République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida

Rapport de Master

Spécialité : Informatique

Option : Génie Logiciel

Thème : Un nouveau modèle temporel de recherche d'information dans les microblogs

---

Réalisé par :

- **LADLI Noureddine**
- **HEMIS Hamza**

Encadré par :

- **Mme Z.BOUCETTA**

Année : 2022/2023

REMERCIEMENTS

Nous commençons par remercier DIEU, le tout-puissant et clément, pour sa bienveillance, pour nous avoir accordé l'énergie et la persévérance nécessaires pour mener à bien ce travail. Nous souhaitons exprimer notre gratitude sincère et profonde à l'endroit de l'universite de Blida.

Nous sommes également très reconnaissants envers notre encadrante, Mme BOUCETTA ZOUHEL, pour le temps qu'elle nous a accordé, pour ses conseils avisés et pour son intérêt et sa compréhension à notre égard. Nous sommes aussi très reconnaissants à l'ensemble des membres du jury Mme. MEZZI et Mr. FERFERA pour avoir bien voulu examiner et juger notre travail. Merci pour votre présence, qui nous honore.

Nos sincères remerciements vont également au chef du département d'informatique. Nous tenons également à exprimer notre gratitude la plus sincère à tous les professeurs qui nous ont offert une formation multidisciplinaire de très haut niveau, parfaitement adaptée à la réalité de l'ingénierie informatique.

Enfin, nous adressons nos remerciements à nos parents et à tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce document.

Dédicace

Je dédie ce modeste travail,

A mes parents en guise de reconnaissance et de gratitude pour les sacrifices qu'ils ont faits,

A mon frère, à mes sœurs, à qui je dois tout l'amour, avec tous mes vœux de les voir réussir dans leurs vies,

A la boite d'informatique « PCONLINE »,

A mes ami(e)s, à qui je souhaite le succès pour l'amitié qui nous a toujours unis,

A tous ceux qui me sont chers.

H.HEMIS

Dédicace

A mes chers parents qui m'ont soutenu et encouragé durant ces années d'etudes,
Qu'ils trouvent ici le temoignage de ma profonde reconnaissance,
A mon frère , ma soeur et ma cousine , mes grand-parents et ceux qui m'ont partagé avec moi tous les moments d'emotions lors de la realisation de ce travail , Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours ,
A ma famille , mes cousins , mes proches et a ceux qui me donnent de l'amour et de la vivacité ,
A mon binome hamza et a tous mes amis qui m'ont encouragé , et à qui je souhaite plus de succès .
A tous ceux que j'aime .

N.LADLI

Résumé

Twitter devient une source d'information majeure avec plus de 2,1 milliards [1] de requêtes quotidiennes. Cependant, la nature des tweets, souvent courts, mal orthographiés et avec une syntaxe particulière, pose des défis aux modèles traditionnels de recherche d'informations basés sur la fréquence des termes.

Afin d'accroître l'efficacité de la recherche de tweets pertinents, nous avons formulé un nouveau modèle intégrant les dimensions temporelle et thématique. Notre contribution réside dans l'identification de groupes de tweets partageant un lexique commun, tout en présentant des caractéristiques temporelles congruentes avec celles de la requête.

Par la suite, nous avons procédé à une analyse comparative de nos résultats avec les travaux antérieurs, visant à démontrer l'efficacité et la pertinence de notre modèle, avec pour objectif d'atteindre des performances supérieures dans la recherche d'informations sur Twitter.

Mots clés : Twitter , tweets , requete , biterme , recherche temporelle , topics modeling , ri dans les microblogs.

1. https://kinsta.com/fr/blog/statistiques-twitter/

Abstract

Twitter is becoming a major source of information with more than 2.1 billion daily requests. However, the nature of tweets, often short, misspelled and with a particular syntax, poses challenges to traditional patterns of searching for information based on term frequency.

In order to increase the effectiveness of the search for relevant tweets, we have formulated a new model integrating temporal and thematic dimensions. Our contribution lies in the identification of groups of tweets sharing a common lexicon, while presenting temporal characteristics congruent with those of the query.

Subsequently, we conducted a comparative analysis of our results with previous work, aiming to demonstrate the effectiveness and relevance of our model, with the objective of achieving superior performance in the search for information on Twitter.

keywords : Twitter , tweets , query , biterm , temporal search , topics modeling , searching for information in microblogs.

ملخص

أصبح تويتر مصدرًا رئيسيًا للمعلومات مع أكثر من 2.1 مليار طلب يوميًا. ومع ذلك، فإن طبيعة التغريدات، التي غالبًا ما تكون قصيرة وخطأ إملائي وبشكل خاص، تشكل تحديات للأنماط التقليدية للبحث عن المعلومات بناءً على تكرار المصطلح.

من أجل زيادة فعالية البحث عن التغريدات ذات الصلة، قمنا بصياغة نموذج جديد يدمج الأبعاد الزمنية والمواضيعية. تكمن مساهمتنا في تحديد مجموعات التغريدات التي تشارك معجمًا مشتركًا، مع تقديم خصائص زمنية تتوافق مع خصائص الاستعلام.

بعد ذلك، أجرينا تحليلًا مقارنًا لنتائجنا مع العمل السابق، بهدف إظهار فعالية وأهمية نموذجنا، بهدف تحقيق أداء متفوق في البحث عن معلومات على تويتر.

كلمات مفتاحية:تويتر ،التغريدات، الاستعلام ، البتيرم ، البحث الزمني ، البحث في المدونات الصغيرة ، نمذجة المواضيع .

Table des matières

Table des figures

Liste des tableaux

Acronymes

API Application Programming Interface. 21, 22

BTM Biterm Topical Modal. 26, VIII

CLEF Cross-Language Evaluation Forum. 16

GPS Global Positioning System. 11

GUI Graphical User Interface. 33

HTML HyperText Markup Language. 21

IDF Inverse Document Frequency. 9, VII

JSON JavaScript Object Notation. 21

KL Kullback Leibler. 28

MAP Mean Average Precision. 17, 18

NLTK Natural Language ToolKit. 23

NTCIR National Center for Science Information Systems. 16

RE Regular Expression. 23

RI Recherche d'Informations. 1, 4–7, 12, 13

SRI Système de Recherche d'Informations. 4, 7, 9

TF Term Frequency. 9, VII

TREC Text REtrieval Conference. 16, 21, 22

URL Uniform Ressource Locator. 10

Introduction générale

La recherche d'information (RI) consiste à trouver et à fournir des documents pertinents à un utilisateur en fonction de ses besoins. Le challenge réside dans la sélection des documents les plus pertinents parmi un grand nombre de ressources disponibles. Les systèmes de recherche d'information sont les outils informatiques qui implémentent cette tâche en comparant la requête de l'utilisateur avec le contenu des documents via une fonction d'appariement.

Twitter fondu en 2006, est devenu rapidement l'une des plateformes de microbloggings les plus populaires au monde. Il a connu une croissance exponentielle au fil des ans, car les utilisateurs ne se contentent plus de consommer de l'information, mais contribuent également à sa production. Ce flux de publications complique l'accès à l'information pour les microblogueurs.

La recherche d'informations dans le corpus des microblogs présente un véritable challenge pour les modèles de recherche d'informations classiques, vu la spécificité des tweets et le volume massif du corpus.Parmi les problèmes rencontrés lors d'une recherche basée sur un appariement thématique tweet-requête : la taille limitée du tweet qui ne dépasse pas 140 caractères, la non-redondance des termes dans le texte du tweet, l'écriture du texte avec du largo et des abréviations.

Pour améliorer la sélectivité des tweets pertinents, les travaux connexes ont proposé plusieurs approches, comme ils ont évalué plusieurs facteurs de pertinences en plus des facteurs liés au contenu. Ils sont souligné l'efficacité de l'introduction de l'aspect temporel dans les modèles de recherche des microblogs. Dans ce contexte s'inscrit notre travail, il vise à améliorer la sélectivité des tweets pertinents via l'usage de l'aspect temporel en plus de l'aspect thématique (contenu).

L'objectif de ce travail est la réalisation d'un modèle de recherche d'information dans les microblogs. Ce modèle va permettre d'améliorer la sélectivité des tweets pertinents sur la base de l'aspect temporel ainsi que l'aspect thématique.Notre contribution s'agit de la recherche du groupes de tweets qui partagent le même vocabulaire comme possede les mêmes caracteristiques temporelles que la requête.

Notre mémoire est organisée de la manière suivante :

Chapitre 1 : ce chapitre du mémoire offre une introduction aux concepts de base de la recherche d'information, tout en se concentrant spécifiquement sur la recherche d'information dans les microblogs, avec une attention particulière portée sur Twitter.

Chapitre 2 : ce chapitre met en avant notre contribution, qui consiste en une nouvelle approche pour la recherche de microblogs pertinents. L'objectif principal de cette approche est d'améliorer les performances de recherche dans le corpus des tweets. Notre contribution repose sur l'utilisation des techniques temporelles pour la recherche de tweets pertinents par rapport à une requête. Nous proposons une méthode novatrice qui exploite l'aspect temporel et thématique pour trouver le meilleur groupe des tweets qui partagent le même vocabulaire et possède les mêmes caractéristiques temporelles que la requête.

Chapitre 3 : ce chapitre est crucial, il présente les outils d'implémentation utilisés dans notre projet, examiner les résultats de nos tests et procéder à une évaluation rigoureuse de ces derniers. Ces informations nous permettront d'apprécier pleinement l'efficacité de notre solution et de fournir des perspectifs pour des éventuelles améliorations futures.

Chapitre I

ETAT DE L'ART

I.1 Introduction

Dans ce chapitre, nous aborderons deux sujets : la recherche d'informations (RI) et les services de microblogging.La RI consiste à satisfaire les besoins d'informations des utilisateurs à partir d'une collection de documents.Les systèmes mêmes de recherche d'informations automatisent cette tache en évaluant la pertinence des résultats.

Les services de microbloggings sont des plateformes de communication et de partage de messages textuels. Ils permettent aux utilisateurs de partager des informations sur leurs activités et leurs pensées.Ces services sont devenus des outils de collaboration rapides et pratiques, autant pour les entreprises que pour les communautés virtuelles.

Twitter est le service de microblog le plus populaire sur Internet, attirant des millions de visiteurs mensuels.Cependant, la grande quantité de données générées rend difficile la recherche des dernières nouvelles.La recherche d'informations dans les microblogs est également complexe en raison de la taille réduite des articles et de la limitation des recherches par mots-clés.Les chercheurs se dirigent donc vers l'intégration de contextes supplémentaires via les aspects sociaux et temporels.

En résumé, ce chapitre examine la recherche d'informations et les services de microblogging. Il met en évidence les défis lies à la recherche d'informations dans les microblogs et explore quelques approches basées sur les aspects temporels pour améliorer cette tache.

I.2 Définition :

« La recherche d'informations (RI) est la science de la recherche de l'information dans les documents, de la recherche de documents eux-mêmes, de la recherche de métadonnées qui décrivent les documents, ou de la recherche dans les bases de données, qu'elles soient relationnelles, à base de schémas ou distribuées sur un réseau » [2].

I.2.1 Composantes d'un sytème de recherche d'information :

La recherche d'information, se compose de trois concepts de base :

I.2.1.1 Requête

C'est l'expression du besoin de l'utilisateur. La requête est l'interface entre le SRI et l'utilisateur. Une requête peut être soit un ensemble de mots clés, ou exprimée en langage naturel, booléen ou graphique [3].

I.2.1.2 Modèle de représentation

"Un modèle de représentation en RI est une formalisation mathématique pour exprimer la pertinence d'un document à une requête. Il définit la façon dont les documents et les requêtes sont représentés, et fournit une fonction de classement qui attribue à chaque paire document-requête un score de pertinence" [2].

I.2.1.3 Modèle de recherche

"Un modèle de recherche en RI est une théorie ou une spécification qui décrit à la fois la représentation des données et le mécanisme d'accès aux données (y compris la formulation de la requête) pour répondre aux besoins d'information des utilisateurs. Les modèles de recherche sont souvent basés sur un modèle de représentation, qui est une formalisation mathématique de la façon dont les documents et les requêtes sont représentés" [2].

I.2.2 Modèles de recherche d'information :

Il existe plusieurs modèles de recherche d'information, chacun ayant ses propres forces et faiblesses en fonction de la nature de l'information recherchée et du contexte de la recherche. Voici quelques modèles les plus couramment utilisés :

I.2.2.1 Modèle booléen :

Le modèle booléen [1] est le plus simple des modèles de RI. C'est aussi le premier qui s'est imposé dans le monde de la recherche d'information. Il est basé sur la théorie des ensembles et l'algèbre de Boole. Le modèle booléen considère que les termes de l'index sont présents ou absents d'un document. En conséquence, les poids des termes dans l'index sont binaires. Une requête q est composée de termes liés par les trois connecteurs logiques ET, OU, NON. La similarité entre un document et une requête est définie par : rsv(q, d) = (1 si d appartient à l'ensemble décrit par la requête 0 sinon)

I.2.2.2 Modèle vectoriel :

Le modèle vectoriel [1] fait partie des modèles statistiques. L'utilisation des statistiques a pour but d'une part de caractériser d'un point de vue quantitatif les termes et les documents et d'autre part de mesurer le degré de pertinence d'un document vis à vis d'une requête. Le but final est d'arriver à retourner une liste ordonnée de documents selon ce degré. Un autre avantage réside dans l'expression des besoins de l'utilisateur : contrairement au modèle booléen où les termes de la requête doivent être reliés par des connecteurs logiques, l'utilisateur peut ici aussi exprimer son besoin en information en langage naturel

ou sous forme d'une liste de mots clés. La mesure de similarité entre le document fourni et la représentation des documents de la collection est utilisée pour ordonner ces documents. Le critère de similarité est ainsi défini : plus deux représentations contiennent les mêmes éléments, plus la probabilité qu'elles représentent la même information est élevée. Une telle définition revient en fait à compter le nombre d'éléments que partagent la requête et la représentation du document. Pour ce faire, considérons la représentation d'un document comme un vecteur → dj= w1,j , w2,j , . . . , wt,j , où wi,j est le poids (0 ou 1) des termes dans le documents, t étant le nombre total de termes de l'index, et considérons la représentation de la requête comme un vecteur → q = w1,q, w2,q, . . . , wt,q, avec les mêmes notations. La mesure de similarité la plus simple est alors le produit scalaire :

$$RSV(\vec{dj}, \vec{q}) = \sum_{i=1}^{t} w_{i,j} \cdot w_{i,q} \tag{I.1}$$

I.2.2.3 Modèle probabiliste :

Le modèle probabiliste dans la recherche d'information est une méthode qui se base sur la théorie des probabilités pour évaluer la pertinence des documents par rapport à une requête donnée. Il calcule la probabilité qu'un document soit pertinent pour la requête de l'utilisateur en se basant sur des attributs comme la présence ou l'absence de certains termes dans le document.

