République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**
Filière : **Informatique**
Spécialité : **Systèmes Informatiques**

Présenté par
BENFEDDA Kahina
BENALI MOHAMED Hayet

Thème

Contextualisation des tweets

Mémoire soutenu publiquement le 04/07/2017 devant le jury composé de :

Président : M^{r} N.AMIROUCHE
Promotrice : M^{me} F.AMIROUCHE
Examinatrice : M^{me} S. ILTACHE
Examinatrice : M^{me} L. BELKACEMI

Promotion 2016/2017

Remerciements

***D**e prime abord, nous tenons à remercier le Bon Dieu tout puissant de*
***N**ous avoir donné patience, courage et volonté pour réussir*
***N**otre mémoire.*

.

***N**ous tenons à remercier vivement notre promotrice F.Amirouche qui*
***N**ous a aidées et orientées pour la réalisation*
***D**e ce travail.*

***N**os remerciements vont également aux membres de jury qui ont accepté*
***D**'évaluer notre travail.*

***N**ous souhaitons également exprimer notre profonde gratitude à tous ceux*
***Q**ui du près ou de loin ont participé à la réalisation de ce*
***M**odeste travail.*

Kahina & Hayet

Dédicaces

***J**e dédie ce modeste travail avant tout à mes parents qui ont toujours su **ê**tre présents à mes côtés et qui m'ont toujours encouragée tout au long de mes études.*

***A** la mémoire de mon oncle Mohamed*

***A** Ma chère grande mère Sadia*

***A** Mes deux frères Nabil et Fares et à mes deux sœurs : Sara et Souhila*

***A** toute ma famille qui m'aime de loin ou de prés*

***A** tous ceux qui me sont chèrs*

***A** tous ceux qui ont souhaité ma réussite*

***A** tous ceux que j'aime*

Kahina

Dédicaces

***J**e dédie ce modeste travail avant tout à mes très chers parents pour l'éducation qu'ils m'ont donnée, leurs sacrifices et leurs encouragements, que Dieu les garde et les protège.*

***A** mon cher frère Mohand Arabe, les mots ne suffisent pas pour exprimer l'attachement et l'amour que je porte pour toi.*

***A** mes chères sœurs Lynda et Zoubida, Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite. Spécialement à ma deuxième moitié, ma très chère sœur jumelle Lynda.*

***A** mon fiancé Hichem, Pour son soutien moral, sa gentillesse sans égal, son profond attachement qui m'ont permis de réussir dans ce travail.*

***A** mon binôme Kahina.*

Et à tous ceux qui me connaissent de près ou de loin.

Hayet

Table des matières

II La contextualisation des tweets

III Approche de contextualisation se basant sur la conversation sociale

IV Implémentation et expérimentations

Liste des figures

Liste des tableaux

Introduction générale

Depuis toujours, l'information joue un rôle important dans le quotidien des individus. Les individus échangent de l'information et décident en fonction des informations qu'ils acquièrent. Le développement de l'informatique dans tous les domaines mais aussi le développement du web et en particulier l'émergence des réseaux sociaux sur le web (dont Facebook et Twitter), véritables plateformes de production et d'échange d'information, mais aussi véritables sources d'informations, ont conduit à la production d'un volume inestimable d'information. En effet la quantité d'information disponible, le web, se mesure en milliards de pages.

Il est cependant, de plus en plus difficile de localiser précisément ce que l'on cherche dans cette masse importante d'information. La recherche d'information (RI) est le domaine de l'informatique qui s'intéresse à répondre a ce problème. L'objectif principal de la recherche d'information est de fournir des modèles, des techniques et des outils, les systèmes de recherche d'information (SRI), pour stocker et organiser des informations et retrouver et retourner celles qui répondent à un besoin en information de l'utilisateur exprimé sous forme d'une requête.

Nous nous intéressons dans le cadre de notre travail à la recherche d'information dans Twitter, l'un des réseaux sociaux numériques les plus populaires du moment.

Twitter compte plus de 300 millions d'utilisateurs actifs par mois avec 500 millions de tweets (Messages publiés sur Twitter) envoyés par jour, et est disponible dans plus de 40 langues. Les tweets sont des messages très courts limités à 140 caractères et peuvent contenir outre le texte du tweet (utilisant généralement un langage non conventionnel SMS, émoticônes, abréviations), d'autres caractéristiques telles que : les hashtags, Ou les URL, ou encore des images….

Les SRI ont été mis en œuvre initialement dans le cadre de recherches bibliographiques dans des textes conventionnels. Un SRI classique utilisé dans le cadre de recherche d'information dans les tweets retourne comme résultat des bribes de texte (issues du tweet) incompréhensibles pour l'utilisateur.

Pour pallier à ce problème, une solution consiste à enrichir les tweets retournés par des informations liées permettant ainsi de les étendre et par conséquent de le rendre plus compréhensibles. Les informations liées au tweet constituent son contexte. Cette approche est dite « contextualisation des tweets ».

Notre travail dans le cadre de ce mémoire consiste à étudier et à mettre en œuvre une approche de contextualisation des tweets.

Notre mémoire est organisé en quatre chapitres comme suit :

Chapitre 1 : Ce chapitre introduit la recherche d'informations, à travers ses concepts fondamentaux, ses techniques et ses outils.

Chapitre 2 : Ce chapitre aborde la contextualisation des messages courts et en particulier la contextualisation des microblogs, présente quelques approches existantes.

Chapitre 3 : Ce chapitre décrit notre approche de contextualisation des tweets.

Chapitre 4 : Ce chapitre représente les aspects d'implémentation et de mise en œuvre de notre approche ainsi que son évaluation

Une conclusion résume notre travail, et introduit nos perspectives et propositions futures.

Chapitre I

Généralités sur la recherche D'information

I.1 Introduction

La recherche d'information (RI) n'est pas un domaine récent, elle est apparue dans les années 40, et définie classiquement comme une activité de recherche documentaire dont la finalité est de délivrer un ensemble de documents à un utilisateur en fonction de son besoin en information exprimé sous forme de requête.

Notre but à travers ce chapitre est de définir la RI et des SRI classiques et d'en présenter les concepts de base, puis d'introduire les spécificités de la RI dans Twitter.

I.2 La recherche d'information (RI)

I.2.1 Définition

La RI est l'ensemble de techniques permettant de retrouver dans une collection de documents ceux qui sont susceptibles de répondre au besoin informationnel de l'utilisateur. Plus précisément, la RI est la tâche qui prend en charge la gestion, l'acquisition, l'organisation, le stockage et la recherche d'information dans une collection de documents préalablement stockée sur ordinateur. La RI est mise en œuvre à travers un SRI.

I.2.2 Définition d'un SRI

Un SRI est un outil permettant de sélectionner à partir d'une *collection documentaire*, les *documents pertinents* en réponse à une *requête* utilisateur.

La définition d'un SRI fait ressortir les concepts de base suivants :

Requête: Il s'agit d'une représentation du besoin d'information de l'utilisateur. La requête joue le rôle d'interface entre le SRI et l'utilisateur. Elle est exprimée dans un langage des requêtes qui

peut être le langage naturel, un langage booléen (avec ou sans opérateurs : and, or, not) ou encore comme une liste de mots clés.

Document: Le document constitue l'unité de base manipulable et accessible par le SRI. Il peut représenter un texte, une page web, une image, une bande vidéo etc.

Collection de documents: C'est l'ensemble des informations (ie. documents) exploitables et accessibles par le SRI.

Pertinence: C'est le degré de similarité d'un document avec la requête utilisateur.

On distingue deux types de pertinence :

-Pertinence système : Dans ce cas, le SRI utilise une fonction de correspondance pour juger le score de pertinence des documents par rapport à la requête.

-Pertinence utilisateur : C'est l'ensemble des jugements émis par l'utilisateur sur un ensemble de documents retourné en fonction du degré de satisfaction de ses besoins en information. Ce jugement peut être différent d'un utilisateur à un autre.

I.2.3 Processus de recherche

Le processus de recherche se base sur trois étapes principales représentées schématiquement par le processus classique en U de la recherche d'information (voir Figure I.1):

1. **L'indexation**

   Dans cette étape, le système extrait les termes les plus représentatifs des documents (et de la requête) dans l'objectif de construire une représentation interne qui couvre au mieux leurs contenus respectifs.

2. **L'appariement requête-document**

   À ce niveau, le système compare la représentation interne de chaque document de la collection avec la représentation interne de la requête sur la base d'un score de pertinence qui estime le degré de pertinence du document vis- à vis de la requête.

3. **La reformulation de la requête**

   La reformulation du besoin en information est l'étape qui permet de redéfinir le besoin de l'utilisateur au fur et à mesure de la session de recherche. L'objectif est d'affiner progressivement l'expression du besoin en information afin d'arriver à des résultats de plus en plus pertinents.

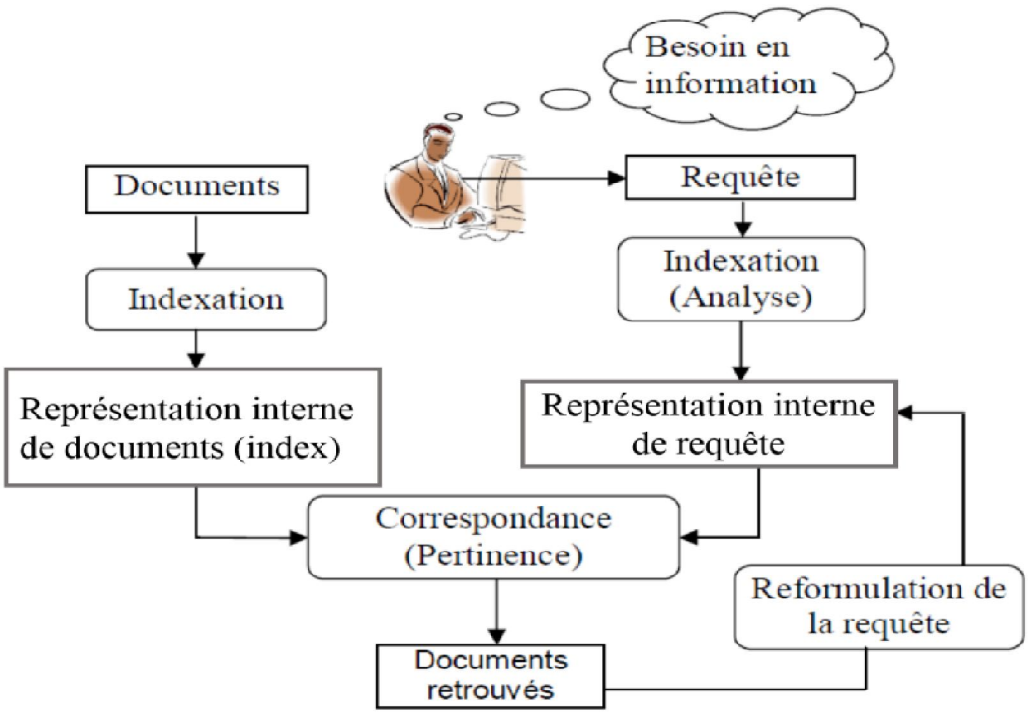

Figure I.1-Schéma de processus de RI en U **[1]**

Dans ce qui suit, nous détaillons chacune de ces étapes.

I.2.3.1 L'indexation

L'indexation c'est l'opération qui garantit la meilleure représentation des documents et des requêtes pour la recherche. Cette représentation consiste en une liste de mots clés appelés identificateurs des documents ou de la requête.

La finalité de l'indexation est de construire un index qui contient un ensemble d'identificateurs représentant le contenu de document ou de la requête. L'index est une structure de données exploitable par le SRI lors de la recherche.

On distingue trois types d'indexation :

- **L'indexation manuelle :** Ce type d'indexation est réalisé par un indexeur humain qui choisit les mots les plus significatifs pour représenter le document.

  Cette approche est couteuse et ne peut être envisagée pour de grandes collections de documents. De plus, elle est subjective du fait que pour un même document, des termes différents peuvent être affectés par des documentalistes différents en fonction de leurs connaissances propres.
- **L'indexation semi-automatique (mixte) :** Les termes de document sont d'abord extraits automatiquement, puis le documentaliste prend en charge le choix des termes significatifs et l'établissement de relation entre eux.
- **L'indexation automatique :** Cette approche est la plus fréquemment utilisée, c'est une indexation entièrement réalisée par un programme informatique. Nous la détaillons ci-après.

L'indexation automatique base sur les étapes suivantes :

1. **Analyse lexicale:** Elle consiste à identifier les mots clés du texte (encore appelés termes d'index) en procédant par reconnaissance des espaces de séparation des mots, des caractères spéciaux, des chiffres, de la ponctuation… .
2. **L'élimination des mots vides :** La liste des mots retournés dans l'étape précédente peut contenir des mots non significatifs appelés mots vides.

   Un mot vide est un mot non informatif pour un document ou une requête. Les prépositions (à, pour …), les déterminants (de la,..), les opérateurs arithmétiques (opérateurs d'appartenance, opérateurs d'inclusion,..) sont des exemples des mots vides.

   Afin d'éliminer les mots vides, on utilise une liste prédéfinie de mots vides (appelée anti dictionnaire ou stop- list).
3. **La pondération des termes d'index:** L'objectif de la pondération des termes est d'apporter une solution au principal problème des algorithmes de prédiction par reconnaissance partielle, à savoir la difficulté de concevoir un modèle statistique capable de représenter n'importe quel type de données que l'on voudrait pouvoir compresser. La majorité des approches de pondération calcule l'importance d'un terme en utilisant des statistiques. On distingue :

– **Approches basées sur la fréquence locale (Tf)**

Un terme qui apparait souvent dans le document représente un terme important. Le poids du mot est alors égale au nombre d'occurrence du mot t dans le document d qui est définit comme suit :

$$TF = \frac{frq(t,d)}{maxfreq(t,d)} \qquad (1)$$

Où

d: document, t:terme

– **Approche basée sur la fréquence globale (IDF)**

Dans cette approche, l'importance est donnée au terme qui apparait moins fréquemment dans toute la collection. Le poids d'un terme est ainsi inversement proportionnel à sa fréquence documentaire.

