REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

Mémoire de fin d'études de MASTER ACADEMIQUE

Domaine : Mathématique et Informatique
Filière : Informatique
Spécialité : Système Informatique

Thème

Recherche d'information temporelle dans les microblogs Cas de Twitter

Présenté par
M^lle^ BELKAID Cylia.
M^lle^ DOUADI Celia.

Devant le jury composé de :
Président: Mr AMIROUCHE Mohamed Nabil.
Encadreur : Mme AMIROUCHE Fatiha.
Examinatrice : Mme BELKACEMI Lila.
Examinatrice : M^lle^ ILTACHE Samia.

Promotion 2016/2017

''Almmud ar timtelt''

(L'apprentissage doit durer jusqu'au tombeau)

Proverbe Amazigh

Remerciements

Au terme de la rédaction de ce mémoire, c'est un devoir agréable d'exprimer en quelques lignes la reconnaissance que nous devons à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail, qu'ils trouvent ici nos vifs respects et notre profonde gratitude.

Tout d'abord, nous adressons toute notre gratitude à notre encadreur ***Mme AMIROUCHE FATIHA,*** *pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion. Nous la remercions également de nous avoir fait confiance et encouragé tout au long de ce projet.*

Nous tenons à remercier également les membres de jury d'avoir accepté d'évaluer notre travail.

Nous voudrions exprimer notre reconnaissance envers le doctorant ***FERROUK Massinissa*** *pour son aide, et les amis(es), qui nous ont apporté leur support moral et intellectuel tout au long de notre démarche.*

Dédicaces

À mes très chers parents

Je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos innombrables sacrifices. Que ce modeste travail, soit pour vous une petite compensation et reconnaissance de ce que vous avez fait d'incroyable pour moi. Quoi que je fasse ou que je dise, je ne saurais point vous remercier comme il se doit. Que DIEU, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

À mes très chers sœurs et frères

Lydia et son mari Mourad, Kamelia et son mari Lyes, Samir, Aghiles.

Ainsi à mes petits neveux que j'adore : Momoh, Aylan, Yanis et Alycia.

Aucune dédicace ne serait exprimer assez profondément de ce que je ressens envers vous. Je vous dirais tout simplement un grand merci, je vous adore.

À ma meilleure amie, et binôme Cylia

Je ne peux trouver les mots justes et sincères pour t'exprimer mon affection et mes pensées, tu es pour moi ma sœur sur qui je peux compter. En témoignage de l'amitié qui nous uni et des souvenirs de tous les moments que nous avons passé ensemble, je te dédie ce travail et je te souhaite une vie pleine de santé et de bonheur.

À mes très chers amis(es)

En témoignage de l'amitié sincère qui nous a lié(e)s et des bons moments passés ensemble je dédie ce travail à Kahina et Hayet, et à tous (tes) mes amis(es), en vous souhaitant un avenir radieux et plein de bonnes promesses.

CELIA DOUADI

Dédicaces

À mes très chers parents

Je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos innombrables sacrifices. Que ce modeste travail, soit pour vous une petite compensation et reconnaissance de ce que vous avez fait d'incroyable pour moi. Quoi que je fasse ou que je dise, je ne saurais point de vous remercier comme il se doit. Que DIEU, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

À mes très chers sœurs et frères

Fateh et sa femme Fazia, Karima et son mari Said, Ali et sa femme Hafida, Farid et sa femme Karima , Yacine et Amine. Ainsi à mes petits neveux que j'adore : Aghiles, Ilias, Rayane, Anna et Dylane. Aucune dédicace ne serait exprimée assez profondément de ce que je ressens envers vous. Je vous dirais tout simplement un grand merci, je vous aime.

À ma meilleure amie, et binôme Celia

Je ne peux trouver les mots justes et sincères pour t'exprimer mon affection et mes pensées, tu es pour moi ma sœur sur qui je peux compter. En témoignage de l'amitié qui nous uni et des souvenirs de tous les moments que nous avons passé ensemble, je te dédie ce travail et je te souhaite une vie pleine de santé et bonheur.

À mes très chers amis(es)

En témoignage de l'amitié sincère qui nous a lié(e)s et des bons moments passés ensemble je dédie ce travail à Kahina et Hayet, et à tous (tes) mes amis(es), en vous souhaitant un avenir radieux et plein de bonnes promesses.

CYLIA BELKAID

Résumé

Notre travail s'inscrit dans le domaine scientifique de la recherche d'information temporelle, et s'intéresse particulièrement à la recherche de microblogs dans la plateforme (numérique) de microblogging Twitter.

Les plateformes (numériques) de microblogging représentent un modèle de réseau social. À travers ces plates-formes, les utilisateurs (ou bloggeurs) publient et s'échangent des publications (ou posts) appelés microblogs. Les microblogs sont des messages courts à travers lesquels les bloggeurs publient des informations sur différents sujets d'actualités, et/ou des opinions sur ces sujets. L'information portée par un microblog est d'autant plus intéressante qu'elle provient de sources (bloggeur) fiables et pertinentes, mais aussi qu'elle est fraîche et récente.

La recherche d'information dans Twitter a pour objectif de retrouver, parmi l'ensemble des microblogs publiés (encore appelés *tweets*), ceux qui portent sur un sujet (ou thématique) donné (i.e. On parle de pertinence thématique), et qui ont été publié par des bloggeurs populaires et/ou experts (on parle de pertinence sociale). La recherche d'information temporelle intègre en plus la dimension temporelle (fraicheur et récence) aux dimensions thématique et sociale dans la recherche de microblogs.

C'est dans ce contexte de la recherche d'information temporelle dans les microblogs que se situent nos travaux. Plus particulièrement, afin de prendre en compte l'aspect temporel dans la recherche des microblogs pertinents pour une requête donnée, nous proposons une approche de recherche d'information qui intègre la temporalité dans le calcul de la pertinence d'un microblog. L'approche proposée est une amélioration d'une approche existante de recherche d'information sociale dans Twitter

Mots-clés : Recherche d'information, temporalité, microblogs, Twitter.

Table des matières

Introduction générale

Chapitre I : Généralités sur la recherche d'information.

Chapitre II : Recherche d'information temporelle dans Twitter

Chapitre III : Contribution à la RI dans Twitter

Chapitre IV : Implémentation et test

Conclusion générale

Bibliographie

Annexe

Liste des figures

CHAPITRE I

CHAPITRE II

CHAPITRE IV

Liste des tableaux

CHAPITRE IV

Introduction générale

Contexte et problématique

Avec les progrès techniques du web et des capacités de stockage et d'échanges sur internet, les réseaux sociaux (en particulier Twitter) connaissent une explosion en termes de volume d'informations produites et manipulé, et du nombre d'utilisateurs à travers le monde.

Le succès du Twitter a atteint un niveau sans précédent. Actuellement Twitter est considéré comme une source géante d'informations, avec quelques 320 million d'utilisateurs actifs et plus de 500 million de tweets qui sont publiés chaque jour. Vu ce volume important d'informations qui y circule, Twitter est devenu une source d'information inestimable. Cependant, retrouver l'information pertinente dans cette source gigantesque et évolutive au fil du temps, nécessite des techniques et des outils de recherche d'information spécifiques.

La recherche d'information(RI) dans Twitter, s'intéressent à la définition et a la mise en œuvre d'approches et techniques permettant de retrouver des informations pertinentes, concises et précises sur un sujet actuel, mais aussi de récupérer en temps réel, des informations sur un évènement qui vient de se produire.

Dans notre travail, nous nous intéressons à la recherche d'information temporelle dans Twitter.

Plus particulièrement, nous proposons une approche de recherche d'information qui intègre la temporalité dans le calcul de la pertinence d'un microblog. L'approche proposée est une amélioration d'une approche existante de recherche d'information sociale dans Twitter.

Organisation du mémoire

Le présent mémoire s'articule autour de quatre chapitres :

Chapitre 1 : *Généralités sur la recherche d'information*, dans ce chapitre nous présentons les concepts généraux de la recherche d'information, puis nous introduisons les spécificités de la recherche d'information dans Twitter.

Chapitre 2 : *Recherche d'information temporelle dans Twitter*, nous présentons différentes approches qui exploitent le temps dans la recherche d'information d'une manière générale, ensuite nous passerons en revue les approches exploitant ce facteur dans la recherche de microblogs d'une manière particulière.

Chapitre 3 : *Contribution de la RI temporelle dans Twitter*, dans ce chapitre nous présentons notre contribution en recherche temporelle dans Twitter. Notre approche est une amélioration d'une approche existante qui est aussi détaillée dans ce chapitre.

Chapitre 4: *Implémentation et tests*. Ce dernier chapitre présente les détails d'implémentation et de mise en œuvre de notre approche, ainsi que les résultats de son évaluation.

Enfin, nous terminons notre mémoire par une conclusion et des perspectives.

Chapitre I: *Généralités sur la recherche d'information.*

1. Introduction

De nos jours, l'information joue un rôle très important dans notre quotidien. Cependant le développement de l'internet et des nouvelles technologies, ainsi la naissance des réseaux sociaux, ont conduit à la production d'un nombre élevé d'informations. A ce stade il est difficile aux utilisateurs, de localiser leurs besoins dans cette masse d'informations, ce qui a provoqué un énorme fossé entre ce qu'ils cherchent et ce qu'ils trouvent.

Face à ce problème, est né la nécessité de mettre en place des systèmes et mécanismes facilitant l'accès aux informations, appelés systèmes de recherche d'information(SRI). L'objectif des SRI, étant de retrouver des documents susceptibles de répondre, au mieux à un besoin en informations d'un utilisateur, exprimé sous forme de requête.

Dans ce chapitre, nous commençons par définir la recherche d'information (RI) et les systèmes de recherche d'information(SRI), nous détaillerons ensuite leurs concepts de base, enfin nous intéresserons à la recherche d'information dans Twitter.

2. Recherche d'information et systèmes de recherche d'information

2.1. Définition de la RI

La RI est une démarche systématique de l'informatique qui s'intéresse à l'acquisition, l'organisation, le stockage, la recherche et la sélection d'informations [Salton, 1968]. La finalité de la RI, est de localiser et délivrer dans une masse de documents existante, un document qui répondent au besoin informationnel de l'utilisateur, exprimé sous forme de requête.

2.2. Définition d'un SRI

Un SRI est un système informatique, composé d'un ensemble de programmes, dont la finalité est de retrouver, dans une collection de documents préalablement enregistrés, les informations (documents) pertinentes, répondant au mieux au besoin de l'utilisateur exprimé sous forme de requête [1].

2.3. Concepts de base d'un SRI

La définition d'un SRI fait ressortir les quatre concepts clés suivants :

- ✓ **Document :** le document représente l'information élémentaire exploitable par le SRI. Un document est représenté sous différents formats : un texte, une page web, une image, une vidéo…

  Un texte peut être structuré (HTML, XML,…), ou non structuré (texte plat).

  Nous focalisons dans la suite de notre mémoire, sur les documents textuels non structurés.

  L'ensemble des documents sur lesquels porte une recherche, forme une collection de documents.

- ✓ **Besoin en information :** cette notion est souvent assimilée au besoin de l'utilisateur. Il existe trois types de besoin en information [Ingwersen, 1992]:
  - Besoin vérificatif : c'est à l'utilisateur de vérifier une information avec des données connues, dont il sait comment y accéder. Le besoin ici est plutôt précis.
  - Besoin thématique connu : l'utilisateur cherche à trouver de nouvelles informations, et à compléter des connaissances dans un sujet connu. Le besoin peut être exprimé de façon incomplète, c'est-à-dire l'utilisateur n'énonce pas nécessairement tout ce qu'il sait dans sa requête.
  - Besoin thématique inconnu: l'utilisateur cherche de nouvelles informations dans un sujet qui ne lui est pas familier, ce besoin est toujours exprimé d'une façon incomplète.
- ✓ **Requête:** la requête est l'expression du besoin en information de l'utilisateur. Elle est exprimée en langage de requête qui peut être naturel, graphique, ou booléen. La requête représente l'interface entre l'utilisateur et le SRI.
- ✓ **Pertinence:** la pertinence est une notion fondamentale dans le domaine de la RI. Elle définit le degré de correspondance entre un document et une requête [Borlund, 1998]. Cette correspondance peut être considérée de point de vue utilisateur on parle alors de pertinence utilisateur, ou du point de vue système on parle alors de pertinence système.

- Pertinence système : est une mesure d'évaluation par le SRI de la similarité entre le contenu des documents vis-à-vis de la requête [Boughanem et al., 2008].
- Pertinence utilisateur : c'est une mesure subjective, qui représente la satisfaction de l'utilisateur vis-à-vis des documents retournés par le système.

2.4. Processus en U de recherche d'information

Le fonctionnement d'un SRI est résumé, à travers le processus de recherche appelé processus en U de la RI illustré en figure ci-dessous :

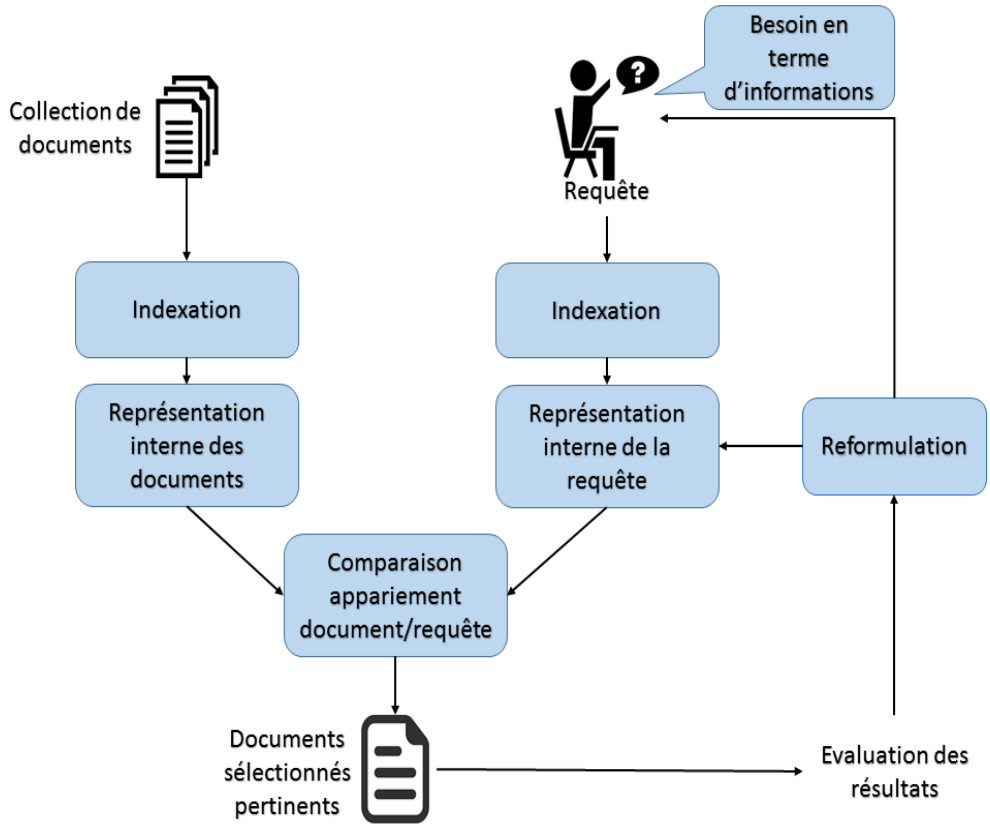


Figure I-1 : *Processus en U de RI.*

En RI, l'utilisateur interroge le SRI à l'aide d'une requête. Ce dernier lui renvoi l'ensemble des documents censés correspondre au mieux à la requête pour cela, le SRI

utilise trois principales phases : l'indexation, l'appariement document-requête et la reformulation du besoin en information.