Dans ce modèle, chaque document est considéré comme un événement indépendant, et la pertinence de ce document pour une requête donnée est déterminée par le calcul de la probabilité conditionnelle de pertinence de ce document sachant la requête [4] [5].

I.2.2.4 Les modèles de langues

Le modèle de langue est un cadre puissant pour la récupération d'informations (RI) qui a été développé en réponse aux limites des modèles de RI traditionnels, y compris le modèle probabiliste. Alors que le modèle probabiliste se concentre sur la modélisation de la pertinence en tant que processus aléatoire, le modèle de langue aborde la RI comme un problème de modélisation du langage [6]. Dans le modèle de langue, on suppose qu'un document est pertinent pour une requête si le document est susceptible de "générer" la requête. Cela est souvent formalisé comme la probabilité de la requête étant générée à partir d'un modèle de langue construit à partir du document [6]. De cette façon, la tâche de la RI est de classer les documents en fonction de la probabilité de la requête donnée le document, c'est-à-dire P(Q|D) [6]. L'un des principaux avantages du modèle de langue est qu'il offre une explication probabiliste naturelle de plusieurs heuristiques communes en RI,

y compris le TF-IDF et le lissage de document [7]. De plus, il fournit un cadre flexible pour incorporer des informations de requête et de document complexes [8]. En outre, alors que le modèle probabiliste de RI traditionnel nécessite un ensemble de documents pertinents pour la formation, le modèle de langue peut être appliqué sans un tel ensemble, ce qui le rend particulièrement utile pour les nouvelles requêtes où aucun document pertinent n'est connu [6]. Il est important de noter qu'il existe différentes variantes du modèle de langue, y compris le modèle unigramme, le modèle bigramme et le modèle à mélange de thèmes, qui capturent différents niveaux de dépendance entre les termes dans un document [9].

I.3 Processus de recherche d'information :

Selon Carol Kuhlthau [10], le processus de recherche d'information est caractérisé comme une suite d'activités interconnectées. Ces activités comprennent l'identification d'un besoin d'information, la localisation et l'examen de sources d'informations pertinentes, l'évaluation de l'information récoltée, et l'assimilation de ces informations pour satisfaire un besoin spécifique ou résoudre un problème donné. Kuhlthau [10] souligne que ce processus est itératif, nécessitant souvent une évaluation et une révision continues des besoins en information, des critères de recherche et des sources consultées. Ce processus, qui peut être mené de manière individuelle ou collaborative, peut mobiliser une variété d'outils et de ressources, allant des bibliothèques et des bases de données aux technologies de l'Internet, et même incluant l'interaction avec des personnes possédant des connaissances appropriées. Le processus de recherche, couramment appelés Processus en U de Recherche d'Information (voir la figure I.1) .

Ce processus est composé de trois fonctions principales :

— l'indexation des documents et des requêtes ;

— l'appariement requête-document, qui permet de comparer la requête et le document ;

— et la fonction de modification, qui intervient en réponse aux résultats obtenus.

Les modifications éventuelles concernent les documents (ajout ou suppression éventuels de la base de données) ou la requête. Les modifications les plus courantes concernent la requête seulement : pour cette raison, on parlera dans la suite de reformulation de la requête. Avant de décrire en détail ces différentes fonctions du SRI, nous allons brièvement définir les deux acteurs nécessaires à son fonctionnement, à savoir d'une part l'information disponible, c'est à dire le corpus documentaire, et d'autre part l'utilisateur et son besoin en information exprimé à travers d'une requête.

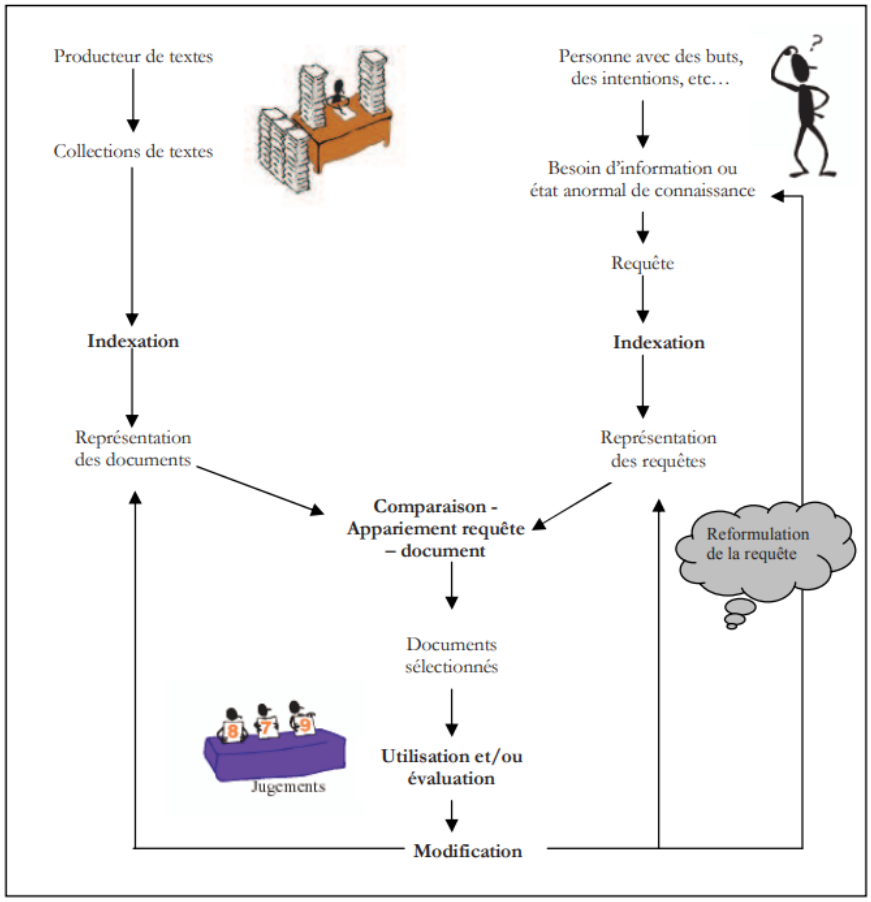

Figure I.1 — Processus de Recherche d'Information [1]

.

I.3.1 L'indexation :

Pour permettre une recherche rapide et efficace, il s'agit d'établir des index pour une collection de documents ou de données. Le processus d'indexation consiste souvent à localiser les mots essentiels, les idées ou d'autres éléments importants dans chaque document ou donnée, puis à organiser ces éléments dans une structure d'index. L'index peut être une simple liste de mots-clés avec des liens vers les documents où ils apparaissent, ou il peut avoir une structure plus complexe qui prend en compte la proximité des mots, la fréquence des mots et d'autres facteurs.

I.3.2 Le requêtage :

Il s'agit du processus de création d'une question pour communiquer un besoin de connaissances. Une requête peut être rédigée de différentes manières, par exemple à l'aide de mots-clés, d'une phrase en langage naturel, d'une image ou d'un son.En outre l'interrogation, les systèmes de recherche d'informations peuvent interpréter les requêtes en utilisant des outils de traitement du langage naturel, de reconnaissance d'images, etc.

I.3.3 L'appariement :

Le système de recherche d'information (SRI) effectue un appariement entre la requête de l'utilisateur et les documents indexés. Ce processus génère un score de pertinence pour mesurer la similarité entre la requête et le document. Les SRI actuels calculent des scores décimaux pour classer les documents. Un bon classement est essentiel, car les utilisateurs examinent généralement les premiers documents renvoyés. Différents modèles de recherche d'information, allant des modèles basés sur l'appariement exact aux modèles plus complexes basés sur l'appariement flou, ont été proposés dans la littérature [11].

I.4 Mesure de similarité

I.4.1 Fréquence des termes (TF) :

La fréquence des termes (Term Frequency, TF) est une mesure utilisée en recherche d'information et en traitement du langage naturel pour quantifier l'importance d'un terme spécifique dans un document ou une collection de documents. Elle représente simplement le nombre de fois où un terme apparaît dans un document donné [12].

I.4.2 Fréquence de documents inverse (IDF) :

La fréquence des documents inverses (IDF, pour "Inverse Document Frequency" en anglais) est une mesure utilisée en recherche d'information et en traitement automatique du langage naturel pour évaluer l'importance d'un terme dans un corpus de documents. L'IDF est calculée en prenant le logarithme inverse de la proportion de documents qui contiennent le terme, avec une pondération pour réduire l'importance des termes très fréquents [12].

Il se calcule selon la formule suivante :

$$IDF_t = \log\left(\frac{N}{d_{ft}+1}\right) \tag{I.2}$$

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.

I.4.3 TF-IDF :

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.

Il se calcule selon la formule suivante :

$$TFIDF_{t,d} = TF_{t,d} \times IDF_t \tag{I.3}$$

I.5 Présentation générale de Twitter

Sa fondation remonte à l'année 2006. Il a été créé par Jack Dorsey, Biz Stone et Evan Williams. Il est devenu rapidement l'une des plateformes de médias sociaux les plus populaires au monde. Il a connu une croissance exponentielle au fil des ans, devenant un outil essentiel pour la diffusion d'informations en temps réel, le partage d'opinions et l'engagement des utilisateurs. De nombreux évènements mondiaux majeurs ont été marqués par l'utilisation intensive de Twitter, notamment les élections, les catastrophes naturelles et les mouvements sociaux [13].

I.6 Specifications des microblogs

Dans une étude menée par Jansen et al. [14] sur le langage utilisé sur Twitter, il a été découvert que la longueur moyenne d'un tweet est d'à peine 15 mots. Cela contraste fortement avec d'autres sources d'information en ligne, comme les articles de Wikipédia qui comptent en moyenne 320 termes. Cette caractéristique de Twitter pose un défi pour les méthodes traditionnelles de recherche d'information basées principalement sur la fréquence des termes dans les documents [11].

En outre, les utilisateurs de Twitter ont développé des "normes de balisage" spécifiques qui incluent divers signes dans un tweet en plus du contenu textuel. Par exemple, le symbole "@" suivi d'un nom d'utilisateur sert à mentionner ou adresser un message à une personne spécifique. Le symbole # suivi d'un mot forme un hashtag, utilisé pour mettre en avant un mot important qui peut être utilisé pour naviguer dans la recherche, catégoriser les tweets par contexte, et suivre des événements en temps réel. Les tweets peuvent également contenir des URL, souvent sous forme raccourcie en raison de la limite de caractères de Twitter. Deux services couramment utilisés pour cela sont bit.ly et tinyurl.com. Les images peuvent aussi être insérées dans les tweets, avec un aperçu affiché sur l'interface utilisateur de Twitter.

En plus du contenu directement posté par les utilisateurs, les tweets contiennent aussi

diverses métadonnées, telles que des informations de géolocalisation (quand le tweet est posté à partir d'un appareil mobile avec GPS), un horodatage indiquant la date de publication du tweet, des informations d'auteur, des informations sur le nombre de fois qu'un tweet a été marqué comme favori et sur le nombre de fois qu'il a été retweeté [11].

I.7 Spécificités des recherches dans les microblogs

Les moteurs de recherche pour les microblogs sont uniques en termes de données d'entrée et de résultats. Un utilisateur peut combiner des mots-clés, des comptes d'utilisateurs, des hashtags et même des URL dans sa recherche. Les résultats varient selon le type de données utilisées : par exemple, si un compte utilisateur est sélectionné, le profil de ce compte sera affiché, tandis que pour les autres cas, une liste de microblogs contenant les termes, le hashtag ou l'URL recherchés sera affichée comme le montre la figure I.2 .

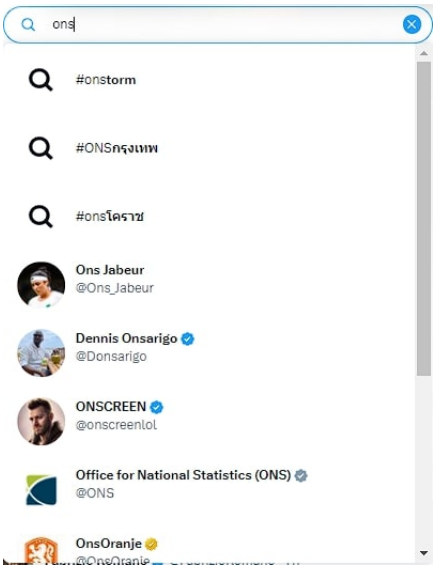

Figure I.2 — Capture d'écran de résultats suggerés dans le moteur de recherche de Twitter

Les résultats sont généralement présentés par ordre chronologique inverse, mais l'utilisateur peut choisir d'afficher tous les résultats, qui sont alors triés en fonction de leur pertinence, telle que la popularité [11].

Dans une étude réalisée par Teevan et al. [15], les motivations des utilisateurs pour rechercher des informations sur Twitter ont été analysées, ainsi que les méthodes de recherche des microblogueurs. Parmi les 54 utilisateurs actifs de Twitter observés, il a été constaté que les utilisateurs recherchent sur Twitter pour obtenir des informations récentes (49% des participants), des informations sociales (26% des participants), et des informations sur des sujets spécifiques (36% des participants). Les chercheurs ont également analysé les journaux de recherche pour identifier les différences entre les recherches

effectuées sur Twitter et celles effectuées sur les moteurs de recherche web. Ces différences se manifestent à plusieurs niveaux, notamment la longueur des requêtes, la présence de noms de célébrités, l'utilisation de hashtags, la fréquence des requêtes, et la durée des sessions de recherche .

En conclusion, les plateformes de microblogging, en particulier Twitter, représentent un nouveau type de source d'information en constante évolution, grâce à des caractéristiques spécifiques en termes de fonctionnalité et de forme. Cela a conduit à de nouveaux usages de la part des individus et des organisations [11].

I.8 Recherche d'information adhoc dans les tweets

Contrairement à la recherche traditionnelle sur les moteurs de recherche web, la recherche ad hoc dans les microblogs se concentre sur la récupération d'informations pertinentes et récentes, en tenant compte de la nature dynamique et en temps réel des messages courts.

Les microblogs présentent des caractéristiques uniques telles que la contrainte du nombre de caractères par message et la rapidité de diffusion des informations, ce qui pose des défis spécifiques pour la recherche ad hoc.Les modèles de RI classiques, qui se basent principalement sur le contenu textuel des documents et sur des statistiques des fréquences de termes, ne sont plus adaptées aux spécificités des microblogs.

