République Algérienne Démocratique et populaire Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERIE DE TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de fin d'études du MASTER ACADEMIQUE

Domaine : Mathématique et Informatique

Filière : Informatique Spécialité : ingénierie des systèmes d'information

---

Présenté et soutenue le 09/10/2019 par :

Katia CHIBA

Recherche d'information dans Twitter : Proposition d'une approche integrant le profil utilisateur

---

Devant le jury composé de :

Président : Mme AMIROUCHE Fatiha

Examinateur : Mr S. SADI

Encadré par : Mme BELKACEMI Lilia

Remerciements

Au terme de la rédaction de ce mémoire, c'est un devoir agréable d'exprimer en quelques lignes mon extrême reconnaissance envers tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

Tout d'abord, j'adresse toute ma gratitude à mon encadreur MME BELKACEMI LILA, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma reflexion. Je la remercie également de m'avoir fait confiance et encourager tout au long de ce projet.

Je tiens à remercier également les membres de jury d'avoir accepter d'évaluer mon travail.

Je voudrai exprimer ma reconnaissance envers les amis (es) qui m'ont apporté leur support morale et intellectuel tout au long de ma démarche, et particulièrement mademoiselle HANACHI Feriel et mademoiselle SELMI Meliza.

Dédicaces

A mes chers parents

Ma fierté, mon bonheur et ma raison d'avancer dans la vie.

Les mots ne suffiront jamais pour exprimer ma reconnaissance et ma gratitude

Je vous dois ce que je suis aujourd'hui et ce que je deviendrai demain

Que dieu vous préserve et vous procure santé et longue vie

Me donne la chance de vous contenter tant que je vie Et la puissance de vous être demain ce que vous m'êtes aujourd'hui.

A mon petit frère

Ou devrais-je dire mon grand frère

Une personne dont je suis dextrement fière

Toujours présent à mes cotés Et prêt à tous pour m'épauler

A toute ma famille

A mes amis

Table des matières

Table des figures

Introduction

Sur les réseaux sociaux plus précisément les plateformes de microblogings tel que Twitter, les usagers produisent beaucoup de contenu. Pour n'importe quel sujet donné le nombre de publications atteint facilement des centaines de milliers. De ce fait, mettre la main sur des informations fraiches et pertinentes pourrait s'avérer être une tache trés difficile pour l'utilisateur. En outre, l'utilisateur peut se retrouver face au problème de la désorientation pour trouver des informations qui correspondent vraiment à ses besoins. La recherche d'information s'interesse à la mise en oeuvre d'approches et techniques permettant de trouver des informations pertinentes sur un sujet donné.

Dans le cadre de la recherche d'information(RI), l'importance d'un tweet est assimilée à son éditeur, où ce dernier est modélisé par l'ensemble d'informations qui lui sont relatives dans un profil appelé le profil utilisateur. Cependant, comment modéliser un profil utilisateur ? comment exploiter ce dernier pour trouver les tweets les plus pertinents pour un utilisateur ? C'est ce que nous tacherons d'expliquer au fil de notre manuscrit.

Dans notre travail, nous proposons une approche qui intégre le profil utilisateur et le temps dans Twitter. Nous nous sommes focaliser sur le nombre de followers et de tweets qui représentes deux facteurs de pertinences de l'utilisateur sur Twitter.

Organisation de la thèse Notre travail est réparti en 5 chapitres :

- **Chapitre1 :** Généralitées sur la recherche d'information, dans lequel nous présentons les différents concepts liés à la recherche d'information
- **Chapitre 2 :** Recherche d'information sociale , ce chapitre parlera des réseaux sociaux en générale et Twitter en particulier.
- **Chapitre 3 :** Etat de l'art, dans lequel nous étallerons les differents traveaux fais dans le cadre d'intégration du profil utilisateur et du temps dans Twitter.
- **Chapitre 4 :** Approches proposées : dans ce chapitre nous expliquons les deux approches que nous avons proposées.
- **Chapitre 5 :** Implémentation et teste : dans lequel nous exposerons les differents résultats obtenus en les évaluons.

Nous terminons notre mémoire sur une conclusion générale.

[Auteur]« Ma citation préférée. »" Titre de la partie 1

Chapitre 1

Généralités sur la recherche d'information

1.1 Introduction

L'objectif de la recherche d'information (RI) est de permettre à l'utilisateur un accès facile à l'information pertinente répondant au mieux à son besoin informationnel exprimé par une requête. Le présent chapitre a pour objectif de présenter brièvement les concepts de base de la RI. Nous commençons par définir ce qu'est un système de recherche d'information, puis nous nous intéresserons au processus général de la RI. Nous détaillerons par la suite les principaux modèles de la recherche d'information. Enfin nous nous interésserons aux méthodes d'évaluation d'un système de recherche d'information (SRI).

1.2 Recherche d'information

La recherche d'information est une discipline de recherche qui intègre des modèles et des techniques dont le but est de faciliter l'accès à l'information pertinente pour un utilisateur ayant un besoin en information.

1.3 Système de recherche d'information

Un SRI est un ensemble de techniques qui assurent les fonctions nécessaires pour la RI. Il a pour rôle de sélectionner dans une collection de documents préalablement enregistrés, les informations pertinentes qui répondent au mieux au besoin de l'utilisateur formulé par une requête de recherche. De cette définition, on fait sortir quelques concepts clés auxquels s'articule un système de recherche d'information à savoir : le besoin en information, la requête, le document et collection de documents, ainsi que la pertinence entre un document et une requête.

1.3.1 Le besoin d'information

Le besoin d'information est l'expression mentale d'un utilisateur, on définit trois types de besoin utilisateur :

- **Besoin de vérification :** l'utilisateur souhaite vérifier une information ou retrouver des éléments aux caractéristiques spécifiques. Il veut, par exemple, retrouver des références bibliographiques précises, Il sait que l'information existe, et parfois où il va la retrouver. La précision des recherches est alors déterminante.
- **Besoin dirigé :** l'utilisateur veut clarifier, passer en revue ou approfondir certains aspects d'un sujet connu. Il possède déjà un certain nombre de connaissances et de données relatives au sujet, comme des termes, des concepts, des représentations imagées, et il veut les compléter.
- **Besoin flou sur un sujet :** l'utilisateur veut explorer de nouveaux concepts ou relations en dehors des domaines qu'il connaît, ou les données qu'il connaît sont vagues et incomplètes. L'usager ne dispose souvent pas du vocabulaire adéquat pour formuler sa demande. Souvent, il ne connaît

pas non plus les sources qui pourraient l'aider. Ce besoin est toujours exprimé d'une façon incomplète.

1.3.2 La requête

La requête est l'expression du besoin en information de l'utilisateur. Elle est exprimée par un ensemble de mots spécifiques traduisant le besoin formulé. La requête représente l'interface entre l'utilisateur et le SRI.

1.3.3 Document /collection de documents

- **Document :** représente l'information élémentaire manipulable par le SRI. Il est représenté sous différents formats : du texte, une page web, une vidéo, une image ....
- **Collection de document :** c'est l'ensemble de documents sur lesquels porte une recherche, également appelé corpus ou fond documentaire.

1.3.4 La pertinence

La pertinence est le degré de correspondance entre un document et une requête[Borlund et al, 1998]. La pertinence peut être évaluée d'un point de vue utilisateur il s'agit alors de la pertinence utilisateur, ou d'un point de vue système on parle alors de la pertinence système.

- **La pertinence utilisateur** : Elle représente la satisfaction de l'utilisateur vis à vis des documents retournés par le système. c'est une mesure subjective c'est à dire que deux utilisateur peuvent juger différement un même document.
- **La pertinence système :** C'est une mesure d'évaluation définie à travers les modèles de RI. Souvent traduite par un score évaluant la similarité entre le contenu des documents et celui de la requête.[Boughanem et al., 2008]. -

1.4 Le processus de la recherche d'information

De façon générale, nous pouvons résumer le fonctionnement d'un SRI à travers un processus de recherche appelé processus en U de la RI illustré ci-dessous :

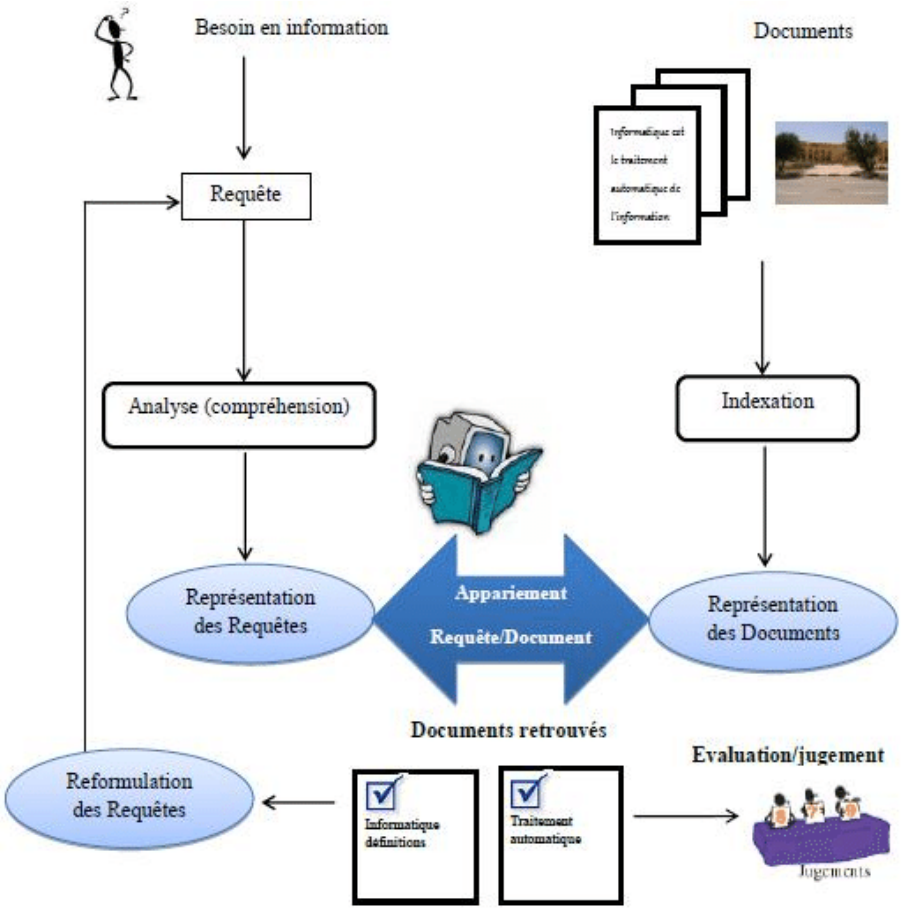


FIGURE 1.4.1 – Processus de recherche d'information [Adil Toumouh, 2015]

Le SRI utilise trois principales phases afin de renvoyer l'ensemble des documents correspondant au mieux à une requête formulé par l'utilisateur, à savoir : l'indexation, l'appariement, et la reformulation de la requête.

1.4.1 La phase d'indexation

Elle consiste à déterminer et extraire les termes représentatifs du contenu d'un document ou d'une requête, qui couvrent au mieux leur contenu sémantique. La qualité de la recherche dépend en grande partie de la qualité de l'indexation, c'est une étape capitale dans le processus de RI. Le résultat de l'indexation constitue, ce que l'on nomme le descripteur du document ou de requête. Ce dernier est souvent une liste de termes significatifs pour l'unité textuelle correspondante, généralement assortis de poids représentant leur degré de représentativité du contenu sémantique de l'unité qu'ils décrivent. l'indexation

peut être manuelle, automatique ou semi-automatique.

- **Indexation manuelle :** chaque document est analysé par un spécialiste du domaine correspondant ou par un documentaliste.
- **Indexation semi-automatique :** le choix final reste au spécialiste du domaine correspondant, qui intervient souvent pour établir des relations sémantiques entre mots-clés et choisir les termes significatifs.
- **Indexation automatique :** consiste à analyser chaque document à l'aide d'un processus entièrement automatisé.

L'indexation la plus utilisée est l'indexation automatique, celle ci effectue un ensemble de traitements sur un document dont : l'éxtraction automatique des termes du document, l'élimination des mots vides, la normalisation, la pondération, ainsi que la création de l'index.

1.4.2 L'appariement requête-document

Le SRI procède à l'appariement entre la requête et les documents. De cette mise en correspondance résulte un score de pertinence reflétant le degré de similarité entre la requête et le document, ils intègrent l'information jugée pertinente pour l'utilisateur[Hammache, 2011]. Il existe deux types d'appariement :

- **Appariement exact :** le résultat est une liste de documents respectant exactement la requête spécifiée avec des critères précis. Les documents retournés ne sont pas triés.
- **Appariement approché :** le résultat est une liste de documents censés être pertinents pour la requête. Les documents retournés sont triés selon un ordre de mesure. Cet ordre reflète le degré de pertinence document/requête

1.4.3 La reformulation de la requête

Dans les SRI, la requête initiale formulé par un utilisateur est souvent insuffisante pour répondre à son besoin informationnel. La reformulation de la requête consiste à modifier la requête initiale de l'utilisateur par l'ajout de termes significatifs ou la réestimation de leurs poids. On distingue deux approches principales pour la reformulation de la requête :

- **L'expansion automatique de requête :** il s'agit d'etendre la requête initiale de l'utilisateur avec des mots du fond documentaire afin de construire une nouvelle requête qui definie encore mieux le besoin informationnel de l'utilisateur.
- **La réinjection de pertinence :** elle permet une reformulation de la requête initiale sur la base des jugements de pertinence de l'utilisateur. Ce dernier sélectionne les documents pertinents, et les non pertinents issus de sa requête initiale. Son jugement de pertinence est ensuite exploité par le SRI pour reformuler sa requête en modifiant le poids des termes qu'elle contient, et/ou en rajoutant de nouveaux termes utiles. La requête est dans ce cas là est étendu à partir des résultats de recherche obtenus avec

la requête initiale. Plusieurs formules sont utilisées, la plus populaire est celle de [Rachio, 1971] définie comme suit :

$$Q_N = \alpha.Q_0 + \beta.\frac{1}{|R|}\sum r \epsilon R^r - \gamma.\frac{1}{R'}\sum r' \epsilon R'^{r'}$$

- Q_N représente le vecteur de la nouvelle requête (requête reformulée)
- Q_0 est le vecteur de la requête initiale
- R est l'ensemble des vecteurs r des documents jugés pertinents par l'utilisateur
- R' est l'ensemble des vecteurs r' des documents jugés non pertinents par l'utilisateur
- α, β, γ sont des paramètres de reformulation.

1.5 Modèles de RI

Les modèles de RI visent à fournir un cadre théorique pour interpréter la notion de pertinence et permettent ainsi de classer les documents par rapport à un besoin d'information. Un modèle de recherche d'information est définit par un quadruplet [D, Q,F,R(q,d)] :

- D : est l'ensemble de documents.
- Q : est l'ensemble de requêtes.
- F : est le schéma de représentation des documents et des requêtes.
- R (q, d) : est la fonction de pertinence du document d à la requête q.

Il existe plusieurs modèles de recherche dont les modèles booléen, vectoreil, et probabiliste.

1.5.1 Le modèle booléen

Le modèle Booléen est un modèle qui se base sur la théorie des ensembles et l'algèbre de Boole [Salton et al 1968]. Les documents sont représentés par une conjonction des termes qui constituent leur contenu et les requêtes sont formulées à l'aide d'expressions logiques (AND, OR, NOT). Un document est jugé pertinent si et seulement si son contenu respecte la formule logique de la requête. Ce modèle vérifie si le document satisfait les conditions représentées par les termes de la requête. Il évalue si un document est pertinent ou non pertinent. Le score de chaque document sera ainsi représenté respectivement par 0 ou 1.