2.4.1. La phase d'indexation

L'indexation vise à transformer les documents(ou requête), en substituts ou en descripteurs capables de représenter leurs contenus. Ces descripteurs forment un langage d'indexation, représentés selon une structure basée sur un ensemble de mots clés [Moulahi, 2016].

L'objectif de l'indexation consiste à détecter les termes les plus représentatifs, et à créer une représentation interne (index) des documents, pour faciliter la recherche.

Différents modes d'indexations existent en RI :

- ✓ **Indexation manuelle :** lors de l'indexation manuelle, un expert dans le domaine se charge de définir les mots-clés représentatifs du contenu du document. l'indexation manuelle assure une meilleure précision de recherche en réponse à une requête utilisateur, mais le temps nécessaire à sa réalisation est très important.
- ✓ **L'indexation automatique :** l'indexation automatique est un processus entièrement automatisé, il repose sur des algorithmes pour extraire les termes caractéristiques du document.
- ✓ **L'indexation semi-automatique:** est une combinaison des deux approches d'indexations précédentes, où le choix final des termes à indexer revient à l'expert.

D'une manière générale, l'indexation automatique se fait en plusieurs étapes :

- **L'analyse lexicale (Tokénisation) :** consiste à découper le texte d'un document (ou d'une requête), en plusieurs unités lexicales représentant les termes d'index (Tokens) [Fox, 1992].
- **L'élimination des mots vides :** vise à éliminer les mots non porteurs de sens ou mots vides. Ces mots peuvent être des mots outils tels que : les déterminants (Le, La…), des prépositions (sur, contre…), comme ils peuvent être les mots les plus fréquents par exemple : si un mot apparait dans plus de 80% des documents, alors il est jugé non utile pour la recherche. L'élimination des mots peut se faire en

utilisant une liste prédéfinie de mots vides (anti-dictionnaire) dite stoplists, ou en écartant les mots trop fréquents ou trop rares dans la collection.

- **La normalisation :** permet de représenter les variantes morphologiques des termes, issus d'une même famille sous une forme normale. La normalisation ce base sur l'une des deux procédures :
  - La racinisation (troncature) : vise à supprimer les affixes pour avoir des mots sous une forme tronquée, commune à toutes les variantes morphologiques.
  - La lemmatisation : permet l'obtention d'une forme canonique à partir d'un mot, les verbes sont transformés à l'infinitif, les noms et les adjectifs… sont transformés en masculin singulier.
- **Le choix des descripteurs:** dans cette étape, il faut faire un choix sur les index qui représenteront les documents, afin d'assurer une moindre perte d'information sémantique.
- **Création de l'index:** un ensemble de structures de données sont créés lors du processus d'indexation, permettant un accès efficace aux documents tel le fichier inverse. Le fichier inverse associe les termes d'index aux documents qui les contiennent.

2.4.2. Appariement document-requête

Elle permet de mesurer la valeur de pertinence d'un document vis-à-vis d'une requête. Le SRI représente le document et la requête avec un même formalisme, puis il compare les deux représentations, afin d'obtenir un résultat qui détermine le degré de ressemblance du document avec la requête [Hammache, 2011].

Il existe deux types d'appariements :

- ✓ **Appariement exact:** les documents retournés respectent exactement la requête spécifiée avec des critères précis, ces documents sont non triés.
- ✓ **Appariement approché:** les documents retournés répondent à tout ou à une partie de la requête, ces documents sont triés selon un ordre de mesure. Cet ordre reflète le degré de pertinence document-requête.

2.4.3. Les différents modèles de RI

Un modèle en RI décrit le processus computationnel, par exemple comment les documents retournés sont ordonnés ? Et comment sont-ils stockés ? Comme il peut décrire le processus humain tel que le besoin d'information, l'interaction. Son rôle le plus important étant de fournir un cadre théorique pour la modélisation de la mesure de pertinence.

Ces modèles sont divisés en deux catégories : on a les modèles exacts qui retournent les documents répondant exactement à la requête (modèle booléen), et les modèles approchés qui retournent des documents, qui répondent à tout ou à une partie de la requête (vectoriel, probabiliste) [Baeza-Yates, 1999].

Un modèle de RI est définit par un quadruplet (D, Q, F, R (Q, d)) ou :

D : est l'ensemble de documents.

Q : est l'ensemble de requêtes.

F : est le schéma du modèle théorique de représentation des documents et des requêtes.

R (Q, d) est la fonction de pertinence du document d à la requête Q.

2.4.3.1. Le modèle booléen

Dans ce modèle, un document d_i est représenté par un ensemble de termes descriptifs. Une requête Q est une expression booléenne, composée de mots-clés reliés par des opérateurs logique (AND, OR, NOT).

Pour évaluer la pertinence d'un document pour une requête, ce modèle se base sur la présence ou l'absence des termes de la requête dans les documents, en utilisant la fonction booléenne RSV (d_i, Q), qui calcule la mesure de pertinence document-requête comme suit :

$$\text{RSV}(d_i, Q) = \begin{cases} 1 \; si \; d_i \; contient \; les \; expressions \; booléennes \; décrites \; par \; Q \\ 0 \; sinon \end{cases}$$

Malgré la large utilisation de ce modèle, il présente un certain nombre de faiblesses :

- ✓ Formulation difficile de la requête : elle n'est pas évidente pour beaucoup d'utilisateurs, vu qu'elle nécessite une connaissance des opérateurs booléens.

- ✓ Les documents sélectionnés ne sont pas ordonnés.

2.4.3.2. Le modèle vectoriel

Est un modèle algébrique, introduit par [Salton, 1968] où les documents et les requêtes sont représentés, par des vecteurs de poids dans l'espace vectoriel des termes d'index.

Formellement :

Un document d_i est représenté par un vecteur de dimension n, $d_i = (w_{i1}, w_{i2}, \ldots\ldots w_{ij}, w_{in})$

Où : wij : le poids du terme t_j dans le document d_i.

Une requête Q est aussi représentée par un vecteur de poids de dimension n :

$Q = (w_{Q1}, w_{Q2}, \ldots, w_{Qj}, w_{Qn})$

Où : w_{Qj} : le poids du terme t_j dans la requête Q.

Dans ce modèle, chaque mot a un poids dans chaque document. Deux types de pondération sont utilisés : pondération locale et globale [Lv, 2010].

- ✓ **Pondération locale** (*tf* pour *t*erm *f*requency) : elle prend en compte les informations locales du terme qui ne dépendent que du document. Elle représente la fréquence d'occurrence du terme, dans le document elle est donnée par :

$$tf_{ij} = 1 + \log (f (t_i, d_j)) \quad (1)$$

  Où : $f(t_i, d_j)$ est la fréquence du terme t_i dans le document d_j.

- ✓ **Pondération globale** (*idf* pour *i*nvented *d*ocument *f*requency) : elle prend en compte les informations concernant le terme dans la collection, où le poids le plus important doit être assigné aux termes, qui apparaissent moins fréquemment dans la collection, elle est donnée par :

$$idf = \log (N/n_i) \quad (2)$$

Où : n_i est la fréquence en document du terme considéré, N est le nombre total de documents dans la collection.

La fonction de pondération combinant la pondération locale et globale est référencée sous le nom de la mesure tf*idf, qui donne une bonne approximation de l'importance du terme dans les collections de documents.

Dans le modèle vectoriel, la pertinence du document d_i vis-à-vis de la requête Q est mesurée par le degré de corrélation de leurs vecteurs correspondants, qui peut être exprimée par l'une des mesures suivantes :

- Le produit scalaire :

$$\text{RSV } (d_i,Q)= \sum_{j=1}^{n} w_{Qj} * w_{ij} \quad (3)$$

- La mesure de Dice :

$$\text{RSV } (Q,d_i)=\frac{2*\sum_{j=1}^{|n|} w_{Qj}*w_{ij}}{\sum_{j=1}^{|n|} w_{Qj^2}+\sum_{j=1}^{|n|} w_{ij^2}} \quad (4)$$

- La mesure de Jaccard :

$$\text{RSV } (Q,d_i)=\frac{\sum_{j=1}^{|n|} w_{Qj}*w_{ij}}{\sum_{j=1}^{|n|} w_{Qj^2}+\sum_{j=1}^{|n|} w_{ij^2}-\sum_{j=1}^{|n|} w_{Qj}*w_{ij}} \quad (5)$$

- La mesure de cosinus :

$$\text{RSV } (Q,d_i)=\frac{Q*d_i}{||Q||*||d_{i||}}=\frac{\sum_{j=1}^{|n|} w_{Qj}*w_{ij}}{\sqrt{\sum_{j=1}^{|n|} w_{Qj^2}\ \sum_{j=1}^{|n|} w_{ij^2}}} \quad (6)$$

2.4.3.3. Le modèle probabiliste

On a deux types de modèles probabilistes [Robertson, 1977] :

✓ **Modèle probabiliste de base :** permet de classer les documents, selon leurs probabilités de pertinence vis-à-vis d'une requête.

La fonction de classement est définit comme suit :

$$\text{RSV } (d_i, Q)=\frac{P(per\backslash d_i,Q)}{P(Nper\backslash d_i,Q)} \quad (7)$$

Où : P (per\d_i, Q) : la probabilité qu'un document d_i soit pertinent (*per*) vis-à-vis de la requête *Q*.

P (Nper\d_i, Q) : la probabilité qu'un document d_i soit non pertinent(*Nper*) vis-à-vis de la requête *Q*.

Cette fonction de classement permet de sélectionner, les documents ayant à la fois une forte probabilité, d'être pertinents et une faible probabilité d'être non pertinents à la requête.

- ✓ **Le modèle de langue:** Dans ce modèle, l'idée de base admet que la pertinence d'un document pour une requête est en rapport, avec la probabilité que la requête puisse être générée par le document.

  Formellement :

$$RSV(d_i, Q) = P(Q = (t_1, t_2, \ldots t_n)/Mdi) \quad (8)$$

  Où Mdi : le modèle de langue de document *di*.

  P (Q/Mdi) : la probabilité que la requête *Q* soit générée par Mdi.

2.4.4. La reformulation de la requête

Le but d'un SRI est d'offrir à l'utilisateur un résultat, satisfaisant à son besoin exprimé par une requête. La reformulation de la requête initiale, fera en sorte que le résultat retourné soit pertinent, on distingue trois méthodes [Hammache, 2011] :

- **La reformulation manuelle**: consiste à présenter à l'utilisateur une liste de documents, jugés pertinents en réponse à la requête initiale. C'est à l'utilisateur de sélectionner parmi les documents pertinents, ceux dont lesquels le système va extraire les termes à rajouter à la requête initiale, pour effectuer une nouvelle recherche.
- **La reformulation par réinjection de la pertinence** : elle nécessite l'intervention de l'utilisateur, pour sélectionner les documents pertinents, et les non pertinents à partir des résultats, issus de sa requête initiale. Ce jugement de pertinence de l'utilisateur est ensuite exploité par le SRI, pour reformuler la requête initiale, en modifiant le poids des termes qu'il contient, et/ou en ajoutant

de nouveaux termes utiles [Hammache, 2011]. La technique de réinjection de pertinence à l'origine a été mise dans le modèle vectoriel. [Rocchio, 1971] a proposé le modelé de reformulation de requête suivant :

$$Q_N = \alpha . Q_O + \beta . \frac{1}{|R|} \sum_{r \in R} r - \frac{1}{|R'|} \sum_{r' \in R'} r' \quad (9)$$

Où :

Q_N : est le vecteur de la nouvelle requête (reformulée).

Q_O : est le vecteur de la requête originale.

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 : Sont des paramètres de la reformulation.

- **La réinjection par pseudo feedback :** nommée aussi pseudo-réinjection de pertinence, où les méthodes de reformulation sont effectuées de manière automatique. Les documents les mieux classés (les premiers) sont considérés comme pertinents, le système alors utilise ces documents pour reformuler la requête.

  La réinjection automatique de la requête est exprimée par la formule suivante :

$$Q_N = \alpha . Q_O + \beta . \frac{1}{|R|} \sum_{r \in R} r \quad (10)$$

2.5. Evaluation des SRI

L'évaluation d'un SRI permet de vérifier l'efficacité des modèles mis en œuvre pour l'identification des documents pertinents.

2.5.1. Corpus de tests

L'évaluation d'un système se fait à l'aide d'un corpus de test qui contient :

- Un ensemble de documents
- Un ensemble de requêtes
- La liste de documents pertinents pour chaque requête.

Pour qu'un corpus soit significatif, il faut qu'il possède un nombre assez élevé de documents.

Différentes collections de tests sont utilisées en la recherche d'information, parmi elles nous citons :

2.5.1.1. Les collections TREC (Text REtrieval Conference)

Le projet TREC est un programme international initié au début des années 90, qui offre des moyens homogènes d'évaluation, des systèmes de recherche d'information. L'objectif du projet TREC est de proposer une plate-forme qui réunit des collections de tests, des tâches spécifiques et des protocoles d'évaluation pour chaque tâche afin de mesurer, les différentes stratégies de recherche.

Les différents éléments qui constituent le projet TREC sont :

- ✓ Les tâches : les tâches proposées changent d'une année à une autre, elles reflètent l'intérêt des chercheurs et les besoins réels : la RI ad-hoc (tache classique de RI, qui consiste à soumettre des requêtes sur une collection statique), la RI dans le web, la RI médicale, RI dans les microblogs, RI temporelle…etc.
- ✓ Les participants : ce sont les différents groupes de personnes qui ont participés au projet TREC, issus de différents pays.
- ✓ Structure et principe de construction de la collection : un document TREC est présenté sous un format SGML, il est identifié par un numéro et décrit par un auteur, une date de production et un contenu textuel. Une requête TREC est aussi identifiée par un numéro, elle est décrite par une description sur les caractéristiques des documents pertinents, associé à la requête.

