République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITÉ DE SAAD DAHLEB – 1 –

Faculté des Sciences

Département d'Informatique

MEMOIRE DE FIN D'ETUDES EN VUE DE

L'OBTENTION DU DIPLÔME DE MASTER EN

INFORMATIQUE

Option : Traitement Automatique de Langue (TAL)

Extraction et classification des informations médicales des réseaux sociaux

Réalisé par :

- **Mlle. Bouziane Salima**
- **Mlle. Abeddou Imene**

Encadré par :

- **M.Abbas.M**

Promotrice :

- **M^{me}. Benblidia.N**

Devant le jury composé de :

- **M. Hamouda.M**
- **M^{me}. Lahiani .N**

Soutenu le : 08 / 10 / 2020

Remerciement

Nous tenons, tout d'abord, à remercier ALLAH le tout puissant pour toute la volonté et le courage qu'il nous a données pour l'achèvement de ce travail.

Nous tenons à remercier :

***Mr** Abbas encadreur de projet pour nous avoir accordé sa confiance pour la réalisation de ce projet à distance les unes des autres et pour le grand honneur qu'il nous a fait en nous proposant le sujet de ce mémoire de fin d'étude.*

***Mr** Lichouri Mohamed Chercheur au Centre de Recherche Scientifique et Technique, qui a crus en nous et nous a soutenu dans notre démarche qu'a toujours répondue présente quand nous avons besoin de lui. Les mots nous manquent pour lui exprimer nos profonds remerciements.*

M^me^ *Benblidia Nadjia professeur dans l'université Saad Dahleb, pour leurs conseils dans le cheminement de ce mémoire.*

Nous remercions toutes les personnes qui ont contribué de manière directe ou indirecte à l'aboutissement de ce travail.

Dédicace

J'ai le grand plaisir de dédier cet évènement marquant de ma vie :

A la mémoire de mon grand-père, mon deuxième père disparu trop tôt,

Je ne saurais exprimer mon grand chagrin pour ton absence.

A mes très chers parents, sources de vie, pour leur sacrifices afin que

rien n'entrave le déroulement de mes études.

Ma mère, tu m'as donné la vie, la tendresse et le courage pour réussir.

Mon père l'épaule solide, l'œil attentative compréhensive.

Que dieu vous préserve et vous procurez santé et longue vie.

A ma chère sœur Malika et mon cher frère Mohamed Rafik

Source de joie et de bonheur.

A tous ceux que j'aime et ceux qui m'aiment.

A mes chères amies pour leur sympathie

Abir, c'était ma base de continuation

Meriem et Kaouther

Enfin, je remercie mon binôme Salima qui a contribué à la réalisation de ce modeste travail.

Imène

Dédicace

Je dédie ce modeste travail ;

A ma très chère mère Je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et à vos sacrifices innombrables. Que ce modeste travail, soit pour toi une petite compensation et reconnaissance envers ce que tu as fait d'incroyable pour moi. Que Dieu, le puissant, te préserves et te procures santé et longue vie afin que je puisse à mon tour vous remplir.

A la mémoire de mon Père qui nous a quittées en 2016. Aucune dédicace ne saurait exprimer l'amour , l'estime , le dévouement et le respect que j'ai toujours eu pour vous.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.

Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

A mes chères sœurs Nabila, Naima, Souhila , Aicha et Dounia.

A mes chers frères krimo, Chaaban , Fouad, Bachir , Raouf et Billal

A mes chères amies Ikram , Mimi et Roumaissa

A toute ma belle famille

A tous ceux qui me sont chers.

A mon binôme Imene

Salima

Résumé

Le traitement automatique de la langue est un domaine multidisciplinaire impliquant la linguistique, l'informatique et l'intelligence artificielle. Il vise à créer des outils de traitement de la langue naturelle pour diverses applications. Parmi ces applications, nous pouvons citer celles qui relèvent du traitement des réseaux sociaux. Les sites de médias sociaux, tels que Twitter, sont une source riche de nombreux types d'informations, notamment en matière de santé. Ce travail s'inscrit dans le cadre de la réalisation d'un système d'extraction et classification des informations médicales des réseaux sociaux (twitter). Notre objectif consiste aussi à détecter avec précision des entités telles que les médicaments et les symptômes et faire la séparation des textes contenant des informations médicales de ceux qui n'en contiennent pas. Ces fonctionnalités ont été réalisées en s'inspirant des techniques de classification automatique de données textuelles. La première phase de ce travail consiste à la préparation des données nécessaires à la réalisation de cette tâche. Nous avons appliqué ensuite un prétraitement au corpus. Nous avons utilisé des méthodes d'apprentissage automatique. Pour atteindre notre objectif, plus précisément, nous avons testé différents algorithmes de classification et comparé leurs performances.

Mots clés :

Extraction d'information, Twitter, Diagnostique médicale, Traitement automatique de la langue, Classification de texte.

Abstract

Natural Language Processing is a multidisciplinary that brings linguistics, computer science and artificial intelligence, together to create natural language processing tools for various applications. Applications of Natural Language Processing are numerous, we can cite those related to the processing of social networks. Social media sites, such as Twitter, are a rich source of many types of information, especially in the field of health. The main goal of this work is to develop a system for extracting and classifying medical information from social networks (twitter). In addition, we aim at accurately detecting entities such as drugs and symptoms, and to separate texts containing medical information from those that do not, using automatic classification techniques of textual data. The first phase of this work consists in preparing the data necessary to perform this task. We then apply pre-processing to the corpus. We have used machine learning methods, more precisely, we have tested several classification algorithms compared their performance.

Keywords:

Information extraction, Twitter, Diagnostic medical, Natural language processing, Text Classification.

ملخص

تعد المعالجة التلقائية للغات مجالًا متعدد التخصصات يشمل اللغويات وعلوم الكمبيوتر والذكاء الاصطناعي، ويهدف إلى إنشاء أدوات معالجة اللغة الطبيعية لمختلف التطبيقات.من بين تطبيقات المعالجة التلقائية للغات الطبيعية، يمكننا الاستشهاد بتلك التي تتعلق بمعالجة الشبكات الاجتماعية. تعد مواقع الوسائط الاجتماعية ، مثل تويتر،مصدرًا غنيًا لأنواع عديدة من المعلومات ، بما في ذلك الصحة. هذا العمل هو جزء من إنشاء نظام لاستخراج وتصنيف المعلومات الطبية من الشبكات الاجتماعية (تويتر) و الكشف الدقيق للكيانات مثل الأدوية والأعراض وفصل النصوص التي تحتوي على معلومات طبية عن تلك التي لا تحتوي عليها. مستوحاة من تقنيات التصنيف التلقائي للبيانات النصية. تتكون المرحلة الأولى من هذا العمل من إعداد البيانات اللازمة لتحقيق هذه المهمة. ثم قمنا بتطبيق المعالجة المسبقة على الجسم. نستخدم طرق التعلم الآلي. لتحقيق هدفنا، بشكل أكثر دقة، قمنا باختبار خوارزميات تصنيف مختلفة و مقارنة أدائها.

.

الكلمات المفتاحية

استخراج المعلومات، طب، تويتر، تشخيص ،تصنيف نصي ،المعالجة التلقائية للغة.

Table des matières

Liste des tableaux

Liste des Figures

Introduction générale

Le traitement automatique de la langue est un champ de savoir et de techniques élaborées autour de problématiques diverses. Les concepts et techniques qu'il utilise se trouvent à la croisée de multiples champs disciplinaires : l'IA « traditionnelle », l'informatique théorique, la logique, la linguistique, etc... [1].

Les médias sociaux permettent le partage public d'informations entre toute personne possédant un compte. Alors qu'il était auparavant utilisé principalement pour connecter des personnalités publiques telles que des politiciens, des célébrités et des athlètes avec le grand public, des utilisations supplémentaires des médias sociaux sont apparues. Étant donné que les médias sociaux permettent le partage d'informations, les applications des médias sociaux à la médecine ont récemment attiré beaucoup d'attention. Actuellement, Twitter est la forme la plus populaire de médias sociaux utilisée pour la communication en matière de santé [2]. Les messages Twitter (tweets) contiennent divers types d'informations, dont des informations relatives à la santé. L'analyse des tweets liés à la santé nous aiderait à comprendre les conditions de santé et les préoccupations rencontrées dans notre vie quotidienne en adoptant des techniques de traitement du langage naturel [3].

Certains affirment que les médias sociaux n'ont pas sa place dans les soins de santé, tandis que d'autres affirment que le partage ouvert des informations les médias révolutionneraient l'accessibilité à la médecine. Ici, nous examinons l'utilisation de Twitter en médecine.

Problématique

Des masses de données concernant la santé sont de plus en plus importantes dans les réseaux sociaux tels que Twitter. Il devient fatidique d'utiliser ces données pour aider l'utilisateur à accéder facilement à l'information médicale. De là se pose les questions suivantes :

- Comment extraire ces données médicales à partir de twitter ?
- Comment traiter ces données ?
- Comment classifier ces données ?
    - Classer les tweets à des tweets médicales ou non médicales.
    - Identifier les symptômes et les médicaments de ces tweets médicales.

Beaucoup de questions auxquelles nous essaierons de répondre.

Objectif

Notre objectif est d'extraire et classifier les informations médicales des réseaux sociaux par un apprentissage automatique supervisé, et ceci en se basant sur les méthodes de classification. Dans ce travail, nous utilisons des corpus de tweets de langue anglais et arabe.

Méthodologie adoptée

Pour atteindre cet objectif nous avons extrait des conversations liées à des publications sur la santé et la maladie sur Twitter à l'aide d'un ensemble de mots clés prédéfinis, et validé comment ce protocole de recherche a pu détecter les tweets de maladie pertinents. Dans notre travail nous procédons à l'analyse et la classification des textes extraits de Twitter. Nous sommes particulièrement concentrés sur la phase de prétraitement des tweets avant d'utiliser un algorithme de Classification. La classification de texte est l'activité du traitement automatique des langues naturelles qui consiste à classer de façon automatique des ressources textuelles, généralement en provenance d'un corpus. L'algorithme de classification comme « 'KNN' KNeighborsClassifier, 'LSVM' LinearSVM, 'LR' LogisticRegression, 'DT' DecisionTreeClassifier, 'MNB' MultinomialNB, 'BNB' BernoulliNB, 'RF' RandomForestClassifier, 'GB' GradientBoostingClassifier» sont utilisé pour former et tester les protocoles d'identification des tweets, les symptômes et les médicaments.

Nous utilisons deux corpus de tweets de langue anglais et arabe, nous visons à identifier les tweets qui contiennent des informations médicales en considérant trois entités : symptômes, médicaments et topiques. Dans le cas des tweets, la classification consiste à annoter les différentes phrases d'un tweet avec des classes. Pour chaque classe C, on trouve des termes importants considérés comme des indicatifs pour la classe C. Par exemple, les noms de médicaments, les symptômes sont des indicatifs du sujet médical.

A cet effet, notre travail s'appuie sur la réalisation d'une application web qui permet d'identifier les tweets qui contiennent des informations médicales en considérant deux entités : symptômes et médicaments.

Organisation du mémoire

Afin d'aborder tous ces aspects, le présent mémoire s'articule autour de quatre chapitres **:**

Chapitre 1 : Notions fondamentales sur le TAL.

Chapitre 2 : Conception de la solution proposée.

Chapitre 3 : Expérimentation et comparaison des résultats.

Chapitre 4 : Implémentation et réalisation.

Enfin, ce mémoire se termine avec une conclusion générale.