1.5.2 Le modèle vectoriel

Le modèle vectoriel représente un document D et une requête Q par des vecteurs d'un espace à n dimensions, les dimensions étant constituées par les termes du vocabulaire d'indexation. La pertinence d'un document est définie par des mesures de distance dans un espace vectoriel. La fonction de correspondance mesure la similarité entre le vecteur requête et les vecteurs documents. Parmi

les mesures les plus utilisées nous citons : la mesure du produit scalaire qui est la plus simple, La mesure de Jaccard, La mesure cosinus, et la mesure de Dice.

- La mesure de Jaccard :

$$RSV(q, d_j) = \frac{\sum_{i=1}^{n} w_{iq} * w_{ij}}{\sum_{i=1}^{n} {w^2}_{iq} + \sum_{i=1}^{n} {w^2}_{ij} - \sum_{i=1}^{n} \mathrm{w}_{iq} * w_{ij}}$$

- Le produit scalaire :

$$RSV(q, d_j) = \sum_{i=1}^{n} \mathrm{w}_{iq} * w_{ij}$$

- La mesure cosinus :

$$RSV(q, d_j) = \frac{\sum_{i=1}^{n} w_{iq} * w_{ij}}{\sqrt{\sum_{i=1}^{n} {w^2}_{iq} + \sum_{i=1}^{n} {w^2}_{ij}}}$$

- La distance de Dice :

$$RSV(q, d_j) = \frac{\sum_{i=1}^{n} w_{iq} * w_{ij}}{\sum_{i=1}^{n} {w^2}_{iq} + \sum_{i=1}^{n} {w^2}_{ij}}$$

1.5.3 Le modèle probabiliste

Dans les modèles probabilistes, La pertinence entre un document d et une requête q est mesurée par le rapport entre la probabilité que d soit pertinent pour q, notée p (R/d, q), et la probabilité qu'il soit non pertinent, notée par p ($\bar{R}$/d, q), où R est l'événement de pertinence et $\bar{R}$ de non-pertinence. Il existe plusieurs modèles probabilistes, mais aujourd'hui l'un des modèles les plus performants observés en RI est le modèle BM25. Le système Okapi est un exemple d'implémentation de ce modèle [Robertson et al 1996]

$$W_{ij} = \frac{(k_1 + 1) * tf_{ij}}{(k_1 * ((1 - b) + b * (\frac{dl}{avgl}) + tf_{ij}} * log(\frac{N - df_i + 0.5}{df_i + 0.5})$$

- dl : la taille du document di et Avgdl est la taille moyenne des documents
- tfij : le nombre d'occurrences du terme tj dans le document di
- k1 : un paramètre qui permet de contrôler la saturation de $tfij$
- b : un paramètre qui permet de contrôler la normalisation par rapport à la taille des documents
- N : le nombre de documents dans la collection
- dfi : le nombre de documents qui contiennent le terme tj

1.6 Evaluation des SRI

La première évaluation de SRI est effectué dans les années60 sur le projet Cranfield par Cleverdon [Cleverdon et al 1967].Afin de comparer plusieurs systèmes,Cleverdon a proposé une collection de test (CRANFIELD II), composée de 1400 articles scientifiques. Les auteurs des articles ont rédigé une requête résumant la problématique de leur article. Ils ont également évalué sur une échelle de 1 à 5 la pertinence par rapport à ces requêtes de tous les articles les référençant. Ceci apermis de collecter pour chaque requête l'ensemble des documents pertinents dans la collection.Cleverdon a ainsi posé les bases de l'évaluation utilisant des collections de test, sur une collection de documents fixe, le ou les systèmes à évaluer exécutent une re-quête, et les résultats sont comparés à la réponse idéale, c'est à dire à l'ensemble des documents pertinents pour la requête.

1.6.1 Mesures d'évaluation

Afin d'évaluer la pertinence du système en terme de documents retournés pour un besoin d'information spécifique. Plusieurs mesures d'évaluation standard peuvent être appliquées, telles que le Rappel, la Précision, la F-mesure, la précision@X, la précision moyenne, la R-précision, ainsi que la MAP.

— **Rappel :** le rappel mesure la capacité d'un système à sélectionner tous les documents pertinents de la collection. La valeur du rappel est comprise entre 0 et 1, plus le rappel est proche de 1, plus la réponse du SRI est pértinente. Le rappel se calcule par la formule suivante :

$$Rappel = \frac{(nombre\ de\ documents\ pertinents\ selectionnés)}{nombre\ total\ de\ documents\ pertinents}$$

— **Précision :** la précision est la capacité d'un système à ne sélectionner que des documents pertinents de la collection, c'est à dire rejeter tous les documents non pertinents. La valeur de la précision est comprise entre 0 et 1, elle est donnée par la formule suivante :

$$Precision = \frac{(nombre\ de\ documents\ pertinents\ selectionnés)}{nombre\ total\ de\ documents\ selectionnés}$$

— **La F-mesure :** elle permet de combiner la précision et le rappel en une seule et unique mesure, c'est une moyenne harmonique pondérée de la précision et du rappel. Elle se calcule par la formule suivante :

$$F - mesure = \frac{2 * (Précision * Rappel)}{Précision + Rappel}$$

— **la précision@X :** c'est la précision à différents niveaux de coupe de la liste. Cette précision mesure la proportion des documents pertinents retrouvés parmi les X premiers documents restitués par le système.

— **La précision moyenne :** c'est la moyenne des valeurs de précisions après chaque document pertinent, elle se calcule par la formule suivante :

$$AP_q = \frac{1}{R}\sum_{i=1}^{N} p(i) * R(i)$$

- R(i) = 1 si le i ème document restitué est pertinent, R(i) = 0 si le ième document restitué est non pertinent

- p(i) la précision à i documents restitués.
- R le nombre de documents pertinents pour la requête q
- N le nombre de documents restitué par le système

— **La R-precision :** c'est la précision en R pour une requête donné Q, où R est le nombre de documents pertinents pour Q. Autrement dit, s'il y a r documents pertinents parmi les premiers documents R récupérés, alors R-précision est r/R

— **La MAP (Mean Average Precision) :** c'est la moyenne des precisions moyennes.Cette mesure peut être qualifiée de globale puisqu'elle combine différents points de mesure, elle se calcule comme suit :

$$MAP = \frac{\sum_{q\epsilon Q} AP_q}{|Q|}$$

- APq est la précision moyenne d'une requête q.
- Q est l'ensemble des requêtes.
- |Q| est le nombre de requêtes.

1.6.2 Collections de test

D'une manière générale, pour évaluer un SRI à l'aide d'une collection de test, on procède de la façon suivante : le système exécute les requêtes une par une sur la collection de documents et renvoie pour chacune une liste ordonnée de documents qu'il considère comme potentiellement pertinents. Ces réponses sont ensuite comparé aux jugements de pertinence et des mesures d'efficacité sont calculées.Une collection de test est composée des éléments suivants [Sanderson et al, 2010][Voorhees et al 2002] :

– unecollection de documents, chaque document possédant un identifiant unique.

– un ensemble de besoins d'informations (topics) sur cette collection, un besoin étant généralement exprimé sous forme de descriptif textuel, et traduit en une suite de mots-clés formant ainsi la requête qui sera envoyée aux SRI.

– des jugements de pertinence, permettant d'indiquer quels documents sont pertinents pour chacun des besoins. Ces jugements de pertinence sontgénéralement binaires (un document est pertinent ou non) et produits manuellement. Les collections de test sont généralement construites dans le cadre de campagnes d'évaluation.

1.6.3 Compagnes d'évaluation

Les campagnes d'évaluation fonctionnent toutes sur le même principe : tous les ans,des tâches de recherche à évaluer sont définies, une collection de documents et de besoins est distribuée aux participants qui doivent renvoyer les résultats d'exécution correspondants. Une conférence est ensuite organisée pour que les participants confrontent leurs résultats et points de vue. De nombreuses compagnes sont apparues, parmi elles, on trouve la compagne TREC que nous décrivons dans ce qui suit.

TREC (Text REtrieval Conference)

La première campagne à avoir vu le jour [Voorhees et al, 2005] Mise en place en 1992 avec une tâche de recherche adhoc, elle propose aujourd'hui de très nombreuses tâches de recherche, parmi lesquelles on peut citer pour 2012 la tâche de recherche Web, la tâche de RI médicale, la tâche de recherche dans des microblogs,...

1.7 Conclusion

Dans ce premier chapitre nous avons essentiellement parler de la recherche d'information classique. Nous avons défini ce qu'est un systéme de recherche d'information, son processur de fonctionnement, ses modèles, ainsi que les différentes mesures d'évaluation.

Dans le prochain chapitre, nous allons nous intéresser à la recherche d'information sociale et plus particulièrement la recherche d'information dans les microblogs.

Chapitre 2

Recherche d'information sociale

2.1 Introduction

Dans ce chapitre, nous allons parler de la recherche d'information en générale, nous allons par la suite intégrer les réseaux sociaux, nous parlerons plus particuliérement de Twitter, son fonctionnement, caractéristiques et spécificités. Et enfin nous allons aborder la recherche d'information sur Twitter.

2.2 Recherche d'information sociale

La RI sociale est un domaine de recherche innovant qui a émergé au début des années 2000. Elle rassemble deux domaines de recherche, à savoir la recherche d'information et l'analyse des réseaux sociaux. [Kirsch et al, 2006] ont défini la recherche d'information sociale par la prise en compte des données des réseaux sociaux dans le processus de recherche d'information. Nous considérons en effet la RI sociale selon 3 axes :

1. La recherche d'information de nature sociale : Il s'agit de trouver les informations sociales qui répondent àl'utilisateur. On distingue par exemple la recherche d'information dans les blogs, microblogs, ou encore des réponses à des questions spécifiques auprès des amis, familles, collègues, ou même des personnes inconnues.
2. L'exploitation des contenus sociaux pour améliorer la RI : l'information sociale est utilisée afin d'améliorer le processus de RI, par exemple, les tags et les annotations sociales s'avérent utiles pour améliorer la recherche web et la recherche personnalisée, le reclassement des résultats de recherche, la reformulation (expansion)de requête,, etc.
3. La recherche collaborative : qui est une recherche effectuée par plusieurs personnes.

2.3 Réseaux sociaux

Les réseaux sociaux sont des éspaces dans lesquels les internautes interagissent ; publient, partagent, annotent, ou commentent, des ressources, des images ou encore des informations professionnelles. Les réseaux sociaux représentent aussi un moyen de communication et d'échange efficace en permettant aux utilisateurs de rentrer en contact avec d'autres pérsonnes. Au sens large, un réseau social désigne un ensemble d'entités sociales reliées entre elles par des liens créés lors d'interactions sociales, ces interactions se résument par :

• La recherche d'amis : cela permet de partager des outils avec eux tels que les photos et les messages.

• La recherche de professionnels : ce qui permet de rencontrer des partenaires potentiels, trouver un nouvel emploie, trouver des collaborateurs, annoncer des événements ou des activités professionnels.

La communication est un élément central des réseaux sociaux, leurs principe est de retrouver des personnes que vous connaissez, qui à leur tour, vous per-

mettes de rentrer en contacte avec d'autres personnes, ainsi, votre réseau peut très vite devenir considérable.

FIGURE 2.3.1 – Exemple d'un réseau social

D'un point de vue théorique, un réseau social est essentiellement un grand graphe où les nœuds représentent les utilisateurs et les arêtes représentent les relations entre les eux. Il est représenté par un graphe G= (V,E) où l'ensemble des nœuds V représente les utilisateurs et l'ensemble des arêtes E=V×V représente les relations entre eux. Dans le cas d'un réseau social non-orienté, une arête (vi,vj) représente une relation symétrique, associant des utilisateurs **vi** et **vj**. L'amitié est un exemple typique de relation non-orientée dans le réseau social. Dans le cas d'un réseau social orienté, une arête (vi,vj) représente une relation orientée de vi à **vj**. Par exemple, la communication électronique est représentée par une bordure directe (vi,vj) où **vi** et **vj** représente l'expéditeur et le destinataire, respectivement. Pour indiquer l'importance d'un utilisateur dans le réseau ou les points forts d'une relation sociale, les poids du réseau sont associés à des nœuds et des arêtes, respectivement.

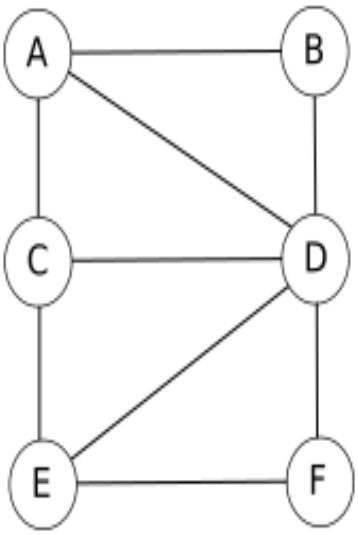


FIGURE 2.3.2 – Représentation graphique d'un réseau social

2.3.1 Exemples de réseaux sociaux

Nous citons ci-dessous quelques exemples de réseaux sociaux :

FIGURE 2.3.3 – Réseau social

— **Facebook :** Créé en 2004 par Mark Zuckerberg à l'université Harvard. C'est le réseau social le plus populaire au monde avec 2,38 milliards d'utilisateurs actifs chaque mois et 1,56 milliard d'utilisateurs actifs chaque jour. Facebook est un réseau social qui permet à ses utilisateurs de publier des images, des photos, des vidéos, des fichiers et documents, d'échanger des messages, joindre et créer des groupes et d'utiliser une variété d'applications.
— **Google plus :** Créé en 2011 par l'équipe de Google et lancé à la fin de cette année. Produit proposé par le géant du web, Google+ est assez séduisant : une interface agréable et rapide, et des fonctionnalités inspirées des meilleures des concurrents du web social. Les utilisateurs de Google+ peuvent partager des informations avec un ou plusieurs de leurs cercles, faire un tchat avec leurs contacts, et suivre éventuellement des actualités par centres d'intérêts.

— **LinkedIn :** Créé en mai 2003 par Reid Hoffman et Allen BlueLinkedIn est un réseau social à utiliser dans un contexte d'affaires. Les pages des utilisateurs exposent leurs carrières professionnelles et leur permettent de préciser leurs intérêts en matière de débouchés professionnels, d'emplois, et cela en partageant des liens, textes, vidéos, etc.
— **Pinterest :** Créé en 2010 par Paul Sciarra, Evan Sharp et Ben Silbermann. Le nom du site est un mot composé des mots anglais "pin" et "interest", signifiant respectivement "épingler" et "intérêt".Pinterest est un site web américain permettant d'épingler des images, de les partager, et de créer des albums. C'est un réseau social, un peu comme Facebook et Google+, mais qui est principalement orienté sur les images et leur partage.

L'un des réseaux sociaux les plus actifs et populaires de nos jours est Twitter. Dans notre travaille, on s'interessera particuliérement à ce réseau sociale.

2.4 Twitter

Twitter est créé en mars 2006 et lancé en juillet de la même année par Jack Dorsey à san Francisco. C'est une plateforme de microblogging qui permet à un utilisateur, d'envoyer gratuitement de bref messages appelés Gazouillis « tweets ». ces messages sont limités à 140 caractères. Au deuxième trimestre 2019, Twitter comptait 139 millions d'utilisateurs quotidien actifs dans le monde. Récement Le réseau social Twitter annonce le doublement de la taille maximale des messages de 140 à 280 caractères afin notamment d'inciter les utilisateurs à écrire plus.