2.5.2. Mesures d'évaluation

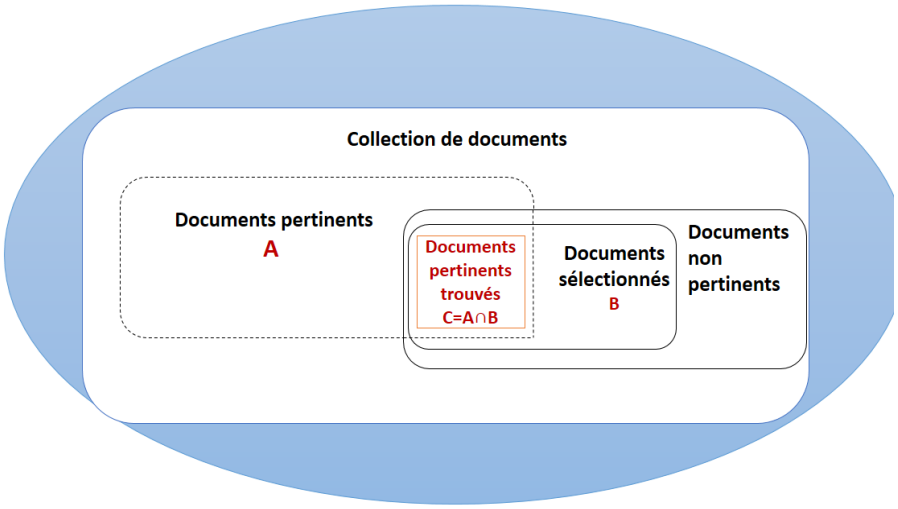
L'évaluation correcte de la capacité d'un SRI, est de retourner des documents satisfaisants l'utilisateur, se fait selon plusieurs facteurs dont les plus utilisés sont : le rappel et la précision.

- ✓ **Rappel:** est la capacité d'un système à sélectionner tous les documents pertinents de la collection.

  Où : rappel $= \frac{\text{Nombre de documents pertinents selectionnés}}{\text{Nombre total de documents pertinents}}$

✓ **Précision:** est la capacité d'un système à nous sélectionner que les documents pertinents.

Où : précision=$\frac{\text{Nombre de documents pertinents selectionnés}}{\text{Nombre total de documents selectionnés}}$



Rappel =$\frac{|C|}{|A|}$

Précision =$\frac{|C|}{|B|}$

Figure I-2 : *Partition d'une collection pour une requête*

Les deux métriques rappel et précision, ne sont pas indépendantes il y a une forte relation entre elles quand l'une augmente l'autre diminue. Ainsi pour un système on a une courbe de précision-rappel qui a en général l'aspect suivant :

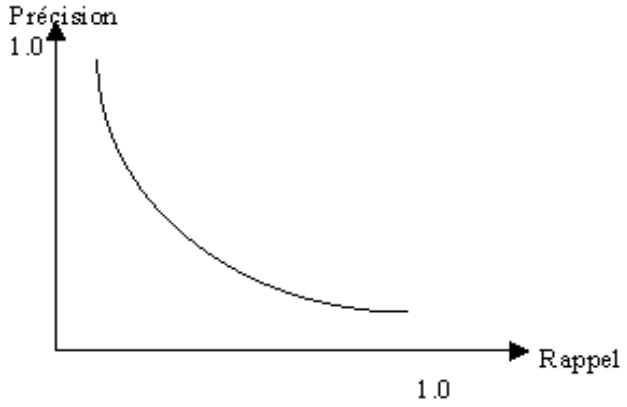


Figure I-3: *Courbe de rappel-précision*

Des mesures complémentaires au rappel et précision ont été définies, il s'agit de bruit et de silence :

- ✓ Bruit : est une notion complémentaire à la précision, définie par B=1-P, où *P* est la précision du SRI.
- ✓ Silence : est notion complémentaire au rappel, définie par S= 1-R, où *R* est le rappel du SRI.

Le comportement d'un système peut varier en faveur de la précision où en faveur du rappel, ainsi un bon SRI est celui qui est capable de renvoyer les bons documents en évitant de retourner les documents non pertinents(en faisant le moindre bruit possible), et de restituer le maximum de documents pertinents (silence de système).

3. 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 index de microblogs, et sélectionner ceux qui sont pertinents [Ounis, 2010], dans cette partie on s'intéressera à la RI dans les tweets. On commencera d'abord par présenter Twitter, ensuite on parlera de la RI dans Twitter.

3.1. Réseau social Twitter

L'outil Twitter a été créé par la société américaine Odeo en 2006, il s'agit d'un service de microblogging qui permet à un utilisateur, d'envoyer gratuitement de brefs messages appelés « tweets » (gazouillis). Ces messages sont limités à 140 caractères [5].

A la base, twitter a été créé pour permettre aux amis, familles, collaborateurs de communiquer, d'exprimer et de partager les histoires, les nouvelles, leurs opinions sur différents sujets.

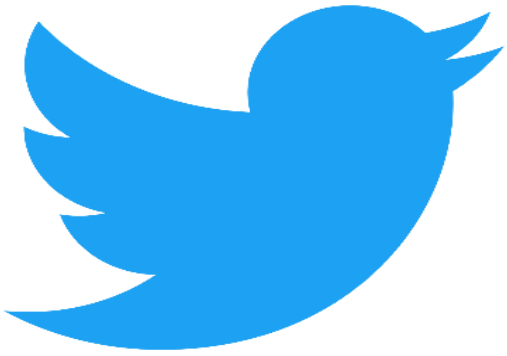


Figure I-4: *Logo de twitter.*

3.1.1. Fonctionnement de Twitter

La figure I.5 montre l'interface de twitter, qui est composée de plusieurs sections telles la section tweet où l'utilisateur peut voir le flux de ses tweets ou ceux de ses amis, la section tendance qui contient les 10 sujets les plus populaires, la section suggestions et la section concernant son propre compte.

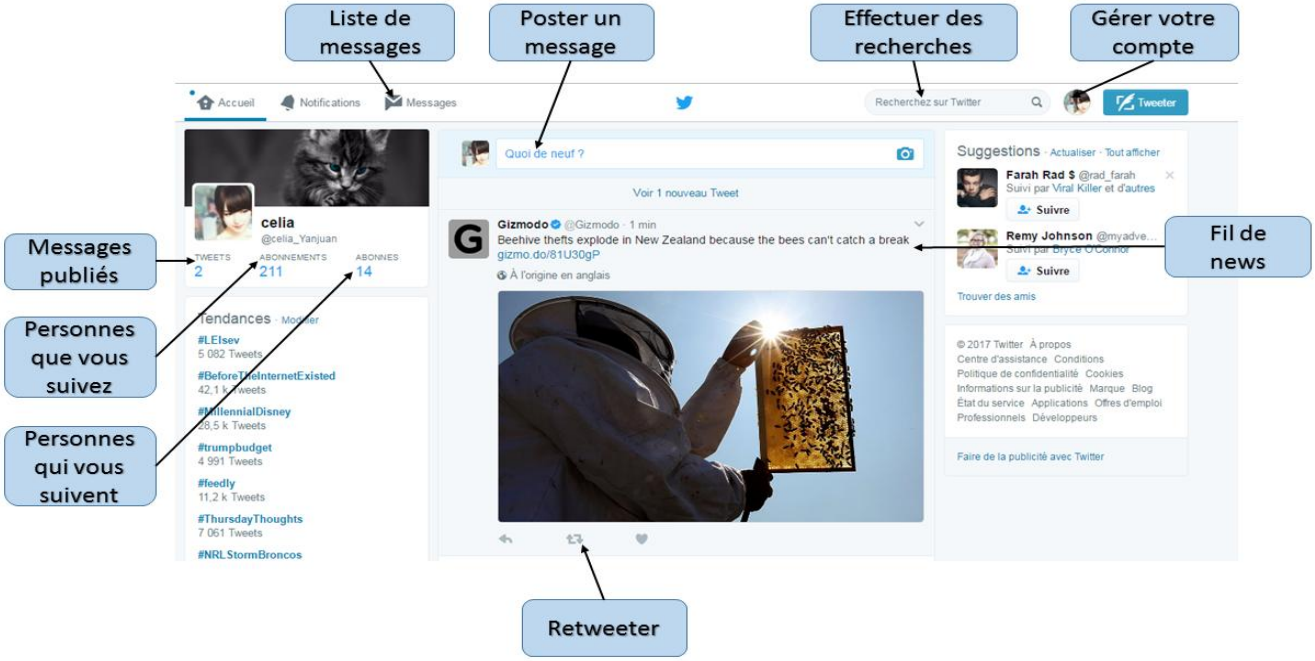


Figure I-5: *L'interface d'accueil de Twitter.*

3.1.2. Vocabulaire de Twitter

Le vocabulaire de Twitter est définit comme suit **[2] :**

- ✓ Twitt où tweet : message publié sur twitter.
- ✓ Username : identifiant qui caractérise un utilisateur sur twitter.
- ✓ Follower où abonnés : sont les personnes qui suivent votre actualités.
- ✓ Following où abonnements : correspond aux comptes twitter que vous suivez
- ✓ Timeline : ensemble des tweets générés par les utilisateurs auxquels on est abonné.
- ✓ DM Direct message : il s'agit des messages tweets envoyé de manière privée à un abonné.

- ✓ RT (retweet) : est un message déjà publié par une première personne et republié par une autre personne, le message est constitué comme tel : RT@auteurtweetmessage.
- ✓ Le signe @ : lorsqu'un utilisateur cite ou mentionne un compte, il le fait précédé par un @, ce qui permet de faire savoir à son destinataire que vous lui adressez un message.
- ✓ Le Hashtag où # : les hashtags vous permettent de découvrir de nouvelles personne qui parlent ou s'intéressent aux mêmes sujet que vous, il fonctionne comme étant un mot-clé ou un tag, il permet de définir d'une manière générale le sujet principal du tweet. Donc le faite d'ajouter un # à un mot dans un tweet ce terme devient un mot-clé.

3.1.3. Le réseau social d'information

A la différence des autres réseaux sociaux, twitter se positionne par la relation sociale d'abonnement. Cette association permet aux utilisateurs, d'exprimer leurs intérêts pour les articles d'un autre microblogueur, mais Twitter ne se limite pas aux relations d'abonnements et aux blogueurs, il inclut également les acteurs et les données qui interagissent entre eux, dans les deux contextes de publication et l'utilisation des articles [Lamdjeb Ben Jabeur, 2011].

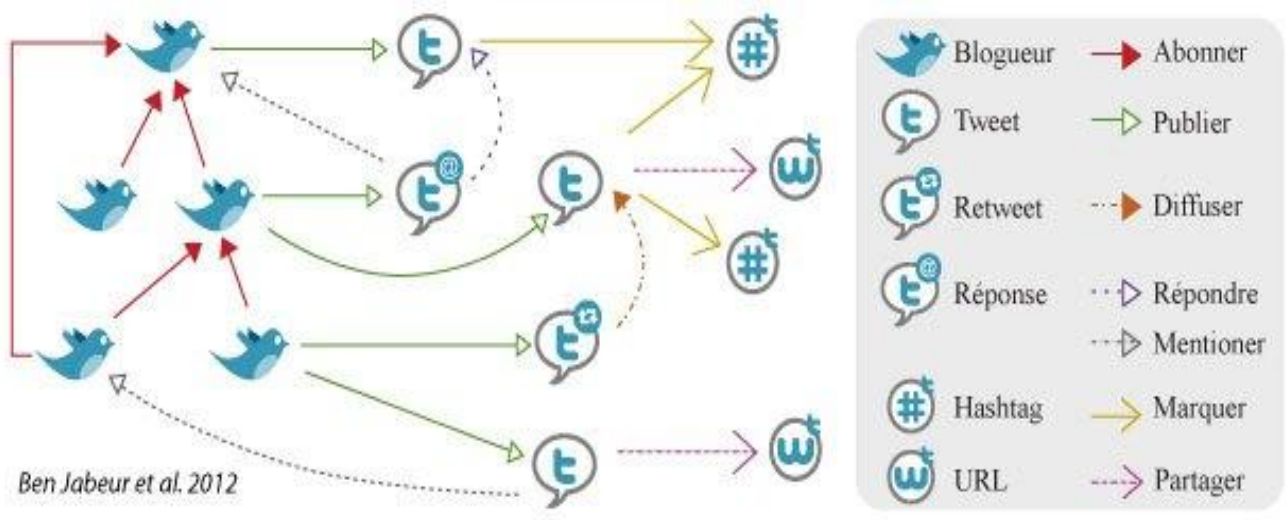


Figure I-6: *Réseau social d'information de Twitter* [Lamdjeb Ben Jabeur, 2011].

3.1.4. Spécificité et avantage de twitter

Twitter permet une communication d'actualité et les échanges y sont plutôt qualitatifs, de nombreux journalistes l'utilisent afin de se tenir informés, de recevoir une information immédiate, il permet également de suivre en temps réel un débat ou un séminaire et de faire des commentaires, les personnes qui ne sont pas sur place peuvent ainsi participer ou du moins accéder au contenu des conférences [3].

Parmi ses avantages :

- Permet à l'utilisateur d'obtenir, filtrer, trier, échanger et classer le flot d'un grand nombre d'information et de données, donc twitter peut être vu comme étant un outil d'acquisition de gestion des connaissances.
- Twitter peut être vu aussi comme un objet d'étude ou support d'étude.
- Twitter demeure un outil simple et efficace et potentiellement riche en termes de partage d'informations.
- La création du compte est gratuite.

3.2. La recherche d'information dans Twitter

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, le trafic routier…etc.
- Des informations sociales : telle la recherche d'autres utilisateurs.
- Des informations sur des sujets spécifiques.

Il existe plusieurs facteurs qui reflètent la pertinence dans la recherche de microblogs, que nous détaillerons dans (3.2.1).

3.2.1. Etude des facteurs de pertinences

Nous présentons ici les différents facteurs de pertinence à prendre en compte dans la conception des approches de recherche de microblogs [Damak, 2014] :

Nous utiliserons les notations suivantes dans la suite :

- *q*: la requête composée par des mots-clés « topic » et caractérisé par une date.

- C_q : le corpus des tweets publiés avant la date de la requête.
- T_q : l'ensemble des tweets restitués par un moteur de recherche donné calculant la pertinence par rapport à la requête q.
- t : est un tweet qui appartient à T_q et sur lequel on applique le facteur de pertinence.

