d'intérêt d'un auditeur (appelé profil utilisateur) $p = \{w_0^p, ..., w_k^p\}$ composé de k termes w_j^p, et un intervalle de temps Δ modélisant la période de validité de p (par exemple un mois), notre objectif est de filtrer $\mathcal{T}$ en ne conservant qu'un nombre limité N de tweets pertinents en fonction de p au cours d'intervalles plus restreints δ (par exemple une heure) de Δ.

Chaque tweet est représenté comme suit : $t_i = (\mathcal{CO}_{t_i}, \mathcal{M}_{t_i}, ts_{t_i})$, où $\mathcal{CO}_{t_i} = \{w_0^{t_i}, ..., w_n^{t_i}\}$ est l'ensemble des termes $w_j^{t_i}$ composant le contenu du tweet t_i, $\mathcal{M}_{t_i}$ est l'ensemble de méta-données associées à ce tweet (hashtags, urls, images, likes, auteur...), et ts_{t_i} est l'étiquette temporelle associée, c'est-à-dire la date de publication initiale du tweet.

Nous définissons la fonction de décision φ comme suit :

$$\varphi(t_i, p, \mathcal{T}_s, \mathcal{R}) = \begin{cases} \{rt_i\} & \text{si } t_i \text{ est sélectionné pour être retweeté} \\ \emptyset & \text{sinon} \end{cases} \quad [1]$$

où $\mathcal{T}_s$ est l'ensemble de tweets déjà sélectionnés de $\mathcal{T}$ pendant Δ, $\mathcal{R}$ est un ensemble de ressources, et $rt_i = (\mathcal{CO}_{t_i}, \mathcal{M}_{t_i}, ts_{rt_i})$ est le tweet correspondant au retweet de t_i à l'étiquette temporelle ts_{rt_i}, avec $ts_{rt_i} > ts_{t_i}$.

La fonction φ doit considérer certaines problématiques :

– pour éviter la surcharge de l'utilisateur, φ ne doit pas renvoyer plus de N tweets au cours d'un intervalle donné δ (**P1**),

– pour ne pas notifier un utilisateur plusieurs fois pour le même sujet, la nouveauté d'un tweet t_i vis à vis de $\mathcal{T}_s$ doit être prise en compte (**P2**),

– pour empêcher l'obsolescence d'un tweet, φ doit minimiser $\sigma = ts_{rt_i} - ts_{t_i}$ (**P3**),

– lorsqu'aucun tweet pertinent n'apparaît durant δ, φ doit se comporter comme la fonction $vide$ qui ne renvoie jamais aucun message (**P4**). En revanche, elle doit bien sûr identifier les tweets pertinents lorsque ceux-ci apparaissent.

3. État de l'art

Twitter attire l'attention des chercheurs depuis de nombreuses années maintenant, et ce dans de nombreux domaines différents comme la détection d'événements, l'analyse de sentiments, les systèmes considérant le contexte, etc. Néanmoins, très peu de travaux avant 2015 se concentrent sur la recommandation de retweets (Kywe *et al.*, 2012).

Le premier travail proche de notre approche a été réalisé par (Zhao et Tajima, 2014). Quatre algorithmes ont été proposés pour automatiser la sélection de tweets, mais seulement deux d'entre eux fonctionnent en véritable temps-réel (c'est-à-dire en retweetant instantanément après le post initial). La sélection de tweet est basée sur la similarité par cosinus entre le contenu du tweet et les centres d'intérêts. Le premier algorithme utilise un seuil global pour chacun des tweets et ajuste ensuite ce seuil en exploitant des données provenant de l'unité de temps précédente (l'heure précédente), en supposant que « les seuils optimaux pour des intervalles de temps consécutifs ne varient que très peu ». Le second algorithme utilise un seuil stochastique. Toutefois, aucun de ces algorithmes proposés n'utilise d'informations relatives au contexte, et les algorithmes en véritable temps-réel sont nettement moins efficaces que ceux en pseudo temps-réel.

En 2015, la tâche TREC Microblog a fourni à la communauté RI un corpus temps-réel réutilisable et a ainsi permis au problème de recommandation de retweets d'être au centre d'une attention nouvelle. Plusieurs approches ont été proposées lors de la tâche. La plus efficace a été réalisée par l'Université de Waterloo (Tan *et al.*, 2015). Cette approche considère en premier lieu la qualité du message en supprimant tous ceux avec trop peu de mots significatifs ou au contraire trop de mots-clés (hashtags). Dans un deuxième temps, une expansion de requête est réalisée grâce à 17 mois de données Twitter collectées. Pour sélectionner les tweets pertinents, l'approche proposée utilise une combinaison de modèles vectoriels appliqués à toutes les différentes parties des requêtes fournies par TREC (c'est-à-dire le titre, la description mais aussi la partie narrative). Enfin, afin de sélectionner les messages à retweeter, deux stratégies de fenêtrage temporel sont appliquées : des fenêtres temporelles fixes et des fenêtres avec seuils dynamiques. La première méthode sélectionne le tweet avec le plus haut score au sein de chaque fenêtre de x minutes au cours de la journée. La seconde utilise des fenêtres dynamiques (x évolue tout au long de la journée selon le temps écoulé entre deux notifications) pour sélectionner le meilleur candidat parmi les tweets. Une deuxième approche a été proposée par l'Université de Pékin (Fan *et al.*, 2015). Une expansion de requête (ou de profil utilisateur) est réalisée en utilisant l'API de Google

et une divergence de Kullback-Leibler est appliquée aux tweets sélectionnés. La principale différence réside dans le traitement manuel effectué pour mettre à jour le seuil utilisé lors de la sélection des tweets. Un être humain doit parcourir les cents meilleurs tweets sélectionnés la veille pour chacun des profils utilisateurs et ensuite déterminer la limite inférieure du seuil pour les sélections du jour à venir. Une autre approche, proposée par l'Université de Changsha (Zhu *et al.*, 2015), combine certains des principes de chacune des deux approches précédentes, comme la prise en considération de la qualité du tweet en plus du traitement sur le contenu, mais aussi l'expansion de requête grâce à l'API de Google, et enfin la mise à jour dynamique des seuils de sélection.