Les chercheurs ont proposé différentes approches pour améliorer la pertinence des résultats de recherche dans les microblogs, notamment l'enrichissement des modèles classiques par les signaux sociaux temporels .Selon le type d'information recherchée, nous citons quelques travaux par la suite [16].

I.8.1 Recherche temps-réel de microblogs

Dans le domaine de la recherche d'information en temps réel, obtenir des données pertinentes et actuelles est primordial [17]. Généralement, ces informations nécessitent un certain temps pour être disponibles et indexées sur le web [18]. La date de publication s'avère être un critère essentiel de pertinence. La tâche peut consister à classer les documents en ordre décroissant de publication, en éliminant ceux qui sont non pertinents [17]. Diverses études ont suggéré des critères supplémentaires de pertinence, comme la fraîcheur de l'information, la popularité de l'auteur et la présence d'URLs [19]. Ces critères, lorsqu'ils sont utilisés en complément de la pertinence textuelle, ont démontré leur efficacité [20].

I.8.2 Recherche de microbloggeurs

La recherche de microbloggeurs est similaire à la recherche d'experts dans le domaine traditionnel de la recherche d'information (RI). Les objectifs principaux sont l'identification des utilisateurs les plus en vue, ceux partageant des intérêts communs avec l'utilisateur actuel ou les experts dans certains domaines spécifiques.

Des travaux comme TwitterRank [21], basé sur l'algorithme PageRank [22], mesurent l'influence des utilisateurs en fonction des scores de leurs abonnés. Un autre algorithme similaire à PageRank a été proposé par Ben Jabeur, Tamine et Al [23] , qui évalue la popularité d'un auteur dans un réseau formé par les retweets, les mentions et les réponses.

Une formule proposée par Tunkelang évalue la popularité d'un utilisateur "u" en se basant sur l'algorithme PageRank, en prenant en compte le facteur de renvoi des messages par les abonnés d'un utilisateur. En se concentrant sur les tendances de diffusion de l'information, Lee et al. [24] ont constaté que l'information est généralement diffusée le plus largement lors de sa première apparition. Par conséquent, ils ont proposé une approche basée sur l'ordre temporel de diffusion de l'information pour identifier le meilleur diffuseur.

Enfin, Cappelletti et Sastry [25] ont suggéré que l'importance d'un utilisateur devrait être dynamique dans un environnement en temps réel. Ils ont proposé un modèle qui se base sur le potentiel d'un utilisateur à amplifier la diffusion d'une information, qui change en fonction de l'évolution du réseau social de l'utilisateur.

I.8.3 Détection d'opinions

La détection des opinions est une composante clé de la recherche d'information, notamment dans le contexte des blogs et des microblogs [26] . Elle implique l'identification des documents exprimant des points de vue sur une requête spécifique, souvent en utilisant des ressources lexicales d'opinions et des techniques d'apprentissage automatique [27].

La détection d'opinion dans les microblogs a été observée comme particulièrement utile pour obtenir des réactions et des opinions immédiates sur des produits et événements [28].

I.8.4 Classification thématique des microblogs

La classification thématique des microblogs vise à filtrer les flux d'informations en identifiant les sujets discutés, permettant ainsi de classifier les utilisateurs selon leurs intérêts [29]. Plusieurs approches ont été adoptées pour atteindre cet objectif. Par exemple, Ramage et al. [30] ont utilisé l'Allocation de Dirichlet Latente pour extraire les tags caractérisant les utilisateurs et les microblogs. Song et al. [31] ont utilisé des informations spatio-temporelles pour identifier et classer des tags corrélés. Bernstein et al. [32], en

revanche, ont développé une méthode qui détecte les entités nommées dans un microblog et les soumet à un moteur de recherche pour identifier le sujet, en utilisant l'algorithme de pondération TF-IDF [33].

I.8.5 Détection de tendances

La détection de tendances dans les microblogs, comme Twitter, repère en temps réel les thèmes d'intérêt croissant parmi les utilisateurs [34]. C'est un outil précieux pour les journalistes, les analystes et les marketeurs pour suivre les sujets populaires. Plusieurs applications ont été développées pour cette fonction, dont Trendsmap et What The Trend. Certains chercheurs ont même utilisé Twitter pour alerter sur les catastrophes, comme les tremblements de terre [16], ou suivre la propagation des épidémies comme Lampos [35].

I.9 La recherche d'information temporelle

La recherche d'information temporelle dans les Tweets concerne la récupération d'informations spécifiques à des périodes temporelles précises sur la plateforme de microblogging Twitter. Cela implique de trouver des tweets pertinents, des événements ou des tendances qui se sont produits à des moments spécifiques.

Les caractéristiques temporelles uniques de Twitter, telles que les flux continus de tweets en temps réel et l'archivage des données historiques, posent des défis particuliers pour la recherche d'information temporelle. Les chercheurs ont proposé différentes approches pour relever ces défis [36] , par la suite nous discutons les travaux les plus connexes a notre travaille dans le tableau ci-dessous .

D'après les travaux cités dans le tableau I.1, on peut sortir avec les points suivants :

— La pertinence du tweet pour une requête sensible au temps est estimée via un modèle subdivisé en deux pertinences indépendantes, thématique et temporelle. Cela signifie que l'estimation de la pertinence temporelle d'un tweet pour une requête est indépendante de l'estimation de sa pertinence thématique.

— Nous subdivisons les contributions proposées selon l'aspect temporel en deux sous-classes. La première sous-classe comprend les travaux qui ont introduit la preuve temporelle « récente »dans leurs modèles temporels. La deuxième sous-classe comprend les travaux qui visaient à favoriser les tweets qui appartiennent aux rafales temporels.

— La détection du type temporelle de la requête est primordial pour connaitre les régions temporelles où se focalise les tweets pertinents.

— Plusieurs mesures sont proposées pour estimer la pertinence temporelle d'un tweet.

— Les tweets pertinents ce cluster ensemble dans les régions temporelles importantes pour la requête.

Donc, ça sera interressant de proposer un nouveau modèle qui améliore la recherche d'information dans les tweets qui se base sur l'aspect thématique et temporel.

Chercheur	**Travaux**
Efron et al [37]	— Approche de Sélection des Tweets Pertinents et Récents basée sur le profil temporel. — Méthode de recherche initiale avec la requête Q et génération de listes de tweets. — Construction du profil temporel à l'aide des travaux de Jons et Diaz. — Calcul du score final avec la concaténation du score lexical et du score temporel.
Miyanishi et al [38]	— Approche TVQE (Temporal Variation Query Expansion) pour sélectionner les meilleurs termes d'expansion des meilleurs tweets. — Estimation de la pertinence des termes basée sur la distance des profils temporels via la divergence de Kullback et Leibler. — Sélection des 10 meilleurs termes pour l'expansion de la requête.
Han et al [39]	— Études sur les caractéristiques temporelles du microblog et constat que cela peut améliorer les performances de récupération. — Proposition d'un modèle de langage de segment temporel (TSLM) pour modéliser les caractéristiques temporelles du microblog. — Utilisation de TSLM pour estimer les modèles de requête et de document, calcul de la similarité. — Résultats expérimentaux montrant que les approches surpassent plusieurs performances de lignes de base.

Table I.1 — Résumé des travaux des chercheurs dans le domaine de la recherche d'information dans les microblogs.

I.10 Evaluation

L'évaluation d'un système de recherche d'informations peut être réalisée en soumettant des questions de test et en comparant les réponses fournies par le système avec les réponses attendues.

I.10.1 Les campagnes d'évaluation :

Les campagnes d'évaluation dans la recherche d'information comprennent des compétitions organisées pour évaluer et comparer les performances des systèmes de recherche d'information. Une campagne d'évaluation bien connue est TREC (Text REtrieval Conference), qui se tient annuellement depuis 1992 et vise à stimuler la recherche dans le domaine de la recherche d'information [40].

Les campagnes TREC sont considérées comme la référence en matière d'évaluation des systèmes, mais il convient également de mentionner les campagnes CLEF [41](Cross-Language Evaluation Forum), qui se concentrent spécifiquement sur les systèmes multilingues, ainsi que les campagnes NTCIR [42].

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. Les pistes principales explorées sont le filtrage, la tâche adhoc et la tâche question-réponse [43]. La collection de test Tweets2011 utilisée dans notre travail comprend :

— 16 millions de tweets (10,5 Go) exprimés dans différentes langues et publiés sur Twitter entre le 23 janvier 2011 et le 8 février 2011.

— Elle comprend également 50 sujets de recherche, dont un exemple est présenté dans la figure suivante. La balise <titre> décrit le besoin d'information exprimé à un moment précis (querytime),querytweettime est l'identifiant du tweet recemment soumis sur le sujet.

```
<top>
<num> Number: MB002 </num>
<title> 2022 FIFA soccer  </title>
<querytime> Tue Feb 08 18:51:44 +0000 2011 </querytime>
<querytweettime> 35048150574039040 </querytweettime>
</top>
```

Figure I.3 — Exemple d'un topic pour la tâche Microblog de TREC2011.

I.10.1.1 Mesures d'évaluation

Les mesures d'évaluation dans la recherche d'information permettent de quantifier et d'évaluer la performance des systèmes de recherche d'information. Elles servent à évaluer la pertinence des résultats de recherche fournis par un système par rapport à une requête donnée [44]. Voici quelques-unes des mesures d'évaluation les plus couramment utilisées :

1. **La précision** [44] : mesure la proportion des documents pertinents parmi les documents récupérés. Elle est calculée en divisant le nombre de documents pertinents

récupérés par le nombre total de documents récupérés.

$$Prcision = \frac{|Documents\ pertinents\ restitus|}{|Documents\ restitus|} \in [0.1] \tag{I.4}$$

2. **Le rappel** [44] : mesure la proportion des documents pertinents récupérés parmi tous les documents pertinents existants. Il est calculé en divisant le nombre de documents pertinents récupérés par le nombre total de documents pertinents. Il est exprimé par :

$$Rappel = \frac{|Documents\ pertinents\ restitus|}{|Documents\ pertinents|} \in [0.1] \tag{I.5}$$

   Le rappel et la précision sont calculés sans tenir compte de l'ordre des résultats (ce sont des mesures ensemblistes). Cependant, pour évaluer des systèmes où l'ordre des documents est important, tel que les moteurs de recherche Web, des mesures prenant en compte cet ordre sont nécessaires. Deux mesures principales sont utilisées à cet égard : la **précision@X** et la **précision moyenne.**

3. La **précision@X** [44] : mesure la précision à différents niveaux de découpage de la liste de résultats. Elle évalue la proportion des documents pertinents parmi les X premiers documents récupérés par le système

4. La **précision moyenne** [44] : est calculée en prenant la moyenne des valeurs de précision après chaque document pertinent dans la liste de résultats. Elle met particulièrement l'accent sur l'évaluation du premier document pertinent classé dans les premières positions.

$$AP_q = \frac{1}{R} \sum_{i=1}^{N} p(i) \cdot R(i) \tag{I.6}$$

   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** [44] : (Mean Average Precision - MAP) est calculée pour l'ensemble des requêtes. Cette mesure calcule la moyenne des valeurs de précision moyenne non interpolées pour tous les documents pertinents. La formule suivante est utilisée pour calculer la MAP :

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP_q \tag{I.7}$$

Avec APq représente la précision moyenne d'une requête q, Q est l'ensemble des requêtes et |Q| est le nombre total de requêtes. La MAP est considérée comme une mesure globale car elle combine différents points de mesure.

I.11 Conclusion

Dans ce chapitre nous avons abordé le domaine de recherche d'information en se focalisant sur les systèmes qui utilisent Twitter comme source de données . Nous avons passé en revue la littérature existante sur la recherche temporelle dans les microblogs.Comme nous avons présenté quelques travaux connexes.

Chapitre II

CONCEPTION

II.1 Introduction

Dans ce chapitre nous allons présenter une nouvelle approche pour la recherche des Microblogs. Notre contribution consiste à proposer une technique de recherche des tweets pertinents pour une requête via l'usage des aspects thématique et temporel.

L'utilisation de l'aspect thématique se résume dans deux notions : effectuer une recherche thématique via le modèle de langue d'Indri, l'utilisation du modèle de topic pour regrouper les tweets qui partagent le même vocabulaire. L'usage de l'aspect temporel se voit dans la comparaison des profils temporels de la requête et des topics.

II.2 Description de notre approche

Dans ce chapitre, nous allons présenter une nouvelle approche pour la recherche des tweets pertinents pour une requête Q . Notre hypothèse est : « les tweets pertinent pour une requête se regroupent ensemble dans l'espace de termes et possèdent les mêmes caractéristiques temporelles que la requête ». Notre intuition est de faire une évaluation de pertinence par groupe de tweets au lieu de faire une evaluation individuelle de tweet. Il s'agit alors de chercher le groupe de tweets le plus pertinent puis considérer les tweets de ce groupe comme meilleur résultat de recherche pour la requête. Pour la réalisation de notre approche nous avons proposé l'architecture détaillée dans figureII.1.