✓ **Facteurs de pertinence liés au contenu :** consiste à étudier les quatre facteurs relatifs au contenu qui sont : 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 la popularité d'un tweet dans *Tq*, un tweet est populaire si seulement si on trouve plusieurs autres tweets ayants un contenu similaire. On utilise le modèle vectoriel qui calcule la similarité entre chaque pair de tweet $sim\ (t_i, t_j)$ où t_i représente le vecteur.

  Ce facteur de pertinence est calculé de la manière suivante :

$$f_1\ (t_i, q) = \frac{\sum_{tj \in Tq, i \neq j} sim(ti,tj)}{|Tq|-1} \qquad (11)$$

- Longueur du tweet: est calculé on comptant le nombre de termes dans un tweet, on note l (t_i) qui est le nombre de terme dans un tweet t_i dans *Tq*.

$$f_2\ (t_i) = \frac{l(ti)}{\max_{tj \in Tq}\ l(tj)} \qquad (12)$$

- la correspondance exacte des termes : consiste à calculer le nombre de termes en commun entre t_i et q.

$$f_3\ (t_i, q) = \frac{nb(ti,q)}{max_{ti \in Tq}\ nb(tj,q)} \qquad (13)$$

- La qualité du langage : représente la proportion des termes qui existent dans un dictionnaire, par rapport à tous les termes du tweet t_i, la valeur retournée est en binaire : 1 si le terme existe dans le dictionnaire 0 sinon.

$$f_4\ (t_i) = \frac{\sum_{term \in ti} dic(term)}{l(ti)} \qquad (14)$$

✓ **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 *Cq*.

✓ Facteurs de pertinence basés sur les hashtags :

- Présence de hashtag : retourne une valeur binaire : 1 si le tweet contient un hashtag, 0 sinon.
- Fréquence de hashtag de tweet : noté par la fréquence d'un hashtag dans le corpus Cq par freq(h).

$$f_5(t_i) = \sum_{h \in ti} freq(h) \qquad (15)$$

- Hashtag de la requête dans le tweet : qui calcule le nombre de termes d'une requête q qui apparaissent sous forme d'un hashtag dans un tweet t_i.

✓ **Facteurs de pertinence relatifs à la qualité des tweets :**

La prise en compte du temps est primordiale dans la recherche de microblogs car le microblogging est un système temps-réel qui incite les utilisateurs à exprimer leurs opinions et discuter en temps-réel. Dans cette section on doit étudier les deux critères particularisants les tweets :

- Retweet : le principe consiste à étudier les messages si ils ont été marqués comme étant retweet, en les voyant précédés par RT un exemple : si un utilisateur aime ce que l'un de ses amis à publier, il va probablement le commenter et le partager, dans ce cas le nouveau message sera précédé 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 t_i et la date de soumission de la requête q mesurée en seconde.

- ✓ **Facteur de pertinence basé sur la popularité de l'auteur :** deux facteurs de pertinence spécifiques aux auteurs de microblogs, ont été définit :
  - Nombre de tweets de l'auteur : l'objectif de ce facteur de pertinence, est de valoriser les tweets publiés par les auteurs actifs par rapport aux tweets publiés par des auteurs moins actifs. On note par a (t_i) l'auteur de tweet t_i, et $N\ (a\ (t_i))$ le nombre de tweets publiés par l'auteur de tweet t_i, dans le corpus C_q.

$$f\ (t_i) = N\ (a\ (t_i)) \quad (16)$$

  - Nombre de citation de l'auteur : plus un auteur est mentionné, plus il est populaire. $M\ (a\ (t_i))$ indique combien de fois un auteur de tweet t_i a été mentionné dans le corpus C_q.

$$f\ (t_i) = M\ (a\ (t_i)) \quad (17)$$

3.2.2. Evaluation de la RI dans les microblogs

La RI dans les microblogs, se fait avec la mise en place de la tache microblog, dans la campagne d'évaluation TREC [Damak, 2014].

3.2.2.1. La tâche microblog de TREC

Pour un moteur de recherche il s'agit de fournir des tweets les plus pertinents et aussi les plus récents en réponse à la requête exprimée sous forme de mots clés. Les résultats doivent être publiés avant la date de soumission de requête.

La collection de test tweets2011 comprend :

- 16 millions de tweets exprimés dans différentes langues et publiés sur twitter en 2011, ou chaque tweet est caractérisé par un identifiant, son auteur, et sa date de publication.
- 49 topics ou chacun est composé de plusieurs balises. La balise *title* décrit le besoin exprimé à un moment donné (querytime).
- Jugement de pertinence (qrels) associé au 49 topics. Un tweet peut être non pertinent, moyennement pertinent et hautement pertinent.

La collection de test tweets2012 comprend :

- Le même fond de tweets que celui de 2011.
- 60 nouvelles requêtes avec leurs jugements de pertinence.

La collection de test tweets2013 comprend :

- Une nouvelle collection de 240 millions de tweets qui accessible uniquement à travers une API.
- 60 nouvelles requêtes avec leurs jugements de pertinence.

3.2.2.2. Les mesures d'évaluation

Deux mesures ont été considérées dans les trois versions de la tâche [Lamjed Ben Jabeur, 2012] :

- La précision p@30: 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 : 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.

4. Conclusion

Au cours de ce chapitre nous avons passé en revue les concepts principaux de la recherche d'information et des SRI, ensuite nous avons introduit Twitter et ses spécificités ainsi son fonctionnement, puis nous avons introduit la RI dans Twitter.

Dans le chapitre suivant, nous nous intéressons à la RI temporelle en générale, et à la RI temporelle dans Twitter en particulier.

Chapitre II: *Recherche d'information temporelle dans Twitter.*

1. Introduction

De nos jours, avec l'expansion des outils de communication et des réseaux sociaux tel que Twitter, des millions d'utilisateurs accèdent au web tous les jours pour rechercher différents types d'informations. Des études récentes ont soulignés la haute temporalité des informations publiées par Twitter, couvrant principalement les dernières nouvelles et les évènements de toutes sortes (politique, économique, culturelle…). Ces informations sont utiles pour de nombreuses applications de traitement du langage, comme la recherche et l'extraction d'informations temporelles à partir du texte.

Dans ce chapitre, nous nous intéressons à la recherche d'information temporelle dans twitter, nous commençons d'abord par quelques généralités sur la RI temporelle, puis nous abordons la recherche d'information temporelle dans twitter.

2. La recherche d'information temporelle

2.1. Objectif

La RI temporelle est un nouveau domaine de recherche, dont l'objectif principal est l'amélioration des modèles de RI classiques, par l'exploitation des informations temporelles qui peuvent exister dans les requêtes et les documents d'une façon générale. Les modèles de RI temporels combinent la notion de pertinence traditionnelle (thématique) avec la dimension temporelle pour proposer aux utilisateurs des documents récents, qui répondent à leurs requêtes.

Le temps prend de plus en plus d'intérêts dans le domaine de recherche d'information, car la pertinence de la recherche dépend du temps.

Le temps est généralement représenté par la date de création des documents, la date de soumission d'une requête, ou par les expressions temporelles contenues dans les documents. Dans cette section nous présentons différentes approches qui exploitent le temps au niveau de la recherche d'information (document, requête, modèle de recherche).

2.2. Utilisation du temps en RI

En ce qui concerne la recherche d'information basée sur le temps et la pertinence, nous pouvons distinguer trois principales approches : le temps au niveau de la requête, le temps dans les documents et le temps en tant que facteur dans le modèle de recherche. La figure II.1 résume les approches d'utilisation du temps en RI [Moulahi, 2016].

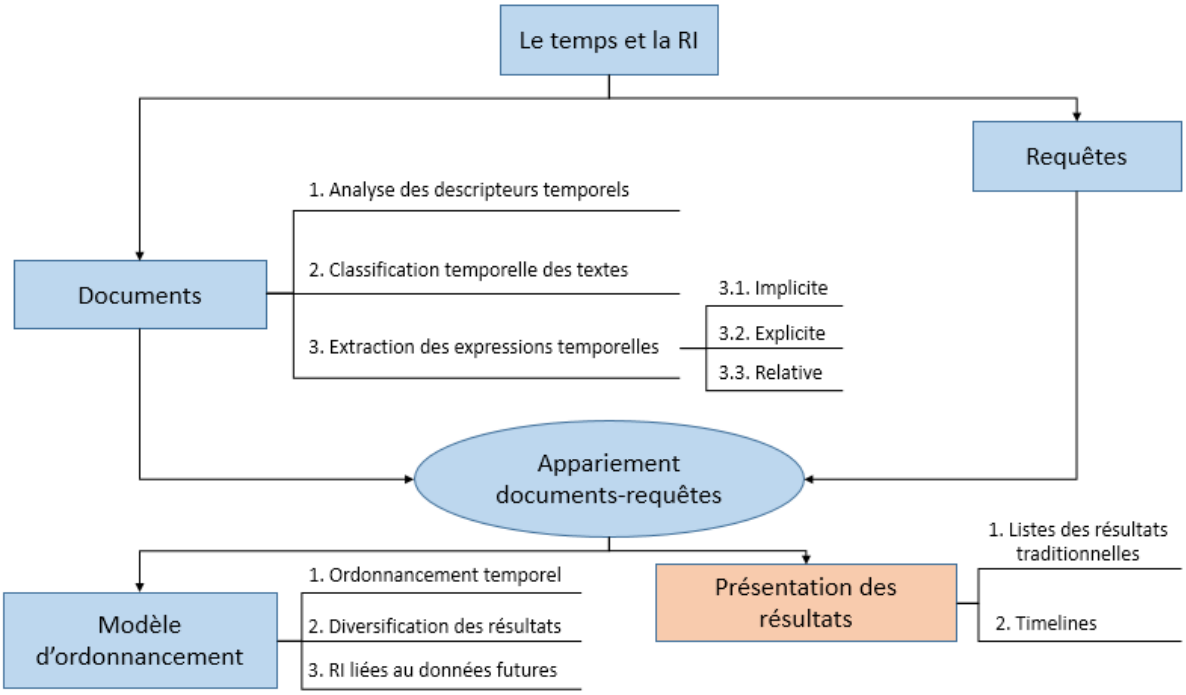


***Figure II-1**: Le temps au niveau de RI suivant 3 niveaux (requête, document, modèles de RI) [Moulahi, 2016].*

2.2.1. Approches de RI basées sur le temps dans le document

Certains documents contiennent des informations temporelles. Ces informations sont définies comme étant des expressions langagières naturelles, qui se réfèrent directement au temps ou aux intervalles. Ces expressions transmettent non seulement les informations temporelles, mais servent également d'ancres pour localiser les évènements mentionnés dans le texte.

Trois catégories (ou occurrences) des expressions temporelles dans les documents sont identifiées :

- ✓ **Explicites :** ces expressions temporelles se réfèrent à un point précis dans le temps, où les faits sont relatés de façon claire et précise, et les événements sont

exposés tels qu'ils se sont passés. Ces expressions peuvent être des expressions de date, de temps, de durée, ou d'ensemble [Berrazega, 2012].

- **Date** : les expressions de date se rapportent à une période particulière basée sur le calendrier grégorien où l'unité de base sur laquelle s'appuie le calendrier est le jour. Exemple le « 02 mai 2017 ».
- **Le temps** : les expressions du temps désignent une subdivision particulière d'un jour, cela peut correspondre aux moments que nous mesurons sur une horloge, par exemple « à 9 heures du matin ».
- **Durée** : les expressions de durée se réfèrent à une période prolongée. Elles sont mesurées en utilisant des unités calendaires (année, mois, jour, etc.) ou des unités d'horloge (heures, minutes, secondes, etc.). Par exemple « l'examen a duré 3 heures de 9 heures jusqu'à 12 heures ».
- **Ensemble :** les expressions d'ensemble font référence à la régularité ou à la réapparition d'une éventualité, soit dans l'absolu, soit par rapport à une période de temps. Par exemple « deux fois la semaine ».

✓ **Implicites :** les expressions temporelles imprécises telles que « deux jours auparavant », « la semaine dernière »,… etc. c'est au lecteur de faire la déduction et l'interprétation pour arriver à dégager une information précise.

✓ **Relatives :** les expressions temporelles relatives, représentent des entités temporelles qui ne peuvent être ancrés dans une ligne de temps que par référence à une autre expression temporelle explicite ou implicite comme « hier », « vendredi »… etc.

Dans le but d'exploiter les descripteurs temporels au niveau du contenu du document, le défi principal est l'identification et l'extraction des expressions temporelles. Cette étape fait partie de la tâche des systèmes d'annotation temporelle. La première étape d'un système d'annotation consiste à segmenter le texte du document en un ensemble de phrases, dans la deuxième et troisième étape le système identifie les phrases et effectue leurs étiquetages morphosyntaxiques, puis reconnait les entités contenues dans le texte. Les expressions temporelles extraites sont enfin normalisées.

Le système HeidelTime **[4]** qui est un système multilingue, permet d'identifier et d'extraire les expressions temporelles.

Les expressions temporelles extraites sont essentielles, pour créer un système de recherche temporel. L'ajout du temps dans le système d'indexation permet de chercher efficacement les documents pertinents pour une période donnée [Nattiya, 2012].

2.2.2. Approches de RI basées sur le temps dans la requête

L'objectif principal est souvent de comprendre l'intention temporelle derrière les requêtes des utilisateurs, et d'adapter le modèle de recherche en fonction de cette requête. Le principal défi est d'identifier la période de temps à laquelle la requête se réfère et de faire face à l'ambiguïté temporelle de la requête.

Trois types de profils temporels de requêtes sont identifiés dans [Jones et Diaz, 2007] : requêtes atemporelles qui font référence à des sujets non sensibles au temps, requêtes temporellement non ambiguës qui font référence à une période de temps précise, et les requêtes temporellement ambiguës qui font référence à des périodes de temps imprécises.

Nous présentons deux principaux types de requêtes qui ont fait l'objet d'études approfondies :

- ✓ **Requêtes orientées récence :** sont des requêtes qui surviennent juste après les dernières nouvelles ou les événements les plus récents. [Lie et Croft, 2003] ont classé les requêtes en fonction de la distribution temporelle des documents sur une collection de requêtes de TREC. Dans le cas où la période pertinente des requêtes sensible au temps ne peut être déterminée, les auteurs suggèrent de calculer une probabilité p (q|t) pour chaque instant *t* et requête *q* en utilisant les modèles de vraisemblance.
- ✓ **Requêtes périodiques et rafales :** sont des requêtes qui sont soumises d'une façon récurrente dans les mêmes périodes de temps. Il existe trois classes qui regroupent les requêtes rafales (bursts): celles qui disparaissent complètement après une certaine période, les bursts sur des sujets existants et les bursts qui créent d'autres sujets.

Du point de vu de la RI temporelle, l'intégration du facteur temps dans les requêtes permet d'avoir des résultats plus pertinents et qui répondent aux besoins de l'utilisateur.