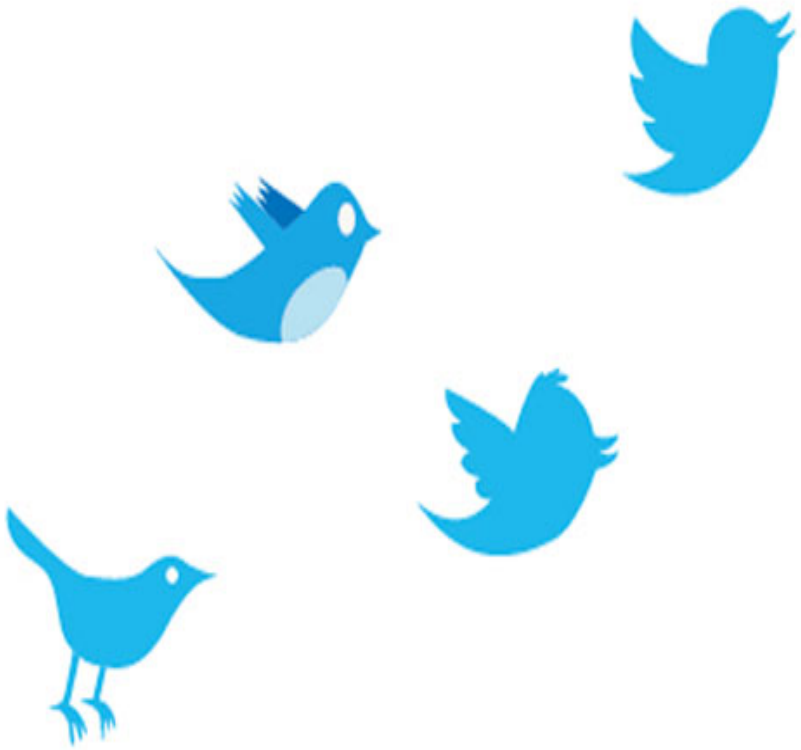


FIGURE 2.4.1 – Logos de twitter

2.4.1 Fonctionnement de Twitter

Pour ouvrir un compte Twitter, il suffit de s'inscrire sur le site officiel « http ://twitter.com/ ». Une fois le compte créé et validé, vous accédez à votre profil sur une interface d'accueil comme le montre la figure qui suit.

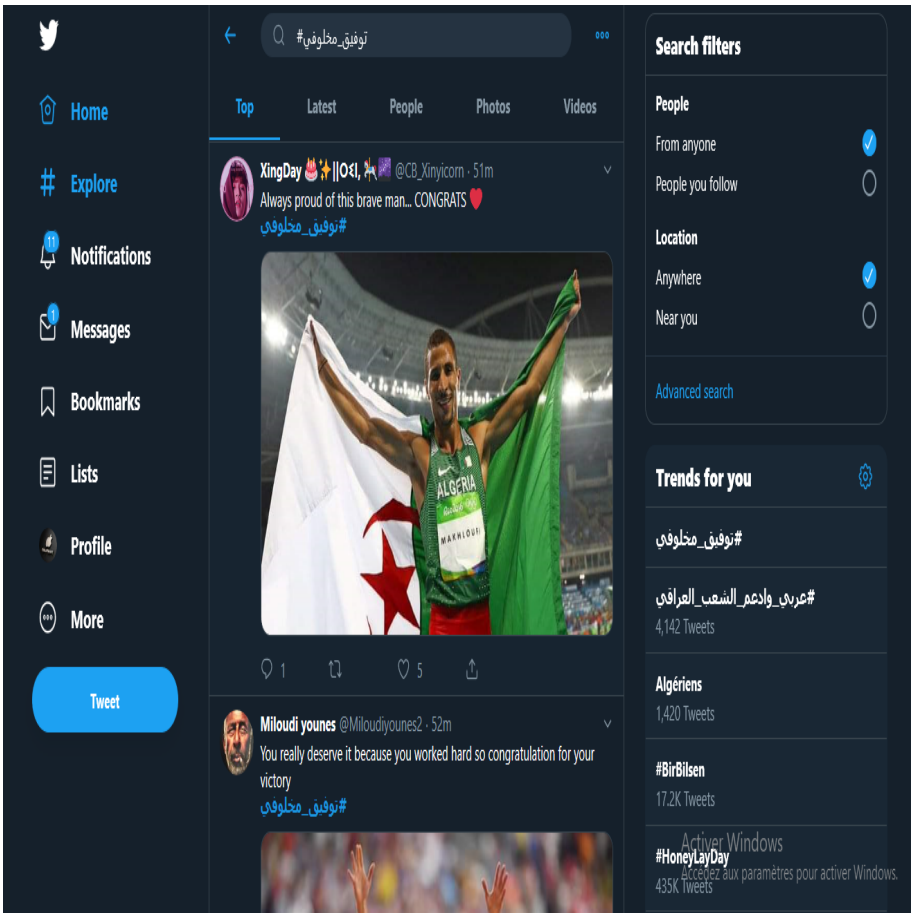


FIGURE 2.4.2 – Interface Twitter

2.4.2 Vocabulaire de Twitter

Entre tweets, twittos, retweets, et autre mots spécifiques à Twitter. Voici un petit récapitulatif des principaux mots et signes utilisés dans Twitter :

- **Tweet :** Littéralement « gazouillis ». Les tweets sont les messages postés sur Twitter. Ils sont limités à 140 caractères.
- **Twitto :** est un utilisateur de Twitter. Si vous avez un compte, vous faites donc partie des Twittos.
- **Retweeter :** Action qui consiste à rediffuser le message d'un autre utilisateur à vos abonnés. Un retweet (RT) est donc un message rediffusé.
- **Timeline :** Il s'agit du flux d'actualités de Twitter. La timeline générale présente l'ensemble des tweets postés par vos abonnements, et votre timeline personnelle affiche les différents tweets que vous avez mis en ligne.
- **Follower :** Un follower est une personne qui s'est abonnée à vos publi-

cations. Elle verra donc apparaître vos tweets dans sa timeline. On dit qu'elle vous suit « vous follow ». Un résumé du profil de vos followers apparaît quand vous cliquez sur votre nombre d'abonnés.

• **Following :** Ce terme désigne les comptes twitters que vous suivez vos « abonnements ». Une icône bleu écrit « abonné » est indiquée lorsque vous consultez le compte d'un twitto auquel vous êtes abonné.

• **Mention(@) :** Comme son nom l'indique, elle permet de mentionner quelqu'un dans un tweet. La mention s'exprime par le symbole @ accolé à un pseudo. Ces messages sont publics et pourront être lus par l'ensemble de vos followers. Leur destinataire pourra quant à eux les repérer aisément dans l'interface « @ Connecter » de Twitter.

• **Hashtag(#) :** Il s'agit d'un mot-clé qui permet de catégoriser votre tweet et de faciliter la recherche des utilisateurs. Un hashtag se constitue du symbole dièse # suivi d'un mot, comme #referencement par exemple. Les hashtag permettent à un twitto de repérer facilement des comptes à suivre, ou des tweets susceptibles de l'intéresser.

• **MP :** Abréviation de « Message Privé », pour « Direct Message » en anglais. Cette fonction permet d'envoyer un message privé à un utilisateur. Les MP sont eux-aussi limités à 140 caractères mais ils n'apparaissent pas dans les timeline : ils arrivent sur une messagerie interne à Twitter. On ne peut envoyer un MP à une personne que lorsqu'on la suit sur Twitter, et elle ne peut nous répondre que si elle nous suit également.

• **Réponse :** Une réponse est une réaction à un Tweet d'une autre personne. Vous pouvez publier une réponse en cliquant ou en appuyant sur l'icône Répondre depuis un Tweet.

• **Tendances :** Les tendances désignent en quelque sorte les sujets à la mode sur Twitter. Elles sont personnalisées en fonction de votre localisation et de vos abonnements.

2.4.3 Spécificités et avantages

Twitter permet aux utilisateurs d'etre sur la pointe de l'actualitée. Le concept initial de Twitter est fondé sur une idée simple : permettre à ses utilisateurs de dire ce qu'ils font en temps réel, d'où son premier slogan : What are you doing ? (Qu'est-ce que vous faites). De nombreuses personnes l'utilise afin de se tenir informés de façon immediate des événements récents. A la différence des autre réseaux sociaux, ou les membres vérifient la plupart du temps qu'ils se connaissent et sont plus hésitants à l'idée d'ajouter quelqu'un qu'ils ne connaissent pas, le premier intérêt dans twitter est de voir ce que l'autre a à partager et de partager des informations intéressantes en retour. Les connexions se font donc rapidement partout dans le monde.

2.5 La recherche d'information dans Twitter

La recherche des tweets est une tâche de recherche d'information ad hoc, qui consiste à répondre à une requête via un ensemble de tweets, et sélectionner ceux qui sont pertinents. Selon une étude menée par [Teevan et al, 2011] sur 54 utilisateurs de Twitter, dans le but d'étudier leurs motivations, pour chercher des informations dans les microblogs, ils ont constaté que les utilisateurs utilisent Twitter pour avoir :

— Des informations récentes : sur les actualités, les sujets tendance, les évènements récents. . . etc.
— Des informations sociales : en cherchant à établir des relation sociales
— Des informations sur des sujets en particuliers.

2.5.1 Etudes de facteurs de pertinences

Dans cette section, nous présentons les différents facteurs de pertinence à prendre en compte dans la conception des approches de recherche de microblogs [Damak, 2014] :

— **Facteurs de pertinence liés au contenu :** consiste à étudier les quatre facteurs relatifs au contenu à savoir, la popularité du tweet, la longueur du tweet, la correspondance exacte des termes entre les tweets et la requête, et la qualité du langage d'écriture du tweet.
  —- **Popularité du tweet :** ce facteur de pertinence estime qu'un tweet est populaire si seulement si on trouve plusieurs autres tweets ayants un contenu similaire.
  —- **Longueur du tweet : :** ce facteur compte le nombre de termes d'un tweet, plus un tweet est long plus il est informatif.
  —- **La correspondance exacte des termes :** consiste à calculer le nombre de termes en commun entre un tweet et une requête.
— **Facteurs de pertinence basés sur l'hyper textualité :** ce sont des facteurs liés aux URLs, on distingue trois facteurs qui ont été employés pour indiquer la qualité de l'information publiée dans les tweets
  —- **Présence de l'URL dans les tweet** : les microblogueurs partagent également des URLs dans leurs statuts pour attirer l'attention de leurs amis sur un contenu présent sur le web, ainsi la présence d'un URL indique que le tweet à un caractère informatif, la valeur retournée par ce facteur est en binaire : 1 si le tweet contient un URL, 0 sinon.
  — **Fréquence des URLs :** compte le nombre d'URLs publiés dans un tweet.
  — **Fréquence de l'URL dans le corpus :** il calcule le nombre de fois où l'URL apparait dans le corpus.
— **Facteurs de pertinence relatifs à la qualité des tweets :** Ce facteur prend en considération les critères spécifiques liés auw tweets :
  — **Retweet :** le principe consiste à étudier si les tweets ont été retweet, autrement dit précédés par RT. La valeur retournée par ce facteur est binaire : 1 si le tweet contient RT, 0 sinon.

— **Fraîcheur :** c'est la différence entre la date de la publication du tweet et celle de soumission de la requête.

2.5.2 Evaluation de la RI dans les microblogs

Dans cette section, nous allons aborder l'évaluation de la RI selon TREC microblog, ainsi que les différentes mesures d'évaluation

2.5.2.1 TREC microblog

Créé en 2011, Trec microblog est une tache de la compagne TREC qui est consacrée à la RI dans les microblogs, décrite également comme étant une tache ad hoc temps réel. dans le cadre de TREC2011, Twitter fourni des identificateurs pour environs 16 milions de tweets récupérés en 2011.

2.5.2.2 Mesures d'évaluation

— **La précision p@30 :** c'est la mesure officielle pour l'évaluation de la tâche de recherche en temps réel dans TREC microblog 2011. Cette mesure évalue la capacité d'un système à retourner les tweets pertinents, parmi les 30 premiers de la liste des résultats.
— **La précision moyenne MAP :** elle est utilisée comme une mesure supplémentaire pour évaluer l'efficacité de recherche, tout en tenant compte de la précision, du rappel et du rang des documents.

2.6 Conclusion

Dans ce chapitre nous avons introduit la recherche d'information sociale, nous avons par la suite passer en revu les différents réseaux sociaux dont Twitter que nous avons détaillé par la suite. Et pour terminer nous avons parler de la recherche d'information dans Twitter. Dans ce qui suit, nous allons établir un état de l'art des différents traveaux de recherche dans le cadre d'intégration du profil utilisateur et du temps dans Twitter.

[Auteur]« Ma citation préférée. »" Titre de la partie 2

Chapitre 3

Etat de L'art sur le profil utilisateur

3.1 Introduction

Dans ce chapitre nous nous interessons à l'integration du profil utilisateur dans le temps sur la plateforme de microblogging Twitter, nous commençons par définir le profil utilisateur ainsi que les differentes manières de l'integrer dans un modèle de recherche. Nous citerons également les travaux éxploitant ce dernier dans la RI. Nous poursuiverons par la suite avec la RI temporelle, son objectif, et les approches intégrants le temps dans les microblogs.

3.2 Le profil utilisateur

Le profil utilisateur dans le contexte des systèmes de personnalisation d'informations, peut être défini comme une structure qui permet de modéliser et stocker des informations relatives à l'utilisateur [Brusilovsky, 1996]. Le profil utilisateur peut contenir ses données personnelles qui sont relativement stables dans le temps et ne demandent pas a priori de mise à jour automatique (son identité, ses données démographiques, ses données professionnelles), son historique, ses annotations associées aux documents(les tags de pages ou de personnes), ainsi que ses les préférences et intérêts qui tendent à changer au fil du temps. Différentes autres informations peuvent être exploitées au sein d'un microblog pour construire le profil de l'utilisateur telles que les traces de l'utilisateur, les tweets, les annotations et les relations sociales. De ce fait, on constate que le profil utilisateur peut etre modélisé et exploité selon differentes manieres. Nous pésentons essentiellement dans cette section les travaux intégrant le profil utilisateur comme vecteur de poids de termes dans la recherche d'information, en permettant à un systéme de s'adapter à l'utilisateur, et comme facteur de pertinence dans le reclassement des resultats de recherche.

3.2.1 Intégration du Profil utilisateur dans la RI

Avec l'émergence des réseaux sociaux et la disponibilité de différentes informations sociales, les travaux de RI ont commencé à s'intéresser d'avantage aux utilisateurs initiateurs des requêtes dans l'objectif de mieux identifier leurs besoins d'information et d'améliorer ainsi les résultats retournés, notamment en les personnalisant. Il est souvent difficile, voire impossible d'interpréter de manière précise le besoin d'information représenté par une requête formulée par l'utilisateur sur un système de RI. De plus plusieurs utilisateurs peuvent formuler la même requête sous la forme d'une liste de quelques mots clés mais en ayant chacun des besoins d'information différents en fonction de leurs centres d'intérêts. Si on suppose que deux utilisateurs Alice et Bob formule tous les deux la même requête composée de deux termes Smartphone et Android. Sachant qu'Alice est intéréssée par les smartphones android, et Bob par la systèmes d'exploitation android, un système de RI non classique devrait estimer qu'un document **d** pertinent pour Alice sera aussi pertinent pour Bob sans tenir compte du fait que le profil d'Alice montre qu'elle est plus intéressée par les Smartphones que par An-

droid. Ainsi, son besoin d'information est probablement centré autour du mot clé smartphone avec une ouverture sur Android. Le terme de la requête smartphone devrait donc être considéré de manière prioritaire pour Alice contrairement à Bob. L'objectif d'exploiter un profil utilisateur dans un système de recherche est de considérer la pertinence d'un document pour un utilisateur selon ses centre d'interets et ses besoin en information indépendament des autres utilisateurs.