Le facteur de pondération globale IDF (Inverse of Document Frequency) est calculé par

$$IDF = \log\left(\frac{N}{ni}\right) \qquad (2)$$

Ou $$IDF = \log\left(\frac{N-ni}{N}\right) \qquad (3)$$

Où

ni: Le nombre de documents contenant le terme ti

N : Le nombre total de document dans la collection.

– **Approche basée sur TF*IDF**

Souvent, l'approche globale et l'approche locale sont combinées en un seul score : TF*IDFqui donne une bonne approximation de l'importance d'un terme dans le document.

À la fin de processus d'indexation, chacun des documents de la collection est représenté par un ensemble de mots clés pondérés stocké dans une structure de données, qui permet un accès rapide document. Parmi ces structures, les fichiers inverses[1] sont les plus utilisés.

[1] **Le fichier inverse :** est un fichier composé d'ensemble des mots associés à un document et une liste de toutes leurs positions (posting).

I.2.3.2 L'appariement requête-document

Cette étape détermine le degré de similarité d'un document pour une requête, et d'ordonner ainsi les documents retournés selon leurs degré de pertinence pour la requête. Elle classe éventuellement les documents par ordre de pertinence pour les requêtes.
La fonction d'appariement est notée RSV (d,q) est définie par les modèle de recherche[2].

Où : d : un document de la collection et q : est une requête.

I.2.3.3 La reformulation de la requête

La requête est une représentation possible d'un besoin en information de l'utilisateur, mais parfois, l'utilisateur n'arrive pas à bien exprimer sa requête. De ce fait, le résultat de la recherche est approximatif et ne le satisfait souvent pas.
Afin d'améliorer les résultats retournés, on améliore la reformulation de la requête initiale en ajoutant de nouveaux termes ou en supprimant des termes inutiles pour rapprocher la pertinence système de la pertinence utilisateur.
Il existe plusieurs méthodes de reformulation de requêtes telles que :

Combinaison des présentations des requêtes

C'est une méthode efficace dans les SRI. Son principe est d'exploiter des représentations multiples de requêtes ou des algorithmes de recherche différents ou encore en utilisant différentes techniques de réinjection.
Une combinaison des représentations de requêtes peut augmenter le rappel d'une requête.

Réinjection de la pertinence

Elle consiste à ajouter des termes issus des premiers documents supposés pertinents pour améliorer la requête de l'utilisateur.

Parmi les approches les plus utilisées, nous citons celle de Rocchio qui propose le modèle de reformulation de requête suivant :

$$Q_N = \alpha . Q_0 + \beta . \frac{1}{|R|} \sum_{r \in R} r - \frac{1}{|R'|} \sum_{r' \in R'} r' \qquad (4)$$

Où :

QN est le vecteur de la nouvelle requête (requête reformulée),

Q0 est le vecteur de la requête initiale,

[2] Voir la section des modèle de RI

R est l'ensemble des vecteurs r des documents jugés pertinents par l'utilisateur,

R' est l'ensemble des vecteurs r' des documents jugés non pertinents par l'utilisateur,

α, β Sont des paramètres de la reformulation,

Le résultat de cette formule est une nouvelle requête dont le vecteur se rapproche des vecteurs des documents jugés pertinents et s'éloigne des documents jugés non pertinents.

Expansion automatique des requêtes

L'expansion de la requête consiste à rajouter à la requête initiale des termes extraits des documents pertinents retournés, ces termes sont ensuite ajoutés à la requête originale comme des termes d'expansion. **[w5]**

I.2.4 Les modèles de RI

Les modèles de RI fournissent une formalisation du processus de recherche d'information et un cadre théorique pour la modélisation de la pertinence. Cette dernière est le rôle le plus important qui est assuré par tous les modèles de RI.

Un modèle de RI est défini par un quadruplet (D, Q, F, RSV (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.
- RSV (d, q) est la fonction de pertinence du document d par rapport à la requête q.

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.

On distingue trois grandes catégories de modèles : [**w8]**

I.2.4.1 Le modèle booléen

Le modèle booléen est le premier modèle utilisé dans la RI. Il est simple, fondé sur la théorie des ensembles et l'algèbre de bool. Dans ce modèle, le document est une conjonction de mots clés.

La requête est une expression logique qui contient des termes avec des opérateurs booléens : AND| OR| NOT.

La pertinence d'un document pour une requête est calculée comme suit :

$$RSV(d,q) = \begin{cases} 1 \text{ si } q \in d \\ 0, \text{ sinon} \end{cases} \quad (5)$$

$$RSV(d, qi \wedge qj) = RSV(d, qi) \wedge RSV(d, qj) \quad (6)$$

$$RSV(d, qi \vee qj) = RSV(d, qi) \vee RSV(d, qj) \quad (7)$$

$$RSV(d, \neg qi) = 1 - RSV(d, qi) \quad (8)$$

Le modèle booléen présente d'importants avantages tels que : le formalisme propre, facile à mettre en œuvre. Il présente cependant quelques inconvénients dont :

- La correspondance exacte peut récupérer peu ou trop de documents,
- Difficile de traduire une requête en une expression booléenne,

Pour y remédier, des extensions sont proposées : Le modèle booléen étendu et le modèle booléen basé sur les ensembles flous.

I.2.4.2 Le Modèle vectoriel

Il représente les requêtes et les documents sous forme de vecteurs de poids dans l'espace vectoriel des termes d'index.

La pertinence d'un document di pour une requête q est une mesure de similarité entre les vecteurs correspondant, calculée comme suit :

$$cos(d_j, q) = = \frac{\overrightarrow{dj} * \vec{q}}{||\overrightarrow{dJ}|| * ||\overrightarrow{q}||} = \frac{\sum_{i=1}^{n} wi,j * wi,q}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} * \sqrt{\sum_{i=1}^{n} w_{i,q}^2}} \quad (9)$$

Il existe plusieurs d'autres mesures plus courantes, parmi eux nous citons :

- Le produit scalaire calculé comme suit :

$$RSV(q_i, d_j) = \sum_{K=1}^{M} wki.wkj \quad (10)$$

- La mesure de Jaccard :

$$RSV(d_i, q_j) = \frac{\sum_{K=1}^{M} wki.wkj}{\sum_{K=1}^{M} w_{ki}^2 + \sum_{K=1}^{M} w_{kj}^2 - \sum_{K=1}^{M} wk,i*wk,j} \qquad (11)$$

Le modèle vectoriel présente plusieurs avantages par rapport au modèle booléen standard cité précédemment comme :

- Les mesures de similarité utilisées permettant d'ajouter à la notion de pertinence un degré d'approximation. Un document peut ainsi être considéré comme pertinent même s'il ne contient pas tous les termes de la requête.
- Le classement ordonne par ordre décroissant de pertinence.

Ce modèle présente aussi des inconvénients, parmi eux nous citons:

- Sensibilité sémantique: Sensibilité sémantique; Les documents ayant un contexte similaire, mais un vocabulaire différent ne sera pas associé, ce qui entraînera une « fausse correspondance négative ».
- L'ordre dans lequel les termes apparaissent dans le document n'est pas conservé dans la représentation vectorielle.

I.2.4.3 Le Modèle probabiliste

Dans ce modèle, le document et requête sont représentés par des vecteurs de poids.

Ce modèle calcule la probabilité qu'un document d soit pertinent (respectivement non pertinent) pour une requête q. Supposant que cette probabilité de pertinence dépend de la requête et des représentations de documents.

Pour cela, on distingue deux classes de documents pour une requête q :

R l'ensemble des documents pertinents et $\overline{R}$l'ensemble des documents non pertinents.

La fonction de classement de ce modèle est donnée par la formule suivante:

$$RSV(d, q) = \frac{P(R/d)}{P(\bar{R}/d)} \qquad (12)$$

Où

P(R/D) est la probabilité que le document D appartienne à l'ensemble des documents pertinents.

P ($\overline{R}$/D) est la probabilité que le document D appartienne à l'ensemble des documents non-pertinent.

La figure suivante illustre les différents modèles définis précédemment.

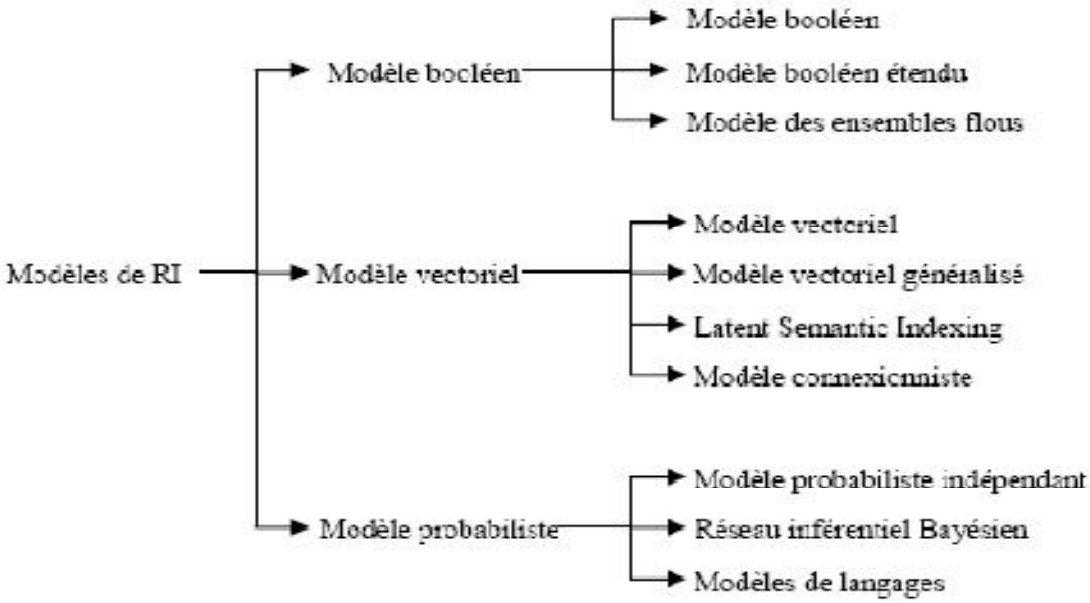

Figure I.2 -Modèle de RI **[w8]**

I.2.5 Evaluation des SRI

L'évaluation des systèmes peut être abordée selon le temps et l'espace (Plus le temps de réponse est court et plus l'espace occupé par le système est faible, meilleur est considéré le système), et selon la capacité du système à sélectionner un maximum de documents pertinents et un minimum de documents non pertinents.

I.2.5.1 Mesures d'évaluation

La capacité intrinsèque d'un SRI peut être mesurée à travers diverses métriques, dont: le rappel, la précision, le bruit et le silence. [**W5**]

- **La précision:** C'est le pourcentage des documents pertinents retrouvés parmi tous les documents retrouvés par le système:

$$Précision = \frac{|\text{Documents pertinents retournés par le SR}|}{|\text{Documents totals retrouvés par le système}|} \in \overline{0,1} \qquad (13)$$

La valeur précision à 1 signifie que le système n'a retrouvé que des documents pertinents.

- **Le rappel :** c'est le taux de documents pertinents retournés par le SRI par rapport à tous les documents pertinents présents dans la base documentaire:

$$\textbf{Rappel} = \frac{|\text{Documents pertinents retourné par le SRI}|}{|\text{Documents pertinents dans la base}|} \in \overline{0,1} \qquad (14)$$

Un rappel égal à 1 signifie que tous les documents pertinents ont été retrouvés.

La précision et le rappel ne sont pas indépendants. Il y a une forte relation entre eux: l'un augmente, l'autre diminue.

La mesure précision/rappel est représentée par la courbe suivante :

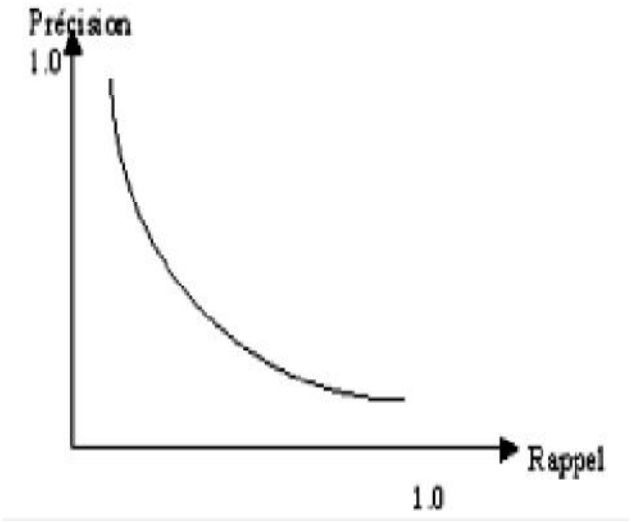

Figure I.3-Courbe de Rappel-Précision

Un SRI est performant s'il fournit des taux de précision et de rappel égaux.

- **Bruit :** ensemble de documents non pertinents retournés par les SRI en réponse à une requête donnée.

$$\text{Bruit} = 1 - \text{Rappel} \quad (15)$$

- **Silence :** ensemble de documents pertinents non sélectionnés par le système lors d'une recherche d'information.

$$\text{Silence} = 1 - \text{précision} \quad (16)$$

Dans même duales du rappel et de la précision sont définis comme suit :

– Les quatre facteurs peuvent être représentés avec les relations suivantes :

Précision+ bruit=1

Rappel+ silence = 1

Un bon SRI est celui qui est capable de renvoyer les bons documents en rejetant les documents non pertinents (en faisant le moindre bruit possible), et de restituer le maximum de documents pertinent (silence

I.2.5.2 Campagnes d'évaluation

Pour évaluer l'efficacité d'un SRI, il faut le soumettre à un jeu d'essai qui consiste à prévoir un ensemble de documents-test et interroger un SRI grâce à un ensemble de requêtes représentant les thèmes de ces documents. En mesurant la réponse, on obtiendra une mesure de qualité sur les performances du système de recherche d'information.
Une des collections les plus utilisées en RI est la collection TREC.

