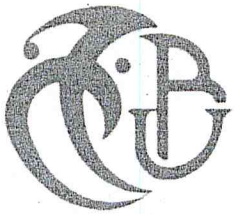
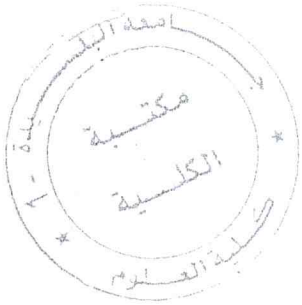
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD-DAHLEB-BLIDA





Faculté des sciences
Département d'informatique

Mémoire Présenté par :

Abdellaoui Aicha **Belkacem Khadidja**

En vue d'obtenir le diplôme de master

Domaine : mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie de logiciel

Un outil pour améliorer la recherche d'information dans le corpus des tweets à base temporelle , sémantique et lexicale

Mme:	**Madani Amina**	Présidente
Mr :	**Nehal djelali**	Examinateur
Mme :	**Boucetta Zouhal**	Promotrice

PROMOTION : 2016/2017

Résumé

De nos jours, les plates-formes de microblogging sont les réseaux sociaux les plus récents et les plus utilisés du Web 2.0. Elles présentent une masse volumineuse d'informations. Aujourd'hui Twitter est parmi les services de microblogging le plus populaire avec 320 millions d'utilisateurs actif par mois et plus de 500 millions de tweets envoyés par jour[1]. Ce volume de publications complique l'opération d'accès à l'information par les Microblogeurs. Le tweet est un document court dont la longueur ne dépasse pas 140 caractères. Souvent écrit avec un langage mal orthographier, contenant des abréviations et des argots à fin de transcrire l'information avec un nombre de caractères minimum. La recherche d'informations dans le corpus des tweets présente un véritable défi pour les modèles de recherche d'informations actuelles, cela est dû au volume du corpus d'une part et aux caractéristiques des tweets d'autre part. En effet, quand l'utilisateur soumet une requête, le modèle de recherche sera confronté à deux problèmes : d'abord l'absence des termes de la requête dans le tweet, et le fait que chaque terme apparaît au plus une seule fois dans le texte. La sélection des meilleurs tweets se base sur un appariement lexical entre la requête et les tweets. De ce fait, il y a une grande probabilité que dans le Top de liste figure des tweets non pertinents. Pour améliorer le classement des tweets pertinents beaucoup de travaux ont introduit les évidences temporelles dans leurs propositions en les combinant avec l'évidence lexicale pour le reclassement des tweets résultats de la première recherche. De notre tour, nous avons proposé un système qui se base sur un nouveau mécanisme pour générer des nouveaux classements des résultats à base temporelle, sémantique et leur combinaison avec la pertinence lexicale (le score de Lucene).

Mots clés :

Twitter , microblogging, le corpus des tweets, requête, modèle de recherche, le reclassement des tweets, recherche sémantique, recherche temporelle, recherche lexicale.

[1]http://blogdumo derateur.com/chiffres-réseaux-sociaux/

Abstract

Nowadays, microblogging platforms are the newest and most used social networks of Web 2.0. They have a large mass of information. Twitter is the most popular microblogging service with 320 million active users per month more than 500 million tweets sent per day. This volume of publications complicates the access to information by microblogs operation. The tweet is a short document whose length does not exceed 140 characters. Often with a misspelled language, containing abbreviations and slangs to transcribe the information with a minimum number of characters. The search for information in the corpus of tweets presents a real challenge for the current information retrieval models, this is due to the volume of the corpus on the one hand and the characteristics of the tweets on the other hand. Indeed, when the user submits a query, the search model will be confronted with two problems: first, the absence of the terms of the query in the tweet, and the fact that each term appears at most once in the text. The selection of the best tweets is based on a syntactic pairing between the query and the tweets. Because of this, there is a high probability that in the list top there are irrelevant tweets. To improve the ranking of relevant tweets a lot of works have introduced temporal evidences into their proposals by combining them with syntactical evidences for ranking the tweets results from the first search. In our turn, we proposed a system based on a new mechanism to generate new rankings of temporal-based results, semantics is their combination with lexical relevance (the score of lucene).

Keywords:

Twitter, microblogging ,the corpus of tweets, query search pattern, ranking of tweets,semantic search, Temporal search,lexical search.

ملخص

في الوقت الحاضر، منصات المدونات الصغيرة هي الأحدث والأكثر استخداما من الشبكات الاجتماعية من ويب 2.0. لديهم كتلة هائلة من المعلومات. تويتر هو من بين خدمات المدونات الصغيرة الأكثر شعبية مع 320 مليون مستخدم نشط شهريا وأكثر من 500 مليون تغريدة في اليوم الواحد. ويعرقل هذا العدد من المنشورات عملية الوصول إلى المعلومات من طرف المدونين. التغريدة هي وثيقة قصيرة طولها لا يتجاوز 140 حرفا. غالبا ما تكون مكتوبة بلغة ذات أخطاء إملائية،لغات عامية و تحتوي على الاختصارات لتدوين المعلومات مع الحد الأدنى لعدد الأحرف. البحث عن المعلومات في مجموعة من التغريدات يمثل تحديا حقيقيا لنماذج استرجاع المعلومات الحالية، وهذا يرجع إلى حجم التغريدة من ناحية وخصائصها من ناحية أخرى. في الواقع، عندما يقوم المستخدم بإرسال عريضة البحث، سيواجه نموذج البحث مشكلتين: الأولى، عدم وجود مصطلحات العريضة في التغريدة، وحقيقة أن كل مصطلح من العريضة يظهر مرة واحدة في النص. ويستند اختيار أفضل نتيجة على الاقتران المعجمي بين العريضة و التغريدة. ونتيجة لذلك، هناك احتمال كبير أن في أعلى القائمة هناك تغريدات ليست ذات صلة.و لتحسين ترتيب التغريدات ذات الصلة هناك الكثير من الأعمال أدخلت الخصائص الزمنية في مقترحاتهم من خلال الجمع بينها مع الخصائص المعجمية من أجل إعادة تصنيف النتائج من البحث الأول. في المقابل، اقترحنا نظاما يقوم على آلية جديدة لتوليد تصنيفات جديدة من النتائج ذات الصلة الزمنية والدلالية والمعجمية (نتيجة لوسين).

الكلمات الرئيسية :

تويتر، المدونات الصغيرة، مجموعة التغريدات ، عريضة البحث، نمط البحث، إعادة التصنيف، البحث الدلالي، البحث الزمني،البحث المعجمي .

Remerciements

Premièrement et avant toute chose, nous rendons grâce à **Allah**, le tout puissant, de nous avoir permis de suivre le chemin du savoir, et donné le courage d'achever ce travail.

Nous tenons, également, à exprimer notre sincère reconnaissance et notre profonde gratitude à tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire, notamment notre promotrice, **Mme. Zouhal Boussatta,** qui grâce à elle nous avons eu l'opportunité de découvrir le domaine de la recherche d'information sur Twitter. Ses conseils illuminés et son aide précieux nous ont permis de mener à bien ce modeste travail et on la remercie très chaleureusement.
Un grand Merci au Melle **Amina Benssialt** et Monsieur **Bilel Garoui** pour l'aide et le soutien et conseils qu'il m'apportés tout long de ce travail

Nous tenons à remercier au corps enseignant ainsi qu'à l'administration de l'université de Blida pour tout le savoir qu'ils ont su nous transmettre durant ces cinq dernières années, et aussi d'être toujours là pour nous guider à retrouver le bon chemin par leur sagesse et leurs précieux conseils.

Mes remerciements vont également à nos parents, qui nous ont aidées de près ou de loin par le fruit de leur connaissance pendant toute la durée de notre parcours éducatif.

Enfin, nous remercions les membres du jury d'avoir accepté d'évaluer notre modeste travail, ainsi que toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce mémoire.

Dédicaces

Je dédie ce modeste travail à mes très chers parents, qui n'ont pas cessé de m'encourager durant toutes mes études, que Dieu les protège

A ma promotrice, *Zouhel Boucetta qui sans elle, ce modeste travail n'aurait pas vu ce jour.*

A mes sœurs (Khadîdja et Leila) et mes frères (Ahmed, Abdelkader, Abdelkarim,Mohammed et soufiane).À mes chères amis Amina, ,Hadjer, Leila, Soumia,Samira et Khadîdja ,Ainsi que Ayoube , S Mohammed, Younesse, Ainsi qu'à toutes mes amis et les personnes qui m'ont aidé à réaliser ce travail.

À tous mes enseignants, à tous mes collègues du cycle Master 2 informatique 2016/2017et à toutes mes amies.

Dédicaces

Je commence par rendre grâce à dieu et à sa bonté, pour la patience, la compétence et le courage qu'il m'a donné pour arriver à ce stade d'étude.

Je dédie ce modeste travail et ma profonde gratitude à celle qui m'a transmis la vie, l'amour, le courage, A toi chère maman « Aida »

A Mon très cher père « Ali », pour tous ses conseils et pour toute la confiance qu'il a mise en moi et pour son dévouement pour mon bonheur.

A mes frère « Mouloud,Sahraoui,Mohamed,Youcef,Ismail »

A ma seule sœur « Houria »

A mon mari « Zohier Korchi »

A Tous mes enseignants

A Tous mes amis :

Nour Elhouda,Aicha,Nassima,Zineb,Khalida,Asma,Khadidja

A Toute ma famille

A Tous ceux que j'aime.

KHADIDJA

Sommaire

Listes des figures

Tableaux :

INTRODUCTION GENERALE

1. Contexte

Actuellement, le monde connaît une avancée technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique d'information et elle joue un rôle important dans la société d'information d'aujourd'hui. La recherche d'informations est un domaine qui s'intéresse à la structure, l'analyse, l'organisation, la recherche et le classement de l'information. Le défi est de pouvoir parmi le volume important de documents disponibles trouver ceux qui correspondent au mieux à l'attente de l'utilisateur. L'opérationnalisation de la RI est réalisée par des outils informatiques appelés Systèmes de recherche d'Informations (SRI). Ce mémoire s'inscrit dans les domaines et les techniques de la recherche d'informations dans les tweets. La recherche d'informations dans les tweets est principalement effectuée par Twitter. Il est parmi les plates-formes de microblogging et les réseaux sociaux les plus récents et les plus utilisés du Web 2.0. Ils présentent une masse volumineuse d'informations puisque les utilisateurs ne se limitent plus à la consommation d'information, mais ils contribuent également à la production des contenus.

Twitter est le service de microblogging le plus populaire avec 320 millions d'utilisateurs actif par mois et plus de 500 millions de tweets envoyés par jour. Ce volume de publications complique l'opération d'accès à l'information par les Microblogeurs. Le tweet est un document court dont la longueur ne dépasse pas 140 caractères. Écrit souvent avec un langage mal orthographie, contenant des abréviations et des argots afin de transcrire l'information avec un nombre minimum de caractères.

La recherche d'informations dans le corpus des tweets présente un véritable défi selon [Choi, 2012] pour les modèles actuels de recherche d'informations, cela est dû au volume du corpus d'une part et aux caractéristiques des tweets d'autre part. En effet, quand l'utilisateur soumet une requête, le modèle de recherche sera confronté à deux problèmes :
D'abord l'absence des termes de la requête dans le tweet, et le fait que chaque terme apparaît au plus une seule fois dans le texte.

La sélection des tweets pour les modèles classiques se base sur un appariement lexical entre la requête et les tweets. De ce fait, il y a une grande probabilité que dans le Top de la liste figurent des tweets non pertinents. Pour améliorer le classement des tweets pertinents beaucoup de travaux ont introduit les évidences temporelles et d'autres les évidences sémantiques pour le reclassement des tweets résultats de la première recherche. C'est dans ce contexte que s'inscrit notre travail qui vise à améliorer l'efficacité de la recherche dans le corpus des tweets, mais en combinant trois sources d'évidence (lexicale, temporelle et sémantique).

2. Problématiques

Les modèles de recherche d'informations classiques se retrouvent handicapés devant le volume accru du corpus des tweets d'une part et devant la taille courte des tweets et la qualité du langage utilisé pour écrire ces documents d'un autre part. Pour remédier à ces problèmes, et amélioré le rang des tweets pertinents dans la liste des résultats, des travaux récents ont introduit les aspects sémantiques, temporelles et sociales en complément avec l'aspect thématique (lexical) pour estimer la pertinence d'un tweet vis à avis une requête et reclasser les tweets selon leurs scores de pertinence. Notre travail s'inscrit dans ce directif.

3. L'objectif

L'objectif de ce travail est la réalisation d'un outil de recherche d'information dans les microblogs « tweets ».Cet outil il va permet d'améliorer le rang des tweets pertinents pour la requête. Ceci via le reclassement de la première liste résultat d'un appariement lexical, selon trois approches: temporelle, sémantique, leur combinaison avec l'évidence lexicale. La première approche consiste à proposer deux techniques temporels pour le calcul de la pertinente temporelle d'un tweet vis à vie une requête, qui prend en considération la pertinence sociale du tweet à savoir, la technique basée sur la fraicheur et celle basée sur la concentration des tweets. La deuxième approche consiste à enrichir là sémantiquement de la requête via WordNet et le contenu informationnel des tweets par les titres des pages Web dont leurs adresses URLs figure dans ces derniers. La dernière consiste à combiner les trois évidences, temporelles, sémantiques et lexicales pour le reclassement des tweets.

Pour l'aspect expérimentation, nous avons mené nosévaluations sur la collection du test de la tache microblogs de TREC2011[2].

4. Organisation du mémoire

Ce mémoire s'articule en 4 chapitres principaux :

Le premier chapitre : résume les concepts de base de la RI, nous commençons par donner une définition de la RI, puis nous décrivons les différents modèles servants comme cadre théorique pour la modélisation du processus de RI. Nous illustrons également les systèmes de la RI en présentant leur définition et les étapes d'un processus de recherche (d'indexation, … etc.), par la suite nous présentons les mesures de similarité. Enfin, nous terminons par détailler quelques moteurs de recherche à accès libre.

Le deuxième chapitre : présent un flash sur la recherche d'informations dans les microblogs. Nous commençons par la présentation des spécificités des plateformes de microblogging : cas de Twitter. Par la suite nous présentons quelques aspects de la recherche d'information sémantique et de la recherche d'informations temporelles dans les tweets puis nous discutons les mesures d'évaluation des systèmes de recherche dans les tweets .Enfin, nous terminons par détailler quelques travaux voisins..

Le troisième chapitre : présente notre contribution, il s'agit d'une nouvelle approche pour la recherche des Microblogs pertinents, dont l'objectif principal est d'améliorer les performances de recherche dans le corpus des tweets. Notre contribution est divisée en trois grandes parties: la première consiste à proposer deux techniques temporels pour la recherche des tweets pertinents à la requête, qui prennent en considération les signaux sociaux à savoir, la technique basée sur la fraicheur, et celle basée sur la concentration des tweets. La deuxième prend en considération l'aspect sémantique de la requête et enrichir les tweets par les titres des pages Web dont leurs adresses URLs figure dans ces derniers. La dernière consiste à combiner les évidences, temporelles et sémantiques et lexicales pour améliorer le classement des tweets pertinents.

[2]http://trec.nist.gov/data/microblog2011.html

Le quatrième chapitre : englobe le détaillé concerne les outils d'implémentation utiliser dans notre travail et la collection de test (TREC2011 Microblogs track) et les interfaces de notre application, aussi les résultats de nos expérimentations et enfin l'évaluation des résultats.

CHAPITRE I :

Recherche d'information

1. Introduction :

La recherche d'Informations (RI) peut être définie comme une activité dont la finalité est de localiser et de délivrer un ensemble de documents à un utilisateur en fonction de son besoin en informations. Le défi est de pouvoir, parmi le volume important des documents disponibles, trouver ceux qui correspondent au mieux à l'attente de l'utilisateur. L'opérationnalisation de la RI est réalisée par des outils informatiques appelés Systèmes de recherche d'Informations, ces systèmes ont pour but de mettre en correspondance une représentation du besoin de l'utilisateur (requête) avec une représentation du contenu des documents (index) au moyen d'une fonction de comparaison (ou de correspondance).

Dans ce chapitre nous présentons les concepts de base de la recherche d'informations (RI) et les différents modèles qui ont été proposés pour fournir un cadre théorique pour la modélisation du processus de recherche d'information (RI). Nous avons aussi décrit le processus de recherche d'information RI, à savoir les étapes d'indexation, d'interrogation et de mise en correspondance. À la fin nous avons présenté quelques moteurs de recherche d'informations à accès libre.

2. Recherche d'information :

La recherche d'information est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie qui ont toujours eu le souci d'établir des représentations de documents dans le but d'en récupérer des informations à travers la construction d'index. L'informatique a permis le développement d'outils pour traiter l'information et établir la représentation des documents au moment de leur indexation, ainsi que pour rechercher l'information. On peut aujourd'hui dire que la recherche d'information est un champ transdisciplinaire qui peut être étudié par plusieurs disciplines utilisant des approches qui devraient permettre de trouver des solutions pour améliorer son efficacité [Damak,2014].

2.1 Concepts de base de la RI :

a. **Requête :** la requête constitue l'expression du besoin en information de l'utilisateur. Elle représente l'interface entre le SRI et l'utilisateur. Divers types de langages d'interrogation sont proposés dans la littérature. Une requête est un ensemble de mots

clés, mais elle peut être exprimée en langage naturel, booléen ou graphique. [Bouramoul,2011].

b. Modèle de représentation :un modèle de représentation est un processus permettant d'extraire d'un document ou d'une requête, une représentation paramétrée qui couvre au mieux son contenu sémantique. Ce processus de conversion est appelé indexation. Le résultat de l'indexation constitue le descripteur du document ou de la requête, qui est une liste de termes ou groupes de termes (concepts), significatifs pour l'unité textuelle correspondante, auxquels sont associés généralement des poids, pour différencier leurs degrés de représentativité du contenu sémantique de l'unité en question. L'ensemble des termes reconnus par le SRI est rangé dans une structure appelée dictionnaire constituant le langage d'indexation. Ce type de langage garantit le rappel de documents lorsque la requête utilise dans une large mesure les termes du dictionnaire. En revanche, il y a risque important de perte d'informations lorsque la requête s'éloigne de ce vocabulaire. [Bouramoul,2011]

c. Modèle de recherche : il représente le modèle du noyau d'un SRI. Il comprend la fonction de décision fondamentale qui permet d'associer à une requête, l'ensemble des documents pertinents à restituer. Il est utilisé pour la recherche d'information proprement dite et est étroitement lié au modèle de représentation des documents et des requêtes. [Bouramoul,2011]

2.2 Les modèles de RI :

Un modèle de RI a pour rôle de fournir une formalisation du processus de RI et un cadre théorique pour la modélisation de la mesure de pertinence. Il existe un grand nombre de modèles de RI textuelle développé dans la littérature. Ces modèles ont en commun le vocabulaire d'indexation basé sur le formalisme mots clés et diffèrent principalement par le modèle d'appariement requête-document. Le vocabulaire d'indexation

$V = \{ti\}$, $i \in \{1, ..., n\}$ est constitué de n mots ou racines de mots qui apparaissent dans les documents. Selon [Baeza, 1999], un modèle de RI est défini par un quadruplet **(D, Q, F, R(q,d))** : où

– **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**

Nous présentons dans la suite les principaux modèles de RI : le modèle booléen, le modèle vectoriel et le modèle probabiliste.

a. Modèle booléen :

Le modèle booléen [Salton,71] est basé sur la théorie des ensembles. Dans ce modèle, les documents et les requêtes sont représentés par des ensembles de mots clés. Chaque document est représenté par une conjonction logique des termes non pondérés qui constitue l'index du document. Un exemple de représentation d'un document est comme suit :

$$d = t1 \wedge t2 \wedge t3 ... \wedge tn ..... (01)$$

Une requête est une expression booléenne dont les termes sont reliés par des opérateurs logiques (OR, AND, NOT) permettant d'effectuer des opérations d'union, d'intersection et de différence entre les ensembles de résultats associés à chaque terme. Un exemple de représentation d'une requête est comme suit :

$$q = (t1 \wedge t2) \vee (t3 \wedge t4) .....(02)$$

La fonction de correspondance est basée sur l'hypothèse de présence/absence des termes de la requête dans le document et vérifie si l'index de chaque document ***dj***implique l'expression logique de la requête ***q***. Le résultat de cette fonction est donc binaire est décrit comme suit :

$$RSV(q, d) = \{1,0\} ..... (03)$$

b. Modèle vectoriel :

Dans ces modèles [Salton,71], la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance dans un espace vectoriel. Le modèle vectoriel représente les documents et les requêtes par des vecteurs d'un espace à **n** dimensions, les dimensions étant constituées par les termes du vocabulaire d'indexation. L'index d'un document **dj** est le $\overrightarrow{vecteur} = (w1j, w2j, w3j, .., wnj)$où $w_{kj} \in [0\ 1]$ dénote le poids du terme **tk** dans le document **dj** . Une requête est également représentée par

$\overrightarrow{vecteur} = (w1q, w2q, w3q, .., wnq)$ où **wkq** est le poids du terme **tk** dans la requête **q**.