Dans ce contexte, plusieurs approches tentent de modéliser le profil utilisateur par des vecteurs de centres d'intérêt basés sur les informations sociales de ce dernier [Xie et al., 2012].

$$\overrightarrow{prof} : (t_1 : w_{u,t1}; t_2 : w_{u,t2}; .....t_n : w_{u,tn})$$

- t_i : est un terme dans le contexte sociale de l'utilisateur : annotations, relations sociales....etc

- $w_{u,t}$: est le poid de $t_i$dans le profil sociale de l'utilisateur **U**. Il peut être calculé par une simple fonction de pondération à base de fréquences d'occurrence comme suit :

$$w_{u,t} = tf_{u,t}$$

-tf_{ut} : est le nombre de fois où l'utilisateur **U** a employé **t** dans son contexte sociale exploité.

Une fois le PU modélisé, se pose la question de son intégration dans le modèle de RI, particulièrement le niveau où il peut être impacté. Dans cette section, nous présentons des travaux de l'état de l'art consacrés à l'intégration du PU à l'indexation des documents, ainsi qu'à l'expanssion de requête.

3.2.1.1 Indexation sociale

L'intégration du profil utilisateur au niveau de l'indexation revient à considérer que cette dernière est personnalisée pour chaque utilisateur. [Bouadjenek et al, 2013] proposent de calculer un score de similarité entre la requête et l'indexation sociale personnalisée du document pour un utilisateur donné.[Bouhini et al, 2013] quand à eux proposent d'extraire et de repondérer les termes du document qui se trouve aussi dans le profil de l'utilisateur.

3.2.1.2 Reformulation et expansion de requête

Dans la plupart des approches d'expansion de requêtes utilisant le profil utilisateur, il est question d'identifier les meilleurs termes candidats à l'expansion de la requête de ce dernier. Dans [Xie et al, 2012], la requête initiale formulée par un utilisateur **U** est représentée par un ensemble de termes pondérés, comme suit :

$$q_u = (t_1^{qu} : w_{qu,t1}; t_2^{qu} : w_{qu,t2}; .....t_n^{qu} : w_{qu,tn})$$

- t :représente un terme **t** dans la requête **qu** de l'utilisateur **U**
- $W_{qu,t}$: est le poids du terme **t** dans la requête **qu** de l'utilisateur **U**

La requête étendue composée des termes initiaux de la requête, combinés aux termes en provenance du profil social de l'utilisateur, est donnée par la formule suivante :

$$q'_u = (t_1^{q'} : w'_{qu,t1}; t_2^{q'} : w'_{qu,t2}; .....t_n^{q'} : w'_{qu,tn})$$

- $w'_{qu,t}$: est le poids d'un terme dans la requête étendue par les termes du profil utilisateur. IL est donné par une combinaison linéaire des termes initiaux de la requête de l'utilisateur et les termes du profil social de ce dernier :

$$w'_{qu,t} = \alpha w_{qu,t} + (1 - \alpha) w_{u,t}$$

- $w_{u,t}$: est le poids du terme t dans le profil de l'utilisateur
-α : coéficient d'amortissement.

3.2.2 Intégration du profil utilisateur dans le reclassement des résultats

La prise en compte du profil utilisateur au niveau des fonctions de correspondance et calcul de score a permis une amélioration significative des résultats de la recherche par un reclassement des documents retournés [Cai and Li, 2010] .Dans une première catégorie de combinaison de scores, les approches proposées se basent sur une combinaison du score thématique et d'un score social. Le score social calculé suivant différents modèles et fonctions de pondération.

3.3 Approches basées sur le Profil Utilisateur

3.3.1 Approche de Kacem

Au fur et à mesure que les intérêts des utilisateurs évoluent, [Kacem et al, 2016] propose de créer un profil utilisateur évolutif et sensible au facteur temps En partant du principe que les anciens termes fréquents ne doivent pas surpasser les termes récents et non fréquent, ils proposent de combiner le profil à court termes autrement dit : les tweets publiés par un utilisateur le jour de soumission de la requete, et le profil à long térme à savoir : les tweets publiés par un utilisateur depuis la creation de son compte jusqu'au jour de la soumission de la requete en excluant le profil à court terme. Ils proposent d'extraire les termes des documents et de calculer leurs poids en combinant à la fois leur fréquence et leur fraicheur, et leur fréquence de terme normalisée (nTF), comme suit :

$$nTF(ti)^{si} = \frac{Freq^{si}(ti)}{\sum_{k \epsilon D^{si}} Freq^{si}(ti)}$$

- Freq$^{si}(ti)$: est la fréquence relative d'un terme **ti** dans D^{si}.

- $\sum_{k \epsilon D^{si}} Freq^{si}(ti)$: représente la somme des fréquences de tous les termes apparus dans D

Pour mesurer la fraîcheur, ils passent en revue la notion de fréquence de terme en l'ajustant avec une fonction biaisée dans le tempsen se basant sur le fait qu'un terme frequent est le terme proche de la date actuelle. ils utilisent la fonction de Kernel Gaussian comme fonction temporelle :

$$K(S^c, S_j) = \frac{1}{\sqrt{2\prod}\sigma}.exp[\frac{-(S^c - S_j)^2}{2\sigma^2}]$$

- : est le coefficient d'interpolation
- Sc : est la date du jour .
- Sj : est une date antérieure.

Ils proposent la construction d'un profil évolutif, à chaque date Sj la construction d'un vecteur de poids de termes modélisant le profil de l'utilisateur est donné comme suit :

$$\overrightarrow{U} = (t_i^{sj}:w_1^{sj}, t_2^{sj}:w_2^{sj}, \ldots\ldots\ldots\ldots\ldots t_n^{sj}:w_n^{sj})$$

- t : est un terme dans le document
- w_{tk}^{sc} : est le poids temporel d'un terme **t** dans le profil, calculé par le produit de sa fréquence relative temporelle et sa fréquence normalisée comme suit :

$$w(t_k)^{sc} = \sum nTf(t_k)^{sj}.\ K(S^c, S_j)$$

3.3.2 Approche de Bouhini

[Bouhini et al 2014] choisissent l'intégration du contexte informationnel social de l'utilisateur, modélisé à partir de ses annotations sociales et son voisinage au sein du document afin de personaliser l'indexation, au niveau de la requête ainsi qu'à l'interrogation. Afin d'améliorer les résultats retournés par le système pour une requête de l'utilisateur en renvoyant les documents pertinents pour chaque utilisateur selon son contexte informationnel social propose [Bouhini et al 2014] la repondération des termes importants pour l'utilisateur dans la requête. En choisissant un modèle de pondération BM25, qui est adapté au traitement de documents et de requêtes de tailles très variables, grâce à des versions normalisées des fréquences dans le TF(Poids du terme au sein du document), l'IDF(Pouvoir discriminant du terme) et le QTF(Poids du terme au sein de la requête). Le score d'un document pour une requête calculé dans le modèle **BM25** est donné par la formule suivante :

$$BM25(q, d) = \sum_{t \epsilon dq} w_{d,t} * w_{q,t}$$

$$BM25(q,d) = \sum_{t \epsilon dq} \frac{(k1+1)tf_{dt}}{k1(1-b+b*\frac{dl}{Avgl})+tf_{dt}} * log \frac{N-df_t+0.5}{N-df_t} * \frac{(k3+1)tf_{qt}}{k3+tf_{qt}} .....(3.3.3)$$

- W(d,t) : est le poids du terme **t** dans le document **d**
- W(q,t) : est le poids du terme **t** dans la requête **q**

Ce qui les a amener à proposer deux approches d'intégration du contexte social de l'utilisateur au sein d'un modèle de RI, selon qu'il est combiné au document (figure 3.3.1) ou à la requête (figure 3.3.2), et à définir plusieurs modèles de recherche sociale pérsonalisée d'information.

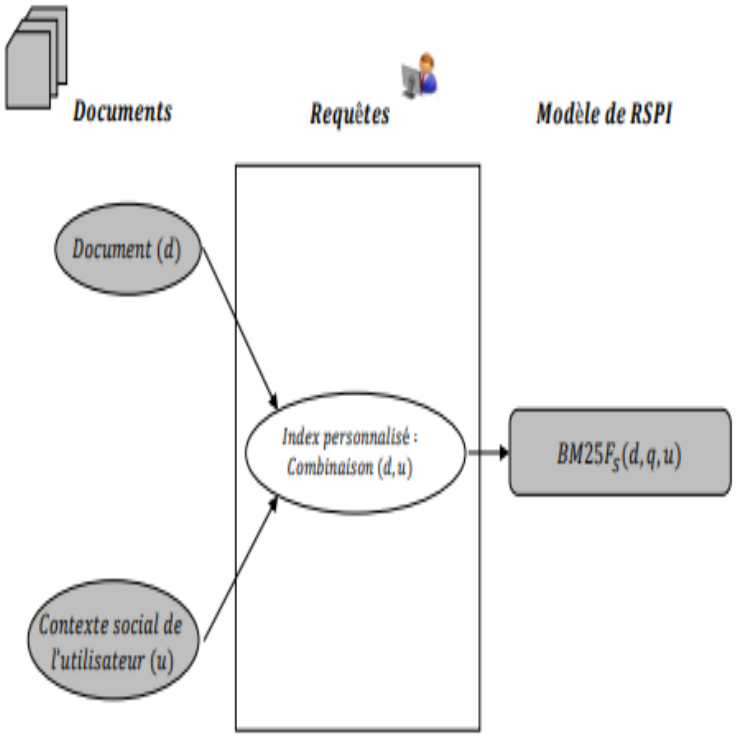


FIGURE 3.3.1 – Personnalisation de l'indexation [Bouhini et al, 2014]

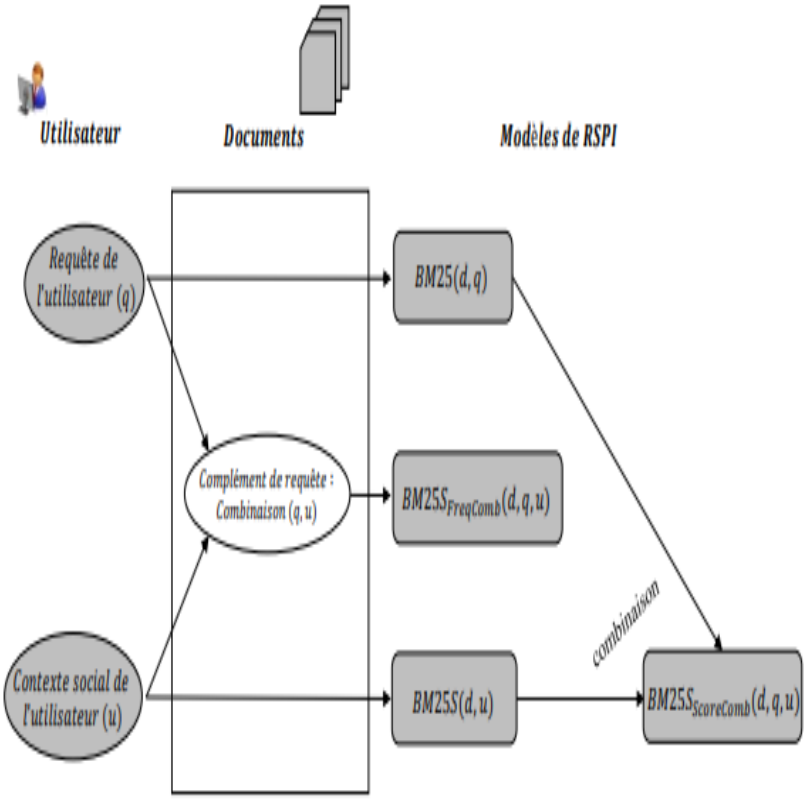


FIGURE 3.3.2 – Intégration du profil utilisateur à l'interrogation [Bouhini et al , 2014]

Ils proposent trois modèles de recherche sociale personnalisée d'information, notés respectivement **BM25S**, **BM25SFreqComb** et **BM25SScoreComb**, basés sur le modèle de pondération **BM25**, et tiennent compte des trois façons de contrôler la saturation de la requête via le paramètre **k3**.

— **BM25S(d, u)** : dans ce modèle, le profil utilisateur peut être utilisé pour remplacer sa requête. Ce modèle retourne une liste classée de documents pertinents pour un utilisateur **U** en considérant son contexte informationnel social.

$$BM25(d,u) = \sum_{d,cis_u} w_{u,t} * w_{d,t}$$

$$w_{dt} = TF_{dt} * IDf_t$$

$$w_{ut} = \frac{(k3+1) * [w_u * tf_{ut} + w_v * tf_{vt}]}{k + [w_u * tf_{ut} + w_v * tf_{vt}}$$

– TF(d,t) et IDFt : représentent respectivement **TF(d,t)** et **IDFt** classiques dans la formule **BM25** donnée dans l'équation (3.3.2)

– W(u,t) : est le poids d'un terme **t** dans le contexte de l'utilisateur **U**.

- Wu et Wv : représentent respectivement deux paramètres déterminés expérimentalement et correspondant au poids du profil de l'utilisateur et au poids du profil du voisinage de l'utilisateur.

En tenant compte de la saturation, **BM25S** se décline en trois variantes :

1. **BM25Sbin(d, u)** : pour un k3 = 0
2. **BM25Stf (d, u) :** pour un k3 = 1000
3. **BM25Sw(d, u) :** pour un k3 optimisé

— **BM25SF reqComb(d, q, u)** : ce modèle retourne une liste classée de documents pertinents pour un utilisateur **U** en considérant sa requête combinée à son profil au niveau des fréquences d'occurrence des termes.

$$BM25SFreqComb(d,q,u) = \sum_{t \epsilon dq} TF_{dt} * IDF_t * QTF_{Squt}$$

$$QTF_{Squt} = \frac{(k3+1) * [tf_{qt} + w_u * tf_{ut} + w_v * tf_{vt}]}{k + [tf_{qt} + w_u * tf_{ut} + w_v * tf_{vt}}$$

– TF(d,t) et IDFt : représentent respectivement **TF(d,t)** et **IDFt** classiques dans la formule **BM25** donnée dans l'équation (3.3.2). En tenant compte des 3 nivaux de saturation de la meme façon que dans le modèle **BM25S(d, u).**

— **BM25SScoreComb(d, q, u) :** ce modèle retourne une liste classée de documents pertinents pour un utilisateur **U** en combinant la requête à son contexte.

$$BM25SScoreComb_{bin}(d,q,u) = RSV(q,d) + w_u * BM25S_{bin}(d,u)$$

$$BM25SScoreComb_{tf}(d,q,u) = RSV(q,d) + w_u * BM25S_{tf}(d,u)$$

$$BM25SScoreComb_{w}(d,q,u) = RSV(q,d) + w_u * BM25S_{w}(d,u)$$

3.3.3 Approche de Masaki Aono

Une approche (Masaki Aono) propose de reclasser les résultats obtenus lors d'une recherche effectuée avec un modèle de RI classique en utilisant différents critères. Cette approche consiste à extraire des critères specifiques du Tweet, puis les combiner au score thématique , puis classer ces documents selon ce score. Afin de reclasser les tweets retournés lors de la recherche avec un modèle classique, il a proposé un modèle linéaire qui combine les valeurs des différents critères de pertinence de chaque tweet. Pour une requête Q et un document T, le score de pertinence SCORE(Q,T) est estimée par la formule :

$$Score(Q, D) = RSV(Q, T) + \sum_{i=1}^{N} fi(Q, T)$$

- N est le nombre de facteurs de pertinence
- RSV (Q,T) la pertinence thématique.