TREC

TREC (TextRetrievalConference) est une campagne annuelle d'évaluation des travaux de RI. Elle est financée par la DARPA (Agence des Projets de Défense Avancée) et le NIST (L'institut national des normes et de la technologie), et a débuté en 1992 par le NIST à Washington. Son but est de soutenir et d'encourager la recherche au sein de la communauté de recherche d'information en fournissant l'infrastructure nécessaire à l'évaluation à grande échelle des méthodes de recherche de texte : collections de test, tâches à investir, critères et procédures d'évaluation, etc.

Les collections TREC sont composées d'un ensemble important de documents, d'un ensemble de topics correspondant aux requêtes et un ensemble de jugements de pertinence associé aux requêtes. Ces jugements consistent à comparer les résultats réels des systèmes aux résultats théoriques établis par les juges.

Plusieurs tâches sont définies dans TREC. On peut citer plutôt les taches qui intéressent notre travail comme la tache TREC MICROBLOG qui définit comme une recherche ad hoc en temps réel sur Twitter c'est-à-dire, l'utilisateur cherche à satisfaire son besoin en consultant les tweets récents.

INEX

INEX (INitiative for the Evaluation of Xmlretrieval) lancé en 2002, avec plus de 90 organisations participantes à travers le monde. INEX encourage la recherche de récupération d'informations XML en fournissant une infrastructure pour évaluer l'efficacité des systèmes de récupération de l'information XML.

Elle a été parmi les campagnes qui ont proposé une évaluation formelle pour la recherche dans les plateformes de microblogging, notamment Twitter en 2012.Pour cela, elle a lancé la tache *Tweet Contextualisation*. L'objectif de cette tâche était la compréhension d'un tweet en lui fournissant un bref résumé explicatif (500 mots).

Cette tache peut engendrer deux sous taches:

- La première consiste à rechercher les articles Wikipédia les plus pertinents en utilisant un SRI,
- La deuxième consiste à extraire les passages les plus représentatifs d'un tweet donné, en les résumant par la suite en utilisant un système de résumé automatique (SRA).

I.3 La RI dans Twitter

I.3.1 Présentation de Twitter

Twitter désigne un service de microblogging, c'est le nom commercial d'un service d'échange de messages courts. Il a été créé le *21 mars 2006* par *Jack Dorsey*, *Evan Williams*, *Biz Stone* et *Noah Glass*, et lancé en juillet de la même année à new York.

Twitter est devenu populaire jusqu'à réunir plus de *313 millions* d'utilisateurs actifs par mois au *5 mars 2017*avec *500 millions* de tweets envoyés par jour et est disponible en plus de *40* langues.

La figure suivante représente le logo de Twitter qui est un oiseau qui gazouille.

Figure I.4-Logo de Twitter.

I.3.2 Structure d'un tweet

Un tweet est un énoncé produit sur la plateforme des microblogging Twitter, il prend la forme d'un message court limité à 140 caractères pour échanger sur internet un maximum d'informations en minimum de caractères.

Après avoir créé un compte Twitter, l'utilisateur de Twitter « microbloggeur » le décrit par l'ajout des informations personnelles telles que sa biographie, sa photo, ses centres d'intérêt et sa localisation, etc.

La figure suivante montre l'exemple d'un profil au sein de Twitter.

Figure I.5-Exemple d'interface principale de l'utilisateur Twitter

Le microbloggeur peut envoyer « tweeter » ou « gassouiller » des brefs et courts messages appelés « tweets » ou « microblogs ».

Le microbloggeur qui suit nos tweets est appelé abonné « Follower » et les microbloggeurs dont nous suivons les tweets dits « Followees ».

Chaque tweet apparait sur la page de profil de son auteur et est transmis à ses abonnés qui le reçoivent dans leurs « timeline ». La timeline est la page principale sur laquelle apparaissent les tweets des comptes auxquels le microbloggeur s'est abonné.

Si un tweet est intéressant, le microbloggeur peut le partager « retweeter » et il peut envoyer un message privé « direct message DM » à un abonné qui est apparait dans son timeline.

Sur Twitter, on peut suivre des microbloggeurs qui nous intéressent sans avoir besoin de leur accord.

Les microbloggeurs utilisent un lexique particulier en incluant différents symboles dans un tweet en plus de son contenu (comme les hashtags, mentions, retweeter, message directe incluant possiblement des hyperliens), parmi ces symboles nous citons :

@ Suivi par un nom d'utilisateur appelé « Mention », permet d'indiquer qu'on mentionne ou s'adresse à une personne ou entité particulière. (Exemple : @informatique).

Suivi par un mot appelé «Hashtag » qui indique le regroupement des tweets intéressants pour que les utilisateurs puissent trouver un contenu spécifique.

On peut trouver aussi des tweets contenant des URL envoyés à des pages web.

Dans ce qui suit, nous présenterons un exemple qui englobe la pluparts des formats de tweet échangés.

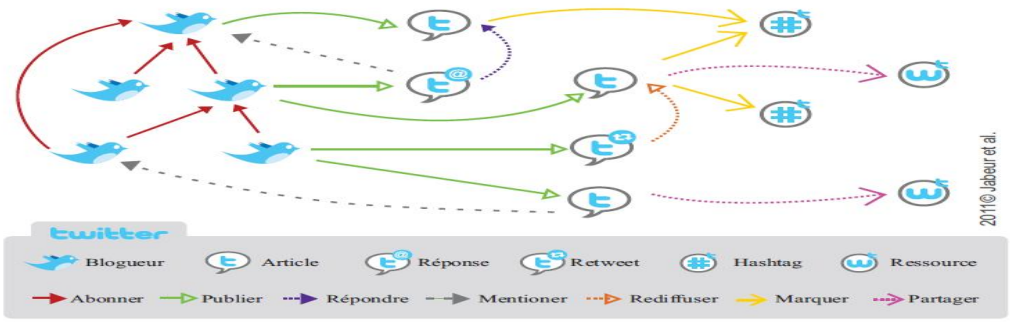

Figure I.6-Exemple d'utilisation de Twitter **[w7]**

I.3.3 RI dans Twitter

Les différentes caractéristiques des tweets rendent la recherche d'information dans les plateformes de microblogging différente de celle du web. Dans ce qui suit, nous présenterons les approches d'accès à l'information dans les microblogs.

I.3.3.1 Accès à l'information dans les microblogs

Les chercheurs qui s'intéressent à la RI dans les microblogs ont proposés plusieurs approches qui sont classées selon le type d'information recherché. On distingue :

- **La recherche temps réel**

  La recherche d'information permet ici de retourner des informations essentielles sur un événement récent en temps réel. La date de publication du document est le facteur le plus important dans la recherche des microblogs en temps réel.

- **La recherche d'opinion**

  L'objectif de cette recherche consiste à retrouver les documents exprimant des opinions à propos de sujet d'une requête donnée.

  D'après Bernard Jansen l'un des chercheurs qui sont intéressés à la recherche d'opinion et de réactions immédiates de microbloggeurs sur des produits et des marques. **[6]**

- **La détection de tendances**

  Elle a comme objectif l'identification des sujets les plus parlants (ou tendance) dans le réseau social notamment Twitter. Cette détection se base sur la recherche d'expressions les plus populaires dans les microblogs.

- **La recherche de microbloggeurs**

  Plusieurs travaux se sont concentrés sur l'identification des microbloggeurs les plus populaires. TwitterRank est une approche qui mesure la popularité des microbloggeurs. Cette approche prend en considération les similarités des sujets discutés entre les microbloggeurs, ainsi que la structure des liens d'abonnements.

- **La recherche thématique**

  L'objectif de la recherche thématique de microblogs est de créer des filtres thématiques sur les flux d'information. Ceci est réalisé en identifiant les sujets les plus parlent dans les microblogs. La recherche thématique des microblogs nous permettra, de classer les utilisateurs en fonction de leurs centres d'intérêts. Dans **[w5]** ont utilisé une implémentation de Latent Dirichlet Allocation(LDA) afin d'extraire les mots clés et de les utiliser pour caractériser les utilisateurs et les microblogs.

I.3.3.2 Facteurs de pertinence des microblogs

Nous considérons cinq groupes de facteurs de pertinence : celui lié au contenu des microblogs, celui lié à leur hyper textualité, celui qui se base sur les hashtags, celui lié aux auteurs des microblogs et enfin un groupe de facteurs relatifs à la qualité des microblogs.

- **Facteurs de pertinence basés sur le contenu des tweets**

  Nous avons considéré deux facteurs de pertinence relatifs à certaines spécificités de contenu des microblogs :

  1. La popularité du Tweet : ce facteur estime la popularité d'un Tweet dans un corpus. Un Tweet est supposé populaire si seulement si on trouve plusieurs autre tweets ayant un même contenu.
  2. Longueur d'un Tweet: plus une phrase est longue, plus elle contient plus d'information, on calcule se facteur de pertinence en comptant le nombre de termes dans un Tweet.

- **Facteurs de pertinence basés sur l'hypertextualité**

  On considère deux facteurs de pertinences qui peuvent indiquer la qualité de l'information publié dans les tweets :

  1. Présence d'URL dans le Tweet: Ce facteur suppose que la présence d'une URL implique que le Tweet a un caractère informatif renforcé. **[2]**
  2. Fréquence de l'URL dans le corpus : Ce facteur permet de calculer la popularité des URLs publiée dans un Tweet présent dans le corpus.

–**Facteurs de pertinence basés sur la popularité des auteurs**

Afin de tenir compte de la popularité des auteurs, nous avons considéré deux facteurs spécifiques :

1. Nombre de tweets de l'auteur : le but de ce facteur de pertinence est d'augmenter la publication des tweets par des auteurs actifs par rapport aux tweets publiés par des auteurs moins actifs. **[2]**
2. Nombre de citation de l'auteur : ce facteur évalue la popularité de l'auteur en fonction du nombre de ces publications, plus un auteur est mentionné, plus il est populaire.

–**Facteurs de pertinence basés sur les hashtags**

1. Présence d'un hashtag: La présence des hashtags peut être considérée comme un facteur de pertinence. **[w7]**
2. Fréquence du hashtags du tweets : Ce facteur permet de calculer la popularité du hashtag dans un corpus. **[w2]**

Nous avons également analysé deux autres critères :

–**Facteurs de pertinence relatifs à la qualité des tweets**

1. Retweet : si un microbloggeur intéresse à un Tweet publié par un de ces abonnés, il va le commenter et le partager. Et dans ce cas, le nouveau message devrait être précédé par RT (retweet). **[w5]**
2. Fraicheur: ce facteur est basé sur la différence entre la date de publication d'un Tweet donné et la date de soumission de la requête, mesuré en secondes, plus la différence est petite, plus le Tweet a de chance d'être pertinent. **[2]**

I.3.3.3 Evaluation de la RI dans les microblogs

Pour évaluer une approche de RI dans les microblogs, nous aurons recours à une collection de test afin de mettre en œuvre les différentes approches de restitution de microblogs pertinents.

Pour cela, on se base sur la tache *TREC MICROBLOG*

La tâche TREC Microblog

TREC microblog est une tâche de la campagne TREC, dédié à la recherche d'information dans les microblogs, elle est organisée annuellement depuis 2011.

La tache TREC Microblog est décrite comme une tache de recherche ad hoc temps réel.

La collection de tests

La première campagne TREC MICROBLOG : TREC 2011, fournit le corpus Tweets2011, qui comprend environ 16 millions de tweets qui ont été publiés sur une période approximative de deux semaines *(du 23 Janvier 2011 au 8 Fevrier 2011)*. Le corpus est considéré comme un échantillon fiable de la "*twittosphère*". Tweets2011 comprend 50 "Topics" (ou requêtes) dont chacun représente un besoin en information à un moment donné. **[w5]**

I.4 Conclusion

Dans ce premier chapitre, nous avons en premier lieu défini les notions fondamentales de la RI. En particulier, nous avons expliqué le processus d'indexation, passé en revue différents modèles de la RI et nous avons détaillé le principe de l'évaluation des systèmes de RI.

En deuxième lieu, nous avons présenté le réseau social Twitter et ses particularités avant d'introduire les principes généraux de la RI dans Twitter.

Dans le prochain chapitre, nous nous intéressons à la contextualisation des tweets qui est la thématique principale abordée dans ce mémoire.

Chapitre II

La contextualisation des tweets

II.1 Introduction

Le service de microblogging Twitter offre aux microbloggeurs un système de communication et de collaboration qui permet de partager leurs connaissances, opinions et évènements, etc. en postant des messages courts «tweets» limités à 140 caractères, ce qui oblige les microbloggeurs d'utiliser un vocabulaire particulier ce qui rend parfois ces tweets ambigus.

Dans ce chapitre, nous allons positionner notre thématique sur l'ensemble des approches portant sur la contextualisation des messages courts notamment tweets.

II.2 La contextualisation

La contextualisation est le processus de fournir un contexte pertinent et des descriptions significatives pour une chaine de caractère donnée. Dans notre thématique, la contextualisation consiste à enrichir le contenu des microblogs en générant un bref texte représentant son contexte en utilisant des approches appropriées comme l'approche de résumé automatique qui consiste à résumer un ensemble du documents pour générer un texte cohérent formant le contexte.