Chapitre 1 : Notions fondamentales sur le TAL

1.1. Introduction

Dans ce chapitre, nous présentons un aperçu sur le traitement automatique du langage naturel, ainsi une définition d'extraction d'information et ses domaines d'applications. Par la suite, nous citons quelques applications médicales appliquées sur les réseaux sociaux, ensuite une brève définition sur l'apprentissage automatique supervisée et non supervisée. Nous terminons ce chapitre en citons quelques définitions de la classification de texte, l'intérêt et ses difficultés.

1.2. Traitement automatique de la langue

1.2.1. Définition

Le Traitement Automatique des Langues est une suite d'actions ou calculs à faire effectuer par la machine, qui a pour objectif de traiter des données linguistiques (textes) exprimées dans une langue dite "naturelle" [4]. Il permet la conception de programmes capables de traiter automatiquement des données linguistiques de type : Textes écrits, dialogues écrits ou oraux, unités linguistiques (mots, phrases, énoncés).

1.2.2. Objectif

L'objectif du traitement automatique du langage naturel (TAL) est la conception de logiciels, capables de traiter de façon automatique des données linguistiques, c'est-à-dire des données exprimées dans une langue (dite "naturelle") [5]. Ces données linguistiques peuvent être des textes écrits, ou bien des dialogues écrits ou oraux, ou encore des unités linguistiques de taille inférieure à ce que l'on appelle habituellement des textes (par exemple : des phrases, des énoncés, des groupes de mots ou simplement des mots isolés).

1.2.3. Applications du TAL

- **Traduction automatique (TA) :** Il est probable que la TA fasse l'objet d'améliorations importantes dans les années à venir.
- **Correction orthographique :** Elle est intégrée à toute application informatique impliquant la rédaction, correction basée sur des lexiques [6]. Exemple : traitement de texte, courrier électronique, navigateur Internet (zone de saisie).
- **La reconnaissance de la parole :**Est une discipline ayant fait des progrès considérables , Grandes étapes : {Segmentation du flux continu de paroles en unités discrètes ,identification du phonème correspondant à chaque unité ,regroupement des unités pour constituer des mots, prise en compte de la syntaxe pour finaliser le texte écrit } , logiciels de dictée vocale (Via Voice, Dragon Dictate…), reconnaissance de la parole ou commande vocale (Reconnaissance vocale de Windows, Systèmes de

navigation routière GPS, Smartphone…), prototype Google de sous-titrage automatique de You tube.

1.3. Extraction d'information « EI »

L'extraction d'information « EI » recherche des informations précises dans les documents, sans les comparer, en tenant compte de l'ordre et de la proximité des mots pour discriminer des énoncés différents ayant des mots-clés identiques [7]. L'extraction d'information consiste en l'alimentation d'une base de données structurée à partir de données exprimées en langage naturel. Il s'agit de détecter dans le texte en langage naturel les mots correspondant à chaque champ de la base de données. L'analyse est locale. L'extraction d'information est plus complexe, car elle nécessite d'effectuer une analyse lexicale et morphosyntaxique pour reconnaître les constituants du texte (phrases, mots, verbes, adjectifs), leur nature pour détecter les phrases pertinentes et en extraire les informations voulues.

Depuis quelques années, une masse grandissante de données est générée de toute part et dans différents domaines [8]. Les techniques usuelles analysant ces données sont insuffisantes d'où le besoin d'une nouvelle génération d'outils et de théories pour aider à extraire les informations utiles (les connaissances) à partir des volumes de données numériques qui croissent rapidement. Ces théories et outils sont le sujet d'un nouveau domaine appelé extraction de connaissances à partir de données dont le cœur est la fouille de données.

1.3.1. Domaines d'application

Le succès des méthodes de data mining a intégré cette nouvelle science dans tous les domaines d'intelligence artificielle [8]. Les méthodes de data mining sont applicables au problème d'estimation, préventions, analyse de risque, catégorisation, reconnaissance, etc.

Nous citons ici quelques applications connues du data mining :

• **Web mining :** le web mining est l'analyse des données du web par les techniques du data mining. On distingue différentes tâches du web mining : web content mining (texte, image, …), Web structure mining (liens hypertextes, ...) et Web usage mining (analyse des fichiers logs client et serveur).

• **Texte Mining :** la fouille de texte est utilisée dans divers domaines, les moteurs de recherche, la reconnaissance de l´écriture imprimée ou manuscrite et etc.

• **Médecine :** la fouille de données s'applique aussi dans le domaine médical, tel que le diagnostic automatique ou l'aide au diagnostic (découverte de la maladie du patient d'après ses symptômes), recherche du médicament le plus approprié à une maladie et etc.

1.3.2. Exemple d'extraction d'information automatique dans le domaine médical

L'extraction d'information automatique en domaine médical par projection inter-langue vers un passage à l'échelle est une recherche issue de leur volonté de tester de nouvelles méthodes automatiques d'annotation ou d'extraction d'information à partir d'une langue L1 en exploitant des ressources et des outils disponibles pour une autre langue L2. Cette approche repose sur le passage par un corpus parallèle (L1-L2) aligné au niveau des phrases et des

mots. Pour faire face au manque de corpus médicaux français annotés, Les chercheurs intéressent au couple de langues (français anglais) dans le but d'annoter automatiquement des textes médicaux en français. En particulier, ils intéressent à la reconnaissance des entités médicales. Ils évaluent dans un premier temps notre méthode de reconnaissance d'entités médicales sur le corpus anglais. Dans un second temps, Ils évaluent la reconnaissance des entités médicales du corpus français par projection des annotations du corpus anglais. Ils abordent également le problème de l'hétérogénéité des données en exploitant un corpus extrait du Web et ils proposent une méthode statistique pour y pallier [9].

1.4. Exemples d'applications médicales appliquées sur les réseaux sociaux

Les patients s'expriment plus spontanément à propos de leur pathologie sur les réseaux sociaux qu'auprès des professionnels de santé. Ils y décrivent leur humeur, évoquent leurs conditions de vie, l'expérience qu'ils font du traitement, les bénéfices qu'ils en retirent, l'inconfort qu'ils constatent ou les effets indésirables qu'ils ressentent [10], voici deux exemples d'application :

- **Les réseaux sociaux et la santé : un enjeu pour le suivi patient et la recherche scientifique**

Convaincu que l'exploitation des données de santé ne peut se faire que dans l'intérêt des patients, mais aussi avec les patients, le HealthcareData Institute a initié, au début de l'année 2017, une réflexion sur les patients et leurs données. Au mois d'avril 2017, une réunion de lancement était organisée avec des patients et représentants d'associations de patients. Les participants ont échangé sur l'accès des patients à leurs données de santé et le partage de ces données. Ces échanges ont montré que si le partage de données avec des équipes de recherche pouvait susciter des interrogations ou des appréhensions, le partage de ces mêmes données avec une communauté d'amis, de famille ou de patients sur les réseaux sociaux était en revanche une pratique fréquente et spontanée. Il décrit les principaux usages des réseaux sociaux de santé et les catégories de données générées par les patients sur ces réseaux : échanges au sein de communautés de patients sur leur pathologie et leur prise en charge, débats sur des enjeux de santé publique, partage par les utilisateurs d'informations sur leur santé avec leurs « amis » ou « abonnés » n'appartenant pas à une communauté de patients. Il précise les principaux lieux de partage (Facebook et Twitter). Il décrit également les modalités actuelles d'accès par des tiers aux données échangées sur les principaux réseaux (Facebook et Twitter) [10].

- **Détection de mésusages de médicaments dans les réseaux sociaux**

Ce travail propose l'identification des messages contenant un mésusage de médicament. Un mésusage de médicament apparaît lorsqu'un patient ne respecte pas sa prescription : sous dosage, surdosage, utilisation de médicaments pour des raisons autres de celles de la prescription, consommation de médicaments prescrits pour une autre personne. Ces situations sont dangereuses car elles mettent en danger la santé de la personne. Comme les patients

reportent rarement les mésusages à leurs médecins, il est nécessaire de consulter d'autres sources d'informations pour découvrir ce qui se passe en réalité. Les chercheurs proposent d'étudier les réseaux sociaux, où les patients communiquent librement et facilement sur leur processus de santé, et sans doute sur l'usage de médicaments. Actuellement, les réseaux sociaux sont largement étudiés par plusieurs disciplines et en poursuivant différents objectifs : identification de géolocalisation, fouille d'opinions, extraction d'événements, traduction et résumé automatique. Dans le domaine médical, les réseaux sociaux peuvent être exploités pour fournir des informations pour la surveillance épidémiologique, la qualité de vie des patients et les effets indésirables de médicaments. Cependant, peu de travaux s'intéressent au mésusage de médicaments. Les chercheurs peuvent citer par exemple l'étude des tweets concernant l'usage non-médical de médicaments avec des méthodes non supervisées et la création d'une plateforme générique pour l'étude de sur-usages [11].

1.5. Apprentissage automatique

Le domaine de l'apprentissage automatique s'intéresse aux méthodes inductives permettant d'acquérir des connaissances à partir d'observations d'un phénomène. Cette connaissance peut être exploitée pour des tâches de décision ou de prévision : c'est le cadre de l'apprentissage supervisé, ou à des fins d'analyse exploratoire ou de structuration d'un ensemble de données : c'est le cadre de l'apprentissage non-supervisé [12], nous intéressant à l'apprentissage supervisé.

1.5.1. Apprentissage supervisé

L'apprentissage supervisé consiste à construire un modèle basé sur un jeu d'apprentissage et des labels (nom des catégories ou des classes) et à l'utiliser pour classer des données nouvelles [13]. Cette technique est utilisée dans plusieurs applications telles que les diagnostics médicaux, la prédiction des pannes et la détection des opinions trompeuses dans les réseaux sociaux.

1.5.2. Apprentissage non supervisé

Le cadre de l'apprentissage non-supervisé concerne des tâches d'analyse exploratoire des données. Il s'agit à partir de la seule observation de x, d'extraire, de visualiser et de résumer les corrélations entre des composantes de ce vecteur ; ou bien encore de construire des regroupements homogènes des observations [12].

1.5.3. Avantages et inconvénients de l'apprentissage supervisé et non supervisé

Parmi les avantages et inconvénients liés aux deux approches [7], on peut citer :

- Les groupes ou clusters obtenus par la technique supervisée est de meilleure qualité et plus précise que la technique non-supervisée.

- Dans la technique supervisée, on sait ce qui est attendu favorisant de meilleurs résultats par rapport au non supervisée.

- Un avantage des techniques non supervisées, est qu'elles accomplissent la tâche de similarité sans avoir besoin des données expertisées.

- Un inconvénient des approches supervisées, repose sur le fait qu'il peut être difficile de se procurer des données expertisées.

- L'inconvénient majeur des approches non supervisées qu'elle demande dans l'étape d'évaluation des résultats l'intervention d'un expert.

1.6. Classification automatique des textes

Actuellement, la classification de textes est un domaine de recherche très actif et l'automatisation de cette opération est devenue un enjeu pour la communauté scientifique, les travaux évoluent considérablement depuis une vingtaine d'années et plusieurs modèles ont vu le jour comme le filtrage (classification supervisée bi-classe), le routage (classification supervisée multi-classe) ou le classement ordonnée (classement des textes par ordre de pertinence pour chaque catégorie) [14].Avec ces modèles, des méthodologies de tests et des outils d'évaluation ont été mises en place. Les méthodes de représentation ainsi que les prétraitements correspondants sont maintenant bien connues. Les algorithmes de classification fonctionnent correctement mais déterminer les avantages des uns par rapport aux autres reste souvent délicat ou même améliorer les performances de la même méthode en intégrant d'autres paradigmes comme nous le faisons-nous ici dans le présent mémoire reste toujours un domaine de recherche très prometteur.