La fonction de correspondance mesure la similarité entre le vecteur requête et les vecteurs documents. Une mesure classique utilisée dans le modèle vectoriel est le cosinus de l'angle formé par les deux vecteurs :

$$RSV(q,d) = cos(\vec{q}\vec{d}) \ldots\ldots\ldots(4)$$

Plus les vecteurs sont similaires, plus l'angle formé est petit et plus le cosinus de cet angle est grand. A l'inverse du modèle booléen, la fonction de correspondance évalue une correspondance partielle entre un document et une requête, ce qui permet de retrouver des documents qui ne reflètent pas la requête qu'approximativement. Les résultats peuvent donc être ordonnés par ordre de pertinence décroissante.

c. Modèle probabiliste :

Le modèle probabiliste a été proposé par Robertson et Sparck Jones. Il propose une solution à la problématique de la RI dans un cadre probabiliste : la fonction de pertinence du modèle probabiliste se base sur le calcul de probabilités de pertinence des documents pour les requêtes données. Le principe de base consiste à retrouver des documents qui ont, dans le même temps, une forte probabilité d'être pertinents, et une faible probabilité d'être non pertinents. Ainsi, on distingue deux classes de documents pour une requête qi : les pertinents (R) et les non pertinents ($\bar{R}$). Par conséquent, deux mesures de probabilité sont calculées : **P(R|dj)** la probabilité que le document **dj** soit dans **R** et **P($\bar{R}$|dj)** la probabilité que ce document soit dans **R**. Ainsi, la pertinence entre le document **dj** et la requête **q** est calculée par :

$$RSV(q,dj) = P(R|dj)/P(\bar{R}|dj) \ldots\ldots\ldots(5)$$

En appliquant la règle de Bayes et après quelques transformations, la formule précédente donne :

$$RSV(q,dj) = P(dj|R)/P(dj|\bar{R}) \ldots\ldots\ldots (6)$$

Dans le modèle probabiliste de base, la représentation des documents est composée par des poids binaires indiquant la présence ou l'absence des termes, si on suppose que les termes sont indépendants, la formule **(6)** devient

$$RSV(q,dj) = \sum_{t_i \in T} x_i \,.\, log \frac{p_i(1-q_i)}{q_i(1-p_i)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(7)$$

Avec T est l'ensemble de tous les termes, $xi = 0$ si le terme i n'apparaît pas dans le document j ou bien $xi = 1$ si le terme i apparaît dans le document j.

$pi = P(ti\ 2\ D|R)$, $qi = P(ti\ 2\ D|\bar{R})$……..……..……..(8)

Et $1 - pi = P(ti\ /2\ D|R)$ et $1 - qi = P(ti\ /2\ D|\bar{R})$ ……..(9)

Lorsque des données d'apprentissage pour l'évaluation ne sont pas disponibles, on retrouve le facteur **idf** probabiliste intégré dans le modèle vectoriel :

$$RSV(q,d_j) = \sum_{t_i \in T} x_i \,.\log(\frac{(N-R_i)}{R_i}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(10)$$

Avec N le nombre de tous les documents et ***Ri*** est le nombre de documents contenant ***ti***.

Nous rappelons que, dans le modèle de base, les termes ont des poids binaires dans les documents, indiquant leur présence ou absence. La prise en compte des fréquences des termes dans les documents a fait l'objet de plusieurs modèles variant du modèle de base. Par exemple, dans le modèle BM25(Robertson et al.,1996) le calcul du poids d'un terme dans un document intègre différents aspects relatifs à la fréquence locale des termes (***tfi***), leur rareté et la longueur des documents :

$$S = \frac{(k_i+1).tf_i}{k_1*((1-b)+tf_i+b*\frac{b*dl}{avgdl}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(11)$$

Avec ***dl***est la taille du document ***dj***,***avgdl***est la moyenne des tailles des documents dans la collection et ***k1*** ,***b***sont des paramètres qui dépendent de la collection ainsi que du type des requêtes.

3. Système recherche d'information :

3.1 Définition :

Un système de recherche d'information est défini par un langage de représentation des documents (qui peut s'appliquer à différents corpus de documents) et des requêtes qui expriment un besoin de l'utilisateur (sous forme de mots-clés par exemple), et une fonction de mise en correspondance du besoin de l'utilisateur et du corpus de documents en vue de fournir comme résultats des documents pertinents pour l'utilisateur, c'est-à-dire répondant à son besoin d'information [Tambellini,07]. La figure1 présente un système de recherche d'information.

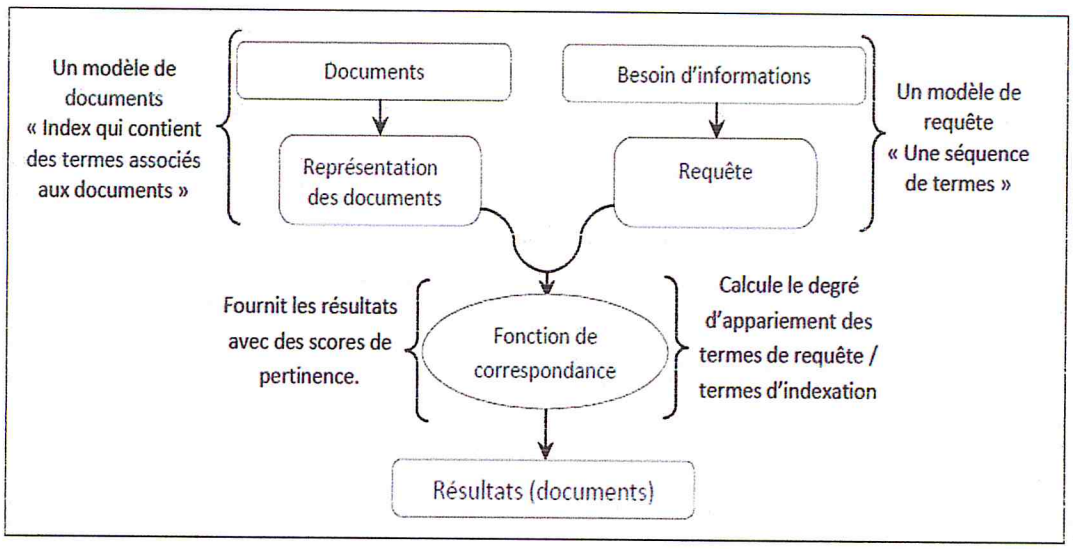


***Figure 1** :Système de Recherche d'Information.*

3.2 Processus de recherche d'information :

Un système de recherche d'information manipule un corpus de documents qu'il transpose à l'aide d'une fonction d'indexation en un corpus indexé. Ce corpus lui permet de résoudre des requêtes traduites à partir de besoins utilisateur. Un tel système repose sur la définition d'un modèle de recherche d'information qui effectue ces deux transpositions et qui fait correspondre les documents aux requêtes. La transposition d'un document en un document indexé repose sur un modèle de document. De même, la transformation du besoin utilisateur en requête repose sur un modèle de requête. Enfin, la correspondance entre une requête et des documents s'établit par une relation de pertinence [Maisonnasse,08].

La figure 2 présente les différentes étapes d'un processus de recherche d'information.

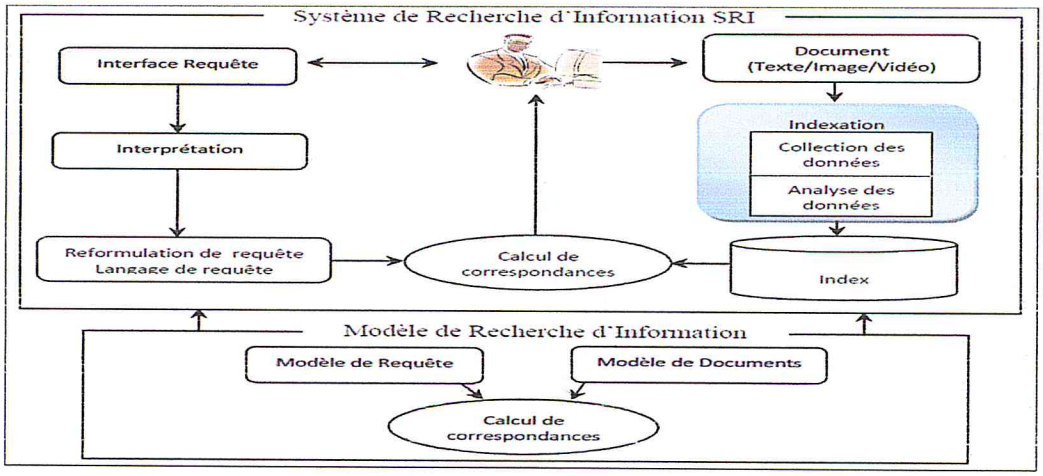


Figure 2 : *Processus de recherche d'information.[Charhad ,05]*

3.2.1. L'indexation :

Crée un index à partir d'un corpus de documents. L'objectif de l'indexation est l'homogénéisation des représentations, tout en rendant l'accès rapide et efficace à l'ensemble des documents. Elle permet d'extraire les mots importants et caractéristiques d'un document et l'indexation peut être manuelle, semi-automatique ou automatique.

3.2.2. Le requêtage :

C'est l'étape durant laquelle l'utilisateur exprime son besoin d'information. Cette étape peut engendrer une reformulation de la requête initiale. La requête soumise par l'utilisateur subit les mêmes traitements que ceux réalisés sur les documents au cours de leur indexation.

3.2.3. L'appariement :

Consiste à mesurer la similarité entre le besoin d'information et les descripteurs des documents dans l'index.

4. Mesure de similarité :

Plusieurs mesures figurent dans la littérature pour la pondération des termes d'un document ou d'une requête ou pour calculer les degrés d'appariement requête/document par la suite nous détaillons la plus importante :

4.1. TF (Term Frequency) :

Cette mesure est proportionnelle au nombre d'occurrences d'un terme dans un document (pondération locale). Toutefois, il existe différentes variantes de cette mesure qui dépendent de la façon dont la pertinence est mesurée. L'inconvénient du TF se situe au niveau de la pertinence globale. Certains termes sont plus significatifs que d'autres, bien qu'apparaissant avec la même fréquence dans un document. Par exemple, dans une collection de documents traitant de la compétition Roland Garros, le terme *Nadal* est plus important que le terme *tennis*, même si ces deux termes apparaissent équitablement dans un document. Pour cette raison le TF est souvent couplé avec la mesure IDF.[Damak, 2014]

4.2. IDF (Inverse Document Frequency) :

Ce facteur mesure l'importance d'un terme dans toute la collection (pondération globale). Un terme qui apparaît souvent dans la base documentaire ne doit pas avoir le même impact qu'un terme moins fréquent. Il est généralement exprimé comme suit : log (N/df), où df est le nombre de document contenant le terme et N est le nombre total de documents de la base documentaire.[Damak, 2014]

Il se calcule selon la formule suivante :

$$IDFt = log\left(\frac{N}{dft} + 1\right) \ldots\ldots\ldots\ldots\ldots (12)$$

N : est le nombre de documents dans la collection et

dft : est le nombre de documents dans lesquels le terme *t* apparaît.

Cette mesure calcule la fréquence d'un terme dans la collection (pondération globale). Cette mesure met en valeur les termes rares et limite l'importance des termes fréquents dans la collection.

La combinaison de TF et IDF fournit une autre mesure importante

$$TFIDF_{t,d} = TF_{t,d} * IDF_t \ldots\ldots\ldots\ldots\ldots (13)$$

Cette mesure donne pour un terme *t* un score important s'il apparaît fréquemment dans peu de documents et un score faible si le terme apparaît rarement dans un même document ou dans beaucoup de documents.

5. Moteur de recherche :

Un moteur de recherche est une application permettant, de trouver des ressources à partir d'une requête sous forme de mots. Les ressources peuvent être des pages web, des articles de forums Usenet, des images, des vidéos, des fichiers, etc, par la suite nous citons les principaux moteurs.

5.1. Moteur de recherche Lucene :

Lucene[3] est une bibliothèque open source écrite en Java qui permet d'indexer et de chercher du texte. Il est utilisé dans certains moteurs de recherche.C'est un projet de la fondation Apache mis à disposition sous licence Apache. Il est également disponible pour les langages Ruby, Perl, C++, PHP, C#.

Lucene est d'abord mis en téléchargement par Doug Cutting sur le site SourceForge.net en mars 2000. Il est alors publié sous licence publique générale limitée GNU. Son transfert vers Apache Jakarta est annoncé en octobre 2001.

5.2. Moteur de recherche Terrier :

Terrier[4] est une plate-forme dédiée à la recherche d'information. Elle implémente les différents modules intervenant dans le processus de RI classique et offre en plus un cadre pour l'évaluation des résultats de recherche pour différentes applications [Ounis, 2006]. Terrier a été largement éprouvée [Middleton ,2007]. Le choix de cette plate-forme est dû aussi à sa capacité à traiter de grandes collections de documents telles que les collections TREC.

L'architecture de la plate-forme Terrier distingue les deux phases classiques : l'indexation et la recherche. Un corpus documentaire est fourni en entrée au module d'indexation. Les documents de la collection passent par un ensemble de prétraitements tels que la tokenisation. Les tokens sont ensuite injectés dans une chaîne de traitement TermPipLines, à savoir le StopWords PipeLine pour l'élimination des mots vides de sens, ou

[3]http://lucene.apache.org/
[4]http://terrier.org/

encore les Stemming pipeline et qui dépendent de la langue en question. La phase d'indexation conduit à la construction de l'index (Data structures).

La phase de recherche comprend le Manager, un module qui interagit avec l'application, réalise la mise en correspondance à travers les calculs des pondérations (selon le schéma de pondération (Weighting Model) choisi : PL2, BM25, etc.) ainsi que les scores des documents. Le résultat renvoyé à l'utilisateur, est la liste des documents jugés pertinents et leurs scores respectifs.

5.3. Moteur de recherche INDRI :

INDRI[5] est un module qui fait partie du projet LEMUR mené par un laboratoire d'université par the university of Massachusetts et the school of computer science at Carnegie Mellon University. Cette application est existante depuis 2004 et poursuit aujourd'hui sont évolution. C'est une solution totalement libre et non commerciale.« La solution de Indri sépare le stockage des données de l'indexation fulltext, ce qui pourrait permettre des modifications et évolutions sans devoir tout changer » .

INDRI avec des fichiers XML est un outil d'indexations qui permet de référencer des mots, des dates, des ordinaux et des balises XML. Les requêtes permettent ensuite de retrouver des documents ou des sous documents contenants ces mots ou intervalles de valeurs ou date… Enfin, un modèle de langage permet de rechercher des documents proches.

En résumé, ***INDRI*** est une solution libre, rapide, fiable, évolutive, qui peut tout à fait être utilisée pour l'indexation et le traitement de requêtes de très nombreux type de document notamment les XML. Il ne manque qu'une interface utilisateur adaptée au grand publique pour que cette solution devienne la référence.

6. Conclusion :

Dans ce chapitre nous avons présenté les principales notions et concepts de la recherche d'informations, des systèmes de recherche d'informations et des moteurs de recherche libre accès.

[5] http://www.lemurproject.org/indri

CHAPITRE II :

Recherche d'information adhoc dans les microblogs

1. Introduction :

Les microblogs sont une forme réduite des blogs. Ils représentent une source d'information récente. Les utilisateurs emploient des plates-formes de microblogging pour partager et accéder à des microblogs. Ces plateformes prennent la forme de réseaux sociaux qui se distingue par des interactions sociales intenses et une diversité dans les sujets discutés, par rapport aux autres sources d'information. Il existe plusieurs plateformes de microblogging. Les plateformes les plus utilisées sont Twitter, Friend Feed, Tumbir, Posterous. Parmi elles, Twitter est sans conteste le plus utilisé. Cette plate-forme compte plus de 650 millions d'utilisateurs, publiant en moyenne 58 millions de tweets par jour. Twitter est utilisé également comme source d'information. En moyenne, 2,1 milliards de requêtes sont soumises chaque jour sur son moteur de recherche. La RI dans les microblogs est différente de la recherche dans le Web. Ceci est dû aux différences de forme des microblogs par rapport aux documents du web, à la spécificité de leur contenu court et mal orthographie et également aux motivations des recherches (informations fraîches).cela a conduite à un véritable défi selon [Choi ,2012] pour les modèles de recherche d'informations actuelles.

Dans ce chapitre nous détaillons Twitter et tout ce qui concerne la recherche d'informations dans le corpus des tweets.

2. Présentation et spécificités des plate-formes de microblogging : cas de Twitter.

2.1. Historique de Twitter :

Twittera été créé en 21 mars 2006 à San Francisco au sein de la société américaine startupOdeo fondée par Noah Glass et Evan Williams, et Jack Dorsey. Cette société proposait une plateforme d'hébergement, de diffusion et d'enregistrement de podcast. L'idée de départ lancée par Jack Dorsey était de permettre aux utilisateurs de partager facilement leurs petits moments de vie avec leurs amis. Le 21mars2006, M. Dorsey envoyait son premier tweet : « Just setting up my twttr » (« Suis en train d'installer mon twttr »). Le marché du podcast étant déjà très concurrentiel, Jack Dorsey et Noah Glass et Evan Williams furent chargés de développer un nouveau service ouvert au public le 13juillet2006, la première version s'intitulait *Stat.us* puis *Twittr*, en référence au site de partage de photos Flickr puis *Twitter*, son nom actuel. Le 25octobre2006, les actifs de la société Odeo ont été rachetés par

Obvious Corp. Puis en avril2007, une entité indépendante est créée comme nom Twitter avec Jack Dorsey à sa tête jusqu'en octobre2008 date à laquelle Evan Williams lui succéda. En mars 2008, Twitter compte un million d'utilisateurs. La société compte 29 employés en février2009, 300 en octobre 2010 et 900 en avril 2012. En juin2012, les mots « Twitter » (nom propre), « Twitt » ou « tweet », « Twitteur » ou « Twitteuse », ainsi que « Twitter » ou « Tweeter », font leur apparition dans Le Petit Larousse édition 2013.Twitter dont le prix d'introduction est fixé à 26 dollars entre à la bourse de New York le 31octobre2013 sous le symbole « TWTR » avec une première cotation qui s'effectue à 45,10 dollars. L'action atteindra un pic à 73,31 dollars en décembre 2013 avant d'amorcer une chute jusqu'à 31,85 dollars à la fin du lock-up (période durant laquelle un actionnaire ou un investisseur ne peut se défaire de ses actions) le 6 mai 2014. Dick Costolo démissionne de son poste de PDG de Twitter en juin 2015, sur fond de désaveu de sa stratégie. Il est remplacé de façon intérimaire par l'un de ses fondateurs, Jack Dorsey.[67]

2.2. Présentation générale de Twitter :

Twitter est l'exemple le plus populaire des plateformes de microblogging. Cesplate-formes sont les réseaux sociaux les plus récents du Web 2.0. Elles sont considérées comme une nouvelle forme de blogs, où les informations diffusées sont courtes et publiées plus rapidement. Ces informations concernent différents sujets. Les utilisateurs parlent de leur quotidien, des événements, des tendances parfois à la mode SMS et en partageant des messages de faible longueur (par exemple 140 caractère au plus dans le cas de Twitter).Twitter a connu une croissance exponentielle durant ces dernières années. Nous présentons ci-dessous les principales spécificités de cette plate-forme, ainsi que l'information qui y est produite.

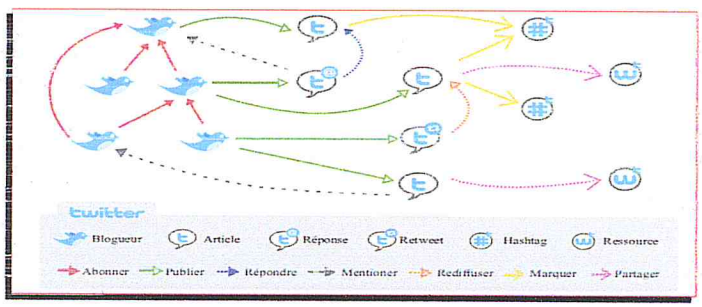


Figure 3 *:Réseau social d'information de Twitter.*

[6] URL: http://www.numerama.com/startup/twitter
[7] URL : http://oseox.fr/twitter/histoire-twitter.html

2.3. Les Followers :

Twitter a mis en place un concept de *followers* (suivre les gens). Donc, on a des followers (des personnes qui nous suivent) et ont suit les gens (on est leur follower), c'est-à-dire que l'on suit les informations qu'ils postent et dès qu'un certain utilisateur met à jour son statut, tous les followers sont informés. Ce résultat est obtenu en ajoutant la nouvelle entrée à leur page personnelle, un aperçu est représenté sur la Figure.