II.2.1 Le résumé automatique

Le résumé automatique est l'une des taches de TALN[5] apparu dans de nombreux domaines d'activités (recherche d'information, finance, etc.) qui est un intérêt considérable car
L'humain n'est plus capable de traiter convenablement ces données pour les résumer et les analyser dans un temps bref. Dans le cas des moteurs de recherche, un système de résumé automatique (SRA) consiste à générer un texte bref et pertinent par rapport à une requête de l'utilisateur. **[7]**

[5] **TALN** : Traitement Automatique des Langues Naturelles

On distingue deux types de résumé automatique :

Résumé extractif : Il se limite à extraire des phrases complètes censées être les plus pertinentes du document et à les concaténer de façon à produire un extrait. Il est aujourd'hui l'approche la plus répandue et la plus simple. L'objectif de cette approche est de pouvoir fournir rapidement, sans analyse en profondeur du texte, un résumé à l'utilisateur.

Résumé abstractif :Il vise à rédiger un résumé en générant des phrases pas forcément contenues dans le texte source. Cette approche est la plus difficile. Elle s'efforce de produire un vrai résumé tel que le ferait un être humain à l'aide d'une analyse et d'une représentation sémantique plus ou moins profonde et Développé.

Les caractéristiques des résumés

***Portée du résumé* :** Le SRA peut être mono document ou multi document, et peut être plus ou moins adaptés à des tailles différentes de documents (résumer un article ne pose pas tout à fait le même problème que résumé un rapport scientifique).

Le SRA multi-document est plus récent, génère des résumés de taille ajustable d'un ensemble de documents.

***Générité du résumé* :** un résumé de texte peut être générique ou orienté. Dans le premier, le résumé est produit à partir de contenu du texte source indépendamment de contexte. Par contre, le résumé orienté est guidé par une tache ou une requête et dans ce cas, seule l'information qui est en relation avec la tâche ou la requête qui sera sélectionnée. Ce type de résumé dépend donc fortement du contexte, il peut être défini comme un ensemble de facteurs d'entrée du système de résumé automatique. **[7]**

Style du résumé

Un résumé peut être informatif ou indicatif. Informatif est un modèle réduit du texte d'origine relatant le plus largement possible les informations de documents.

Cependant, le résumé indicatif prend en considération les sujets les plus importants évoqués par le texte.

II.3 les travaux de contextualisation dans les microblogs

Nous allons nous intéresser dans cette partie à deux approches relatives à la contextualisation des tweets, une approche basée sur le SRI et le résumé automatique qui génère un bref résumé explicatif (500 mots dans INEX). Ce dernier devrait être construit automatiquement en utilisant des ressources externes dans le but d'extraire les passages pertinents et en les regroupant en un résumé cohérent, et une autre approche se concentrant sur la conversation sociale qui prend en considération les différents types des signaux (sociaux, temporels et textuels).

II.3.1 Approche se basant sur le résumé automatique

Cette approche considère le contexte d'un tweet donné comme un résumé prévenant d'articles Wikipédia [6] en suivant les deux étapes suivantes :

- La première consiste à rechercher les articles Wikipédia les plus pertinents contenants des informations liées au tweet initial, en utilisant un SRI,
- La deuxième consiste à résumer ces articles pertinents en extrayant les phrases pertinentes pour construire le contexte en utilisant un système de résumé automatique(SRA). **[8]**

La figure suivante montre les étapes de contextualisation des tweets en utilisant un SRI et le SRA.

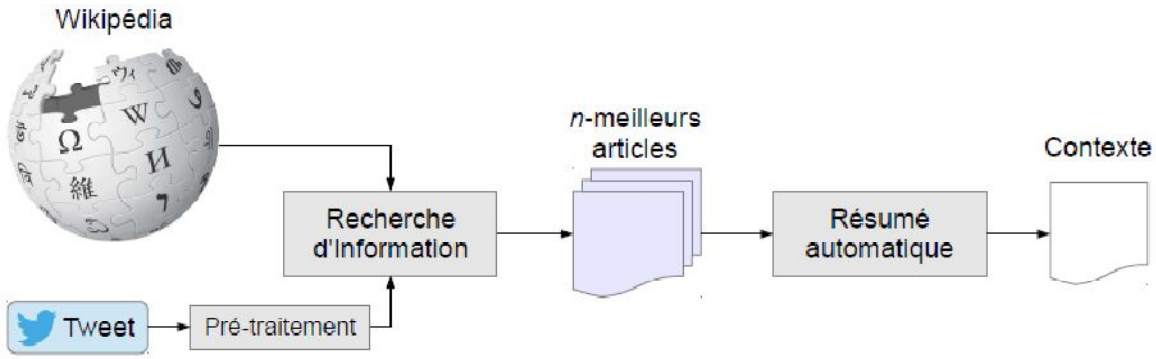

Figure II.1- Schéma de contextualisation d'un tweet à partir de Wikipédia **[9]**

[6]**Wikipédia** : est une source de connaissances croissante de haute qualité en langage naturel qui joue un rôle très important dans le web sémantique.

II.3.1.1 Récupération des articles Wikipédia pertinents

Dans cette étape, on extraie une information structurée et précise contenant des mots clés à partir d'un tweet initial en appliquant un ensemble de prétraitements. L'information extraite correspondre à des requêtes traitées par un SRI dans le but de sélectionner à partir d'une base documentaire (source d'informations comme Wikipédia) des articles les plus pertinents portants des informations contextuelles par rapport au tweet initial.

a. Interprétation des #hashtags et formatage des tweets

Le tweet peut contenir différents symboles comme les hashtags qui indiquent une information importante fournie directement par le microbloggeur.

Les microbloggeurs utilisent les hashtags avant un mot clé ou phrase pertinente (sans espace) de leurs tweets. Pour cela, les hashtags sont importants pour la récupération des articles Wikipédia liés à un tweet. Mais leurs difficultés viennent du fait qu'ils sont pour la plupart composés de plusieurs mots concaténés, comme dans l'exemple suivant.

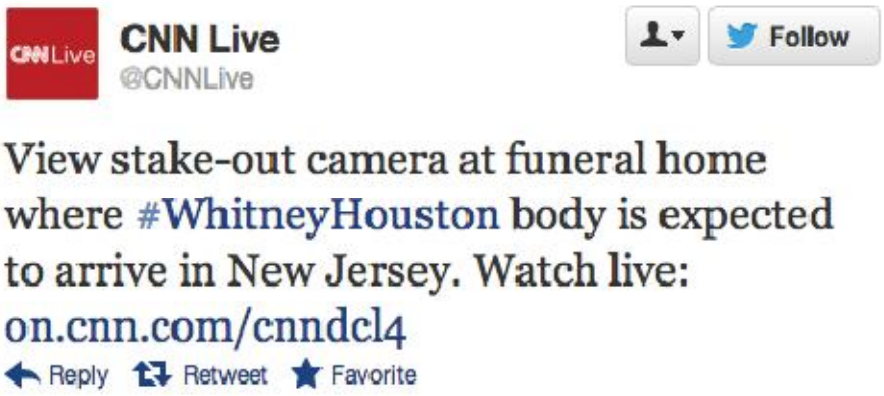

Figure II.2- Exemple d'un Tweet issu de la collection INEX Tweet Contextualization pour l'année 2012.

Pour remédier à ce problème, ils ont utilisé l'algorithme de segmentation automatique pour calculer le découpage le plus probable d'un hashtag en utilisant les probabilités d'apparition d'uni-grammes et de bi-gramme au sein du corpus bing N-Gram[7].

Enfin, chaque hashtag présent dans le tweet initial est remplacé par sa version découpé.

Dans l'exemple précédent, le hashtag #WhitneyHouston sera remplacé par les deux mots « Whitney »et « Houston ».

[7] http:/web-ngram.research.microsoft.com/info/

A ce moment-là, il est possible pour un SRI de renvoyer des documents liés à Whitney Houston étant donné que ces deux mots n'apparaissent pas dans le tweet.

Un tweet peut aussi utiliser des mentions (@pseudonyme) et les symboles RT au début d'un tweet (pour indiquer le renvoi du contenu d'un autre utilisateur), mais ces différents mots sont inutiles pour la récupération des articles et pour cela ils les suppriment simplement en utilisant la liste standard Indri[8].

Le résultat final sera un tweet complet mais sans hashtag collés, ni montions inutiles.

La figure suivante représente un exemple de formatage d'un tweet initial.

Figure II.3- Exemple de formatage d'un tweet initial

b. Recherche d'articles Wikipédia

Après les prétraitements appliqués au contenu de tweets on aura un ensemble de mots utiles constituant la requête passée au SRI pour récupérer l'ensemble d'articles Wikipédia les plus pertinents contenant les informations contextuelles du tweet.

[8] http://www.lemurprojet.org/

Modèle de base

Le modèle de base utilisé dans cette approche est le modèle de langue qui est un modèle de vraisemblance de la requête.

Soit θ_D le modèle de langue estimé en se basant sur un document D, le score d'appariement entre D et une requête T est défini par la probabilité conditionnelle suivante :

$$P(T|\theta D) = \prod_{t \in T} f_T(t, D)(t, D) \qquad (17)$$

Un des points importants dans le paramétrage des approches par modèle de langue est le lissage des probabilités nulles. Dans ce travail, θ_D est lissé en utilisant le lissage de Dirichlet,
On aura donc :

$$f_T(t, D) = \frac{c(t,D) + \mu \cdot P(t|C)}{|D| + \mu} \qquad (18)$$

Où C(t, D) est le nombre d'occurrences du mot t dans le document D. C représente la collection de documents et μ est le paramètre du lissage de Dirichlet.

Une des limitations évidente de l'approche par uni gramme est qu'elle ne tient pas compte des dépendances ou des relations qu'il peut y avoir entre deux termes adjacents dans la
Requête. Le modèle MRF (*Markov Randon Field*) est une généralisation de l'approche par modèle de langue et résout spécifiquement ce problème. L'intuition derrière ce modèle est que des mots adjacents de la requête sont susceptibles de se retrouver proches dans les documents pertinents. Trois différents types de dépendances sont considères :

- l'indépendance des termes de la requête (ce qui revient à un modèle de langue standard prenant en compte uniquement les unigrammes),
- l'apparition exacte de bigrammes de la requête,
- et l'apparition de bigrammes de la requête dans un ordre non d´défini au sein d'une fenêtre de mots.

Le modèle propose deux fonctions supplémentaires, La fonction *fO (qi, qi+1, D)* considère la correspondance exacte de deux mots adjacents de la requête. Elle est dénotée par l'indice *O*. La seconde est d´dénotée par l'indice *U* et considère la correspondance non ordonné de deux mots au sein d'une fenêtre de 8 unités lexicales.

Finalement, le score d'un article Wikipédia D par rapport à un Tweet formaté T est donné par la fonction suivante :

$S_{MRF}(T,D)= \lambda T \prod_{t\in T} f_T(t,D)+\lambda O \prod_{i=1}^{|T|-1} f_o(t_i, t_{i+1}, D)+\lambda U \prod_{i=1}^{|T|-1} f_u(t_i,t_{i+1},D)$ (19)

Où

λT, λO et λU sont des paramètres libres dont la somme est égale à 1.

Intégration de hashtags

Considérons le Tweet T suivant :

T=« #Informatique #spécialités domaine vaste et important »

L'ensemble de hashtags est définie parHT= {"informatique", "spécialités"}.

Le reste des mots du Tweet sert à spécifier ses détails.

Soient un Tweet T et ses hashtagsHT, le score d'un article Wikipédia D est donné par :

$$S(T, HT, D) = \alpha\, s_{MRF}(HT, D) + (1-\alpha) s_{MRF}(T, D) \quad (20)$$

Où

Le paramètre α permet de mesurer l'importance donnée aux hashtags seuls dans la requête.

Cependant, les hashtagspeuvent avoir une utilité parfois très limitée, comme dans l'exemple suivant :

«U Just Heard "Hard To Believe" by @andydavis on the @mtv Teen Mom 2 Finale go 2 http://t.co/iwb2JuL8 for info #ihearditonMTV»

Celle-ci «*I heardit on MTV* »est une phrase d'accroche de type publicitaire et n'apporte rien pour la compréhension du Tweet. L'importance des hashtags est aussi contextuelle et dépend de leur pouvoir discriminant. Ils ont choisis d'estimer ce pouvoir en calculant un score de clarté ce score est en réalité la divergence de *Kullback-Leibler* entre le modèle de langue de l'ensemble de hashtags et le modèle de langue de la collection C d'articles Wikipédia :

$$\alpha = \sum_{w\in V} \left(p(w|HT) \left| log \frac{P(W|HT)}{P(W|C)} \right. \right) \quad (21)$$

$$P(W|HT) = \sum_{D\in R} P(W|D)P(D|HT) \quad (22)$$

Où :

V représente le vocabulaire.

Le modèle de langue des hashtags est estimé par retour de pertinence simulé et la probabilité estimée sera calculée par le théorème de bayes suivant :

$$P(D|HT) = \frac{P(HT|D)P(D)}{P(HT)} \qquad (23)$$

Où la probabilité *P(D)=0* pour les documents qui ne contiennent aucun mot de la requête, pour cela, la probabilité *P(HT)* est constante donc elle sera ignorée.

Plus les documents utilisés pour estimer le modèle de langue des hashtags sont homogènes, plus la divergence de *Kullback-Leibler* augmente. Ainsi le paramètre α permet de quantifier à quel point les hashtags sont précis et à quel point ils permettent de sélectionner des documents distincts du reste de la collection. **[8]**

Génération des phrases candidates

Pour un Tweet donné, ils sélectionnent les **n** articles Wikipédia les plus pertinents selon l'équation : $S(T, HT, D) = \alpha\, s_{MRF}(HT, D) + (1 - \alpha) s_{MRF}(T, D)$.(24)

Chaque article est découpé en phrases en utilisant la méthode PUNKT de détection de changement de phrases mise en œuvre dans la boîte à outils *NLTK*[9]. Toutes les phrases des **n** premiers articles sont considérées comme des phrases candidates. Ils calculent ensuite différentes caractéristiques pour chacune de ces phrases qu'ils permettront de les classer et, ainsi, de former le contexte.