Par ailleurs, plusieurs travaux ont étudié l'importance du contexte dans la RI sociale. À propos de l'information disponible sur l'utilisateur, (Jabeur *et al.*, 2012) s'est intéressé à l'influence de l'auteur et à l'importance de la structure du réseau social dans le but de combiner une estimation thématique et sociale de la pertinence d'un tweet. Entre autres choses, (Damak *et al.*, 2013) a montré que la simple présence d'URL dans un tweet est un facteur déterminant de pertinence étant donné un besoin en information précis. La limite majeure de ces différentes approches réside dans le manque d'application en contexte temps-réel. En effet, ces approches utilisent des tweets en tant que collection statique de documents et non pas comme un flux temps-réel. Contrairement à notre approche qui traite les documents les uns après les autres, elles peuvent utiliser des processus de RI classiques appropriés aux collections statiques comme les fonctions d'indexation et de classement. Certains autres travaux se sont concentrés sur des éléments propres à Twitter, tels que que les entités, qui sont partie intégrante de la structure des tweets. Les hashtags créent des liens entre les documents ou des fils de discussion. Les mentions relient les personnes ou les organisations entre elles et modélisent le réseau social. Des exemples tels que (Efron, 2010) en recherche par mots-clés, ou (Guille et Favre, 2014) en détection d'événements, ont montré le rôle prépondérant de ces entités. Néanmoins, ils se sont concentrés uniquement sur ces caractéristiques précises et n'utilisent pas de flux temps-réel. Notre travail est plus proche de l'état de l'art de (Cheng *et al.*, 2013) qui utilise plusieurs caractéristiques de contexte, telles que la richesse du tweet, son autorité, sa fraîcheur, et enfin l'analyse de sentiments associée. Cependant la collection de documents considérée est, une fois encore, statique et non pas un véritable flux.

4. Modèle Contextuel pour la Recommandation de Retweet

L'évaluation de la fonction de décision de retweet φ est divisée en deux parties : la première portant sur le Contenu et la seconde sur le Contexte, comme décrit dans la figure Fig. 1. Tout d'abord, un pré-traitement est effectué à la fois sur le contenu du tweet $\mathcal{CO}_{t_i}$ et sur le profil utilisateur p, dans le but de maximiser le nombre de correspondances possibles. Ce pré-traitement peut être la suppression des mots vides, la lemmatisation, etc. Le résultat de ce pré-traitement est une représentation du profil

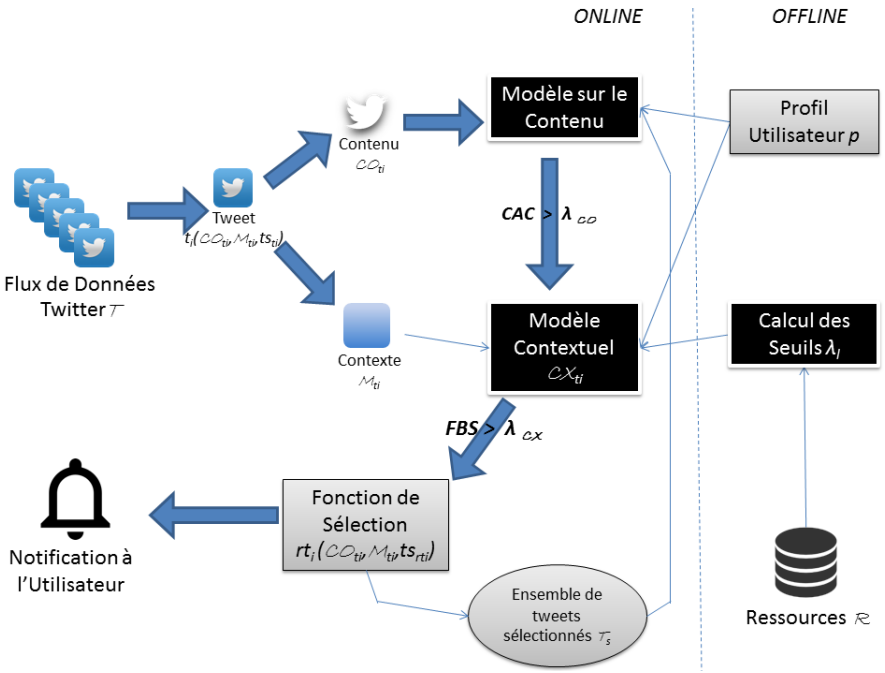

Figure 1 : Processus global basé sur le modèle contextuel pour la recommandation de retweet

$p' = \{w'^{p}_0, ..., w'^{p}_{k'}\}$ composée de $k' \leq k$ termes w'^{p}_j, ainsi qu'un contenu de tweet pré-traité $\mathcal{CO}'_{t_i} = \{w'^{t_i}_0, ..., w'^{t_i}_{n'}\}$ composé de $n' \leq n$ termes $w'^{t_i}_j$.

