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

Université Saad Dahlab Blida

Faculté des sciences

Département d'informatique

Mémoire présenté par : Gousmi Abdelkader

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique
Filière : Informatique
Spécialité : Informatique
Option : Ingénierie de logiciel

Sujet :

Proposition d'un modèle de recherche d'informations sociales à base de machine Learning avec expansion des requêtes de recherche

Encadré par : M. BOUCETTA Zouhel

Promotion
2017 / 2018

Dédicaces

A mes chers parents pour leur soutien moral, encouragements et sacrifices

A ma chère sœur et mon frère qui m'ont soutenu jusqu'au bout

A mes chers amis qui étaient la quand j'en avais le plus besoin

A toute personne qui a contribué de près ou de loin pour l'aboutissement de ce projet

Au bonheur des plus chers

Je dédie cet humble travail

GOUSMI Abdelkader

Remerciements

On tient à remercier dieu tout puissant de nous avoir permis de mener à bien notre mission

On remercie également notre promotrice M Boucetta pour
Sa patience et ses bonnes explications qui nous ont éclairé le chemin de la recherche et sa
Collaboration avec nous dans l'accomplissement de ce modeste travail
Nous tenons à exprimer nos sincères remerciements à tous ceux qui, de près ou de loin, ont
Contribué à la réalisation de ce travail

Pour conclure, Nous adressons nos remerciements les plus respectueux au jury qui ont
Accepté d'évaluer notre travail.

Résumé

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

Les résultats que nous avons obtenus étaient plutôt satisfaisant, l'expérience était très enrichissante et les acquis théoriques et pratiques nous motive à explorer cet univers passionnant encore plus.

Mots clés :

Twitter, microblogging, expansion requête, modèle de RI, le reclassement des tweets.Word2Vec, Machine Learning, IA

[1] -soc http derateur.com/chiffres-réseaux iaux/

Abstract

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

The results we obtained were rather satisfactory; the experience was very rewarding and the theoretical and practical achievements as much on the Electronic side as Informatics motivate us to explore this exciting universe even more.

Keywords:

Twitter, microblogging, Query expansion, IR model, Reranking of tweets.Word2Vec, Machine Learning, IA

ملخص

في الوقت الحاضر ، تعد منصات التدوين المصغر أحدث الشبكات الاجتماعية الأكثر انتشارًا في Web 2.0. لديهم كتلة ضخمة من المعلومات. تويتر هي خدمة المدونات الصغيرة الأكثر شعبية مع 320 مليون مستخدم نشط في الشهر أكثر من 500 مليون تغريدة ترسل يوميا. يعقد هذا العدد من المنشورات عملية الوصول إلى المعلومات من قِبل أصحاب المدونات الصغيرة. إن التغريدة عبارة عن مستند قصير لا يتجاوز طوله 140 حرفًا. غالبًا ما يتم كتابتها بلغة غير صحيحة تحتوي على اختصارات وصفات عامة لتدوين المعلومات باستخدام أقل عدد ممكن من الأحرف. يمثل العثور على معلومات في tweets corpus تحديًا حقيقيًا لنماذج البحث عن المعلومات الحالية ، ويرجع ذلك إلى حجم المجلد من ناحية وخصائص التغريدات من جهة أخرى. في الواقع ، عندما يقدم المستخدم طلبًا ، سيواجه نموذج البحث مشكلتين: أولاً عدم وجود شروط الاستعلام في التغريدة ، وحقيقة أن كل مصطلح يظهر على الأكثر مرة واحدة في النص. يعتمد اختيار أفضل التغريدات على الاقتران المفاهيمي بين الاستعلام والتغريدات. نتيجة لذلك ، هناك احتمال كبير بأن تحتوي أعلى القائمة على تغريدات غير ذات صلة. لتحسين ترتيب التغريدات ذات الصلة ، قامت العديد من الأعمال بعرض الأدلة المؤقتة في مقترحاتها من خلال دمجها مع الأدلة المعجمية لإعادة تصنيف نتائج التغريدات الخاصة بالبحث الأول ، وقد اقترحنا نموذجًا للبحث عن المعلومات استنادًا إلى آلة التعلم لترتيب التغريدات ، كان الهدف من المشروع هو توسيع طلب البحث للحصول على نتائج أفضل.

كانت النتائج التي حصلنا عليها مرضية للغاية، وكانت التجربة مثرية للغاية، وتحفزنا الإنجازات النظرية والعملية على استكشاف هذا العالم المثير أكثر.

كلمات مفتاحية:

تويتر ، المدونات الصغيرة ، توسيع الاستعلام ، نموذج Word2Vec،IR. إعادة التصنيف tweets ، تعلم الآلة ، ذكاء إصطناعي

Table des matières

Liste des tableaux

Introduction générale

Contexte

De nos jours, plus de la moitié de la population mondiale utilise internet, en avril 2017 3.81 milliards (soit 51% de la population) dont 2.91 milliards (soit 39%) inscrits dans les réseaux sociaux[1] alors que ce pourcentage avoisinait les 15% en 2005, et ceci en grande partie grâce à la facilité d'accès à internet partout dans le monde et avec de plus en plus d'appareils (Ordinateurs, téléphones, smart TV…etc.)

Les différents utilisateurs de ces réseaux sociaux ne sont pas limités par la consommation d'informations seulement mais contribuent à leur production, ce qui représente un nombre impressionnant d'informations traités chaque jour ; La recherche d'information ou RI est un des domaines qui s'intéresse principalement à structurer, analyser, organiser, rechercher et classer ces informations, cependant la masse volumineuse d'informations disponible constitue un réel challenge pour pouvoir répondre au différents besoin des utilisateurs et donner des résultats pertinents.

Nous nous somme principalement intéressés à la recherche d'information dans les plateformes de microblogging et plus précisément Twitter, qui est un des réseaux sociaux les plus récents et une des plateformes les plus utilisées actuellement,

Twitter est le service de microblogging le plus populaire avec 320 millions d'utilisateurs actif par mois et plus de 500 millions de tweets envoyés par jour. Ce volume de publications complique l'opération d'accès à l'information par les Microblogueurs.

Le Tweet est un document court dont la longueur ne dépasse pas 140 caractères. Ce nombre assez bas de caractères poussent les utilisateurs à utiliser un langage assez mal orthographié et contenant souvent des abréviations pour transmettre l'information tout en respectant le nombre maximum de caractères

En raison du volume important du corpus des tweets et leur format particulier, la recherche d'information dans Twitter représente un réel défi.

Problématique

Une grande partie des anciens modèles de recherche d'informations font face à plusieurs problèmes ; Le premier étant le volume très important du corpus des tweets, ce qui rend la tâche d'obtenir des résultats de recherche pertinent de plus en plus difficile ;

[1] D'après l'union internationales des télécommunications « http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2016.pdf »

Le deuxième étant la taille assez réduite des tweets ce qui implique que moins de terme seront disponible pour assurer la précision de la recherche; Le troisième est la qualité du texte des tweets, qui est compromise à cause des différentes abréviations ainsi que l'orthographe utilisée dans ces derniers ; Finalement les tweets peuvent contenir une syntaxe spécifique telle que les #hashtags, les @citations ou bien encore des URL. Les plateformes de microblogging représentent également un modèle de réseau social différent des autres réseaux sociaux. Les relations entre les utilisateurs ne sont pas forcément réciproques et les abonnements sont sans restriction entre microblogueurs.

Afin d'améliorer les résultats de recherche des tweets pertinents, plusieurs modèles de recherche d'information ont été introduit, notre travail se positionne dans cette optique.

L'objectif

Les utilisateurs de plateformes de microblogging, outre la publication de microblogs, effectuent également des recherches. Les motivations de ces recherches sont diverses. Certaines sont similaires à la recherche sur le web (comme par exemple la recherche d'actualités), et d'autres sont spécifiques à la recherche de microblogs (comme par exemple la recherche temps réel ou d'informations sociales). Dans Twitter, 1,6 milliards de requêtes sont ainsi émises chaque jour.

Les modèles de RI doivent s'adapter aux spécificités des microblogs : fraîcheur, aspect social et spécificités syntaxiques doivent ainsi être pris en compte. C'est dans ce contexte de recherche d'information dans les microblogs que se situent plus particulièrement nos travaux. Nous nous plaçons plus précisément dans le cadre de la recherche d'information ad hoc tout en introduisant l'apprentissage machine.

L'objectif est de retrouver les microblogs répondant à un besoin d'information spécifié par un utilisateur par l'introduction des aspects sociaux dans un modèle d'apprentissage.

Notre travaille vise d'améliorer la qualité des résultats de recherche d'information adhoc dans les microblogs, en effectuant une expansion des requêtes de recherche afin de maximiser les résultats de ces dernières.

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

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

Organisation du mémoire

Nous avons organisé ce mémoire en Cinque chapitres :

Chapitre 1 :

Dans ce premier chapitre on aborde les concepts de base de la recherche d'information RI, en commençant par définir la RI, comment modéliser un processus de RI ainsi que ses différents modèles.

Nous verrons ensuite ce qu'est un système de recherche d'information et sa définition ainsi que les différentes étapes d'un processus RI c'est-à-dire : Indexation, Requettage, Appariement ;

Et finalement, nous verrons quelques moteurs de recherche OPEN SOURCE (ou accès libre)

Chapitre 2 :

Dans ce deuxième chapitre, nous allons spécifier la recherche d'informations dans les microblogs, en commençant par présenter la plateforme de microblogging choisie «Twitter » tout en détaillant ses différentes caractéristiques. Par la suite nous présentons quelques aspects de la recherche d'information sémantique et de la recherche d'informations temporelles dans les tweets ; Et finalement nous verrons les mesures d'évaluation des systèmes de recherche dans les tweets.

Chapitre 3 :

Dans ce troisième chapitre, nous allons aborder la notion d'intelligence artificielle et ses différent domaines de recherche, nous verrons par la suite ce qu'est l'apprentissage machine (Machine Learning) ainsi que les différents types d'apprentissage, et finalement nous aborderons la notion de Word Embedding et une de ces méthodes qui est Word2Vec ainsi que son utilisation dans notre travail.

Chapitre 4 :

Dans ce quatrième chapitre, nous allons détailler notre approche pour la conception d'un modèle de recherche d'informations à base de Machine Learning sur Twitter, dont l'objectif principal est d'améliorer les performances de recherche dans le corpus des tweets.

Nous travail se divise quatre phases :

Acquisition des données : cette phase consiste à acquérir l'ensemble des tweets constituant le corpus Trec 2011.

Prétraitement : cette phase consiste à traiter les tweets de manière à ce qu'il ne reste que le texte dont on a besoin pour notre recherche, on détaillera donc les différent processus utilisés.

Traitement : cette phase consiste à effectuer une première recherche sur le corpus des tweets traités, et utiliser le résultat afin d'avoir les données nécessaires pour faire une expansion des requêtes et affiner la recherche.

Recherche : C'est la phase finale ou on utilise les nouvelles requêtes traitées pour effectuer la recherche.

Chapitre 5 :

On détaillera dans ce cinquième chapitre l'environnement de travail (hardware et software), ainsi que les différents outils d'implémentation utilisés dans ce projet.