II.3.1.2 Le choix des phrases et formation du contexte

À partir de documents retournés dans l'étape précédente, des phrases candidates seront générées et reliées automatiquement en utilisant le SRA. Pour cela, on prend considération les étapes suivantes :

a. Choix de phrases candidates

Pour sélectionner les phrases candidates il faut prendre en considération quelques caractéristiques qui sont utilisées dans le résumé automatique. On distingue :

- Importance de la phrase vis-à-vis du document d'où elle provient,
- Pertinence de la phrase par rapport au Tweet (y compris les hashtags),

[9]**Nltk** : offre des méthodes pour tenir compte du contexte : pour ce faire, ils calculent les n-grams, c'est-à-dire l'ensemble des cooccurrences successives de mots deux-à-deux (bi-grams), trois-à-trois (tri-grams), etc.

- Pertinence de la phrase par rapport à une page web dont l'URL est dans le Tweet,
- Pertinence du document d'où provient la phrase par rapport au Tweet.

On pose :

T : un tweet nettoyé.

HT : les hashtags du tweet T.

UT : L'URL présente dans le tweet T.

S : une phrase candidate.

À cet effet, les différentes caractéristiques sont basées sur des mesures de recouvrement et de similarité cosinus entre une phrase candidate S= {m1, m2,…., mi} et un tweet T={m1,m2,……mj}.

- Soit |.| le cardinale de l'ensemble «. ».

La formule de recouvrement en mot est donnée par :

$$\text{Recouvrement(T, S)}=\frac{|S \wedge T|}{\min(|s|,|T|)} \qquad (25)$$

Soient $\vec{S}$ et $\vec{T}$ les représentations vectorielles de S et T, et $||\vec{.}||$ la norme du vecteur $\vec{.}$, la similarité cosinus est donnée par :

$$\text{Cosinus (T, S)} = \frac{\overrightarrow{S}.\vec{T}}{\sqrt{||\overrightarrow{S}||.|\vec{T}||}} \qquad (26)$$

Ces deux mesures sont calculées à partir des représentations lexicales nettoyées des phrases et des tweets.

On applique la méthode de racinisation (stemming) de l'algorithme Porter pour calculer les différentes caractéristiques.

–**Importance de la phrase dans le document**

On utilise la méthode TextRank pour estimer l'importance d'une phrase au document dans lequel elle apparait. Ou chaque document est représenté sous forme d'un graphe non orienté *« G »* dans lequel les nœuds *« V »* et les arêtes *« E »* sont définies en fonction d'une mesure de similarité qui correspond au nombre de mots commun entre ces deux phrases.

Soit # (m, s) le nombre d'occurrences du mot m dans la phrase S.

La similarité entre les phrases S_i et S_j est définie par :

$$\text{sim}(S_i, S_j) = \frac{\sum_{m \in s_i \vee s_j} \#(m, s_i) + \#(m, s_j)}{\log(|s_i| + \log(|s_j|)} \qquad (27)$$

L'importance d'une phrase est évaluée par rapport à l'intégralité du graphe. Pour cela on utilise l'adaptation de l'algorithme PAGERANK qui inclut les points des arêtes.

Le score de chaque sommet V_i est calculé itérativement jusqu'à la convergence par :**[9]**

$$P(vi) = (1-d) + d \times \sum_{v_i \in voisins(v_i)} \frac{\text{sim}(si, sj)}{\sum v_k \in voisins(v_j).\text{sin}(sk, si)} \; p(v_j) \qquad (28)$$

Voisins (v_i) représente l'ensemble des nœuds connectés à v_i, le score de la phrase S correspond au score du nœud qui la représente dans le graphe :

$$C_1 = P(S) \qquad (29)$$

–**Pertinence de la phrase par rapport au tweet**

À ce niveau, on sélectionne les phrases donnant des informations contextuelle par rapport au tweet qui est retourné par le recouvrement et la similarité cosinus entre un tweet T et la phrase S donné par :

$$C_2 = \text{recouvrement}(T, S) \qquad (30)$$

$$C_3 = \text{cosinus}(T, S) \qquad (31)$$

Tout en gardant la logique de l'utilisation des hashtags, ils calculent le recouvrement et la similarité cosinus entre chaque phrase et les hashtags.

$$C_4 = \text{recouvrement}(HT, S) \qquad (32)$$

$$C_5 = \text{cosinus}(HT, S) \qquad (33)$$

–**Pertinence de la phrase par rapport à la page web**

Les URL trouvés dans les tweets sont des liens pointés vers les pages web qui portent des informations contextuelles.

Pour calculer la similarité entre la phrase candidate et la page web on utilise aussi le recouvrement et la similarité cosinus entre la phrase candidate et le titre(UT) de la page web.

C_6 = recouvrement (titre(UT), S) (34)

C_7 = cosinus (titre(UT), S) (35)

De la même façon, ils calculent deux mesures entre le contenu entier page UT et la page web et une phrase candidate.

C_8 = recouvrement (page(UT), S) (36)

C_9 = cosinus (page (UT), S) (37)

–**Pertinence du document par rapport au tweets**

Pour mesurer le degré de similarité d'un document par rapport à un tweet et ces hashtags, normalisé sur l'ensemble R de tous les documents renvoyés.

$$C_{10} = \frac{S(T,HT,D)}{\sum_{D' \in R} S(T,HT,D')} \quad (38)$$

–**Score final d'une phrase candidate**

Le score d'importance de chaque phrase candidate est obtenu par la formule suivante :

$$\text{Score} = \sum_x \log(c_x + 1) \quad (39)$$

b. Génération du contexte

Le contexte d'un tweet est un regroupement des phrases candidates les plus pertinentes, Mais il est possible que le contexte obtenu possède des phrases redondantes. Pour remédier à ce problème, on ajoute une étape supplémentaire lors de la génération des contextes. On génère tous les contextes possibles à partir des combinaisons des N phrases ayant les meilleurs scores, en faisant attention à ce que le nombre totale de mot ne dépasseras pas 500 mots et qu'il soit impossible d'ajouter une autre phrase sans dépasser ce seuil.

La valeur N est fixée au nombre minimum de phrases qui ont de meilleur score pour atteindre 500 mots, plus quatre phrases.

On conservant l'ordre original du document, pour améliorer la lisibilité du contexte généré en cas de deux phrases candidates qui sont extraites à partir d'un même document.

II.3.1.3 Protocole d'évaluation

Les organisateurs de la collection de teste de la tache de *Tweetcontextualisation* ont proposé unemesure d'évaluation qui calcule une divergence entre le contexte produit et le contexte de tous les participants.

II.3.2Approche se basant sur conversation la Twitter

Cette approche se base sur des conversations Twitter[10] pour fournir le contexte d'un tweet donné. Elle se décompose en deux étapes :

1. La première consiste à récupérer les conversations Twitter pertinentes contenant des informations relatives au tweets initial (Expansion de tweet),
2. La deuxième consiste à extraire les tweets pertinents pour construire le contexte. **[9]**

La figure suivante montre les deux étapes ci-dessus.

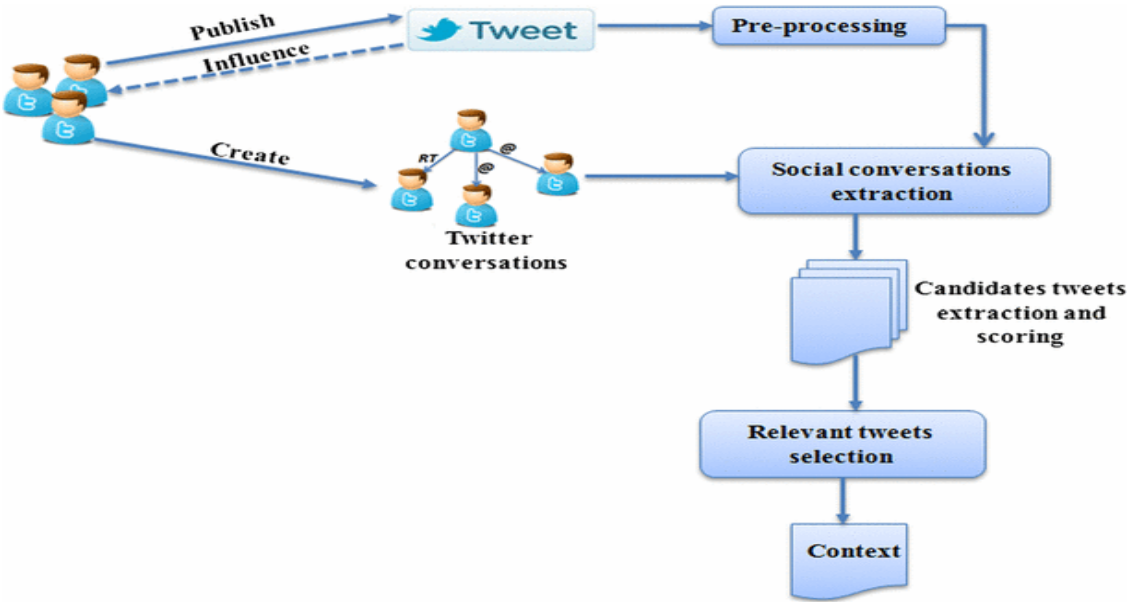

Figure II.4-La contextualisation de tweet en utilisant la conversation Twitter **[9]**

Dans ce qui suit, nous détaillons les processus mis en œuvre dans chacune de ces étapes.

II.3.2.1 Récupération de la conversation Twitter pertinente

Dans cette étape on récupère la conversation Twitter pertinente contenant des informations qui peuvent constituer le contexte d'un tweet donné. Pour récupérer cette conversation on précède comme suit :

[10]**La conversation Twitter** est un ensemble de tweets publiés par les microbloggeurs portant sur un même sujet donné. Ces tweets peuvent être directement répondu à d'autres utilisateurs en utilisant « @username » ou indirectement par retweeter d'autres interactions possibles (favori).

a. Formatage du tweet initial

La première étape qu'ils effectuent consiste à appliquer un ensemble de pré- traitements au Tweet initial. Il s'agit de formater le contenu de ce dernier en supprimant toutes les mentions retweet (RT), les mentions d'utilisateur (@username) et les mots d'arrêt des tweets.

La sortie finale du processus de mise en forme est un ensemble de hashtags. En outre, l'ensemble final des hashtags est utilisé comme une requête en mots-clés courts pour faciliter la récupération des conversations.

b. Récupération de la conversation Twitter

Pour construire une conversation Twitter pertinente, les tweets qui sont en relation avec le tweet initial sont sélectionnés, en retournant les données qui le concerne (date de publication, le texte du message, son auteur, et les entités contenus dans les messages tels que : hashtags, mentions, URLs, etc.) **.**D'autre part, l'ensemble des tweets ayant les même hashtags/URLs que le tweet initial sont tirées de Twitter pour compléter la collection de conversation utilisée par la suite pour construire le contexte.

II.3.2.2 Génération du contexte

Dans cette étape on calcule le score de chacun des tweets récupérés dans l'étape précédente pour les classer et sélectionner les tweets de meilleur score comme suit :

a. Calcul de score du tweet

Cette étape consiste à déterminer si un tweet donné est intéressant en tenant comptes de ces trois éléments suivants :

1. La pertinence sociale du tweet : L'importance d'un tweet donné est calculée en utilisant l'influence sociale.
2. La pertinence d'un tweet par rapport au tweet initial : On calcule la similarité cosinus entre le tweet candidat t et le tweet initial t_i.
3. La pertinence d'un tweet par rapport à l'URL : On calcule le chevauchement des mots et la similarité cosinus entre un tweet candidat t et le contenu de la page liée.
4. La pertinence d'un tweet par rapport aux hashtags.

1. Pertinence sociale du tweet

L'influence de tweet est basée sur l'influence sociale qui utilise le modèle d'interaction User-Tweet, comme : poster, répondre, mention et retweet, etc. qui sont indiqués dans la figure suivante :

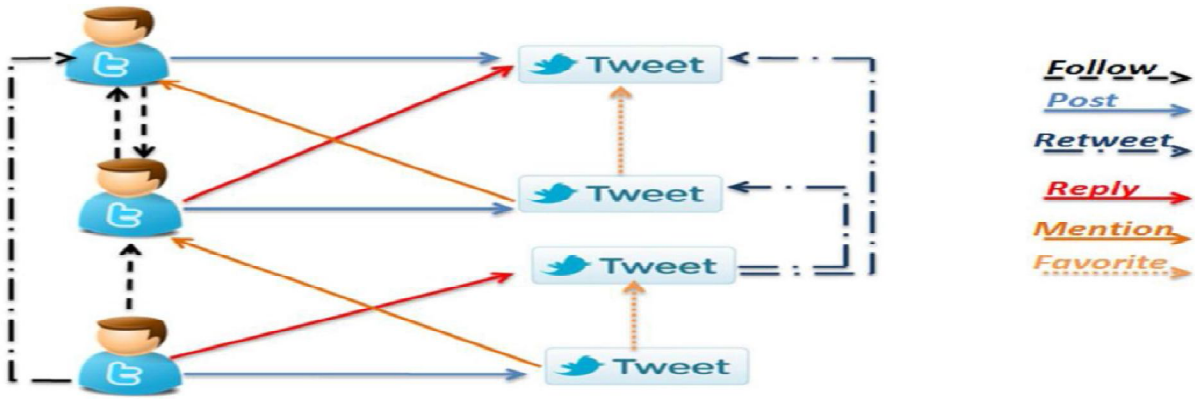

Figure II.5- Les relations existantes entre Microbloggeur-tweet **[10]**

Ces différentes relations sont utilisées pour mesurer le score d'influence du tweet et sélectionner les tweets candidats. On distingue deux éléments essentiels pour la mesure d'influence du tweet :

a. Score d'influence du tweet : désigne les caractéristiques particulières d'un tweet donné,
b. Score d'influence de l'auteur du tweet : désigne les différentes caractéristiques qui représentent l'influence de l'auteur U du tweet.

a. Mesure d'influence du tweet

L'influence du tweet est déterminée par *reply, retweet* et *l'influence préférée*.