4.1. *Modèle sur le Contenu*

Le principal objectif du modèle sur le contenu est d'évaluer la pertinence du tweet vis à vis d'un profil utilisateur, c'est-à-dire trouver une correspondance entre $\mathcal{CO}'_{t_i}$ et p'. Un critère d'acceptation du contenu *CAC* (Content Acceptance Criterion) est alors évalué comme suit :

$$CAC(p', \mathcal{CO}'_{t_i}, \mathcal{T}_s) = \begin{cases} \frac{\sum_{j=1}^{n'} \mathbb{1}_{\{w'^{t_i}_j \in p'\}}}{\sum_{j=1}^{k'} \mathbb{1}_{\{w'^{p}_j \in p'\}}} \text{ si } sim(\mathcal{CO}'_{t_i}, \mathcal{T}_s) < \lambda_{nov} \\ 0 \text{ sinon} \end{cases} \quad [2]$$

où $\mathbb{1}$ est une fonction indicatrice qui est égale à 1 si la condition liée est satisfaite et 0 sinon (par exemple, $\mathbb{1}_{\{x \in \mathbb{R}\}} = 1$; si $x \in \mathbb{R}$).

Concrètement, nous calculons le rapport entre le nombre de mots communs entre le contenu du message $\mathcal{CO}'_{t_i}$ et le profil utilisateur p', sur le nombre de mots dans

p'. t_i doit également être assez différent des autres messages déjà retweetés appartenant à $\mathcal{T}_s$. La nouveauté de t_i selon $\mathcal{T}_s$ est évaluée grâce à une fonction de similarité $sim(\mathcal{CO}'_{t_i}, \mathcal{T}_s)$ dont le résultat doit être inférieure à un seuil λ_{nov}.

4.2. *Modèle sur le Contexte*

En plus d'un traitement sur le contenu, nous associons à chaque tweet un contexte autour du message $\mathcal{CX}_{t_i}$ composé d'informations supplémentaires. Ces informations sont soit directement extraites soit calculées à partir des métadonnées $\mathcal{M}_{t_i}$. $\mathcal{M}_{t_i}$ est défini par $(st_{t_i}, hash_{t_i}, men_{t_i}, url_{t_i}, med_{t_i}, us_{t_i})$ où :

– st_{t_i} est le statut du retweet (message initial ou retweet d'un autre utilisateur),

– $hash_{t_i}$ est l'ensemble des hashtags dans t_i,

– men_{t_i} est l'ensemble des mentions dans t_i,

– url_{t_i} est l'ensemble des urls dans t_i,

– med_{t_i} est l'ensemble des médias (image, son, vidéo, etc.) dans t_i,

– us_{t_i} regroupe des informations sur l'auteur de t_i. $us_{t_i} = \big(fol_{t_i}, stat_{t_i}, fr_{t_i}, list_{t_i}, fav_{t_i}, desc\big)$ où fol_{t_i} est le nombre de followers, $stat_{t_i}$ est le nombre de statuts (nombre de tweets et retweets créés par l'auteur), fr_{t_i} est le nombre d'amis, $list_{t_i}$ est le nombre de listes publiques dont l'auteur est membre, fav_{t_i} est le nombre de favoris, et $desc_{t_i}$ est la description de profil de l'auteur composée de *nDesc* termes $w^d{}_j$.

$\mathcal{CX}_{t_i}$ est défini comme un ensemble de caractéristiques f_l, dont les valeurs sont estimées en utilisant $\mathcal{M}_{t_i}$. Chaque caractéristique f_l est associée à un seuil qui lui est propre λ_l, au-dessus duquel un message est considéré comme pertinent au niveau de cette caractéristique particulière. Nous distinguons deux types de caractéristiques, les majeures et les mineures, en fonction de leur importance déterminée selon l'état de l'art. Le tableau 1 résume toutes les caractéristiques f_l avec leurs seuils associés λ_l, ainsi que leur importance (Majeure ou Mineure). f_9 est par exemple catégorisée comme une caractéristique majeure, puisque (Damak *et al.*, 2013) a montré que la simple présence d'une URL dans un tweet est un bon indicateur de pertinence. λ_l peut être déterminé grâce à un ensemble de ressources $\mathcal{R}$ détaillé dans la section 5. Enfin, les caractéristiques sont organisées en trois catégories : Contenu, Entités et Auteur.

Pour chaque caractéristique f_l, un score $Sc_l(f_l, p, \mathcal{R})$ est défini comme suit :

$$Sc_l(f_l, p, \mathcal{R}) = \begin{cases} \sum_{y=1}^{|hash_{t_i}|} \mathbb{1}_{\{f_l > \lambda_l\}} & \text{si } l = 6 \\ \sum_{y=1}^{|men_{t_i}|} \mathbb{1}_{\{f_l > \lambda_l\}} & \text{si } l = 8 \\ \mathbb{1}_{\{f_l > \lambda_l\}} & \text{sinon} \end{cases} \quad [3]$$

Tableau 1 : Caractéristiques f_l étudiées pour le modèle contextuel. L'obtention des caractéristiques est extraite à partir des métadonnées (M) ou calculée (C). Le mode de détermination des seuils λ_l est indiquée soit par $*$ pour la méthode statique utilisant $\mathcal{R}$, soit † pour la méthode par étude pilote, ou encore ‡ si définie dans la littérature. Si besoin, la référence est indiquée dans la description.