Chapitre 1

Recherche d'information

1. Introduction :

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

La recherche d'Informations (RI) peut être définie comme une activité dont la finalité est de localiser et de délivrer un ensemble de documents à un utilisateur en fonction de son besoin en informations. Le défi est de pouvoir, parmi le volume important des documents disponibles, trouver ceux qui correspondent au mieux à l'attente de l'utilisateur. L'opérationnalisation de la RI est réalisée par des outils informatiques appelés Systèmes de recherche d'Informations, ces systèmes ont pour but de mettre en correspondance une représentation du besoin de l'utilisateur (requête) avec une représentation du contenu des documents (index) au moyen d'une fonction de comparaison (ou de correspondance). (Amini, 2013)

2. Définition :

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

2.1 Composante d'une recherche d'information :

Toute recherche d'information RI, est composée de trois éléments :

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

ensemble de mots clés, mais elle peut être exprimée en langage naturel, booléen ou graphique. (BOURAMOUL, 2011)

2) **Modèle de représentation** : un modèle de représentation est un processus permettant d'extraire d'un document ou d'une requête, une représentation paramétrée qui couvre au mieux son contenu sémantique. Ce processus de conversion est appelé indexation. Le résultat de l'indexation constitue le descripteur du document ou de la requête, qui est une liste de termes ou groupes de termes (concepts), significatifs pour l'unité textuelle correspondante, auxquels sont associés généralement des poids, pour différencier leurs degrés de représentativité du contenu sémantique de l'unité en question. L'ensemble des termes reconnus par le SRI est rangé dans une structure appelée dictionnaire constituant le langage d'indexation. Ce type de langage garantit le rappel de documents lorsque la requête utilise dans une large mesure les termes du dictionnaire. En revanche, il y a risque important de perte d'informations lorsque la requête s'éloigne de ce vocabulaire. (BOURAMOUL, 2011)
3) **Modèle de recherche** *:* il représente le modèle du noyau d'un SRI. Il comprend la fonction de décision fondamentale qui permet d'associer à une requête, l'ensemble des documents pertinents à restituer. Il est utilisé pour la recherche d'information proprement dite et est étroitement lié au modèle de représentation des documents et des requêtes. (BOURAMOUL, 2011)

2.2 Modèles de recherche d'information:

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

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

– **D** est l'ensemble de documents

– **Q** est l'ensemble de requêtes

– **F** est le schéma du modèle théorique de représentation des documents et des requêtes

– **R(q,d)** est la fonction de pertinence du document **d** à la requête **q**

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

a. Modèle booléen :

Le modèle booléen (Salton, 1970)est basé sur la théorie des ensembles. Dans ce modèle, les documents et les requêtes sont représentés par des ensembles de mots clés. Chaque document est représenté par une conjonction logique des termes non pondérés qui constitue l'index du document. Un exemple de représentation d'un document est comme suit :

$$d = t1 \wedge t2 \wedge t3 ... \wedge tn$$

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

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

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

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

b. Modèle vectoriel :

Dans ces modèles (Salton, 1970)la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance dans un espace vectoriel. Le modèle vectoriel représente les documents et les requêtes par des vecteurs d'un espace à **n** dimensions, les dimensions étant constituées par les termes du vocabulaire d'indexation. L'index d'un document **dj** est

le $\vec{vecteur} = (w1,j,w2,j,w3,j,..,wn,j)$où **wk ,j ∈ [0 1]** dénote le poids du terme **tk** dans le document **dj** . Une requête est également représentée par

$\vec{vecteur} = (w1,q,w2,q,w3,q,..,wn,q)$ où **wk,q** est le poids du terme **tk** dans la requête **q**.

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

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

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

c. Modèle probabiliste :

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3. Système recherche d'information :

3.1 Définition :

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

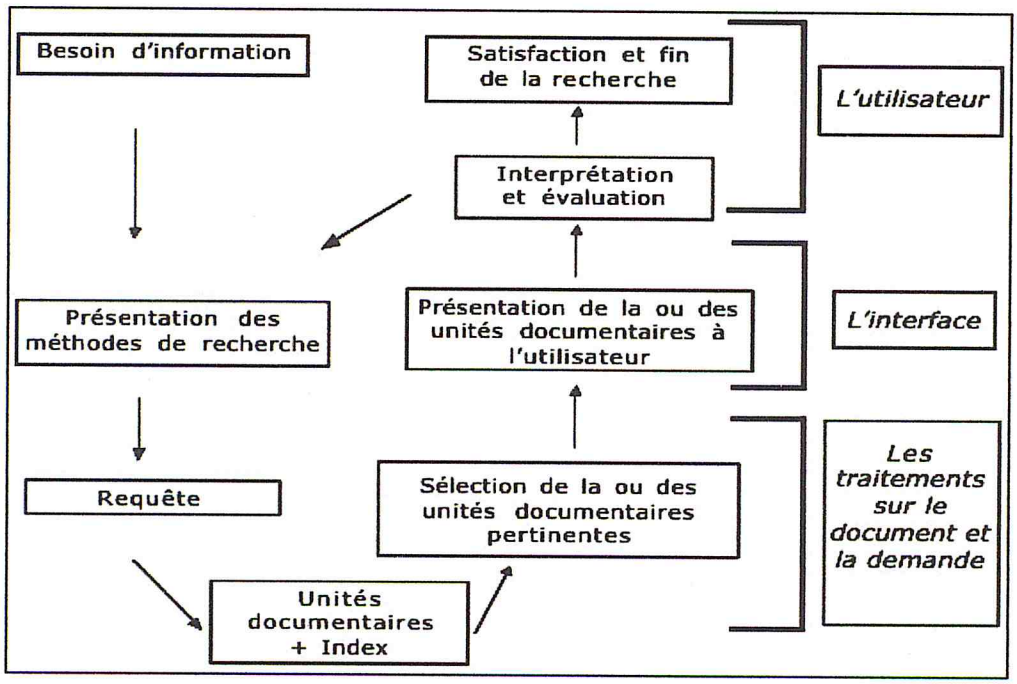


Figure 1 : Système de Recherche d'Information.

3.2 Processus de recherche d'information :

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

utilisateur en requête repose sur un modèle de requête. Enfin, la correspondance entre une requête et des documents s'établit par une relation de pertinence [Maisonnasse,08].

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

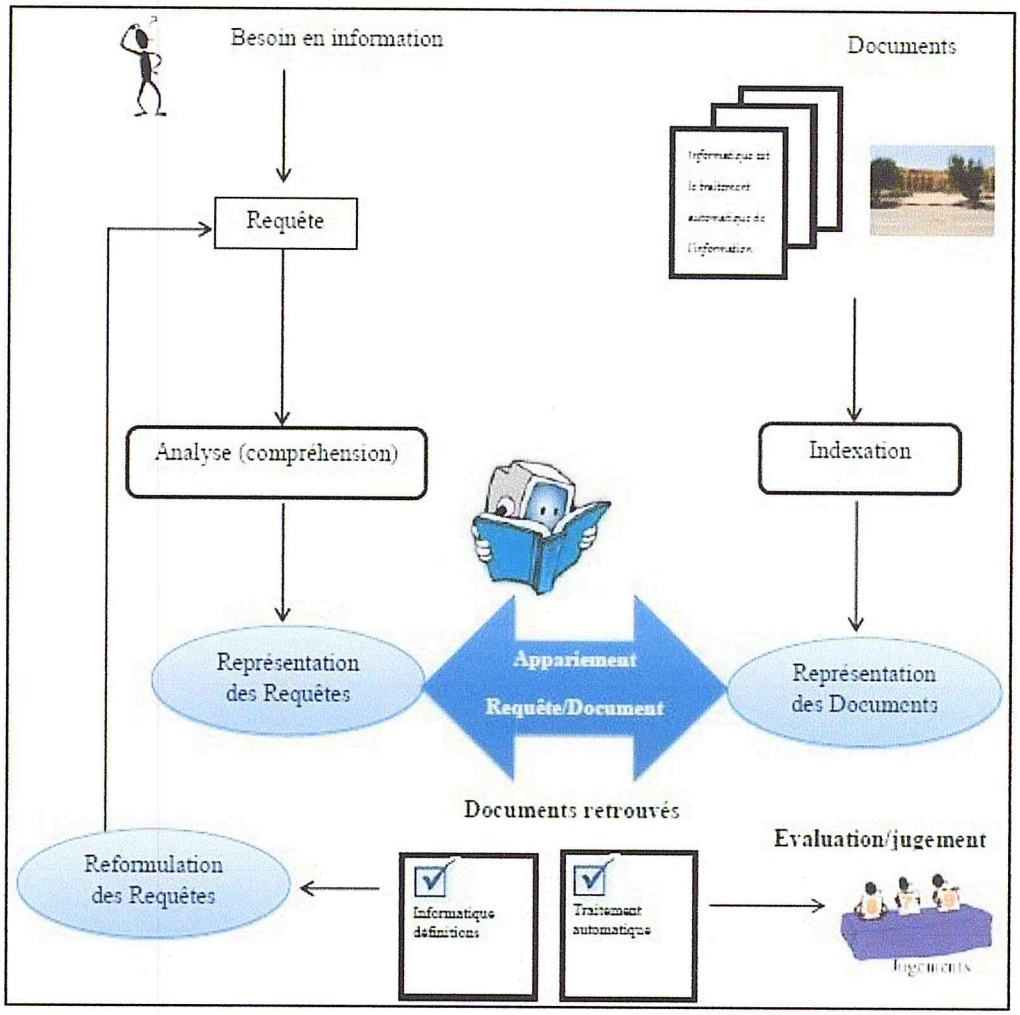


Figure 2 : *Processus de recherche d'information*

3.2.1. L'indexation :

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

3.2.2. Le requêtage :

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

3.2.3. L'appariement :

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

4. Mesure de similarité :

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

- **Fréquence des termes (TF) :**

Cette mesure est proportionnelle au nombre d'occurrences d'un terme dans un document (pondération locale). Toutefois, il existe différentes variantes de cette mesure qui dépendent de la façon dont la pertinence est mesurée. L'inconvénient du TF se situe au niveau de la pertinence globale. Certains termes sont plus significatifs que d'autres, bien qu'apparaissant avec la même fréquence dans un document. Pour cette raison le TF est souvent couplé avec la mesure IDF. (Damak, 2014)

- **Fréquence de documents inverse (IDF) :**

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

Il se calcule selon la formule suivante :

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

N : est le nombre de documents dans la collection et

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

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

La combinaison de TF et IDF fournit une autre mesure importante

$$TFIDF_{t,d} = TF_{t,d} * IDF_t$$

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

5. Moteur de recherche :

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

5.1. Moteur de recherche Lucene :

Lucene[1] est une bibliothèque open source écrite en Java qui permet d'indexer et de chercher du texte. Il est utilisé dans certains moteurs de recherche. C'est un projet de la fondation Apache mis à disposition sous licence Apache. Il est également disponible pour les langages Ruby, Perl, C++, PHP, C#.

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

5.2. Moteur de recherche Terrier :

Terrier[2] est une plate-forme dédiée à la recherche d'information. Elle implémente les différents modules intervenant dans le processus de RI classique et offre en plus un cadre pour

[1] http://lucene.apache.org/
[2] http://terrier.org/

l'évaluation des résultats de recherche pour différentes applications (Ounis, 2006)]. Terrier a été largement éprouvée (Middleton, 2007).Le choix de cette plate-forme est dû aussi à sa capacité à traiter de grandes collections de documents telles que les collections TREC.

L'architecture de la plate-forme Terrier distingue les deux phases classiques : l'indexation et la recherche. Un corpus documentaire est fourni en entrée au module d'indexation. Les documents de la collection passent par un ensemble de prétraitements tels que la tokenisation. Les tokens sont ensuite injectés dans une chaîne de traitement TermPipLines, à savoir le StopWords PipeLine pour l'élimination des mots vides de sens, ou encore les Stemming pipeline et qui dépendent de la langue en question. La phase d'indexation conduit à la construction de l'index (Data structures).

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

5.3 . Moteur de recherche INDRI :

INDRI[1] est un module qui fait partie du projet LEMUR mené par un laboratoire d'université par the university of Massachusetts et the school of computer science at Carnegie Mellon University. Cette application est existante depuis 2004 et poursuit aujourd'hui sont évolution. C'est une solution totalement libre et non commerciale. « La solution de Indri sépare le stockage des données de l'indexation fulltext, ce qui pourrait permettre des modifications et évolutions sans devoir tout changer ».

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

En résumé, INDRI est une solution libre, rapide, fiable, évolutive, qui peut tout à fait être utilisée pour l'indexation et le traitement de requêtes de très nombreux type de document

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

notamment les XML. Il ne manque qu'une interface utilisateur adaptée au grand publique pour que cette solution devienne la référence.

6. Conclusion :

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

Chapitre 2

Recherche d'information sociale « Cas Twitter »

1. Introduction :

Les microblogs sont une forme réduite des blogs. Ils représentent une source d'information récente. Les utilisateurs emploient des plates-formes de microblogging pour partager et accéder à des microblogs. Ces plateformes prennent la forme de réseaux sociaux qui se distingue par des interactions sociales intenses et une diversité dans les sujets discutés, par rapport aux autres sources d'information. Il existe plusieurs plateformes de microblogging. Les plateformes les plus utilisées sont Twitter, Friend Feed, Tumblr, Posterous. Parmi elles, Twitter est sans conteste la plus utilisée. Cette plate-forme compte plus de 650 millions d'utilisateurs, publiant en moyenne 58 millions de tweets par jour. Twitter est utilisée également comme source d'information. En moyenne, 2,1 milliards de requêtes sont soumises chaque jour sur son moteur de recherche. La RI dans les microblogs est différente de la recherche dans le Web. Ceci est dû aux différences de forme des microblogs par rapport aux documents du web, à la spécificité de leur contenu court et mal orthographie et également aux motivations des recherches (informations fraîches.). cela a conduit à un véritable défi selon (Yoo, 2014) pour les modèles de recherche d'informations actuelles.