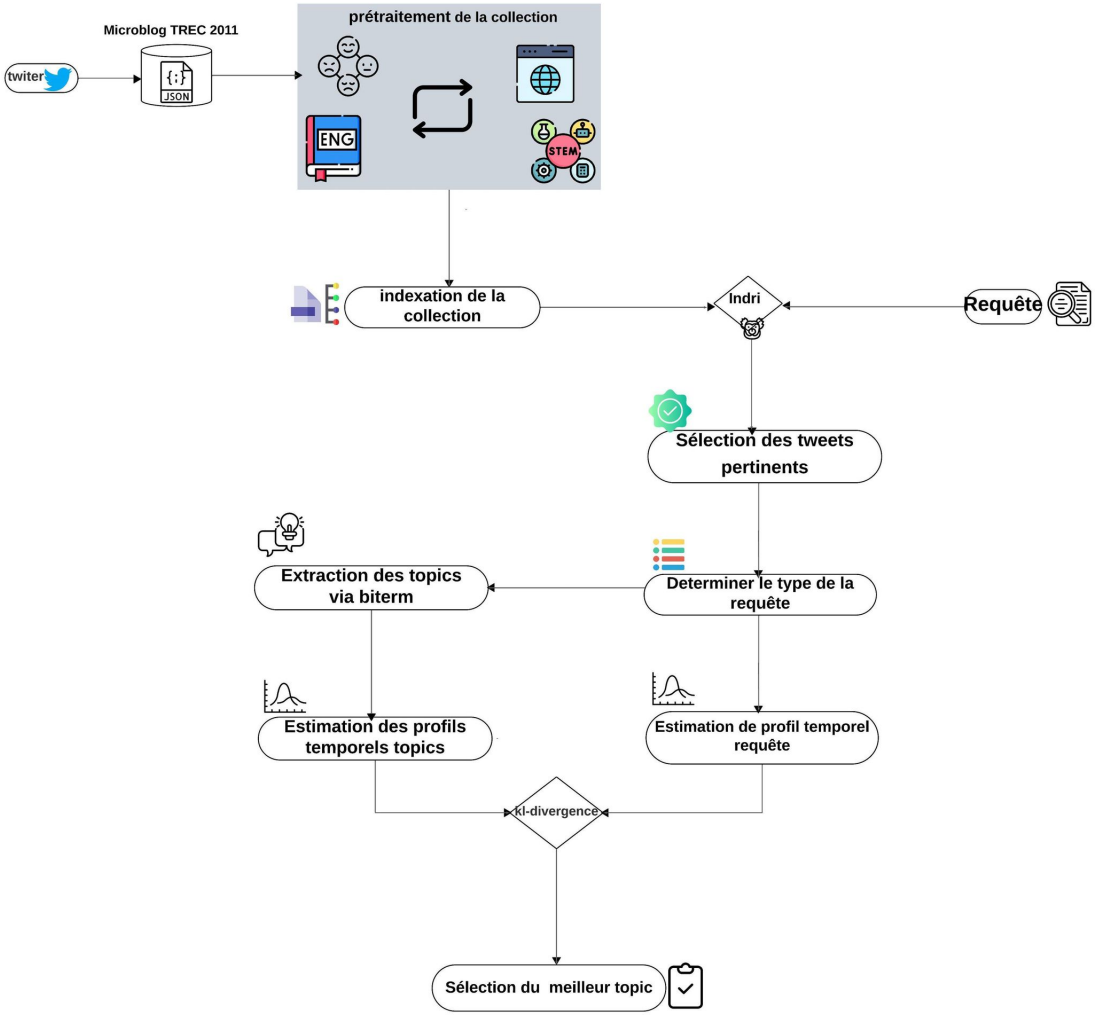

Figure II.1 — Shéma recapitulatif de notre travail.

II.2.1 Collection de données

Pour évaluer notre approche nous avons utilisé le Corpus Tweets2011 de la compagne TREC. Il se compose d'environ 16 millions de tweets discutent des évènements qui ont déroulé entre le 23 janvier 2011 et le 8 février 2011. TREC a fourni trois outils pour le téléchargement du Corpus :

— Le premier outil utilise l'API "Twitter Tools"[1], il est très lent. Il peut explorer environ 150 tweets par heure. Il collecte les tweets sous format JSON. Ces fichiers sont riches en informations puisqu'ils contiennent le profil de l'utilisateur Twitter.

— Le deuxième outil explore uniquement les pages HTML des tweets. Il est plus rapide que le premier outil. Son inconvénient réside dans l'absence des informations sociales dans les fichiers HTML collectés.

— L'API utilisée pour l'évaluation en tant que « service », a été mise en œuvre par

1. `https://github.com/lintool/twitter-tools`

Lin(2013) [2]. Cette API est plus efficace que les outils précédents. Il permet de télécharger les informations souhaitées (nombre de tweets, contenu) en peu de temps.

II.2.2 Prétraitement :

Pour effectuer une recherche efficace des microblogs, il est crucial de prétraiter ces données afin de les transformer en un format plus facile à gérer et plus utile pour la recherche. Le corpus microblog TREC 2011 nécessite un prétraitement minutieux pour une meilleure restitution des tweets pertinents.

1. **Élimination de tweets vides, non anglais et de retweets**

   Cette étape vise à filtrer les tweets de manière à ne conserver que les tweets écrit en anglais, en supprimant les tweets vides et les retweets.

   Par conséquent, après cette étape, nous avons un ensemble de tweets nettoyés qui sont tous écrits en anglais, non vides et ne sont pas des retweets.

2. **Remplacement des URLs par des mots-clés**

   Dans cette étape on a suivi plusieurs étapes pour extraire les URLs à partir des tweets, obtenir les mots clés de ces URLs, puis remplacer les URLs dans les tweets par leurs mots clés correspondants :

   — **extraction des URLs à partir des tweets :** identifier toutes les URLs dans les tweets à l'aide d'une expression régulière (re). Les URLs extraites sont ensuite stockées dans un dictionnaire, où chaque clé est un ID de tweet et chaque valeur est une liste des URLs dans ce tweet.

   — **Obtention des mots-clés des URLs :** nous avons utilisé les bibliothèques 'requests' et 'BeautifulSoup' pour extraire les mots-clés de chaque URL. Pour chaque URL, nous avons extrait les mots-clés à partir de la balise <meta name=”keywords”> et <title>.

   — **Remplacement des URLs par les mots clés dans les tweets :** enfin, nous avons remplacé chaque URL dans les tweets par les mots-clés correspondants en utilisant le dictionnaire que nous avons créé.

   Cette étape nous a permis de remplacer les URLs dans les tweets par leurs mots-clés pertinents, ce qui pourrait enrichir le contenu informationnel de chaque tweet.

3. **Élimination des mots d'arrêt (stop words)**

   Cette étape comprend l'élimination des mots d'arrêt (stop words) et des mentions

2. `https://github.com/lintool/twitter-tools`

(@). Pour cela nous avons utilisé la bibliothèque NLTK [3] (Natural Language Toolkit), qui est un outil populaire en Python pour le traitement du langage naturel :

- **elimination des mots d'arrêt :** les mots d'arrêt sont des mots communs qui n'apportent généralement pas beaucoup d'information pour l'analyse de texte, par exemple "the", "is", "at", "which", et ainsi de suite. NLTK fournit une liste de mots d'arrêt en anglais que nous avons utilisé pour filtrer ces termes dans les tweets. nous avons divisé chaque tweet en mots puis gardé uniquement ce qui ne figure pas dans la liste.
- **elimination des mentions :** les tweets contiennent souvent des mentions d'autres utilisateurs (par exemple, @username). Ces mentions ont été éliminées en utilisant une expression régulière qui correspond à la syntaxe d'une mention Twitter.

4. **Remplacement des abréviations :** les abréviations sont couramment utilisées sur Twitter en raison de la limite du nombre de caractères pour écrire un tweet. Nous avons remplacé les abréviations par leurs formes complètes via les étapes suivantes :
   - **lecture des abréviations :** nous avons conçu une liste d'abréviations dans un fichier texte, chaque ligne contenant une signification et ses differentes abbréviations .
   - **modification des tweets :** nous avons remplacé chaque abbréviations par ça signification via l'usage du module 'RE' et le fichier des abbréviations. Cette étape aide à normaliser le texte d'un tweet en remplaçant les abréviations par leurs significations, ce qui pourrait faciliter l'analyse du texte et améliorer la qualité des résultats de la recherche de tweets .
5. **Stemming :**le stemming est le processus de réduction des mots à leur forme de racine, nous avons utilisé la fonction *apply_stemming* qui utilise a son tour l'algorithme de Porter Stemmer de NLTK , pour réduire chaque mot d'un tweet à sa forme racine.
6. **Traitement des hashtags :** Les hashtags sont extraits du texte du tweet à l'aide des informations d'entités du tweet. Ensuite, chaque hashtag est segmenté en mots individuels à l'aide de la bibliothèque "wordsegment". Par exemple, le hashtag "# WorldCup2022" serait segmenté en "World" et "Cup2022". Les segments obtenus sont ensuite remplacés dans le texte d'origine du tweet, conservant ainsi la structure générale du tweet tout en segmentant les hashtags. Ce processus permet d'enrichir le texte des tweets avec des informations supplementaires pour une meilleure lisibilité du contenu des tweets.
7. **Traitement des emoticônes et des ponctuation :** nous avons supprimé les émoticones et les ponctuations .

3. `https://github.com/nltk/nltk`

En conclusion, cette étape de prétraitement a permis de réduire le bruit et à normaliser le texte des tweets, ce qui pourrait améliorer la qualité des informations que nous pouvons obtenir de chaque tweet pour les étapes suivantes.

II.2.3 Indexation

Nous avons choisi le moteur de recherche Indri pour effectuer l'indexation et la recherche de tweets pertinents dans le corpus microblog TREC 2011 prétraité. Indri est un moteur puissant de recherche du texte, qui fait partie de la famille des moteurs de recherche du lemur Toolkit [4].

Le moteur de recherche Indri est capable d'indexer de grandes collections de documents et de gérer des requêtes complexes avec des opérateurs booléens [45]. Il supporte également plusieurs modèles de récupération d'information comme le modèle vectoriel et probabiliste [46].

Dans notre travail nous avons utilisé la commande suivante pour indexer le corpus :

IndriBuildIndex.exe [chemin vers le fichier de paramètre xml]

II.2.4 Recherche des tweets

La conférence TREC 2011 microblogs trek a proposé pour ses participantes 50 requêtes "topics" sur un corpus de 16 millions de tweets. Chaque requête se compose d'un numéro, du texte (la requête elle-même), d'une date de soumission et d'un ID du tweet le plus proche temporellement à la requête. Voici un exemple de requête :

```
<top>
<num> Number: MB002 </num>
<title> 2022 FIFA soccer  </title>
<querytime> Tue Feb 08 18:51:44 +0000 2011 </querytime>
<querytweettime> 35048150574039040 </querytweettime>
</top>
```

Figure II.2 — Exemple d'une requete pour la tâche Microblog de TREC2011.

Nous avons utilisé Indri pour récupérer les 1000 tweets les plus pertinents thématiquement pour chaque requête. Ces derniers ont été publiés avant la soumission de la requête.

La fonction de recherche d'Indri est basé sur le modèle de langue, qui est une approche statistique pour la récupération d'information. L'idée principale du modèle de langue est d'évaluer la pertinence d'un document à une requête en évaluant la probabilité que le document génère la requête. Formellement, la probabilité d'une requête Q étant générée

4. `https://www.lemurproject.org/indri.php`

par un document D dans le modèle de langue d'Indri est exprimée comme suit :

$$P(Q|D) = P(qi|D) \; pour \; i \; allant \; de \; 1 \;\; n \tag{II.1}$$

où n est le nombre de termes dans la requête et P(qi|D) est la probabilité d'occurrence du terme qi dans le document D. Cette probabilité est généralement calculée en utilisant le lissage de Dirichlet ou le lissage de Jelinek-Mercer [47].

La commande qui permet de faire une recherche avec indri est la suivante :

IndriRunQuery.exe [chemin vers la requête xml] -index=[chemin vers l'index] -count=[nombre de tweets a retourné]

II.2.5 Typage de requêtes

Cette étape consiste a classifier les requêtes sur la base du moment où les tweets ont été postés par rapport a la date de la requête , Nous distingons deux classes : insensibles au temps, sensibles au temps. Les requêtes insensibles au temps se caractérisent par la non-sensibilité à la variation temporelle des dates de soumission des tweets.Par exemple, une requête comme "NSA" serait insensible au temps, car les informations pertinentes pour cette requête ne changeraient pas beaucoup au cours des temps. Les requêtes sensibles au temps ce sont les requêtes où les tweet les plus pertinents sont ceux publiées très récemment (pour la date de soumission d'une requête) ou publiées autour d'une certaine date ou d'un certain évènement.Par exemple, une requête comme "Special Olympics athletes" ou une requête comme "BBC World Service staff cuts " sont du type sensible au temps.

Pour élaborer notre algorithme de classement des requêtes nous nous sommes inespérés du travail, ce dernier propose une classification différente des requêtes (non sensible au temps, un seul pic dominant et plusieurs pics). Nous avons calculé la proportion de tweets publiés chaque jour par rapport au nombre total de tweets résultat de la première recherche (Pr = (NB tweets par jour/NB total des tweets)). Ensuite nous avons établi un seuil (p). En fonction de ce seuil et en fonction des proportions calculées, nous avons déterminé le type de la requête :

— **Les requêtes insensibles au temps :** si la plus grande proportion de tweets publiés en un jour (max Pr) est inférieure ou égale à un certain seuil p (défini dans le code comme 0.05), alors la requête est considérée comme "insensible au temps". Cela signifie qu'il n'y a pas de forte concentration de tweets à un moment spécifique, indiquant que la requête n'est pas liée à un événement ou une actualité particulière.[voir figII.3]

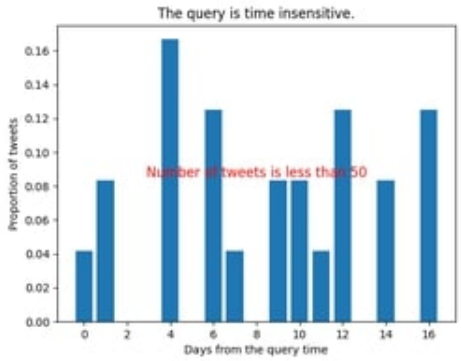

Figure II.3 — Exemple d'une requete insensible au temps.

— **Les requêtes sensible au temps :** si le jour avec la plus grande proportion de tweets supérieur a p (Max Pr > p), alors la requête est considérée comme "du type sensible au temps ". Cela indique que la plupart des tweets liés à la requête sont très récents ou autour d'une certaine date d'un certain événement.[voir fig II.4].

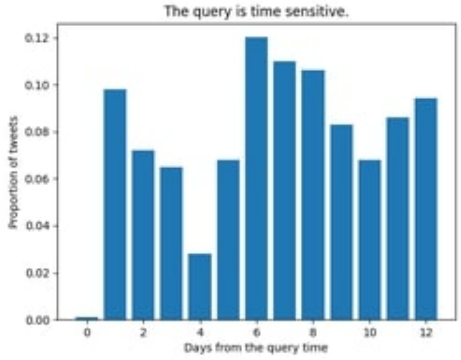

Figure II.4 — Exemple d'une requete sensible au temps.

II.2.6 Application de Topical Modal « BTM »

Dans notre approche de recherche d'information dans les microblogs, nous avons exploité le Biterm Topic Model (BTM), un modèle thématique adapté aux courts textes tels que les tweets. Contrairement à d'autres modèles comme le Latent Dirichlet Allocation (LDA), BTM identifie des "bitermes" dans le texte, formés de paires de mots co-occurrents dans le même contexte de court texte. Cette approche efficace permet d'extraire des thèmes pertinents des courts textes en capturant les co-occurrences de mots dans l'ensemble du corpus.