Les facteurs de pertinence utilisés :

— Le nombre de retweet : un tweet à fort caractère informatif est retweeté par plusieurs autres utilisateurs. En calculant une valeur indiquant le nombre de fois où un tweet est retweeté, ainsi pour mesurer la popularité d'un tweet.

— Les nombre de followers : En calculant le nombre de followers que l'auteur d'un tweet a, pour mesurer la crédibilité d'un utilisateur.

— Le nombre de publication d'un utilisateur : Le paramètre status count d'un utilisateur, indique le nombre de tweets qu'un utilisateur a publier.

3.4 La RI temporelle

Le temps prend de plus en plus d'intéret dans le domaine de recherche d'information dans les microblog.En partant du principe quel'information la plus fraiche possible est l'information la plus pertinente, le temps est généralement représenté par la date de soumission d'une requete, la date de création d'un document, ou par les expression temporelles contenus dans les documents.

La recherche d'information temporelle a pour objectif d'améliorer les modèles de RI classique en combinant la pertinence thématique avec un score basé sur la temporalité afin de retourner les documents les plus fraic pour un utilisateur.

3.5 Approches basées sur la temporalité

3.5.1 Approches de Damak

3.5.1.1 Approche I

[Damak, 2014] propose dans un premier temps d'introduire la fraicheur dans la mesure de pertinence. Ils proposent de renforcer le score thématique par un

score temporelle calculé entre la date de soumission de la requete et la date de publication d'un tweet. Ils favorisent la fraicheur à la pertinence, du fait qu'un tweet pertinent mais pas recent est moins important qu'un tweet recent et peut pertinent. Le score de chaque tweet est donné par :

$$RSVT_1(d, q, \sigma) = RSV(d, q) * k\sigma(t_q, t_d)$$

$k\sigma(t_q, t_d)$: est le score de Kernel Gaussien calculé par :

$$k\sigma(t_q, t_d) = \frac{1}{\sqrt{2\prod}\sigma}.exp[\frac{-(t_q - t_d)^2}{2\sigma^2}]$$

- tq : la date en jour de soumission de la requete
- td : la date en jour de publication du document
- σ : valeure d'emplification(en l'augmentant le score ce raproche du RSV thématique).

3.5.1.2 Approche II

[Damak et al, 2014] propose dans sa deuxiéme approche de favoriser les termes fréquement utilisés au moment de la soumission de la requete. Le score finale de son approche est donné comme suit :

$$RSVT_2(d, q, \sigma) = IDF * k\sigma(t_q, t_d)$$

$$IDF = log\frac{N - Ri_{time}}{Ri_{time}}$$

$$Ri_{time} = \sum |Ri|_t * k\sigma(t_q, t_d)$$

- t : période temporelle exprimée en jours
- $|Ri|_t$: nombre de documents dans cette période

3.5.1.3 Approche III

[Damak, 2014] propose dans sa troisième approche de calculer le score d'un terme dans un tweet, en fonction de la fréquence d'emploi de ce terme dans une période donnée. un meme terme aura des scores différents selon la date de publication du document auquel il apparait. Le score sera plus élevé si le terme appartein à un document publié dans une période ou le ce terme est fréquement utilisé. Le score finale est donné par :

$$RSVT_3(d, q) = IDF_{new}$$

$$IDF_{new} = IDF * 1/IDF_{local}$$

$$IDF_{local} = log\frac{N - Ri_{time}}{Ri_{time}}$$

- Ri_{time} : le nombre de tweets contenant le terme t le jour de publication d'un tweet

- IDF_{local} : l'idf d'un terme sur une période d'une journée. IDF est plus important dans une journée ou le terme est peut utilisé, de ce fait l'auteur utilise l'inverse du facteur : IDF_{new}

3.5.2 Approche de Masaki Aono

[Masaki et al] proposent d'exploiter le temps comme facteur de mesurant la proximité temporelle entre da date de publication du tweet et la date de soumission de la requete, comme suit :

$$TimeScore(Q,T) = \frac{1}{\sqrt{Query_{Time(Q)} - Tweet_{Time(T)} + 1}}$$

3.5.3 Approche de Massoudi

[Massoudi, 2011] propose une approche permettant de classer les résultats du plus encien vers le plus récent, tout en favorisant la fraicheur qui est mesurée par la difference temporelle entre la date de soumission de la requete et la date de publication du document. Dans un premier temps, l'auteur propose une approche d'expanssion de requete, dans laquelle il supose que tous les documents contenant au moins un terme de la requete sont considérés comme termes condidats à l'expanssion de la requete. Il calcule un score permettant d'évaluer l'importance d'un terme t pour une requete q comme suit :

$$score(t|q) = n_{cooccur}(t,q) * log(\frac{|N|}{n_{tweet}(t,N)})$$

- N : la collection totale des documents

-$n_{cooccur}(t,q)$: nombre de documents dans lesquels le terme t correspond avec au moins un des termes de la requete q

- $n_{tweet}(t,N)$: nombre de document de N qui contiennet le terme t.

La requete étendu q' peut alors etre formulée comme suit :

$$P(t|q') = \frac{|score(t|q)}{\sum_{t' \epsilon k} score(t'|q)})$$

- k : represente les meilleurs termes pour l'expanssion de la requete.

Par la suite, il intégre le notion du temps et propose un autre moèle pour l'expansion de requete comme suit :

$$score(t|q,c) = log(\frac{|N|}{n_{tweet}(t,Nc)}) * \sum_{d \epsilon cooccur(t,q,c)} e^{-\beta(c-c_d)}$$

- c : le moment ou la requete de l'utilisateur a été exécuté

- c_d : le moment de publication du document

- Nc : la collection totale des documents qui ont été affichés avant c

- $n_{tweet}(t, Nc)$: le nombre de documents dans la collection N qui contiennent le terme t

- cooccur(t,q,c) : le sous ensemble de documents dans Nc, dans lequel le terme t correspond à au moins un des termes de la requete

-β : controle la contribution de chaque document au score de terme t en fonction de la date de publication

La requete étendu q' basée sur le temps peut etre formulée comme suit :

$$P(t|q') = \frac{|score(t|q,c)}{\sum_{t'' \epsilon k} score(t'|q,c)})$$

3.6 Conclusion

Dans ce chapitre, nous avons passé en revue la notion du profil utilisateur et la temporalité dans la recherche d'information dans Twitter. Le chapitre suivant portera sur l'approche que nous proposerons.

Chapitre 4

Approche proposée

4.1 Introduction

Dans le chapitre précédent, nous avons passé en revue de nombreuses approches intégrant le profil utilisateur et le temps dans Twitter pour répondre au mieux aux besoins utilisateur. Dans ce chapitre nous présentons nos deux approches, la premiére tente d'intégrer le profil utilisateur et le temps tandis que la deuxième intégre uniquement le profil utilisateur.

4.2 Approches proposées

4.2.1 Approche I

Dans cette approche, nous nous sommes inspirés de l'approche de [Kacem]celle de [Bouhini]pour la modélisation et l'exploitation de profil utilisateur, ainsi que celle de [Masaki]pour intégrer la notion du temps. Nous proposons donc de modéliser le profil utilisateur par un vecteur de poids de terme dans un premier temps, nous l'intégrons par la suite à la reformulation de la requête ainsi qu'à l'étape de l'interrogation. Nous avons choisis de modéliser ce dernier à partir des termes composants les différents tweets qu'il a retweeté. Notre choix s'est porté sur les retweets en partant du principe qu'un tweet retweeté par l'utilisateur intéresse ce dernier, autrement dit, son contenu concrétise ces préférences ou centres d'intérêts. Pour chaque terme t_i on associe un poid $W_{(U,ti)}$ calculé par une simple fonction de pondération. On propose par la suite de calculer un Score(T,U) concrétisant le score sociale d'un tweet pour un utilisateur. Enfin, on associe le score thématique au score sociale qu'on combinera avec la fonction temporelle de Masaki Aono [Masaki]. Le principe de cette approche est de retourner les tweets les plus pertinents et les plus frais pour une requête formulée par un utilisateur selon son profil.

4.2.2 Formule de l'approche proposée

4.2.2.1 Modélisation du profil

On propose de modéliser le profil utilisateur par un vecteur de poids de termes comme suit :

$$\overrightarrow{Pu} : (t_1 : w_{U,t_1}; t_2 : w_{U,t_2}; .....t_n : w_{U,t_n})$$

- t_i : terme du profil utilisateur
- $W_{U,t}$: poid du terme **t** dans le profil **Pu** calculé par une fonction de pondération :

$$W_{U,t} = \sum_{t=1}^{t \epsilon T} T f_{t,T} * Idf_t$$

où **Idf**$_t$ est calculé comme suit :

$$Idf_t = log(\frac{N}{n})$$

- Tf$_{t,T}$: est la fréquence d'un terme **t** dans un Tweet **T**
- N : le nombre de tweets publiés par l'utilisateur
- n : le nombre de tweets publiés par l'utilisateur qui contiennent le terme **t**

4.2.2.2 Reformulation de la requête

On propose de repondérer les termes de la requête **q** qui apparaissent dans le profil utilisateur pur formuler une requête **q'**. Dans un premier temps on sélectionne les termes candidats de la requête en calculant un Score d'appartenance **Sa** , ce dernier est valeur booléenne représenté comme suit :

$$S_a = \begin{cases} 1 & S\ un\ terme\ t\ appartient\ à\ la requete\ et\ au\ profil \\ 0 & Si\ non \end{cases}$$

On associe par la suite ce score S_a pour un terme **t** avec son poid dans le profil $W_{U,t}$ ceci nous permet de calculer un score qu'on appelera score d'ajoutS_q. Il sera egale au poid du terme dans le profil ou à 0, ce dernier est calculé comme suit :

$$S_q = S_a * W_{U,t}$$

- $W_{U,t}$: poid du terme **t** dans le profil **Pu .**
- S_a : score d'appartenance.
- S_q : score d'ajout

On perpondére alors les poids des termes dans la requête comme suit :

$$W_{t,q'} = W_{t,q} + S_q$$

-$W_{t,q}$: le poid initiale d'un terme **t** dans la requête **q**
- $W_{q'}$:le nouveau poids du terme **t** dans la requête repondérée **q'**

4.2.2.3 Calcule du score

Notre approche combine le score thématique RSV(Q,T), le score d'un tweet pour un utilisateur et la fonction temporelle Masaki Aono :

$$Score(T,U,Q) = (\ RSV(Q,T) + Score(T,U)\ ) * \frac{1}{\sqrt{Query_{Time(Q)} - Tweet_{Time(T)} + 1}}$$

- RSV(Q,T) : score thématique calculé par Lucene.

- QueryTime : est la date de soumission de la requete de l'utilisateur
- TweetTime : est la date de publication du tweet
- Score(T,U) est le score d'un tweet pour l'utilisateur, calculé en combinant le poid du terme **t** dans un profil **Pu** avec son poid dans un tweet **T** comme suit :

$$score(T,U) = \sum_{te T \bigcap U} w_{U,t} * w_{T,t}$$

- $W_{T,t}$: est la fréquence d'un terme **t** dans un tweet **T**, autrement dit le nombre de fois où **t** est répété dans **T**.
- $W_{u,t}$: est le poid d'un terme **t** dans un profil **U**

4.2.3 Discussion

Afin de modéliser cette approche, nous avons besoin des requetes de chaque utilisateur de notre collection, or la collection don't on dispose ne posséde que des requetes prédéfinie posés par les chercheurs pour estimer la pertinence d'un modèle de recherche. malheureusement, méme sur l'API de twitter, les requete d'un utilisateur sont confidentielles, propre à lui ce qui veut dire que ce sont des données innaccessible pour le grand public à la difference des tweets, commentaires, j'aimes ...etc.

Comme on ne peut plus modéliser cette approche, et en nous inspirons de l'approche [Masaki aono] sur le reclassement des résultats, nous avons été amené à proposer une deuxiéme approche quand décrira dans la section qui suit

4.3 Approche II

Notre approche propose l'intégration de l'utilisateur comme facteur de pertinence, nous partant donc du principe qu'un tweet est pertinent si son éditeur est pertinent. On propose de mesurer l'importance d'un utilisateur par rapport à sa credibilité, et son activité sur twitter .Un utilisateur à qui on fais confiance est un utilisateur important, d'autre part un utilisateur qui publie beaucoup est aussi important. De ce fait on propose de calculer le nobmre de followers d'un utilisateur qu'on combinera avec le nombre des tweets qu'il a publié et le nombre de fois ou il a été mentionné(tagué).

4.3.1 Formule de l'approche proposée

Notre approche combine donc le score thématique d'un tweet pour une requete avec son score sociale calculé comme suit :

$$Score(Q,T_i) = \alpha RSV(Q,T_i) + \beta Score(U,T_i)$$

- RSV(Q,Ti) : est le score thématique calculé par Lucene.

- Score(U,Ti) est un score sociale

Le score sociale combine les trois facteurs liés à l'utilisateur, à savoir le nombre de ses tweets, le nombre de ses followers, et le nombre de ses mentions. Malheureusement, nous n'avons pas pu récupéré le nombre de mentions d'un utilisateur sur twitter. Dans ce qui suit, on considérera que le score sociale est calculé avec le deux permiers facteurs seulement.

$$Score(U, T_i) = log\ Nbr_{(followers)} + log\ Nbr_{(tweets)}$$

-Nbr$_{followers}$: nombre de followers d'un utilisateur sur twitter

-Nbr$_{tweets}$: nombre de tweets(y compris retweets) publiés par un utilisateur sur twitter

4.4 Conclusion

Dans ce chapitre, nous avons décrit nos deux approche intégrant le profil utilisateur dans twitter. Dans ce qui suit, nous allons expérimenter la deuxiéme approche afin de voir si elle apporte une amélioration par rapport aux résultats obtenus par le score thématique.

[Auteur]« Ma citation préférée. »" Titre de la partie 3

Chapitre 5

Implémentation et expérimentation

5.1 Introduction

Dans le chapitre précédent, nous avons présenté notre approche qui intègre le profil utilisateur dans la recherche d'information, en se basant sur le nombre de followersainsiquelenombredetweetspubliésparcedernier.Dansleprésentchapitre nous présentons les différents outils utilisés pour l'implémentation de notre approche.Nous exposerons également les résultats des testes obtenus et les discuterons

5.2 Outils de développement

Pour réaliser nos tests, nous avons utilisés la collection trec microblog 2011. Pour l'implémentation de notre approche, nous avons utilisé le langage JAVA avec l'importation des librairies LUCENE.Afin d'extraire les tweets sur Twitter via leurs ids récupérés depuis la collection, nous avons utilisé l'API TWITTER 4j.Les tweets obtenus sont fournis en format json, ce qui nous a amemés à utiliser l'API JACKSON pour les indexer sous java.Enfin, pour l'evaluation des résultats de notre approche nous avons utilisé TREC_EVAL à travers des commandes tapées sur ubuntu.

5.2.1 Eclipse IDE

Eclipse est un environement de développement (IDE) placé en open source. En plus de java, Eclipse peut être utilisé avec d'autre langages de programmation comme PHP et C/C++. La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plug-in .

FIGURE 5.2.1 – Logo éclipse

5.2.2 Langage Java

java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton employés de Sun Microsystème. Sa particularité repose sur le fait qu'il est facilement portable sur plusieur systèmes d'exploitation tels que linux,Windows, Mac.....avec aucune ou peut de modification.