La classification de textes est définie comme une opération qui identifie des classes d'équivalence entre des segments de textes en tenant compte de leur contenu informationnel (mots, n-gram, etc.) [7].

Elle connaît ces derniers temps un fort regain d'intérêt. Cela est dû essentiellement à la forte croissance des documents numériques disponibles et à la nécessité de les organiser de façon rapide [14].

1.6.1. Objectifs et intérêts

Les intérêts des méthodes de classification sont multiples, il peut s'agir d'améliorer les performances des moteurs de recherche documentaire ou aussi classer les documents en fonction de leurs références communes à d'autres documents pour faire apparaître les liens qui les unissent [14].

1.6.2. Difficultés particulières de la catégorisation des textes

Le traitement de données textuelles est plus difficile que le traitement des données numérique. Le langage naturel est ambigu, il y a plusieurs façons d'exprimer la même idée (la redondance), ce qui est exprimé possède souvent plusieurs interprétations (l'ambiguïté) et tout n'est pas exprimé dans le discours (l'implicite).

Ajoute à ces particularités, une des difficultés majeures de la catégorisation, il s'agit de la dimension très élevée de l'espace de représentation qui peut prendre plusieurs centaines de milliers pour une collection de textes [15]. En plus, le sur-apprentissage est un problème

pouvant survenir dans les méthodes mathématiques et informatiques de catégorisation comme les réseaux de neurones. Il est en général provoqué par un mauvais dimensionnement de la structure utilisée pour la catégorisation.

Conclusion

Le traitement automatique du texte médical permet d'obtenir une information suffisamment normalisée pour une exploitation en santé publique. L'extraction d'information (EI) est une technologie visant à reconnaitre dans un corpus de documents textuels un ensemble d'informations spécifiques à les extraire et à les structurer dans un format prédéfini. La classification de textes a essentiellement progressé ces dix dernières années grâce à l'introduction des techniques héritées de l'apprentissage automatique qui ont amélioré très significativement les taux de bonne classification.

Chapitre 2 : Conception de la solution proposée

2.1. Introduction

Ce chapitre présente en détail les processus qui vont être utilisés pour la représentation, par la suite nous intéressons aux différentes techniques de prétraitement et de classification sur les données textuelles.

2.2. Cas d'étude

2.2.1. Corpus Anglais

L'objectif de notre travail est la classification des tweets à partir de twitter qui nécessite d'abord un prétraitement pour réduire les informations indésirables (voir la figure 2.1) :

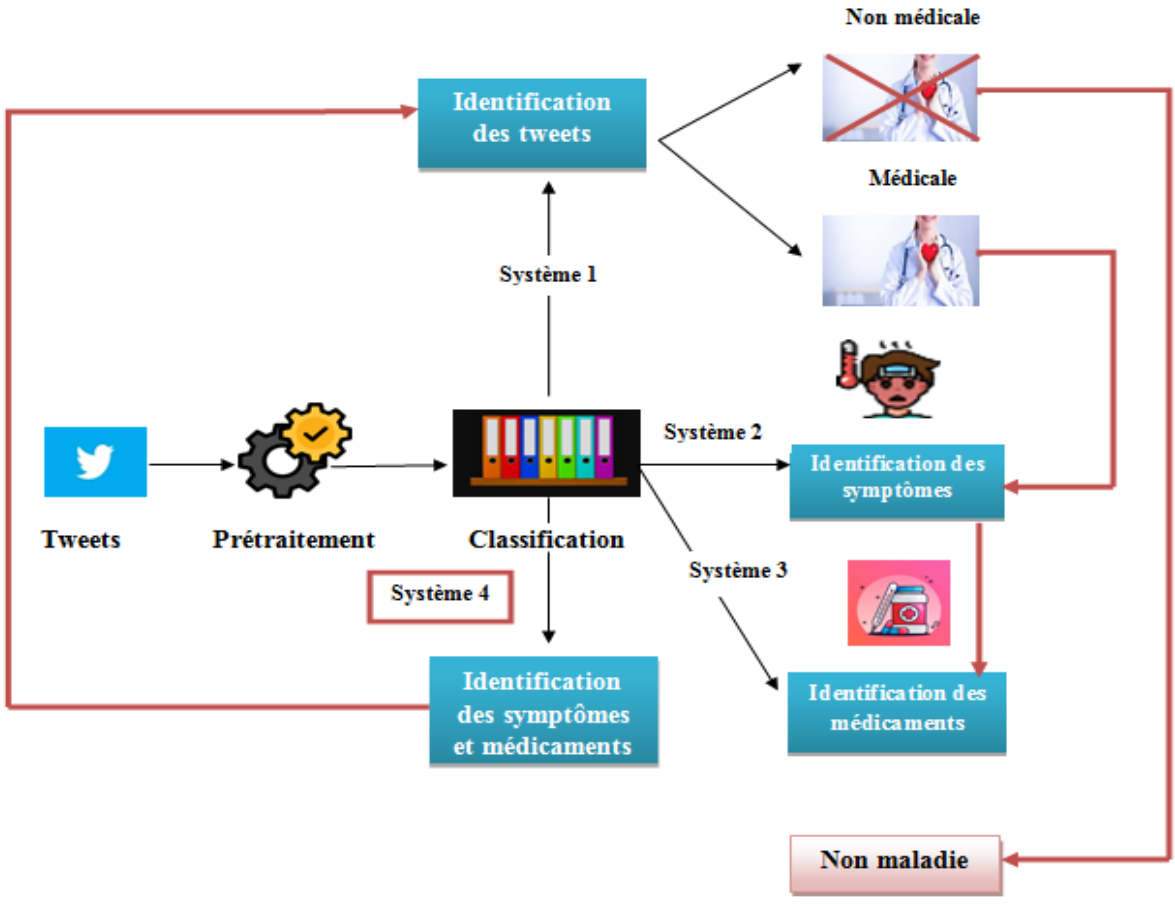


Figure 2.1: processus générale de cas d'étude « Anglais »

2.2.1.1. Système 1 : Identification des tweets

Cette phase a pour objectif de faire la classification des tweets s'ils sont médicaux ou non.

2.2.1.2. Système 2 : Identification des symptômes

Cette phase a pour objectif d'identifier les symptômes des tweets médicales. ·

2.2.1.3. Système 3 : Identification des médicaments

Cette phase a pour objectif d'identifier les médicaments des tweets médicales. ·

2.2.1.4. Système 4 : Identification des symptômes et des médicaments

Cette phase qui porte sur l'identification des symptômes et médicaments pour les tweets médicaux. On a fait une reconnaissance multilabels (tweets médicaux ou non, identification des symptômes et des médicaments) à la fois.

Si le tweet est médical, le système identifie le symptôme ensuite le médicament de ce tweet si les symptômes sont reconnus. Sinon le système affiche non maladie.

2.2.2. Corpus Arabe

L'objectif de notre travail est de faire l'extraction et la classification des tweets à partir de twitter qui nécessite d'abord un prétraitement pour réduire les informations indésirables (voir la figure 2.2) :

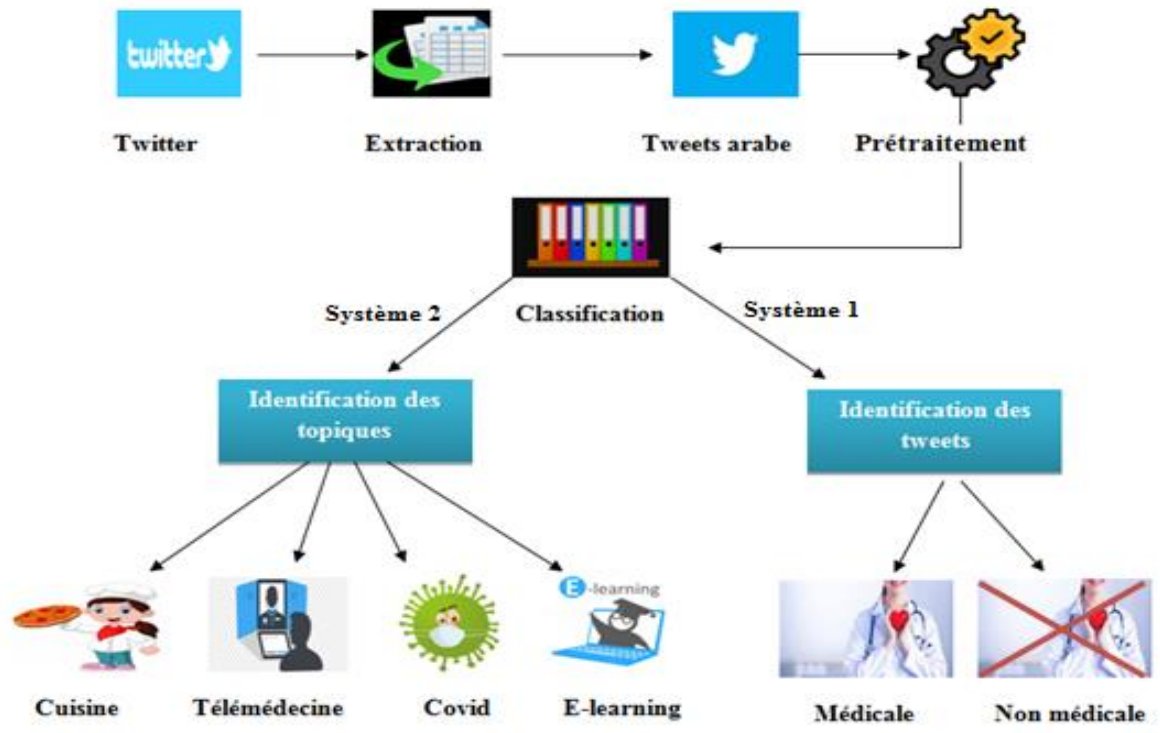


Figure 2.2: processus générale de cas d'étude « Arabe »

2.2.2.1. Système 1 : Identification des tweets arabe

La première étape est pour détecter si les tweets sont médicaux ou non.

2.2.2.2. Système 2 : Identification des topiques

La deuxième étape est pour détecter les topiques soit : Covid, Cuisine, Télémédecine,

E-learning.

2.2.2.3. Twitter

Twitter est une mine d'or de données. Contrairement aux autres plateformes sociales, presque tous les tweets de l'utilisateur sont entièrement publics et extractibles. C'est un énorme avantage si vous essayez d'obtenir une grande quantité de données pour exécuter des analyses. Les données Twitter sont également assez spécifiques. L'API de Twitter vous permet de faire des requêtes complexes comme extraire chaque tweet sur un certain sujet au cours des vingt dernières minutes, ou extraire les tweets non retweetés d'un certain utilisateur [16]. Comme vous pouvez le voir, les données Twitter peuvent être une grande porte d'accès aux informations du grand public et à la façon dont il reçoit un sujet. Cela, combiné à l'ouverture et à la limitation généreuse du débit de l'API de Twitter, peut produire des résultats puissants.

2.2.2.4. Etapes d'extraction

Nous avons utilisé twitter comme ressources. Pour utiliser l'API de Twitter, nous devons créer un compte développeur sur le site des applications Twitter [17], pour accéder à les API il faut Installer tweepy

Nous permettons de faire une recherche sur twitter pour obtenir les tweets qui parlent sur ces 4 sujets :

"covid19, télémédecine, E-learning, cuisine" avec les hashtags suivants :

- **Covid19 :**

#19كوفيد ,#19_كوفيد, #حظر_التجوال, #كورونا,#SARS_COV_2, #التباعد_الاجتماعي, #19-كوفيد, #كوفيد-١٩#,كوفيد, #كورونا_الجديد, #فيروس_كورونا, #كوفيد١٩.