Le processus d'entraînement du modèle BTM consiste à apprendre quels mots co-occurrent fréquemment, permettant l'identification de topics communs. L'apprentissage se fait par itérations, où le modèle attribue probabilistiquement chaque biterme à un topic et met à jour les distributions de probabilité pour chaque topic. Une fois le modèle

entraîné, nous utilisons la méthode "transform" pour transformer les bitermes en topics, obtenant ainsi des distributions de probabilités sur les thèmes pour chaque tweet.

Après la génération des topics à partir des résultats d'une requête, nous passons à la création des profils temporels pour la requête ainsi que pour les trois topics, complétant ainsi notre approche intégrée de recherche dans les microblogs.

II.2.7 Estimation des profils temporels

Un profil temporel illustre comment la quantité ou la fréquence de quelque chose change avec le temps. Dans notre cas, cela montre comment la fréquence des tweets varie au fil du temps. Le profil temporel est souvent utilisé dans la recherche d'information pour comprendre l'évolution temporelle des sujets de recherche, ce qui peut être particulièrement utile pour les requêtes sensibles au temps [48].

La densité de noyau permet d'estimer la fonction de densité de probabilité d'une variable aléatoire continue pour obtenir une courbe fluide qui reflète la densité de tweets à différents moments. Les sommets de la courbe correspondent aux périodes où un volume plus conséquent de tweets a été publié. Cette courbe est centrée sur chaque point de données, et la somme de ces courbes est utilisée pour obtenir l'estimation finale de la densité. Dans notre cas les données sont les dates de publications des tweets [48].

Dans notre contexte, nous avons utilisé la densité de noyau gaussien [49] pour générer le profil temporel de la requête et des topics. Pour chaque tweet, nous avons converti le timestamp en une représentation numérique, et puis utiliser ces timestamps pour calculer la densité de noyau gaussien, La formule générale de la densité du noyau est la suivante :

$$f(x) = \frac{1}{n \cdot h} \sum K\left(\frac{x - x_i}{h}\right) \tag{II.2}$$

où :

- x est la date de publication du tweet t, sa densité est calculée via la formule.
- n est le nombre total de points de données, elle a la valeur :
  - Le nombre total des tweets, dans l'estimation du profil temporel d'une requête.
  - Le nombre de tweets d'un topic, dans l'estimation du profil temporel d'un topic.
- h est la largeur de bande (un paramètre qui contrôle le lissage de l'estimation fixé à 0,25) .
- xi sont les points de données (dans notre cas, les timestamps)
- K est la fonction du noyau (dans notre cas, une fonction gaussienne)

Après le calcul de la densité pour chaque tweet, nous obtiennent un profil temporel pour la requête et des profils pour les topics (le schéma II.5 illustre un exemple de profil temporel

pour la requête et les topics)

II.2.8 Séléction des meilleurs topics « aspect temporel »

Après la génération du profil temporel de la requête et les trois profils des topics via la densité de kernel. La prochaine étape est de choisir un seul topic en se basent sur sa pertinence temporel pour la requête. Nous avons opté pour la divergence KL nommé aussi Kullback-Leibler (KL) [50], pour estimer la divergence entre le couple de profils temporels(requête, topic). Nous l'avons utilisé pour comparer les deux distributions de tweets et mesurer à quel point elles sont différentes l'une de l'autre.

KL mesure aussi la différence entre deux distributions de probabilité. Elle donne une mesure de la quantité d'information perdue lorsqu'on utilise une distribution de probabilité pour approximer une autre.

La (KL) pour deux distributions de probabilité PT et PQ est définie comme suit :

$$KL(P\|Q) = \sum P(i) \log \left(\frac{P(i)}{Q(i)} \right) \tag{II.3}$$

où la somme est sur toutes les valeurs possibles de i, et P(i) et Q(i) sont les probabilités de ces valeurs selon les distributions P et Q, respectivement. Dans notre cas, P serait la distribution de la densité des profils temporels des topics et Q serait la distribution de la densité du profil temporel d'une requête.

En fin, le topic dont le profil temporel est le plus proche à celui de la requête (c'est-à-dire qui a la plus petite divergence KL) est sélectionné comme meilleur topic. Ses tweets représentant la liste finale des résultats de la recherche par Q, ils sont classé par leur probabilité d'appartenance à ce topic.

Dans la figure II.5 nous présentons le profil temporel de la requête 09, et le profil temporel des topics (T0, T1, T2). Nous avons sélectionné topic2, vu que son profil est le plus proche à celui de la requête. Nous avons considéré ses tweets comme les plus pertinents .

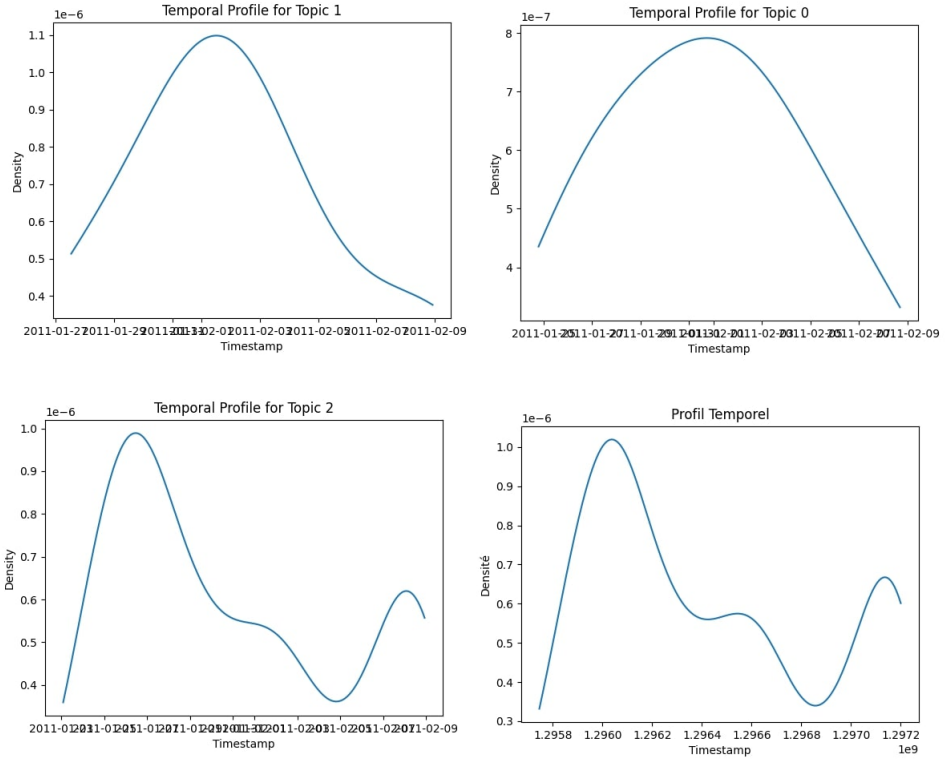

Figure II.5 — Exemple d'un profile temporel de la requete 9 et ses 3 topics.

II.3 Conclusion

Dans ce chapitre nous avons décrit notre approche de recherche des tweets pertinents, suite à une requête utilisateur. L'objectif final est de sélectionner le meilleur groupe de tweets pertinents qui partagent le même vocabulaire et possède les mêmes caractéristiques temporels que la requête . Notre modélisation se base sur l'aspect thématique et temporel.

Chapitre III

TESTS ET IMPLÉMENTATION

III.1 Introduction

Ce chapitre décrit les outils et logiciels utilisés pour notre projet, ainsi que les tests effectués et les fonctionnalités de notre interface utilisateur.

III.2 L'environnement de développement

III.2.1 Python

Python [1] est un langage de programmation polyvalent, facile à lire et à apprendre. Il est utilisé dans de nombreux domaines, tels que le développement web, l'analyse de données, l'intelligence artificielle et l'automatisation de tâches. Python se distingue par sa syntaxe claire et sa large bibliothèque standard qui facilite le développement d'applications. Il est également apprécié pour sa capacité à s'adapter à différents paradigmes de programmation. En résumé, Python est un langage puissant et populaire offrant de nombreuses possibilités aux développeurs.

III.2.2 Visual Studio Code

Visual Studio Code [2] est un éditeur de code source qui peut être utilisé avec une variété de langages de programmation, notamment Java, JavaScript, Go, Node.js et C++. Il est basé sur le cadre Electron, qui est utilisé pour développer des applications Web Node.js qui s'exécutent sur le moteur de présentation Blink. Visual Studio Code utilise le même composant d'éditeur (nom de code Monaco) utilisé dans Azure DevOps (anciennement appelé Visual Studio Online et Visual Studio Team Services). Le logiciel prend en charge le Windows Subsystem for Linux et, permet ainsi par exemple, de programmer facilement en C/C++ depuis un ordinateur Windows 10.

III.2.3 Linux

Linux [3] est un système d'exploitation open source créé par Linus Torvalds en 1991 basé sur le noyau Linux, offrant une alternative fiable, sécurisée et flexible pour les utilisateurs. Il est utilisé dans une variété de domaines, des serveurs d'entreprise aux smartphones, en raison de sa stabilité, de sa sécurité et de sa grande compatibilité matérielle. Grâce à sa nature open source, Linux bénéficie d'une communauté active de développeurs qui contribuent à son amélioration continue et à l'expansion de son écosystème de logiciels.

1. `https://docs.python.org/3/reference/index.html`
2. `https://code.visualstudio.com/`
3. `https://www.britannica.com/technology/Linux`

III.3 Les bibliothèques

III.3.1 JSON

"Le module **json** [4] fournit des fonctionnalités pour travailler avec le format de données JSON (JavaScript Object Notation). Il permet de sérialiser (encoder) des objets Python en JSON et de désérialiser (décoder) des données JSON en objets Python. Le module json prend en charge la conversion des types de base Python (tels que les dictionnaires, les listes, les chaînes, les nombres, les booléens) en JSON et vice versa. Il offre également des fonctionnalités pour personnaliser la sérialisation et la désérialisation en utilisant des encodeurs et des décodeurs personnalisés."

III.3.2 NLTK

NLTK [5] est une bibliothèque Python open source destinée à l'éducation et à la recherche en traitement automatique du langage naturel, avec des interfaces pour plus de 50 corpus et ressources lexicales telles que WordNet. Il est accompagné d'une suite de didacticiels permettant aux étudiants de commencer rapidement à travailler avec des données linguistiques, d'acquérir de l'expérience pratique en matière de méthodes de traitement du langage, et d'accompagner les cours en ligne courants sur le traitement du langage naturel."

III.3.3 BTM

"Le modèle **B**iterm **T**opic **M**odel (BTM) est un modèle de traitement automatique du langage naturel (NLP) qui permet de découvrir des sujets à partir de documents textuels non supervisés. Contrairement à d'autres modèles de sujets qui considèrent les documents comme des collections de mots, le BTM se concentre sur les co-occurrences de paires de mots appelées "bitermes". Il modélise la génération de bitermes à partir de sujets latents et est capable d'inférer les distributions de sujets pour chaque document ainsi que les distributions de mots pour chaque sujet". [51]

III.3.4 LangDetect

La bibliothèque langdetect [6] est un détecteur automatique de langues écrit en Java. Elle fournit une interface simple pour détecter la langue d'un texte donné. La bibliothèque utilise des profils de langues préalablement construits pour effectuer la détection de langues

4. `https://docs.python.org/3/library/json.html`
5. `https://www.nltk.org/`
6. `https://github.com/shuyo/language-detection`

basée sur des modèles statistiques. Elle prend en charge de nombreux langages courants et est largement utilisée dans diverses applications de traitement de texte et de traitement automatique du langage naturel.

III.3.5 Requests

Requests [7] est une bibliothèque Python simple et élégante qui permet de réaliser facilement des requêtes HTTP. Elle simplifie les tâches liées à l'envoi de requêtes HTTP, telles que la gestion des en-têtes, des paramètres, des cookies et des sessions. Requests permet également de gérer les réponses HTTP, y compris l'accès aux en-têtes, au contenu et à d'autres informations. Elle est largement utilisée pour l'interaction avec des API Web et le scraping de contenu à partir de pages Web.

III.3.6 BeautifulSoup

BeautifulSoup [8] est une bibliothèque Python permettant de parcourir et de manipuler facilement des documents HTML et XML. Elle fournit des méthodes puissantes pour extraire des informations spécifiques à partir de pages Web en utilisant des techniques de parsing et de recherche souples. BeautifulSoup facilite la navigation dans la structure du document, l'extraction de balises, d'attributs et de contenu, et la recherche basée sur des critères spécifiques. Elle est couramment utilisée pour le web scraping et le traitement de données HTML/XML.

III.3.7 Regex

La bibliothèque regex [9] est une alternative avancée au module intégré 're' de Python pour les expressions régulières. Elle offre une prise en charge étendue des fonctionnalités d'expressions régulières, y compris les caractères Unicode, les opérations de recherche et de remplacement avancées, les captures nommées, les lookaheads et lookbehinds, les groupes imbriqués et bien plus encore. La bibliothèque regex est réputée pour sa compatibilité avec les spécifications officielles de l'expression régulière, sa rapidité d'exécution et sa facilité d'utilisation.

III.3.8 Tkinter

Tkinter [10] est une bibliothèque Python pour le développement d'interfaces graphiques utilisateur (GUI). Elle fournit un ensemble d'outils pour la création de fenêtres, de bou-

7. `https://requests.readthedocs.io/en/latest/`
8. `https://www.crummy.com/software/BeautifulSoup/bs4/doc/`
9. `https://pypi.org/project/regex/`
10. `https://pypi.org/project/regex/`

tons, de menus, de zones de texte et d'autres éléments d'interface utilisateur.

Tkinter est une enveloppe de la bibliothèque Tk, qui est un multiplateforme standard de Tcl/Tk, un langage de script populaire et un ensemble d'outils de développement d'interfaces utilisateur. Python, avec Tkinter, peut être utilisé pour développer des applications de bureau multiplateformes.

III.3.9 WordSegment

WordSegment [11] est une bibliothèque Python pour la segmentation de mots, généralement utilisée pour séparer des chaînes de caractères en mots individuels. Elle est particulièrement utile pour le traitement de textes qui ne contiennent pas d'espaces, comme les hashtags ou les URL. WordSegment est basé sur un corpus d'un million de phrases et environ 250 000 mots anglais uniques pour fournir des résultats précis.

III.4 Présentation de notre application

Une fois l'application lancée, une interface épurée se présente, illustrant la description de notre application (voir figure III.1).