5.2.3 Lucene

Lucene est un moteur de recherche textuelle développé sous java par la fondation apache. Lucene est open source, Il est utilisé dans les applications java pour ajouter une fonction de recherche de documents de manière très simple et efficace.Lucene permet d'indexer des documents pour ensuite soumettre des requêtes au moteur de recherche. Le processus d'indexation et de recherche met en œuvre les classes suivantes :

5.2.3.1 Les classes d'indexation

- **IndexWriter :** La classe IndexWriter est le composant central du processus d'indexation. Elle agit comme un composant principal qui crée et met à jour des index pendant le processus d'indexation.
- **Directory** : La classe Directory représente l'emplacement de l'index de lucene. IndexWriter utilise une des implémentations de Directory, FSDirectory, pour créer son index dans un répertoire dans le Système de fichiers. Une autre implémentation, RAMDirectory, prend toutes ses données en mémoire.
- **Analyzer :** Avant que le texte soit dans l'index, il passe par l'Analyser. Il s'agit d'un ensemble de classes qui ont pour but le découpage du texte en tokens et la normalisation du texte à indexer.
- **Document :** La classe Document représente un rassemblement de champs. Les champs d'un document représentent le document ou les métadonnées associées avec ce document. Les métadonnées comme l'auteur, le titre, le sujet, la date, etc. sont indexées et stockées séparément comme des champs d'un document.
- **Field :** Chaque document est un index contenant un ou plusieurs champs, inséré dans une classe intitulé Field. Chaque champ (field) correspond à une portion de donnée qui est interrogé ou récupéré depuis l'index durant la recherche

5.2.3.2 Les classes de recherche

- **IndexSearcher :** La classe IndexSearcher est à la recherche de ce que IndexWriter a indexé. On peut la représenter comme une classe qui ouvre un index en mode lecture seule.
- **Term :** Un terme est une unité basique pour la recherche, similaire à l'objet field. C'est une chaîne de caractère.

— **Query :** représente la requête de l'utilisateur.
— **QueryParser :** La classe QueryParser est utilisée pour générer un décompositeur analytique qui peut chercher à travers un index.
— **Hits :** La classe Hits est un simple conteneur d'index pour classer les résultats de recherche de documents qui apparaissent pour une interrogation donnée.

5.2.4 L'Api Twitter

Twitter propose plusieurs Apis permettant d'accéder à ses services, permettant des simples consultations des tweets, followers, amis,etc ainsi que des oppérations de modifications tels que la suppression ou l'ajout d'un ami, la publication d'un tweet, etc.

L'API twitter 4j est la plus répondu, c'est une bibliothèque Java permettant d'intégrer facilement son application Java au service Twitter, il suffit juste d'ajouter twitter4j-core-(version).jar au chemin d'accès aux classes(Build Path).

5.2.5 L'Api Jackson

Jackson est l'une des bibliothéques Java les plus efficaces, elle assure la sérialisation ou le mappage des objets Java en Json et inversement. Elle est open source, facile à ytiliser, et ne nécessite aucune autre bibliothèque que jdk.

5.2.6 Collection TREC microblogs2011

TREC microblogs2011 nous fourni un corpus de tweets publiés en 2011 classé par ordre chronologique. Il fourni également un ensemble de 50 requetes et des jugements de pértinences associés à ces requetes.

5.2.7 Trec eval

C'est un outil utilisé pour évaluer les classement des documents triés par pertinence. Pour cela, on utilise les deux fichiers Qrels et Results définis commes uit :

— Qrels : répertorie les jugements de pertinence pour chaque requete
— Résults : contient le classement des résultats renvoyés par l'SRI.

Trec_eval est open source, aprés l'avoir télécharger on y accéde via des commande tapées sur Ubuntu. Pour pouvoir executer Trec_eval, il suffit de taper la commande suivante :

$ ./trec_eval -c qrel_file result_file

- trec_eval : le nom du programme exécutable.

-c : effectue une moyenne sur l'ensemble des requêtes dans les jugements de pertinence

- qrels_file : le chemin vers le fichier contennat les jujements de pertinences pour chaque requete

- results_file : le chemin vers le fichier contenant les documents retournés par l'SRI (les scores)

qrels_file et results_file porte l'extension .test

plusieur autre possibilitées d'évalutaion sur trec_eval, il suffit de remplacer -c par d'autre mesure comme suit :

-q : pour pouvoir matcher les qrels et les résultats

-m : affiche uniquement une mesure spécifique.

-m all_trec : affiche toutes les mesures

-m official : affiche uniquement les mesures principales.

pour plus de commandes et l'interpretation correcte des resultats, il suffit de tapper :**$ ./trec_eval -h qrel_file result_file**

5.3 Implémentation de l'approche II

5.3.1 Notre collection de tests

Pour l'implémentation de notre approche, nous avons utilisé la collection réduite de TREC microblogs 2011, elle est composée de :

— 40603 tweets

— 50 requetes

— les jugements de pertinence associés à ces requetes

De cette collection nous avons extrait une collection composé de

— 5 requetes (numérotées de 20 à 24 sur la collection)

— les tweets associés à ces requetes, pour plus de réalité nous avons ajouté des tweets aléatoir à la collection

— les jugements de pertinence associés à ces requetes

5.3.2 Les classes implémentées

Pour l'implémentation de notre approche, nous avons utilisé les indexe issus de Lucene et etendu ce dernier avec les classes suivantes :

Resultats et InfluenceBoosting.

5.3.2.1 La classe Resultats

Dans cette classe, nous avons mis deux fonction qui prennet l'id d'un utilisateur en parametre et nous retourne respéctivement le nombre de followers et de tweets publiés par un utilisateur. Cette classe est appelé par la suite à partir de InfluenceBoosting afin de récupérer les valeurs dont on a besoin pour l'implementation de notre approche.

```java
package com.tutorialspoint.lucene;
import java.util.ArrayList;
public class TESTE {



    public static int recup(int id) throws TwitterException, InterruptedException
    {
       Twitter twitter = new TwitterFactory().getInstance();
        TwitterResponse response = twitter.showUser(id);
           RateLimitStatus status = response.getRateLimitStatus();

           int user = twitter.showUser(id).getStatusesCount();


                       return (user );}

   public static int recupf(int id) throws TwitterException, InterruptedException
    {
       Twitter twitter = new TwitterFactory().getInstance();
        TwitterResponse response = twitter.showUser(id);
           RateLimitStatus status = response.getRateLimitStatus();
             int followers = twitter.showUser(id).getFollowersCount();

           return followers;
       }

}
```

FIGURE 5.3.1 – Classe Results

5.3.2.2 La classe InfluenceBoosting

C'est une classe qui étend CustomScoreQuery de lucene, c'est ici que l'ajout du score thématique au score sociale se fait. Dans cette classe, nous avons implémenté notre approche, qui calcule le log du nombre de followers associés au log du nombre de tweets d'un utilisateur. Pour récupérer les valeurs tels que le nombre de followers et le nombre de tweets d'un utilisateur on peut procéder de

deux méthodes différentes :

- A partir de l'indexer : durant la phase de l'indexation, lucene offre la possibilité de récupérer en meme temps les informations sur twitter qu'il stocke dans un field (le cache).
- A partir d'une classe externe (Results) : exécutant des fonctions qui prennent l'Id user comme paramettre et retournent le nombre de followers et de tweets de ce meme utilisateur.

```
private class RecencyBooster extends CustomScoreProvider {
    final String[] values;
    final long [] id_tweet;
    final int[] id;
    final long [] abn;
    final long [] tws;

public RecencyBooster(IndexReader r) throws IOException  {

         super(r);

         values    = FieldCache.DEFAULT.getStrings(r, LuceneConstants.AUTH);
         id_tweet  = FieldCache.DEFAULT.getLongs(r, LuceneConstants.ID_TWEET);
         id        = FieldCache.DEFAULT.getInts(r, LuceneConstants.ID_AUTHEUR);
         abn       = FieldCache.DEFAULT.getLongs(r, LuceneConstants.ABONNE);
         tws       = FieldCache.DEFAULT.getLongs(r, LuceneConstants.TWEETS);


     }

public float customScore(int doc, float subQueryScore,float valSrcScore) {
        float score = 0;



        String auth= values[doc];
        int idu = id [doc];



        // récupération depuis le cache
        long followers = abn [doc];
        long tweets = tws [doc];
```

FIGURE 5.3.2 – Récupération des valeurs à partir des fields cache

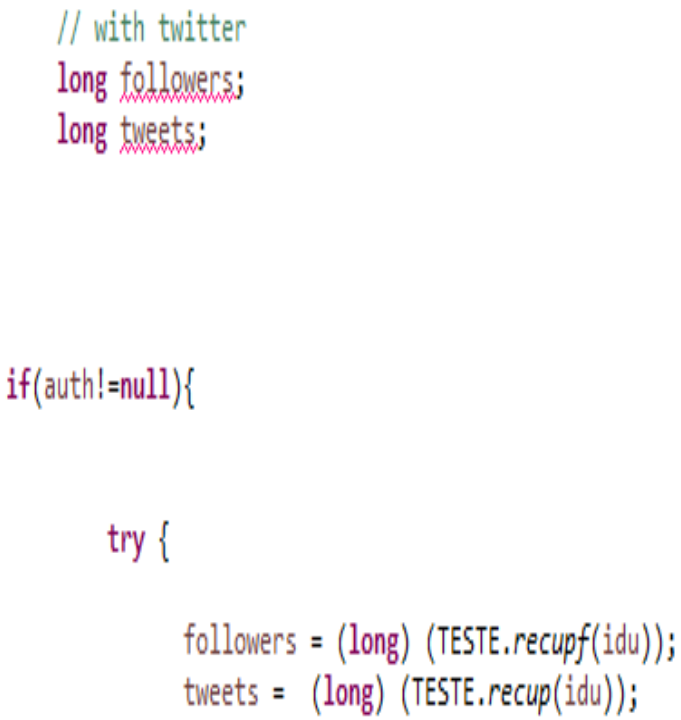


FIGURE 5.3.3 – Récupération des valeur à partir d'une classe extérne

Une fois les valeurs récupérer, on procéde au calcule du score sociale par notre approche qu'on combinera avec le score thématique. Cette classe est appelé lors de la phase de la recherche, en exécutant la classe Searcher du lucene.

```
        score = (float) ( Math.log( tweets) + Math.log(followers));

            } catch (Exception e) {

                e.printStackTrace();

            }

            //return  0.01f*subQueryScore + 0.99f*score;
            return  0.3f*subQueryScore + 0.7f*score;

    }else return    subQueryScore;



    }



     }
```

FIGURE 5.3.4 – Calcule du score finale en combinant les deux score thématique et sociale

```
for(String queryString : top_inf.getTopics()){

    System.out.println(j+" "+queryString);
    String qs = queryString;

     Query q = queryParser.parse(qs);
     Query q2 = new  InfluenceBoosting(q);
     Sort sort = new Sort(new SortField[] { SortField.FIELD_SCORE, new SortField("********", SortField.STRING)});

     TopDocs hits = searcher.search(q2, null, 1000, sort);//fixer les resultats à 100


    System.out.println("tttt");
    writeInFile(LuceneConstants.RESULTS,hits,queries[j]+"");


    //int agb = hits.scoreDocs.length
for (int i = 0; i < hits.scoreDocs.length; i++) {
```

FIGURE 5.3.5 – Appel à la classe InfluenceBoosting depuis le Searcher

5.4 Résultats et évaluations

Pour réaliser nos tests, nous avons commencer par récupérer le score thématique dans un premier temps, puis par la suite nous avons combiner ce score avec un score sociale incluant uniquement le nombre de tweets ou de followers. Cela nous a permis de voir l'impacte de ces deux facteur sur les résultats de recherche. Et enfin, nous avons combiner le score des deux facteurs avec le score thématique. Chaque fois qu'un score est récupéré, on évalue ce dérnier par les différentes mesures d'evaluation.

5.4.1 Résultats avec le score thématique

Voici quelques résultats des score obtenus suite à la recherche thématique effectuée sur les requetes de notre collection :

$$Score(Q, T_i) = RSV(Q, T_i) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots A$$

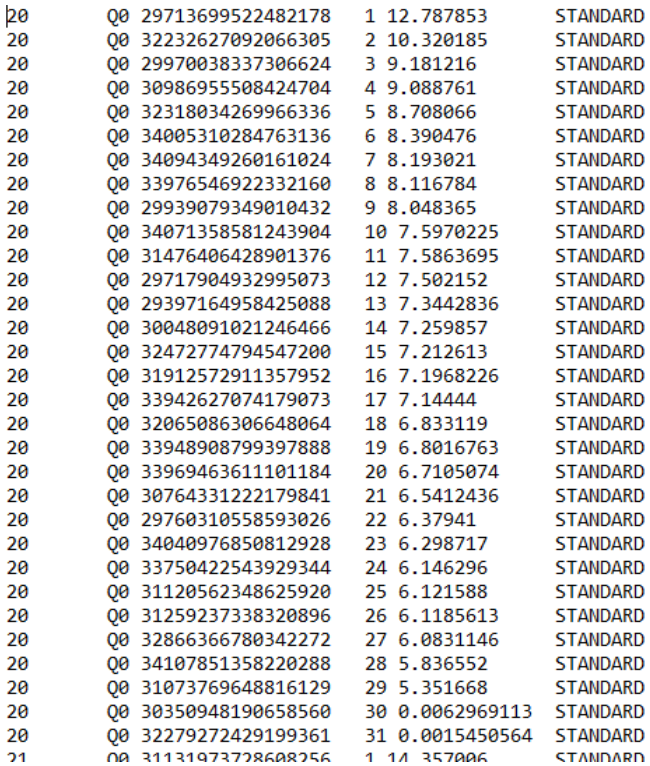
```
20      Q0 30986955508424704   1 1.2899866      STANDARD
20      Q0 31073769648816129   2 1.1976221      STANDARD
20      Q0 30048091021246466   3 1.1225663      STANDARD
20      Q0 29939079349010432   4 0.8982165      STANDARD
20      Q0 32866366780342272   5 0.89051783     STANDARD
20      Q0 31912572911357952   6 0.89051783     STANDARD
20      Q0 33750422543929344   7 0.77920306     STANDARD
20      Q0 30350948190658560   8 0.6296911      STANDARD
20      Q0 32065086306648064   9 0.53915733     STANDARD
20      Q0 32472774794547200   10 0.47226837    STANDARD
20      Q0 32232627092066305   11 0.47226837    STANDARD
20      Q0 34040976850812928   12 0.44525892    STANDARD
20      Q0 34005310284763136   13 0.44525892    STANDARD
20      Q0 31259237338320896   14 0.44525892    STANDARD
20      Q0 29970038337306624   15 0.44525892    STANDARD
20      Q0 29760310558593026   16 0.44525892    STANDARD
20      Q0 29713699522482178   17 0.44525892    STANDARD
20      Q0 29397164958425088   18 0.44525892    STANDARD
20      Q0 32318034269966336   19 0.40436798    STANDARD
20      Q0 30764331222179841   20 0.33394417    STANDARD
20      Q0 29717904932995073   21 0.33394417    STANDARD
20      Q0 34094349260161024   22 0.19124702    STANDARD
20      Q0 33976546922332160   23 0.17657788    STANDARD
20      Q0 33969463611101184   24 0.17657788    STANDARD
20      Q0 32279272429199361   25 0.15450564    STANDARD
20      Q0 33948908799397888   26 0.13523206    STANDARD
20      Q0 33942627074179073   27 0.13523206    STANDARD
20      Q0 31120562348625920   28 0.13243341    STANDARD
20      Q0 34107851358220288   29 0.10928401    STANDARD
20      Q0 34071358581243904   30 0.099177256   STANDARD
20      Q0 31476406428901376   31 0.099177256   STANDARD
21      Q0 30817677584900096   1 0.48099002     STANDARD
```