- **Télémédecine :**

#المريض_الرقمي, #الصحة_الإلكترونية, #الصحة_الرقمية, #الخدمات_الصحية_عن_بعد, #التطبيب_عن_بعد, #التحول_الرقمي.

- **E-learning :**

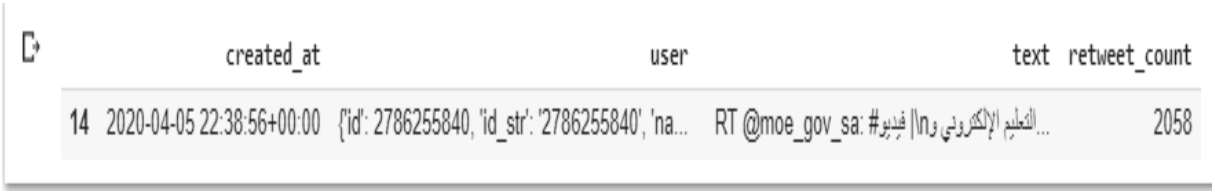
#التعليم_الإلكتروني,#الدراسة_مستمرة, #التعليم_عن_بعد,#التعليم,#انفوجرافيك,#التعلم الإلكتروني, #تجارب_ناجحة_في_التعليم_الإلكتروني,#استشارات_تعلم_الكتروني,#عمادة_التعلم_الإلكتروني_والتعلم_عن_بعد #التعلم_الالكتروني,#افضل_الممارسات_في_التعلم_الالكتروني.

- **Cuisine :**

#محشي,#مطبخي,الطبخ_بالبيت#,طبخات#,فوود#,# طبخ,مطبخ_دلع#,اطبخ_بالبيت#,الطاجين_العربى#
كيك#,وصفاتي# ,وصفات_سهلة#,#ومضة_للتدريب,#مطبخ_ومضة.

Le résultat que nous recevons de l'API Twitter est reçoit au format JSON et contient une quantité importante d'informations. Par souci de simplicité, ce didacticiel se concentre principalement sur l'attribut « texte » de chaque tweet et les informations sur l'utilisateur qui a créé le tweet.

Voici l'exemple de l'objet JSON (représentés dans la figure 2.3) :



	created_at	user	text	retweet_count
14	2020-04-05 22:38:56+00:00	{'id': 2786255840, 'id_str': '2786255840', 'na...	RT @moe_gov_sa: #فيديو \n| التعليم الإلكتروني...	2058

Figure 2.3 : Exemple d'un tweet extracter

2.3. Processus de notre système

Dans le schéma suivant « voir la figure 2.4 », nous résumons les étapes de notre système global :

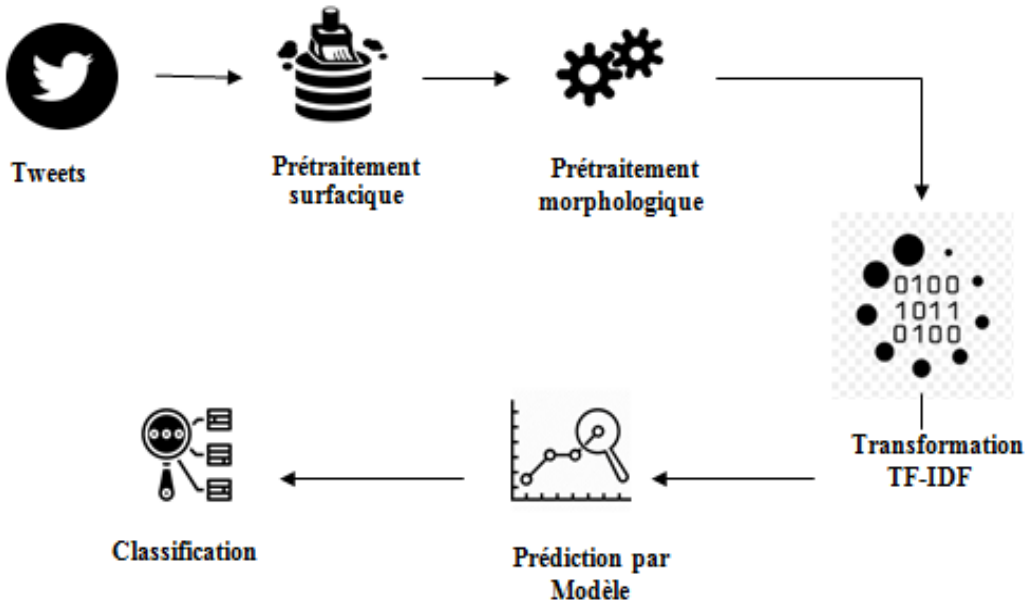


Figure 2.4 : Processus de notre système

2.3.1. Préparation des données (prétraitement)

L'exploitation directe du corpus d'apprentissage constitue un vrai obstacle pour l'algorithme d'apprentissage. Si on utilise directement tous les mots des documents d'entraînement, on se retrouve face à un espace vectoriel énorme.

L'exploration et l'analyse d'un tel espace nécessitent beaucoup de ressources, en fonction de la mémoire, de la puissance et du temps de calcul. En plus, utiliser tous ces mots influencera négativement la fidélité de la classification. La préparation des données est une étape importante qui nous permet de déterminer les attributs utilisés dans la phase de représentation du corpus d'apprentissage. Le choix de la nature de l'attribut d'un document est l'enjeu principal d'une classification automatique efficace [18]. Pour cela, les chercheurs se divisent entre eux en ce qui concerne ce choix. Nous citons, entre autres, ceux qui utilisent les mots comme attributs, et ceux qui préfèrent utiliser les lemmes ou les racines comme, d'autres, préfèrent utiliser les n-grammes.

Le processus de prétraitement passe essentiellement par les étapes illustrer dans « la figure 2.5 » :

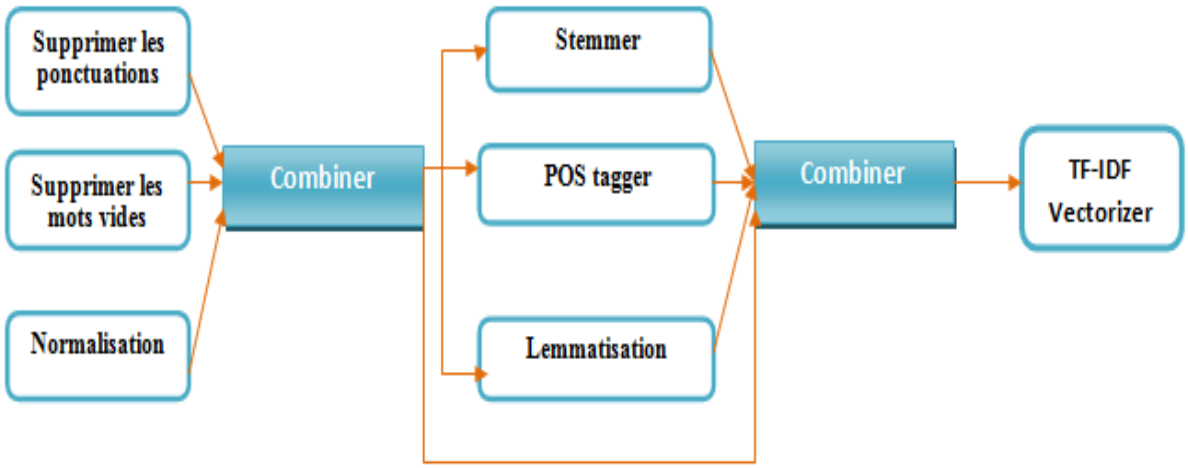


Figure 2.5: Processus de prétraitement de tweets

2.3.1.1. Prétraitement surfacique

- **Normalisation lexicale des tweets**

Les différentes représentations morphologiques de la lettre ا (alef) (ex.إ آ أ ٱ) sont normalisées et remplacées par ا et la lettre finale ي par ى et remplacement de la lettre finale "ئ" par "ء" et "ؤ" par "ء", voici un exemple illustratif (voir figure 2.6) :

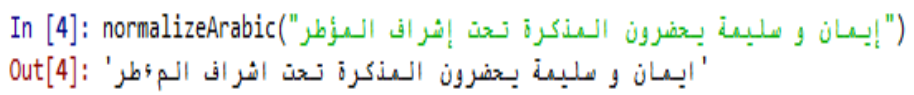


Figure 2.6: Exemple de Normalisation (Arabe)

- **Détection des mots vides (stop liste, stop-word) :**

Les mots vides (ou stop-word) sont des mots qui sont tellement communs qu'il est inutile de les traiter ou de les utiliser dans une recherche d'informations. En Anglais, certains de ces mots sont « the », ~~ is », « far », etc. Un mot vide est un mot non significatif figurant dans un texte. La signification d'un mot s'évalue à partir de sa distribution (au sens statistique) dans une collection de textes [13]. Un mot dont la distribution est uniforme sur les textes de la collection est dit « vide » et ne permets pas de distinguer les textes les uns par rapport aux autres, voici des exemples illustratifs (voir figures 2.7 et 2.8) :

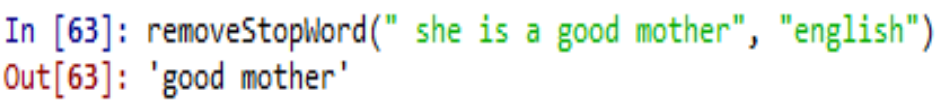


Figure 2.7: Exemple de suppression des mots vides (Anglais)

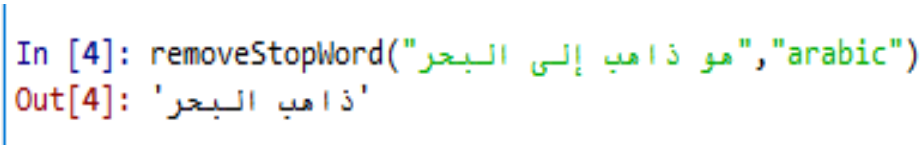


Figure 2.8: Exemple de suppression des mots vides (Arabe)

.