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

2. Présentation de la plate-forme de microblogging Twitter :

2.1. Historique de Twitter :

Twitter a été créé en 21 mars 2006 à San Francisco au sein de la société américaine startup Odeo fondée par Noah Glass et Evan Williams, et Jack Dorsey. Cette société proposait une plateforme d'hébergement, de diffusion et d'enregistrement de podcast. L'idée de départ lancée par Jack Dorsey était de permettre aux utilisateurs de partager facilement leurs petits moments de vie avec leurs amis. Le 21 mars 2006, M. Dorsey envoyait son premier tweet : « Just setting up my twttr » (« Suis en train d'installer mon twttr »). Le marché du podcast étant déjà très concurrentiel, Jack Dorsey et Noah Glass et Evan Williams furent chargés de développer un nouveau service ouvert au public le 13 juillet 2006, la première version s'intitulait Stat.us puis Twittr, en référence au site de partage de photos Flickr puis Twitter, son nom actuel. Le 25 octobre 2006, les actifs de la société Odeo ont été rachetés par Obvious Corp. Puis en avril 2007, une entité indépendante est créée comme nom Twitter avec Jack Dorsey à sa tête jusqu'en octobre 2008 date à laquelle Evan Williams lui succéda. En

mars 2008, Twitter compte un million d'utilisateurs. La société compte 29 employés en février 2009, 300 en octobre 2010 et 900 en avril 2012. En juin 2012, les mots « Twitter » (nom propre), « Twitt » ou « tweet », « Twitteur » ou « Twitteuse », ainsi que « Twitter » ou « Tweeter », font leur apparition dans Le Petit Larousse édition 2013.Twitter dont le prix d'introduction est fixé à 26 dollars entre à la bourse de New York le 31 octobre 2013 sous le symbole « TWTR » avec une première cotation qui s'effectue à 45,10 dollars. L'action atteindra un pic à 73,31 dollars en décembre 2013 avant d'amorcer une chute jusqu'à 31,85 dollars à la fin du lock-up (période durant laquelle un actionnaire ou un investisseur ne peut se défaire de ses actions) le 6 mai 2014. Dick Costolo démissionne de son poste de PDG de Twitter en juin 2015, sur fond de désaveu de sa stratégie. Il est remplacé de façon intérimaire par l'un de ses fondateurs, Jack Dorsey.[1] [2]

2.2 . Présentation générale de Twitter :

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

2.3. Les Followers :

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

[1] URL: http://www.numerama.com/startup/twitter

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

Figure 3 : *Capture d'écran de la page personnelle d'ensemble* [Twitter(1)].

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

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

2.4. Lexique de Twitter :

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

***Figure 4** : Capture d'écran de la page profile de l'utilisateur de Twitter*

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

***Figure 5** : Capture d'écran d'un exemple de tweet.*

- **J'aime** : Cliquer sur J'aime est un moyen simple de montrer que vous appréciez un Tweet. Vous pouvez par ailleurs utiliser cette fonctionnalité pour

facilement retrouver ce Tweet plus tard. Cliquez sur l'icône en forme de cœur pour aimer un Tweet ; l'auteur verra ainsi que vous l'appréciez.

- **Following / Abonnements** : correspondent aux nombre des comptes Twitter que vous suivez. Pour connaitre le nombre d'abonnements, allez sur votre page d'accueil Twitter le nombre se trouve dans la colonne de droite tout en haut. Et pour voir tous vos following (personnes que vous suivez) cliquez sur le nombre ou « Abonnements».

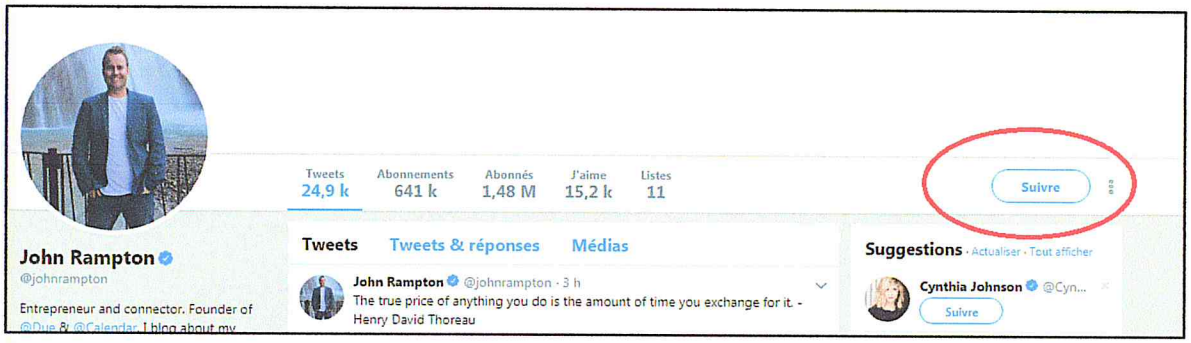


Figure 6 **:** *Capture d'écran d'un exemple d'abonnement.*

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

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

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

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

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

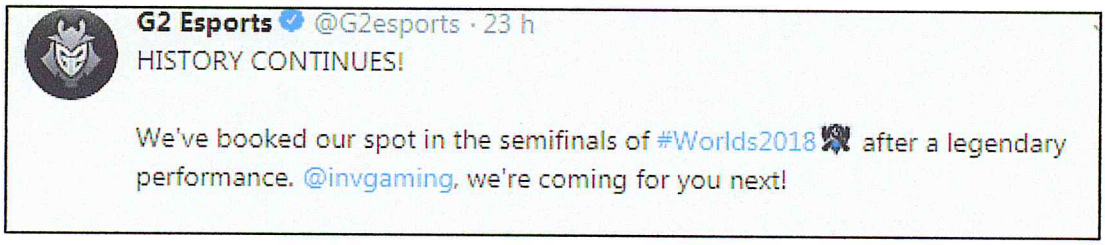


Figure 8 **:** *Capture d'écran d'un exemple de mention.*

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

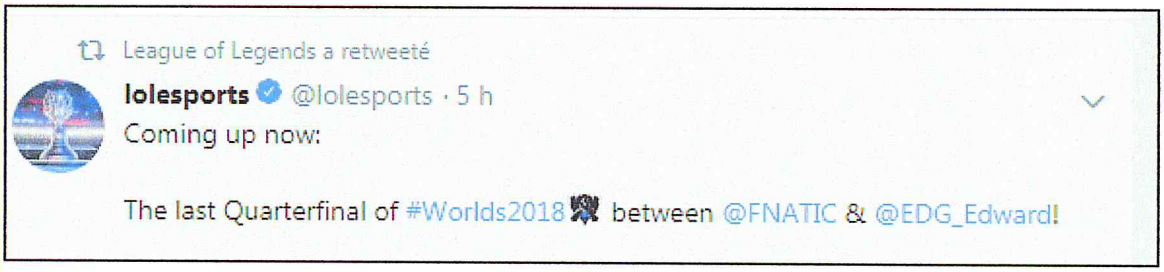


Figure 9: Capture d'écran d'un exemple de RT

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

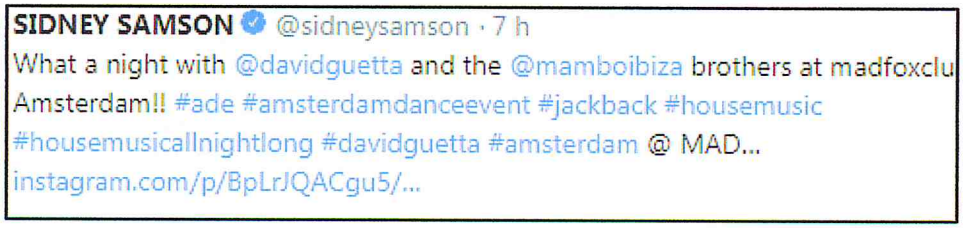


Figure 1 **:** *Capture d'écran d'un exemple d'un Hashtag.*

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

Figure 2 **:** *Capture d'écran d'un exemple des tendances.*

2.5 Type des tweets :

Il existe plusieurs types de tweets sont :

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

3. Recherche adhoc des microblogs

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

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

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

4. La recherche d'information temporelle :

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

5. Evaluation :

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

5.1. Les campagnes d'évaluation :

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

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

1) La campagne d'évaluation TREC est une série d'évaluations annuelles des technologies pour la recherche d'informations. Les participants sont en général des chercheurs pour de grandes compagnies commercialisant des systèmes et voulant les améliorer et des

groupes de recherche universitaires. Aujourd'hui le TREC est considéré comme le développement le plus important dans la recherche d'informations expérimentales, et demeure le plus cité et utilisé par la communauté de RI. Les pistes principales explorées sont le filtrage, la tâche adhoc et la tâche question-réponse.[Ben Jabeur,2013]

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

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

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

Figure 3 **:** *Exemple d'un topic pour la tâche Microblog de TREC2011.*

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

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

3) La campagne INEX : INEX[1] (INitiative for the Evaluation of XML Retrieval) est la seule campagne d''evaluation des différents SRI pour la recherche d'information sur les documents XML. Elle est mise en place chaque année depuis 2002. Elle offre un forum international non seulement pour permettre aux différentes organisations participantes

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

d'évaluer et de comparer leurs résultats, mais aussi pour discuter les différentes problématiques qui se présentent. La collection de test consiste en un ensemble de documents XML, requêtes, taches de recherche et jugements de pertinence.

5.2. mesures d'évaluation :

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

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

$$Précision = \frac{|Documents\ pertinents\ restitués|}{|Documents\ restitués|} \in [0,1]$$

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

$$Rappel = \frac{|Documents\ pertinents\ restitués|}{|Documents\ pertinents|} \in [0,1]$$

Le rappel et la précision sont calculés indépendamment de l'ordre dans lequel les résultats sont représentés (ce sont des mesures ensemblistes). Des mesures tenant compte de l'ordre des documents sont également nécessaires. Elles permettent par exemple d'évaluer des systèmes tels que les moteurs de recherche du web où l'ordre d'apparition des documents est crucial. À cet égard, les mesures principales proposées sont la **précision@X** et la **précision moyenne**.

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

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

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

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

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

$$\boldsymbol{MAP} = \frac{\sum_{q \in Q} \mathbf{AP_q}}{|\boldsymbol{Q}|}$$

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

6. Travaux voisins :

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

Titre	Résumé	Méthode	Auteur et l'année
Effectiveness of State-of-the-art Features for Microblog Search	Dans ce travail : -ils ont utilisé wordnet pour enrichir la requête sémantiquement avec les synonymes des termes.	-Technique de *Roucchio* -Mécanisme naturel de Pseudo-Pertinence Feedback (PRF) -Le modèle *BM25* - La mesure de précision et rappel - TF (Term Frequency) + IDF (Inverse Document Frequency).	**Firas Damak *2013*** (Damak, 2014)

Article: Use of Twitter and Semantic Resource Recovery in the Educational Context	Dans cet article : Ils ont développé un plugin contextuel qui intègre Twitter dans Moodle, Ce plugin effectue la recherche sémantique des tweets et des documents dans des dépôts externes en utilisant la requête fournie par l'utilisateur et un Contexte spécifié par Moodle.	Protocol OAI-PMH' (Open Archives Initiative – Protocol for Metadata Harvesting)	2012 IEEE 21st International WETICE (Wetice, 2012)
Combining Temporal and Content Aware Features for Microblog Retrieval	Dans cet article, ils ont proposé une méthode pour redéfinir le résultat de la recherche en fonction des caractéristiques temporelles, des fonctionnalités liées au compte et des fonctionnalités spécifiques au twitter, ainsi que des fonctionnalités textuelles des tweets. Ils ont également appliqué une technique d'expansion de requête en deux étapes pour améliorer la pertinence de sélection des tweets. Ils effectuent leurs expériences sur la collection TREC 2011	-Modèle de langue avec lissage Dirichlet -Modèle d'espace vectoriel -URL - Compte Retweet -Compte de statut	-Abu Nowshed Chy -Md Zia Ullah -Masaki Aon (Abu Noshed chy, 2015)
Combining Recency and Topic-Dependent Temporal Variation for Microblog Search	Ils ont proposé trois méthodes pour l'expansion temporelle de la requête. Deux méthodes individuelles basées sur la variation temporelle et la fraicheur (TVQE et TRQE)	-Fraicheur -Estimations de la densité du noyau(KDE)	-Taiki Miyanishi - Kazuhiro Seki -Kuniaki Uehara.