FIGURE 5.4.1 – Aperçu des résultats de la recherche thématique

5.4.2 Résultats avec le nombre de Followers

La recherche thématique combinée avec le score sociale calculé par la formule **B** ci-dessous, effectuée sur les requêtes de notre collection nous a donné les résultats suivant :

$$Score(U, T_i) = log\ Nbr_{(followers)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots B$$



```
20      Q0 29713699522482178   1 12.787853      STANDARD
20      Q0 32232627092066305   2 10.320185      STANDARD
20      Q0 29970038337306624   3 9.181216       STANDARD
20      Q0 30986955508424704   4 9.088761       STANDARD
20      Q0 32318034269966336   5 8.708066       STANDARD
20      Q0 34005310284763136   6 8.390476       STANDARD
20      Q0 34094349260161024   7 8.193021       STANDARD
20      Q0 33976546922332160   8 8.116784       STANDARD
20      Q0 29939079349010432   9 8.048365       STANDARD
20      Q0 34071358581243904   10 7.5970225     STANDARD
20      Q0 31476406428901376   11 7.5863695     STANDARD
20      Q0 29717904932995073   12 7.502152      STANDARD
20      Q0 29397164958425088   13 7.3442836     STANDARD
20      Q0 30048091021246466   14 7.259857      STANDARD
20      Q0 32472774794547200   15 7.212613      STANDARD
20      Q0 31912572911357952   16 7.1968226     STANDARD
20      Q0 33942627074179073   17 7.14444       STANDARD
20      Q0 32065086306648064   18 6.833119      STANDARD
20      Q0 33948908799397888   19 6.8016763     STANDARD
20      Q0 33969463611101184   20 6.7105074     STANDARD
20      Q0 30764331222179841   21 6.5412436     STANDARD
20      Q0 29760310558593026   22 6.37941       STANDARD
20      Q0 34040976850812928   23 6.298717      STANDARD
20      Q0 33750422543929344   24 6.146296      STANDARD
20      Q0 31120562348625920   25 6.121588      STANDARD
20      Q0 31259237338320896   26 6.1185613     STANDARD
20      Q0 32866366780342272   27 6.0831146     STANDARD
20      Q0 34107851358220288   28 5.836552      STANDARD
20      Q0 31073769648816129   29 5.351668      STANDARD
20      Q0 30350948190658560   30 0.0062969113  STANDARD
20      Q0 32279272429199361   31 0.0015450564  STANDARD
21      Q0 31131973728608256   1 14.357006      STANDARD
```

FIGURE 5.4.2 – Aperçu des résultats de la recherche thématique avec score social exploitant le nombre de followers

5.4.3 Résultats avec le nombre de Tweets

Ci-dessous quelques résultats des score obtenus suite à la recherche thématique combinée avec le score sociale effectuée sur les requetes de notre collection. Le score sociale est calculé comme suit :

$$Score(U, T_i) = log\ Nbr_{(tweets)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots C$$

20	Q0	30986955508424704	1	0.012899865	STANDARD
20	Q0	31073769648816129	2	0.011976219	STANDARD
20	Q0	30048091021246466	3	0.011225663	STANDARD
20	Q0	29939079349010432	4	0.008982166	STANDARD
20	Q0	32866366780342272	5	0.008905178	STANDARD
20	Q0	31912572911357952	6	0.008905178	STANDARD
20	Q0	33750422543929344	7	0.0077920305	STANDARD
20	Q0	30350948190658560	8	0.0062969113	STANDARD
20	Q0	32065086306648064	9	0.005391573	STANDARD
20	Q0	32472774794547200	10	0.0047226837	STANDARD
20	Q0	32232627092066305	11	0.0047226837	STANDARD
20	Q0	34040976850812928	12	0.004452589	STANDARD
20	Q0	34005310284763136	13	0.004452589	STANDARD
20	Q0	31259237338320896	14	0.004452589	STANDARD
20	Q0	29970038337306624	15	0.004452589	STANDARD
20	Q0	29760310558593026	16	0.004452589	STANDARD
20	Q0	29713699522482178	17	0.004452589	STANDARD
20	Q0	29397164958425088	18	0.004452589	STANDARD
20	Q0	32318034269966336	19	0.0040436797	STANDARD
20	Q0	30764331222179841	20	0.0033394417	STANDARD
20	Q0	29717904932995073	21	0.0033394417	STANDARD
20	Q0	34094349260161024	22	0.0019124701	STANDARD
20	Q0	33976546922332160	23	0.0017657788	STANDARD
20	Q0	33969463611101184	24	0.0017657788	STANDARD
20	Q0	32279272429199361	25	0.0015450564	STANDARD

FIGURE 5.4.3 – Aperçu des résultats de la recherche thématique avec score social exploitant le nombre de tweets

5.4.4 Résultats avec le nobmre de tweet et de followers

En combinant le score thématique avec le score sociale calculé par la formule **D** on a eu les résultats suivants :

$$Score(U, T_i) = log\ Nbr_{(tweets)} + log\ Nbr_{(tweets)} \ldots\ldots\ldots\ldots\ldots\ldots D$$

```
20      Q0 29713699522482178   1 24.50602       STANDARD
20      Q0 32232627092066305   2 22.694784      STANDARD
20      Q0 30986955508424704   3 21.82583       STANDARD
20      Q0 34005310284763136   4 20.078014      STANDARD
20      Q0 32318034269966336   5 20.049429      STANDARD
20      Q0 34071358581243904   6 19.852707      STANDARD
20      Q0 29970038337306624   7 19.786772      STANDARD
20      Q0 32472774794547200   8 19.452778      STANDARD
20      Q0 29939079349010432   9 19.314384      STANDARD
20      Q0 34094349260161024   10 19.182602     STANDARD
20      Q0 29717904932995073   11 19.147715     STANDARD
20      Q0 31912572911357952   12 19.05537      STANDARD
20      Q0 32065086306648064   13 17.898174     STANDARD
20      Q0 31476406428901376   14 17.395906     STANDARD
20      Q0 29397164958425088   15 17.367863     STANDARD
20      Q0 33976546922332160   16 17.251614     STANDARD
20      Q0 33942627074179073   17 17.199568     STANDARD
20      Q0 29760310558593026   18 16.907953     STANDARD
20      Q0 33750422543929344   19 16.856213     STANDARD
20      Q0 32866366780342272   20 16.659304     STANDARD
20      Q0 34040976850812928   21 16.595842     STANDARD
20      Q0 34107851358220288   22 16.224533     STANDARD
20      Q0 30764331222179841   23 16.095587     STANDARD
20      Q0 33948908799397888   24 16.089378     STANDARD
20      Q0 30048091021246466   25 15.974205     STANDARD
20      Q0 33969463611101184   26 15.891647     STANDARD
20      Q0 31259237338320896   27 15.848237     STANDARD
20      Q0 31073769648816129   28 15.520685     STANDARD
20      Q0 31120562348625920   29 15.202706     STANDARD
20      Q0 30350948190658560   30 0.0062969113  STANDARD
20      Q0 32279272429199361   31 0.0015450564  STANDARD
```

FIGURE 5.4.4 – Aperçu des résultats de la recherche thématique avec score social exploitant le nombre de tweets et le nombre de followers

Voici un apperçu de l'évaluation de cette approche sur Trec_eval :

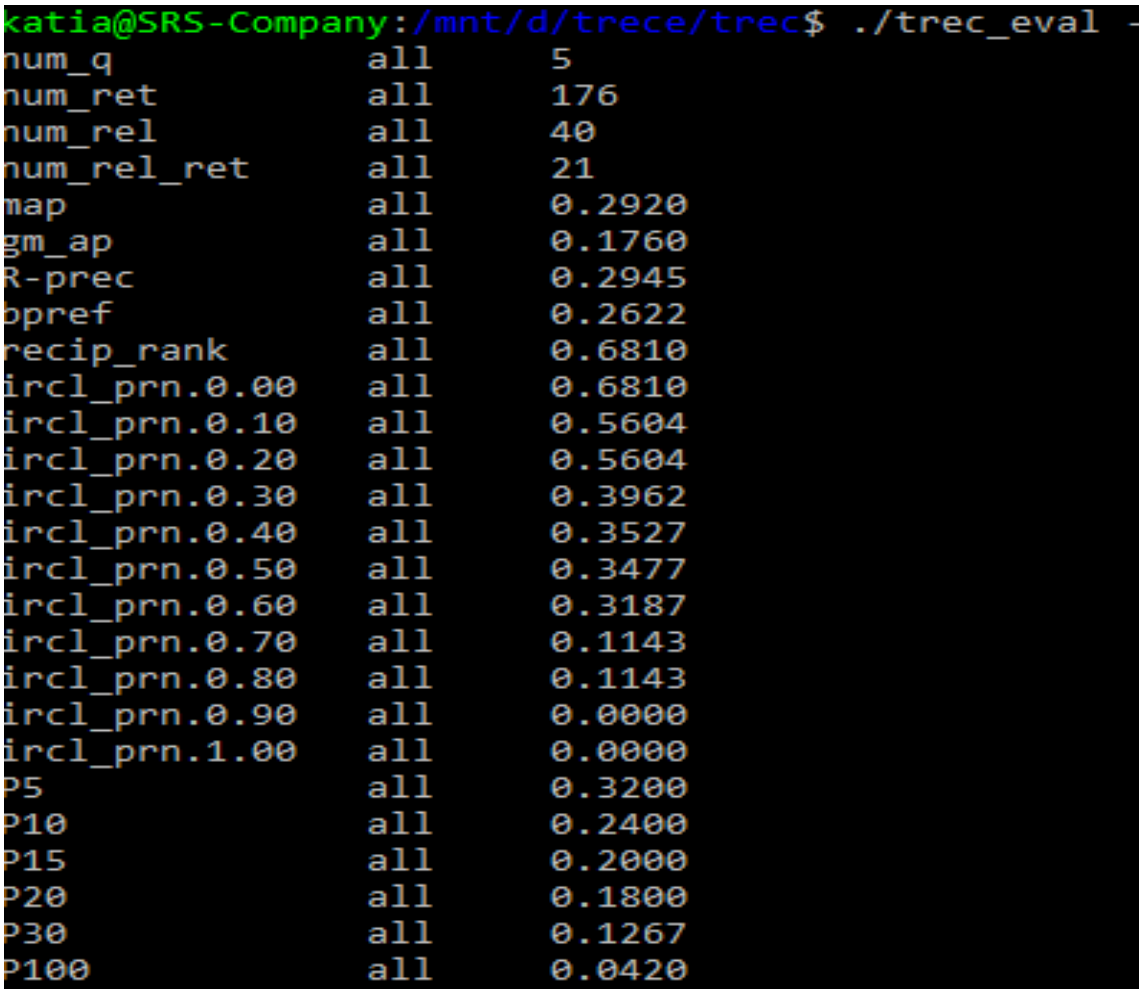


FIGURE 5.4.5 – Aperçu sur l'évaluation Trec_eval

5.4.5 Evaluation des résultats

Pour estimer la qualité des listes de résultats produites selon les différentes approches cités dans cette phase d'évaluation, nous avons utilisé les mesures d'évaluation suivantes : La précision@X, MAP , Précision Moyenne et R-précision, Rappel, Précision et F-mesure, et le Rappel interpolé.

5.4.5.1 La précision@X

	Formule (A)	Formule (B)	Formule (C)	Formule (D)
P5	0.3600	0.3600	0.3200	0.3200
P10	0.2200	0.2200	0.2200	0.2400
P15	0.2000	0.2000	0.1867	0.2000
P20	0.1700	0.1700	0.1600	0.1800
P30	0.1267	0.1267	0.1267	0.1267
P100	0.0420	0.0420	0.0420	0.0420
P200	0.0210	0.0210	0.0210	0.0210
P500	0.0084	0.0084	0.0084	0.0084
P1000	0.0042	0.0042	0.0042	0.0042

FIGURE 5.4.6 – L'évaluation par la precision@X de notre approche

Nous constatons que la formule D, c'est à dire le calcule du score sociale en combinant le nombre de tweets et de followers d'un utilisateur apporte une légére amélioration par rapport à l'approche thématique. La précision à 10 documents ainsi qu'à 20 documentes marque une légére augmentation, tandis que la précision à 5 documents baisse légèrement.

5.4.5.2 La MAP, Précision Moyenne et R-précision

	Formule (A)	Formule (B)	Formule (C)	Formule (D)
MAP	0.2252	0.2252	0.3076	0.2920
Précision Moyenne	0.1675	0.1675	0.1805	0.1760
R-précision	0.2461	0.2461	0.3345	0.2945

FIGURE 5.4.7 – L'évaluation par La MAP, Précision Moyenne et R-précision de notre approche

Les valeurs de la Map, Précision moyenne, et R-précision de notre approche sont supérieur à celles obtenus par le score thématique. que ce soit en intégrant le nombre de followers seulement ou en combinant ce dernier avec le nombre de tweets

5.4.5.3 Le Rappel, Précision et F-mesure

	Formule (A)	Formule (B)	Formule (C)	Formule (D)
Rappel	0.525	0.525	0.525	0.525
Précision	0.119	0.119	0.119	0.119
F-mesure	0.1940	0.1940	0.1940	0.1940

FIGURE 5.4.8 – L'évaluation par le Rappel,Précisions et F-mesure de notre approche

On constate que les valeurs du Rappel, Précision et F-mesure sont pareil pour l'approche thématique et la notre.

5.4.5.4 Rappel interpolé - Précision moyenne à Y rappel

	Formule (A)	Formule (B)	Formule (C)	Formule (D)
ircl_prn.0.00	0.6743	0.6743	0.6810	0.6810
ircl_prn.0.10	0.6632	0.6632	0.5667	0.5604
ircl_prn.020	0.6632	0.6632	0.5462	0.5604
ircl_prn.0.30	0.3442	0.3442	0.4325	0.3962
ircl_prn.0.40	0.1991	0.1991	0.3890	0.3527
ircl_prn.0.50	0.1511	0.1511	0.3018	0.3477
ircl_prn.0.60	0.1282	0.1282	0.2862	0.3187
ircl_prn.0.70	0.0471	0.0471	0.2000	0.1143
ircl_prn.0.80	0.0471	0.0471	0.2000	0.1143
ircl_prn.0.90	0.0000	0.0000	0.0000	0.0000
ircl_prn.1.00	0.0000	0.0000	0.0000	0.0000

FIGURE 5.4.9 – L'évaluation par la Rappel interpolé de notre approche

En évaluant avec le Rappel interpolé, on constate une nette amélioration de résultats obtenus par notre approche vis à vis des résultats thématique

5.4.6 Evaluation de l'approche

Notre approche améliore nettement les résultats selon les métriques utilisés. Les valeurs de la précision à 10 ainsi qu'à 20 documents de notre approche sont supérieurs à la thématique, ainsi qu'à la map, précison, et r-précisionle. Le rappele interpolé est nettement amélioré, quand au rappel, précision et f-mesure sont restés stables sur les deux approches. Ci-dessous les graphes illustrants les comparaisons de notre approche avec le score thématique selon les différentes mesures utilisées.