	f_l	Obtention	λ_l	Importance	Description
Contenu	f_1	M	0 ‡	Majeure	Message initial ou retweet d'un autre utilisateur. 1 si message initial ; 0 sinon (Cheng *et al.*, 2013)
	f_2	C	10 †	Mineure	Nombre de termes après pré-traitement (n')
	f_3	C	0.6 †	Mineure	Rapport entre n' et n (nombre de termes avant pré-traitement) (Cheng *et al.*, 2013)
	f_4	C	0.6 †	Mineure	Rapport entre la taille de $hash_{t_i}$ et n
Entités	f_5	C	1 †	Mineure	Taille de $hash_{t_i}$ (Aisopos *et al.*, 2012)
	f_6	C	0 ‡	Mineure	Pour chaque h de $hash_{t_i}$, présence de h dans le profil p
	f_7	C	0 †	Mineure	Taille de men_{t_i} (Aisopos *et al.*, 2012)
	f_8	C	0 †	Mineure	Pour chaque m de men_{t_i}, présence de m dans le profil p
	f_9	M	0 ‡	Majeure	Taille de url_{t_i} (Damak *et al.*, 2013)
	f_{10}	M	0 †	Mineure	Taille de med_{t_i}
Auteur	f_{11}	M	945 $*$	Majeure	fol_{t_i} (Aisopos *et al.*, 2012)
	f_{12}	M	27689 $*$	Majeure	$stat_{t_i}$ (Cheng *et al.*, 2013 ; Aisopos *et al.*, 2012)
	f_{13}	M	759 $*$	Majeure	fr_{t_i} (Aisopos *et al.*, 2012)
	f_{14}	M	7 $*$	Mineure	$list_{t_i}$
	f_{15}	M	3166 $*$	Mineure	fav_{t_i}
	f_{16}	C	0.3 †	Mineure	Similarité Cosinus entre la description de l'auteur $desc_{t_i}$ et le profil p

Chaque caractéristique est associée à un score égal à 1 si sa valeur est plus élevée que λ_l. La seule différence provient de f_6, respectivement f_8, dont le score est un cumul pour chacun des hashtags présents, respectivement pour chaque mention, s'il est présent dans le profil utilisateur p.

Un score basé sur les caractéristiques de contexte FBS (Feature-Based Score) est calculé pour chaque tweet comme suit :

$$FBS(\mathcal{CX}_{t_i}, p, \mathcal{R}) = \sum_{l=1}^{m} [(Sc_l(f_l, p, \mathcal{R}) \,.\, \mathbb{1}_{\{l \in \mathcal{CX}^m_{t_i}\}}) + 2\,(Sc_l(f_l, p, \mathcal{R}) \,.\, \mathbb{1}_{\{l \in \mathcal{CX}^M_{t_i}\}})] \quad [4]$$

où $\mathcal{CX}^m_{t_i}$ est l'ensemble des caractéristiques mineures et $\mathcal{CX}^M_{t_i}$ est l'ensemble des caractéristiques majeures, avec $\mathcal{CX}_{t_i} = \mathcal{CX}^M_{t_i} \cup \mathcal{CX}^m_{t_i}$ et $\mathcal{CX}^M_{t_i} \cap \mathcal{CX}^m_{t_i} = \emptyset$.

4.3. *Fonction de Décision*

Au final, la fonction de décision φ est définie de la manière suivante :

$$\varphi(t_i, p, \mathcal{T}_s, \mathcal{R}) = \begin{cases} \{rt_i\} & \text{si } CAC(p', \mathcal{CO}'_{t_i}, \mathcal{T}_s) > \lambda_{\mathcal{CO}} \text{ et } FBS(\mathcal{CX}_{t_i}, p, \mathcal{R}) > \lambda_{\mathcal{CX}} \\ \emptyset & \text{sinon} \end{cases} \quad [5]$$

où $\lambda_{\mathcal{CO}}$ et $\lambda_{\mathcal{CX}}$ sont les seuils associés au CAC et au FBS. Un tweet est alors retweeté si son contenu et son contexte sont tous les deux considérés suffisamment pertinents selon le profil utilisateur p ciblé.

Utiliser des valeurs appropriées pour $\lambda_{\mathcal{CO}}$ et $\lambda_{\mathcal{CX}}$ permet de répondre à **(P1)** et **(P4)**. Ces valeurs peuvent être fixes, ou peuvent évoluer avec le temps, comme proposé dans (Tan *et al.*, 2015). **(P2)** est pris directement en compte dans le calcul du CAC, puisque chaque tweet est comparé à l'ensemble des messages déjà retweetés $\mathcal{T}_s$ (cf. Eq.2). Pour répondre à **(P3)**, tout en conservant σ aussi faible que possible, le FBS est calculé uniquement si $CAC > \lambda_{\mathcal{CO}}$.

5. Expérimentations

Nous avons choisi d'évaluer notre modèle sur le corpus de la tâche Microblog 2015 de TREC (Text REtrieval Conference). Cette tâche correspond au problème que nous abordons car elle utilise un flux de données temps-réel véritable, et l'objectif est d'envoyer des notifications à des utilisateurs selon leurs centres d'intérêt.
Nous avons implémenté notre modèle pour répondre aux directives de la tâche. Notre adaptation se concentre sur le temps de réponse : ne jamais dépasser la minute quel que soit le nombre de tweets entrants. Le processus et son évaluation dans son ensemble n'impliquent à aucun moment une intervention humaine, ils sont complètement automatiques.