Figure 4 *:Capture d'écran de la page personnelle d'ensemble Twitter*[8].

Cette opération est réalisée en cliquant sur le bouton suivre ou (Follow) sur une page Twitter. On peut suivre tous les autres utilisateurs à moins que cet utilisateur à mis son profil en mode privée. Dans ce cas, une demande d'approbation doit être envoyée en premier.

2.4. Lexique de Twitter :

- **Twitto :** est un utilisateur de Twitter.

Figure 5 *:Capture d'écran de la page profile de l'utilisateur de Twitter*[9].

[8]https://twitter.com/?lang=fr
[9]http://www.alesiacom.com/blog/maitriser-le-vocabulaire-de-twitter

- **Tweets** « gazouillis » : sont les messages postés sur Twitter. Ils sont limités à 140 caractères.

Figure 6 *:Capture d'écran d'un exemple de tweet.*

- **J'aime** : Cliquer sur J'aime est un moyen simple de montrer que vous appréciez un Tweet. Vous pouvez par ailleurs utiliser cette fonctionnalité pour facilement retrouver ce Tweet plus tard. Cliquez sur l'icône en forme de cœur pour aimer un Tweet ; l'auteur verra ainsi que vous l'appréciez.
- **Following / Abonnements** : correspondent aux nombre des comptes Twitter que vous suivez. Pour connaitre le nombre d'abonnements, allez sur votre page d'accueil Twitter le nombre se trouve dans la colonne de droite tout en haut. Et pour voir tous vos following (personnes que vous suivez) cliquez sur le nombre ou « Abonnements».

Figure 7 *:Capture d'écran d'un exemple d'abonnement.*

- **Followers / Abonnés** : c'est le nombre de comptes Twitter qui suit cette personne. Tout comme pour les abonnements, le nombre se situe sur la page d'accueil dans la colonne de droite et vous pouvez voir qui vous suit en cliquant +sur le nombre ou «Abonnés ».

Figure 8 **:** *Capture d'écran d'un exemple d'un abonné.*

- **@Réponses** : si vous souhaitez répondre à un tweet, vous pouvez envoyer un tweet en débutant par le nom du compte précédé par un "@". Si nous prenons par exemple le tweet "@Antoine Bonjour !", vous allez ici envoyer le message "Bonjour" au compte d'Antoine, celui-ci verra votre réponse dans l'onglet "Réponses" de son profil. A noter que votre réponse est visible par tout le monde, du moins ceux qui vous suivent et qui suivent également le destinataire de votre message, et apparaît dans votre historique de tweets.

- **Timeline** : Il s'agit du flux d'actualités de Twitter. La timeline générale présente l'ensemble des tweets postés par vos abonnements, et votre timeline personnelle affiche les différents tweets que vous avez mis en ligne. La timeline affiche les messages par ordre antéchronologique, c'est-à-dire du plus récent au plus ancien.

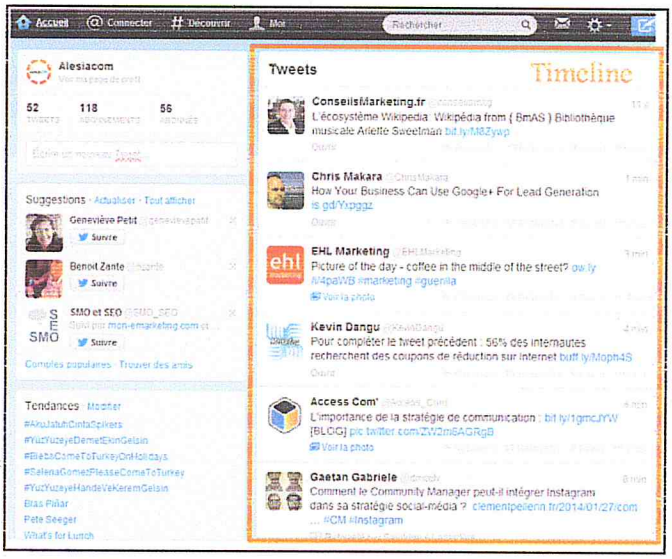


Figure 9 : *Capture d'écran d'un exemple de Time line.*

- **Les mentions (@) :** Un nom précédé d'arobase « @ » est un lien vers le compte Twitter de l'utilisateur de ce nom (qui permet de voir tous ses tweets, sauf s'ils sont protégés). Chaque utilisateur peut consulter les mentions qu'il a reçues dans l'onglet « @ Connect ». Si un tweet débute par une mention, seuls les followers suivant le compte mentionné verront le tweet dans leur fil d'actualité (par exemple @Eve rédige un tweet en commençant par @Bob, donc parmi les followers de @Eve, seuls ceux qui suivent également @Bob liront le tweet depuis leur fil d'actualité).

Figure 10 **:** *Capture d'écran d'un exemple de mention.*

- **RT (retweeter) :** Action qui consiste à rediffuser le message d'un autre utilisateur à vos abonnés. Un retweet (également désigné par l'abréviation RT) est donc un message rediffusé.

Figure 11: Capture d'écran d'un exemple de RT

- **Message Privé (MP)** : (se dit « DM », pour « Direct Message » en anglais). Cette fonction permet d'envoyer un message privé à un utilisateur. Les MP sont eux-aussi limités à 140 caractères mais ils n'apparaissent pas dans les timeline : ils arrivent sur une messagerie interne à Twitter. On ne peut envoyer un MP à une personne que lorsqu'on la suit sur Twitter, et elle ne peut nous répondre que si elle nous suit également.
- **Hashtag(#) :** Le « # » suivi d'un mot (sans espace et éviter les accents et autres caractères spéciaux) fonctionne un peu comme un mot clé ou un tag. Il permet de définir de manière générale le sujet principal du tweet. Lors d'un événement, il permet de suivre toutes les conversations sur Twitter relatives à cet événement. Ce qui est intéressant avec les hashtags, ils permettent de découvrir de nouvelles personnes qui parlent ou s'intéressent aux mêmes sujets que vous.

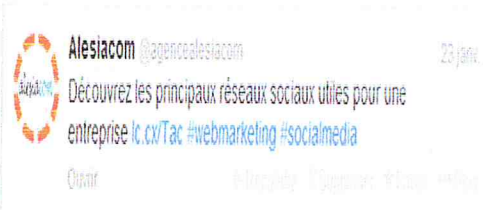


Figure 12 **:** *Capture d'écran d'un exemple d'un Hashtag.*

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

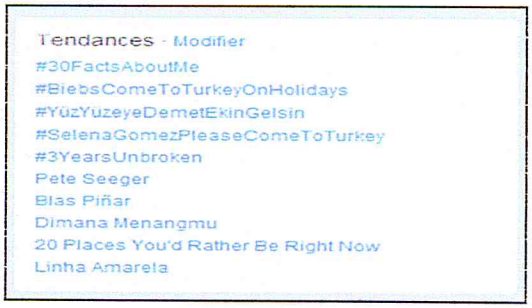


Figure 13 **:** *Capture d'écran d'un exemple des tendances.*

2.5Type des tweets :

Il existe plusieurs types de tweets sont :

- **Tweet normal :** tout message de 140 caractères maximum publié sur Twitter.
- **Réponses :** Tweet qui commence par le @nomdutilisateur d'un autre utilisateur et qui répond à l'un des Tweets de celui-ci, par exemple : @Assistance Je n'arrive pas à croire que tu n'as pas aimé ce film !
- **Mention :** Tweet contenant le nom d'utilisateur d'un autre utilisateur de Twitter précédé du symbole @, par exemple : Bonjour @Assistance ! Quoi de neuf ?
- **Message direct (DM) :** Un tweet privé envoyé à une personne qui vous suit, vous ne pouvez pas envoyer un message direct à quelqu'un qui vous ne suit pas.

3. Recherche adhoc des microblogs

Le principe de la recherche adhoc de microblogs est similaire à la RI adhoc classique. Il s'agit de répondre à une requête via un index de microblogs et sélectionner ceux qui sont pertinents [Efron, 2011]. La différence entre la RI adhoc dans les tweets et la RI adhoc dans les documents du Web réside dans la nature de l'information traitée et des sessions de recherches. Ces différences sont principalement dues aux spécificités des microblogs par rapport aux autres sources d'information et les motivations des utilisateurs pour chercher dans cette source d'information.

[Efron,2011] ont posé la question : quels sont les facteurs reflétant la pertinence dans la recherche de microblogs ? Les facteurs tels que la popularité de l'auteur et l'horodatage ont probablement leur importance pour juger l'utilité d'un microblog par rapport à un autre. Cependant, la manière de considérer ces qualités n'est pas évidente.

Ainsi, il existe plusieurs facteurs de pertinence à prendre en compte dans la conception des approches de recherche de microblogs, en plus de la pertinence textuelle : facteurs sociaux, facteurs de popularité des auteurs, facteurs de fraîcheur, facteurs liées aux URLs.

4. Les Ontologies :

La notion d'ontologie intéresse à la fois l'ingénierie des connaissances, la linguistique et la philosophie. Initialement, l'ontologie était un domaine de la philosophie concernant « l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe ». Or, progressivement les objets produits par cette discipline ont pris le nom d'ontologies [Ottens,07]

4.1 . Définition :

Cette notion a été reprise par les chercheurs dans le domaine de l'intelligence artificielle et utilisée dans le cadre de construction des systèmes à base de connaissances. L'idée était de séparer, d'un côté, la modélisation des connaissances d'un domaine, et d'un autre côté, l'utilisation de ces connaissances (i.e. le raisonnement).

4.2. Rôles des ontologies :

Historiquement, la notion d'ontologie est apparue pour satisfaire des besoins d'interopérabilité dans les systèmes informatiques et de réutilisation. On attend d'elles qu'elles améliorent la communication non seulement entre machines, mais aussi entre humains et machines ou encore entre humains par le biais de logiciels. Les propriétés de ce type de structure de données ont permis de diversifier leur utilisation à différentes applications, en particulier la gestion des connaissances et le Web sémantique. Elles sont utilisées pour : [Ottens, 07]

– Résoudre des problèmes de compréhension et faciliter le partage des connaissances entre personnes de spécialités différentes ;
– Assurer l'interopérabilité entre applications à base de connaissances
– Accéder à des ressources hétérogènes.
– Permettre la réutilisation de modèles de connaissances;
– Faciliter la communication entre agents logiciels.
– Annoter des ressources à l'aide de méta-données.
– Améliorer les processus de recherche d'informations.

4.3. Recherche d'information guidée par les ontologies :

En effet, les ontologies peuvent améliorer la pertinence d'une recherche et ce, en recherchant des documents faisant référence à un concept précis au lieu de se baser sur des mot-clés qui peuvent être ambigus.

Cette recherche basée sur les ontologies se présente comme une recherche intelligente qui repose sur la sémantique des ressources et sur les concepts contenus dans les documents qui leur sont associés. Ces ontologies peuvent ainsi, d'une part, guider la création d'annotations sous la forme de métadonnées sur les ressources, et d'autre part, décrire leurs contenus de manière à la fois formelle et signifiante pour être exploitable aussi bien par les humains que par les machines.

5. La recherche d'information temporelle :

C'est une nouvelle tendance pour la recherche d'informations dans les tweets . En raison du court texte des tweets, seules les résultats de recherche liées à la pertinence du contenu ne peuvent pas satisfaire les besoins d''information des utilisateurs. La recherche temporelle montre une amélioration des performances de récupération pour les tweets qui discute des news ou des récents récents ou bien anciens, les travaux proposés dans ce domaine on peut les classer dans deux catégories. La première considère que les tweets récents sont les plus pertinents pour une requête et ils sont présenté leurs modèles pour les sélectionner. La deuxième opte pour l'idée que les documents intéressants sont ceux qui figurent dans les grandes concentrations des tweets, plusieurs approches ont été proposées aussi dans ce contexte.

6. Evaluation :

Pour évaluer un système de recherche d'informations, il suffira de lui soumettre les questions tests, et de comparer les réponses qu'il fournira aux réponses attendues.

6.1. Les campagnes d'évaluation:

Les campagnes d'évaluations représentent le modèle actuel dominant. En effet, c'est sur l'expérience des tests de Cranfield que s'est basé le NIST (National Institute of Science and Technology) pour créer la campagne d'évaluation TREC (Text REtrieval Conference) en1992.

Les campagnes de TREC sont devenues la référence en ce qui concerne l'évaluation des systèmes mais on peut également citer les campagnes CLEF (Cross-Language Evaluation Forum) qui se rattachent plus particulièrement aux systèmes multilingues, les campagnes NTCIR et Amaryllis.

1) La campagne d'évaluation TREC est une série d'évaluations annuelles des technologies pour la recherche d'informations. Les participants sont en général des chercheurs pour de grandes compagnies commercialisant des systèmes et voulant les améliorer et des groupes de recherche universitaires. Aujourd'hui le TREC est considéré comme le développement le plus important dans la recherche d'informations expérimentales, et demeure le plus cité et utilisé par la communauté de RI. Les pistes principales explorées sont le filtrage, la tâche adhoc et la tâche question-réponse.[Ben Jabeur,2013]

La collection de test Tweets2011 que on utiliser dans notre travaille comprend :

- 16 millions de tweets (0,5 Go) exprimés dans diverses langues et publiés sur Twitter entre le 23 janvier 2011 et le 8 février 2011.
- 50 *topics* dont on trouvera un exemple en figure suivant. La balise titre décrit le besoin exprimé à un moment donné (querytime). Ce moment correspond concrètement à la date de publication du tweet le plus récent de la requête.

```
<top>
<num> Number: MB038 </num>
<title> protests in Jordan </title>
<querytime> Tue Feb 01 12:46:40 +0000 2011  </querytime>
<querytweettime> 32419560749531136 </querytweettime>
</top>
```

Figure 14 **:** *Exemple d'un topic pour la tâche Microblog de TREC2011.*

2) La campagne CLEF est lancée en 2000 comme un projet européen d'évaluation des SRI. Le but de ce projet est de promouvoir la recherche dans le domaine des systèmes multilingues en organisant des campagnes d'évaluations annuelles. L'intention est d'encourager l'expérimentation de toutes sortes d'accès à l'information multilingues, allant du développement des systèmes de recherche monolingue opérant sur de nombreuses langues à la mise en oeuvre des services de recherche multilingues et multimédia. L'objectif est aussi d'anticiper les nouveaux besoins de la communauté R&D et d'encourager le développement des SRI multilingue de prochaine génération (Petes., 2009).

CLEF 2009 s'est focalisé sur huit tâches principales, les plus importantes d'entre elles sont : recherche de documents textuels multilingues, recherche dans les collections d'images et l'analyse des fichiers log.[Bouramoul,2011]

3) La campagne INEX : INEX[10] (INitiative for the Evaluation of XML Retrieval) est la seule campagne d''evaluation des différents SRI pour la recherche d'information sur les documents XML. Elle est mise en place chaque année depuis 2002. Elle offre un forum international non seulement pour permettre aux différentes organisations participantes d''evaluer et de comparer leurs résultats, mais aussi pour discuter les différentes problématiques qui se présentent. La collection de test consiste en un ensemble de documents XML, requêtes, taches de recherche et jugements de pertinence.

6.2. Discussion sur les mesures d'évaluation

Tout l'enjeu du processus de recherche d'information est de minimiser la distance entre la pertinence système et la pertinence utilisateur. Plusieurs mesures standards en RI ont été proposées pour évaluer les performances des SRI. Nous nous basons sur les travaux de [Kompaoré, 08] pour présenter ces mesures.

1) La mesure de précision calcule la capacité du système à rejeter tous les documents non pertinents pour une requête. Elle est donnée par le rapport entre les documents sélectionnés pertinents et l'ensemble des documents sélectionnés :

$$Précision = \frac{|Documentspertinentsrestitués|}{|Documentsrestitués|} \in [0,1] \ldots\ldots\ldots (14)$$

2) Le rappel calcule la capacité du système à restituer le maximum de documents pertinents pour une requête. Il mesure la proportion de documents pertinents restitués par le système relativement à l'ensemble des documents pertinents contenus dans la base documentaire. Il est exprimé par :

$$Rappel = \frac{|Documentspertinentsrestitués|}{|Documentspertinents|} \in [0,1] \ldots\ldots\ldots (15)$$

Le rappel et la précision sont calculés indépendamment de l'ordre dans lequel les résultats sont représentés (ce sont des mesures ensemblistes). Des mesures tenant compte de l'ordre des documents sont également nécessaires. Elles permettent par exemple d'évaluer des

[10]https://inex.mmci.uni-saarland.de/

systèmes tels que les moteurs de recherche du web où l'ordre d'apparition des documents est crucial. À cet égard, les mesures principales proposées sont la **précision@X** et la **précision moyenne**.

3) La **précision@X** est la précision à différents niveaux de coupe de la liste. Cette précision mesure la proportion des documents pertinents retrouvés parmi les X premiers documents restitués par le système.

4) La **précision moyenne** est la moyenne des valeurs de précisions après chaque document pertinent. Elle se focalise en particulier sur le document pertinent classé dans les premiers rangs.

$$AP_q = \frac{1}{R}\sum_{i=1}^{N} p(i) * R(i) \text{ .....(16)}$$

Où $R(i)$ = 1 si le ième document restitué est pertinent, $R(i)$ = 0 si le ième document restitué est non pertinent, $p(i)$ la précision à i documents restitués. R le nombre de documents pertinents pour la requête q et N le nombre de documents restitué par le système.

5) La **moyenne des précisions moyennes** (Mean Average Precision-MAP) est obtenue sur l'ensemble des requêtes, Cette mesure calcule la moyenne des valeurs de précision moyenne non interpolées sur l'ensemble des documents pertinents. La formule suivante donne la méthode de calcul de la MAP :

$$\boldsymbol{MAP = \frac{\sum_{q \in Q} AP_q}{|Q|}} \text{.....(17)}$$

Avec AP_q est la précision moyenne d'une requête **q**, **Q** est l'ensemble des requêtes et **|Q|** est le nombre de requêtes. Cette mesure peut être qualifiée de globale puisqu'elle combine différents points de mesure.

7. Travaux voisins :

Ils existent plusieurs travaux qui ont contribué dans le domaine de recherche d'information sémantique et temporelle dans les tweets, nous résumons ci-après les plus importants :

Titre	Résumé	Méthode	Auteur et l'année
Effectiveness of State-of-	Dans ce travail : -ils ont utilisé wordnet pour enrichir la requête sémantiquement avec les	-Technique de *Roucchio* -Mécanisme naturel de	**Firas Damak *2013* [Damak,*2012]**

the-art Features for Microblog Search	synonymes des termes.	Pseudo-Pertinence Feedback (PRF) -Le modèle *BM25* -La mesure de précision et rappel - TF (Term Frequency) + IDF (Inverse Document Frequency).	
Use of Twitter and Semantic Resource Recovery in the Educational Context	Dans cet article : Ils ont développé un plugin contextuel qui intègre Twitter dans Moodle, Ce plugin effectue la recherche sémantique des tweets et des documents dans des dépôts externes en utilisant la requête fournie par l'utilisateur et un Contexte spécifié par Moodle.	Protocol OAI-PMH' (Open Archives Initiative – Protocol for Metadata Harvesting)	2012 IEEE 21st International WETICE [WETICE,2012]
Combining Temporal and Content Aware Features for Microblog Retrieval	Dans cet article, ils ont proposé une méthode pour redéfinir le résultat de la recherche en fonction des caractéristiques temporelles, des fonctionnalités liées au compte et des fonctionnalités spécifiques au twitter, ainsi que des fonctionnalités textuelles des tweets. Ils ont également appliqué une technique d'expansion de requête en deux étapes pour améliorer la pertinence de sélection des tweets. Ils effectuent leurs expériences sur la collection TREC 2011	-Modèle de langue avec lissage Dirichlet -Modèle d'espace vectoriel -URL - Compte Retweet -Compte de statut	-Abu Nowshed Chy -Md Zia Ullah -Masaki Aon [Chy, 2015]
Combining	Ils ont proposé trois méthodes pour	-Fraicheur	-Taiki

Recency and Topic-Dependent Temporai Variation for Microblog Search	l'expansion temporale de la requête. Deux méthodes individuelles basées sur la variation temporelle et la fraicheur (TVQE et TRQE) et leur combinaison (TVRQE) pour surmonter les limites des méthodes individuelles.	-Estimations de la densité du noyau(KDE)	Miyanishi - Kazuhiro Seki -Kuniaki Uehara. [Miyanishi, 2013]
Incorporating Temporal Informationin Microblog Retrieval	Ils ont proposé trois méthodes pour la recherche temporelle des tweets, la première favorise les termes récents ayant une cooccurrence élevée avec tous les termes de la requête, la deuxième favorise les tweets pertinents qui appartiennent aux périodes de grande concentration des tweets, la troisième favorise les termes qui appartiennent à des tweets pertinents qui figurent dans les grandes concentrations des tweets et qui ont une occurrence élevée avec tous les termes de la requête.	-Peak-Finding - Fraicheur	-Willis -Medlin -Arguello [Willis, 2012]
Temporal Feedback for Tweet Search with Non-Parametric Density Estimation	Ces derniers ont hypothèse qu'il existe une densité fq au cours du temps de corpus, de sorte que fq est grand pour les moments où les documents pertinents sont susceptibles d'apparaître et de petits dans le cas inverse. Alors pour promouvoir les tweets dont leur temps coïncide avec	-Fraicheur -Estimations de la densité du noyau(KDE) avec Trois pondérations différentes.	-Miles Efron. Jimmy Lin. Jiyin He . Arjen de Vries. [Efron ,2014]

	une grande valeur de la densité. Ils ont utilisé la densité du kernel d'une loi normal. Comme ils ont pondéré chaque kernel par le score thématique du tweet correspondant vit à vie la requête. Se la va permettre d'amplifier la densité des régions temporelles ou figure des tweets pertinents.	-La méthode « The moving window ».	