2.2.3. Approches de RI basées sur la pertinence temporelle

Deux critères de pertinence exploitent la dimension temporelle :

- **La récence (recency)** : ce critère favorise les documents publiés récemment. Elle peut correspondre à des requêtes qui retournent les documents les plus pertinents et récents. Ce critère est calculé comme la différence entre le temps de publication du document, et le temps de soumission de la requête.
- **La fraicheur d'information (freshness)** : elle est interprétée en fonction des requêtes. Prenons comme exemple les requêtes liées aux sujets d'actualités, la fraicheur concerne les documents qui traitent les nouvelles informations.

La figure II.2 montre comment le temps pourrait être utilisé au niveau de correspondance du document requête.

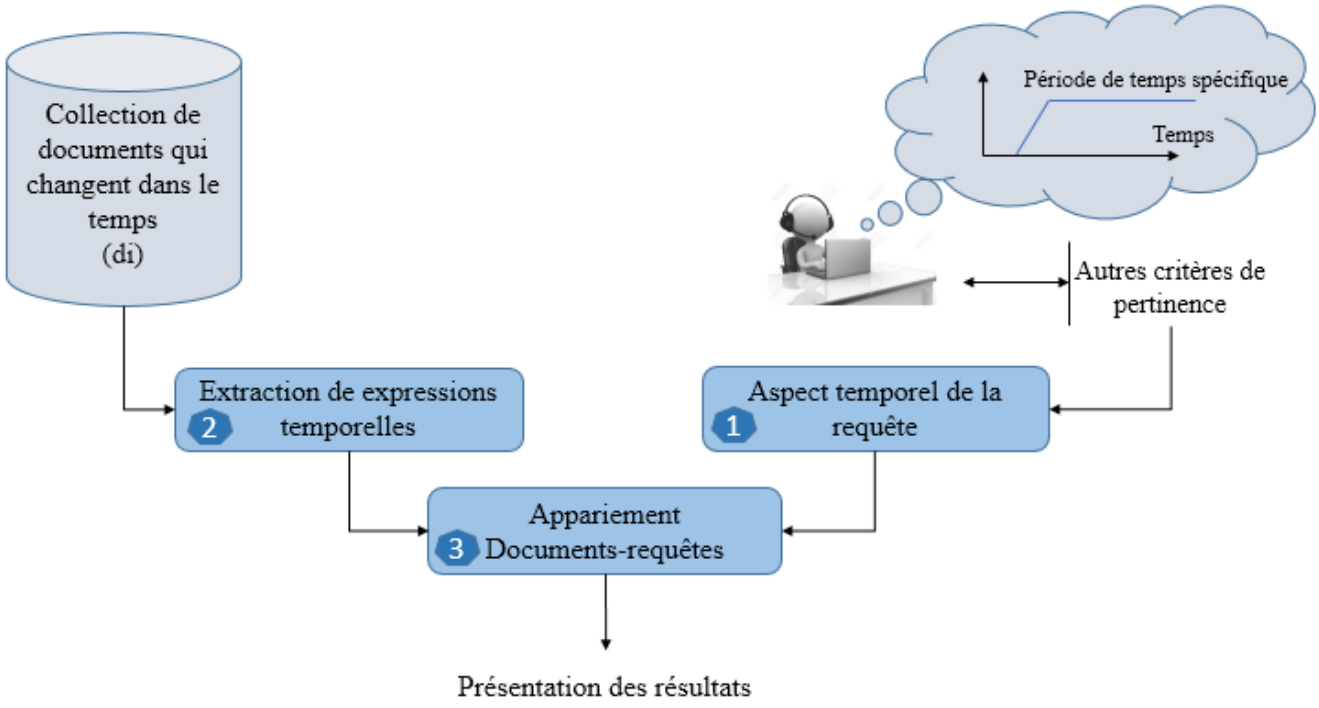


Figure II-2: *Processus général d'ordonnancement dans une approche de RI sensible au temps [Moulahi, 2016].*

L'approche la plus utilisée consiste à incorporer les expressions temporelles dans un modèle de recherche classique, pour classer les documents temporellement pertinents, en combinant la pertinence thématique avec les caractéristiques de pertinence temporelle [Lie et Croft, 2003].

[Lie et Croft, 2003] ont défini aussi un modèle de langue qui répond aux requêtes dont le besoin est temporel, les documents les plus récents ont eu le score le plus élevé :

$$p(d|T_d) = p(T_d) = \lambda\, e^{-\lambda(T_c - T_d)} \quad (18)$$

Où:

T_c: date la plus récente dans la collection de documents.

T_d : date de création du document.

L'évaluation de cette méthode, montre que les modèles temporels sont meilleurs que les modèles de langue classiques.

[Dakka et al., 2012] ont développé un modèle qui définit la pertinence comme une combinaison de la dimension thématique et temporelle représentés respectivement par les probabilités P(q|d) et P(t|q). Ce modèle identifie les intervalles de temps pour les requêtes sensibles au temps, et les utilise afin de favoriser les documents publiés dans ces périodes. La pertinence totale est donnée par :

$$P(d_t|q) = P(d, t|q) \propto P(q|d)\, P(t|q). \quad (19)$$

Où :

$\propto$: **[8]** désigne une proportionnalité (quand on peut passer de l'une à l'autre en multipliant ou en divisant la première par une même constante non nulle).

P (q|d) : est le modèle de vraisemblance sur le document *d*.

P (t|q) : c'est l'importance relative du temps *t* pour la requête *q*.

3. La recherche d'information temporelle dans twitter

La recherche temporelle dans Twitter, consiste à exploiter et à intégrer le facteur temps, comme facteur de pertinence dans les modèles de recherche.

L'utilisateur cherche à avoir l'information la plus récente, et pertinente par rapport à un besoin d'information. Plusieurs approches exploitent le facteur temps pour mettre au point des modèles de recherche dans les microblogs. Ces approches se basent sur plusieurs critères soit sur la fraicheur, soit sur la pertinence, ou bien sur le profil utilisateur. Dans cette section, nous citons quelques travaux qui ont été fait dans le but d'exploiter ces critères.

3.1. Approche basée sur la fraicheur

Cette approche vise à retourner les documents les plus récents en réponse aux besoins des utilisateurs.

Dans cette catégorie, nous citons les travaux de [Massoudi, 2011] où l'approche proposée permet de classer les résultats du plus ancien vers le plus récent, en favorisant les tweets les plus récents. Cette approche vise à calculer un score de fraicheur basée sur la différence temporelle entre la date de la soumission de la requête et la date de publication du document.

Premièrement [Massoudi, 2011] propose une approche d'expansion de requête, dans laquelle il suppose que tous les tweets qui contiennent au moins un des termes de la requête sont considérés comme des tweets pertinents pour l'expansion de requête.

Il construit un poids qu'il appelle score pour évaluer l'importance d'un terme *t* à la requête *q*, en utilisant le concept tf-idf comme suit :

$$\text{score}(t|q) := n_{cooccur}(t,q) * \log\left(\frac{|N|}{n_{tweet}(t,N)}\right) \quad (20)$$

Où

N : est la collection totale de tweets.

$n_{cooccur}(t,q)$: nombre de tweets dans lesquels le terme t correspond avec au moins un des termes de la requête $q_1, \ldots, q_r$ (*r* est le nombre de termes dans la requête d'origine).

$n_{tweet}(t,N)$: nombre de tweets dans la collection *N* qui contiennent le terme *t*.

La requête étendue q' peut être formulée comme suit:

$$p(t,q') = \frac{score(t,q)}{\sum_{t' \in k} score(t'|q)} \quad (21)$$

Où : *k* représente les meilleurs termes pour l'expansion de requête.

Deuxièmement L'auteur propose un modèle basé sur le temps pour l'expansion de requête, vu que les tweets récents contiennent le terme t qui est un aspect important, il l'intègre dans le score comme suit :

$$score(t|q, c) := \log\left(\frac{|N_c|}{n_{tweet}(t,N_c)}\right) * \sum_{d \in cooccur(t,q,c)} e^{-\beta(c-c_d)} \quad (22)$$

Où :

c : le moment où la requête de l'utilisateur a été exécuté.

c_d : est le moment de publication du tweet.

N_c : est la collection totale de tweets qui ont été affichés avant le temps *c*.

$n_{tweet}(t,N_c)$: est le nombre de tweets dans la collection *N* qui contient le terme *t*.

cooccur(t,q,c) : est le sous ensemble de tweets dans N_c, dans lequel le terme *t* correspond à au moins avec un des termes de la requête.

β: poids qui contrôle la contribution de chaque tweet au score de terme t en fonction de la date de publication.

La requête étendue q' basée sur le temps peut etre formulée comme suit :

$$p(t,q') = \frac{score(t|q,c)}{\sum_{t' \in k} score(t'|q,c)} \quad (23)$$

Cette méthode n'a pas donné de bons résultats, vu qu'elle favorise les termes d'expansion avec une coordination disproportionnée élevées avec l'un des termes de la requête, mais pas avec tous les termes de la requête.

3.2. Approches basées sur la pertinence

Les approches basées sur la pertinence visent à intégrer le facteur temps avec différentes manières, afin d'améliorer et de retourner des résultats plus pertinents en réponse à une requête.

3.2.1. Approche de Croft

[Croft et al., 2012] ont proposé une méthode de sélection des périodes de temps pour l'expansion de requête basée sur le Retweet, cette méthode intègre le facteur temps dans le modèle de langue.

Pour identifier la période de temps pertinente pour la requête, [Croft et al., 2012] ont besoin de *N* documents retournés. Les auteurs calculent d'abord P(t|RT, Q) et P(t|D, Q) comme décrit respectivement dans l'équation (24) et (25) :

$$P(t|RT, Q)=\frac{\#docs\ (t,RT,Q)}{\sum_{t\prime} \#docs(t',RT,Q)} \quad (24)$$

$$P(t|D, Q)=\frac{\#\text{docs}\ (t,D,Q)}{\sum_{t\prime} \#\text{docs}(t',D,Q)} \quad (25)$$

Où:

#docs(t, RT, Q) est le nombre de retweets affichés à l'instant *t* dans le top *N* documents retournés par la requête *Q*.

#docs(t, D, Q) est le nombre de documents publiés à l'instant *t* dans top *N* documents retournés par la requête *Q*.

En tenant compte de ces probabilités les auteurs définissent une fonction d'indicateur φ qui a la valeur 1 si P(t|RT, Q) est supérieure à P(t|D, Q) et 0 sinon, puis φ est normalisée pour tous les temps t et comme dans l'équation (26).

$$P(t|Q)= \frac{\varphi(t,Q)}{\sum_{t\prime} \varphi(t',Q)} \quad (26)$$

3.2.2. Approche de Willis

L'objectif des travaux de [Willis et al, 2012], est d'explorer différentes façons d'intégrer l'information temporelle pour améliorer la recherche dans les microblogs. Les auteurs proposent une approche d'expansion de requête basée sur le temps, qui s'intéresse à la récence et à la priorité temporelle, cette dernière effectue une recherche initiale et promeut les documents prévenant de périodes de temps avec une forte concentration pour des meilleurs résultats.

Premièrement [Willis et al., 2012] favorisent les termes qui ont une haute cooccurrence avec tous les termes de la requête comme suit :

$$\text{score}(w,Q)=\left(\frac{1}{|Q|}\sum_{q\in Q}\left(\sum_{\{D:q,w\in D\}} e^{-\beta(t_Q-t_D)}\right)^{-1}\right)^{-1} * \log\left(\frac{N}{df_w}\right) \quad (27)$$

Où :

t_Q : temps de la requête Q émise.

t_D : temps de publication du document.

w : terme candidat d'expansion.

N : nombre de documents dans la collection (qui comprend que les documents publiés avant t_Q).

df_w : indique le nombre de documents où w apparait.

β: Paramètre qui contrôle la contribution de chaque document au score du terme w en fonction de sa date de publication.

Deuxièmement [Willis et al., 2012] marquent les termes candidats d'expansion basée sur la récence en utilisant la technique de regroupement. La technique de regroupement consiste à identifier les périodes de temps pertinentes en effectuant une recherche initiale, en regroupant les meilleurs résultats par date/ heure. Les groupes sont classés selon la taille par ordre décroissant (c'est-à-dire le nombre de tweets assignés) et indexé par : i= {1,…, T}. Le premier groupe (i=1) correspond à celui qui a le plus grand nombre de tweets du top n résultats, le dernier groupe (i=T) correspond à celui avec le plus petit nombre de tweets du top n résultats, puis favorisent les résultats des plus grands groupes (c'est-à-dire période associée aux meilleurs résultats). Le score final est donné comme suit :

$$\text{score}(w,Q)=\left(\frac{1}{|Q|}\sum_{q\in Q}\left(\sum_{\{D:q,w\in D\}} e^{-\lambda(bin(t_D))}\right)^{-1}\right)^{-1} * \log\left(\frac{N}{df_w}\right) \quad (28)$$

où : la fonction $bin(t_D)$ renvoie l'index du groupe associé à t_D dans la plage [1,T].

3.2.3. Approche de Damak

[Damak, 2014] Propose d'intégrer le temps de différentes manières dans le calcul de la pertinence des tweets.

✓ Emploi de la fraicheur dans la mesure de la pertinence

Premièrement L'auteur propose de renforcer et d'amplifier les scores de pertinence du contenu du tweet en fonction de sa date de proximité temporelle, avec la date de la requête. L'intuition est que certain tweets même ayant un score de pertinence de contenu élevé, ne sont pas pertinents du fait de leurs distance temporelle importante par rapport à la date de soumission de la requête. En d'autre part, des tweets même ayant un score de pertinence de contenue faible sont pertinents, du fait de leurs fraicheurs par rapport à la date de soumission de la requête.

Le score de chaque tweet est donné par :

$$RSVT_1(q,d,\sigma)= RSV(q,d)* k_\sigma(t_q,t_d) \quad (29)$$

Où :

$k_\sigma(t_q,t_d)$: est le score du facteur kernel Laplace définit par :

$$k_\sigma(t_q,t_d)= \frac{1}{2b}\exp\left(\frac{-|t_q-t_d|}{b}\right) \quad (30)$$

$$\text{Avec } \sigma = 2b^2$$

Où :

t_q : représente la date en jour de la soumission de la requête.

t_d : représente la date en jour de publication du document (tweet).

Et σ est le facteur qui permet de modifier le degré d'amplification des scores, en variant sa valeur.

L'auteur remarque qu'en augmentant le σ, l'effet de l'amplification des scores du modèle de recherche $RSVT_1$ diminue, et les résultats se rapprochent des résultats du modèle de recherche de base RSV(q,d).