–**Score d'influence (t)** de **Reply** : Cette influence peut être mesurée par le nombre de réponses que le tweet reçoit. L'influence de réponse est définie comme suit :

$$\text{Influence-Reply}(t)=\alpha*\text{nombre\_reply}(t) \quad (40)$$

Où $\alpha \in (0, 1]$

–**Score d'influence(t) de Retweet :** Cette influence peut être quantifiée par le nombre de retweet. Il est défini comme suit :

$$\text{Influence-Retweet}(t)=\beta*\text{nombre\_retweet}(t) \quad (41)$$

Où $\beta \in (0, 1]$

–**Score d'influence préférée(t) :** Lorsqu'un utilisateur marque un tweet favori, le contenu de ce tweet est donc utile et pertinent. Cette influence peut être déterminée par le nombre de favori que le tweet reçoit. Il est défini comme suit :

$$\text{Influence-Favorite}(t)=\gamma*\text{nombre\_favorite}(t) \quad (42)$$

Où $\gamma \in (0, 1]$.

L'utilisation du concept temporel peut fournir des informations précieuses pour le problème de la contextualisation des tweets c'est-à-dire, un tweet récent a une grande chance d'avoir de plus grandes influences en le comparants avec un ancien tweet. Pour cela, on utilise le noyau gaussienne pour calculer une différence Δt entre le temps de tweet racine t_i et le temps des autres tweets t dans la même conversation, à savoir, $\Delta t = | t - t_i |$. Il est défini comme suit :

$$\Gamma(\Delta t) = e^{-\Delta t^2 / 2\sigma^2} \quad (43)$$

Où $\sigma \in R_+$

Enfin, le score d'influence tweet est défini comme suit :

Influence-Tweet(t)=Γ(Δt)*Influence-Reply(t)+Influence-Retweet(t)+Influence-Favorite(t)(44)

b. La mesure d'influence de l'auteur Tweet

Dans cette approche, la relation de Follow et la relation de Mention.

- **Influence de Mention :** Le score d'influence de mention est défini comme suit :

Influence-Mention=δ*nombre_Mention(U) (45)

Où δ∈ (0, 1] et U signifie l'utilisateur.

- **Influence de Follow :** Un microbloggeur peut être suivi par de nombreux autres utilisateurs. Le score suivi d'influence est définie comme suit :

Influence-Follow(U)=ω*nombre_Follow(U) (46)

Où ω∈(0, 1] et U signifie l'utilisateur.

Enfin, le score d'influence de l'auteur du tweet est défini comme suit :

Influence-Auteur-Tweet(U)= Influence-Mention+ Influence-Follow(U)

2. La similarité d'un tweet par rapport au tweet initial

On attribue un score à chaque tweet de la conversation. Pour cela, à partir de chaque tweet candidat t dans une conversation c, un vecteur V est dérivé comme suit :

$\vec{V}$= {W_1, W2,……W_i}qui représentons un ensemble de mots en utilisant le modèle vectoriel.

Pour déterminer si un tweet est fortement similaire au tweet initiale on utilise la similarité cosinus qui calcule la similarité entre le vecteur tweet initiale V_{tin}et les autres vecteurs V_t de tweets de conversation. Pour mesurer au quel point un tweet serait lié au contenu de tweet initial on précède comme suit :

$$\text{Cosinus}\ (\overrightarrow{V_T},\vec{V}_{Ti}) = \frac{\overrightarrow{V_T}.\overrightarrow{V_{Ti}}}{||\overrightarrow{V_T}||.||\overrightarrow{V_{Ti}}||} \quad (47)$$

3. La pertinence d'un tweet par rapport à l'URL

Pour les tweets contenant des URLs, on télécharge la page et on extrait son titre ainsi que son contenu.

On calcule pour chaque tweet candidat t les éléments suivants :

– Le chevauchement de mot entre un tweet candidat t et les titres de la page web, et entre t et le contenu de la page.

– La similarité cosinus entre t et le titre de la page, et entre t et le contenu de la page web.

4. La pertinence d'un tweet par rapport aux hashtags

Le symbole #, appelé hashtags, c'est une information importante sur un tweet, sont des étiquettes qui ont été générées par des microbloggeurs .Hashtag est utilisé pour marquer un sujet dans un tweet ou pour suivre une conversation. Les hashtags peuvent aussi être utilisés pour créer un complexe tweet implicite. Pour collecter des tweets candidats qui partagent les mêmes informations que le tweet initial Ils ont utilisé cette fonctionnalité :

$$F1\,(t, ti) = \begin{cases} 1 \text{ si t contient le même hashtag,} \\ 0 \text{ , sinon} \end{cases} \quad (48)$$

b. Génération du contexte

Pour générer le contexte du tweet initial, les tweets sont passés pour un système d'apprentissage superviseur qui peut les convertir en fonctionnalités, puis lancer la tâche de récapitulation du contexte de tweets en un problème d'apprentissage supervisé. Après avoir formé un modèle, ils pourraient annoncer quelques tweets en tant que résumé pour tous les tweets dans quelques arborescences contextuelles. Ils ont choisi l'algorithme de l'arborescence de décision améliorée (GBDT) pour apprendre un modèle non-linéaire. GBDT est un algorithme de régression additive composé d'un ensemble d'arbres.

II.3.2.3Protocole d'évaluation

L'évaluation de la contextualisation des tweets se base à la fois sur l'informativité et la lisibilité **:**

- **L'informativité** : vise à mesurer à quel point le résumé explique le tweet ou la façon dont le résumé aide l'utilisateur à comprendre le contenu du tweet. Son objectif est d'évaluer la sélection des tweets pertinents, c'est-à-dire, le résumé des 20 meilleurs tweets, pour chaque tweet initial est sélectionné pour l'évaluation. Ce choix est effectué en fonction du score attribué par la contextualisation automatique des tweets du système. Pour cela on a utilisé La di-similarité entre un résumé humain sélectionné et le résumé proposé (selon notre approche) qui sera représenté par la formule suivante :

$$\text{Dis(T, S)} = \sum_{t \in T} (P - 1) \times \left(1 - \frac{\min(\log(P), \log(q))}{\max(\log(P), \log(q))}\right) \quad (49)$$

Où $P = \frac{f_T(t)}{f_T} + 1$ et $q = \frac{f_S(t)}{f_S} + 1$

S est l'ensemble des tweets informatifs présentés dans notre résumé proposé. Et T est l'ensemble des termes présentés dans le résumé de référence. Pour chaque terme t∈ T $f_T(t)$ Représente la fréquence d'occurrence de t dans le résumé de référence et $f_S(T)$ Sa fréquence d'occurrence dans le résumé automatique.

Plus le *Dis (T, S)* est faible, plus le résumé proposé est similaire à la référence.

T peut prendre les trois formes suivantes :

- Unigrams fait de lemmes uniques.
- Bigrams fait de paires de lemmes consécutifs (dans la même phrase).

- **La lisibilité** : vise à mesurer et la clarté et la facilité de la compréhension du résumé.

En revanche, la lisibilité est évaluée manuellement, où chaque résumé a été évalué en considérant les paramètres suivants :

Pertinence : juger si le tweet avait du sens dans leur contexte (c'est-à-dire après avoir lu les autres tweets dans le même contexte). Chaque évaluateur humain (étudiants dans notre cas) a dû évaluer la pertinence avec trois niveaux, à

savoir hautement pertinent (valeur égale à 2), pertinente (valeur égale à 1) ou non pertinente (valeur égale à 0).

Non-redondance : l'évaluation de la capacité du contexte ne contient pas trop d'informations redondantes, c'est-à-dire des informations qui ont déjà été données dans un précédent tweet. Chaque évaluateur humain (étudiants) a dû évaluer la redondance avec trois niveaux, à savoir non redondant (valeur égale à 2), redondant (valeur égale à 1) ou hautement redondant (valeur égale à 0). **[9]**

II.3.3 La comparaison entre les deux approches

Dans la première approche citée précédemment, Wikipédia est la source utilisée pour extraire des passages pertinents représentant le contexte d'un tweet donné.

Cependant, cette source n'est pas directement disponible pour informer les microbloggeurs sur un évènement d'actualité tels que les catastrophes naturelles. Pour cela, la deuxième approche basée sur la conversation sociale sur Twitter permet d'obtenir plus d'informations en utilisant plusieurs signaux tels que les signaux sociaux (hashtags et Url), des signaux temporels et des signaux textuels. La conversation Twitter est une source d'information produite tous les jours, ce qui peut améliorer la tache de contextualisation des tweets. On s'intéresse à cette dernière dans le chapitre suivant.

II .4 Conclusion

Dans ce chapitre, ils ont passé en revue deux approches distinctes qui exploitent plusieurs systèmes afin de résoudre la problématique de rendre les tweets en contexte.

Dans le prochain chapitre, nous présenterons le principe général de l'approche de contextualisation.

Table des matières

Chapitre III

Approche de contextualisation se basant sur la conversation sociale

III.1 Introduction

Dans le chapitre précédent, nous avons passé en revue deux approches de recherche effectuées dans le domaine de contextualisation des microblogs. La première se basant sur le résumé automatique et la deuxième se basant sur la conversation sociale Twitter.

Nos travaux portent sur la proposition d'une approche de RI dans Twitter qui permet La contextualisation des tweets. Dans ce contexte nous nous basons sur l'approche de conversation sociale de **[w3]** que nous proposons d'améliorer.

III.2 Principe général de l'approche

Notre approche de contextualisation est basée sur la conversation sociale dans Twitter qui tient compte de la pertinence textuelle des tweets et de la pertinence sociale estimée au travers des informations sociales issues du réseau Twitter.

Nous avons commencé à chercher des conversations Twitter en tenant compte de leurs tweets et des éléments sociaux qui leur sont relatifs.

Pour construire une conversation, nous avons commencé par obtenir un ensemble important de tweets. D'autre part, nous avons décidé d'associer à la conversation proprement dit l'ensemble des tweets contenant le même hashtag que les tweet initial ainsi que ses réponses (Reply).

Nous illustrons tout cela dans la table d'algorithme suivante :

Algorithme1 : Récupérer la conversation Twitter pertinente

Début

Var C : chaine ; // Conversation Twitter pertinente

t, t_i, U: chaine ; // tweet, tweet initial, utilisateur

IdTweet : entier ; // id de tweet initial

H : liste ; // liste des hashtags présents dans t_i

- Initialiser le Tweet initial : t_i ;

Pour chaque tweet t de twitter **Faire**

Si (InreplytoStatusId(t)= IdTweet) **alors**

C←C+t ;

Fait

H ← Hashtags contenant dans ti ;

Pour chaque hashtag de H **Faire**

Pour chaque tweet t de twitter **Faire**

Si t contient le même hashtag que ceux du H **alors**

C← C+t ;

Fait

Fait

Fin

Pour déterminer si un tweet donné est influant, nous avons calculé sa pertinence sociale, sa pertinence par rapport au tweet initial, sa pertinence par rapport à l'URL et sa pertinence aux hashtags.

Nous avons commencé par calculer la pertinence sociale en se basant sur le modèle d'interaction User-Tweet, comme : poster, retweeter, etc. Pour cela, nous avons exploité deux types de score pour la mesure de l'influence des tweets :

- Score d'influence du tweet : Se réfère à ces caractéristiques qui représentent les caractéristiques particulières du tweet telles que l'influence de la réponse.
- Score d'influence de l'auteur du tweet : se réfère à ces caractéristiques qui représentent l'influence de l'auteur des tweets.

Nous avons calculé le score d'influence du tweet en utilisant la table d'algorithme suivante :

Algorithme2 : score d'influence du tweet

Var t : chaine ; //tweet de collection c

Début

Pour chaque tweet t **faire**

Influence-Reply(t) ← $\propto * nombre\_Reply(t)$;

Influence-Retweets(t) ← $\beta * nombre\_Retweet(t)$;

Influence-Favorite(t)← $\gamma * nombre$_Favorite(t);

Fait

// $\propto, \beta\ et\ \gamma \in [0,1]$.

$\Delta t = |T_t - T_{ti}|$ // T c'est le taux de tweet

Gausienne_Kernel ← exp $(- {\Delta t\text{^}2}/{2\sigma\text{^}2})$

// Δt^2 Peut être reconnu comme la distance euclidienne au carré entre les deux vecteurs caractéristiques. σ est un paramètre gratuit $\in$ R+

- Sommer les quatre résultats précédents pour obtenir l'influence de tweet lui même.

Influence _Tweet (t) ←Gausienne_Kernel∗ Influence-Reply(t) + Influence- Retweets(t) + Influence-Favorite(t) ;

Fin

Et le score d'influence de l'auteur de tweet, est représenté par la table d'algorithme suivante :

Algorithme3 : score d'influence de l'auteur de tweet

// Calcul de l'influence de l'auteur U de Tweet t représenter par les formules suivante

Début

Pour chaque utilisateur U **faire**

Influence-Mention(U) ←δ*nombre_Mention(U) ;

Influence-Follow(U) ←ω*nombre_Follow(U) ;

Fait

// δ *et* $\omega \in [0,1]$.