- **Élimination des ponctuations :**

Filtrage des résultats indésirables : définir des schémas syntaxiques pour éliminer des résultats indésirables dus à des erreurs d'extraction.
Par exemple, les candidats termes contenant des caractères de ponctuation comme (!, ?, ., :, etc.) ou des caractères improbables pour un terme comme (#, &, %, {, etc.) ou enfin contenant des nombres comme (1, 2, 3, 4, etc.) sont considérés comme des erreurs d'extractions [19]. Voici des exemples illustratifs (voir figures 2.9 et 2.10) :

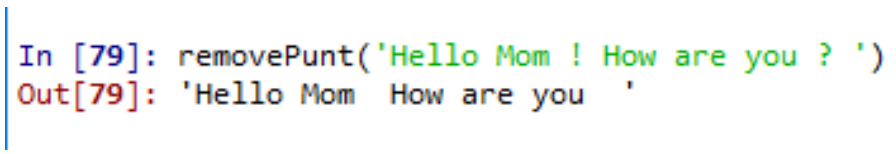


Figure 2.9: Exemple de suppression des ponctuations (Anglais)

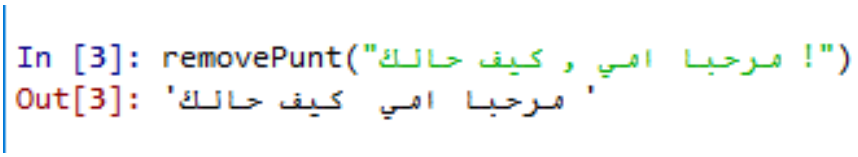


Figure 2.10: Exemple de suppression des ponctuations (Arabe)

2.3.1.2. Prétraitement morphologique

- **Stemmer**

La racine est une technique de réduction des termes utilisés qui est utilisée en supprimant les suffixes tels que « ed », « ing » et « ily ». Il réduit la complexité et permet une récupération d'informations plus efficace, en particulier dans les applications d'exploration de données. Néanmoins, le radical peut créer des mots non réels car le radical ne vérifie pas les règles grammaticales pendant le processus de radical [20]. Voici des exemples illustratifs (voir figures 2.11 et 2.12) :

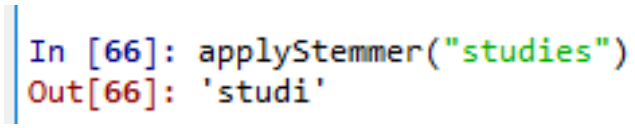


Figure 2.11: Exemple de Stemmer (Anglais)

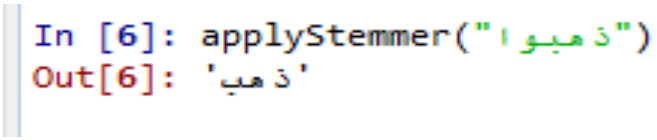


Figure 2.12: Exemple de Stemmer (Arabe)

- **POS Tagger (part-of-speech tagger)**

L'étiquetage grammatical *(*part -of- speech tagging eu anglais abrégé par POS) est un processus qui associe aux mots d'un texte les informations grammaticales comme la partie du discours, le genre, le nombre, etc. à l'aide d'un outil informatique. Les étiqueteurs grammaticaux qui analysent les textes courts et les tweets sont nombreux. On cite TweetNLP, TreeTagger [13]. Nous avons choisi StanfordNLP comme outil de prétraitement des tweets qui peut utiliser des modèles pour l'étiquetage grammatical construits par l'outil Gâte.

❖ **Classes de tagger :** Le tagger de stanford fonctionne avec les textes, il noté les mots par leur catégorie [7], Si le mot est verbe concaténer le mot avec « _V », sont classes est :

✓ Nom noté (N).

- ✓ Adjective noté (A).
- ✓ Adverbe noté (ADV).

Voici des exemples illustratifs (voir figures 2.13 et 2.14) :

```
In [74]: applyPosTag("The baby is sleeping")
Out[74]: 'DT NN VBZ VBG'
```

Figure 2.13: Exemple de POS Tagger (Anglais)

```
In [7]: applyPosTag("دخل الطالب المدرج")
Out[7]: 'JJ NNP NN'
```

Figure 2.14: Exemple de POS Tagger (Arabe)

Ça marche avec les bons ajustements mais les résultats obtenus ne sont pas satisfaisants ce qui veut dire que ces outils ne sont pas performant pour l'arabe comme pour l'anglais.
Les POS Tagger ne sont pas tous juste, car l'outil a des performances limitées pour l'arabe, c'est une limite de l'outil et pas de notre expérience, donc l'expérience est juste, mais on peut dire que pour améliorer les résultats un posTag dédié pour l'arabe va donner de meilleurs résultats.

- **Lemmatisation**

Cette méthode consiste à remplacer les mots du document par leurs lemmes, elle doit utiliser l'analyse grammaticale afin de remplacer les verbes par leurs formes infinitives et les noms par leurs formes au singulier [15]. En effet, Un mot donné peut avoir différentes formes dans un texte, mais leur sens reste le même. Par exemple, les mots vol, volant et vole seront remplacés par leurs lemmes : vol, volant et voler selon le contexte. Cette représentation est simple mais elle peut causer une perte d'informations donnée par le contexte nécessaire à la distinction des lemmes polysémiques (possèdent plusieurs sens) et la présence de synonymes, considérés comme des lemmes différents même s'ils font référence au même concept. Voici des exemples illustratifs (voir figures 2.15 et 2.16) :

```
In [65]: applyLemme("studies")
Out[65]: 'study'
```

Figure 2.15: Exemple de lemmatisation (Anglais)

```
In [9]: applyLemme("الطلاب يدخلون القاعة")
Out[9]: 'الطلاب يدخلون القاعة'
```

Figure 2.16: Exemple de lemmatisation (Arabe)

Pour l'anglais ça marche sans faute, en arabe ça marche avec les bons ajustements mais les résultats obtenus ne sont pas satisfaisants ce qui veut dire que ces outils ne sont pas performant pour l'arabe comme pour l'anglais.

2.3.1.3. Transformation TF-IDF

- **Représentation avec les n-grammes**

Cette méthode consiste à représenter le document par des n-grammes. Le n-gramme est une séquence de n caractères consécutifs. Cette technique présente plusieurs avantages. Les n-grammes capturent automatiquement les racines des mots les plus fréquents sans passer par l'étape de recherche des racines lexicales, indépendante de la langue, les espaces sont pris en considération parce qu'en effet, la non prise en compte de ces derniers introduit du bruit.

- **Codage des termes**

Une fois choisies les composantes du vecteur représentant un texte *j*, il faut décider comment coder chaque coordonnée de son vecteur d_j .
Il existe différentes méthodes pour calculer le poids w_{kj}. Ces méthodes sont basées sur les deux observations suivantes :

- Plus le terme t_k est fréquent dans un document d_j , plus il est en rapport avec le sujet de ce document.

- Plus le terme t_k est fréquent dans une collection, moins il sera utilisé comme discriminant entre documents.

Soient $\#(t_k, d_j)$ le nombre d'occurrences du terme t_k dans le texte dj , |Tr| le nombre de documents du corpus d'apprentissage et $\#\text{Tr}(t_k)$ le nombre de documents de cet ensemble dans lesquels apparaît au moins une fois le terme t_k. Selon les deux observations précédentes, un terme t_k se voit donc attribuer un poids d'autant plus fort qu'il apparaît souvent dans le document et rarement dans le corpus complet. La composante du vecteur est codée $f(\#(t_k, d_j))$, où la fonction f reste à déterminer [21].
Deux approches triviales peuvent être utilisées. La première consiste à attribuer un poids égal à la fréquence du terme dans le document :

$$w_{kj} = \#(t_k, d_j) \qquad (1)$$

Et la deuxième approche consiste à associer une valeur booléenne :

$$w_{kj} = \begin{cases} 1 & \text{Si } ((t_k, d_j) > 1 \\ 0 & \text{Sinon} \end{cases} \quad (2)$$

- **Codage TF × IDF**

Plutôt que de représenter le vecteur en fonction de la fréquence du concept dans le document, nous utilisons le score TFIDF. Ce score permet de donner une importance au concept en fonction de sa fréquence dans le document (TF = TermFrequency) pondérée par la fréquence d'apparition du concept dans tout le corpus (IDF =Inverse Document Frequency). Ainsi un concept très spécifique au document (n'apparaissant que dans ce document) aura un score correspondant à sa fréquence d'apparition, par contre, un concept apparaissant dans tous les documents du corpus aura une pondération maximale.

La formule est la suivante :

$$TFIDF_{c,d} = TF_{c,d} . IDF_{c,d} \quad (3)$$

Avec:
- c : un concept
- d: le document

TFIDF voulant dire "TermFrequency, Inverse Document Frequency"
Soit :

$$TFIDF_{c,d} = TF_{c,d} . ((\log_2 N/DF_c) + 1) \quad (4)$$

Avec:
- c : un concept
- d: le document
- $TF_{c,d}$: la fréquence d'apparition du concept dans le document.
- DF_c : le nombre de documents du corpus contenant le concept.
- N : le nombre de documents du corpus.

Ainsi, quand DF_c est égal à 1 (concept n'apparaissant que dans ce document), le
TFIDF sera fort, quand DF_c est proche de N (concept apparaissant dans tous les documents), le TFIDF sera faible.
Après la phase d'indexation du corpus de textes, nous calculons donc, pour chaque concept dans un document, son score TFIDF. Nous verrons que, dans toutes les applications de l'indexation, nous utiliserons ce score TFIDF comme métrique de l'importance du concept dans le document.
Par contre, l'ajout d'un nouveau document dans le système nécessite de recalculer tous les scores TFIDF. Il s'avère néanmoins, que, lorsque le nombre de document est élevé, l'ajout d'un nouveau document ne modifie pas beaucoup les autres scores TFIDF. Le recalcule complet peut donc être différé [22].

- **Les plongements utilisés**

Nous avons appliqué ces plongements :
-Vec_char : Vec_char applique une segmentation (tokenisation) par caractère avant d'appliquer la TF-IDF, exemple : Hello = 'H' 'e' 'l' 'l' 'o'.
-Vec_word : Vec_word applique une segmentation (tokenisation) par mot par exemple:
Hello everybody = 'Hello' ,'everybody'.

-Vec char_wb : Char_wb applique une segmentation en tenant compte des espaces par exemple :
Hello = 'H' ' ' 'e' ' ' 'l' ' ' 'l' ' ' 'o'.

-Vec_all : Applique une concaténation des vecteurs obtenus.

2.3.2. Classification des données

Il existe plusieurs algorithmes et techniques utilisés pour la classification supervisée telles que :

- **RégressionLogistique "LR"**

Les prévisions de régression logistique sont des valeurs discrètes, c'est-à-dire un ensemble fini de valeurs (Vrai ou faux par exemple). La régression logistique convient mieux à la classification binaire. Par exemple, on peut considérer un ensemble de données où y = 0 ou 1, où 1 représente la classe par défaut. Pour illustrer on peut imaginer que l'on veuille prédire s'il pleuvra ou non. On aura 1 pour s'il pleut et 0 le cas contraire [23]. La régression logistique, propose le résultat sous forme de probabilités de la classe par défaut. Le résultat appartient donc à l'intervalle [0 :1]. C'est-à-dire qu'il est compris entre 0 et 1, vu qu'il s'agit d'une probabilité. La valeur y de sortie est générée par la transformation de la valeur x, à l'aide de la fonction logistique h (x) = 1 / (1 + e ^ -x). Un seuil est ensuite appliqué pour forcer cette probabilité dans une classification binaire.

- **Méthode des k plus proches voisins "KNN"**

La méthode des k plus proches voisins (KNN : K-Nearest Neighbors), vise à classifier une instance en fonction de ces K plus proches voisins, et ceci, en calculant une distance quelconque [24].

- **Pseudo algorithme :**

1. Choisir le nombre K : nombre des voisins.

2. Pour chaque instance test :

- Trouver la distance (Cosinus, Euclidienne …) avec toutes instances d'apprentissage.

- Trié les distances dans une liste.

- Choisir les K premier classes (étiquettes) relative au K premières distances.

- Assigner une classe à l'instance test en se basant sur la majorité des classes des K premier point.

❖ **Bernoulli Naïve Bayes "BNB"**

Bernoulli Naive Bayes est une variante de Naive Bayes. Alors, parlons d'abord de Naive Bayes en bref. Naive Bayes est un algorithme de classification de Machine Learning basé sur le théorème de Bayes qui donne la probabilité d'occurrence de l'événement. Le classifieur Naïve Bayes est un classifieur probabiliste, ce qui signifie que, étant donné une entrée, il prédit la probabilité que l'entrée soit classée pour toutes les classes. Elle est également appelée probabilité conditionnelle [25]. Dans le cas du modèle d'apprentissage Bernoulli Naïve Bayes (BNB), un document d est représenté comme un vecteur de mot (w_1,w_2, . . .,w_m) et la probabilité conditionnelle P (d|c) peut être estimée [26], selon l'hypothèse de NB par :

$$P(d|c) = \prod_{i=1}^{m} (w_i P(w_i|c) + (1-w_i)(1-P(w_i|c))) \quad (5)$$

Où m est le nombre de mots, $w_i$est une valeur booléenne qui représente la présence du i^emot dans le document d ou non, et la probabilité conditionnelle P (w_i|c) [26] est estimée par :

$$P(w_i|c) = x = \frac{\sum_{j=1}^{n} w_{ji}\delta(c_j,c)+1}{\sum_{j=1}^{n} \delta(c_j,c)+2} \quad (6)$$

Où n est le nombre de documents d'entraînement, c_j est la catégorie du j^edocument, etw_{ji} est un booléen indiquant la présence ou non du i^e mot dans lej^edocument d'entraînement.