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

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

***Tableau 1** : Résumé des travaux voisin.*

7. Conclusion :

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

Chapitre 3

Machine Learning et Word Embedding

1. Introduction :

Durant ces dernières années, l'intelligence artificielle ou IA a connu une avancée d'une vitesse exponentielle, et ceci d'une part à cause de l'avancée technologique en matière de hardware, et donc en puissance de calcul et quantité d'informations traitées, et d'une autre part à cause des progrès en matière de recherche théoriques et applications pratiques, et donc une meilleure compréhension du domaine en général.

Le champ d'application de l'IA devient de plus en plus vaste, et touche plusieurs autres domaines (Médecine, Robotique, Finance, Jeux vidéo...Etc.),

Dans ce chapitre on va présenter brièvement ce qu'est l'intelligence artificielle et le machine learning, ainsi qu'un produit du ML qui est le **Word Embedding**, et comment on peut utiliser cette méthode dans la recherche d'information sociale.

2. Intelligence artificielle :

1.1. Définition :

L'intelligence artificielle est « L'Ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine. » (Larousse)

L'intelligence artificielle consiste à mettre en œuvre un certain nombre de techniques visant à permettre aux machines d'imiter une forme d'intelligence réelle. L'IA se retrouve implémentée dans un nombre grandissant de domaines d'application.

La notion voit le jour dans les années 1950 grâce au mathématicien Alan Turing. Dans son livre « Computing Machinery and Intelligence », ce dernier soulève la question d'apporter aux machines une forme d'intelligence. Il décrit alors un test aujourd'hui connu sous le nom « Test de Turing » dans lequel un sujet interagit à l'aveugle avec un autre humain, puis avec une machine programmée pour formuler des réponses sensées. Si le sujet n'est pas capable de faire la différence, alors la machine a réussi le test et, selon l'auteur, peut véritablement être considérée comme « intelligente ». (Futura)

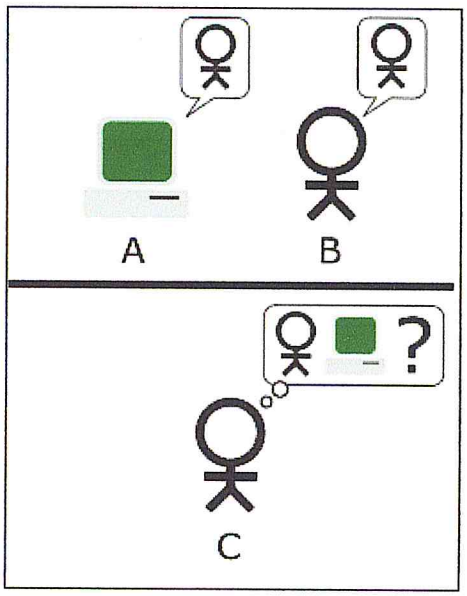


Figure 1 : Schéma simplifié d'un test de Turing

1.2. Domaines de recherche en IA :

L'IA s'étend sur plusieurs domaines de recherche, on en citera quelques-uns:

- Reconnaissance des formes :

  L'IA a permis d'automatiser la perception dans certaines situations, par exemple la reconnaissance de la parole et des traits du visage (système de repérage d'individus dangereux dans une foule), la lecture optique de documents et la synthèse d'images (détection automatique de langue et traduction)

- Robotique :

  L'IA permet aux robots d'avoir des capacités telles que la perception, pour leur permettre de sentir, se déplacer ainsi que raisonner pour trouver des solutions a certain problèmes

- Indexation multimédia :

  Le grand nombre de données disponible sur le Web rend la tâche de fouille et d'extraction de connaissances très dure, voir impossible dans certain cas, et c'est là qu'entre l'IA en jeu avec des outils permettant d'effectuer ces taches.

- Apprentissage machine (Machine Learning) :

  C'est un processus qui permet à une machine, d'effectue des taches qu'elle ne pouvait pas faire auparavant, ou d'effectuer certaines tâches qu'elle faisait déjà de manière plus efficace et rapide.

2. Apprentissage machine (Machine Learning) :

Nous nous intéresserons dans cette partie à l'apprentissage machine car c'est le champ d'étude de l'IA qu'on utilisera plus tard durant ce projet.

2.1. Définition :

« Unes des familles essentielles de techniques pour l'intelligence arfificielle, elle donne la capacité à un système à améliorer ses performances via des interactions avec son environnement, et ceci en concevant ou en améliorant le modèle ou le comportement de l'agent en question » (Fabien, 2011)

2.2. Principe :

Le principe du Machine Learning est d'utiliser des algorithmes qui vont permettre à un système piloté ou assisté par ordinateur d'adapter ses analyses et comportements en réponse à des données qu'il aura récoltées à travers une base de données ou des capteurs.

Le problème majeur dans ce cas réside dans le nombre de situation possibles (qui sont des entrées) qui va être de plus en plus grand, jusqu'à devenir impossible à traiter, et c'est pour cela qu'on délègue ce rôle à un programme qui devra simplifier la complexité. (Lecun, 2015)

2.3. Types d'apprentissage :

a. Apprentissage supervisé :

On parle d'apprentissage supervisé, si le système apprend à classer selon un modèle de classement donné, et donc avec des classes prédéterminées et des exemples déjà connus.

Ceci s'effectue en deux phases, lors de la première phase (hors ligne) le système détermine un modèle de données précis, et dans la deuxième(en ligne) le système essaye de classer la nouvelle donnée par rapport au système de données qu'il a appris dans la première phase.

b. Apprentissage non supervisé :

On parle d'apprentissage non supervisé lors ce que le système ne dispose pas d'un modèle de classement définit, et dispose donc seulement d'exemples, dans ce cas, l'algorithme devra découvrir tout seul comment structurer ces données, avec des méthodes comme le Data clustering qui regroupe les exemples en groupes similaires, pour les classer.

c. Apprentissage semi supervisé :

Ce type d'apprentissage est mis en œuvre quand le système dispose d'un modèle de classement de données, mais certaines données ne figurent pas dans ce modèle.

d. Apprentissage par renforcement :

En se basant sur une observation, l'algorithme effectue une action sur l'environnement qui produira une valeur de retour qui le guidera l'algorithme d'apprentissage. (Mitchell, 1997)

e. Apprentissage par transfert :

L'algorithme se base ici sur des actions antérieures, pour appliquer les compétences apprises, sur des nouvelles actions similaires. (Yang, 2010)

3. Word Embedding (plongement de mots)

3.1. Définition :

« Le word embedding (plongement de mot) est une méthode se focalisant sur l'apprentissage d'une représentation de mots. Cette technique permet de représenter chaque mot d'un dictionnaire par un vecteur de nombres réels correspondant. Ceci facilite notamment l'analyse sémantique des mots. Cette nouvelle représentation a ceci de particulier que les mots apparaissant dans des contextes similaires possèdent des vecteurs correspondants qui sont relativement proches. Par exemple, on pourrait s'attendre à ce que les mots « chien » et « chat » soient représentés par des vecteurs relativement peu distants dans l'espace vectoriel où sont définis ces vecteurs. » (Vukotic, 2015)

Les plongements de mots constituent notamment une méthode pour mitiger un problème récurrent en intelligence artificielle, soit celui du fléau de la dimension (curse of dimensionality). En effet, sans les plongements de mots, les objets mathématiques utilisés pour représenter les mots ont typiquement un grand nombre de dimensions, tant et si bien que ces objets se retrouvent isolés, et deviennent épars. La technique des Word Embeddings diminue le nombre de ces dimensions, facilitant ainsi les tâches d'apprentissage impliquant ces mots.

3.2. Principe :

Le principe du Word Embedding est de projeter un ensemble de mots dans un espace continu, ce qui va générer des vecteurs de mots (Dean, 2013), cette projection doit identifier et situer les vecteurs de mots similaires ou proches par rapport au sens selon le contexte où les mots ont été trouvés (corpus).

Exemple : Vecteur (Père) – Vecteur (Homme) + Vecteur (Femme) = Vecteur (Mère)

Voici un exemple ci-dessous d'une représentation d'un ensemble de mot en utilisant le Word Embedding :

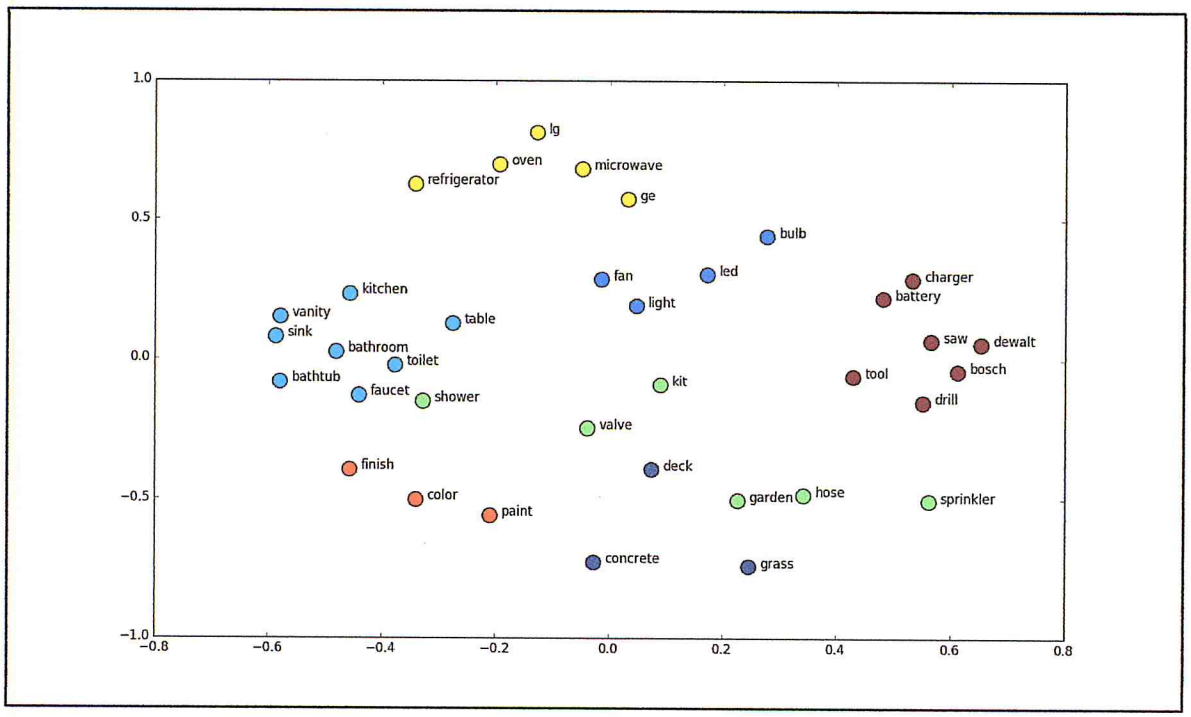


Figure 2 : Représentation d'un ensemble de mot avec le Word Embedding

3.3. Modèles de Word Embedding :

Il existe un bon nombre de modèle de Word embedding, on s'intéressera plus particulièrement à deux modèles : Skip-gram Model et Continuous Bag-of-Words Model (Greg Corrado, 2013)

- **Skip-gram :**

Ce modèle d'apprentissage, apprend en considérant un mot tiré d'une phrase donnée, c'est-à-dire il classifie les autres mots présents dans la phrase par rapport au mot tiré.