1. Cette interface propose trois boutons, détaillés dans la section suivante :
   — Typage
   — Profil de la requête
   — Biterm
   — Best-t-info
   — Best tweets

   À noter que les boutons "Profil de la requête" et "Biterm" sont inactifs au départ.

11. `https://grantjenks.com/docs/wordsegment/`

Figure III.1 — Capture d'écran de l'interface générale

2. Lors de l'exécution de l'application, une liste déroulante apparaît pour choisir une requête. Une fois la requête choisi, en cliquant sur le bouton "typage" l'algorithme de typage commence à déterminer son type et affiche en fin de processus son type et son numéro. (voir figure III.2)

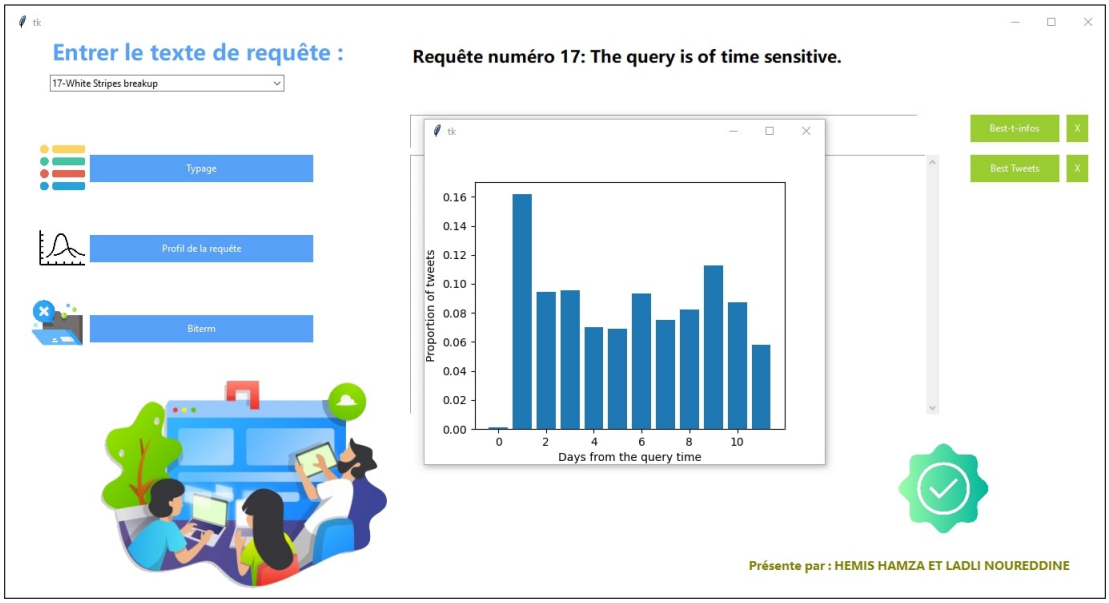

Figure III.2 — Capture d'écran de l'interface de typage

3. Après le typage de la requête, le bouton "Profil de la requête" sera actif en cliquant sur lui , l'algorithme extrait les tweets, génère et affiche un profil temporel, puis stocke ces informations dans un fichier JSON pour une utilisation ultérieure.(voir la figure III.3

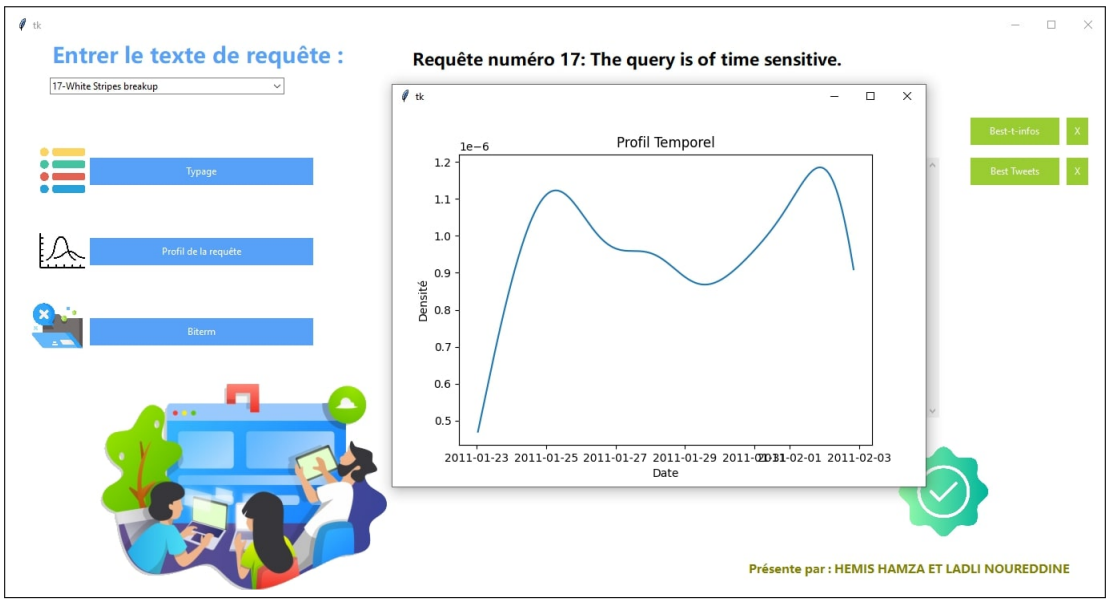

Figure III.3 — Capture d'écran de l'interface de profil temporel

4. Lorsqu'on clique sur "Biterm" , l'algorithme attribue des thèmes aux tweets, identifie le topic le mieux adapté en fonction du profil temporel, sauvegarde les tweets correspondants dans un fichier JSON et affiche le profil temporel de chaque topic.(voir figure III.4)

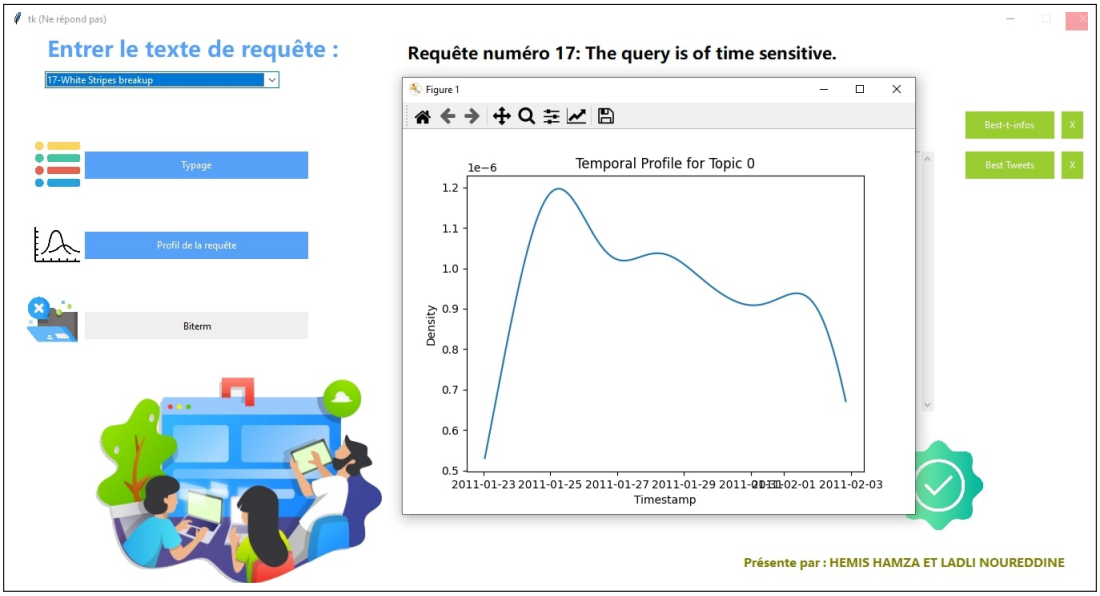

Figure III.4 — Capture d'écran de l'interface de biterm

5. En cliquant sur le bouton " Best-t-info " un message s'affiche indiquant le numero du best topic ainsi que son KL divergeance. [voir figure III.5]

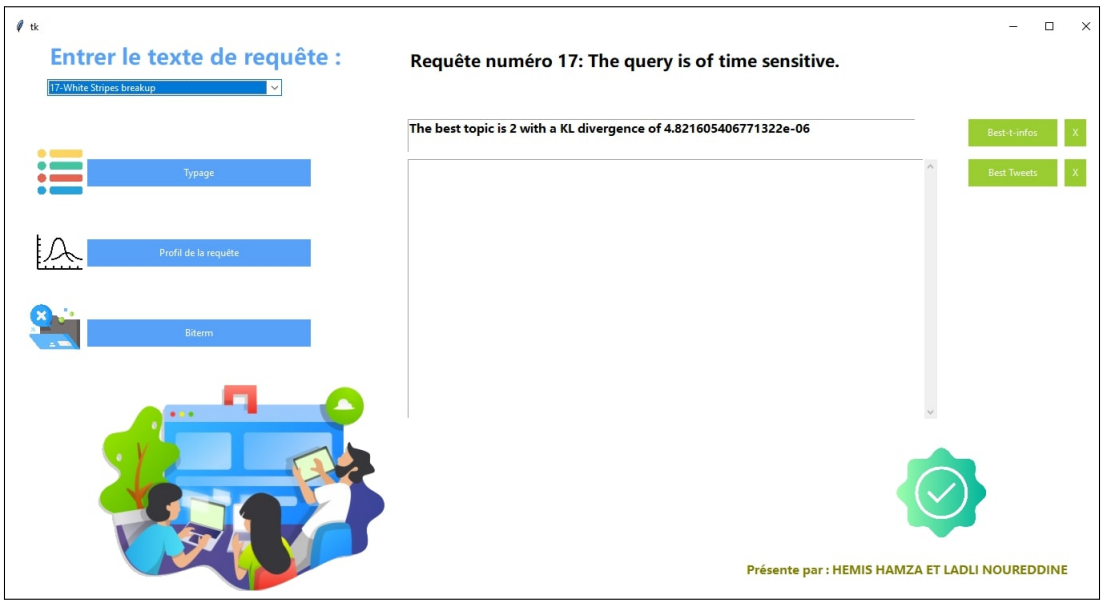

Figure III.5 — Capture d'écran d'information du best topic

6. le bouton "best tweets" affiche les resultats enregistrés après l'execution du biterm .[voir figure III.6]

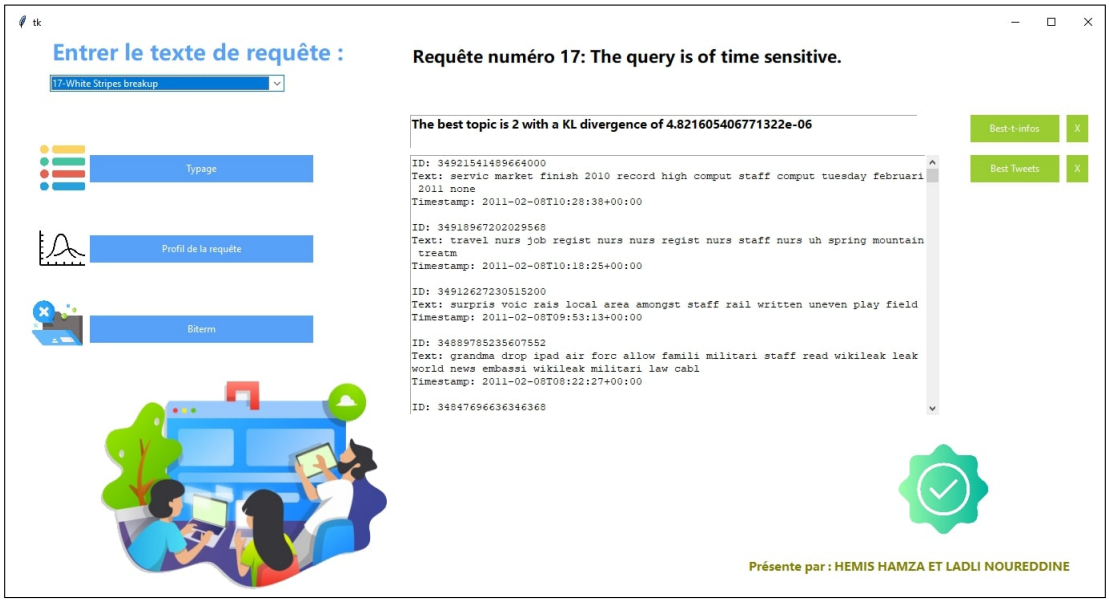

Figure III.6 — Capture d'ecran de visualisation de tweets du best topic

III.5 Évaluation des résultats

Trec_eval [12] est un outil utilisé pour évaluer les classements, soit des documents, soit toute autre information triée par pertinence. L'évaluation est basée sur deux fichiers : le premier, connue sous le nom de "qrels" (informations de requête) énumère les jugements de pertinence pour chaque requête. Le deuxième contient le classement des documents

12. `https://github.com/usnistgov/trec_eval`

retournés par notre système RI.

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

— Trec_eval : c'est le nom du programme exécutable.
— q : donner une évaluation pour chaque requête / sujet
— Qrel_file : chemin du fichier avec la liste des documents pertinents pour chaque requête
— m : montre une mesure spécifique ("-m all_trec" montre toutes les mesures, "-m official" est le paramètre par défaut qui ne montre que les principales mesures)
— Result_file : chemin du fichier avec la liste des documents récupérés par notre application.

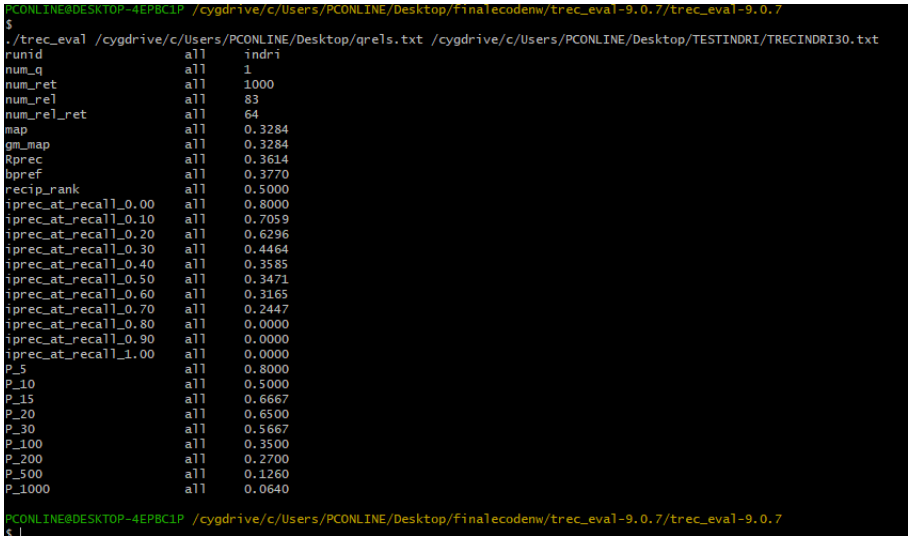

Figure III.7 — Capture d'écran d'un exemple de test

III.6 Discussion des résultats de test

Nous avons etudié dans notre approche 25 requetes qui sont tous de type sensible au temps sauf les requetes 5 et 6 sont insensible, voici ci-dessous les resultats de tests obtenus selon les scores MAP et P@30 .