5.1. *Présentation de la tâche*

En 2015, la tâche TREC Microblog[3] s'intéresse aux problématiques de filtrage temps-réel. Deux scénarios ont été mis en place afin de répondre à des probléma-

3. `https://github.com/lintool/twitter-tools/wiki/TREC-2015-Track-Guidelines`, 2015

tiques précises de filtrage temps-réel. L'idée principale du premier scénario (scénario A) était de simuler l'envoi de notifications sur téléphone mobile en temps-réel, tandis que le second (scénario B) était de créer un résumé par mail régulier à la fin de chaque journée, selon des profils utilisateurs (Lin *et al.*, 2015). Dans cet article nous nous concentrons uniquement sur le scénario A. Afin de simuler au mieux le concept de filtrage temps-réel, nous avons dû utiliser une API de Twitter pour collecter les données (le flux recueilli représente en fait 1% du flux global de Twitter) et ensuite les traiter au fur et à mesure de leur arrivée. Ce faible pourcentage est suffisant pour tester notre approche avec une moyenne de 3000 tweets collectés par minute (comparable aux systèmes de (Paik et Lin, 2015)). Quand les tweets arrivent, nous devons décider dès que possible (le laps de temps maximum pour retweeter un message proposé par TREC est fixé à 100 minutes, c'est-à-dire $\sigma < 6000$ secondes, cf. section 2) s'ils sont pertinents vis à vis d'un ou plusieurs profils utilisateurs. Pour éviter la surcharge de l'utilisateur, le nombre maximum de documents retweetés par centre d'intérêt et par jour ($\delta = 1$ jour) a été fixé à dix notifications (**P1**, $N = 10$, cf. section 2). La période officielle d'évaluation a duré dix jours sans interruption (concrètement, Δ= 10 jours). Les profils utilisateurs ont un format TREC classique, c'est-à-dire composé d'un titre, une description et une partie narrative comme montré en figure Fig. 2. Pour modéliser une situation la plus proche possible de la réalité, nous n'avons utilisé que la partie titre, qui est composée de deux à cinq mots-clés. TREC a fourni un ensemble de 225 centres d'intérêt à traiter au cours de la période d'évaluation, mais seulement 51 d'entre eux ont été jugés. Toutes les requêtes n'ont pas eu le même nombre de tweets pertinents (entre 0 et 1543 selon le profil).

```
<top>
<num> Number: MB242
<title>
Saudi bombing Yemen
<desc> Description:
Find information related to any recent bombing raids by Saudi Arabia
against the Houthi of Yemen.
<narr> Narrative:
The user is interested in the ongoing war in Yemen, and the bombing
raids by the Saudi Air force against the Shea Houthi fighters in Yemen.
He is interested in the number of raids, targets, and damage assessments.
</top>
```

Figure 2 : Exemple de requête type de TREC Microblog

Deux mesures d'évaluation ont été calculées pour chaque profil utilisateur et pour chaque jour. La première mesure est la ELG (*Expected Latency-discounted Gain*) :

$$ELG = \frac{1}{N} \sum G(t) \quad [6]$$

où N est le nombre de tweets retournés et G(t) est le gain de chaque tweet (0 pour les tweets non pertinents, 0.5 pour les tweets pertinents, et 1 pour les tweets très pertinents).

De plus, une pénalité liée au temps de réponse est appliquée à tous les tweets, calculée comme suit : MAX(0,(100-d)/100) où d est le temps écoulé en minutes entre l'émission du tweet et sa notification finale. Cette valeur décroit linéairement de telle sorte qu'au bout de 100 minutes le système reçoive un score de 0 quelle que soit la pertinence du tweet renvoyé.

La seconde mesure est la nCG (*normalized Cumulative Gain*) :

$$nCG = \frac{1}{Z} \sum G(t) \qquad [7]$$

où Z est le gain maximum (selon la limite de dix retweets par jour).

5.2. *Instanciation du modèle*

Pour utiliser la collection de la tâche TREC Microblog 2015, du point de vue du modèle sur le contenu, le même pré-traitement classique est appliqué sur les profils p ainsi que sur le contenu des tweets $\mathcal{CO}_{t_i}$: suppression des mots vides, uniformisation de la casse, et lemmatisation au moyen de l'algorithme de Porter (Porter, 1980). Une détection de la langue est aussi effectuée afin de ne conserver que les messages écrits dans la même langue que les profils étudiés. Pour conserver σ le plus faible possible, et puisque les profils fournis pour la tâche ne concernent que très peu de tweets similaires durant la période Δ, la similarité $sim(\mathcal{CO}'_{t_i}, \mathcal{T}_s)$ n'a pas été considérée dans l'équation 2.

Au niveau du modèle contextuel, l'ensemble de ressources $\mathcal{R}$ utilisé pour calculer $\mathcal{CX}_{t_i}$ est composé de données Twitter collectées au cours des six semaines précédant la période officielle de traitement de la tâche TREC Microblog 2015. $\mathcal{R}$ est utilisé pour fixer les seuils λ_{11} à λ_{15} associés aux caractéristiques f_{11} à f_{15} en calculant le troisième quartile des différentes caractéristiques en question. Les seuils pour les caractéristiques restantes ont été déterminés manuellement par observation de leurs distributions ou encore selon l'état de l'art lié.