Deuxièmement l'auteur propose de favoriser les termes fréquemment utilisés au moment de la soumission de la requête, exemple : un document ancien par rapport à la date de soumission de la requête, s'il contient des termes fréquemment utilisés au moment de la requête, donc il sera plus pertinent qu'un document récent contenant des termes fréquemment utilisés dans des périodes lointaines par rapport à la requête.

Pour cela l'auteur propose de modifier le facteur IDF comme suit :

$$\text{IDF} = \log\left(\frac{N-(R_i)_{temp}}{(R_i)_{temp}}\right) \quad (31)$$

$$(R_i)_{temp} = \sum_t |R_i|_t * k_\sigma(t_q, t) \quad (32)$$

Où :

t : correspond à la fenêtre temporelle exprimée en jour.

$|R_i|_t$: correspond au nombre de documents dans cette fenêtre temporelle.

Le score final est donné par $RSVT_2(q,d,\sigma)$ comme suit:

$$RSVT_2(q,d,\sigma) = IDF * k_\sigma(t_q,t_d). \quad (33)$$

L'utilisation de la fraicheur dans le calcul de la pertinence des tweets dans les méthodes proposées, n'ont pas apporté une amélioration significative.

- ✓ Prise en compte de la fréquence temporelle :

Il propose d'amplifier le score d'un terme dans un tweet publié à un instant t en fonction de la fréquence d'emploi de ce terme dans une période. Un même terme aura des scores différents en fonction de la date de soumission du document auquel il appartient. Ce score sera plus important si le terme appartient à un document publié dans une période de rafale de ce terme, que dans le cas où il appartient à un document publié dans une période où le terme n'est pas fréquemment utilisé.

L'auteur propose un nouveau facteur IDF_{new} :

$$IDF_{new} = IDF * {}^{1}/_{IDF_{local}} \quad (34)$$

$$IDF_{local} = \log\left(\frac{N - (R_i)_t}{(R_i)_t}\right) \quad (35)$$

Où :

$(R_i)_t$: est le nombre de tweets contenant le terme i le jour de publication du tweet.

IDF_{local} : est l'IDF d'un terme sur une fenêtre temporelle d'un jour. Ainsi un terme aura un IDF_{local} différent pour chaque jour. Ce facteur est considéré important dans un jour où le terme n'est pas fréquemment utilisé, que dans le jour où il est fréquemment utilisé. C'est pour cela que l'auteur utilise l'inverse du facteur $1/IDF_{local}$.

Le score final $RSVT_3$ (q, d) est donné par :

$$RSVT_3(q, d) = IDF_{new}. \quad (36)$$

La prise en compte de la fraicheur de cette façon n'a pas montré aussi son effet.

3.3. Approches basées sur le profil utilisateur

Le profil utilisateur est représenté par un vecteur de poids de termes qui correspondent aux intérêts de l'utilisateur. La manière la plus courante pour représenter ce profil est le modèle vectoriel, où les intérêts sont représentés par un vecteur de mots clés et qui peuvent changer dans le temps.

[Kacem et al, 2016] proposent de créer un profil utilisateur sensible au temps en combinant le profil à court terme où les intérêts sont éphémères qui reflètent les besoins de l'information des utilisateurs pendant un court laps de temps, et le profil à long terme où les intérêts sont accumulés par les expériences d'une longue période, et qui présentent des intérêts persistants de l'utilisateur. Les auteurs accordent de l'importance aux intérêts récents. Ces intérêts sont représentés par un vecteur de termes de mots-clés, où l'importance de chaque mot-clé est réglée selon le moment de son utilisation, la spécificité du profil réside dans la pondération des termes combinant leurs fréquences et leurs fraicheurs.

Après avoir collecté des mots clés des interactions des utilisateurs dans le réseau social, les poids sont calculés en combinant à la fois leurs fréquences et leurs moments d'apparitions. Un document D^{S_i}= (t_1, t_2, …t_N), généré au moment S_i (jour, heure, minute…), un document est tout contenu généré par l'utilisateur.

Les auteurs extraient les termes des documents et génèrent leur fréquence de terme normalisée (nTF) comme suit :

$$\text{nTF } (t_i)^{Si} = \frac{freq^{Si}(t_i)}{\sum_{\forall k \in D^{S_i}} freq^{Si}(t_k)} \quad (37)$$

Où : $freq^{Si}(t_i)$ est la fréquence relative d'un terme t_i dans D^{S_i},

Et $\sum_{\forall k \in D^{S_i}} freq^{Si}(t_k)$ représente la somme des fréquences de tous les termes apparus-en D^{S_i}

Pour mesurer la fraicheur d'un terme, il faut examiner la notion de fréquence du terme en l'ajustant avec une fonction de polarisation temporelle, en fait [Kacem et al. 2016] supposent que plus le terme est proche de la date actuelle S^C, plus sa fréquence temporelle serait significative, ils utilisent la fonction Kernel gaussienne comme fonction temporelle :

$$K(S^C, S_j) = \frac{1}{\sigma\sqrt{2.\pi.}} . \exp\left[\frac{-(S^C - S_j)^2}{2.\sigma^2}\right] \quad (38)$$

Où : σ est le coefficient d'interpolation,

S^C: est la date actuelle,

S_j : est la date antérieure.

Dans chaque date S_j, [Kacem et al. 2016] définissent le profil utilisateur comme un vecteur U de termes et leurs poids globaux correspondants W :

$$\underset{U}{\rightarrow} = (t_1^{S_j} : W_1^{S_j}, t_2^{S_j} : W_2^{S_j}, \dots, t_m^{S_j} : W_m^{S_j}) \quad (39)$$

Où : le poids temporel $W(t)^{Sc}$ d'un terme t dans le profil est le produit somme de sa fréquence relative temporelle et la fonction de polarisation temporelle comme suit :

$$W(t_k)^{S_C} = \sum nTF(t_k)^{S_j} . K\ (S^C, S_j) \quad (40)$$

La stratégie de personnalisation que [Kacem et al. 2016] adaptent, consiste à soumettre une requête à un moteur de recherche standard et à mesurer la similitude entre le profil utilisateur, et chaque profil de page web retournée$\underset{WP}{\longrightarrow} = (t_{wp1}, t_{wp2}, \dots, t_{wpk})$, grâce à la mesure de similarité cosinus.

Le profil d'utilisateur basé sur le temps peut être affiné en le lissant avec la similarité de la requête de la page web obtenue, pour personnaliser les résultats de recherche pour toutes les requêtes de l'utilisateur pendant la session de recherche, Les résultats sont réorganisés comme suit :

$$\text{Score (U, Q)} = \alpha . Sim\left(\underset{U}{\rightarrow} \underset{WP}{\longrightarrow}\right) + (1 - \alpha) . Sim(\underset{WP}{\longrightarrow} Q) \quad (41)$$

Où : $Sim(\underset{WP}{\longrightarrow} Q)$ est le score obtenu à partir des résultats originaux reflétant l'appariement entre la requête et la page web et $Sim\left(\underset{U}{\rightarrow} \underset{WP}{\longrightarrow}\right)$ désigne la similarité de la page web et de l'utilisateur. Les deux similitudes sont calculées par la fonction cosinus.

L'approche implémentée par [Kacem et al. 2016] a montré des résultats encourageants par rapport à d'autres approches non temporelles.

4. Conclusion

Dans ce chapitre nous avons passé en revue les approches exploitant le temps dans la recherche d'information au niveau de document, requêtes ainsi qu'au niveau du modèle de recherche. Ensuite nous avons introduit le facteur temps pour la recherche d'information dans les microblogs (Twitter) selon des approches proposées.

Dans le prochain chapitre nous exploitons l'approche proposée par [Damak, 2014] en la combinant avec l'approche proposée par [Kacem et al. 2016], et voir ce que cette approche modifiée apporte dans ce domaine de recherche d'information temporelle.

Chapitre III: *Contribution à la RI dans Twitter.*

1. Introduction

Dans le chapitre précédent, nous avons montré l'importance d'intégration du facteur temps, qui est un facteur pertinent crucial pour la recherche de microblogs.

Dans ce chapitre, nous proposons une approche qui se base sur des approches déjà existantes. Dans ce qui suit nous allons détailler notre approche, et voir comment intégrer ce facteur temps dans le calcul de pertinence.

2. Approche proposée

Notre approche a pour principe, de combiner deux scores temporels, ou bien deux fonctions l'une qui calcule la fréquence temporelle de chaque terme, et l'autre est une fonction de polarisation temporelle.

Un score d'un tweet est important, lorsque les termes constituants ce tweet sont fréquemment utilisés dans une période, c'est ce qu'on appelle une période de rafale de ces termes.

Un même terme aura des scores différents selon la date de publication du tweet auquel il appartient.

Ce score sera important lorsque le terme fréquemment utilisé appartient à un tweet publié dans une période de rafale de ce terme.

Nous partageons le même avis que [Damak, 2014] qui dit que l'emploi de la fréquence temporelle de chaque terme est important. Pour ce faire nous avons utilisé la même fonction que celle décrite par l'auteur dans sa troisième approche qui est un IDF modifié (IDF_{new}).

Cette fonction combine l'IDF thématique avec l'inverse de l'IDF_{local} qui est une autre fonction qui prend en compte la fréquence temporelle de chaque terme du tweet.

Où :

L'IDF est la fréquence inverse du document (nombre de documents dans lesquels le terme *i* apparait). Cela signifie que les termes rares donnent une contribution plus élevée au score total.

IDF_{local} est la fréquence inverse d'un terme, sur une fenêtre temporelle d'un jour. Chaque terme aura un IDF_{local} différent pour chaque jour. Les termes qui sont non fréquemment utilisés dans un jour, auront une contribution élevée. C'est pour cela que l'utilisation de l'inverse de cette fonction est importante, ainsi les termes fréquemment utilisés auront un score élevé.

L' IDF_{new} est définit comme suit :

$$IDF_{new}=IDF* {}^{1}/_{IDF_{local}} \quad (42)$$

$$IDF_{local}= \log \left(\frac{N-(R_i)_t}{(R_i)_t}\right) \quad (43)$$

Où :

$(R_i)_t$: est le nombre de tweets contenant le terme i le jour de publication du tweet. Cette fonction calcule la fréquence de chaque terme selon la date de publication du tweet.

Dans notre cas nous pensons à combiner la fonction IDF_{new} avec une autre fonction temporelle, afin d'améliorer les résultats.

Cette fonction est une fonction de polarisation temporelle qui prend en considération la date de soumission de la requête avec la date de publication du tweet.

Le score d'un tweet est important lorsque sa date de publication est proche de la date de soumission de la requête.

La fonction temporelle a été proposée et utilisée par [Kacem et al, 2016], où l'emploi de cette fonction a donné une bonne contribution à son approche. Nous pensons à utiliser la même fonction.

Cette fonction est définit comme suit :

$$k(q, d)= \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(q-d)^2}{2\,\sigma^2}\right) \quad (44)$$

σ: Le coefficient d'interpolation.

q : la date de soumission de la requête.

d: la date de publication du tweet.

L'intuition derrière notre approche vient du fait à intégrer le facteur temporel de sorte à donner un score élevé à un tweet. Pour ce faire, nous proposons de combiner deux fonctions temporelles l'une est IDF_{new} et l'autre est une fonction temporelle Kernel gaussienne.

Le score final d'un tweet est donné comme suit :

$$Score\ (q, d) = IDF_{new} * k(q, d). \quad (45)$$

Où :

$k(q, d)$ est le score du facteur kernel gaussienne.

Et

IDF_{new} est calculé comme nous l'avons décrit.

Notre choix se porte sur l'emploi de IDF_{new} combiné avec la fonction de polarisation temporelle Kernel gaussienne vu les bons résultats retournés respectivement par les deux approches données par [Damak, 2014] et [Kacem et al, 2016].

3. Conclusion

Dans ce chapitre nous avons décrit notre approche, qui permet d'exploiter le facteur temps afin d'améliorer le score thématique, dans la recherche de microblogs.

Dans ce qui suit, nous allons expérimenter l'approche afin de voir si elle rapporte une amélioration par rapport à la recherche thématique.

Chapitre IV: *Implémentation et tests.*

1. Introduction

Dans le chapitre précédent, nous avons présenté notre approche qui intègre le facteur temps dans la recherche des microblogs, en se basant sur la fréquence temporelle.

Dans le présent chapitre nous décrivons le cadre expérimentale de notre approche, ensuite nous présentons les résultats obtenus et les discuter, en les comparant avec les résultats de l'approche thématique, mais avant de passer à cela on donne les outils qui ont contribués à la réalisation de notre application.

2. Outils de développement

2.1. NetBeans IDE

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000. En plus de java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projet multi-langages, refactoring, éditeur graphique d'interfaces et de pages web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris … etc. un environnement Java Développent Kit JDK est requis pour le développement en Java.

L'IDE NetBeans s'appuie sur une plateforme qui permet le développement d'applications spécifiques (Bibliothèque Swing (Java)) [6].

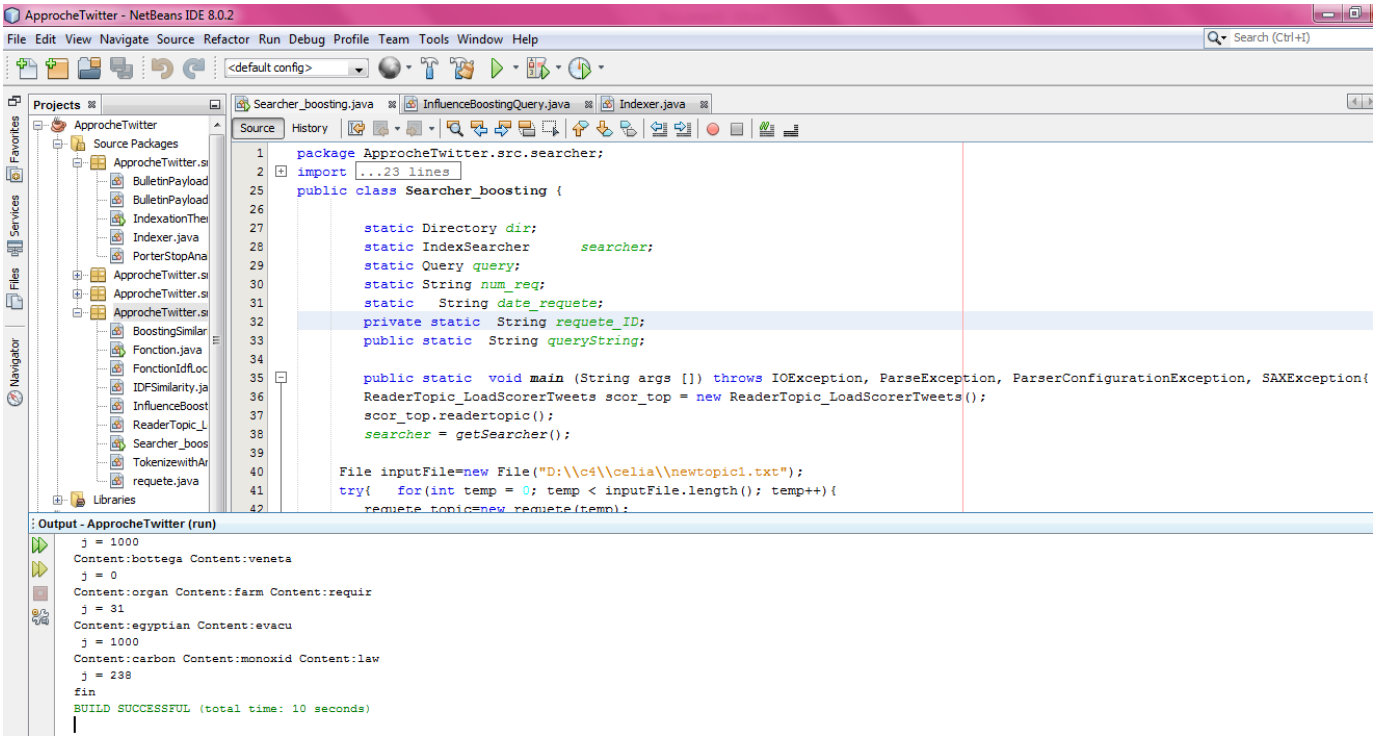


Figure IV-1: *Interface de NetBeans.*

2.2. Le langage Java

Le langage Java est un langage de programmation informatique orienté objet, crée par James Gosling et Patrick Naughton, employé de Sun Microsystems, avec le soutien de Bill Joy (Cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 à SunWorld.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac Os ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java est assurée par ce langage.