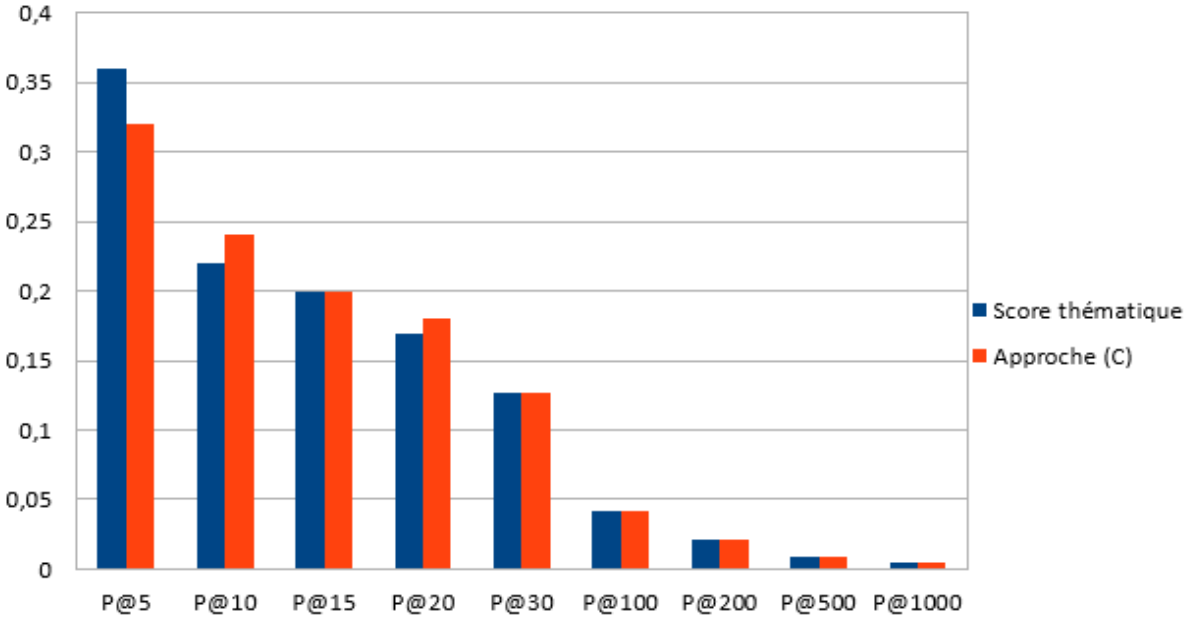


FIGURE 5.4.10 – Comparaison de la p@X entre notre approche avec le score thématique

Notre approche a donné des résultats similaires ou meilleurs par rapport à la thématiques en comparant les P@X des deux approches.

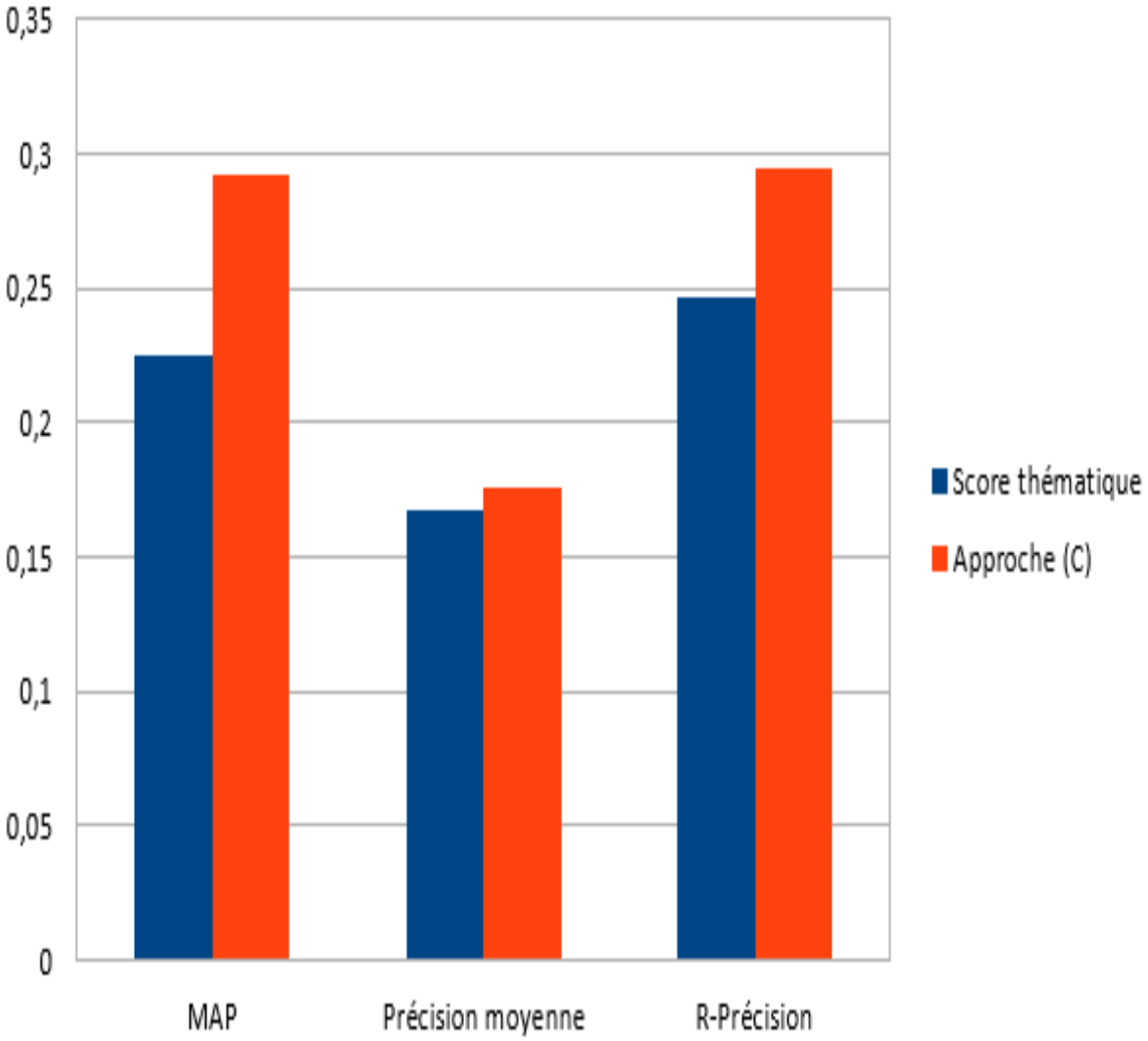


FIGURE 5.4.11 – Comparaison de la Map, Précision moyenne et R-précision entre notre approche avec le score thématique

Les résultatts de la Map, Précision moyenne et R-précision de notre approche sont nettement supérieurs à ceux de l'approche thématique.

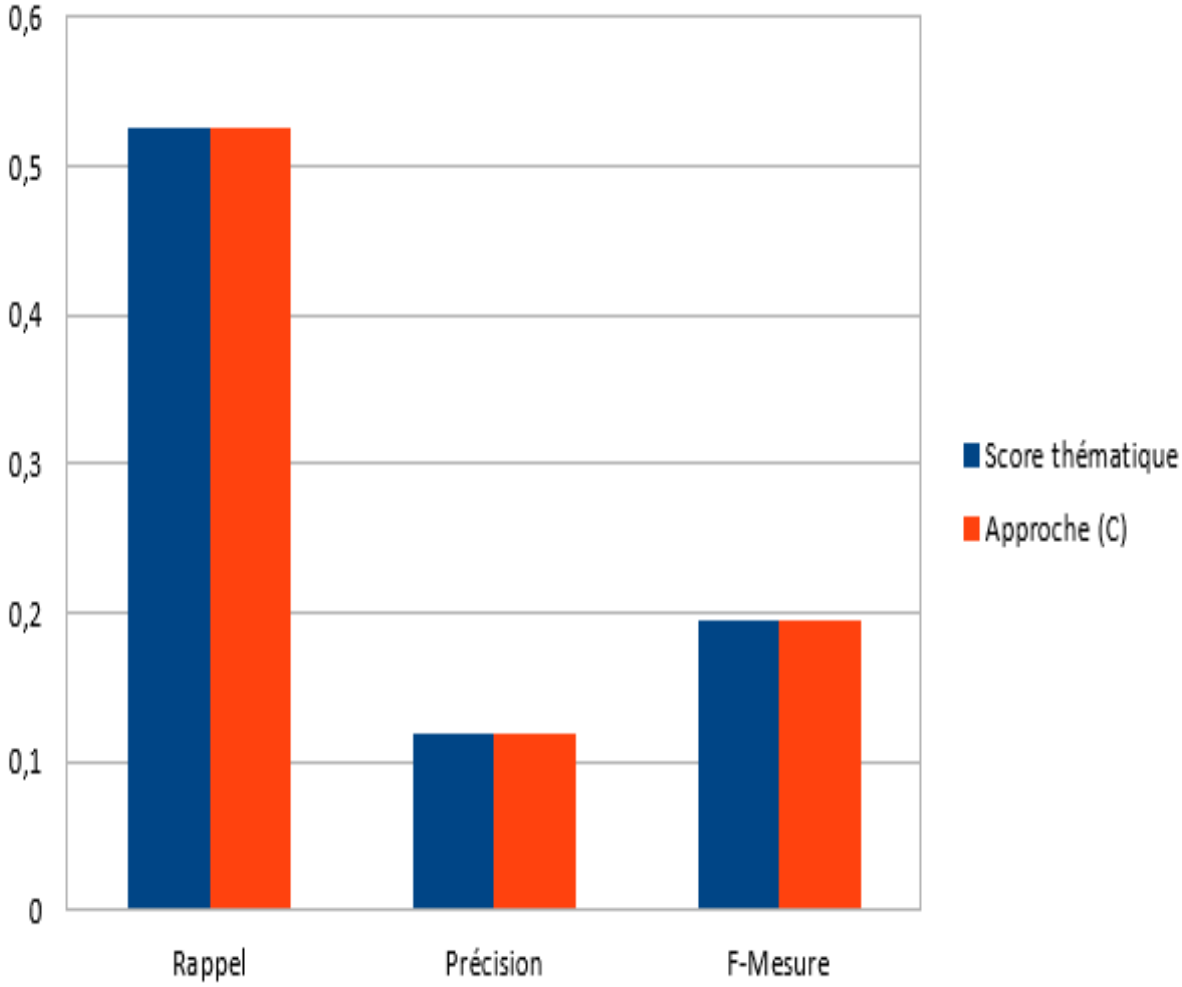


FIGURE 5.4.12 – Comparaison du Rappel, Précision et F-mesure eentre notre approche avec le score thématique

Le Rappel, Précision et R-precision de notre approche restent similaires à ceux de l'approche thématique.

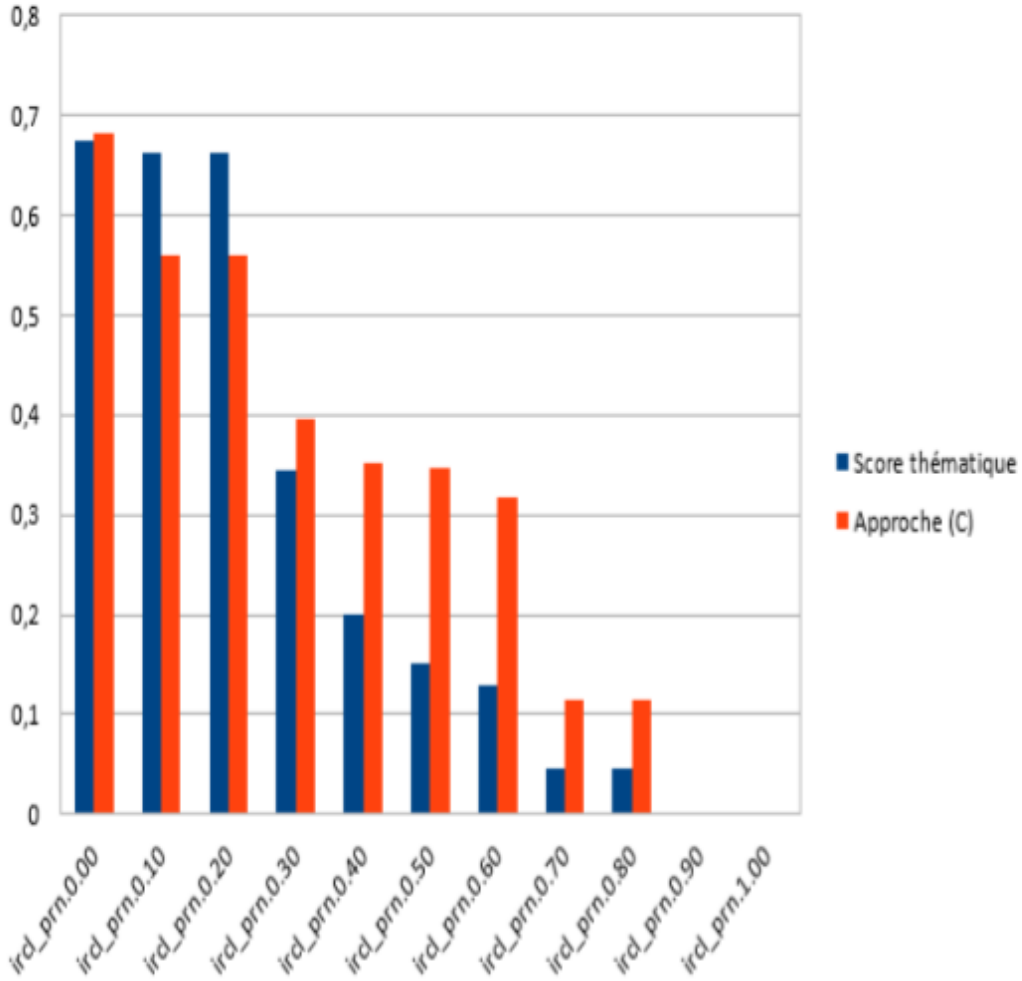


FIGURE 5.4.13 – Comparaison du Rappel interpolé entre notre approche avec le score thématique

D'après la comparaison par le rappel interpolé, on constate une amélioration des résultats obtenus par notre approche comparée à l'approche thématique.

5.5 Conclusion

Après de nombreuses séries de testes, et d'évaluations des résultats obtenus par différentes mesures, nous concluons que notre approche qui intégre le profil utilisateur apporte une nette amélioration par rapport à l'approche thématique. Nous avons obtenu des résultats satisfaisants. Cela témoigne de la pertinence de notre proposition et nous encourage à poursuivre dans ce domaine de recherche.

Conclusion et perspectives

Apports Dans notre travail nous nous sommes intéressées à la recherche d'information intégrant le profil utilisateur dans le réseau social Twitter. L'objectif étant de trouver les tweets les plus pertinents selon l'utilisateur. Nous avons commencer par introduire la recherche d'information classiques, nous nous sommes penchés par la suite vers la recherche d'information sociale, en donnant un bref aperçu sur les réseaux sociaux en générale. Nous avons été amené par la suite à parler sur Twitter plus spécifiquement, et passés en revu différentes approches proposées par les chercheurs. Nous avons finalement parler de notre approche exploitant le profil utilisateur dans Twitter, les différents ainsi que les résultats obtenus aprés son implémentation et son évaluation.

Limites Ce projet nous a ouvert une vue vers le domaine de la recherche d'information, il nous a permis de développer nos capacitées dans le domaine théorique ou pratique.

Nous avons réussi à proposer une approche dans le cadre de la recherche d'information dans Twitter, en intégrant le profil utilisateur. Nous avons réussis à implémenter cette approche, à obtenir des résultats et à évaluer ces dérniers selon différentes métriques. Et enfin, nous avons réussis à démontrer l'efficacité de notre approche dans la recherche d'information sur Twitter.

Perspectives A l'avenir, nous envisageons de tester notre approche sur une collection encore plus volumineuse. Nous essayerons également de récupérer les mentions d'un utilisateur sur Twitter afin de valoriser notre approche. Nous tenterons également de récupérer les requêtes des utilisateurs sur Twitter afin d'implémenter notre premiére approche.