Au niveau des deux seuils globaux $\lambda_{\mathcal{CO}}$ et $\lambda_{\mathcal{CX}}$, associés à CAC et FBS, nous avons tout d'abord implémenté notre modèle avec plusieurs paires de valeurs ($\lambda_{\mathcal{CO}}$, $\lambda_{\mathcal{CX}}$). Dans un premier temps, $\lambda_{\mathcal{CX}}$ a été fixé à 5 (afin d'éliminer les tweets avec des contextes pauvres), puis nous avons testé notre modèle en faisant varier $\lambda_{\mathcal{CO}}$ de 0.1 à 0.9 par pas de 0.1. Dans un second temps, nous avons fixé $\lambda_{\mathcal{CO}}$ à 0.6 (qui est le meilleur seuil obtenu lors des tests précédents), puis nous avons testé notre modèle en faisant varier $\lambda_{\mathcal{CX}}$. Le cas particulier de $\lambda_{\mathcal{CX}}$ égal à 0 correspond à l'omission complète du contexte. Une seconde implémentation de notre modèle (variante avec *fenêtrage temporel*) prend en considération l'évolution dynamique des deux seuils $\lambda_{\mathcal{CO}}$ et $\lambda_{\mathcal{CX}}$ au cours du traitement des tweets, comme proposé dans (Tan *et al.*, 2015) par exemple, mais avec une différence significative. Nous avons travaillé sur la supposition (vérifiée) que le nombre de messages postés n'est pas régulier au cours de la journée, mais au contraire que des pics apparaissent chaque jour approximativement à la même heure. Afin de déterminer nos deux seuils $\lambda_{\mathcal{CO}}$ et $\lambda_{\mathcal{CX}}$ pour un créneau

horaire particulier du jour en cours, nous récoltons les données du même créneau horaire des jours précédents. Chaque heure, les seuils sont mis à jour grâce aux données (CAC et FBS) des jours précédents durant cette heure précise avec un poids plus important pour la veille. Concrètement, $\lambda_{\mathcal{CO}}$, respectivement $\lambda_{\mathcal{CX}}$, est mis à jour par la moyenne entre le dernier $\lambda_{\mathcal{CO}}$, respectivement $\lambda_{\mathcal{CX}}$, et le premier quartile des valeurs de CAC, respectivement FBS, de la veille, dans l'ensemble $\mathcal{T}_s$. Ce principe donne une importance plus grande au jour précédent qu'à l'ensemble des jours encore antérieurs.

Comme présenté en section 5.1, la mesure d'évaluation ELG comprend une pénalité liée au temps de réponse du système depuis la première minute jusqu'à la centième. Afin de pleinement répondre à (**P3**) et de ne pas être pénalisés en termes d'ELG, nous avons fixé notre propre temps de réponse maximum σ à 60 secondes.

5.3. *Résultats globaux*

Le tableau 2 résume certains des résultats les plus représentatifs obtenus avec notre modèle CBM (*Context-Based Model*) pour les mesures d'évaluation ELG et nCG. Notre modèle a été également comparé au système ayant eu les meilleurs résultats à la tâche TREC Microblog 2015 (*Meilleur Run Officiel*) soumis par l'Université de Waterloo (Tan *et al.*, 2015) ainsi que le système *Vide*, qui ne renvoie aucun tweet pendant toute la période d'évaluation ($\mathcal{T}_s = \emptyset$). Étant donné qu'il existe un nombre non négligeable de jours au cours desquels certains profils n'ont pas eu de tweets pertinents associés, ce système obtient des résultats étonnamment élevés. Les résultats pour notre approche CBM sont divisés en plusieurs catégories : la première partie est relative aux différentes valeurs prises par $\lambda_{\mathcal{CO}}$, et la seconde est relative à celles de $\lambda_{\mathcal{CX}}$ (cf. section 5.2). Les valeurs entre crochets correspondent aux valeurs de $\lambda_{\mathcal{CO}}$, respectivement $\lambda_{\mathcal{CX}}$. CBM_d correspond à la variante du modèle simulant une expansion de requête grâce à la partie description des profils fournis par TREC. CBM_tw correspond à la variante qui teste le *fenêtrage temporel* pour la mise à jour progressive des seuils globaux (cf. section 5.2). Tous les autres tests de CBM ont été effectués avec des seuils globaux fixes.

La première conclusion émanant de ce tableau est que pratiquement toutes les versions de notre modèle surpassent significativement la baseline du système *Vide*, nous permettant ainsi de répondre à (**P4**). Dans un deuxième temps, la variante de CBM atteignant le meilleur résultat (parmi toutes les variantes testées) correspond aux seuils globaux $\lambda_{\mathcal{CO}} = 0.6$ pour CAC et $\lambda_{\mathcal{CX}} = 5$ pour FBS. Ce dernier égale pratiquement le meilleur système officiel automatique de TREC selon ELG et le dépasse au niveau de nCG. De plus, les résultats montrent l'efficacité du modèle contextuel. En effet, à traitement égal sur le contenu (même valeur de $\lambda_{\mathcal{CO}}$ égale à 0.6), la non prise en compte du contexte (qui équivaut à $\lambda_{\mathcal{CX}} = 0$) donne une efficacité bien inférieure ($\sim -9\,\%$). En revanche, l'extension des profils à partir de la description des requêtes TREC uniquement (run CBM_d) ne permet pas d'améliorer le modèle.