Java possède plusieurs versions. Pour implémenter notre approche nous avons choisis la version Java SE 8 qui est sortie en mars 2014 [7].

2.3. Lucene

Lucene est une libraire open source écrite en Java par Doug Cutting, mais il existe des APIs pour d'autres langages dont PHP, Python, Ruby…Permettant d'ajouter des fonctionnalités de recherche plein texte ainsi des capacités d'indexation à des applications. *Pour plus de détails voir dans* ***l'annexe A*****.**

3. Protocole d'évaluation

Dans cette partie on présentera la collection sur laquelle s'est portée nos tests, et les mesures standards, qui nous ont aidés à évaluer notre approche.

3.1. Description de la collection de test

Afin de réaliser nos tests, nous avons utilisé une collection réduite de la collection complète de la tâche TREC microblogs 2011. Cette collection contient :

- 40603 tweets.
- 49 requêtes.
- 49 jugements de pertinence.

3.2. Mesures d'évaluation

Pour évaluer notre approche, nous avons utilisé les mesures d'évaluation standards, à savoir :

- La MAP
- La R-Précision
- La précision@X

Nous avons aussi utilisés les mesures de base Rappel, Précision et la F-mesure

3.3. Expérimentation et résultats

Dans cette partie, nous présentons les résultats obtenus lors des expérimentations de notre approche, on la comparant avec les résultats de l'approche thématique.

- **Précision@X**

Score	P@2	P@3	P@4	P@5	P@10	P@15	P@20	P@30
Thématique	0.2347	0.2925	**0.3367**	**0.3469**	**0.3612**	0.3388	0.3306	0.3204
Approche	**0.2449**	**0.3197**	**0.3367**	**0.3469**	0.3571	**0.3401**	**0.3337**	**0.3211**

Tableau IV.2 : Précision@X des deux approches.

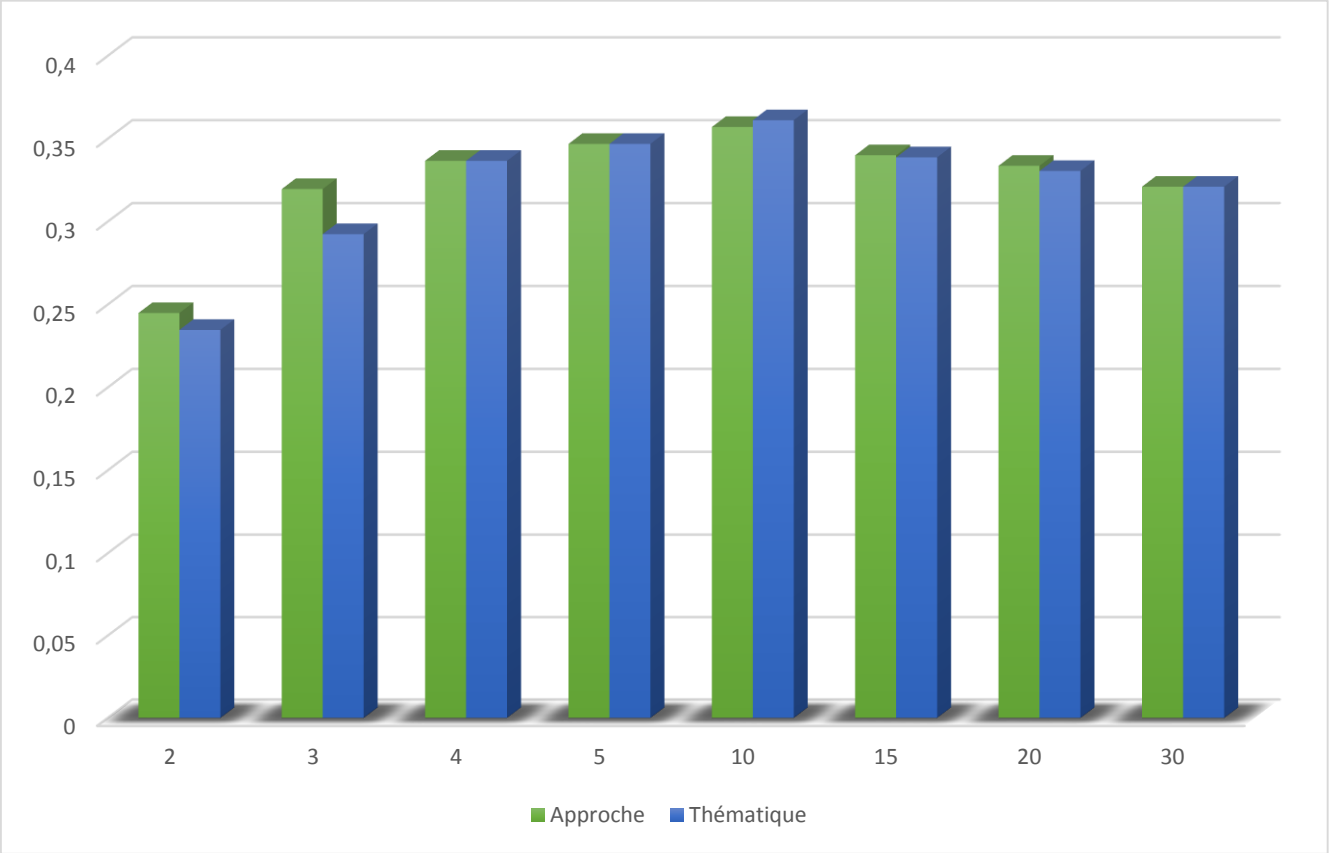


Figure IV -2: *Comparaison de la précision@X du score thématique et du score de l'approche.*

D'après ces expérimentations, les résultats montrent que notre approche améliore légerement les résultats. Par exemple, pour les quinze premiers documents retournés il y a une amélioration de **0.0013**, soit un taux de **0.0038** (0,38 %).

- **Rappel, précision et F-mesure des deux approches**

- **Précision** : une mesure de la capacité d'un système à ne présenter que les documents pertinents.

$$P = \frac{\text{Nombre de documents pertinents récuperés}}{\text{Nombre total de documents récuperés}}$$

$$P_{\textbf{Approche}} = \frac{2257}{37653} = \mathbf{0.0599.}$$

$$P_{\textbf{Approche\_Thématique}} = \frac{2259}{37409} = \mathbf{0.06.}$$

- **Rappel** : une mesure de la capacité d'un système à présenter tous les documents pertinents.

$$R = \frac{\text{Nombre de documents pertinents récuperés}}{\text{Nombre total de documents pertinents}}$$

$$R_{\textbf{Approche}} = \frac{2257}{3081} = \mathbf{0.7325.}$$

$$R_{\textbf{Approche\_Thématique}} = \frac{2259}{3081} = \mathbf{0.7332}.$$

- **F-mesure** :

$$\text{F-score}_{\textbf{Approche}} = (2 * R * P)/(R+P) = \mathbf{0.1107}.$$

$$\text{F-score}_{\textbf{Approche\_Thématique}} = (2 * R * P)/(R+P) = \mathbf{0.1109}.$$

- **Courbes Rappel-Précision**

Rappel	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Précision Thémati-que	0.6901	0.7023	0.6652	0.5669	0.4884	**0.4429**	0.2693	**0.2149**	0.1279	**0.0147**	**0.0001**
Précision Approche	**0.7052**	**0.7313**	**0.6845**	**0.5756**	**0.4488**	0.4169	**0.2937**	0.1995	**0.1297**	0.0146	**0.0001**

***Tableau IV-3**: Rappel et précision des deux approches*

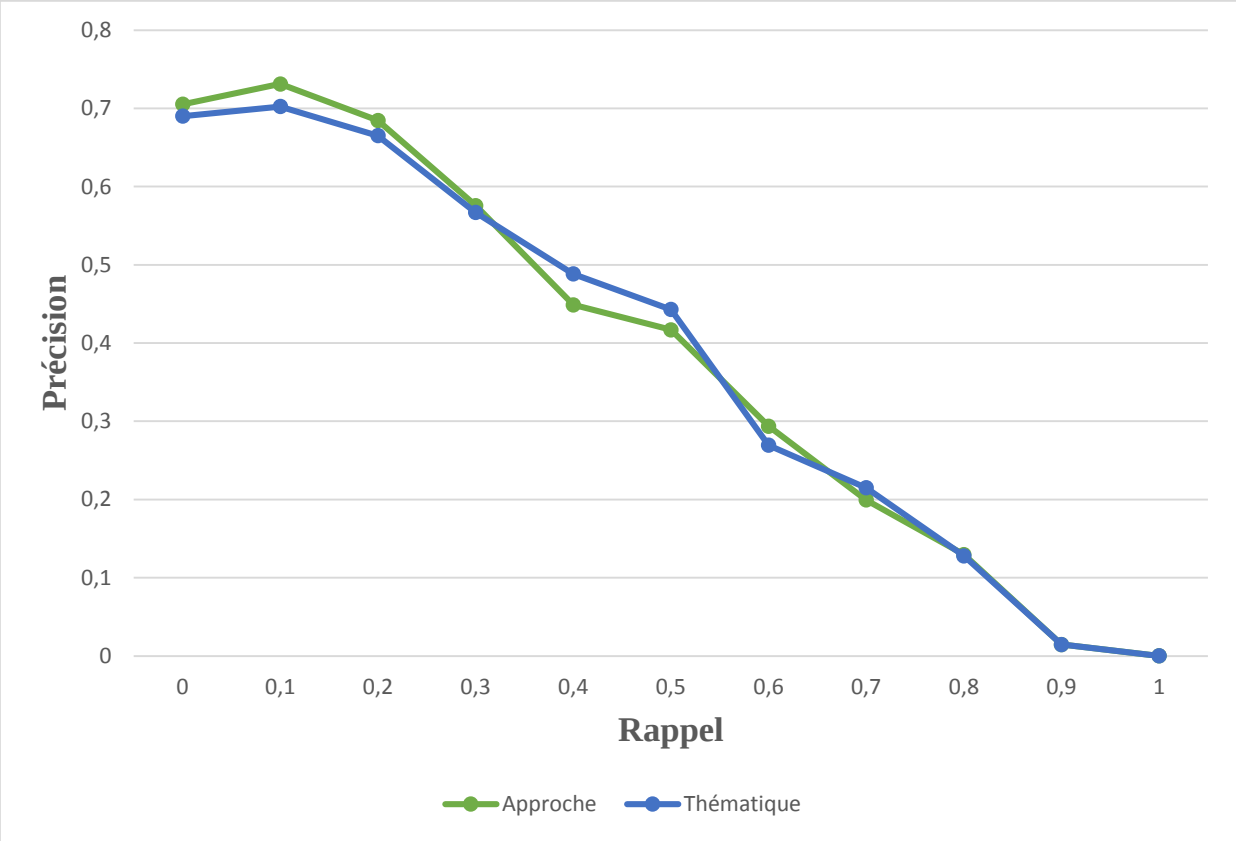


***Figure IV-3**: Comparaison des deux courbes rappel-précision.*

Les deux métriques, Rappel et Précision ne sont pas indépendantes, il y a une forte relation entre elles quand l'une augmente l'autre diminue.

D'après la représentation des deux courbes Rappel et Précision de notre approche et celle de l'approche thématique, nous remarquons une amélioration de notre approche, par exemple au rang 2 on a :

- ✓ Une précision P@2 qui correspond au nombre de documents pertinents parmi les deux premiers documents retournés, qui est égale à **0.2449** pour notre approche et **0.2347** pour l'approche thématique.
- ✓ Un rappel de 1/10=10% qui correspond à deux documents pertinents parmi les dix retrouvés, qui est égale à **0.7313** pour notre approche et **0.7023** pour l'approche thématique.