- **Continuous Bag-of-Words :**

Ce modèle est l'image inverse du Skip-gram, c'est-à-dire à partir d'un ensemble de mots, il peut prédire un mot à partir du contexte (les autres mots présents)

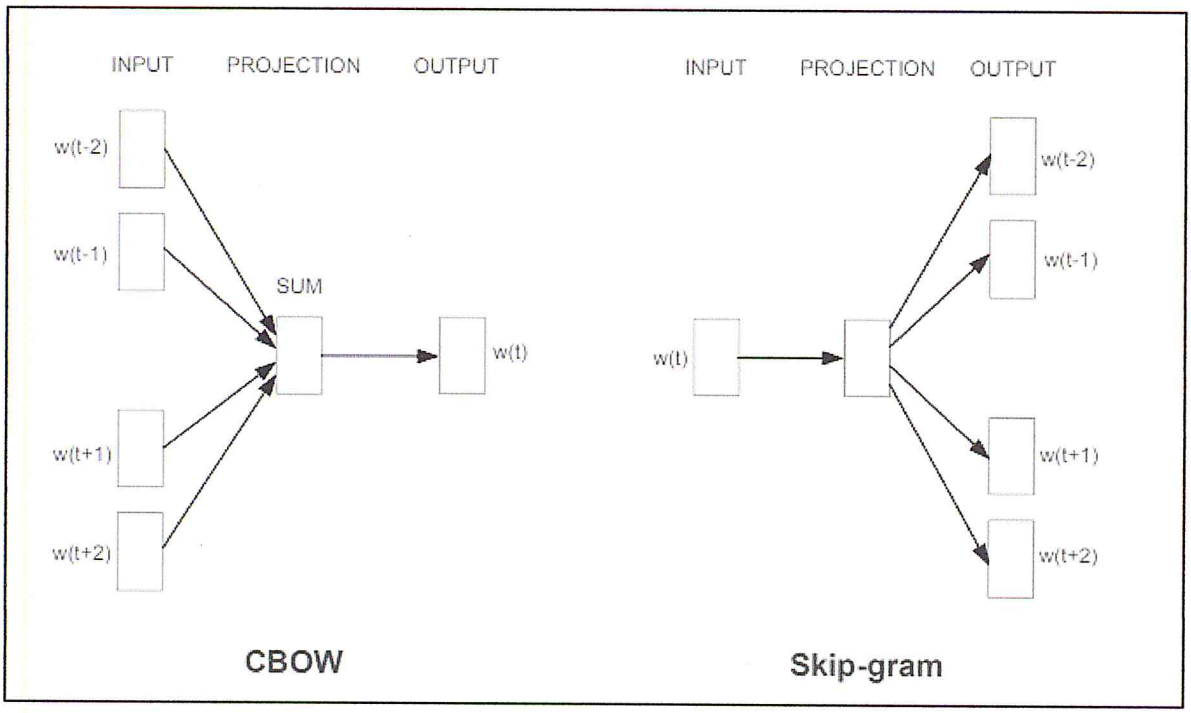


Figure 3 Représentation des modèles SG et CBOW

3.4. Word2Vec :

Word2Vec est une combinaison du modèle Skip gram et Continuous bag of words, développés par Tomas mikolov, Kai chen, Greg Corrado et Jeffrey Dean, qui est une équipe de recherche chez Google, cette combinaison vise à reconstruire le contexte linguistique des mots.

Cette méthode vise à apprendre à partir d'un corpus de phrases, pour former des vecteurs de mots afin de calculer la similarité entre ces différents mots.

On utilisera cette méthode dans notre projet lors de la phase de traitement des requêtes de recherche, pour l'expansion de ces requêtes à l'aide de mots similaires présent dans le corpus des Tweets.

4. Conclusion :

Dans ce chapitre nous avons abordé la notion d'apprentissage machine ou Machine Learning ainsi que le word embedding et ses différents modèles, qu'on utilisera par la suite dans notre modèle de recherche d'information.

Le prochain chapitre comportera la conception de notre modèle de recherche d'information.

Chapitre 4

Conception du modèle

Introduction :

Dans ce troisième chapitre nous allons détailler notre approche pour la conception de notre modèle de recherche d'informations, notre travail se divise en quatre étapes principales ; La première étape est l'acquisition des données (dans notre cas, le corpus des tweets), la deuxième étape consiste à effectuer un prétraitement sur le corpus des tweets, la troisième étape est un traitement qui consiste à effectuer une première recherche sur le corpus en utilisant 50 requêtes différentes, cette première recherche va nous permettre d'extraire les données nécessaires pour effectuer une expansion des requêtes de recherche, la quatrième et dernière étape consiste à effectuer une recherche finale en utilisant les requêtes traitées dans l'étape précédente, pour ensuite pouvoir comparer les résultat de notre recherche avec les résultats fournis par TREC 2011 Web TRACK.

1. Choix du moteur de recherche:

Plusieurs moteurs de recherche d'information open source ont vu la lumière dans ces dernières années. Parmi les plus utiliser nous citons [1] : ht://Dig, Indri, IXE, Lucene, MG4J, IBM OmniFind Yahoo ! Edition, oméga, Terrier, Zettair. Ce sont des bibliothèques de recherche évolutive pour la recherche de texte intégral. Ce sont une base solide, sur laquelle une application de recherche peut être développée. Chaque moteur parmi ces derniers possède des points forts et des points faibles. Si on veut par exemple un meilleur temps de réponse on peut choisir: Indri, IXE, Lucene, XMLSearch. Si on veut des meilleures performances d'indexation Zettair est le Top pour cette tâche. Les meilleurs temps d'indexation sont réalisés par: ht://Dig, Indri, IXE, Lucene, MG4J, Swish-E, Swish++, Terrier, XMLSearch, Zettair . Notre avons choisi le moteur de recherche INDRI pour sa simplicité, ainsi que son interface graphique prête à l'emploi, INDRI fonctionne sous Python ce qui nous permettra d'utiliser un seul langage seulement, car les bibliothèques de Word Embedding sont sous python.

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

2. Description du modèle :

Les modèles de RI classiques utilisent principalement la fréquence des termes et la longueur des documents pour calculer la pertinence textuelle (Ounis, 2011), cependant vu la nature des microblogs (comportant en moyenne 15 termes) les termes n'apparaissent très souvent qu'une seule fois dans un tweet, et donc ces modèles de recherche classiques ne sont pas très adaptés pour être utilisés dans le cas des microblogs, et c'est dans cette optique qu'on a proposé d'effectuer une expansion des requêtes de recherche en utilisant le Machine Learning, ce qui devrait nous permettre d'élargir notre champ de recherche et avoir des résultats plus pertinent qu'un modèle de recherche classique.

Nous allons détailler ci-dessous les quatre étapes principales de notre modèle de RI :

2.1. Acquisition des données :

Cette étape est de loin la plus lente du projet, elle consiste à télécharger la totalité des tweets du corpus TREC2011.

Le corpus des tweets utilisé dans la piste Microblog de TREC 2011, est distribué sous forme de 15 répertoires, chacun contenant environ 100 fichiers .DAT, chacun contenant une liste de (tweet id, user Name, MD5 checksums). Chacun de ces fichiers est appelé bloc d'état (c'est-à-dire bloc de tweets) ; Nous avons commencé par télécharger les tweets depuis tweeter en utilisant leurs identifiant (ID), cependant avec cette méthode, il fallait télécharger les tweets un par an, et vu la taille très importante du corpus (environs 16 millions de tweets) cette méthode n'était pas envisageable et il fallait trouver un autre moyen ;Après plusieurs semaines de recherche nous avons trouvé un API gratuit open source « Tweeter Tools »[(1)] qui nous a permis d'accélérer considérablement le processus (télécharger par millier de tweet au lieu d'un seul) .

On notera que cette étape est seulement présente dans le cadre d'une étude théorique comme celle-ci, car elle ne sera plus utile si on veut effectuer une recherche en temps réel (ce qui est l'objectif final du projet) car l'accès se fera directement aux bases de données de Tweeter ; Par faute de capacité de calcul et de traitement, toutes les étapes du projet vont être réalisées en local pour y remédier.

[1] https://github.com/lintool/twitter-tools

2.2. Prétraitement :

Vu le nombre réduit de caractères dans un tweet (140 lors de la création du corpus TREC 2011, ce nombre a été doublé en 2017) très souvent, le contenu textuel des tweets contient des données non structurées, des abréviations, des incohérences typographiques, des émoticons, des mots vides de sens, …etc. Toutes ces données nécessitent un nettoyage et une normalisation. Cette phase de prétraitement est d'une très grande importance car le recherche doit se faire sur un corpus épuré sinon les résultats n'auront aucun sens.

Le prétraitement se divise en plusieurs étapes citées ci-dessous :

- **Suppression des tweets écrit avec une langue différente :**
  La langue principale du corpus TREC 2011 est l'anglais, donc on a décidé de garder seulement les tweets écrit en anglais et supprimer le reste.
- **Suppression des tweets avec un code d'erreur/spam :**
  Certains tweets comportent un code d'erreur ou sont catégorisés comme étant des spam, ces deux types de tweets seront supprimés.
- **Suppression des fichiers vides :**
  Certains fichiers sont vides, et donc seront automatiquement supprimés.
- **Suppression des retweets :**
  Pour éviter d'avoir un même tweet plusieurs fois, ce qui va forcément fausser les résultats de la recherche, nous avons décidé de supprimer les retweets.
- **Suppression des Stop Words (mots vides) :**
  Un mot vide est un mot non significatif qui figure dans un texte, ces mots sont très communs et très utilisés dans tous les textes (ex : « and », « it », « he », « they ») ces mots n'ont pas d'importance en ce qui concerne la recherche d'information et vont donc seulement ralentir l'exécution de l'algorithme de recherche, et peuvent potentiellement fausser les résultats de ce dernier.
  Pour effectuer la suppression de ces mots, nous avons utilisé une liste de stop words de la langue anglaise, et supprimé tous les mots du bloc « texte » des tweets qui figure dans la liste.
- **Suppression des Abréviation :**
  Au début nous avions décidé de remplacer les abréviations par les mots qui leurs correspondes, mais vu la taille très importante du corpus, certains mots avaient plusieurs abréviations différentes, et dans certains cas une même

abréviation avait plusieurs sens différents, On a donc décidé de les retirer pour éviter d'avoir des résultats non cohérents.

- **Remplacement des URL :**
  Les tweets peuvent contenir des URL, ce qui va fausser les résultats des recherches ; Une des solutions est de supprimer ces URL pour éviter tout problème, mais on a décidé de remplacer ces URL par les mots clés (Key Words) des pages ou renvoi ces URL pour éviter toute perte d'informations pouvant être utiles à la recherche.
- **Racinisation (Stemming) :**
  Le Stemming ou Racinisation est une méthode/procédé qui consiste à transformer un mot pour avoir sa racine, en éliminant ses affixes (Suffixes Préfixes, Postfixes, Antéfixes) en se basant sur des règles de Racinisation.
  Il existe plusieurs algorithmes de racinisation (Stemmer) : Carry (patemostre, P.Franco, J, Wartel, & Saerens, Juillet 2002) qui est un Stemmer pour langue française, ainsi que Porter (porter, 1979) qui est un des plus connus et les plus utilisé pour la langue anglaise.
- **Suppression des Hashtag émoticônes et des ponctuations :**
  Cette étape consiste à éliminer les émoticônes telle que (☺, ☹) ainsi que les ponctuations (, ; ! ?) ainsi que les hashtags (#)

Une fois ces étapes terminées, on devrait obtenir un corpus de Tweets épurés, prêt à l'utilisation, ce qui n'était pas le cas avant le prétraitement.

2.3. Traitement :

Notre corpus est maintenant constitué de Tweets épurés et chacun de ces Tweets est composé seulement de mots qui peuvent avoir un sens pour notre recherche et est donc prêt à l'emploi.

L'objectif final de cette étape est de pouvoir expandre/allonger nos 50 requêtes (Topics) de départ qu'on utilisera ensuite pour faire notre recherche dans la dernière étape.