Tableau 2 : Performances de notre modèle. Le T-test pairé bilatéral par rapport au système vide est indiqué par $*$, indiqué par † par rapport au meilleur run officiel TREC de l'Université de Waterloo et indiqué par ‡ par rapport au système sans prise en compte du contexte (correspondant à CBM[0.6 ;0]). Un seul symbole montre une différence significative ($p\text{-}value < 0.05$) et deux une différence très significative ($p\text{-}value < 0.01$).

Runs	**ELG**	**nCG**
Système Vide	0.2471 †† ‡‡	0.2471
Meilleur Run Officiel	0.3150 **	0.2679
CBM [0.4 ;5]	0.2381 †† ‡‡	0.2363 ‡
CBM [0.5 ;5]	0.3049 **	0.2891 *
CBM [0.6 ;5]	**0.3145** **	**0.2917** **
CBM [0.7 ;5]	0.2525 † ‡	0.2486
CBM_d [0.6 ;5]	0.3078 **	0.2480
CBM [0.6 ;0]	0.2902 **	0.2798
CBM [0.6 ;3]	0.2943 **	0.2824 *
CBM [0.6 ;8]	**0.2996** **	**0.2872** *
CBM [0.6 ;10]	0.2758	0.2680
CBM_tw	0.2764	0.2630

Une autre conclusion est que lorsque notre modèle est trop restrictif, c'est-à-dire avec des seuils trop élevés, l'efficacité décroit de nouveau. À l'inverse, si notre modèle est trop permissif, de trop nombreux tweets non pertinents sont retransmis et l'efficacité est une fois encore réduite. La plupart des tests de significativité présentés ici ne sont pas statistiquement concluant, mais cela peut s'expliquer par le très grand nombre de requêtes obtenant le même score d'un modèle à l'autre à cause du faible nombre de retweets qui leur sont associés. Des différences significatives ne sont observables que sur les requêtes ayant un nombre élevé de notifications pertinentes.

La variante du modèle utilisant les fenêtres temporelles n'améliore pas non plus l'efficacité (run CBM_tw). De manière plus approfondie, les profils avec peu de tweets pertinents sont trop fortement détériorés, tandis que dans le même temps les autres ne profitent pas d'une amélioration suffisante. Un tel comportement pourra certainement être amélioré avec des seuils adaptés à chaque profil utilisateur plutôt que d'utiliser le même seuil tous profils confondus. Nous étudierons cette piste dans de futurs travaux.

Enfin, il faut noter que notre objectif d'efficience (**P3**), qui était d'envoyer chaque notification en moins d'une minute ($\sigma < 60$), est pleinement atteint. En effet, même au cours des périodes à forte affluence de tweets, un message estimé pertinent a été retweeté en un maximum de 5 secondes ($\sigma < 5$). De plus, le processus complet a été bien plus rapide qu'escompté puisque ce σ a été calculé alors que le modèle traitait simultanément les 225 profils utilisateurs et non pas un seul.

5.4. *Analyse par caractéristique*

Dans cette partie nous étudions l'impact de nos différents groupes de caractéristiques (c'est-à-dire Contenu, Entités et Auteur) sur le modèle contextuel global. Les résultats sont présentés dans le tableau 3. $\lambda_{\mathcal{CO}}$ est fixé à 0.6 et les résultats sont présentés ici uniquement pour la valeur optimale de $\lambda_{\mathcal{CX}}$ pour chacune des configurations. Les résultats de la colonne $\emptyset$ correspondent à notre modèle sans la partie sur le contexte (run CBM[0.6 ;0] du tableau 2).

Tableau 3 : Résultats par groupe de caractéristiques. Ces groupes sont : Contenu (C), Entités (E), et Auteur (A), comme décrit dans le tableau 1. La colonne $\emptyset$ correspond à notre modèle sans la partie sur le contexte.

	Groupes de caractéristiques							
	$\emptyset$	**C**	**E**	**A**	**C+E**	**C+A**	**E+A**	**C+E+A**
ELG	0.2902	0.2859	0.2963	0.3026	0.2994	0.3092	0.3088	**0.3145**
nCG	0.2798	0.2791	0.2825	0.2844	0.2826	0.2878	0.2879	**0.2917**

Tout d'abord, aucune combinaison des groupes de caractéristiques n'obtient de meilleur résultat que le modèle contextuel complet (comprenant donc Contenu, Entités et Auteur – C+A+E), ce qui démontre l'utilité de ce modèle global. En regardant de manière plus précise les résultats impliquant un seul groupe, nous remarquons qu'utiliser uniquement le Contenu dégrade nettement les résultats même face au modèle sans contexte du tout. Le groupe de caractéristiques Auteur semble être le plus important ici (la variante A comparée aux variantes C et E mais aussi les variantes C+A et E+A comparées à C+E). Ainsi, les trois groupes sont nécessaires à CBM, et le groupe Contenu devrait être améliorée pour apporter autant au modèle que les deux autres groupes.