P@30, dans le domaine de la recherche d'information dans les microblogs, est une métrique d'évaluation des résultats de recherche. Le "P" signifie "précision", et le "30" indique que la métrique mesure la précision des 30 premiers résultats retournés par un système de recherche.

Plus précisément, P@30 est calculé en prenant le nombre de documents pertinents parmi les 30 premiers résultats et en le divisant par 30. Cela donne une mesure de la proportion de résultats pertinents parmi les 30 premiers retours du système de recherche.

Cette métrique est souvent utilisée dans le contexte des microblogs, comme Twitter, où les requêtes peuvent être courtes et les résultats doivent être pertinents et rapidement accessibles à l'utilisateur. P@30 permet d'évaluer la qualité des résultats dans un contexte où la rapidité et la pertinence sont cruciales.

Requete	MAP	P@30	Requete	MAP	P@30
1	0,0578	0,1667	28	0,0769	0,0333
2	0,159	0,2333	29	0,1841	0,4667
3	0,26	0,3333	30	0,3698	0,6
4	0,1985	0,6333	31	0,2508	0,4506
7	0,2262	0,7333	32	0,0159	0
8	0,0096	0,0333	33	0,2108	0,4106
9	0,1845	0,3	34	0,1673	0,4
10	0,2304	0,3667	37	0,056	0,2
11	0,262	0,583	42	0,1539	0,3
17	0,0941	0,1333	44	0,312	0,683
20	0,2443	0,8	48	0,282	0,603
21	0,4124	0,8333	49	0,3336	0,6056
25	0,0093	0,0667			

Table III.1 — Score des 25 requetes etudiées.

La table III.2 résume les valeurs des deux critères (précision au rang 30 et map) issues de l'évaluation des résultats de test de cinq requete de l'algorithme étudié. Nous remarquons que les résultats obtenus sont quasiment meilleurs que ceux de indri . Cependant, ce qui est plus intéressant, c'est qu'on a pu augmenter la mesure (P@30 et map) d'une façon générale dans notre contribution.

Requete	Indri		Notre approche	
	Map	P@30	Map	P@30
21	0.41	0.83	0.41	0.83
42	0.15	0.33	0.15	0.30
30	0.33	0.57	0.37	0.60
09	0.17	0.23	0.18	0.30
04	0.19	0.57	0.20	0.63

Table III.2 — Résultats de test de Indri et nos résultats sur quelque requetes

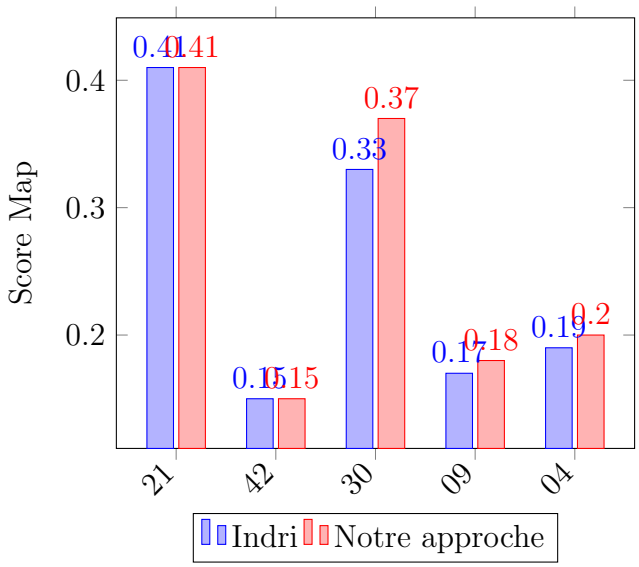

Figure III.8 — Comparaison des résultats du tableau III.2

	HEMIS2023	ECIR2013	GHELLAL2023	HAN2021	INDRI	Lucene baseline
MAP	**0.1904**	0.2741	0.1994	0.4208	0.1280	0.1413
P@30	**0.4253**	0.4830	0.3167	0.4769	0.2381	0.1007

Table III.3 — MAP et P@30 des traveaux voisins

III.7 Comparaison avec travaux voisins

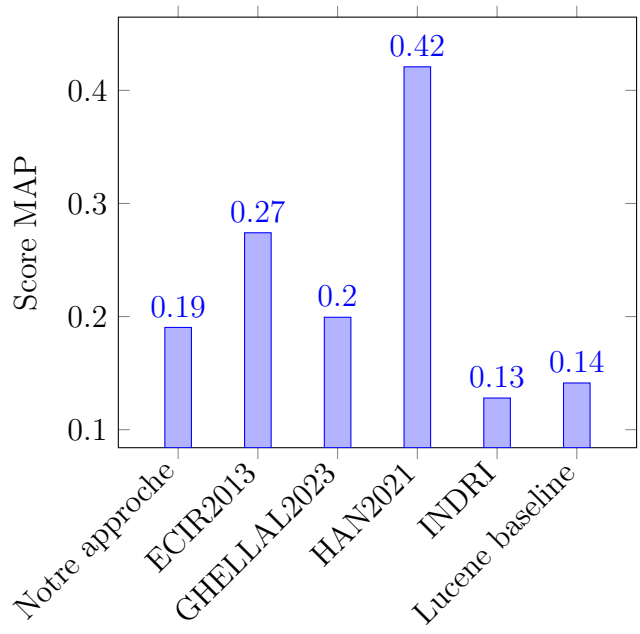

Figure III.9 — Score MAP des travaux voisins

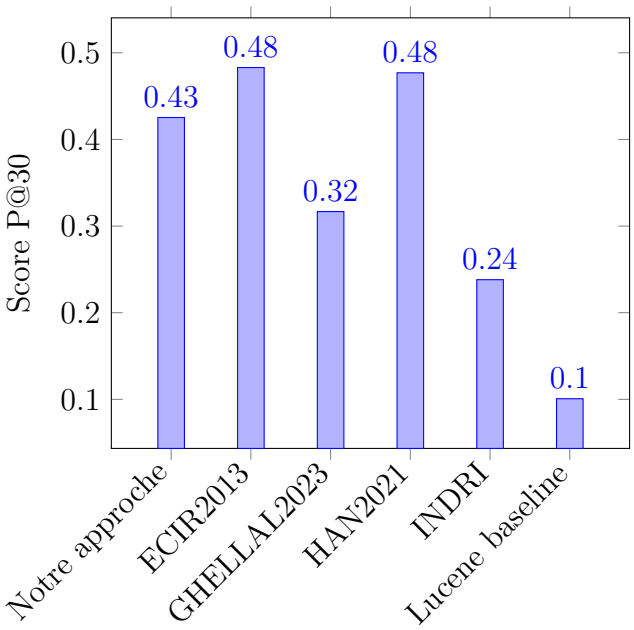

Figure III.10 — Score P@30 des travaux voisins

Selon le tableau présenté III.3, notre approche a démontré des améliorations significatives en termes de performance par rapport à certaines des méthodes comparées.

1. Notre méthode dépasse "INDRI" et "Lucene baseline" en précision moyenne (MAP) et précision à 30 (P@30), indiquant une récupération plus efficace des résultats.
2. Face à "GHELLAL2023", nous observons une légère amélioration de la précision moyenne sur l'ensemble du jeu de données.
3. Malgré la performance respectable de notre approche (HEMIS2023), la comparaison avec "HAN2021" révèle un potentiel d'amélioration. Leur performance supérieure nous pousse à envisager des ajustements pour optimiser davantage notre modèle.
4. Bien que "ECIR2013" ait une meilleure performance MAP, notre approche se rapproche de celle-ci et la surpasse en P@30, indiquant une efficacité supérieure pour les 30 documents les plus pertinents.

III.8 Fiche de Synthèse :

III.8.1 Description de l'approche :

Notre méthode de recherche d'information est un système novateur qui se distingue des approches traditionnelles en intégrant des aspects thématiques et temporels. Cette approche est particulièrement adaptée pour améliorer la recherche d'information dans les

microblogs, un domaine qui présente des défis uniques tels que le volume élevé d'information et la concision des tweets.

III.8.2 Résultats obtenus :

L'évaluation de notre approche a montré des résultats prometteurs. Selon les métriques MAP (Mean Average Precision) et P@30 (Precision at 30), notre méthode a surpassé plusieurs systèmes de recherche d'information concurrents, dont "INDRI", "Lucene baseline", et "GHELLAL2023".

De plus, bien que l'approche "ECIR2013" ait montré une performance légèrement supérieure en termes de MAP, notre méthode a surpassé "ECIR2013" en termes de P@30. Cela indique que notre système est capable de renvoyer des résultats plus pertinents en haut de la liste.

III.8.3 Points forts de l'approche :

Notre approche offre plusieurs avantages clés :

1. Pertinence accrue des résultats : Grâce à l'intégration d'aspects thématiques et temporels, notre approche est capable de renvoyer des résultats plus pertinents, comme le démontre notre score P@30 supérieur.
2. Adaptabilité aux microblogs : Notre méthode a été spécialement conçue pour la concision des tweets.

III.9 Conclusion

Dans ce chapitre nous avons décrit l'implémentation du travail effectué : Le choix du python qui a été fait parce qu'il est le langage qu'on maîtrise le plus. les data-sets choisis sont les plus utilisés dans le domaine de la recherche d'information . Les résultats obtenus ont été comparé aux ceux trouvés dans différents travaux publiés.

Conclusion générale et perspectives

Notre travail vise principalement à améliorer la recherche des microblogs (tweets) au sein d'un corpus de tweets. L'objectif est de trouver un modèle qui améliore la sélectivité des microblogs pertinents pour un besoin d'information spécifique exprimé par un utilisateur. Par l'usage de l'aspect thématique en conjonction avec l'aspect temporel.

Nous avons exploré en détail l'état de l'art des systèmes de recherche d'information dans les microblogs, identifiant ainsi leurs forces et leurs faiblesses. Nous avons tenté d'améliorer la recherche d'information dans les microblogs en abordant les lacunes des travaux précédents. D'abord Nous avons constaté que la brièveté des tweets et la qualité du langage utilisé pour rédiger ces documents influencent négativement la fonction de correspondance requête-tweet. Comme nous avons confirmé la nécessité d'introduire d'autre aspect de pertinence.

Afin d'améliorer la pertinence des tweets résultats de la recherche, plusieurs études ont intégré des preuves temporelles à leurs modèles, en les combinant avec des preuves lexicales. De notre côté, nous avons proposé une nouvelle approche pour performer la recherche. Basé sur l'intuition « Les tweets pertinents pour un besoin d'information partagent le même vocabulaire comme possèdent les mêmes caractéristiques temporelles que la requête ». Concernant l'aspect expérimental, nous avons mené nos évaluations sur la collection de test de la tâche microblogs de TREC2011.

Nous avons rencontré certains problèmes au cours de notre travail tel que la difficulté d'utiliser Indri et le Biterm.

Le travail présenté ouvre plusieurs pistes de recherche futures. Il serait intéressant :

- D'introduire l'aspect sémantique pour améliorer l'appariement thématique tweet/requête par l'usage du Deep Learning.
- D'élargir la requête avec les termes les plus probables du meilleur topic, puis effectuer une nouvelle recherche avec Indri.
- Utiliser un autre moteur de recherche vu les difficultées que nous avons rencontré avec Indri .
- En raison de la multitude d'abréviations possibles pour chaque signification, notre

liste d'abréviations n'est pas exhaustive et nécessite donc un enrichissement continu.

— Améliorer les techniques de nettoyage des données peut aider à mieux comprendre le contenu des tweets.

— Les tweets ont souvent des règles de segmentation uniques en raison de leur utilisation unique de la ponctuation et des espaces. L'amélioration de la tokenisation pour gérer ces cas est une perspective importante.

— En optimisant le nettoyage du corpus TREC2011, notamment le filtrage de bruits tels que fautes, liens et emojis, on peut accroître l'efficacité des analyses subséquentes.

Annexe

III.10 Présentation de la plateforme Twitter

Nous présentons ci-dessous les principales spécificités de cette plate-forme, ainsi que l'information qui y est produite.

III.10.1 Les Followers :

Twitter a instauré le système de "followers" permettant de suivre les activités d'autres utilisateurs. Ainsi, nous avons des abonnés (followers) et nous sommes abonnés à d'autres (nous sommes leurs followers), recevant les informations qu'ils partagent et étant informés dès qu'ils mettent à jour leur statut. Cette mise à jour est ajoutée à leur page personnelle, et un aperçu est illustré dans la Figure III.11.

Figure III.11 — Capture d'écran de la page personnelle d'ensemble Twitter

Pour suivre un utilisateur sur Twitter, il suffit de cliquer sur le bouton "Suivre" ou "Follow" sur sa page. Nous avons la possibilité de suivre tous les autres utilisateurs, sauf si l'utilisateur a défini son profil en mode privé. Dans ce cas, il est nécessaire d'envoyer une demande d'approbation avant de pouvoir le suivre..