- **Comparaison de notre approche avec l'approche de Damak**

Mesure	Rappel	P@30	MAP
Approche de Damak	0.6469	0.3198	0.2087
Notre approche	**0.7052**	**0.3211**	**0.2172**

Tableau IV-3 : Rappel et P@30 et MAP de notre approche avec celle de Damak

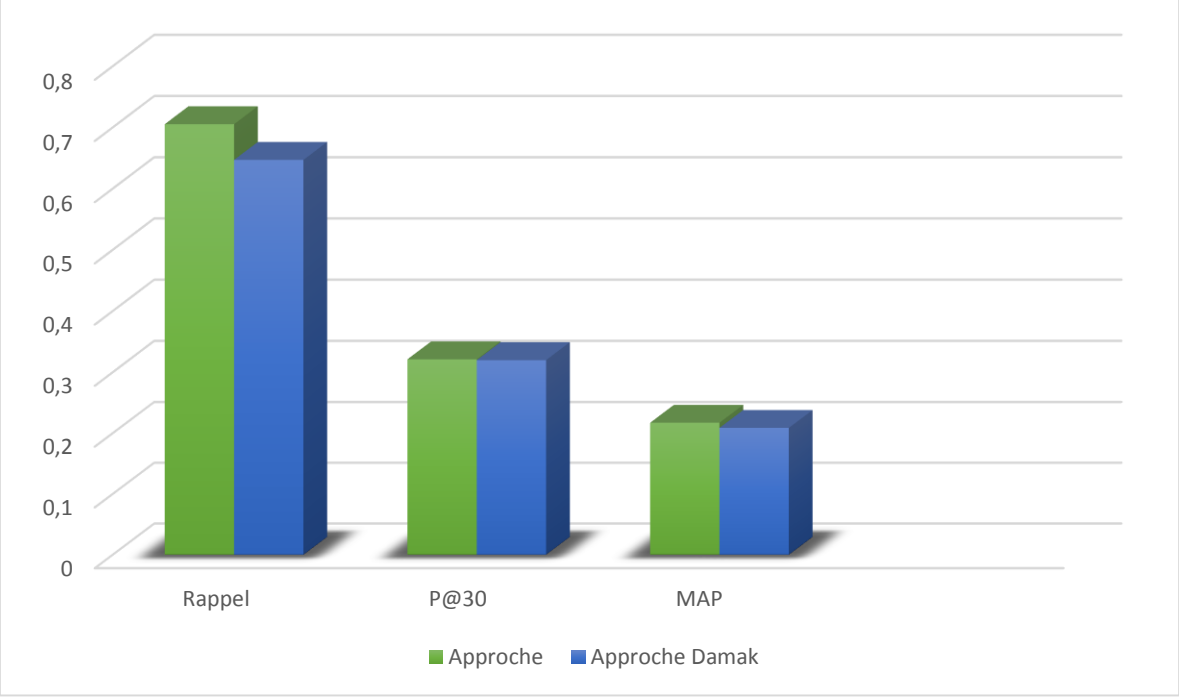


Figure IV-4: *Comparaison de notre approche avec celle de Damak.*

On voie que notre approche améliore les résultats par rapport à l'approche proposée par Damak (IDF_{new}) et ceci en intégrant la fonction Kernel gaussienne.

- **Synthèse**

Enfin d'après les résultats obtenus, nous concluons que notre approche qui intègre la fréquence temporelle, n'apporte pas une amélioration significative par rapport à l'approche thématique, vu qu'elle n'améliore pas la précision, mais n'est en moins elle apporte une amélioration à la troisième approche proposée par [Damak, 2014] citée dans le chapitre II.

4. Conclusion

Dans ce dernier chapitre, nous avons proposé le cadre expérimental de notre approche que nous avons présenté dans le chapitre trois.

L'évaluation que nous avons menée sur une collection de tweets, montre que notre approche n'améliore pas les résultats comme nous le souhaitons par rapport à l'approche thématique.

De là nous pouvons conclure, que la prise en compte de la fréquence temporelle combinée avec la fonction temporelle Kernel Gaussienne, améliore légèrement les performances du système de recherche d'information, mais nous ne pouvons pas conclure que cette approche est meilleure, vu que nos tests se focalisent sur une collection de tweets réduite, ainsi peut être la prise en compte de la condition que la date de soumission de la requête soit supérieure à la date de publication du tweet a influencé sur le nombre de tweets pertinents retournés.

Conclusion générale

Dans notre travail, nous nous sommes intéressés à la recherche d'information temporelle, dans un contexte de microblogings pour la plateforme et le réseau social Twitter, l'objectif est de retrouver les microblogs, répondant à un besoin d'information spécifié par un utilisateur. Pour cela, nous avons cherché comment intégrer le facteur temps, qui est un facteur crucial dans la recherche d'information.

Notre travail a pour objectif d'implémenter et évaluer une approche dans ce contexte. Et pour ce faire, nous avons commencé par présenter la recherche d'information (RI), ainsi la RI appliquée dans Twitter puis nous avons introduit la RI temporelle dans Twitter.

Nous avons proposé une approche, qui intègre le facteur temporel dans la recherche. Cette approche combine l'IDF modifié qui intègre une nouvelle fonction, qui calcule la fréquence temporelle de chaque terme dans un document, avec la fonction temporelle Kernel Gaussienne.

Notre approche améliore légèrement les résultats, mais n'aboutit pas aux résultats que nous voulons, et nous pensons que la combinaison de l'IDF new avec la fonction temporelle Kernel gaussienne d'une autre manière pourra améliorer les résultats (nous pensons à utiliser la somme au lieu de la multiplication).

Bibliographie

Webographie

[1]: http://fr.wikipedia.org/wiki/Systeme_de_recherche_d'infomation.

[2]: ESPACE MULTIMEDIA DU CANTON DE ROCHESERVIERE .Atelier « pour approfondir » Apprivoiser Twitter.

[3]: e-change.ritimo.org, C'est quoi exactement Twitter.

[4]: www.Heideltime.ifi.uni-heidelberg.de/heideltime

[5]: http://fr.wikipedia.org/wiki/Twitter.

[6]: www.fr.wikipedia.org/wiki/NetBeans

[7]: www.fr.wikipedia.org/wiki/Java_(langage)

[8]: www.fr.wikipedia.org/wiki/Proportionnalité

Bibliographie

[Kacem et al., 2016]: Ameni Kacem, Mohand Boughanem, Rim Faiz, Time-Sensitive User Profile for Optimizing Search Personalization, 2016.

[Moulahi, 2016]: Bilel Moulahi. Définition et évaluation des modèles d'agrégation pour l'estimation de la pertinence multidimensionnels pour la recherche d'information. Thèse de doctorat, université Toulouse 3 Paul Sabatier, 2016.

[Baeza-Yates, 1999]: R. A. Baeza-Yates, B A. Rebeiro-Neto. Modern Information Retrieval. ACM Press. Addison-Wesley, 1999.

[Borlund, 1998]: Borlund,P. Measures of relative relevence and ranked halflife: performance indecators for interactive ir. Proceedings of the ACM SIGIR Conference on Research and Developement in Information Retrieval, 1998.

[Boughanem et al., 2008]: Recherche d'information états des lieux et perspectives. Hermès science publications, 2008.

[Willis et al, 2012]: Incorporating Temporal Information In Microblogs Retriaval, 2012.

[Dakka et al., 2012]: Dakka, W., Gravano, L., et Ipeirotis, P. G. Answering general time-sensitive queries. IEEE Translations on knowledge and Data Engineering, 2012.

[Damak, 2014]: Firas Damak : étude des facteurs de pertinence dans la recherche de microblogs.

[Fox, 1992]: Fox, C. Lexical analysis and stoplists,Frakes W B, Baeza-Yates R(eds) Prentice hall, new jersey, 1992.

[Hammache, 2011]: Hammache A. Boughanem, Ahmed Ouamer, New language model combining single and compound terms. IEEE ACM Web Intelligence Conference, 2011.

[Berrazega, 2012]: Berrazega I. TRIME An approach for temporal relation identification between main events, 2012.

[Ingwersen, 92]: P.Ingwersen. Information retrieval interaction.London, Taylor Graham, 1992.

[Jones et Diaz, 2007]: Jones et Diaz, Temporal profiles of queries. ACM trans.Inf.Syst, 2007.

[Nattiya, 2012]: Natiyya K. Time-aware Approaches to Information Retrieval, 2012.

[Lamjed Ben Jabeur, 2011]: Lamjed Ben Jabeur, un modèle de recherche d'information sociale dans les microblogs : cas de Twitter, 2011.

[Lamjed Ben Jabeur, 2012]: Lamjed Ben Jabeur, intégration des facteurs temps et autorité sociale dans un modele bayésien de recherche de tweets, 2012.

[Lie et Croft, 2003]: Li, X. et Croft, W. B. (2003). Time-based language models. In Proceedings of the Twelfth International Conference on Information and Knowledge Management, 2003.

[Lv, 2010]: Lv, Y., Zhai. C. Positional Relavance Model for Pseudo-Relevance Feedback. Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval, 2010.

[Massoudi, 2011]: Massoudi, Incorporating query expansion and credibility into twitter search, 2011.

[Ounis, 2010]: Ounis I., Macdonald C., Lin J., Soboroff I., « TREC 2011 MicroblogTrack », Text REtrieval Conference TREC, November 2010.

[Rocchio, 1971]: Rocchio, J.J Relevance Feedback in Information Retrieval, In The Smart System Experiments in Automatic Document Processing, 1971.

[Robertson, 1977]: Roberston, the probability Ranking principle in IR. Journal of documentation, 1977.

[Salton, 1968]: G. Salton, Automatic Information Organization and Retrieval. New York, McGraw. Hill Book Comapany, 1968

[Teevan et al, 2011] : Teevan, J., Ramage, D., et Morris, M. R. a comparison of microblog search and web search. 2011.

[Croft et al., 2012]: Croft, Temporal Models fo Microblogs, 2012.

Annexe

ANNEXE A

A. Lucene

Lucene est une libraire open source écrite en Java par Doug Cutting, mais il existe des APIs pour d'autres langages dont PHP, Python, Ruby…Permettant d'ajouter des fonctionnalités de recherche plein texte ainsi des capacités d'indexation à des applications.

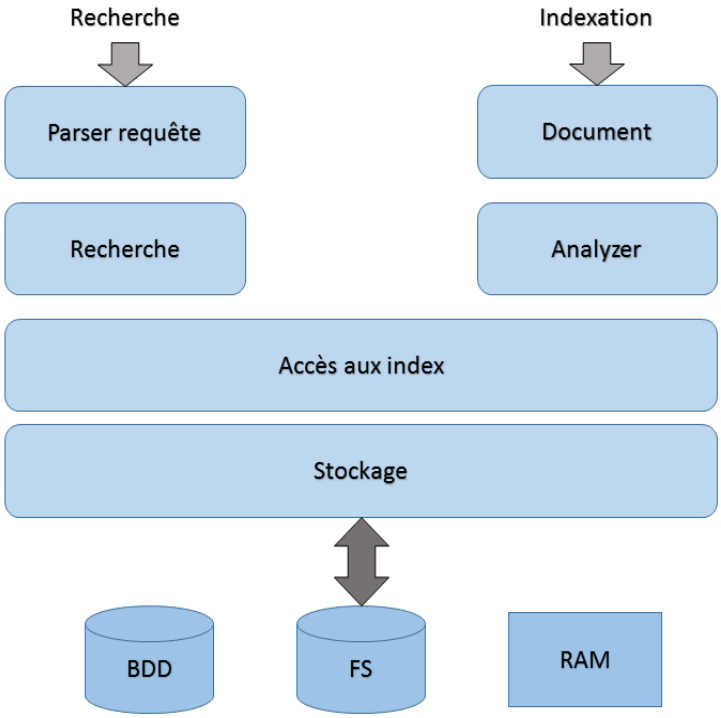


Figure A-1: *Architecture de Lucene*

- Tout en bas se trouvent les objets d'accès aux données, elles sont accessibles par les classes du paquetage « store ».
- Ensuite on a une couche pour accéder aux fichiers d'index.
- La couche recherche et parseur servent à la recherche.

Lucene se découpe en 7 paquetages principaux :

✓ Org.apache.lucene.analysis : il contient du code afin de convertir du texte en élément indexable.

- ✓ Org.apache.lucene.document : contient des classes relatives aux documents.
- ✓ Org.apache.lucene.index : il contient le code pour accéder aux index.
- ✓ Org.apache.lucene.queryparser : son rôle est d'analyser les requêtes afin de générer la requête sous forme d'objet query qui pourront ensuite être réutilisé par le parseur.
- ✓ Org.apache.lucene.search : il se charge de fournir les objets pour chercher dans les indexes.
- ✓ Org.apache.lucene.store : représente une couche d'abstraction d'entrée sortie.
- ✓ Org.apache.lucene.util : les classes sont utilisées dans les autres paquetages.

A.1. Déroulement du lucene

La découverte de lucene se déroulera en deux temps, en premier lieu il faut créer les indexes à partir des documents, pour ensuite soumettre des requêtes au moteur pour effectuer une recherche.

A.1.1. Indexation des données

Les données de recherche effectuées par lucene s'effectuent sur un index. Il s'agit d'une compilation de mots clés et de propriétés identifiant des documents.

L'indexation de données met en œuvre 4 classes lucene qui se trouvent dans les paquetages org.apache.lucene.index et org.apache.lucene.analysis :

- ✓ Indexwriter : c'est la classe qui donne accès aux indexes en écriture (création, ajout de document, optimisation...).
- ✓ Analyzer : il s'agit d'un ensemble de classes qui ont pour but le découpage du texte en token (mot) et la normalisation du texte à indexer.

Et les principaux Analyzer fournis sont :

- SimpleAnalyzer : il découpe le texte en mots et le converti en minuscule.

- StopAnalyzer : même principe que simpleAnalyzer mais il supprime les mots vides tel que (le, la, de, ….).
- StandardAnalyzer : il combine les deux Analyzer précédents.
  - ✓ Document : il représente l'unité élémentaire d'information, il est retourné dans la liste de résultats d'une recherche et il est constitué de champs « field » (nom/valeurs).
  - ✓ Field : c'est le sous élément d'un document, les champs les plus fréquents sont : auteur, titre, date de publication, texte de fichier Word, PDF, HTML.

A.1.2. Effectuer une recherche

La recherche met en œuvre 6 classes lucene qui se situent dans les paquetages org.apache.lucene.search et org.apache.lucene.queryparser :

- ✓ IndexSearcher : c'est la classe qui donne accès aux indexes en recherche.
- ✓ Analyzer : fait partie du processus de recherche pour normaliser les critères de recherche.
- ✓ QueryParser : analyseur de requêtes.
- ✓ Query : représente la requête de l'utilisateur et elle est utilisée par un indexSearcher.
- ✓ Hits : une collection d'éléments résultats de la recherche.
- ✓ Hit : un élément de la collection des résultats.
- ✓ Document : c'est l'unité contenant l'information.

A.2. Fonctionnalités de lucene

Cette bibliothèque libre de la formation apache permet d'indexer et de rechercher du texte en utilisant des analyseurs linguistiques, cette libraire extensible conserve une architecture simple et cohérente on trouve du code, de la documentation, des listes de diffusion pour la recherche plein texte, elle permet aussi le tri de pertinence selon différents algorithmes.

Lucene intéresse les développeurs d'applications de bibliothèques, du secteur linguistique, numérique, des archives…etc.

A.3. Syntaxe du lucene

Lucene supporte une syntaxe très riche et il permet :

- ✓ La recherche d'expression exacte avec les guillemets : «''Jean marche'' ».
- ✓ La recherche booléenne : « Jean and marche », « Jean or marche », « Jean not marche ».
- ✓ La composition avec parenthèse : « Jean or (Marc not Jean) ».
- ✓ La recherche sur les métadonnées : « Marc titre : Jean ».
- ✓ La recherche de termes semblables : « Jaen » pourra trouver « Jean ».
- ✓ La recherche de mots avec caractères omis : « Je ?n » trouvera « Jean ».
- ✓ La recherche avec séquences omises : « Je* » trouvera « Jean ».