***Tableau1** : Résumé des travaux voisin.*

Notre proposition :

- Proposer deux techniques pour la recherche des tweets pertinents temporellement à la requête, qui prennent en considération le comportement des Microblogeurs (retweet + favorite).
- Prise en considération de l'aspect sémantique de la requête.
- La dernière consiste à combiner trois évidences, temporelle, sémantique et lexicale dont le but d'améliorer le classement des tweets pertinents.

8. Conclusion :

Dans ce chapitre nous avons présenté les notions principales auxquelles nous faisons appel comme support pour la modélisation de nos propositions. Il s'agit de Twitter et celle de la sémantique et du temps et des protocoles d'évaluation des compagnes de tests. Nous souhaitons apporter des contributions pour améliorer la recherche d'informations dans les tweets en prenant en compte la sémantique et le temps.

CHAPITRE III :

Conception

1. Introduction :

Dans ce chapitre nous allons présenter une nouvelle approche pour le reclassement des Microblogs. Notre contribution est divisée en trois grandes parties: la première consiste à proposer deux techniques temporels pour la recherche des tweets pertinents à la requête, qui prennent en considération les signaux sociaux à savoir, la technique basée sur la fraicheur, et celle basée sur la concentration des tweets . Le deuxième partie dans notre contribution s'agit de prend en considération l'aspect sémantique de la requête et aussi enrichir le contenu informationnel des tweets par les titres des pages Web dont leurs adresses URLs est englobée par ces derniers. Le dernier partie consiste à combiner les évidences, temporelle et sémantique et lexicale pour le reclassement des tweets.

2. Notre approche :

Dans cette partie du chapitre, nous proposons notre contribution pour la recherche temporelle et sémantique des tweets dans le corpus TREC 2011. L'idée consiste à définir un mécanisme pour générer des nouveaux classements des résultats, à base temporelle, sémantique et de leur combinaison avec le score lexical et cela pour améliorer le classement des tweets similaire sémantiquement ou pertinent temporellement a la requête.

Le calcul de la pertinence temporelle s'agit d'utiliser la date de soumission de la requête et la date de publication du tweet pour le calcul de son score de pertinence temporelle. Alors pour les requêtes sensibles aux tweets importantes et récentes, nous avons proposé la méthode fraicheur afin de sélectionner les tweets proches temporellement de la requête. Tandis pour les requêtes qui vise les tweets soumis suit à des événements importants dans le passé, nous avons proposé la méthode concentration.

Notre contribution par rapport aux travaux voisins consiste à introduire la pertinence sociale (le nombre de retweets, le nombre de favorite) en addition à la pertinence lexicale dans l'estimation de l'importance temporelle d'un tweet. Plus précisément nous avons contribué dans les deux travaux : le travail de [Efron, 2014] et le travail de [Li, 2003]. Les premiers ont hypothèse que les tweets pertinents ce cluster ensemble dans le temps. Alorspour trouver ces clusters temporels, ils ont estimé la distribution des tweets via la fonction densité du noyau et pour valoriser les tweets pertinents ils ont pondéré chaque noyau par l'importance

lexicale du tweet correspondant. Ceci va permettre d'amplifier les intervalles temporels où se trouvent des tweets pertinents, comme il va augmenter les scores temporels (densité) des tweets pertinents. Pour le deuxième travaillé, les autres ont supposé que la distribution des tweets récents suit une loi exponentielle et ils ont proposé une méthode qui se base sur cette loi pour les sélectionner.

Pour la pris en compte de la sémantique de la requête nous avons défini un mécanisme inspiré du travail de [Boucetta, 2017] qui prend en compte les différents sens qui peuvent être portés par la requête utilisateur et qui enrichissent le tweet par le contenu des adresses URLs figurant dans ce dernier . À cet effet, nous avons utilisé l'ontologie WordNet pour identifier les différents sens d'une requête. Cela est fait par la construction d'un 'vecteur sémantique' contenant l'ensemble des termes à pondérer et les concepts qui leur sont sémantiquement liés.

Nous avons aussi expansée chaque tweet par les titres des pages web dont leurs adresses URLs est englobée dans ce dernier, pour enrichir son contenu informationnel. Par la suite le vecteur sémantique est utilisé en association avec le modèle vectoriel pour construire 'les vecteurs tweets' et 'le vecteur requête' en se basant sur des coefficients calculés selon la formule de pondération 'Tf/Idf'. Ces deux vecteurs sont utilisés pour calculer le degré de similarité sémantique entre la requête et chacun des tweets du corpus.

Le reclassement selon la troisième approche consiste à combiner les trois dimensions: temporelle, sémantique et lexicale pour le calcul du score de pertinence d'un tweet résultat de la première recherche.

2.1. Fondements de l'approche proposée :

Nous présentons dans cette section les fondements théoriques sur lesquels se base notre proposition, il s'agit des caractéristiques guidant l'approche de recherche temporelle et sémantique que nous proposons. Dans cette contribution nous nous intéressons plus précisément à la recherche temporelle et sémantique des tweets. Notre système permet de :

- faire un prétraitement du corpus.
- Communiqué avec le moteur de recherche Lucene pour indexer le corpus des tweets et analyser la requête et enfin pour récupérer les tweets pertinents thématiquement a la requête.
- Calculer la pertinence sociale pour chaque tweet.
- Calculer la pertinence temporelle fraicheur ou concentration pour chaque tweet résultat de la première recherche.
- Reclasser les résultats de la première recherche, selon leur score temporel.

- Enrichir chaque tweet avec le titre de l'URL englobé dans ce dernier.
- Projeter la requête utilisateur sur une ressource linguistique WordNet et création du vecteur sémantique.
- Création des vecteurs tweets et requête.
- Reclassé la liste résultat de la première recherche, selon les scores sémantiques des tweets.
- Reclasser la liste résultat de la première recherche, selon la combinaison des trois scores sémantiques, temporelle et lexical.

Notre système est fondé sur :

- Des algorithmes temporels.
- Une ressource linguistique (Word Net).
- Un modèle de calcul vectoriel pour mesurer la pertinence 'tweet/requête'.
- Le moteur de recherche Lucene. Dans ce qui suit nous justifions nos choix.

2.1.1. Choix du modèle de recherche d'information :

Dans notre travail nous avons opté pour deux modèles de recherche d'informations. Pour extraire la première liste des tweets nous avons utilisé le modèle de recherche implémenté par Lucene qui est une combinaison du modèle vectoriel et du modèle booléen. Le rôle de la fonction de score et de déterminer la pertinence d'un document par rapport à une requête en fonction des autres documents composant l'index. L'évaluation du score de chaque document par Lucene dépend fortement de l'indexation. Des poids sont attachés à chaque document en fonction du nombre de termes, de leur présence plus ou moins importante, etc. L'indexation attribue un score à chaque champ du document et son score total est la combinaison de ces scores.

Le score d'appariement d'un document par rapport à une requête est calculé par Lucene via la formule (18) suivante :

$$\text{Score}(q,d) = \text{coord}(q,d)\,\text{queryNorm}(q) \sum_{t \text{ in } q} (\text{tf}(t \text{ in } d)\,\text{idf}(t)^2 t.\,\text{getBoost}()\,\text{norm}(t,d) \qquad (18)$$

– coord est le nombre des termes du texte recherché présents dans un document. Cette valeur est calculée par Lucene lors de la recherche.

– queryNorm est un facteur de normalisation utilisé pour rendre les scores entre des requêtes successives comparables. Ce facteur n'a aucune incidence sur le classement des documents d'une requête donnée. Cette valeur est calculée par Lucene lors de la recherche.

– TF(t,d) (Term Frequency) mesure la fréquence du terme T dans le document D. Les documents qui contiennent le plus d'occurrences d'un terme sont les plus valorisés pour ce terme. TF est une sorte de table « document x terme » contenant ces fréquences. Cette valeur est calculée par Lucene lors de l'indexation.
– IDF(d) (Inverse Document Frequency) mesure la fréquence d'un terme T dans l'espace de document. Les termes communs (apparaissant dans beaucoup de fiches) sont moins valorisés que les termes rares (apparaissant dans peu de fiches). IDF est une table des termes avec leur fréquence globale. Cette valeur est calculée par Lucene lors de l'indexation.
– Boost augmente ou diminue le score sur la recherche d'un champ donné. Globalement, ce score de recherche indique l'importance des termes de la requête dans le document en prenant en compte l'importance de ces termes dans l'espace des documents.

Pour le reclassement sémantique nous avons utilisé le modèle vectoriel, qui préconise la représentation des requêtes utilisateurs et documents sous forme de vecteurs, dans l'espace engendré par tous les termes. De manière formelle, les tweets et requêtes sont des vecteurs dans un espace vectoriel de dimension *N*.
Le choix du modèle vectoriel est principalement motivé par sa simplicité. Plus précisément dans notre cas, le modèle vectoriel est basé sur un vecteur sémantique des termes. Ce vecteur sémantique est le résultat de la projection sémantique de la requête sur l'ontologie WordNet. Ce modèle nous a permis donc de construire « le vecteur requête» et « les vecteurs documents» à base des coefficients calculés à l'aide d'une fonction de pondération. Il était également la base pour mesurer la similarité entre le vecteur de la requête et ceux des tweets en utilisant une fonction de calcul de similarité entre vecteurs. Le mécanisme de pondération des termes et les mesures de similarité utilisées en association avec ce modèle sont les suivants :

a-Pondération des termes : elle permet de mesurer l'importance d'un terme dans un tweet.Dans ce contexte, plusieurs techniques de pondération ont vu le jour, la plupart d'entre elles sont basées sur les facteurs « tf» et « idf», qui permettent de combiner les pondérations locale et globale d'un terme :
- tf (Term Frequency) : cette mesure est proportionnelle à la fréquence du terme dans le tweet (pondération locale). Elle peut être utilisée telle quelle ou selon plusieurs déclinaisons (log(tf), présence/absence,. . .).

- idf (Inverse of Document Frequency) : ce facteur mesure l'importance d'un terme dans toute la collection (pondération globale). Un terme qui apparaît souvent dans la base documentaire ne doit pas avoir le même impact qu'un terme moins fréquent. Il est
généralement exprimé comme suit : log (N/df), où df est le nombre de documents contenant le terme et N est le nombre total de documents de la base documentaire.
La mesure « tf * idf» donne une bonne approximation de l'importance du terme dans le document, particulièrement dans les corpus de documents de taille homogène. Cependant, elle ne tient pas compte de l'aspect longueur c'est pour cette raison nous avons utilisé la (formule 29).

b-Mesure de similarité : la mesure de similarité d'un tweet à une requête est calculée par notre système selon la mesure de distance dans un espace vectoriel (formule 30).

2.1.2 Choix de la ressource linguistique :

Dans notre travail nous avons opté pour l'ontologie WordNet vue s'a disponibilité sur le Web et s'a richesse. WordNet est un réseau thématique électronique développé depuis 1985 à l'université de Princeton par une équipe de psycholinguistes et de linguistes du laboratoire des sciences cognitives [Bouramoul ,2011]. L'avantage de WordNet réside dans la diversité des informations qu'elle contient (grande couverture de la langue anglaise, définition de chacun des sens, ensembles de synonymes, diverses relations sémantiques). En outre, WordNet est librement et gratuitement utilisable.

WordNet couvre la majorité des noms, verbes, adjectifs et adverbes de la langue Anglaise,qu'elle structure en un réseau de noeuds et de liens. Les noeuds sont constitués par des ensembles de termes synonymes (appelés *synsets*).Un terme peut être un mot simple ou une collocation (i.e. deux mots ou plusieurs mots reliés par des soulignés pour constituer le mot composé) [Bouramoul ,2011]. Les concepts de WordNet sont reliés par des relations sémantiques. La relation de base entre les termes d'un même synset est la synonymie. Les différents synsets sont autrement liés par diverses relations sémantiques telles que la subsomption ou la relation d'hyponymie hyperonymie, et la relation de composition méronymie-holonymie. Ces relations sont formellement définies comme suit:

a.La relation taxonomique (ou relation de subsomption), dite relation
d'Hyperonymie/Hyponymie: X est un hyponyme de Y si X est un type de (kind of / isa) Y. Y est alors dit hyperonyme de X.

Exemple : {canine} a pour hyponymes {wolf, wild dog, dog} .

b.La relation d' Holonymie et son inverse la Méronymie :

X est un méronyme de Y si X est une partie constituante (part of), substance de (substance of) ou membre (member of) de Y. Y est alors dit un homonyme de X.

Exemple : {car} a pour méronymes {wheel, engine, …}

2.1.3. Choix des formules temporelles :

a.La sélection des tweets récents :

Plusieurs travaux dans le domaine de recherche d'informations, ont hypothèse que les microblogs récent et important thématiquement sont les plus pertinents pour une requête [Efron, 2011]. Vu que les utilisateurs de Twitter s'intéressent aux news et aux événements récents. Dans la littérature plusieurs méthodes ont était proposée pour sélectionner ces documents parmi ces méthodes celle proposée dans [Li, 2003], les auteurs supposent que la distribution des tweets récent vit à vie une requête suit une loi exponentielle et ils ont proposé la formule (19) pour l'estimation de la pertinence fraicheur des documents.

$$PR = re^{-r|Dq-Dt|} \quad \ldots\ldots\ldots \quad (19)$$

Dt : représente la date de publication du tweet t, Dq : la date de soumission de la requête Q.

r : le taux de la distribution exponentielle. Dans notre travail nous avons opté pour cette formule.

b. La sélection des tweets qui figure dans les Bursts (concentration) :

Pour l'estimation de la pertinence temporelle des tweets qui figue dans les grandes concentrations des tweets. Nous avons opté dans notre travail pour la Densité du Kernel. La méthode du Kernel est une généralisation de la méthode d'estimation par histogramme.

Dans un histogramme, la densité en un point x est estimée par la proportion d'observations x_1, x_2, ..., x_N qui se trouvent à proximité de x. Pour cela, une boîte en x est tracer dont la largeur est gouvernée par un paramètre de lissage h ; ensuite le nombre d'observations qui appartiennent à cette boîte est compter. Cette estimation, qui dépend du paramètre de lissage h, présente de bonnes propriétés statistiques mais elle est par construction non-continue. [Efron , 2014]

La méthode du noyau consiste à retrouver la continuité : pour cela, la boîte est remplacé par une gaussienne centrée en x et de largeur h . Plus une observation est proche du point de support x plus la courbe en cloche lui donnera une valeur numérique importante. À l'inverse, les observations trop éloignées de x se voient affecter une valeur numérique

IBM OmniFind Yahoo ! Edition, oméga, Terrier, Zettair. Ce sont des bibliothèques de recherche évolutive pour la recherche de texte intégral. Ce sont une base solide, sur laquelle une application de recherche peut être développée. Chaque moteur parmi ces derniers possède des points forts et des points faibles. Si on veut par exemple un meilleur temps de réponse on peut choisir: Indri, IXE, Lucene, XMLSearch. Si on veut des meilleures performances d'indexation Zettair est le Top pour cette tâche. Les meilleurs temps d'indexation sont réalisés par: ht://Dig, Indri, IXE, Lucene, MG4J, Swish-E, Swish++, Terrier, XMLSearch, Zettair .

Dans notre application nous avons utilisé Lucene, vu ces performances et vue qu'il est le moteur le plus utiliser dans la tâche Microblogs de TREC 2011.Les trois rôles de Lucene comme schématiser dans la figure 16, pour n'importe quelle application de recherche du texte réside dont : - indexé le corpus.- Analysé la requête.- Recherche par indexe. -affichage des résultats

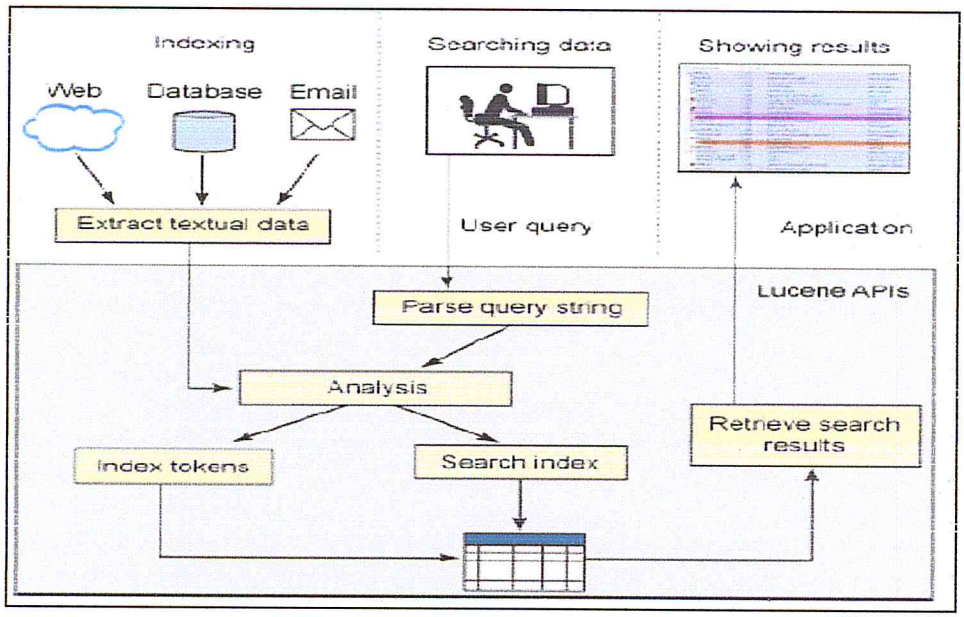


Figure 16 : *L'architecture de L'API Lucene.*

3. Schéma global de notre approche :

Dans cette section nous commencent par détailler la méthode adoptée pour collecter le corpus de test, puis nous détaillons l'architecture de notre système qui est la concaténation d'un ensemble de modules qui se complètent pour améliorer la recherche d'informations dans les tweets. Le score temporal de chaque tweet est calculé selon le type temporel de la requête. La similarité sémantique entre la requête et chaque tweet est calculée pour favoriser les tweets proches sémantiquement de la requête. Notre système combine aussi les évidences

temporelles, sémantiques et lexicales pour reclasser les tweets. Ceci permettra d'avoir plus de documents pertinents dans le top de la liste résultat de la recherche. La vue d'ensemble de notre architecture est représentée dans la figure 17.Par la suite nous détaillons les différents modules de notre architecture.