III.10.2 Lexique de Twitter :

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

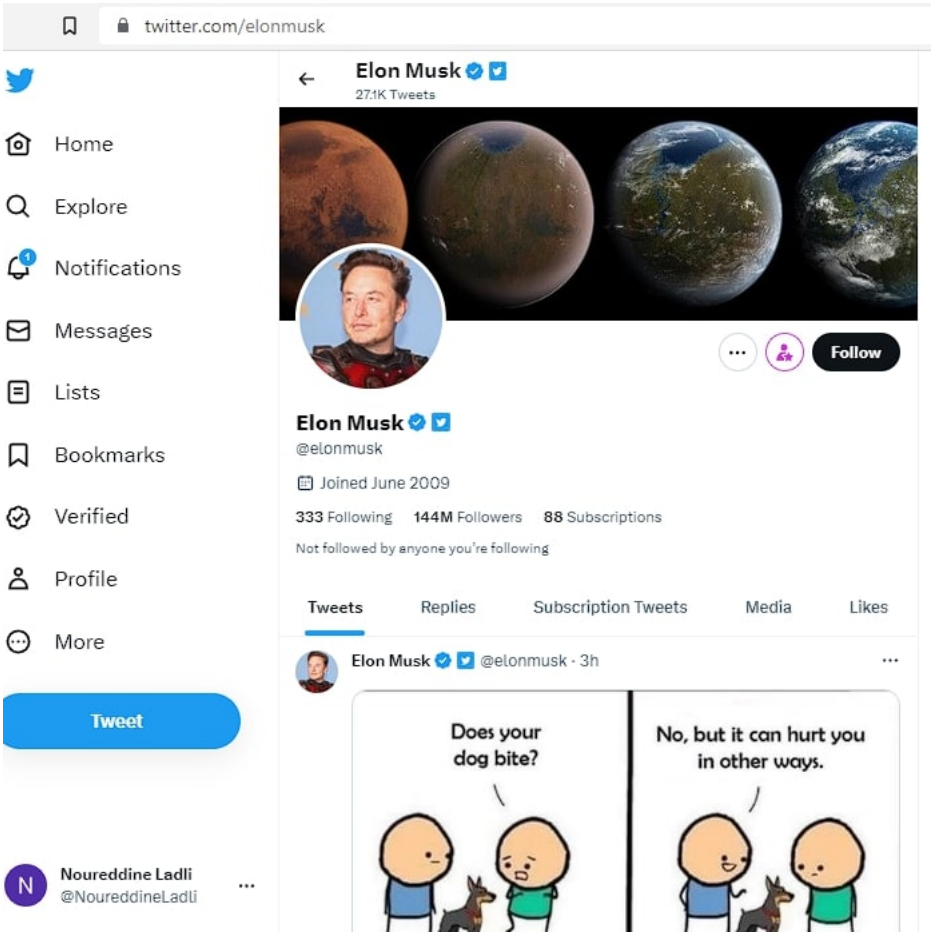

Figure III.12 — Capture d'écran de la page profile de l'utilisateur de Twitter

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

Figure III.13 — Capture d'écran d'un exemple de tweet.

• **J'aime :** Sur Twitter, en cliquant sur le bouton "J'aime" en forme de cœur, vous pouvez exprimer votre appréciation pour un tweet et le sauvegarder pour le retrouver ultérieurement. Cela indique à l'auteur que vous aimez son contenu.

Following / Abonnements : Le nombre de comptes Twitter que nous suivons correspond au nombre d'abonnements. Pour connaître ce nombre, rendez-vous sur la page d'accueil Twitter et il sera affiché dans la colonne de droite, en haut. Pour voir la liste complète des personnes que vous suivez, il vous suffit de cliquer sur le nombre d'abonnements ou sur l'option « Abonnements ».

Figure III.14 — Capture d'écran d'un exemple d'abonnement.

Followers / Abonnés : Le nombre de comptes qui suivent une personne est appelé le nombre d'abonnés de cette personne. Ce nombre est affiché sur la page d'accueil de Twitter, dans la colonne de droite. Pour voir la liste des personnes qui suivent un utilisateur, il suffit de cliquer sur le nombre d'abonnés ou de choisir l'option « Abonnés ».

Figure III.15 — Capture d'écran d'un exemple d'un abonné.

• **@Réponses :**si nous souhaitons répondre à un tweet, nous pouvons envoyer un tweet commençant par le nom du compte précédé d'un symbole "@". Par exemple, si nous tweetons "@Antoine Bonjour !", notre message "Bonjour" sera envoyé au compte d'Antoine. Il pourra voir notre réponse dans l'onglet "Réponses" de son profil.

• **Timeline :**la timeline de Twitter présente les tweets dans un ordre antéchronologique, affichant les messages les plus récents en premier. La timeline générale montre les tweets des comptes suivis, tandis que la timeline personnelle affiche les propres tweets de l'utilisateur.

• **Les tags (@)** sur Twitter créent un lien vers le compte d'un utilisateur mentionné.

Si un tweet débute par une mention, il n'est visible que par ceux qui suivent à la fois l'auteur et la personne mentionnée.

Figure III.16 — Capture d'écran d'un exemple de mention.

• **Retweet (RT) :** est une fonction qui permet de partager le tweet d'un autre utilisateur avec vos abonnés, servant à rediffuser le message à votre audience. L'abréviation "RT" signale généralement qu'un message a été retweeté.

Figure III.17 — Capture d'écran d'un exemple de retweet.

• **Le Message Privé (MP)**, ou Direct Message (DM) en anglais, est une fonctionnalité de Twitter qui permet d'envoyer des messages privés, limités à 140 caractères, qui n'apparaissent pas en public mais dans une messagerie interne. Pour envoyer un MP, il faut suivre la personne concernée qui, à son tour, peut nous répondre seulement si elle nous suit aussi.

• **Hashtag(#) :** Les hashtags sur Twitter organisent les conversations, facilitent le suivi des discussions et aident à se connecter avec une communauté partageant les mêmes centres d'intérêt.

Figure III.18 — Capture d'écran d'un exemple de hashtag.

• **Tendances :** Les tendances sur Twitter, reflétant les sujets populaires du moment, sont personnalisées en fonction de notre localisation et des comptes que nous suivons, présentant ainsi des sujets pertinents pour nous.

Figure III.19 — Capture d'écran d'un exemple de tendance.

III.10.3 Type de tweets :

Il existe plusieurs types de tweets sont :

• **Tweet normal :** tout message de 140 caractères maximum publié sur Twitter.

• **Réponses :** Un tweet qui commence par le nom d'utilisateur (@nomdutilisateur) d'un autre utilisateur et qui répond à l'un de ses tweets, est un exemple de réponse directe sur Twitter.

• **Mention :** Un tweet avec une mention à un autre utilisateur de Twitter en utilisant le symbole @, par exemple : "Salut @Assistance ! Quoi de neuf ?"

• **Message direct (DM) :** Vous pouvez envoyer un tweet privé à une personne qui vous suit, mais vous ne pouvez pas envoyer de message direct à quelqu'un qui ne vous suit pas.

Bibliographie

[1] K. Sauvagnat, "Modèle flexible pour la recherche d'information dans des corpus de documents semi-structurés," Ph.D. dissertation, Université Paul Sabatier-Toulouse III, 2005.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[3] A. Leila, "Reformulation de la requête en recherche d'information en intégrant le profil utilisateur," Ph.D. dissertation, Université Mouloud Mammeri, 2012.

[4] M. Maron and J. Kuhns, "On relevance, probabilistic indexing and information retrieval," *Journal of the Association for Computing Machinery*, vol. 7, pp. 216–244, 1960.

[5] S. E. Robertson and K. Sparck Jones, "Relevance weighting of search terms," *Journal of the American Society for Information Science*, vol. 27, no. 3, pp. 129–146, 1976.

[6] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1998, pp. 275–281.

[7] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 2, pp. 179–214, 2004.

[8] D. Hiemstra, "Using language models for information retrieval," Ph.D. dissertation, University of Twente, 2001.

[9] V. Lavrenko and W. B. Croft, "Relevance based language models," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, September 2001, pp. 120–127.

[10] C. C. Kuhlthau, "Inside the search process : Information seeking from the user's perspective," *Journal of the American Society for Information Science*, vol. 42, no. 5, pp. 361–371, 1991.

[11] F. Damak, "Étude des facteurs de pertinence dans la recherche de microblogs." Ph.D. dissertation, Université Paul Sabatier, 2014.

[12] G. Salton and M. J. McGill, *Introduction to modern information retrieval.* McGraw-Hill, 1983.

[13] A. Bruns and J. E. Burgess, "The use of twitter hashtags in the formation of ad hoc publics," in *Proceedings of the 6th European Consortium for Political Research (ECPR) General Conference*, 2012, pp. 1–20.

[14] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter power : Tweets as electronic word of mouth," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 11, pp. 2169–2188, 2009.

[15] J. Teevan, D. Ramage, and M. R. Morris, "twittersearch : A comparison of microblog search and web search," in *WSDM '11 : Proceedings of the Fourth ACM International Conference on Web Search and Data Mining.* New York, NY, USA : ACM, 2011, pp. 35–44.

[16] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users : real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web.* ACM, 2010, pp. 851–860.

[17] I. Ounis, J. Lin, and I. Soboroff, "Overview of the trec-2011 microblog track," in *TREC '11 : 20th Text Retrieval Conference*, 2011.

[18] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, and et al., "Time is of the essence : Improving recency ranking using twitter data," in *WWW '10 : Proceedings of the 19th International Conference on World Wide Web*, 2010.

[19] M. Magnani, D. Montesi, and L. Rossi, "Conversation retrieval for microblogging sites," *Information Retrieval*, vol. 15, no. 3-4, pp. 354–372, 2012.

[20] F. Damak, K. Pinel-Sauvagnat, G. Cabanac, and M. Boughanem, "Effectiveness of state-of-the-art features for microblog search," in *SAC'13 : ACM Symposium on Applied Computing.* ACM, 2013.

[21] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank : finding topic-sensitive influential twitterers," in *WSDM'10 : Proceedings of the Third ACM International Conference on Web Search and Data Mining.* New York, NY, USA : ACM, 2010, pp. 261–270.

[22] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, pp. 107–117, 1998.

[23] L. Ben Jabeur, L. Tamine, and M. Boughanem, "Active microbloggers : Identifying influencers, leaders and discussers in microblogging networks," in *String Processing*

and Information Retrieval, L. Calderón-Benavides, C. González-Caro, E. Chávez, and N. Ziviani, Eds., vol. 7608. Berlin Heidelberg : Springer, 2012, pp. 111–117.

[24] C. Lee, H. Kwak, H. Park, and S. Moon, "Finding influentials based on the temporal order of information adoption in twitter," in *WWW'10 : Proceedings of the 19th International Conference on World Wide Web*. New York, NY, USA : ACM, 2010, pp. 1137–1138.

[25] R. Cappelletti and N. Sastry, "Iarank : Ranking users on twitter in near realtime, based on their information amplification potential," in *Proceedings of the 2012 International Conference on Social Informatics*. Washington, DC, USA : IEEE Computer Society, 2012, pp. 70–77.

[26] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[27] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA : Association for Computational Linguistics, 2005, pp. 347–354.

[28] D. A. Shamma, L. Kennedy, and E. F. Churchill, "Tweet the debates : Understanding community annotation of uncollected sources," in *Proceedings of the First SIGMM Workshop on Social Media*. New York, NY, USA : ACM, 2009, pp. 3–10.

[29] M. Efron, "Hashtag retrieval in a microblogging environment," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA : ACM, 2010, pp. 787–788.

[30] D. Ramage, S. T. Dumais, and D. J. Liebling, "Characterizing microblogs with topic models," in *ICWSM'10*, 2010, pp. –1–1.

[31] S. Song, Q. Li, and N. Zheng, "A spatio-temporal framework for related topic search in micro-blogging," in *Proceedings of the 6th International Conference on Active Media Technology*. Berlin, Heidelberg : Springer-Verlag, 2010, pp. 63–73.

[32] M. Bernstein, B. Suh, L. Hong, J. Chen, S. Kairam, and E. Chi, "Eddi : Interactive topic-based browsing of social status streams," in *ACM Symposium on User Interface Software and Technology*. New York, NY : ACM, 2010, pp. 303–312.

[33] S. Robertson, "Understanding inverse document frequency : On theoretical arguments for idf," *Journal of Documentation*, vol. 60, 2004.

[34] R. Li, K. H. Lei, R. Khadiwala, and K.-C. Chang, "Tedas : A twitter-based event detection and analysis system," in *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, 2012, pp. 1273–1276.

[35] V. Lampos and N. Cristianini, "Tracking the flu pandemic by monitoring the social web," in *Cognitive Information Processing (CIP), 2010 2nd International Workshop on*, 2010, pp. 411–416.

[36] J. Sankaranarayanan, H. Samet, B. Teitler, and M. D. Lieberman, "Twitterstand : News in tweets," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 42–51.

[37] M. Efron, "The university of illinois graduate school of library and information science at trec 2011," in *TREC*, 2011, (2011a).

[38] D. Tseng, J.-P. Volkmer, S. B. Willingham, H. Contreras-Trujillo, J. W. Fathman, N. B. Fernhoff, J. Seita, M. A. Inlay, K. Weiskopf, M. Miyanishi *et al.*, "Anti-cd47 antibody–mediated phagocytosis of cancer by macrophages primes an effective antitumor t-cell response," *Proceedings of the National Academy of Sciences*, vol. 110, no. 27, pp. 11 103–11 108, 2013.

[39] Z.-y. Han, L.-l. Kong, and H.-l. Qi, "Time segment language model for microblog retrieval," *Neural Computing and Applications*, vol. 33, pp. 4763–4777, 2021.

[40] E. M. Voorhees and L. P. Buckland, Eds., *TREC : Experiment and Evaluation in Information Retrieval*. MIT Press, 2012.

[41] C. Peters, M. Braschler, P. Clough, J. Gonzalo, G. J. Jones, M. Kluck, ..., and E. Toms, Eds., *Information access evaluation. Multilinguality, multimodality, and visual analytics : Third International Conference of the CLEF Initiative, CLEF 2012, Rome, Italy, September 17-20, 2012. Proceedings*. Springer, 2012.

[42] T. Sakai, N. Kando, and H.-H. Chen, "Information retrieval technology : 15th asia information retrieval societies conference, airs 2019, hong kong, china, november 7–9, 2019, proceedings," in *15th Asia Information Retrieval Societies Conference*. Hong Kong, China : Springer Nature, 2019, pp. 1–500.

[43] L. Ben Jabeur, "Leveraging social relevance : Using social networks to enhance literature access and microblog search," Ph.D. dissertation, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2013.

[44] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[45] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft, "Indri : A language model-based search engine for complex queries," in *Proceedings of the International Conference on Intelligent Analysis*, vol. 2, no. 6, August 2005, pp. 2–6.

[46] D. Metzler and W. B. Croft, "Combining the language model and inference network approaches to retrieval," in *Proceedings of the 27th Annual International ACM SIGIR*

Conference on Research and Development in Information Retrieval. ACM, July 2004, pp. 377–384.

[47] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2001, pp. 334–342.

[48] S. Wager, T. Hastie, and B. Efron, "Confidence intervals for random forests : The jackknife and the infinitesimal jackknife," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1625–1651, 2014.

[49] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC Press, 1986.

[50] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[51] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22nd international conference on World Wide Web*, May 2013, pp. 1445–1456.