Voici ci-dessous un tableau comportant les 50 requêtes qu'on va traiter :

Num	requête	Num	requête
1	Ritz carlton lake las vegas	26	US capitol map

2	Fickle creek farm	27	Dutchess county tourism
3	Madam cj walker	28	Atypical squamous cells
4	Indiana child support	29	IOWA food stamp program
5	Sonoma county medical services	30	Fact on uranus
6	Universal animal cuts reviews	31	Equal opportunity employer
7	Cass county missouri	32	Mothers day songs
8	Ralph owen brewster	33	All men are created equal
9	Mayo clinic jacksonville fl	34	Electronic skeet shoot
10	Map of brazil	35	Source of the nile
11	Lymphoma in dogs	36	American military university
12	Kenmore gas water heater	37	Rock and gem shows
13	HP mini 2140	38	Jax chemical company
14	Adobe indian houses	39	Rocky mountain news
15	Pacific northwest laboratory	40	East ridge high school
16	California franchise tax board	41	VA DMV registration
17	Dangers of asbestos	42	Illinois state tax
18	Poem in your pocket day	43	Arkadelphia health club
19	Interview thank you	44	Trombone for sale
20	TV on computer	45	Vines for shade

21	Sit and reach test	46	Sherwood regional library
22	Culpeper national cemetery	47	Tangible personal property tax
23	Von willebrand disease	48	Martha stewart and imclone
24	Bowflex power pro	49	Uplift at yellowstone national park
25	Butter and margarine	50	TN highway patrol

Tableau 2 : Liste des requêtes de recherche (topics)

- On va d'abord effectuer une première recherche en utilisant le moteur de recherche INDRI, Cette première recherche aura comme entrée les 50 requêtes listées dans le tableau ci-dessus.
- Chaque requête à un date (Timestamp) on effectuera la recherche sur le tweets ayant une date <= à la date de la requêtes, pour éviter de rechercher sur des tweets qui sont paru après la requête, et qui n'ont donc aucun sens pour notre recherche.
- Le résultat de cette recherche comportera une liste de tweets, cette liste de tweets sera le corpus sur lequel on effectuera notre apprentissage en utilisant la méthode Word2Vec
- Une fois l'apprentissage terminé, on prendra chaque mot de la requête, et on trouve les 3 premiers mots plus proches en utilisant Word2Vec.
- La dernière étape consiste à élargir (éxpandre) la requête avec les mots similaires récolté avec Word2Vec de chaque mot qui forme la requête.
  Exemple :
  Si on considère la requête (Topic) de recherche suivante « Bonjour Homme » :
  On suppose que le résultat de la recherche avec Word2Vec est le suivant :
  Bonjour est proche de : Bonsoir, salutation, journée
  Homme est proche de : Humain, male, Père
  La requête « Bonjour homme » après expansion deviendra « Bonjour bonsoir salutation journée homme humain male père »

On notera que ceci n'est qu'un exemple théorique pour mieux illustrer les étapes ci-dessus, et que le résultat pratique de la recherche aura probablement plus de sens.

2.4. Recherche :

Une fois que toutes les requêtes auront subit une expansion, on procèdera à la recherche en utilisant ces requêtes traitées, à l'aide du moteur de recherche INDRI.

Cette étape est la dernière étape de notre modèle de recherche, et nous permettra d'avoir une liste de tweets comme résultats.

3. Architecture globale du modèle :

Nous allons présenter ci-dessous l'architecture globale de notre modèle de recherche d'information :

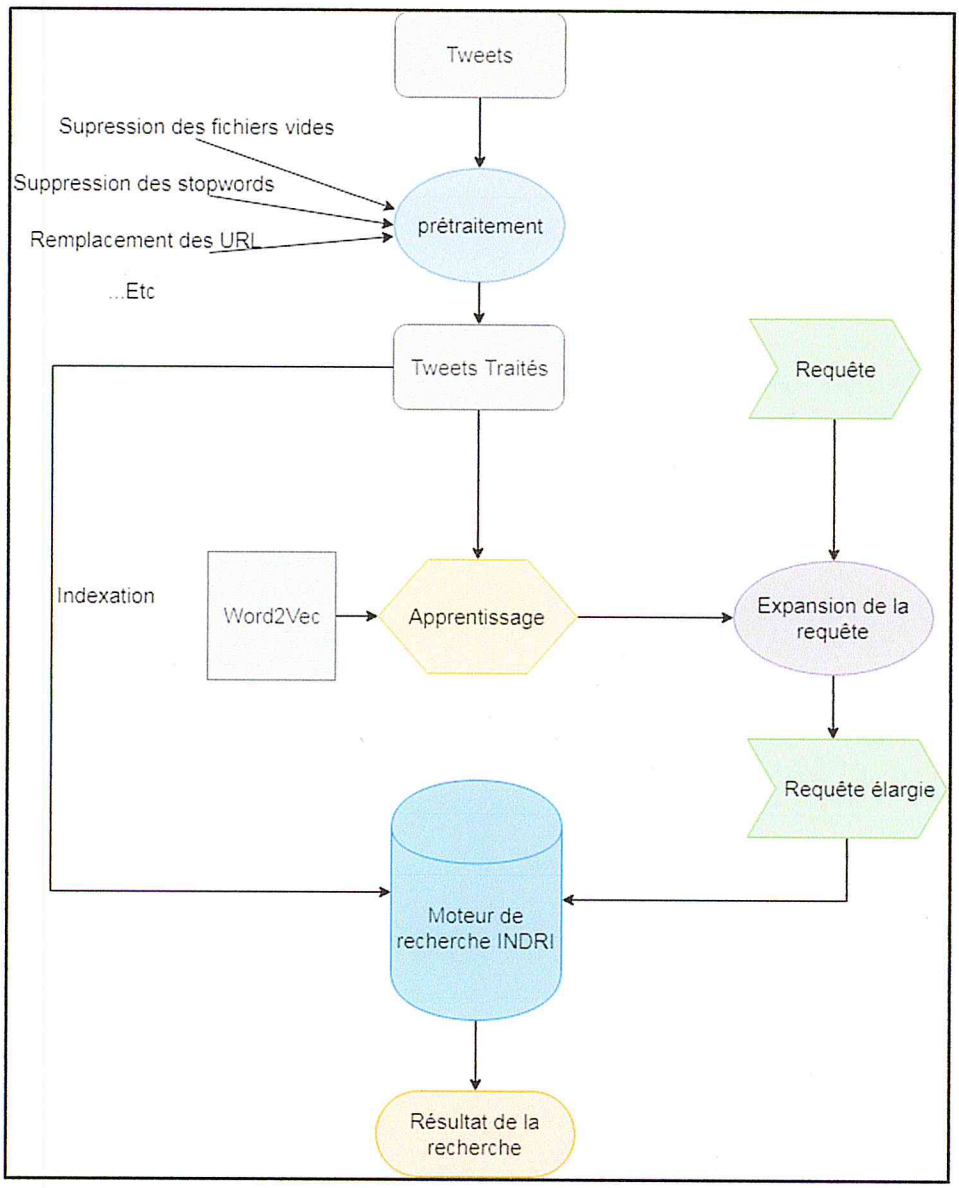


Figure 1 : Architecture globale du modèle de recherche d'information

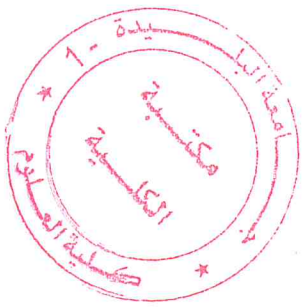
4. Conclusion :

Dans ce chapitre nous avons détaillé notre approche en décrivant le modèle de recherche ainsi que le moteur de recherche utilisé, et nous avons ensuite détaillé quatre étapes principales du modèle de recherche d'information proposé (Acquisition des données, prétraitement, traitement et finalement la recherche) en illustrant le tout avec une architecture globale du modèle proposé.

Dans le prochain et dernier chapitre, nous allons voir les différents composants qui permettent d'implémenter ce system.

Chapitre 5

Implémentation du modèle



1. Introduction :

Nous allons présenter dans ce chapitre l'environnement de travail ainsi que les différents outils utilisés (Software/Hardware), Nous verrons ensuite quelques illustrations des différentes parties de l'application du modèle de recherche proposé, Ce qui constituera la partie pratique de ce mémoire.

2. Environnement de travail :

Cette partie comporte les différents outils (Hardware et Software) utilisés pour l'implémentation du projet

2.1. Matériel utilisé :

Dans le cadre de ce projet, L'ordinateur utilisé comporte les spécifications suivantes :

-Processeur : Intel ® Core™ i5-6500 CPU @3.20GHz

-Mémoire installée (RAM) : 8 Go

-Type du système : Système d'exploitation 64 bits

-GPU : Saphire Radeon rx370 4 Go

Ainsi qu'un ordinateur portable avec les spécifications suivantes :

Acer intel core i5 3200 GHz, 4 Go ram et Système d'exploitation 64 bits

2.2. Python (Ver 2.7.15) :

Python est un langage de programmation objet interprété multiparadigme et multiplateformes crée en 1991 par Guido van Rossum. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl. (A Brief Timeline of Python, 2009)

Le langage Python est placé sous une licence libre proche de la licence BSD et fonctionne sur la plupart des plates-formes informatiques, des Smartphones aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par MacOs, ou encore Android, iOS, et aussi avec Java ou encore .NET. Il est conçu pour optimiser la

productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Il est également apprécié par certains pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des mécanismes de bas niveau, permet une initiation aisée aux concepts de base de la programmation.

Nous avons choisi python pour sa facilité d'utilisation et rapidité d'exécution ainsi que la disponibilité de nombreuses bibliothèques permettant d'effectuer l'apprentissage machine.

2.3. Anaconda :

Anaconda est une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement. Les versions de paquetages sont gérées par le système de gestion de paquets conda. La Distribution Anaconda est utilisée par plus de 6 millions d'utilisateurs, et il comprend plus de 250 paquets populaires en science des données adaptés pour Windows, Linux et MacOS. (Anaconda)

2.4. Windows 7 :

Windows 7 (précédemment connu en tant que Blackcomb et Vienna) est un système d'exploitation de la société Microsoft, sorti le 22 octobre 2009 et successeur de Windows Vista. Bien que le système s'appelle Windows 7, il s'agit de la version NT 6.1. Windows 7 est progressivement remplacé par Windows 8 à partir du 30 octobre 2012, le support de Windows 7 RTM a pris fin le 9 avril 2013 tandis que la version SP1 a vu son support standard se terminer en janvier 2015 et verra son support étendu se terminer en janvier 2020[3]. Cette version de Windows reprend l'acquis de Windows Vista tout en apportant de nombreuses modifications, notamment par divers changements au niveau de l'interface et de l'ergonomie générale, un effort particulier pour la gestion transparente des machines mobiles et le souci d'améliorer les performances globales du système (fluidité, rapidité d'exécution même sur des systèmes moins performants, tels les netbooks) par rapport à son prédécesseur.

En identifiant cette mouture par son numéro de version (il s'agit de la septième version de Windows), Microsoft renoue avec une logique abandonnée depuis Windows 3.11 et Windows NT 4.0. La tradition voulait jusqu'ici que les versions de Windows soient

identifiées par référence à l'année de sortie (Windows 95…) ou par une appellation ad hoc (Windows XP ou Windows Vista). Néanmoins, Windows 7 se base sur le noyau NT 6.1.

2.5. Notepad++ :

Notepad++ est un éditeur de texte libre générique développé par Don Ho, fonctionnant sous Windows, codé en C++, qui intègre la coloration syntaxique de code source pour les langages et fichiers C, C++, Java, C#, XML, HTML, PHP, JavaScript, makefile, art ASCII, doxygen, .bat, MS fichier ini, ASP, Visual Basic/VBScript, SQL, Objective-C, CSS, Pascal, Perl, Python, R, MATLAB, Lua, TCL, Assembleur, Ruby, Lisp, Scheme, Properties, Diff, Smalltalk, PostScript et VHDL ainsi que pour tout autre langage informatique, car ce logiciel propose la possibilité de créer ses propres colorations syntaxiques pour un langage quelconque.

Ce logiciel, basé sur la composante Scintilla, a pour but de fournir un éditeur léger (aussi bien au niveau de la taille du code compilé que des ressources occupées durant l'exécution) et efficace. Il est également une alternative au bloc-notes de Windows (d'où le nom). Le projet est sous licence GPL version 2.

3. Les bibliothèques

3.1. Numpy :

NumPy[1] est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

[1] https://github.com/numpy/numpy/blob/master/LICENSE.txt

3.2. Scipy :

SciPy[1] est un projet visant à unifier et fédérer un ensemble de bibliothèques Python à usage scientifique. Scipy utilise les tableaux et matrices du module NumPy.

Cette distribution de modules est destinée à être utilisée avec le langage interprété Python afin de créer un environnement de travail scientifique très similaire à celui offert par Scilab, GNU Octave, Matlab voire R.