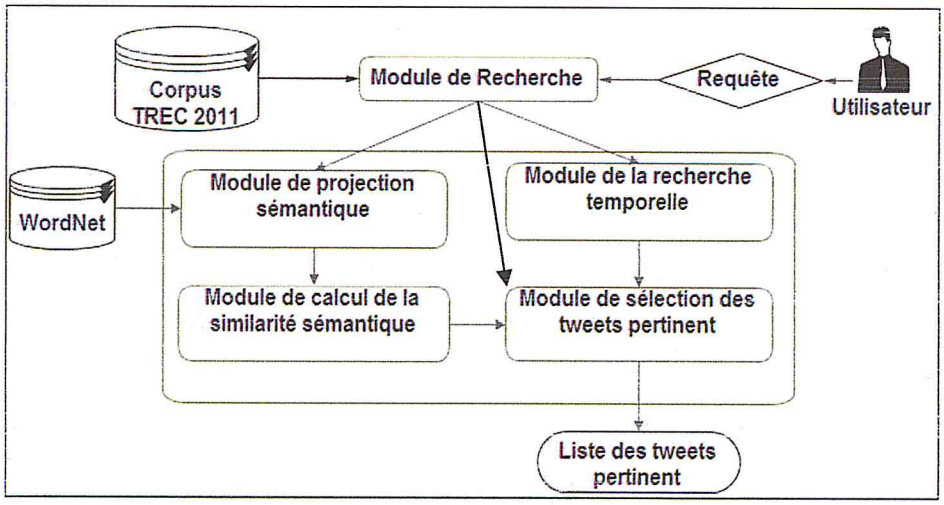


Figure 17 : *L'architecture générale de notre système*

3.1. Collection des données :

Cette phase est la tâche la plus lente. Elle consiste à récupérer l'ensemble des tweets du corpus TREC2011.Le corpus des tweets utilisé dans la piste Microblog de TREC 2011, est distribué sous forme de 15 répertoires, chacun contenant environ 100 fichiers .DAT, chacun contenant une liste de (tweet id, username, MD5 checksums) . Chacun de ces fichiers est appelé bloc d'état (c'est-à-dire bloc de tweets). Le programme de téléchargement (robot d'exploration) de bloc d'état, fonctionne en récupérant les tweets indiqués dans les fichiers .DAT depuis twitter.com. La combinaison de tweet id et de nom d'utilisateur correspond directement à une URL, qui peut être récupérée pour le téléchargement des tweets. Pour le téléchargement du corpus nous avons suivi les étapes suivantes[12] :

- Nous avons créé le dossier "DATA" dans le dossier twitter tools core.
- Nous avons exécuté les deux commandes suivantes sous Unix selon le nombre des blocs qui est 150 : **# Mvn clean package jar : jar appassembler : assembler.**

 # Sh target / appassembler/ bin /AsyncHTML Status Block Crawler -data nom de fichier- out put Json/ nom de fichier.Json.gz.

[12] http://trec.nist.gov/data/tweets/

La stemmatisation est le processus d'élimination le suffixes des mots afin d'obtenir leur racine commune. Cela permet de générer la forme de base (souvent tronquée) appelée le stem (Racine en français). Par exemple :{computer, computing, computation} devient « comput »[13].

- Suppression des tweets écrits avec un langage autre que l'anglais.
- Suppression des tweets dont le code d'état (erreur) est égal à 403 ou 404 et 302 et Spam.
- Suppression des fichiers vides.
- Suppression des retweets.

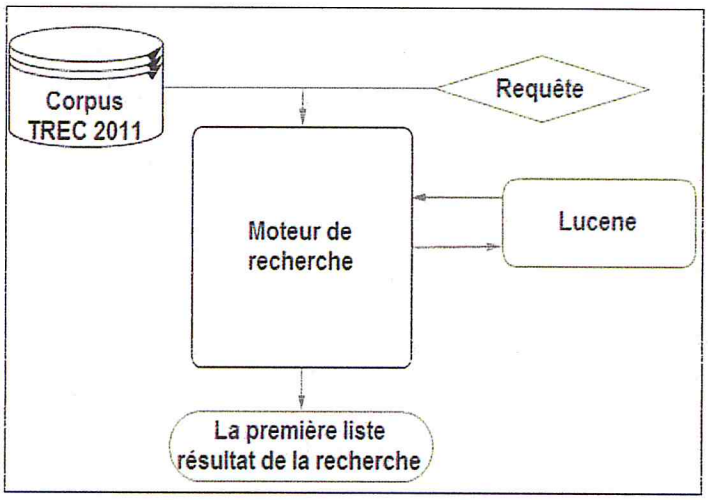


Figure 18 : *Module de recherche*

Après l'épuration du corpus, le module de recherche (figure 18) transmet la requête de l'utilisateur au moteur de recherche, qui analyse la requête puis lance une recherche par index dans le corpus des tweets. Par la suite le module de recherche récupère la liste des documents triés selon leur score de pertinence calculé par Lucene.

3.2.2. Le module de calcul du score temporelle :

Notre idée consiste à exploiter l'aspect temps afin de reclasser les résultats de la première recherche. Pour ce là nous avons proposé un modèle temporel inespéré du travaille de [Li, 2003]. Notre modèle exploite la distance temporelle entre les dates de soumissions de la requête et du tweet pondéré avec certaines propriétés sociales et thématique du tweet, pour estimer la pertinence d'un document vis à vie la requête. La recherche des tweets par la prise

[13] https://stackoverflow.com/questions/9756653/porter-stemmer-code

en compte du temps (figure 19), consiste à estimer leurs importances selon leurs positions temporelles vise à vie la requête (récente, ou bien figurant dans les grandes concentrations des tweets). Nous avons proposé deux techniques pour le calcul du score de pertinence temporelle d'un tweet, que nous détaillons par la suite. Notre contribution majeure par rapport aux travaux voisins consiste a introduire l'importance social et l'importance thématique dans un modèle temporelle.

Nous définissent l'importance sociale d'un tweet selon les deux signaux sociaux suivants :

- le nombre de retweets (le nombre de fois que le tweet a été retweet par les abonnés).
- Favorisé (le nombre de fois que le tweet a été favorisé).

Tendit que l'importance thématique s'agit du score de similarité calculée par Lucene.

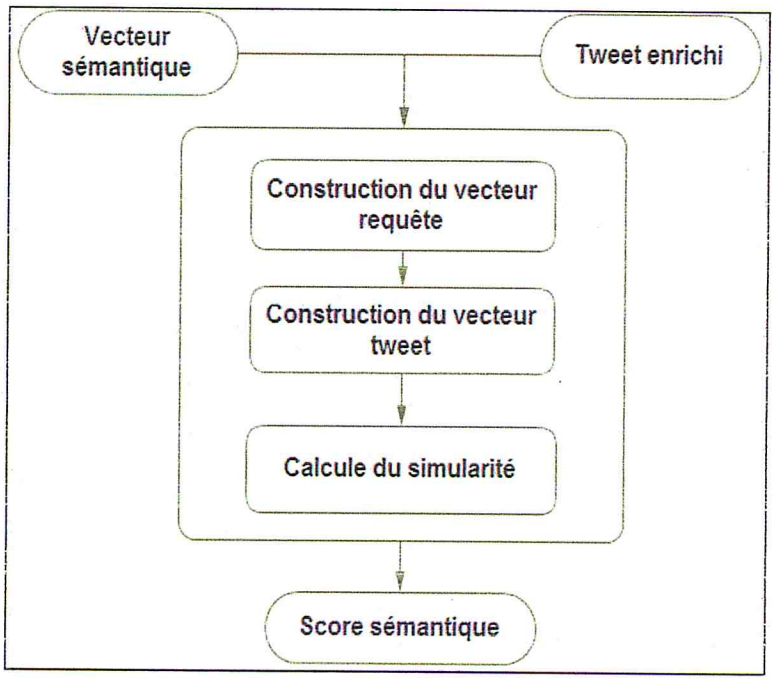


Figure 19 : *Module de calcule du score temporelle.*

a- Le score social :

Vu la spécificité lexicale des microblogs (longueur court ne dépasse pas 140 caractères et le langage ambigu). Seules les propriétés liées au contenu ne pouvant pas qualifier la pertinence d'un tweet vis à vie une requête [Damak, 2014]. Les propriétés sociales selon des travaux récents [Chy, 2015] jouent un rôle important pour évaluer la pertinence d'un tweet. il y a une multitude de propriétés sociales voir [Damak, 2014] [Choi, 2012] [Han, 2012][Duan, 2010]. Dans notre travail nous avons considéré deux propriétés, détaillé par la suite :

– Favori [Damak, 2014]: elle désigne le nombre de fois qu'un microblog a été choisi dans les listes de favoris des autres utilisateurs (voir chapitre 2) ainsi que l'ensemble des utilisateurs qui l'ont sélectionné. Dans Twitter, on peut connaître le nombre de fois qu'un tweet a été favorisé.

– Retweet [Damak ,2014][Choi,2012] : le mécanisme de rediffusion permet aux utilisateurs de partager de nouveau des microblogs qu'ils trouvent intéressants parmi les microblogs publiés par leurs amis. Dans Twitter, on peut connaître le nombre de fois qu'un tweet a été rediffusée.

Nous proposent la formule linéaire (formule 8) pour estimer la pertinence sociale d'un tweet.

$$W= \frac{Nbr}{NbrT} + \frac{FV}{FVT} \text{.....(25)}$$

Nbr : le nombre de retweet pour un tweet.

NbrT: le nombre total de retweet pour tous les tweets de la première liste résultat de la recherche.

FV : le nombre de fois qu'un tweet a été favorisé.

FVT : le cumul de la valeur de l'attribut favorite pour tous les tweets de la première liste résultat de la recherche.

b- Le score fraicheur :

Pour les requêtes sensibles aux documents récents, nous avons proposé la méthode de reclassement selon la fraicheur et la pertinence du tweet. Pour sélectionner les tweets, à la fois récent et pertinent et non seulement récents. Notre contribution consiste à paramétrer le modèle de reclassement selon la fraicheur proposée dans [Li, 2003] avec les deux pertinences thématiques et sociales. L'estimation de la pertinence fraicheur d'un tweet revient à calculer sa proximité temporelle de la requête, pondérée par son importance sociale et son importance thématique. Le score fraîcheur du tweet t pour une requête Q, est calculé comme suit:

$$SPT1(Q,t) = \frac{1}{\sigma} * e^{-\frac{|Dt-Dq|}{\sigma}} * (W + S(t)) \text{ ..... (26)}$$

$$S(t) = \frac{SL(t)}{\sum_{i=1}^{n} SL(ti)} \text{.....(27)}$$

Dt : représente la date de publication du tweet t et **Dq:** la date de soumission de la requête Q.

σ : est le facteur qui permet de modifier le degré d'amplification des scores

(**σ** = 1/ l'écart-type (temps)).

SL(t) : c'est le score de Lucene du tweet t.

les ti: sont les tops des tweets résultats de la première recherche.

S(t) : la pertinence thématique du tweet t,

$\Sigma_{i=1}^{n} SL(ti)$: C'est la somme des scores thématiques des tweets résultats de la première recherche.

c- Le score concentration temporelle :

Nous proposent dans notre travail la pertinence temporelle concentration pour favoriser les tweets pertinents qui figurent dans les grands clusters temporels. Plus précisément nous contribuons dans le travail de [Efron, 2014], ces derniers ont hypothèse où Il existe une densité fq au cours du temps de corpus, de sorte que fq est grand pour les moments où les documents pertinents sont susceptibles d'apparaître et de petits dans le cas inverse. Alors pour promouvoir les tweets dont leur temps coïncide avec une grande valeur de la densité. Ils ont utilisé la densité du kernel d'une loi normal. Comme ils ont pondéré chaque kernel par le score thématique du tweet correspondant vis à vie la requête. Se la va permettre d'amplifier la densité des régions temporelles ou figure des tweets pertinents. Dans notre travail nous avons introduit un deuxième facteur pour l'amplification, c'est le facteur social vu que la pertinence thématique toute seule ne peut pas estimer la pertinence d'un tweet [Damak, 2014]. Nous avons défini la pertinence sociale comme la concaténation de deux propriétés sociales le nombre de retweets et le nombre de fois qu'un tweet a été favorisé. La formule 28 résume notre contribution.

$$SPT2(Q,t) = \frac{1}{nH}\sum_{i=1}^{n} K\left(\frac{x-xi}{H}\right) * \left(w + S(t)\right) \dots\dots\dots\dots\dots (28)$$

3.2.4. Module d'enrichissement des tweets avec les titres des pages web :

Pour remédier au problème du vocabulaire restreint des tweets, qu'il influence négativement le processus du calcul de la similarité sémantique, nous avons enrichi chaque tweet (voir figure 20), par les titres des adresses URLs englobé dans ce dernier. Ceci va lui donner plus du contexte.

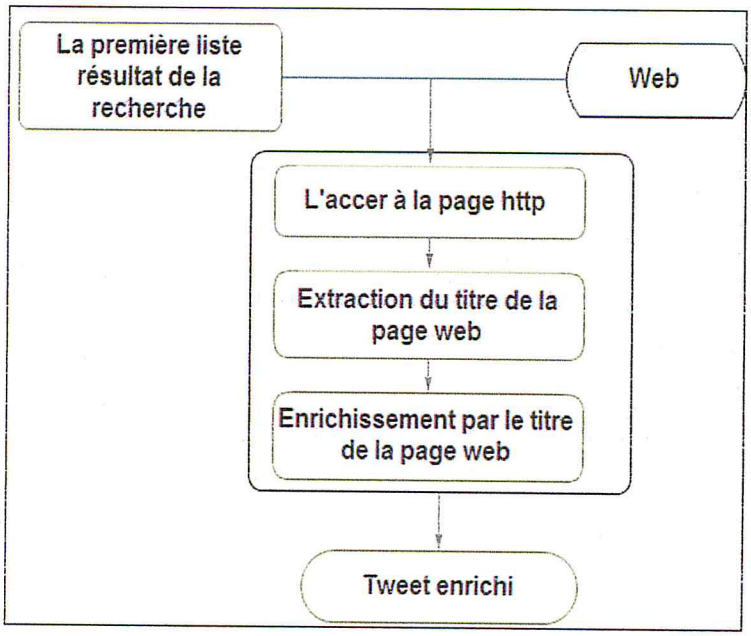


Figure20 : *Module d'enrichissement des tweets par les titres des pages web.*

3.2.5 Module de projection sémantique :

Afin de prendre en compte la sémantique de la requête Q pour favoriser les tweets similaires sémantiquement avec Q, nous associons à chaque terme de la requête l'ensemble des mots qui lui sont sémantiquement liés. L'idée est de projeter les termes de la requête sur les concepts de l'ontologie WordNet en utilisant les deux relations sémantiques : 'Synonymies' et 'Hyperonymies' pour extraire les différents sens de la requête. Par la suite l'ensemble des concepts récupérés pour chaque terme sont utilisés en conjonction avec le terme lui-même lors de la pondération par le module de calcul. L'objectif est de favoriser un tweet qui contient des mots sémantiquement proches à ceux que l'utilisateur cherche, même si ces mots n'existent pas comme termes dans la requête. Nous utilisons, à cet effet, l'ontologie WordNet selon le principe suivant : au départ nous accédons à la partie de l'ontologie contenant les concepts et les relations sémantiques, ces derniers sont utilisés pour récupérer tous les Synsets et Hyperonymies relatifs à chacun des termes de la requête. Ces derniers sont utilisés pour la construction du vecteur sémantique qui contient pour chaque terme de la requête, les synonymes et les hyperonymes appropriés. La figure 21 illustre le fonctionnement du module de projection sémantique.

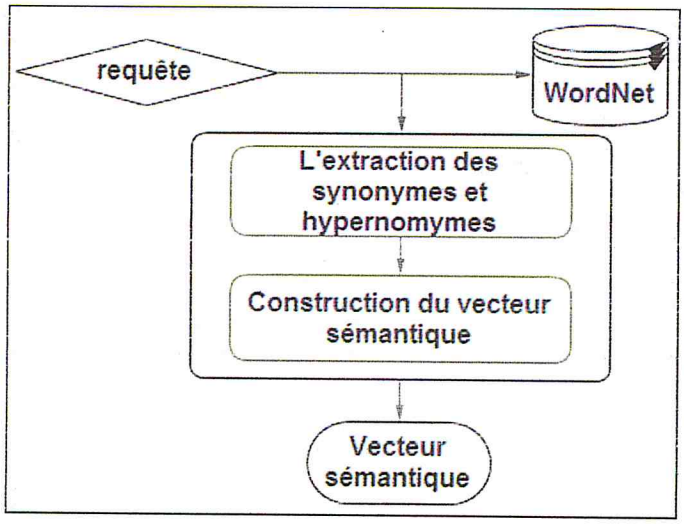


Figure 21 *: Module de projection sémantique.*

3.2.6. Module de calcul de la similarité sémantique :

Une fois que le vecteur sémantique est construit, par le module de projection sémantique, le module de calcul (figure 22) procède à la construction des vecteurs tweets et du vecteur requête à base des coefficients calculés à l'aide de la fonction de pondération appropriée (formule 29). Le module de calcul mesure par la suite la similarité sémantique entre ces deux vecteurs en utilisant la fonction de calcul de similarité entre deux vecteurs (formule 30).

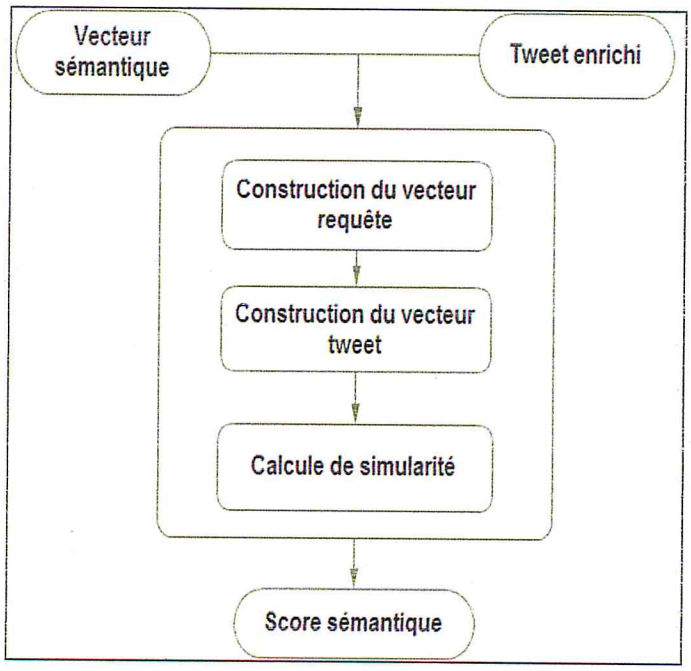


Figure 22 : Module de calcul du score sémantique.

Le fonctionnement de ce module est réalisé donc en deux étapes, les formules utilisées sont proposées dans [Salton, 71]:

a-Pondération des termes : Cette étape calcule le poids de chacun des termes du tweet. Elle se déroule comme suit : Un coefficient t_{ji} du vecteur tweet Tj mesure le poids du terme i dans le tweet j. La formule de pondération des termes des tweets est la suivante :

$$t_{ji} = \frac{occ(w)}{card(t_j)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (29)$$

occ(w) : c'est le nombre d'occurrences du terme w dans le tweet , **card(t_j)** est le nombre de termes du tweet t_j.

Un coefficient q_{ki} du vecteur requête Q_k mesure le poids du terme w dans tous les tweets.

b-Appariement « tweets/requête» : pour le calcul de la similarité entre le vecteur tweet et le vecteur requête nous avons projeté ce problème sur l'espace vectoriel et nous avons quantifié la proximité sémantique via le cosinus entre les deux vecteurs. La proximité d'une requête à un tweet est donnée par:

$$SS(Q_k, T_j) = \sum_{i=1}^{M} |q_{ki} - t_{ji}| \ldots\ldots\ldots\ldots\ldots\ldots\ldots (30)$$

M : est le nombre des termes d'un document, q_{ki}-t_{ji} : est la différence entre le poids d'un terme q_{ki} de la requête et le poids du terme de même rang du tweet t_{ji}.

3.2.7. Sélection des tweets pertinents :

Pour prendre en considération les différentes sources d'évidence (figure 23) : temporelle, sémantique et lexicale, pour le calcul de la pertinence finale de chaque tweet t pour une requête Q, nous avons proposé une formule linéaire qui cumule le score de similarité sémantique SS et le score de pertinence temporelle SPT (SPT1 ou SPT2) et le score de Lucene SL. La formule est détaillée par la suite :

$$PF(Q,t) = 0.4 * SL(Q,t) + 0.3 * SS(Q,t) + 0.3 * SPT(Q,t) \ldots (31)$$

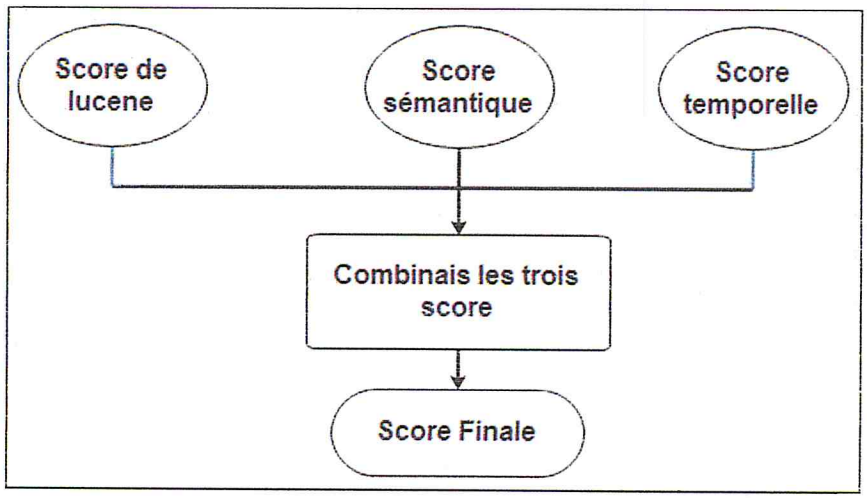


Figure 23 : *Module de combinaison des trois scores.*

Par la suite le score de pertinence final de chaque tweet est utilisé dans le reclassement de la première liste résultat de la recherche, suivant un ordre décroissant.