- Sommer les deux résultats précédents pour obtenir l'influence de l'auteur tweet.

Influence-Auteur-Tweet(U)← Influence-Mention+ Influence-Follow(U) ;

Fin

Pour déterminer la pertinence de chaque tweet de la conversation par rapport au tweet initial, nous avons utilisé le modèle vectoriel pour représenter les tweets sous forme des vecteurs. Par la suite, nous avons calculé la similarité cosinus pour mesurer la similarité entre le vecteur tweet initial et un autre vecteur tweet dans la même conversation.

Nous avons illustré tout cela par la table d'algorithme suivante :

Algorithme 4 : La pertinence de chaque t de C par rapport au ti

Var t, ti : chaine ; //t de C, tweet initial

Début

Appeler Consine_Similarity(t, ti) ;

Fin

Fonction Consine_Similarity (chaine S1, chaine S2) : int {

Var :

V_{S1}, V_{S2} : Vecteur

Début

- Dériver un Vecteur V_{S1} de S1

V_{S1}= {a_1, a_2,..., a_n} ; // Initialisation de vecteur V_{S1}

- Dériver un Vecteur VS2 de S2

V_{S2}= {b_1, b_2,...b_n} ; // Initialisation de vecteur V_{S2}

$\overrightarrow{Vs1} * \overrightarrow{Vs2}$= ($a_1$*$b_1$ + a_2*b_2 + ... + a_n*b_n) ;

$||\overrightarrow{Vs1}||$ = sqrt (a_1^2 + a_2^2 + ... + a_n^2) ;

$||\overrightarrow{Vs2}||$ = sqrt (b_1^2 + b_2^2 + ... + b_n^2) ;

Cosinus $(\overrightarrow{Vs1}\ \overrightarrow{Vs2}) = \overrightarrow{Vs1} * \overrightarrow{Vs2} \Big/ ||\overrightarrow{Vs1}|| * ||\overrightarrow{Vs2}||$

Fin } //Fin de Fonction

Nous avons utilisé aussi les différents signaux sociaux (url, hashtags) pour déterminer si un tweet de la conversation est influent.

La table d'algorithme suivante résume ce que nous avons déjà dit.

Algorithme 5 : La pertinence de chaque tweet t ∈ C par rapport à l'URL et hashtags

Var i , n : int ;

Var t, contenu : chaine ;

Début

L_{url} ← les URL contenant dans chaque t

Retourner la longueur de la liste : $|L_{url}|=n$;

Pour (i de 0 à n) **faire**

Télécharger :

- La page web.

Pour chaque page web **faire**

Extraire le titre de la page web.

Extraire le contenu de la page web.

Fait

- Calculer le nombre de chevauchement de mot entre t et le titre de la page web
- Calculer le nombre de chevauchement de mot entre t et le contenu de la page web

Comme nous avons définit la fonction de similarité cosinus dans l'algorithme précédent, on va utiliser la même fonction ici :

Appeler similarité cosinus (t, titre) ;

Appeler similarité cosinus (t, contenu) ;

Fait

```
        //La pertinence de t par rapport aux hashtags
        hashtagT← les hashtags de t      //Listes des hashtags de chaque tweet
        Pour chaque hashtagT de t Faire
            Si hashtagT présent dans le t_i
                Retourner 1 ;
            Sinon
                Retourner0 ;
            FinSi
        Fin pour
Fin
```

Par comparaison avec l'instar du **[9]**, qui a utilisé les hashtags seulement pour classer les tweets. Nous les avons de plus employés pour enrichir le contexte final du tweet initial en raison des informations importantes que portent leurs signaux.

A cet effet, nous les avons découpés pour recueillir les informations textuelles qu'ils contiennent en appliquant les N-grammes de lettres illustrés dans l'algorithme suivant :

Algorithme 6 : Découpage des hashtags

```
Var Freq : entier ;
Fonction Calcule_fréquence (hashtag : chaine): entier
Début
    Pour chaque hashtag faire
        Freq←Calculer la fréquence de chaque hashtag ;
    Fait
        Retourner Freq ;
        Freq← nouvelle freq ;
Fin
Début
    Extraire les hashtags de C
    ListH← les hashtags pertinent les mieux classés ;
```

```
Retourner la longueur de liste : |ListH|=n ;
list : liste ;
Pour (i de 0 à n) faire
    m←taille de ListH[i] ;          //m taille du i^éme mot de la liste
    Pour (inti=2 ;i<= m ; i++) Faire
        list←N_grams (ListH[i]) ;       //générer les N-grams
    Fait
Fait
    Charger le dictionnaire ;

Pour    chaque élément de list Faire

  Si l'élément est présent dans le dictionnaire alors

    On le garde ;

Fait

Garder l'ordre originale ;
Retourner le découpage des mots de la liste ;

FIN
```

Nous avons utilisé aussi les URLs pour l'ajout d'informations supplémentaires au contexte final, chose qui n'a pas été faite par **[9].**

Algorithme 7 : Extraire le contexte

```
Var L : liste
Fonction retour_texte_brut _de_tweets (hash, url, t : chaine):int {
 Var   Texte_brut : chaine ;  //variable de retour
 Début
    Pour chaque tweet t faire
        Pour tous  hash de t faire
              t ← t / hash ;
              texte_brut← t ;
             Pour tous  url de t faire
                 t← t / url ;
                 texte brut← t ;
             Fait
        Fait
        Si nbrtweet>= 2 alors
            Appeler  Supp_redendance ()
        FinSi
        RETOURNER  texte_brut ;
    Fait
FIN       } //fin fonction retour_texte_brut _de_tweets

Procédure Supp_redendance (tweet : chaine)
 Var  Nbrtweet    // membre de même tweet da&ns la conversation C
 Début
    Pour tous tweet ∈ C faire
        Si Nbrtweet>= 2 alors
            Supprimer la redondance ;
        FinSi
    Fait
Fin   //fin procédure Supp_redendance
Fonction Return_contexte (texte : chaine): chaine {
```

```
Var   resume : chaine ;
 DEBUT
    resume ← hashtags découpés + résumé automatique de texte  + URL
                                                          pertinents
    Retourner resume ;
 Fin     } //fin fonction
 Début
    Pour chaque t faire
        Trier les scores retournés par tri décroissant ;
        Extraire les URL présent dans c ;
       Pour chaque URL faire
            Calculer la fréquence ;
        Fait
        Classer les URL ;
        Retourner liste URL pertinente ;
        Pour chaque hashtags présents dans le t_i de chaque t ∈ C faire
            L ← Supprimer() ;
       Fait
        Extraire les hashtags restés dans chaque t ;
        Hash←Calcule_fréquence () ;
        Classer les Hashtags ;
        Retourner ListH ;    // liste des hashtags mieux fréquents
        Pour chaque élément de ListH faire
            Découpage de hashtags ;
        Fait
        Passé le texte brut au système de résumé automatique ;
        Appeler Return_contexte () ;  // le nombre de mots<=500
    Fait
 Fin
```

III.3 Conclusion

Dans ce chapitre nous avons décrit en détail notre approche ainsi que tout ce qu'elle a apporté comme modifications, nous avons présenté toutes les étapes proposées dans notre méthode sous forme algorithmique.

Dans le prochain chapitre, nous allons voir l'implémentation de notre approche ainsi que l'évaluation de ses résultats.

Chapitre IV

Implémentation et expérimentations

IV.1 Introduction

Dans ce chapitre, nous présentons les expérimentations effectuées pour évaluer l'approche proposée dans le chapitre précédent.

Dans ce qui suit, nous décrivons les différentes collections des tweets utilisées et nous présentons le cadre expérimental ainsi que les mesures d'évaluation utilisées.

Nous allons ensuite comparer les résultats de notre approche avec des résumés humains, dans le but de déterminer la fiabilité et la performance de notre approche.

Enfin, nous allons tirer des conclusions à partir de l'étude et l'analyse des résultats de notre approche que nous avons effectuées.

IV.2 Implémentation

Nous avons implémenté notre approche sous Java en utilisant les bibliothèques Lucene et développé ce projet sous l'IDE NetBeans. Dans notre implémentation les tweets sont extraits à travers l'API Twitter.

IV.2 .1 Description du matériel utilisé

Afin de réaliser cette application dans les conditions les plus favorables, nous avons disposé d'un micro-ordinateur portable ayant les configurations suivantes :

- Micro-processeur : CORE i3,
- Fréquence d'horloge : 2 .10 GHz,
- RAM : 4Go.

IV.2.2 Environnement de développement

–***le langage JAVA***

Le langage Java est un langage de programmation informatique crée par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy

(confondeur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995au Sun World.

Java est inspiré des langages C et C^{++}.Comme C^{++}, Java fait partie de la grande famille « des langages orientés objets ». Il répond donc aux trois principes fondamentaux de l'approche orientée objet (POO) :l'encapsulation, le polymorphisme et l'héritage.

–*NetBeans IDE*

NetBeans est à l'origine un EDI Java, NetBeans fut développé au départ par une équipe d'étudiants à Prague, racheté ensuite par Sun Microsystems. En 2002, Sun a décidé de rendre NetBeans open-source. L'IDE NetBeans[11] est donc un environnement de développement intégré permettant d'écrire, compiler, et déployer des programmes Java. En plus de Java, il supporte différents autres langages, comme Python, C^{++}, XLM, Ruby, PHP et HTML.

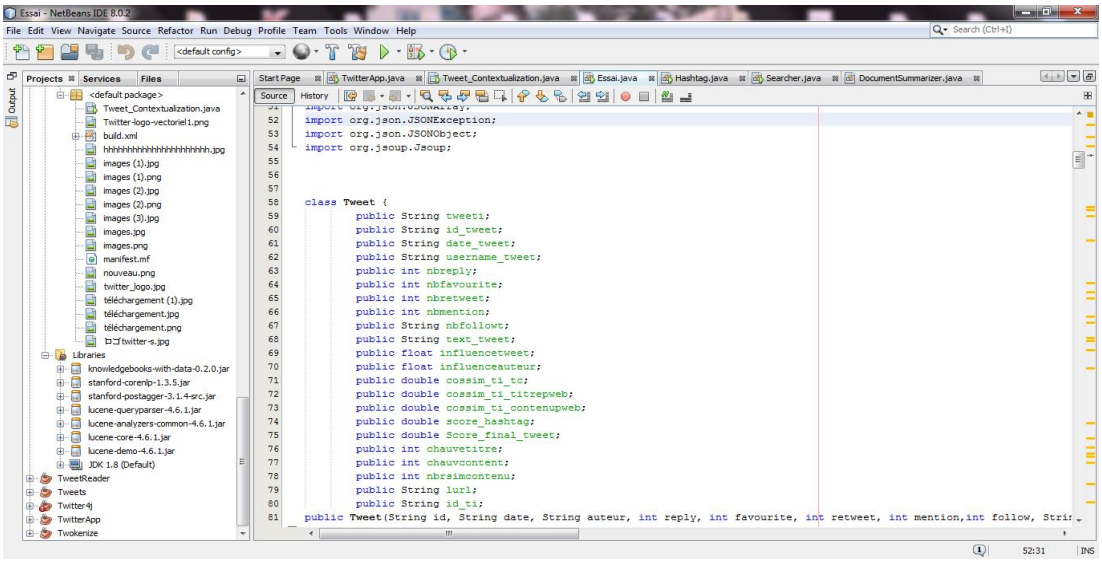

Figure IV.1- Interface de NetBeans ID

[11] https://netbeans.org/

IV.2.3 Bibliothèques java utilisées

Pour réaliser notre travail, nous avons utilisé des bibliothèques java qui contiennent des fonctions utiles que l'on ne désire pas à chaque fois réécrire.

Voici une description des bibliothèques que nous avons utilisées :

– ***Lucene***

Lucene est un moteur de recherche textuelle Open Source, utilisé dans des applications commerciales ou Open Source. Il se concentre surtout sur l'indexation et la recherche.

Lucene apporte aussi des capacités de recherche à l'EDI Eclipse, Nutch (un moteur open source de recherche Web), il a été porté vers beaucoup d'autres langages de programmation, par exemple Perl, Python, C++, et .NET.

– ***Twitter4J***

Twitter4J est une bibliothèque non-officielle de JAVA permettant d'intégrer facilement l'API de Twitter dans toute l'application JAVA, la librairie propose différentes classes et méthodes permettant de manipuler les méthodes qu'offre l'API Twitter.

Parmi les caractéristiques du Twitter4j on distingue :

- Elle fonctionne sur toutes les versions JAVA Plateforme 5 ou version ultérieure,
- Elle fonctionne sur les plateformes Android,
- Zéro dépendance : Aucune autre bibliothèque n'est requise,
- Elle utilise le support d'authentification OAuth.

– ***JSoup***

Est une bibliothèque Java open source de méthodes conçues pour extraire et manipuler des données stockées dans des documents HTML.

IV.2.4 Interface de l'application

Pour mieux éclaircir notre travail, nous avons pensé à ajouter une interface qui se présente comme une fenêtre GUI, permettant de recevoir une requête, dans notre cas, il s'agit d'un tweet initial, dans le but de le contextualiser.