❖ **Multinomial Naïve Bayes "MNB"**

BNB ne tient, cependant, pas compte de la fréquence d'apparition des mots dans un document, qui est une information potentiellement utile à la prédiction des catégories. C'est, justement, ce manque que vient combler Multinomial Naïve Bayes (MNB) [26]. Un document de test, d, est maintenant représenté par un Bag-Of-Words. L'ordre des mots n'est ici pas considéré, mais bien la fréquence de chacun d'eux dans le texte. Dans ce modèle, la même hypothèse de NB est faite : la probabilité du nombre d'occurrences de chaque mot dans un document est indépendante de sa position et du nombre d'occurrences des autres mots du document. Un document d est donc représenté par un vecteur de mots (w_1,w_2, . . .,w_m), et MNB estime la probabilité conditionnelle P (d|c) par :

$$P(d|c) = \left(\sum_{i=1}^{m} f_i\right)! \prod_{i=1}^{m} \frac{P(w_i|c)^{f_i}}{f_i!} \quad (7)$$

Où m est le nombre de mots, wi (i = 1, 2, . . . ,m) indique la présence du i^emot dans le document d, f_i est le nombre d'occurrences de w_i dans d, P (w_i)|c) est la probabilité conditionnelle que le mot w_i apparaisse dans la catégorie c :

$$P(w_i|c) = \frac{\sum_{j=1}^{n} f_{ji}\delta(c_j,c)+1}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ji}\delta(c_j,c)+m} \quad (8)$$

Où n est le nombre de documents d'entraînement, $c_j$est la catégorie du j^edocument, m est le nombre de mots, f_{ji} est le nombre d'occurrences du mot wi dans le j^e document d'entraînement, et δ est une fonction binaire qui vaut 1 si ses paramètres sont identiques, 0 sinon.

- **Machine à vecteurs de support ("SVM")**

SVM ou Support Vector Machines sont également des algorithmes très populaires utilisés dans l'apprentissage supervisé. Ils aident à classer et analyser les données à l'aide d'un hyperplan [27]. L'hyperplan est une ligne ou un plan, qui divise les points de données en deux catégories distinctes. Le but est de trouver un plan optimal, qui divise les deux points de données. En maximisant la marge de l'hyperplan, nous augmentons la distance entre les points de données de chaque côté. Ceci est fait jusqu'au point où les points de données sont distincts les uns des autres. Pour déterminer les marges de l'hyperplan, des vecteurs de support (points de données qui semblent se trouver sur le plan ou en sont proches) sont nécessaires.
Voici un exemple d'un problème de discrimination à deux classes. Un plan (espace à deux dimensions) dans lequel sont répartis deux groupes de points, ces points sont associés à un groupe : les points (+) pour y > x et les points (-) pour y < x. On peut trouver un séparateur linéaire évident dans cet exemple, la droite d'équation y = x. Le problème est dit linéairement séparable. (Voir figure 2.17) :

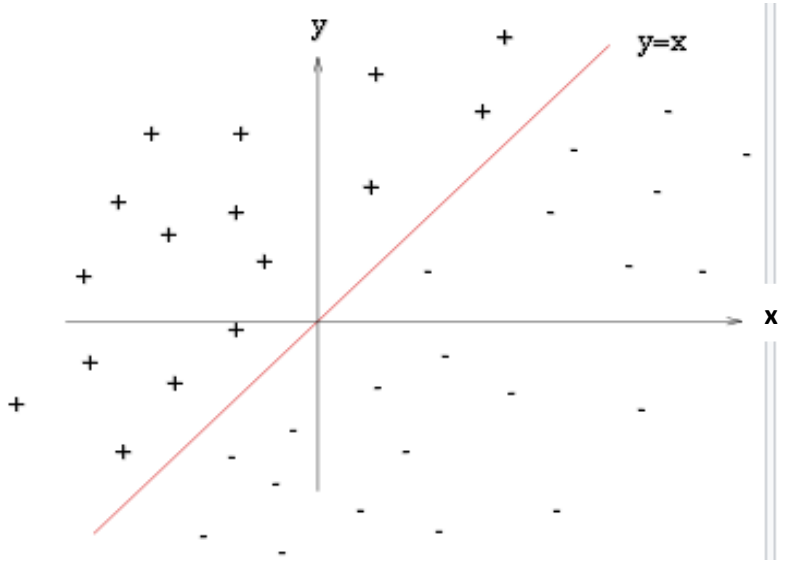


Figure 2.17 : Problème de discrimination à deux classes

❖ **Arbres de décision (DecisionTreeClassifier "DT")**

Les arbres de décision sont des arbres binaires, qui aident à la classification, qui est un type d'approche d'apprentissage supervisé [27]. Il peut avoir une complexité plus élevée en fonction du nombre de feuilles et de nœuds de l'arbre. Cet arbre est utile dans les situations «oui ou non» et «si et alors».

Un chemin, partant de la racine jusqu'à une feuille de l'arbre, constitue une règle d'affectation du type « Si condition Alors conclusion ». L'ensemble de ces règles constitue le modèle de prédiction [21]. Voici un exemple dans la (figure 2.18) :

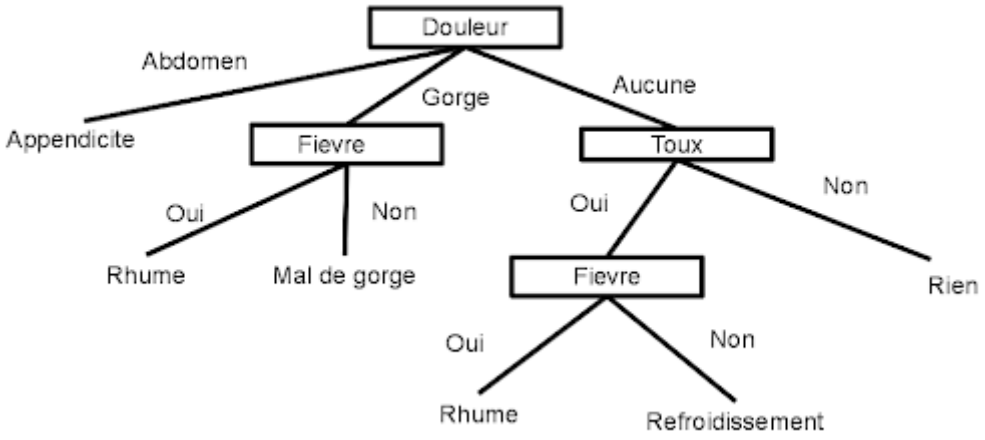


Figure 2.18: Exemple d'arbre de décision

❖ **Forêts d'arbres décisionnels (RandomFoRest "RF")**

La forêt aléatoire est une énorme collection d'arbres de décision. Donc, pour une situation, il y a de nombreux résultats possibles que la forêt aléatoire nous aide à voir. Il est différent d'un arbre de décision (car les arbres de décision sont toujours binaires et constituent une seule unité) car il a plusieurs arbres [27]. Cet algorithme aide à trouver de nouveaux modèles et possibilités pour tout, car la collection d'arbres aide à analyser les données de plusieurs manières. Il a un algorithme plus complexe qu'un arbre de décision. Par conséquent, cela consommerait beaucoup plus de puissance de calcul.

❖ **Gradient Boosting "GB"**

Le boosting construit un modèle additif de manière progressive ; il permet l'optimisation de fonctions de perte différentiables arbitraires. À chaque étape, n_classes_ les arbres de régression sont ajustés sur le gradient négatif de la fonction de perte de déviance binomiale ou multinomiale. La classification binaire est un cas particulier où un seul arbre de régression est induit [28]. Il s'agit d'une méthode de classification émettant des hypothèses qui sont au départ de moindre importance. Plus une hypothèse est vérifiée, plus son indice de confiance augmente. Ce qui prend de l'importance dans la classification [13].

2.4. Métriques d'évaluations

Il existe plusieurs métriques pour l'évaluation des algorithmes d'apprentissage, et le choix dépend entièrement du cas d'utilisation. L'évaluation consiste à utiliser des benchmarks, afin de mesurer la différence entre un résultat réel et un résultat obtenu par le ML [24]. Dans cette étude, l'accuracy, le rappel, la précision et le F-score sont utilisés.

1.7.1. Accuracy

C'est la métrique le plus utilisés et la plus intuitive pour l'évaluation des algorithmes d'apprentissage.

$$\text{Accuracy}=\frac{\text{Nombre total de documents correctement classifiés}}{\text{Nombre total de documents}} \qquad (9)$$

Parfois, cette métrique est très biaisée. Lorsque 90 % des instances appartiennent à une classe A et 10% des instances à une classe B, il est très probable que le ML (Machine Learning algorithm) prédira que toutes les instances appartiendront la classe A, ainsi l'accuracy sera de 90%.

1.7.2. Rappel

Le rappel ou sensibilité (Recall en anglais), est défini par le nombre de documents pertinents correctement prédits d'une classe i, au regard du nombre de documents total de la classe i.
Un rappel élevé, signifie que le ML a pu prédire correctement un bon nombre de documents pour chaque classe.

$$\text{Rappel} =\frac{\text{Nombre de documents correctement attribués à la classe i}}{\text{Nombre de document appartenant la classe i}} \qquad (10)$$

1.7.3. Précision

La précision est le nombre de documents pertinents retrouvés rapporté au nombre de documents total proposé par le ML. Si elle est élevée, cela signifie que la plupart des documents d'une classe i sont correctement classifiés, ainsi ce dernier peut être considéré comme « précis ».

$$\text{Précision}=\frac{\text{Nombre de documents correctement attribués à la classe i}}{\text{Nombre de document attribué a la classe i}} \qquad (11)$$

1.7.4. F-score

L'apprentissage à partir de données déséquilibrées revient souvent à effectuer un arbitrage entre le rappel et la précision du modèle. Avoir une haute précision signifie que la plupart des documents attribués à une classe i appartiennent à cette classe. Par contre, avoir un taux de rappel élevé signifie qu'on peut identifier la plupart des individus pour chaque classe du corpus (de cette manière on est capable de détecter des individus d'une classe minoritaire).
Idéalement, on voudrait réduire le nombre d'individus mal attribués à une classe i.

La différence entre le rappel et la précision se situe surtout dans la façon de considérer l'objectif :

- Soit l'établissement des classes les plus justes possibles.
- Soit l'établissement d'une classification de textes la plus juste possible.

Les deux visions sont donc plus complémentaires qu'antagonistes.

Le F-score (la F-mesure ou indice de Dice) correspond à un compromis de la précision et du rappel, donnant la performance du système. Ce compromis est donné de manière simple par la moyenne harmonique de la précision et du rappel.

$$\text{F-score} = 2.\frac{\text{Précision.Rappel}}{\text{Précesion+Rappel}} \qquad (12)$$

Conclusion

Ce chapitre est consacré à tous les processus de notre système pour les deux corpus. Nous citons aussi les étapes de prétraitement des données tweets avec une brève description sur le processus de traitement de tweets qui a composé de la suppression des ponctuations, la normalisation, la suppression des mots vides, Stemmer, Postagger et la lemmatisation.