Il contient par exemple des modules pour l'optimisation, l'algèbre linéaire, les statistiques, le traitement du signal ou encore le traitement d'images.

Il offre également des possibilités avancées de visualisation grâce au module matplotlib.

Afin d'obtenir d'excellentes performances d'exécution (point faible des langages interprétés), la plupart des algorithmes de SciPy et NumPy sont codés en C. Le module NumPy permet d'appliquer des opérations simultanément sur l'ensemble d'un tableau permettant d'écrire un code plus lisible, plus facile à maintenir et donc plus efficace.

3.3. Twitter4j:

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

Twitter4j [2] est une librairie facilitant l'utilisation des API Twitter.

3.4. Gensim :

Gensim est une bibliothèque open source robuste de Topic modelling, qui utilise Numpy, Scipy et optionnellement Cython pour plus de performance, cette bibliothèque est conçue pour gérer un collection de données assez importante.

4. Illustrations des étapes du projet :

Nous allons voir ci-dessous des captures d'écran représentant différentes étapes du projet :

[1] https://github.com/scipy/scipy/releases/tag/v1.1.0
[2] http://twitter4j.org

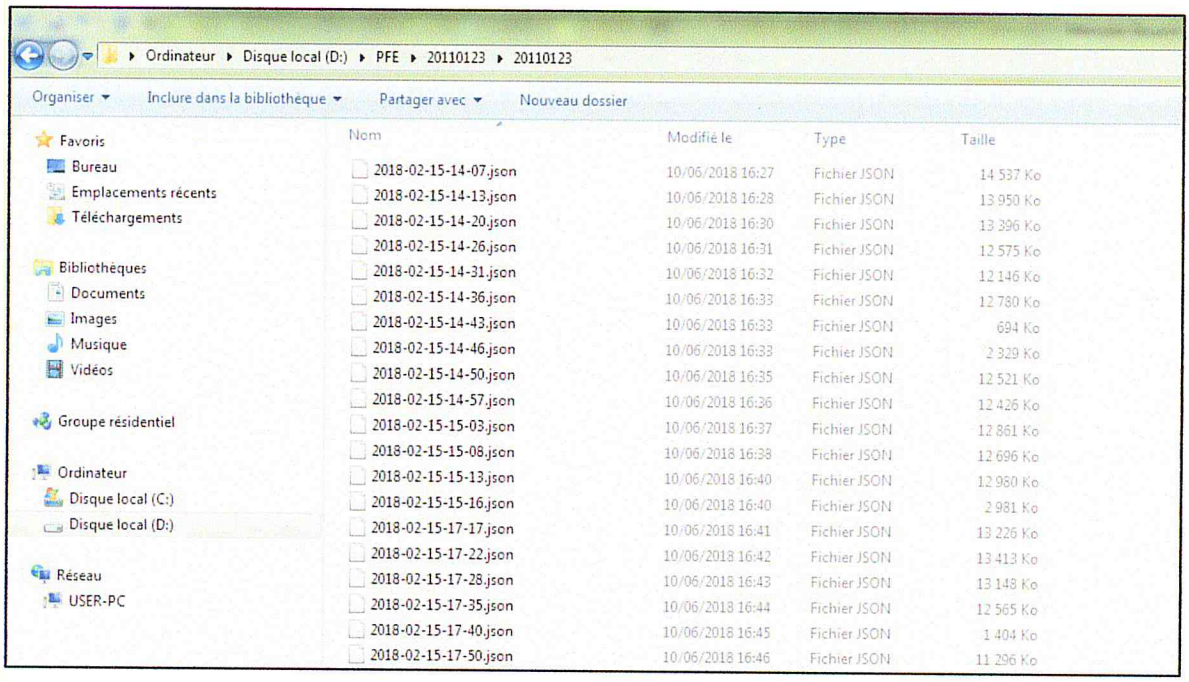


Figure 1 : Screen montant le format initial des tweets téléchargés

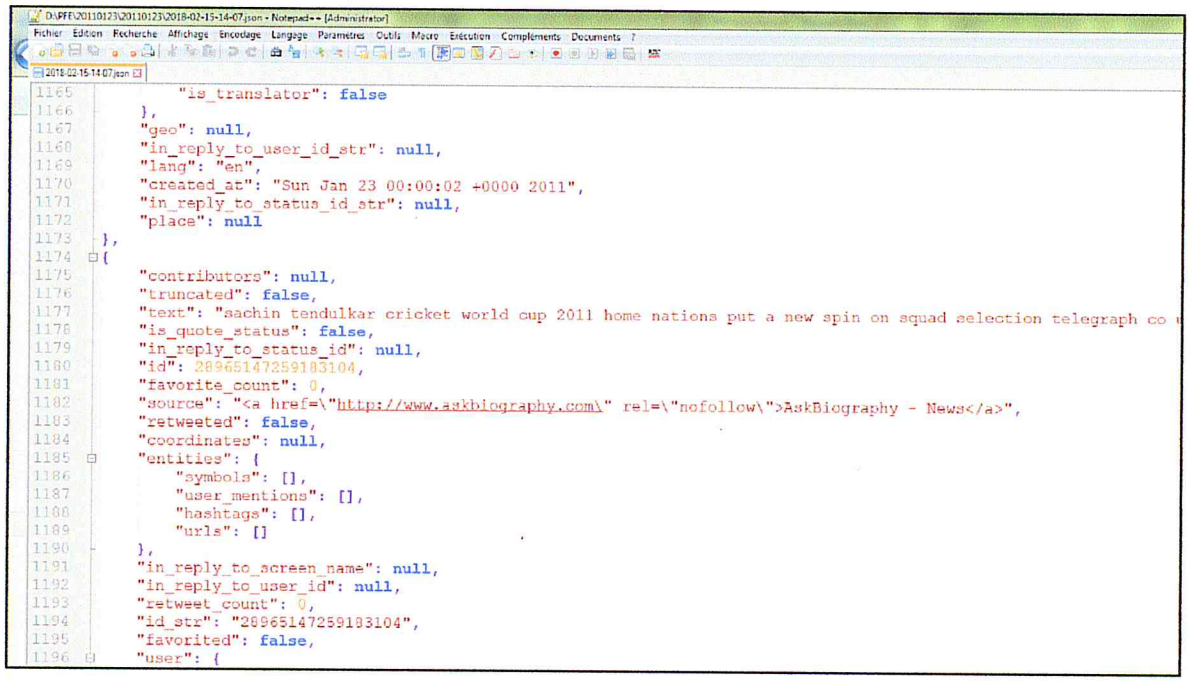


Figure 18 : Screen montrant le contenu d'un fichier Json téléchargé depuis Tweeter