4. Conclusion :

Dans ce chapitre nous avons détaillé notre nouvelle approche pour le reclassement des tweets. Notre contribution est divisée en trois grandes parties: La première consiste à proposer deux techniques pour la recherche des tweets pertinents temporellement à la requête, qui prend en considération la pertinence sociale des tweets à savoir, la technique basée sur la fraicheur et celle basée sur la concentration des tweets. La deuxième consiste à enrichir là sémantiquement de la requête via WordNet et enrichir le contenu informationnel des tweets par les titres des pages Web dont leurs adresses URLs figurent dans ces derniers.

CHAPITRE IV :

Implémentation et Test

Introduction :

Dans ce chapitre, nous allons présenter la partie pratique qui constitue la mise en œuvre de notre application. Nous commençons par la présentation des outils utilisés, puis nous passons à l'illustration de l'application qui sera utilisée pour l'évaluation de l'approche proposée.

1. Présentation de l'environnement de travail :

Dans cette partie, nous présentons les outils utilisés pour l'implémentation de notre application.

1.1. Le langage de programmation Java :

Le langage Java est un langage de programmation informatique orienté objet, créé par James Gosling et Patrick Naughton employés de Sun Microsystems présenté en 1995. Il a la particularité d'être multi-plateforme, c'est-à-dire que les logiciels développés sous ce langage sont utilisables sur plusieurs systèmes d'exploitation tels que Microsoft Windows, Mac OS, Linux ainsi que sur plusieurs appareils mobiles. [Miller, 2010]

Nous avons choisi le langage Java car il convient parfaitement à l'élaboration de notre projet, vu que Lucene est implémenté en java.

1.2. Netbeans :

Afin de programmer la plateforme avec le langage Java, nous avons choisi

L'environnement de développement intégré Netbeans EDI[14] (Environnement de développement intégré) qui permet entre autre de créer des projets en langage Java. Le choix de cet IDE vient de sa simplicité d'utilisation et du nombre important de possibilités proposées par ce dernier, par exemple la génération automatique des composants graphiques et des interfaces.

1.3. WordNet :

Est une base de données lexicale développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. C'est un réseau sémantique de la langue anglaise, qui se fonde sur une théorie psychologique du langage.

[14]https://netbeans.org

La première version diffusée remonte à juin 1991. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Des versions de WordNet pour d'autres langues existent, mais la version anglaise est cependant la plus complète à ce jour.

WordNet est distribué sous une licence libre, permettant de l'utiliser commercialement ou à des fins de recherche. Nous somme utilisé version 2.1 Cette version est par ailleurs consultable en ligne. La dernière version distribuée en avril 2013 est la 3.1.

1.4. Linux :

Linux est un système d'exploitation open source (OS) basé sur UNIX créé par Linus Torvalds en 1991. Les utilisateurs peuvent modifier et créer des variations du code source, connues sous le nom de distributions, pour les ordinateurs et autres périphériques. L'utilisation la plus courante est en tant que serveur, mais Linux est également utilisé dans les ordinateurs de bureau, les smartphones, les lecteurs de livres électroniques et les consoles de jeux, etc. Nous somme choisi ubuntu 14.04 pour notre travail.

2. Les bibliothèques :

2.1. Lucene :

Lucene [15] est une bibliothèque de moteur de recherche de texte hautement évolutive disponible à partir d'Apache Software Fondation. On peut utiliser Lucene dans des applications commerciales et open source. Les API puissantes de Lucene se concentrent principalement sur l'indexation et la recherche de texte. Il peut être utilisé pour créer des fonctionnalités de recherche pour les applications telles que les clients de courrier électronique, les listes de diffusion, les recherches sur le Web, la recherche sur la base de données, etc. Des sites Web tels que Wikipedia, TheServerSide, jGuru et LinkedIn ont été alimentés par Lucene.

[15]https://lucene.apache.org

2.2 Json-simple :

JSON- simple [16] est une simple bibliothèque Java pour le traitement JSON, la lecture et l'écriture de données JSON.Json-simple utilise Map and List pour le traitement des donnees JSON. Nous pouvons utiliser json-simple pour analyser les données JSON ainsi que pour écrire le fichier JSON. L'une des meilleures fonctionnalités de json-simple est qu'il n'a aucune dépendance à l'égard de bibliothèques tierces. Json-simple est une API très légère et fonctionne bien avec les exigences JSON simples.

2.3 Jfreechart :

JFreeChart [17] est une bibliothèque open source disponible pour Java qui permet aux utilisateurs de générer facilement des graphiques et des charts. Il est particulièrement efficace pour un utilisateur qui doit régénérer des graphiques qui changent fréquemment.

2.4 Twitter4j:

Twitter propose plusieurs APIs permettant d'accéder à ses services : cela permet des opérations de consultation de comptes (tweets, listes d'amis et de followers, etc) ou des opérations de modification (supprimer des amis, poster des tweets, etc).

Twitter4j [18]est une librairie facilitant l'utilisation des API Twitter.

2.5 Stanford corenlp:

Stanford CoreNLP [19] fournit un ensemble d'outils d'analyse de langage naturel écrit en Java. Il peut prendre des entrées de texte en langage humain brut et donner les formes de base des mots, leurs parties de la parole, qu'ils soient des noms d'entreprises, de personnes, etc., normaliser et interpréter les dates, les heures et les quantités numériques, marquer la structure des phrases En termes de phrases ou de dépendances de mots, et indiquez quelles expressions de noms se réfèrent aux mêmes entités. Il a été développé à l'origine pour l'anglais, mais fournit maintenant différents niveaux de soutien pour (Modern Standard) arabe, (continent) chinois, français, allemand et espagnol. Stanford CoreNLP est un cadre intégré qui permet d'appliquer très facilement un ensemble d'outils d'analyse de langue à un texte. À partir du texte brut, nous pouvons

[16]https://www.mkyong.com/java/json-simple-example-read-and-write-json
[17]http://www.jfree.org/jfreechart/download.html
[18]http://twitter4j.org
[19]https://stanfordnlp.github.io/CoreNLP/download.html

exécuter tous les outils avec seulement deux lignes de code. Ses analyses fournissent les éléments fondamentaux pour les applications de compréhension de texte de niveau supérieur et spécifiques au domaine.

3. Format des données de trec:

Après le téléchargement de corpus tweets2011 les microblogsont le format suivant :

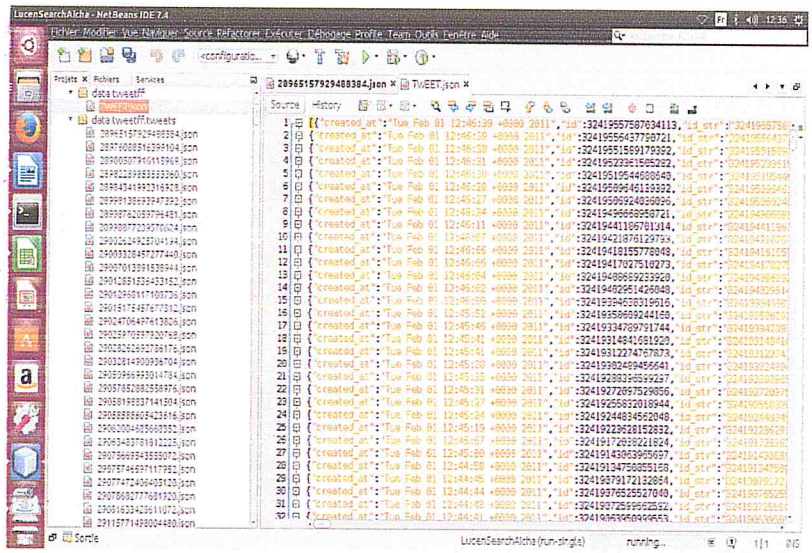


Figure 24 : *Format des données de TREC2011.*

4. Description des fonctionnalités de notre application :

4.1. L'interface principale :

Après exécution de l'application, une interface simple est affichée comportant la description de notre l'application voir figure 25.Cet interface contient trois boutons, nous les détaillons dans la section suivante :

- Recherche sémantique.

- Recherche temporelle.

- Recherche temporelle et sémantique et lexicale.

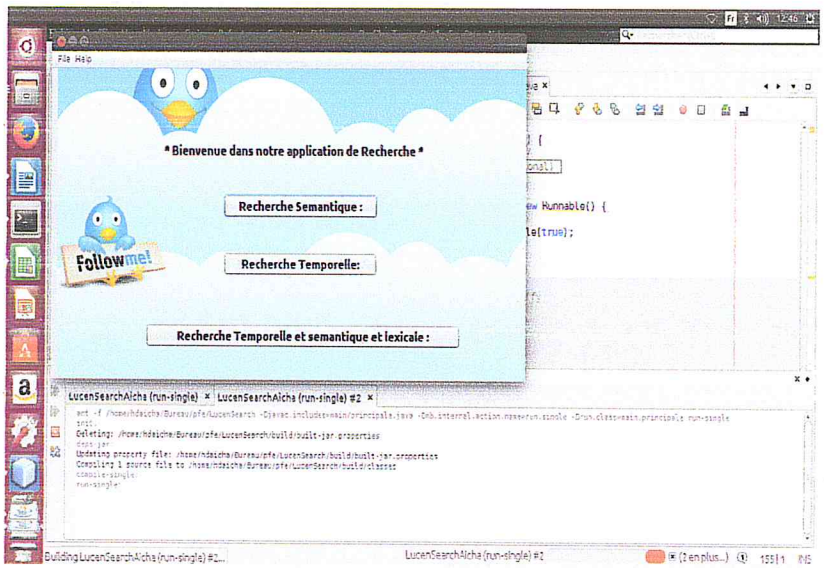


Figure 25 : *L'interface principale.*

4.2. La recherche sémantique :

L'interface principale de la recherche sémantique est illustrée dans la figure elle contient deux menus :

-«**help**» pour aider les utilisateurs à comprendre l'utilisation de l'application.

-«**File**» contient quatre sous-menus détailler par la suite:

Figure 26: *Interface d'accueil de la recherche sémantique.*

a. Chargement des tweets :

L'onglet « **Chargements des tweets** » (voir figure 27) il contient un bouton : «**Chargements des tweets**» pour charger les tweets, et un affichage pour le temps de chargement et le nombre total des tweets chargé.

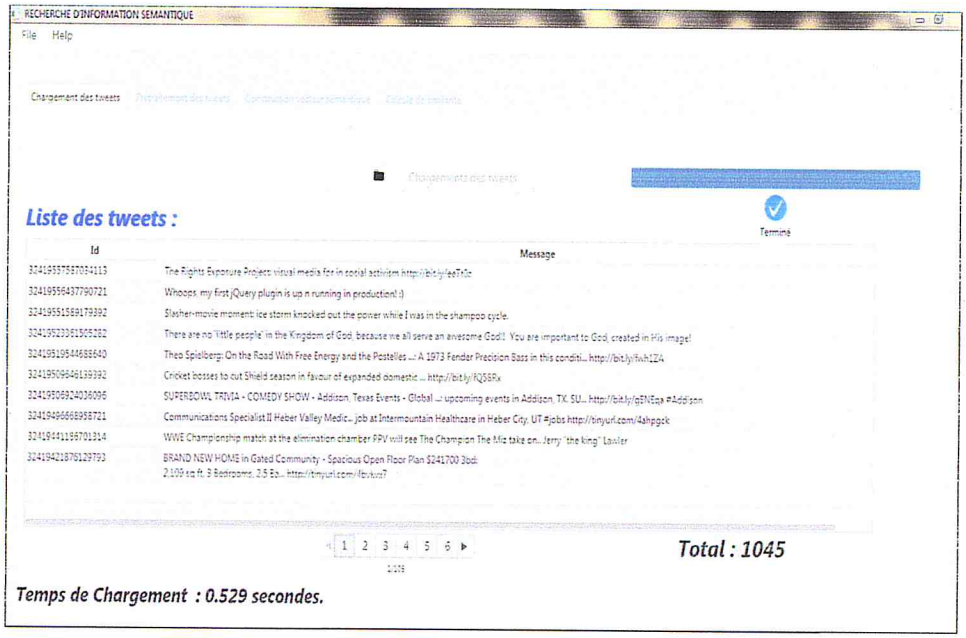


Figure 27: *L'onglet chargements des tweets.*

b. L'ongletprétraitement des tweets :

L'onglet « **Prétraitement des tweets** » (voir figure 28) contient le bouton « **Traitement des** L'interface « **Prétraitement des tweets** » contient le bouton « **Traitement des tweets** » pour lancer le prétraitement, et quatre cases à cocher, chaque case indique une opération de prétraitement a effectué (Enlever les caractères spécieux et URL(http,RT ,@,#,…..),Enlever les mots vides ,Appliquer le stem , Traiter les URLs) voir la figure suivante :

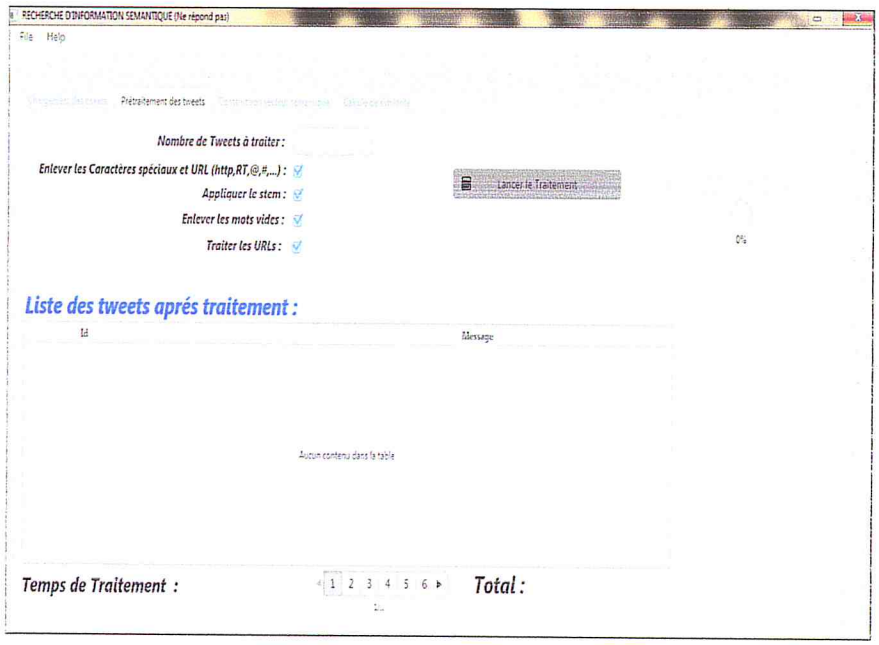


Figure 28 : *L'onglet prétraitement des tweets.*

Après qu'on coche les deux cases **« Enlever les caractères spécieux http,RT,@,#... »** et **« Enlever les mots vides »,** les prétraitements correspondants s'effectuons, et le contenu textuel des tweets résultats de l'opérationest illustré dans la figure 29 suivante :

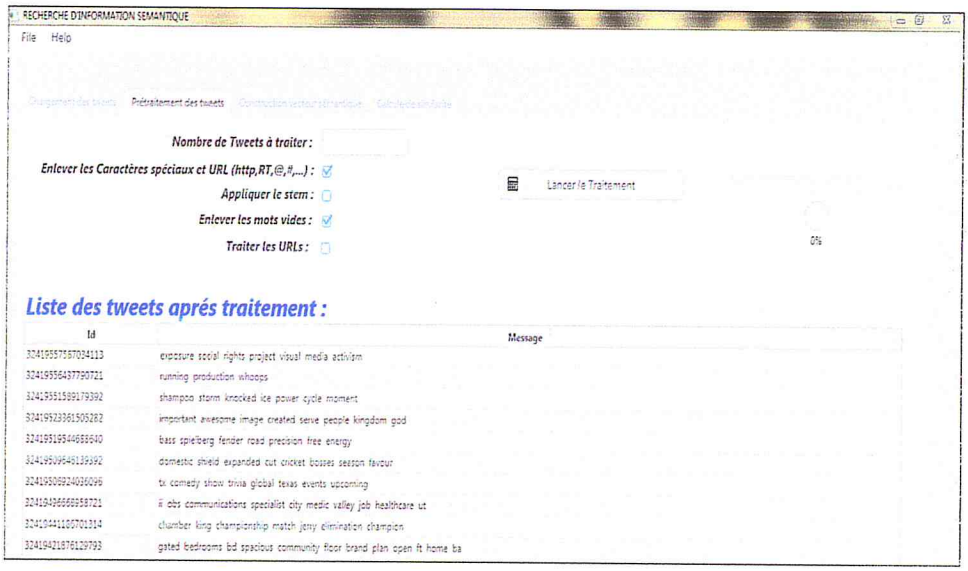


Figure 29 :*Eliminationdes caractères spécieux et les mots vide.*

La case **« Appliquer le stem »** permet de visualiser le contenu textuel des tweet après l'étape de Stemmitisation, voir la figue 30 suivante :

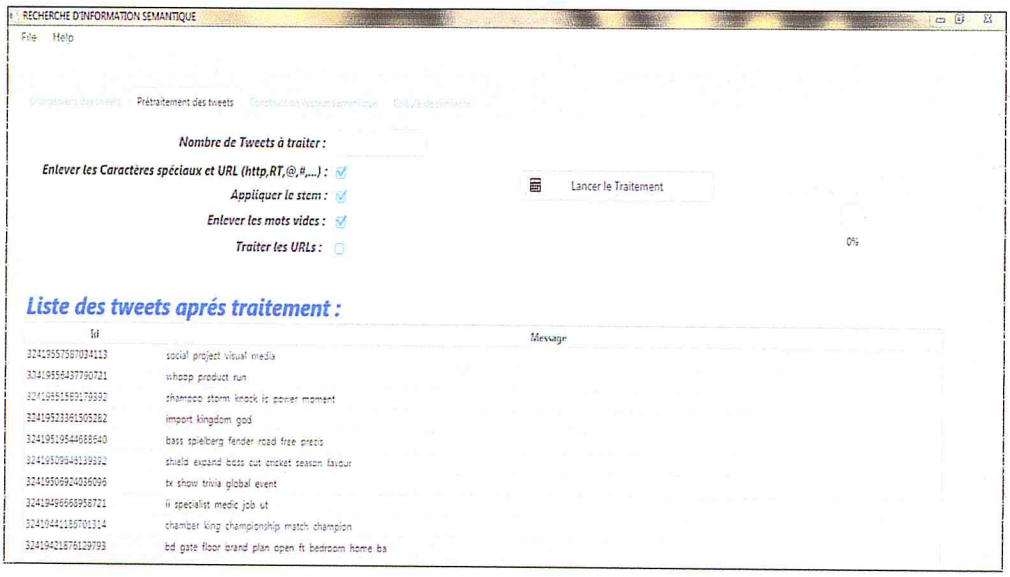


Figure 30 :*L'application de stem.*

La case « **Traiter les URLs** » permet de visualiser le contenu textuel des tweets après l'enrichissement de ces dernier par les titres des adresses URLs des pages web englobées dans ces documents, voir la figue 31 suivante :

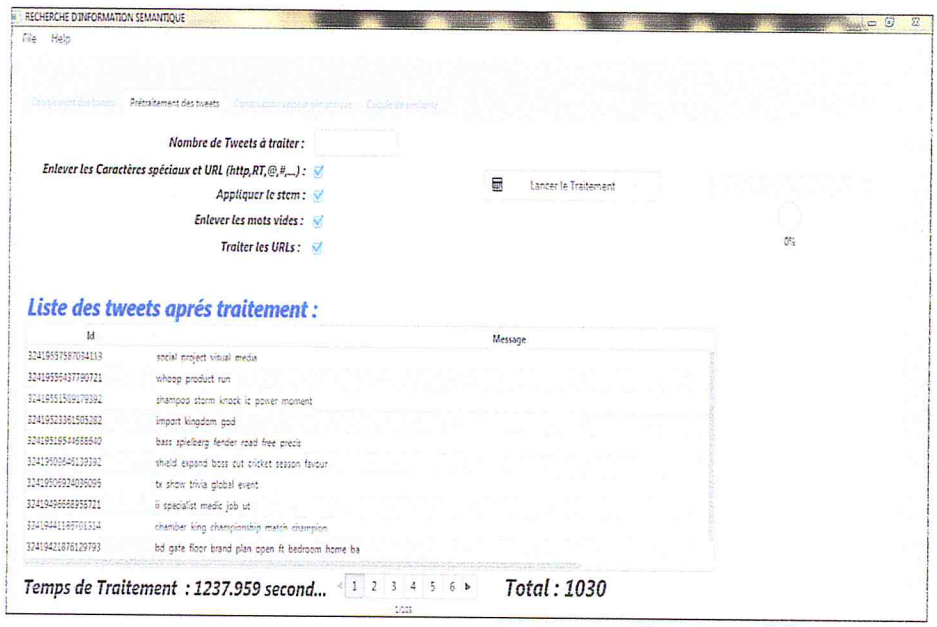


Figure 31 : *Enrichissement des tweets par les titres des pages Web.*

c.*Construction desvecteurs sémantique :*il s'agit de créer le vecteur sémantique et les vecteurs requête et tweet comme détailler par la suite (voir figure 32) :

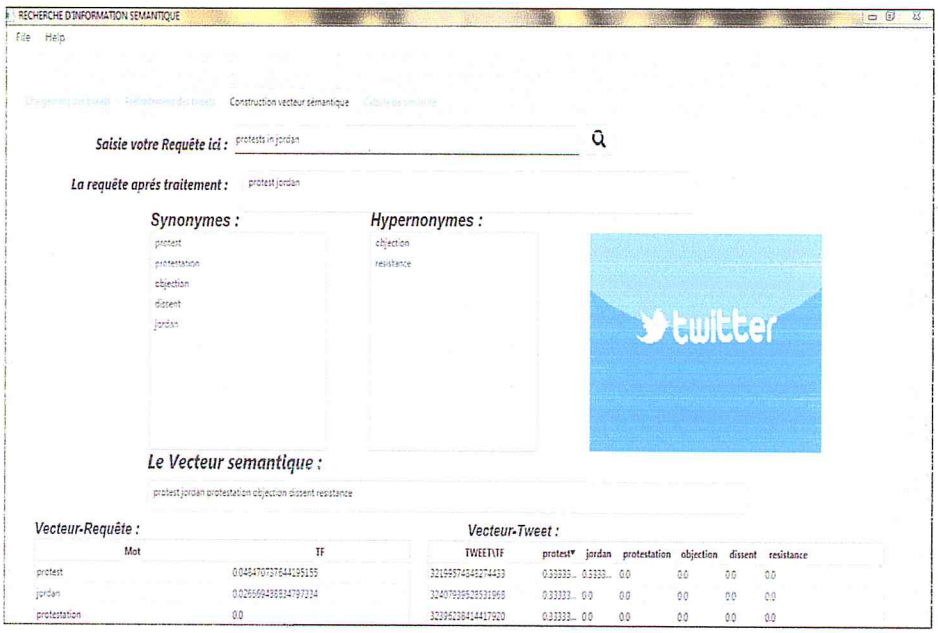


Figure 32 : *L'onglet construction desvecteurs.*

d. Construction du vecteur sémantique :le vecteur sémantique est crée à patrie de Wordnet, il contient les synonymes, les hyperonymes, la racine de chaque terme de la requête voir figure 33.