Nous avons tenté dans ce chapitre de définir la classification et de présenter les principales techniques de classification automatique supervisée, utilisées pour classer des unités textuelles en groupes.

Chapitre 3 : Expérimentation et comparaison des résultats

3.1. Introduction

Dans ce chapitre, nous allons présenter les datasets utilisé et les résultats de différents tests établis en utilisant les techniques présentées dans le chapitre précédent.

3.2. Datasets utilisés et évaluation des résultats

Dans ce travail, deux datasets ont été utilisés pour comparer nos approches pour la classification des tweets : anglais et arabe.

3.2.1. Corpus anglais

Le corpus d'apprentissage (en anglais, « dataset ») est un élément essentiel à la construction d'un système de classification automatique. Plusieurs sites web proposent gratuitement des corpus d'apprentissage et de tests bien structurés pour réaliser des travaux portant sur la classification automatique des documents écrits en caractères latins [18]. Pour mesurer les performances d'un classifieur, nous avons utilisé la fonction train_test_split pour diviser aléatoirement le corpus en deux : Corpus d'entrainement 80%, Corpus de test 20%.

Nous avons travaillé avec deux dataset :

A. Dataset : Drug Review

L'ensemble de données fournit des avis aux patients sur des médicaments spécifiques ainsi que des conditions connexes et une cote de 10 étoiles reflétant la satisfaction globale des patients. Les données ont été obtenues en explorant des sites de revue pharmaceutique en ligne [29]. Les Tableaux 3.1 et 3.2 résument les attributs et les Statistiques de ce corpus :

Tableau 3.1: Résumé des attributs de « Drug Review »

Nom d'attribut	Type d'attribut	Description
DrugName	Catégorique	Nom de médicament
Condition	Catégorique	Symptôme
Review	Texte	Tweet de patient
Rating	Numérique	Évaluation des patients 10 étoiles
Date	Date	Date d'entrée de tweet
UsefulCount	Numérique	Nombre d'utilisateurs qui ont trouvé le tweet est utile

Tableau 3.2 : Statistiques de « Drug Review »

	Train	Test	Totale
Nombre de tweets	161 297	53 766	215 063
Nombre de mots	13 814 570	4 601 711	18 416 281
Nombre de caractères	73 991 166	24 641 188	98 632 354
Nombre de mots par tweet (moyenne)	85.64	85.58	85.63
Nombre de caractère par tweet (moyenne)	458.72	458.30	458.62

B. Dataset : Trip Advisor

L'ensemble de données TripAdvisor est un ensemble de données que nous avons exploré à partir du site Web de TripAdvisor [30]. Il contient des notes pour les points d'intérêts dans la région du Tyrol du Sud en Italie qui sont marquées avec des situations contextuelles décrites par la conjonction de conditions contextuelles provenant de trois facteurs contextuels, à savoir le type (par exemple, couple, voyage en famille ou voyage d'affaires), le mois (par exemple, janvier, Février) et l'année (par exemple, 2015, 2014) du voyage. De plus, cet ensemble de données possède des attributs d'utilisateur bien définis (par exemple, l'emplacement de l'utilisateur, le type de membre) et les attributs de points d'intérêts (par exemple, le type d'élément, les commodités, la localité de l'élément). Le tableau 3.3 fournit les statistiques du corpus Trip Advisor.

Tableau 3.3 : Statistiques de « Trip Advisor »

Nombre de tweets	7 154
Nombre de mots	929 580
Nombre de caractères	6 163 817
Nombre de mots par tweet (moyenne)	129.93
Nombre de caractère par tweet (moyenne)	861.59

C. Dataset : Expérience 1

- Nous avons combiné les deux dataset « Drug Review » et « TripAdvisor » pour avoir un corpus qui contient des tweets médicales et non médicales.

- Nous avons eu un problème de ressource de calcul. Pour pallier à cette problématique, nous avons commencé par un petit corpus (data 1.1) et augmenter la taille pas à pas jusqu'à la limite de calcul de la machine (data 1.2).

Nous avons utilisé deux datasets, le premier data contient 17 154 tweets et le deuxième contient 47 154 tweets. Le tableau 3.4 représente les statistiques de ces deux.

Tableau 3.4 : Statistiques de data1.1 et data 1.2

	Data1.1			**Data 1.2**		
	Train	**Test**	**Totale**	**Train**	**Test**	**Totale**
Nombre de tweets	13 723	3 431	17 154	37 723	9 431	47 154
Nombre de mots	1 326 254	330 500	1 656 754	3 391 368	833 647	4 225 015
Nombre de caractères	7 220 029	1 799 339	9 019 368	18 276 096	4 498 154	22 774 250
Nombre de mots par tweet (moyenne)	96.64	96.32	96.58	89.90	88.39	89.6
Nombre de caractère par tweet (moyenne)	526.12	524.43	525.78	484.48	476.95	482.97

D. Dataset : Expérience 2

Nous avons travaillé dans l'expérience 2 seulement avec le corpus « Drug Review » puisque l'objectif est d'identifier les symptômes. Puisque ce travail consistait une première étape dans un travail qui va se suivre, on s'est limité :

- Le premier data à 20 symptômes les plus fréquents qui est équivalents à faire une classification de 20 classes (data 2.1).
- Le deuxième data à 30 symptômes les plus fréquents qui est équivalents à faire une classification de 30 classes (data 2.2).

Le tableau 3.5 représente les statistiques de ces deux :

Tableau 3.5 : Statistiques de data 2.1 et data 2.2

	Data2.1			Data 2.2		
	Train	**Test**	**Totale**	**Train**	**Test**	**Totale**
Nombre de tweets	28 192	7 048	35 240	43 024	10 757	53 781
Nombre de mots	2 564 020	641 251	3 205 271	3 892 265	970 129	4 862 394
Nombre de caractères	13 691 360	3 422 378	17 113 738	20 798 342	5 189 363	25 987 705
Nombre de mots par tweet (moyenne)	90.94	90.98	90.95	90.46	90.18	90.41
Nombre de caractère par tweet (moyenne)	485.64	485.58	485.63	483.41	482.41	483.21

E. Dataset : Expérience 3

Nous avons travaillé dans l'expérience 3seulement avec le corpus « Drug Review » puisque l'objectif est d'identifier les médicaments. Puisque ce travail consistait une première étape dans un travail qui va se suivre, on s'est limité :

- Le premier data à 30 médicaments les plus fréquents qui est équivalents à faire une classification de 30 classes (data 3.1).
- Le deuxième data à 50 médicaments les plus fréquents qui est équivalents à faire une classification de 50 classes (data 3.2).

Le tableau 3.6 représente les statistiques de ces deux :

Tableau 3.6 : Statistiques de data 3.1 et data 3.2

	Data 3.1			Data 3.2		
	Train	**Test**	**Totale**	**Train**	**Test**	**Totale**
Nombre de tweets	14 720	3 680	18 400	19 478	4 870	24 348
Nombre de mots	1 448 257	363 872	1 812 129	1 902 856	467 487	2 370 343
Nombre de caractères	7 725 056	1 943 540	9 668 596	10 135 624	2 496 698	12 632 322
Nombre de mots par tweet (moyenne)	98.38	98.87	98.48	97.69	95.99	97.35
Nombre de caractère par tweet (moyenne)	524.8	528.13	525.46	520.36	512.66	518.82

F. Dataset : Expérience 4

Nous avons travaillé dans l'expérience 4 avec le corpus qui contient la collection des deux dataset « Drug Review et « Trip Advisor » puisque l'objectif est de faire une reconnaissance multi labels (médicale ou non, symptômes et médicaments) à la fois. On s'est limité : à 30 symptômes les plus fréquents qui est équivalents à faire une classification de 30 classes, et 50 médicaments les plus fréquents qui est équivalents à faire une classification de 50 classes. Le tableau 3.7 représente les statistiques de ce data :

Tableau 3.7: Statistiques de data 4

	Test
Nombre de tweets	11 354
Nombre de mots	1 129 788
Nombre de caractères	6 059 436
Nombre de mots par tweet (moyenne)	99.50
Nombre de caractère par tweet (moyenne)	533.68

3.2.1.1 Etude comparative entre les algorithmes de classification

Nous avons réalisé 5 expériences. Nous avons testé différents algorithmes pour la classification des tweets afin de choisir celui avec les meilleurs résultats pour notre approche. Nous exposons dans ce qui suit les résultats de l'étude comparative de 8 algorithmes de classification supervisée. Le tableau 3.8 représente les classifieurs utilisé avec ses paramètres.

Tableau 3.8 : Les algorithmes utilisés avec ses paramètres

Classifieurs	**Paramètre**
k plus proches voisins (KNN)	n_neighbors=5, metric='minkowski'
Machine à vecteurs de support (LSVM)	loss='squared_hinge', C=1.0, max_iter=1000)
Regression Logistic (LR)	C=1.0, max_iter=100
Arbres de décision (DT)	criterion='gini', splitter='best', max_depth=None
Multinomial NaïveBayes (MNB)	alpha=1.0, fit_prior=True
Bernoulli Naïve Bayes (BNB)	alpha=1.0, binarize=0.0
Forêts d'arbres décisionnels (RF)	n_estimators=100, criterion='gini', max_depth=None
Le Boosting (GB)	loss='deviance', learning_rate=0.1, n_estimators=100, criterion='friedman_mse'

- **Expérience 1**

La première expérience concerne la classification des tweet médicaux.

- **Data1.1**

Dans les tableaux 3.9 et 3.10 on va présenter les résultats de KNN et MNB successivement :

Tableau 3.9 : Résultats de KNN expérience 1 data 1.1

KNN	Prétraitement	Vec_all	Vec_word	Vec_char	Char_wb
		(Stemmer+Lemmatizer+ Postagger)			
Accuracy	Suppression des mots vides+ Elimination des ponctuations	60.44%	96.32%	51.67%	64.67%
F1-score		64,00%	96,00%	60,00%	67,00%

Tableau 3.10 : Résultats de MNB expérience 1 data 1.1

MNB	Prétraitement	Vec_all	Vec_word	Vec_char	Char_wb
		(Stemmer+Lemmatizer+ Postagger)			
Accuracy	Suppression des mots vides+ Élimination des ponctuations	99.59%	99.76%	99.70%	99.70%
F1-score		100%	100%	100%	100%

Dans les tableaux 3.9 et 3.10, on remarque que ***vec_word*** a une influence positive importante sur KNN**,** et moins d'impact sur MNB ; il améliore les résultats comparés aux autres.

Le tableau 3.11 représente les résultats de BNB :

Tableau 3.11 : Résultats de BNB expérience 1 data 1.1

BNB	Prétraitement	Vec_all	Vec_word	Vec_char	Char_wb
		(Stemmer+Lemmatizer+ Postagger)			
Accuracy	Suppression des mots vides+ Élimination des ponctuations	96.18%	78.86%	97.43%	98.57%
F1-score		96,00%	81,00%	97,00%	99,00%

Par contre dans le tableau 3.11 l'introduction du ***vec word*** a diminué les performances de BNB.