5.5. *Analyse par requête*

En analysant les évaluations de pertinence des centres d'intérêts fournis, nous avons pu différencier 5 catégories de profils utilisateurs selon la fréquence d'arrivée des tweets pertinents associés. La première catégorie C1 rassemble les profils avec très peu de tweets pertinents (moins de 20 au cours des 10 jours d'évaluation). C2 est caractérisée par 2 pics de messages pertinents, un au début et l'autre à la fin de la période, tandis que le début de C3 est quasiment vide puis des pics réguliers apparaissent par la suite. C4 quant à elle est caractérisée par un très fort pic au commencement puis seulement quelques messages pertinents arrivent sur le reste de l'évaluation, tandis que C5 voit arriver des pics réguliers au cours de l'ensemble de la période évaluée. Les résultats par catégorie de requêtes sont présentés dans le

tableau 4.

Tableau 4 : Résultats par catégories de profils utilisateurs.

	Catégories				
	C1	**C2**	**C3**	**C4**	**C5**
Nombre de Profils	21	5	10	7	8
ELG	0.56	0.09	0.13	0.10	0.24
nCG	0.55	0.05	0.08	0.09	0.19

Notre approche est efficace pour les profils avec peu de tweets pertinents (C1) mais semble être trop restrictive pour les requêtes ayant un ensemble de résultats plus grand. Par exemple, ces résultats pourraient être améliorés d'environ $36\,\%$ pour le profil intitulé MB371 (de la catégorie C4) en abaissant le seuil global du contenu. De futurs travaux tendront à adapter les seuils à chaque requête ou au moins à chaque catégorie de requêtes préalablement détectée.

6. Conclusion et évolutions

Dans cet article, nous avons proposé un nouveau modèle basé sur le contexte pour des systèmes de recommandation de retweet (comptes portails). Une fonction d'évaluation utilisant 2 seuils, relatifs d'une part au contenu et d'autre part au contexte, est mise en place pour prendre la décision de retransmettre le message capté dans le flux de données. Seize caractéristiques regroupées en trois catégories (Contenu, Entités et Auteur) sont utilisées pour modéliser le contexte. Les résultats obtenus sur la collection de la tâche TREC Microblog 2015 montrent à la fois l'efficacité et l'efficience de cette approche. Une analyse plus en profondeur des résultats nous indique certaines pistes pour la suite de nos travaux. En premier lieu, nous devrions poursuivre nos analyses sur chacune de nos caractéristiques de contexte f_l, leur impact sur le modèle global et les interactions qui les relient. Ensuite, les résultats des analyses d'ores et déjà conduites semblent montrer que les seuils devraient être fixés selon chacun des différents types de requêtes ; leur classification nous aidera à améliorer les résultats.

7. Bibliographie

Aisopos F., Papadakis G., Tserpes K., Varvarigou T., « Content vs. Context for Sentiment Analysis : A Comparative Analysis over Microblogs », *Proceedings of Int. Conf. HT'12*, p. 187-196, 2012.

Cheng F., Zhang X., He B., Luo T., Wang W., « A Survey of Learning to Rank for Real-time Twitter Search », *Proceedings of Joint Int. Conf. ICPCA/SWS'12*, p. 150-164, 2013.

Damak F., Pinel-Sauvagnat K., Boughanem M., Cabanac G., « Effectiveness of state-of-the-art features for microblog search », *Proceedings of Int. Conf. SAC'13*, p. 914-919, 2013.

Efron M., « Hashtag retrieval in a microblogging environment », *Proceedings of Int. Conf. SIGIR'10*, p. 787-788, 2010.

Fan F., Fei Y., Lv C., Yao L., Yang J., Zhao D., « PKUICST at TREC 2015 Microblog Track : Query-biased Adaptive Filtering in Real-time Microblog Stream », *Proceedings of Int. Conf. TREC'15*, 2015.

Guille A., Favre C., « Mention-anomaly-based event detection and tracking in twitter », *Proceedings of Int. Conf. ASONAM'14*, p. 375-382, 2014.

Jabeur L. B., Tamine L., Boughanem M., « Featured tweet search : Modeling time and social influence for microblog retrieval », *Proceedings of Int. Joint Conf. WI-IAT'12*, vol. 1, p. 166-173, 2012.

Kywe S. M., Lim E.-P., Zhu F., « A survey of recommender systems in twitter », *International Conference on Social Informatics*, Springer, p. 420-433, 2012.

Lin J., Efron M., Wang Y., Sherman G., Voorhees E., « Overview of the TREC-2015 Microblog Track », *Proceedings of Int. Conf. TREC'15*, 2015.

Paik J. H., Lin J., « Do Multiple Listeners to the Public Twitter Sample Stream Receive the Same Tweets ? », *Proceedings of Int. Conf. SIGIR'15*, 2015.

Porter M. F., « An algorithm for suffix stripping », *Program*, vol. 14, n° 3, p. 130-137, 1980.

Takemura H., Tajima K., « Tweet Classification Based on Their Lifetime Duration », *Proceedings of Int. Conf. CIKM'12*, p. 2367-2370, 2012.

Tan L., Roegiest A., Clarke C. L. A., « University of Waterloo at TREC 2015 Microblog Track », *Proceedings of Int. Conf. TREC'15*, 2015.

Zhao X., Tajima K., « Online Retweet Recommendation with Item Count Limits », *Proceedings of Joint Int. Conf. WI-IAT'14*, p. 282-289, 2014.

Zhu X., Huang J., Zhu S., Chen M., Zhang C., Zhenzhen L., Dongchuan H., Chengliang Z., Li A., Jia Y., « NUDTSNA at TREC 2015 Microblog Track : A Live Retrieval System Framework for Social Network based on Semantic Expansion and Quality Model », *Proceedings of Int. Conf. TREC'15*, 2015.