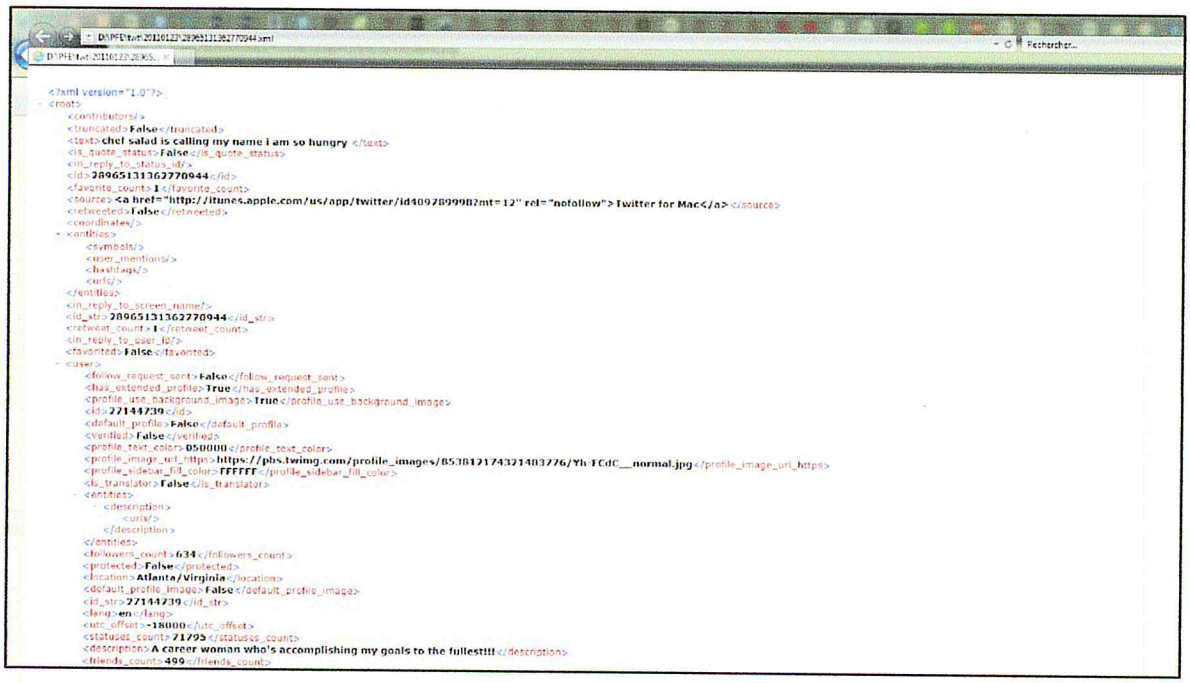


Figure 19 : Screen d'un fichier de Tweets converti en XML et Prétraité

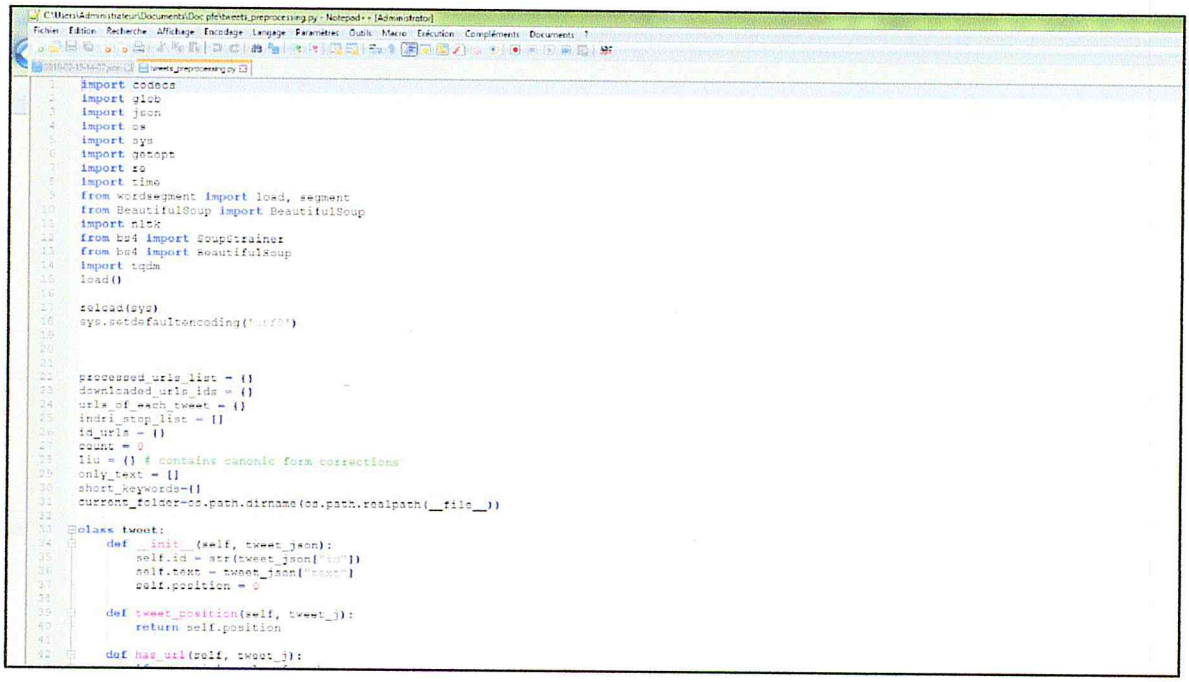


Figure 20 : Screen d'une partie de l'algorithme de prétraitement

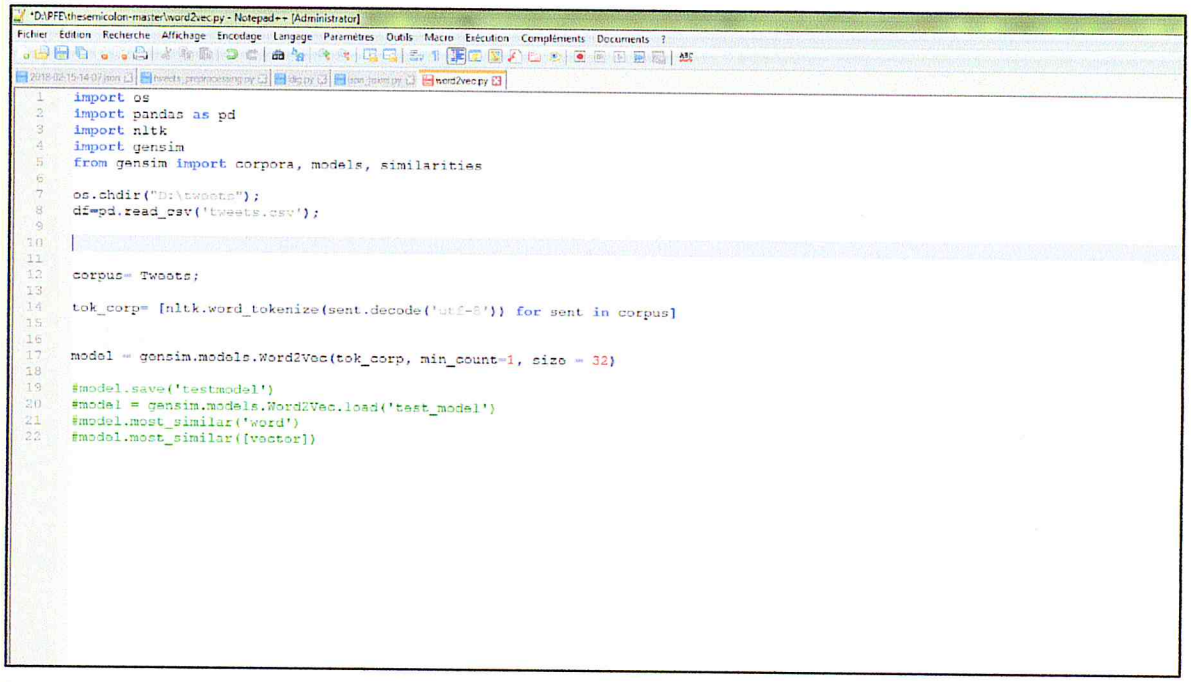


Figure 2 : Algorithme Word2Vec sur python

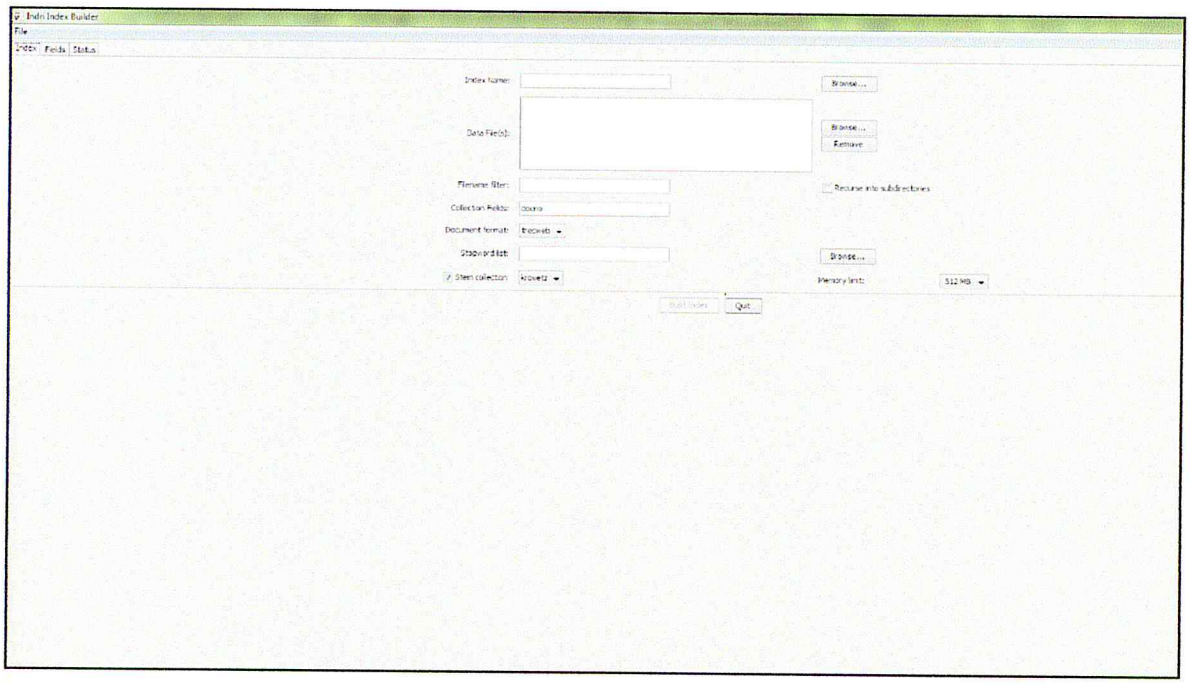


Figure 3 : Interface d'indexation sur INDRI

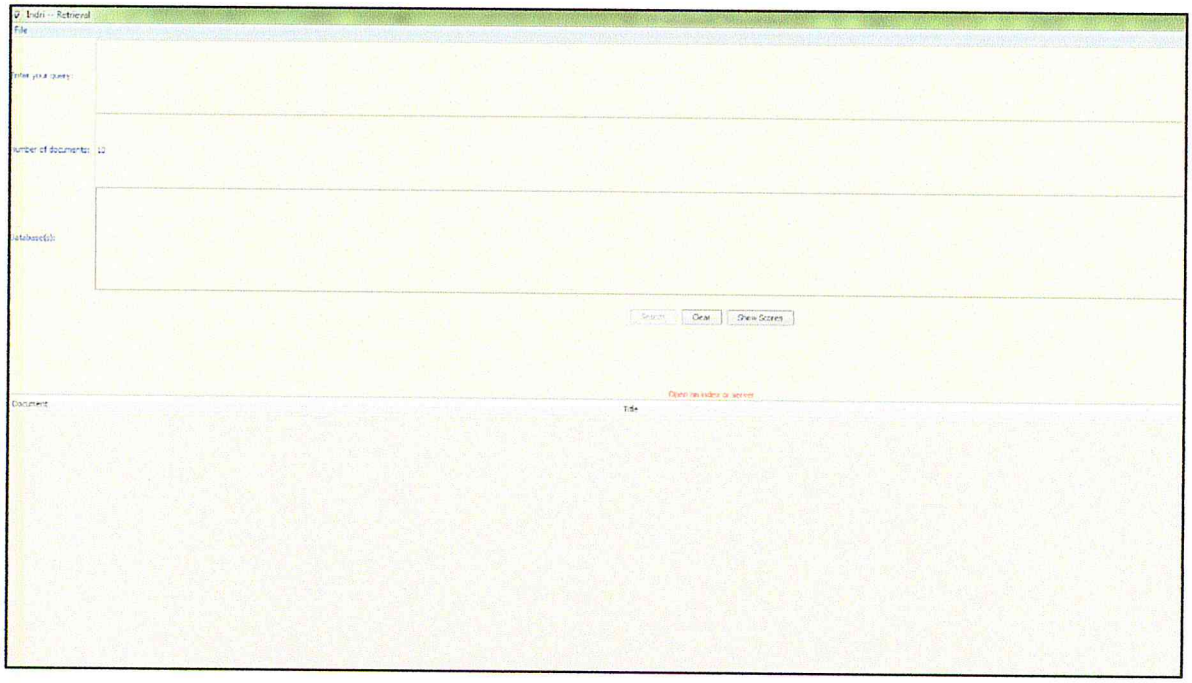


Figure 4 : Interface de recherche sur INDRI

5. Conclusion :

Dans ce dernier chapitre nous avons déterminé l'environnement de travail (Hard/Software), les technologies utilisées ainsi que les différentes bibliothèques utilisées dans le cadre de notre projet, puis on a montré un aperçu de quelques captures d'écran montrant différente phases du projet.

Conclusion générale

Notre travail a porté sur la recherche d'information sociale dans les microblogs, l'objectif initial était de proposer un modèle de recherche d'information en se basant sur l'apprentissage machine pour répondre à un besoin d'informations spécifié par un utilisateur sous la forme de requêtes (topics).

Le corpus utilisé dans le cadre de cette recherche, est TREC microblog 2011 qui est constitué de tweets, notre contribution se situe au niveau de l'expansion des requêtes de recherches afin d'améliorer le résultat de ces dernières.

La première étape de ce travail, était de parcourir l'état de l'art des systèmes de recherche d'informations dans les microblogs, ce qui nous a permis de distinguer les faiblesses de ces systèmes.

Nous avons par la suite abordé la notion de l'intelligence artificielle et l'apprentissage machine ainsi que le word embedding et son utilisation dans le cadre de notre projet.

La première étape de l'exécution du projet était le téléchargement du corpus des tweets, cette étape est de loin la plus lente en raison du volume très important des tweets.

Nous avons par la suite traité les tweets téléchargés pour éviter toute source de conflit lors ce qu'on effectuera une recherche sur ces dernier, nous avons effectué par la suite un apprentissage machine sur le corpus des tweets épurés pour pouvoir effectuer une expansion des requêtes de recherche, et c'est ce qui a été fait.

Enfin nous pouvons humblement estimer que l'objectif principal qui était de proposer un modèle de recherche à base de Machine Learning en effectuant une expansion de la requête, a était atteint.

En perspective nous souhaitons l'application de cette méthode avec des tests et comparaison avec le service d'évaluation fourni par TREC 2011 (TrecEval)

Bibliographie

A Brief Timeline of Python. (2009).

Amini, M.-R. (2013). *Recherche d'Information - Applications, modèles et algorithmes.*

Anaconda, I. (s.d.). *What is anaconda*. Récupéré sur Anaconda: https://www.anaconda.com/what-is-anaconda/

BOURAMOUL. (2011). *RECHERCHE D'INFORMATION Sémantique sur le web.* Constantine.

chy, A., ullah, M. z., & Aon, M. (2015). *Combining Temporal and content aware features for microblog retrieval.*

Damak. (2014). *Etude des facteurs de pertinence dans la recherche de microblogs.* Paris.

Dean, K. C. (2013). *Efficient Estimation of Word Representations in.*

EFRON. (2011). *Hashtag retrieval in a microblogging environment.*

Fabien, M. (2011). *Apprentissage artificiel "machine Learning".* paris.

Firas, D. (2013). *Effectiveness of state of the art features for microblog.*

Futura. (s.d.). *Futura tech*. Consulté le 06 15, 2018, sur https://www.futura-sciences.com/tech/definitions/informatique-intelligence-artificielle-555/

Greg Corrado, T. M. (2013). *Efficient Estimation of Word Representations in.*

Larousse. (s.d.). *L'intelligence artificielle.*

Lecun, Y. (2015). *L'apprentissage prédictif est le défi scientifique de l'IA.*

Medlin, Willis, & Arguello. (2012). *Incorporating Temporal informationing microblog retrieval.*

Middleton. (2007). *A Comparison of Open Source Search Engines, Technical report.*

Miles, Lin, he, J., & vries, A. d. (2014). *Temporal Feedback for tweet search with non parametric density estiamation.*

Mitchell, T. (1997). *Machine Learning.* international édition.

Ounis. (2006). *Overview of trec 2005.*

Ounis, I. L. (2011). *Overview of the TREC-2011 Microblog : 20th Text Retrieval Conference.*

patemostre, M., P.Franco, J, L., Wartel, D., & Saerens, M. (Juillet 2002). *Carry, un algorithme de désuffixation pour le français.*

porter, M. (1979). *New models in probabilistic information retrieval.* London: British library research and development.

Salton. (1970). *le système SMART* .

Uehara, k., Seki, K., & miyanashi, T. (2013). *Combining Recency and Topic dependent temporal variation for microblog search.*

Vukotic, V. (2015). *Word embeddings: Supervised and Unsupervised Methods in Sentiment Analysis.*

Wetice. (2012). Use of Twitter and semantic resources recovery in educational context.

Yang, S. J. (2010). *A surveu on transfer learning.*

Yates, B. (1999). *Modern Information Retrieval.*

Yoo, J. (2014). *Why people use Twitter: social conformity and social value perspectives.*