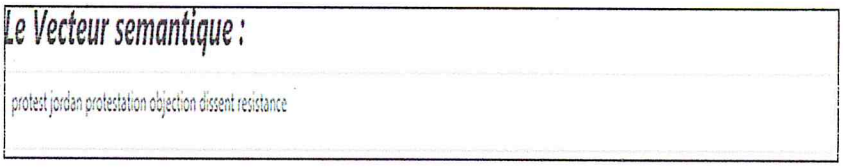


Figure 33 :*Vecteur sémantique.*

1. Construction du vecteur requête :il contient le nombre d'occurrences de chaque mot du vecteur sémantique dans tous les tweets (voir figure 34).

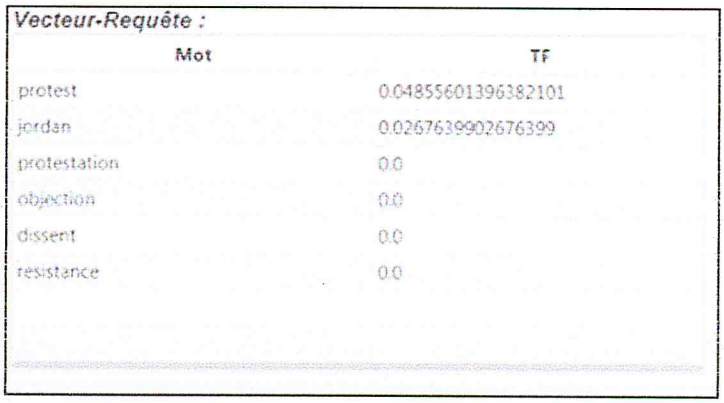


Figure 34 :*Vecteur requête.*

2. Construction du vecteur tweet : il contient le nombre d'occurrence de chaque mot du vecteur sémantique dans un tweet .

Vecteur-Tweet :

TWEET\TF	protest	jordan	protestation	objection	dissent	resistance
32199574848274433	0.33333...	0.3333...	0.0	0.0	0.0	0.0

Figure 35 : *Vecteur tweet.*

d. Calcul de la similarité sémantique :

L'ongle « **Calcule de similarité sémantique»** (voir figure 36)permet de:

- Calculé la similarité entre la requête est chaque tweet.
- Afficher la liste des tweets (id, texte, score) classées selon leurs scores de similarité sémantique.

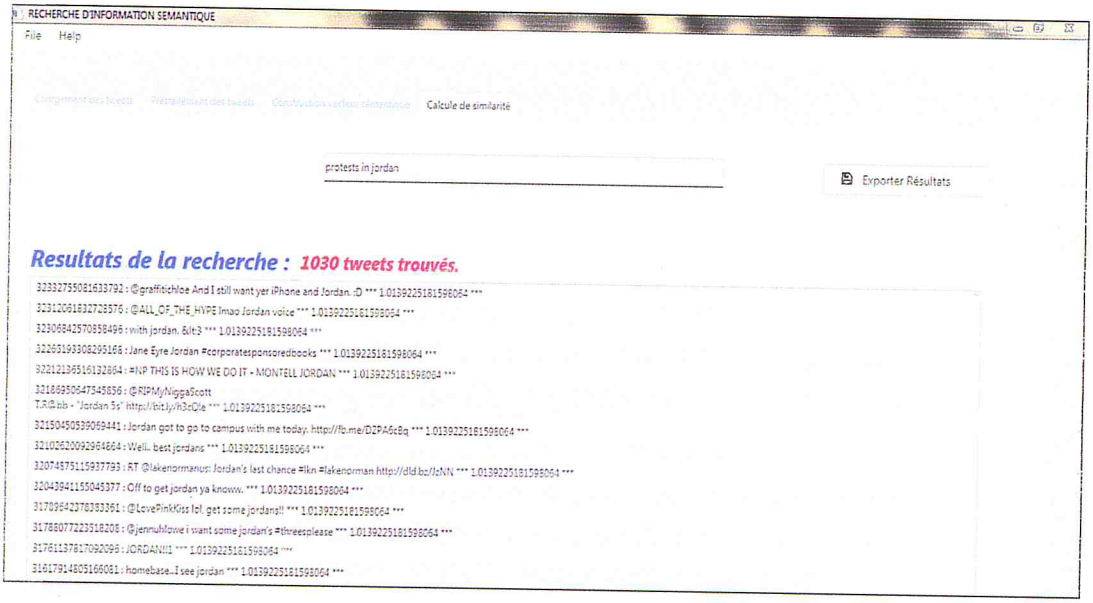


Figure 36 : *L'onglet calcul de la similarité sémantique.*

4.3-La recherche temporelle :

Dans la fenêtre recherche temporelle (voir la figure 37) on distingue deux fonctionnalités de base :

- Analyse des tweets : " **menu Analyse** ".
- Recherche des tweets : " **Menu Recherche**".

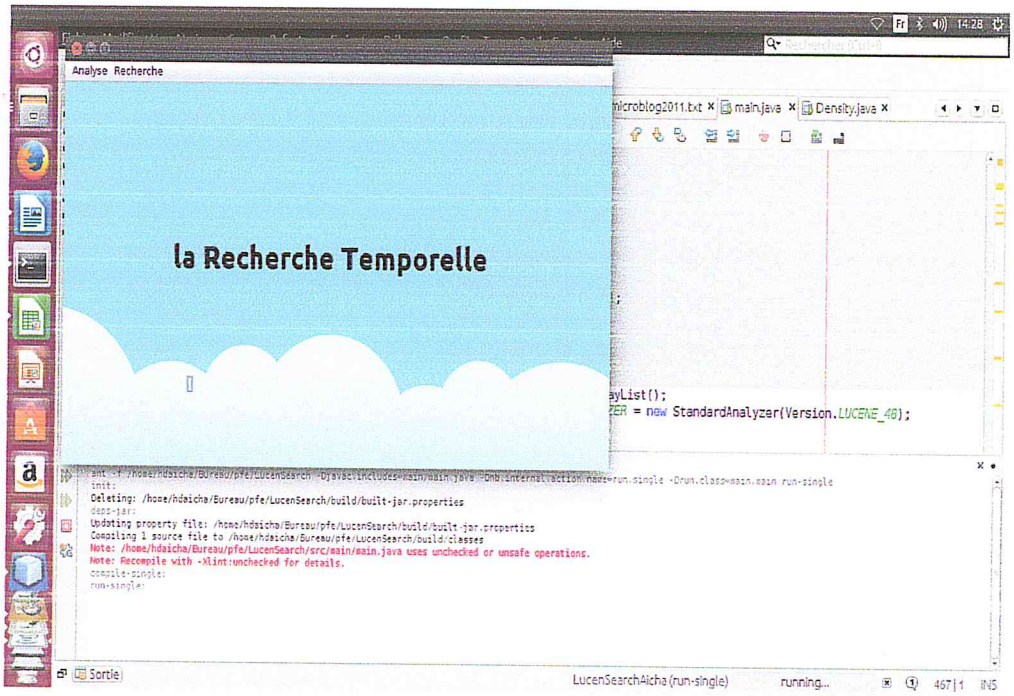


Figure 37 : *Interface principale de la recherche temporelle.*

4.3.1 Analyse des tweets :

Se fait en deux étapes détaillées par la suite, le but final est l'indexation des tweets.

a -L'extraction des tweets :

Consiste à récupérer les contenus des balises « id », « text », « created_at », « favorated_count » et « retweet_count » de chaque tweet puis les maitres dans un fichier Json vide, par la fin sauvegarder ce dernier dans la base des documents (voir figure 38a et figure 38b) .

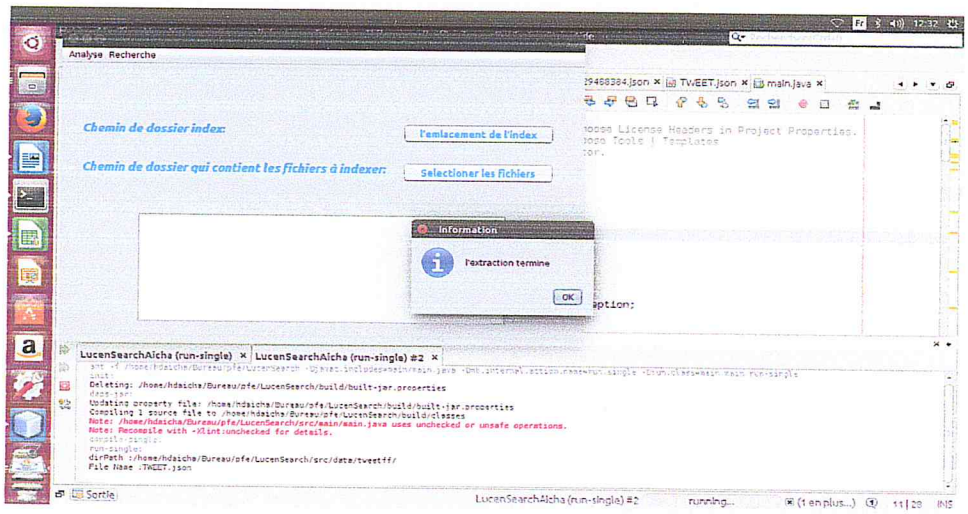


Figure 38a : *Présentation d'un tweet après l'extraction.*

{"created_at":"Sun Jan 23 00:00:06 +0000 2011",
"favorite_count":2,
"text":"Thousands demand ouster of Yemen's president http:\/\/thoora.com\/6V46Vd",
"retweet_count":0}

Figure 39b : *Présentation d'un tweet après l'extraction.*

b. Indexation des tweets :

L'indexation des tweets se fait en deux phases :

- La sélection du dossier des indexes : se fait par un simple clic sur le bouton« l'emplacement de l'index» voir figure 39.

- Puis le choix des tweets a indexé: se fait par un appui sur le bouton « sélectionner le fichier » pour sélectionner les (json) à indexer voir figure 40.

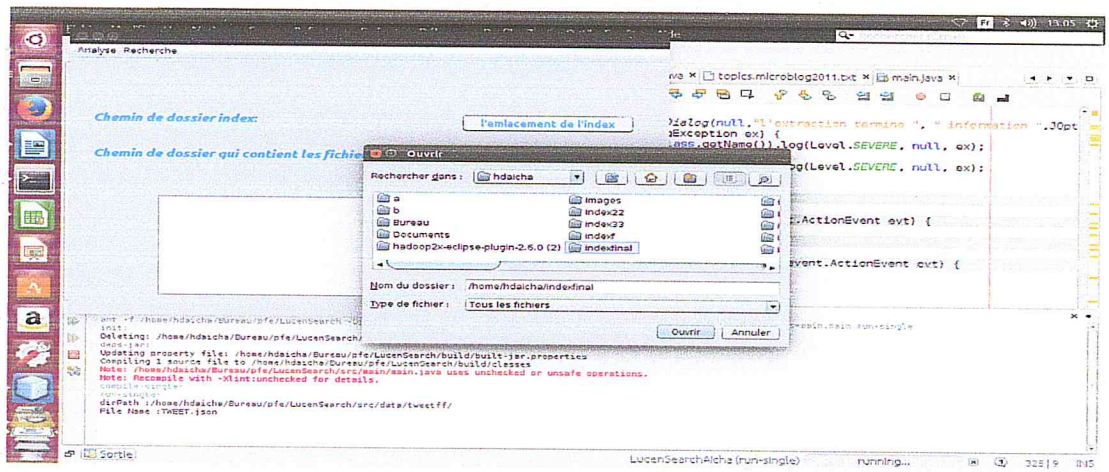


Figure 40 : *L'emplacement de l'index.*

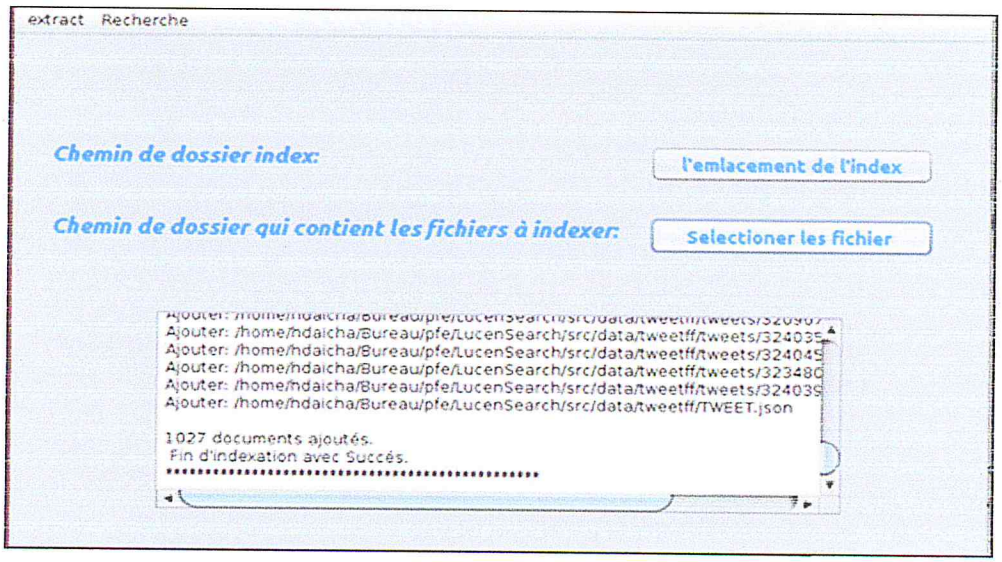


Figure 41 : *La sélection du fichier.*

4.3.2. Recherche :

Trois types de recherche sont offerts par notre système. Dans le sous menu « recherche » on peut choisir :

- Recherche via Lucene (voir la figure 41) c'est une recherche lexicale.
- Recherche selon la fraicheur des tweets (voir la figure 42).
- Recherche selon l'aspect concentration des tweets (voir la figure 43).

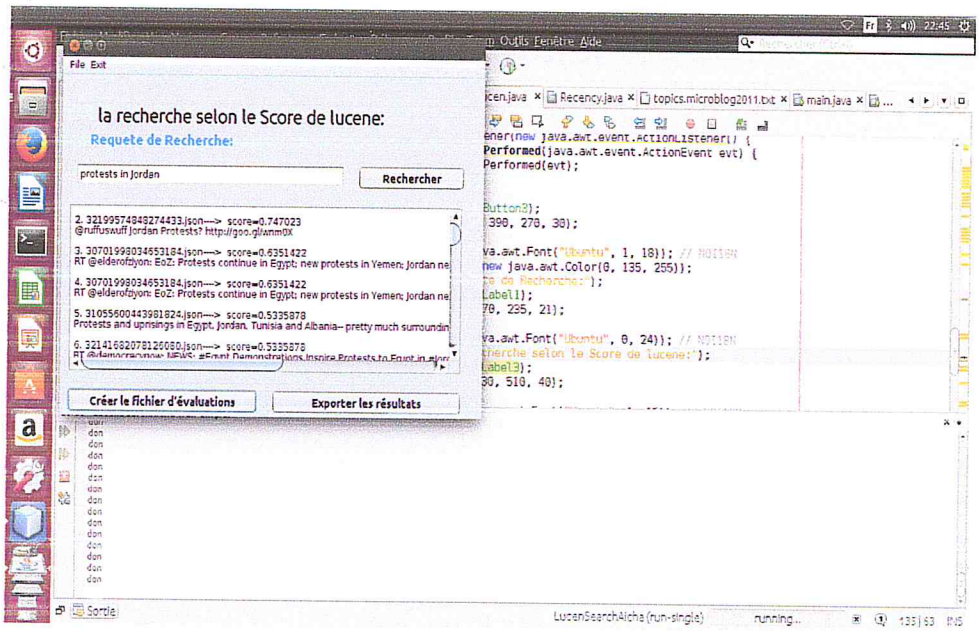


Figure 42 : *Résultat de la recherche par lucene.*

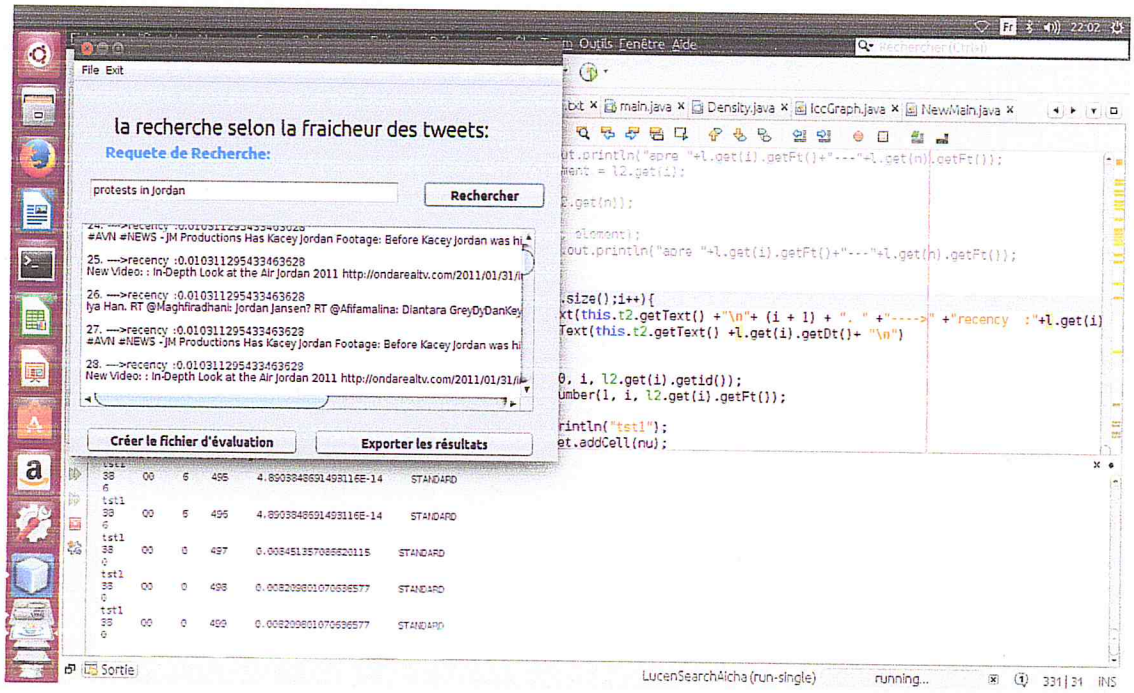


Figure 43 : *Le résultat de la recherche par la fraicheur du topic.*

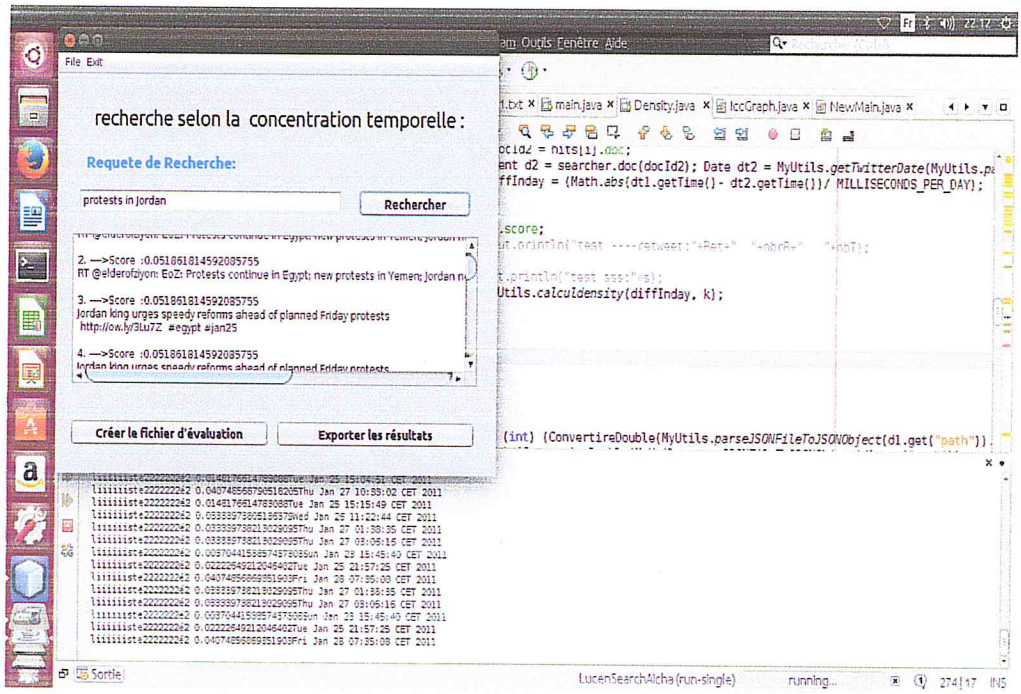


Figure 44: *Le résultat de la recherche via densité de noyau*

5. L'évaluation :

On va utiliser dans cette partie un outil de TREC pour évaluer nos résultats de recherches.