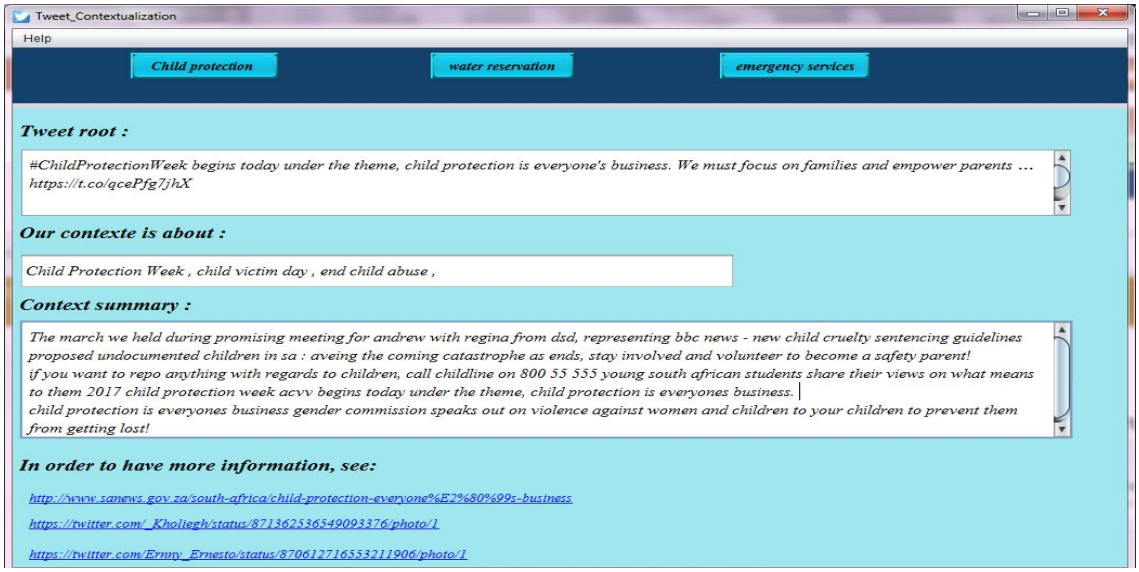

Figure IV.2- Présentation de l'application

Et pour avoir plus d'information, le microbloggeur peut consulter un des trois liens proposés, comme le montre la figure suivante

Figure IV.3- utilisation de l'un des liens de l'application

IV.2.5 Fonctionnement

Toutes les expérimentations réalisées dans ce chapitre ont été effectuées sur trois collections définies comme suit : chaque collection de test utilisée est constituée de tweets extraits à l'aide de l'API Twitter. Ces collections incluent envers 120 tweets chacune.

Nous avons indexé le texte et le score de chaque tweet de la collection avec le moteur de recherche Lucene. Par la suite, ces tweets indexés sont classés selon l'ordre décroissant de leurs scores.

Afin de créer un résumé de référence, nous avons effectué une étude par un groupe d'étudiants du département anglais à « Hasnaoua » qui peut être utile pour évaluer nos résultats. Ainsi, pour chaque tweet initial, nous avons considéré seulement le texte brut des 30 meilleures tweets classés qui sera donné à l'outil de résumé automatique pour qu'il génère le contexte final qui sera passé aux étudiants avec le tweet initial et nous avons demandé à chaque étudiant d'ouvrir chaque URL de celles proposées pour avoir une idée de ce tweet. Ensuite, ils ont généré des résumés en sélectionnant 5 à 10 tweets apparaissent importants pour eux en tant que contexte, qui étendent le tweet initial en fournissant des informations supplémentaires.

Pour estimer la qualité des résultats en termes de lisibilité produits selon notre approche, nous avons utilisé une évaluation manuelle réalisée aussi par un groupe d'étudiants, illustré dans le tableau suivant :

	user 1	user 2	user 3	user 4	user 5	user 6	user 7	user 8	user 9	user 10
Sujet 1										
Pertinence	1	1	1	0	1	1	1	1	1	1
Non redondant	1	1	1	1	0	1	1	1	1	1
Sujet 2										
Pertinence	1	1	0	1	1	1	1	0	1	0
Non redondant	1	1	1	1	1	1	0	1	1	1
Sujet 3										
Pertinence	2	1	0	0	1	1	1	1	1	1
Non redondant	1	1	0	1	1	1	0	1	0	1

Tableau 1- résultats de lisibilité faits manuellement

IV.3 Protocole d'évaluation

Pour évaluer notre approche qui sert à contextualiser les tweets, nous avons utilisé les mesures d'évaluation suivante : **l'informativité** et **la lisibilité**

Notre évaluation va se faire comme suit : nous allons effectuer une série de traitements sur notre approche puis nous allons comparer les résumés obtenus automatiquement avec des résumés de référence.

Puis nous comparons ces résultats et nous les analyserons à l'aide du logiciel Active Perl[12].

Un bon contexte devrait avoir une bonne qualité d'informations mais moins de redondance. L'évaluation de l'information est représentée dans le tableau 2, implique le calcul de deux paramètres : la Di-similarité entre un résumé de référence et le résumé automatique proposé pour les unigrammes et les bi-grammes.Comme l'illustre le tableau suivant

	Unigrams	Bigrams
Sujet 1		
approche proposée	**0,8333**	**0,8333**
Sujet 2		
approche proposée	**0,95**	**0,95**
Sujet 3		
approche proposée	**0,8889**	**0,8889**

Tableau 2 **-** Tableau des résultats d'informativité

[12] https://en.wikipedia.org/wiki/ActivePerl

Notez que la Di-similarité étant une mesure de distance qui implique qu'une valeur inférieure à cette métrique est indicative d'un meilleur résultat.

Nous avons conclu que l'utilisation des hashtags améliore considérablement la clarté de contexte. Nous présentons dans le tableau suivant les résultats de notre contexte produit en termes de lisibilité :

	Pertinence (%)	Non redondance (%)	Moyenne(%)
Sujet 1			
Résumé proposé	90,00	90,00	90,00
Sujet 2			
Résumé proposé	70,00	90,00	80,00
Sujet3			
Résumé proposé	90,00	70,00	80,00

Tableau 3- Tableau des résultats de lisibilité

IV.4 Conclusion

Dans ce chapitre nous avons présenté le cadre expérimental de nos travaux. Ainsi que les étapes de traitements par lesquelles passe l'approche. Puis nous avons réalisé des expérimentations sur notre approche et analysé par la suite ces résultats qui sont des résumés générés automatiquement.

Enfin, nous les avons comparés aux résumés humains. Concluant donc que nos résultats ont apportés des améliorations en termes de clarté du contexte du tweet initial.

Conclusion générale

Les microblogs ont connu récemment un engouement extraordinaire par la communauté de la recherche d'information, du fait de la popularité des plateformes de microblogging. Twitter, étant la plateforme qui a connu le plus de croissance et qui a suscité l'intérêt de plusieurs chercheurs.

Nous nous somme intéressées dans ce mémoire à proposer des solutions pour la contextualisation des tweets, en donnant une vue d'ensemble des différentes approches qui ont été proposées. Plus explicitement nous avons proposés dans ce mémoire deux approches :

1) Une approche permettant de générer un bref résumé explicatif (500 mots dans INEX). Ce dernier devrait être construit automatiquement en utilisant des ressources externes dans le but d'extraire les passages pertinents et en les regroupant en un résumé cohérent.
2) Une approche permettant de prendre en considération les différents types des signaux (sociaux, temporels et textuels).
3) Une approche permettant le découpage des hashtags en appliquant les N_grams de lettres.

Notre objectif était de proposer une approche qui améliore la contextualisation des tweets.

Pour y arriver, nous avons exploité les propriétés présentes dans les tweets en se basant sur la pertinence thématique du tweet et ses propriétés sociales.

Cependant, notre approche présente certaines limites, parmi lesquelles, nous citons :

- La collection sur laquelle nous avons testé notre approche n'est pas assez riche, ce qui n'a pas donné des résultats très clairs à interpréter,
- Utilisation d'une seule source d'information (Twitter),
- L'interface que nous avons rajoutée est simple et limitée, car elle ne peut pas contextualiser n'importe quel tweet de Twitter,
- Utilisation d'un dictionnaire anglais moins riche dans les N_grams de lettres,

De ce fait, nous envisageons comme perspectives d'amélioration de notre travail, les actions suivantes :

- Plusieurs sources d'informations comme la combinaison entre Twitter et Wikipédia,

- nous envisageons de mener nos expérimentations sur une collection tweets de plus grande taille,
- Nous souhaitons que notre interface fasse l'objet des prochaines études d'amélioration par les futurs étudiants,
- Utilisation d'un dictionnaire anglais mieux enrichi pour faciliter le découpage des hashtags.

Pour terminer, nous espérons que le travail que nous avons réalisé puisse être un outil facilitant la contextualisation des tweets, et que notre mémoire soit un guide pour les futurs étudiants.

BIBLIOGRAPHIE

[1]ABBAS,Nacira. « informatique ingénieurie des système de recherche d'information », Mémoire de magister.

[2]HAMMACHE, Arzeki. «recherche d'information» Thèse de doctorat en informatique.

[3] AMIROUCHE , Fatiha. « Modèles de recherche », cours UMMTO.

[4]Friburger.« Métriques et TAL ».

[6]Damak, Firas. « Etude des facteurs de pertinence dans la recherches de microblogs »,Thèse de doctorat.

[7] MNASRI, Maâli. « Résumé Automatique Multi-Document Dynamique : État de l'Art ».

[8] ROMAIN, Deveaud et BOUDIN, Florian . «De quoi parle ce Tweet ? Resumer Wikipedia pour contextualiser des microblogs.» s.d.

[9] BELKRAOUI, Rami et FAIZ, Rim.«Conversational based method for tweet contextualization» .

[10] BELKRAOUI,Rami; FAIZ, Rim et Pascale, Kuntz.« User-Tweet Interaction Model and Social Users Interactions for Tweet Contextualization » .

ANNEXE

LexRank

Est un algorithme de classement le plus populaire, conçu à l'origine pour déterminer l'importance d'unepage Web représenté par le sommet d'un graphe et le lien par l'arc.

TextRank

TextRank est une variante de l'algorithme LexRank pour le résumé automatique mono document fondé sur les graphes, dans laquelle les valeurs de similarités attribuées aux arcs sont utilisées pour l'équilibrage des sommets. De cette manière l'impact des sommets connectés par des arcs de valeurs faibles sera minimisé dans le calcul du score. Le score de chaque sommet s est calculé itérativement jusqu'à la convergence par la formule suivante :

$$P(s) = (1 - d) + d \times \sum_{V \in adj[S]} \frac{Sim(S,V)}{\sum_{V \in adj[V]} Sim(Z,V)} P(v) \quad (50)$$

Où P(v) est le nombre d'arrêtes du sommet v et d est un facteur d'amortissement généralement fixé à 0,85.

Le théorème de Bayes

Le théorème de Bayes est un résultat de base en théorie des probabilités, issu des travaux du « Révérend Thomas Bayes ».

L'utilisation de la règle de Bayes nous amène donc à calculer la probabilité d'observer une certaine classe c pour un tweet t définis comme suit :

$$P(C|t) = \frac{p(C|t)P(C)}{P(t)} \quad (51)$$

Où C est la classe la plus probable pour le tweet t.

Lissage de Dirichlet

Le lissage de Dirichlet permet de lisser les documents en tenant comptede leur taille. Il donne de très bons résultats sans nécessiter de calculs complexes. Ce type delissage est utilisé par des modèles de langue. Il permet l'augmentation des fréquences des n-grammesmi dans un document D**.**

La probabilité PDIR (mi/D) d'un mot selon le modèle de langue du document est la suivante :

$$\text{PDIR (mi|D)} = \frac{t_f(mi,D)+\mu pml(mi|C)}{|D|+\mu} \quad (52)$$

Où |D| est la taille du document (le nombre total d'occurrences de mots), et t_f (mi, D) est lafréquence du mot mi dans D.

L'objectif théorique de ce lissage est de mieux estimer les distributions de probabilitéstirées des documents seuls, et au niveau pratique d'éliminer le problème des probabilités nullesdans le cas de correspondances partielles entre documents et requêtes.

API Twitter

Une API (Application Programmable Interface en anglais, « interface de Programmation » est une série de méthodes permettant d'accéder aux services d'une application, par l'intermédiaire d'un langage de programmation.Elle permet de fournir un certain niveau d'abstraction au développeur, c'est-à-dire qu'elle lui permet d'utiliser certaines fonctionnalités ou d'accéder à des données du site.

Définitions de termes relatifs à l'API Twitter

Dans ce qui suit, nous allons voir quelques définitions de termes liés à l'API Twitter.

Utilisateur(User) :C'est l'entité qui publie des tweets, suit d'autres utilisateurs(Follow),ils peuvent être mentionnés ou eux même suivis par d'autres utilisateurs,ils ont un ensemble de propriétés qui leur est associés. Parmi ces propriétés on peut citer :

- username: c'est l'alias de l'utilisateur, il est unique pour chaque utilisateur, mais peut être sujet à modification.

- followcount: représente le nombre de Followers que l'utilisateur a (ses abonnés).
- Mentioncount : représente le nombre d'utilisateurs à motionnés.

Un tweet a un ensemble de propriétés qui lui sont associées, parmi celles qu'on a utilisées :

- IdTweet: est un entier qui représente l'identifiant unique d'un tweet, stocké sur 64 bits.
- Text : c'est le contenu textuel du tweet (encodé UTF-8).
- createdAt: réfère à la date de création du tweet.
- Retweet_count : représente le nombre de fois qu'un tweet a été partagé pard'autres utilisateurs.
- Replycount : représente le nombre de réponse qu'un tweet aura lieu.
- Favouritecount : représente le nombrede tweets préférées.

Apache OpenNLP

La librairie Apache OpenNLP est une boite à outils pour le traitement automatique detextes en langage naturel, basé sur l'apprentissage machine. Elle fournit aussi un grandnombre de modèles prédéfinis pour une variété de langues, ainsi que les ressources de texteannotées.

Les outils proposés par Apache OpenNLP :

- La tokenisation.
- Détection de phrases.
- Etiquetage morpho-syntaxique (aussi appelé étiquetage grammatical, POS tagging(part-of-speech tagging) en anglais).
- Extraction des entités nommées.
- Catégorisation de documents.
- Support du classifieur Entropie Maximum pour l'apprentissage machine.