5.1. TREC-eval :

Trec_eval est un outil utilisé pour évaluer les classements, soit des documents, soit toute autre information triée par pertinence. L'évaluation est basée sur deux fichiers: le premier, connue sous le nom de "qrels" (informations de requête) énumère les jugements de pertinence pour chaque requête. Le deuxième contient le classement des documents retournés par votre système RI.

```
$ ./trec_eval [-q] [-m measure] qrel_file results_file
```

Trec_eval: c'est le nom du programme exécutable.

-q: donner une évaluation pour chaque requête / sujet

-Qrel_file: chemin du fichier avec la liste des documents pertinents pour chaque requête

-m: montre une mesure spécifique ("-m all_trec" montre toutes les mesures, "-m

official" est le paramètre par défaut qui ne montre que les principales mesures)
Result_file: chemin du fichier avec la liste des documents récupérés par notre application.

5.2. Les fichiers qrel_file :

Ce fichier contient une liste des documents considérés comme pertinents pour chaque requête. Ce jugement de pertinence est fait par des êtres humains qui sélectionnent manuellement des documents qui doivent être récupérés lorsqu'une requête particulière est exécutée. Ce fichier peut être considéré comme la «réponse correcte» et les documents récupérés par votre système IR devraient se rapprocher au maximum. Il a le format suivant:

Query-id 0 document-id pertinence

Le champ query-id est une séquence alphanumérique pour identifier la requête, id-document est une séquence alphanumérique pour identifier le document jugé, et la pertinence est un nombre pour indiquer le degré de pertinence entre le document et la requête (0 pour non pertinent et 1 Pour pertinence). Le deuxième champ "0" n'est pas actuellement utilisé, il suffit de le mettre dans le fichier. Les champs peuvent être séparés par un espace vide ou une tablette.

5.3 Les fichiers resultas_file :

Le fichier des résultats contient un classement des documents pour chaque requête automatiquement générée par notre application. C'est le fichier qui sera évalué par trec_eval en fonction de la «réponse correcte» fournie par le premier fichier. Ce fichier a le format suivant:

Query-id Q0 document-id classement score STANDARD

Le champ query-id est une séquence alphanumérique pour identifier la requête. Le deuxième champ, avec la valeur "Q0", est actuellement ignoré par trec_eval, il suffit de le mettre dans le fichier. Le champ document-id est une séquence alphanumérique pour identifier le document récupéré. Le rang est une valeur entière qui représente la position du document dans le classement, mais ce champ est également ignoré par trec_eval. Le score peut être une valeur entière ou flottante pour indiquer le degré de similitude entre

document et requête, de sorte que les documents les plus pertinents auront des scores plus élevés. Le dernier champ, avec la valeur "STANDARD", est utilisé uniquement pour identifier cette exécution.

5.4. Résultat de l'évaluation

Un résultat de l'évaluation de la recherche par concentration temporelle pour le Topic 38 :

eval-rec38 ×

```
num_q             all     1
num_ret           all     674
num_rel           all     15
num_rel_ret       all     13
map               all     0.2321
gm_ap             all     0.2321
R-prec            all     0.4000
bpref             all     0.1956
recip_rank        all     0.1667
ircl_prn.0.00     all     0.4000
ircl_prn.0.10     all     0.4000
ircl_prn.0.20     all     0.4000
ircl_prn.0.30     all     0.4000
ircl_prn.0.40     all     0.4000
ircl_prn.0.50     all     0.3571
ircl_prn.0.60     all     0.3571
ircl_prn.0.70     all     0.1618
ircl_prn.0.80     all     0.1579
ircl_prn.0.90     all     0.0000
ircl_prn.1.00     all     0.0000
P5                all     0.0000
P10               all     0.2000
P15               all     0.4000
P20               all     0.3000
P30               all     0.3333
P100              all     0.1200
P200              all     0.0600
P500              all     0.0260
P1000             all     0.0130
```

Figure 45 : *Résultat de l'évaluation de la recherchepar concentration temporelle pour le Topic 38*

6. Conclusion :

Ce chapitre a été consacré à la partie implémentation, ou on à commencer premièrement par la définition des outils utilisés, puis on a passé à la présentation de l'application, ainsi que les procédures de son utilisation, et on a terminé par la partie évaluation.

Conclusion générale :

Nous nous sommes intéressés dans ce travail à la RI ad hoc dans les microblogs. L'objectif est de retrouver les microblogs répondant à un besoin d'information spécifié par un utilisateur. Pour réaliser nos expérimentations, nous nous sommes basés sur le corpus fourni par la campagne d'évaluation international TREC (Texte Retrieval Conférence), la tâche Microblog 2011. Nos contributions se situent à trois niveaux sémantiques et temporels et leurs combinaisons avec l'aspect lexical.

Dans ce travail, nous avons parcouru l'état de l'art des systèmes de recherche d'informations dans les microblogs. Nous avons pu distinguer leurs points faibles et leurs points forts. Dans ce mémoire nous avons essayé d'améliorer la recherche d'informations dans les microblogs par la pris en charge des insuffisances des travaux voisins.

Nous avons commencé notre projet par le téléchargement du corpus des tweets puis nous avons nettoyé ce dernier pour éliminer tout source de bruit dans le processus d'appariement requête tweet, en suit nous avons identifié les facteurs empêchant leclassement des tweets pertinents dans le Top de la liste des résultats. Nous avons trouvé que les problèmes viennent de la concision des microblogs.

La concision engendre une correspondance limitée entre les termes des microblogs et les termes des requêtes, même s'ils sont sémantiquement semblables.

Pour améliorer leclassement des tweets pertinents dans la liste des résultats. Nous avons proposé trois méthodes pour le reclassement de cette dernière : sémantique, temporelle, leurs combinaisons avec le score lexical. L'évaluation de notre travail à donner des résultats satisfaisants.

Le travail présenté débouche sur plusieurs perspectives de recherche. Il serait intéressant de :

- Utiliser le Word embeddingpour enrichir les requêtes sémantiquement au lieu d'utiliser Wordnet, vu qu'il prend en considération l'aspect contexte.
- Utiliser le modèle BM25 pour le calculé la similarité entre les requêtes et les tweets. Vu les performances de ce dernier.
-Enrichir chaque tweet par le contenu de la page web dont leurs adresses URLs est englobée par ce dernier au lieu du titre de la page Web seulement.

Annexe 01 :

1. Collection des données :

Cette phase est la tâche la plus lente dans le processus de recherche des tweets,elle consiste à récupérer l'ensemble des tweets du Corpus TREC2011.

Afin de télécharger le Corpus, nous avons suivie les étapes suivantes :

- Accéder au lien de téléchargement des tweet : **https://apigee.com**
- Définir l'authentification pour twitter qui appelé « OAuth 1 »

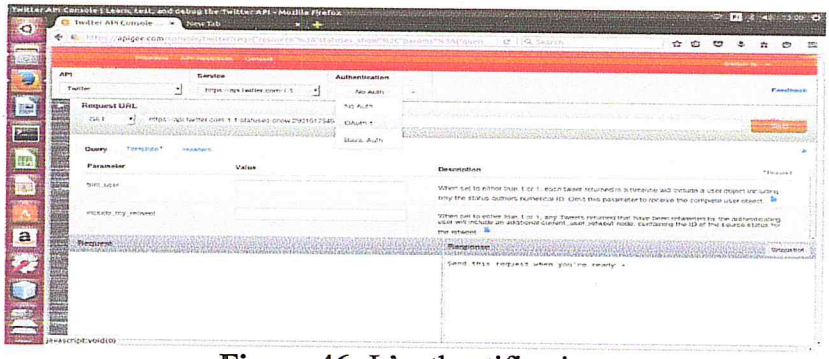


Figure 46 :L'authentification

L'api nous rédigons vers twitter pour activer l'authentification.

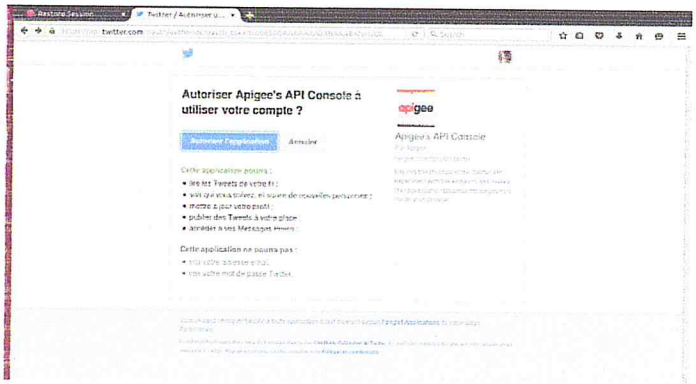


Figure 47 : Activation de l'authentification

- Nous changeons les services et la méthode et nous mettons l'id comme il est illustré dans la figure 20.

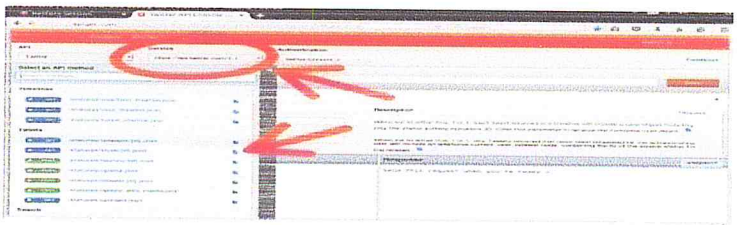


Figure 48 : Choix de services et la méthode

- Nous obtenons le tweet sous la forme suivant

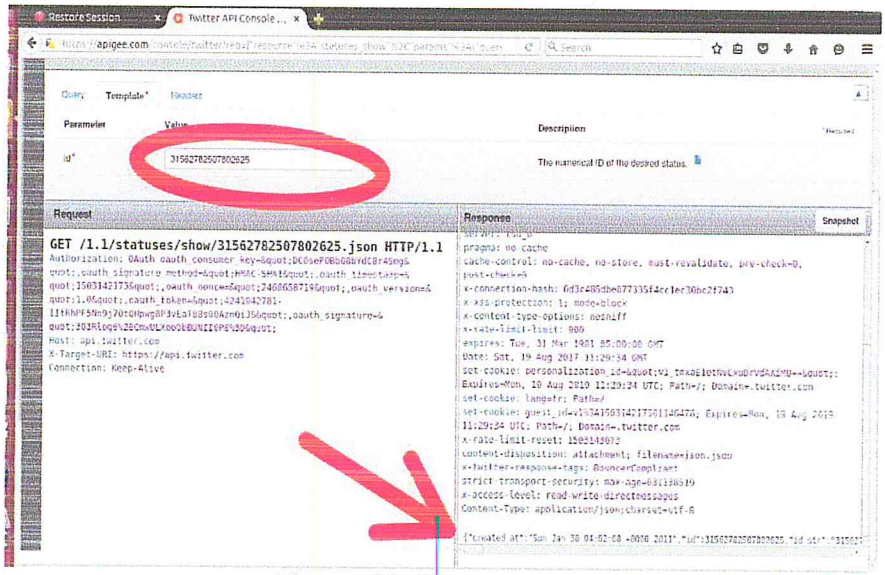


Figure 49 :*Format de tweet*

L'opération de prétraitement se fait en plusieurs étapes en appliquant les règles suivantes :

- Suppression des tweets écrits avec un langage autre que l'anglais.
- Suppression des tweets dont le code d'état (erreur) est égal à 403 ou 404 et 302.
- Suppression des Spams.
- Suppression des fichiers vides.
- Elimination des tweets répétés.

  Suppression des retweets.

Bibliographie

[Baeza, 1999] R. Baeza-Yates and R. A. Ribeiro-Neto. "Modern Information Retrieval". New York: ACM Press; Harlow England: Addison-Wesley, cop., 1999.

[Ben Jabeur,2013] L., Damak, F., Tamine, L., Cabanac, G., Pinel-Sauvagnat, K., et Boughanem,M. (2013). IRIT at TREC Microblog Track 2013. In E. M. Voorhees et (Eds.), *Text REtrieval Conference (TREC), Gaithersburg, USA,*. NationalInstitute of Standards and Technology (NIST).

[Bouramoul,2011] Recherche d'Information Contextuelle et Sémantique sur le Web. PhD Thesis in computer science, Constantine university – Algeria &Supelec , France.

[Boucetta, 2017] Z.Boucetta, **Bouramoul,2017, Bouznada,** "Vers l'utilisation des évidences syntaxiques, sémantiques et temporelles dans le PRF pour améliorer la recherche d'informations dans les tweets", ASD 2017 ,Conférence sur l'avancé des systèmes décisionnels.

[Charhad, 2005] M. Charhad. "Modèles de Documents Vidéo basés sur le Formalisme des Graphes Conceptuels pour l'Indexation et la Recherche par le Contenu Sémantique". pages 24-25, Novembre 2005.

[Choi,2012] Choi, J W. B.Croft, J.Y. Kim (2012). Quality Models for Microblog Retrieval.In CIKM'12.

[Choi, 2012] Choi, J., W. B.Croft, " Temporal Models for Microblogs". In CIKM'12, October 29–November 2, 2012, Maui, HI, USA. Copyright 2012 ACM 978-1-4503-1156-4/12/10

[Chy, 2015] Abu Nowshed Chy, Md Zia Ullah, and Masaki Aono, Combining Temporal and Content Aware Features for Microblog Retrieval,in IEEE , pages ,2015.

[Willis,2012] Rechard medlin jaime arguello incorporating Temporal Information in Microblog Retrieval , 2012 .

Bibliographie

[Dakka, 2012] L.Gravano, P.G. Ipeirotis (2012). Answering general time-sensitive queries. In TKDE 24(2) :220–235.

[Damak, 2013] Pinel-Sauvagnat, K., Cabanac, G., et Boughanem, M. (2013). Effectiveness of State-of-the-art Features for Microblog Search. In SAC'13 : ACM Symposium on Applied Computing. ACM.

[Damak, 2014] F.Damak.2014" 'Etude des facteurs de pertinence dans la recherche de microblogs". thèse de doctorat en informatique l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier) .

[Duan ,2010] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.Y. Shum. An empirical study on learning to rank of tweets. In Proceedings of COLING , pp.295–303, 2010.

[Efron, 2011] M. Efron, Golovchinsky,G.:"Estimation methode for ranking recent information". In : SIGIR.(2011) 495-504.

[Efron, 2014] J. J. Lin, J.He, P. Arjen (2014). Temporal feedback for tweet search with non-parametric density estimation. In SIGIR, 33-42.

[Jones,2007] Jones, R., F. Diaz .2007Temporal profiles of queries. In TOIS 25(3).

[Han,2012] Han,X. Li, M. Yang, H. Qi, S. Li, and T. Zhao. HIT at TREC 2012 microblog track. In Proceedings of TREC , 2012.

[Li,2003] X., W.Croft(2003). Time-based language models. In CIKM, 469–475.

[Maisonnasse, 2008] L. Maisonnasse. "Les supports de vocabulaires pour les systèmes de recherche d'information orientés précision : application aux graphes pour la recherche D'information médicale". Thèse de doctorat en informatique, Université Joseph Fourier- Grenoble I, France, 2008.

Bibliographie

[Middleton,2007] Middleton DA, Starr PJ, Gilbert DJ (2007) Modelling the impact of fisheries bycatch on Hector's dolphin: comment on Slooten 2007. Endang Species Res 3:331–334.

[Miyanishi, T] K. Seki, K.Uehara (2013a).Combining Recency and Topic-Dependent Temporal Variation for Microblog Search. In Proceedings of the 35th European Conference on Information Retrieval, 331– 343.

[Ounis,2006] Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., & Lioma, C. (2006). Terrier: A high performance and scalable information retrieval platform. In OSIR'06: Proceedings of 2nd international workshop on open source information retrieval, Seattle, USA (pp. 18–25).

[Ounis,2011] Ounis, I.,C. Macdonald, J. Lin, I.Soboroff (2011). Overview of the TREC- 2011 Microblog Track. In Proceeddings of the 20 th Text REtrieval Conference.

[Ottens, 07] K. Ottens. (2007) « Un système multi-agent adaptatif pour la construction d'ontologies à partir de textes ». page 14, Octobre.

[Robertson,1988] Robertson S. E., Sparck Jones K., « Document Retrieval Systems », Taylor Graham Publishing, London, UK, UK, chapter Relevance Weighting of Search Terms, p. 143-160, 1988.

[Salton, 1971] G. Salton.(1971) "A comparison between manual and automatic indexing methods. Journal of American Documentation", 20(1) :61–71,

[Smart, 2006] Smart, J. F (2006). Web Mining Lab in UCLA.

[Tambellini, 2007] C. Tambellini. "Un système de recherche d'information adapté aux données incertaines: adaptation du modèle de langue". Thèse de doctorat en informatique,Université de Nice-Sophia Antipolis-UFR sciences, 2007.

Bibliographie

[Willis,2012]. Willis, C., R. Medlin,J. Arguello (2012) Incorporating Temporal Informationin Microblog Retrieval . In Proceedings of the Twenty-First Text REtrieval Conference.

[WETICE,2012] WETICE'12 Proceedings of the 2012 IEEE 21st International Workshop on Enabling Technologie:Inrastructeur for Collaborative Enterprises Page 468-473 June 25-27,2012 IEEE Computer Society Washington,DC,USA

WEB:

[1] http://www.blogdumoderateur.com/chiffres-reseaux-sociaux/

[2]

[3] http://lucene.apache.org/

[4] http://terrier.org/

[5] http://www.lemurproject.org/indri

[6] http://www.numerama.com/startup/twitter

[7] http://oseox.fr/twitter/histoire-twitter.html

[8] https://twitter.com/?lang=fr

[9] http://www.alesiacom.com/blog/maitriser-le-vocabulaire-de-twitter

[10] https://inex.mmci.uni-saarland.de/

[11] http://blog.tuquoque.com/post/2008/03/23/Comparaison-de-moteurs-de-recherche-open-source

[12] http://trec.nist.gov/data/tweets/

[13] https://netbeans.org

[14] https://lucene.apache.org

[15] https://www.mkyong.com/java/json-simple-example-read-and-write json

[16] http://www.jfree.org/jfreechart/download.html

[17] http://twitter4j.org

[18] https://stanfordnlp.github.io/CoreNLP/download.html

[19] http://www.rafaelglater.com