Les tableaux 3.12, 3.13, 3.14 et 3.15 représente les résultats de GB, DT, LR et LSVM successivement :

Tableau 3.12 : Résultats de GB expérience 1 data 1.1

GB	**Prétraitement**	**Vec_all**	**Vec_word**	**Vec_char**	**Char_wb**
		(Stemmer+Lemmatizer+ Postagger)			
Accuracy	**Suppression des mots vides+ Elimination des ponctuations**	99.94%	98.51%	100.0%	99.94%
F1-score		100%	99%	100%	100%

Tableau 3.13 : Résultats de DT expérience 1 data 1.1

DT	**Prétraitement**	**Vec_all**	**Vec_word**	**Vec_char**	**Vec**
		Stemmer+Lemmatizer+Postagger			
Accuracy	**Suppression des mots vides+ Élimination de ponctuation**	99,94%	99,56%	100,00%	99,94%
F1-score		100,00%	97,00%	100,00%	100,00%

Tableau 3.14 : Résultats de LR expérience 1 data 1.1

LR	**Prétraitement**	**Vec_all**	**Vec_word**	**Vec_char**	**Char_wb**
		(Stemmer+Lemmatizer+ Postagger)			
Accuracy	**Suppression des mots vides+ Elimination des ponctuations**	99.82%	99.56%	99.73%	99.70%
F1-score		100%	100%	100%	100%

Tableau 3.15 : Résultats de LSVM expérience 1 data 1.1

LSVM	Prétraitement	Vec_all	Vec_word	Vec_char	Char_wb
		(Stemmer+Lemmatizer+ Postagger)			
Accuracy	**Suppression des mots vides+ Elimination des ponctuations**	100.0%	99.82%	99.97%	100.0%
F1-score		100%	100%	100%	100%

Dans les tableaux 3.9, 3.12, 3.13, 3.14 et 3.15, nous constatons que le ***vec word*** a une influence importante pour le KNN contrairement il n'a aucun impact avec GB, DT, LR et LSVM.

Le tableau 3.16 représente les résultats de RF :

Tableau 3.16 : Résultats de RF expérience 1 data 1.1

RF	Prétraitement	Vec_all			
		Stemmer	Lemmatizer	Postagger	-
Accuracy	Suppression des mots vides				99.06%
	Elimination des ponctuations	99.8%	99,83%	99,74%	98.95%
	-	99.38%	99.15%	98.89%	99.79%
F1-score	Suppression des mots vides				99%
	Elimination des ponctuations	100%	100%	100%	100%
	-	100%	99%	99%	99%

		Vec_word				
		Stemmer	Lemmatizer	Postagger	-	Stemmer +Lemmatizer
Accuracy	Suppression des mots vides				95.83%	
	Elimination des ponctuations	98.95%			96.41%	
	-	96.82%	97.55%	96.73%	98.89%	99,04%
F1-score	Suppression des mots vides				96%	
	Elimination des ponctuations	99%			99%	
	-	99%	98%	97%	98%	99%

		Vec_char				
		Stemmer	Lemmatizer	Postagger	-	Stemmer +Lemmatizer
Accuracy	Suppression des mots vides				98.97%	
	Elimination des ponctuations	99.74%	99,77%		98.92%	
	-	99.06%	98.51%	99.03%	99.70%	99,04%
F1-score	Suppression des mots vides				99%	
	Elimination des ponctuations	100%	100%		100%	
	-	100%	99%	99%	99%	100%

		Char_wb				
		Stemmer	Lemmatizer	Postagger	-	Postagger +Lemmatizer
Accuracy	Suppression des mots vides				99.32%	
	Elimination des ponctuations	99,88%			99.21%	
	-	99.56%	99.65%	99.50%	99.27%	99,88%
F1-score	Suppression des mots vides				99%	
	Elimination des ponctuations	100%			99%	
	-	100%	100%	100%	99%	100%

Dans le tableau 3.16 nous remarquons que :

- Le ***vec all*** et ***vec char*** ont des bons résultats avec ***lemmatisation*** et ***élimination des ponctuations*** par rapport ***Stemmer*** et ***Postagger***.
- Le ***vec word*** a un bon résultat avec ***Stemmer*** et ***l'élimination des ponctuations*** par rapport ***Lemmatizer*** et ***Postagger***.
- Les résultats obtenus par RF sont meilleurs avec le ***char wb***, ***Stemmer*** et ***l'élimination des ponctuations*** par rapport ***vec word***, ***vec char***, ***vec all***.
- D'après nos résultats le meilleur classifieur est le LSVM.

➢ **Data 1.2:**

Nous avons appliqué les mêmes démarches précédentes sur « data 1.2 ».

Dans les tableaux 3.17 et 3.18 on va présenter les résultats de KNN et MNB successivement :

Tableau 3.17 : Résultats de KNN expérience 1 data 1.2

KNN	Prétraitement	Vec_all	Vec_word	Vec_char	Vec char_wb
		Stemmer+Lemmatizer+Postagger			
Accuracy	Suppression des mots vides+ Élimination de ponctuation	68,91%	95,70%	50,99%	83,31%
F1-score		64,00%	96,00%	46,00%	81,00%

Tableau 3.18 : Résultats de MNB expérience 1 data 1.2

MNB	Prétraitement	Vec_all	Vec_word	Vec_char	Vec char_wb
		Stemmer+Lemmatizer+Postagger			
Accuracy	Suppression des mots vides+Élimination de ponctuation	85,33%	94,13%	88,97%	89,92%
F1-score		92,00%	95,00%	92,00%	92,00%

Dans les tableaux 3.17 et 3.18, nous remarquons que le ***Vec word*** toujours a une influence positive sur KNN, et aussi sur MNB mais n'est pas forte.

Le tableau 3.19 représente les résultats de BNB :

Tableau 3.19 : Résultats de BNB expérience 1 data 1.2

BNB	Prétraitement	Vec_all	Vec_word	Vec_char	Vec
		Stemmer+Lemmatizer+Postagger			
Accuracy	Suppression des mots vides+Elimination de ponctuation	95,67%	86,27%	99,01%	99,46%
F1-score		96,00%	92,00%	99,00%	99,00%

Dans le tableau 3.19, nous remarquons que le ***Vec word*** toujours a une influence négative sur BNB il diminue les résultats.

Les tableaux 3.20, 3.21, 3.22 et 3.23 représente les résultats de LR, GB, DT, et LSVM successivement :

Tableau 3.20 : Résultats de LR expérience 1 data 1.2

LR	Prétraitement	Vec_all	Vec_word	Vec_char	Vec
		Stemmer+Lemmatizer+Postagger			
Accuracy	Suppression des mots vides+Elimination de ponctuation	99,92%	99,65%	99,74%	99,80%
F1-score		100%	100%	100%	100%

Tableau 3.21 : Résultats de GB expérience 1 data 1.2

GB	Prétraitement	Vec_all	Vec_word	Vec_char	Vec
		Stemmer+Lemmatizer+Postagger			
Accuracy	Suppression des mots vides+Elimination de ponctuation	99,97%	99,24%	100,00%	99,97%
F1-score		100%	99%	100%	100%

Tableau 3.22: Résultats de DT expérience 1 data 1.2

DT	Prétraitement	Vec_all	Vec_word	Vec_char	Vec
		Stemmer+Lemmatizer+Postagger			
Accuracy	Suppression des mots vides+Elimination de ponctuation	99,98%	98,33%	100,00%	99,98%
F1-score		100,00%	98,00%	100,00%	100,00%

Tableau 3.23 : Résultats de LSVM expérience 1 data 1.2

LSVM	Prétraitement	Vec_all	Vec_word	Vec_char	Vec char_wb
		Stemmer+Lemmatizer+Postagger			
Accuracy	**Suppression des mots vides+Elimination de ponctuation**	99,97%	99,92%	99,96%	99,98%
F1-score		100%	100%	100%	100%

A partir les tableaux 3.17, 3.18, 3.19, 3.20, 3.21, 3.22 et 3.23, nous constatons que :

- Le ***Vec word*** a une influence importante pour le KNN et MNB contrairement il n'a aucun impact avec LR, GB, DT et LSVM.
- Les résultats de KNN GB LR LSVM DT BNB RF dans data 1.2 sont meilleurs que le data1.1 alors nous concluons que quand on a plus des donnés on obtient des meilleurs résultats.
- Contrairement avec MNB qui nous a donné des meilleurs résultats avec le data 1.1 qui a moins de donnés.

Le tableau 3.24 représente les résultats de RF :

Tableau 3.24: Résultats de RF expérience1 data1.2

RF	Prétraitement	Vec_all					
		Stemmer	**Lemmatizer**	**Postagger**	**Postagger+Stemmer**	**Postagger+Lemmatizer**	**-**
	Suppression des mots vides						99,58%
	Elimination de ponctuation						99,61%
Accuracy	-	99,92%	99,71%	99,67%	99,69%	99,67%	99,75%
	Suppression des mots vides						100,00%
	Elimination de ponctuation						100,00%
F1-score	-	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
		Vec_word					
	Suppression des mots vides						98,98%
	Elimination de ponctuation				99,01%		98,98%
Accuracy	-	98,96%	99,00%	98,96%	99,00%	98,94%	98,97%
	Suppression des mots vides						99,00%
	Elimination de ponctuation						99,00%
F1-score	-	99,00%	99,00%	99,00%		99,00%	99,00%
		Vec_char					
	Suppression des mots vides						99,68%
	Elimination de ponctuation			99,73%			99,67%
Accuracy	-	99,78%	99,70%	99,75%	99,76%	99,72%	99,74%
	Suppression des mots vides						100,00%
	Elimination de ponctuation			100,00%			100,00%
F1-score	-	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
		Vec char_wb					
	Suppression des mots vides						99,93%
	Elimination de ponctuation			99,94%			99,92%
Accuracy	-	99,78%	99,90%	99,92%		99,69%	99,92%
	Suppression des mots vides				99,00%		100,00%
	Elimination de ponctuation			100,00%	99,00%		100,00%
F1-score	-	100,00%	100,00%	100,00%		100,00%	100,00%

Dans le tableau 3.24, nous remarquons que :

- ✓ Le ***Vec_all*** et ***Vec_char*** ont des bons résultats avec ***Stemmer*** par rapport la ***lemmatisation*** et ***Postagger***.
- ✓ Le ***Vec word*** a un bon résultat avec ***Stemmer+Postagger*** et ***l'élimination des ponctuations***.
- ✓ Les résultats obtenus par RF sont meilleurs avec le ***Vec char_wb***, ***Postagger*** et ***l'élimination des ponctuations*** par rapport ***Vec Word***, ***vec char*** et ***vec all***.
- ✓ D'après nos résultats le meilleur classifieur est le DT.

- **Expérience 2**

La deuxième expérience concerne l'identification des symptômes seulement pour les tweets médicaux. Nous avons travaillé avec les meilleures classifieurs de l'expérience 1 « LSVM et DT ».

- ➢ **Data2.1**

Les tableaux 3.25 et 3.26 montrent les résultats obtenus :

Tableau 3.25: Résultats de LSVM expérience 2 data 2.1

LSVM	**Prétraitement**	**Vec_char**
		(Stemmer+Lemmatizer+Postagger)
Accuracy	**Suppression des mots vides+ Elimination des ponctuations**	84.87%
F1-score		85%

Tableau 3.26 : Résultats de DT expérience 2 data 2.1

DT	Prétraitement	Vec_char							
		Stemmer	Lemmatizer	Postagger	-	Stemmer+ Lemmatizer	Stemmer + Postagger	Postagger+ Lemmatizer	Stemmer+Postagger + Lemmatizer
Accuracy	Suppression des mots vides	68,88%	69.21%	68,94%	68,76%	68.38%	69.01%	68.54%	68.44%
	Elimination des ponctuations	68,74%	68,87%	68,88%	68.63%	68.87%	68.28%	68.88%	68.94%
	Suppression des mots vides+Elimination des ponctuations	68.64%	68.95%	68.57%	68,60%	68.81%	68.98%	68.54%	68.84%
	-	68,69%	68,67%	68,56%	68.47%	68,57%	68,79%	68,74%	68.82%
F1-score	Suppression des mots vides	69%	69%	69%	69%	69%	69%	69%	69%
	Elimination des ponctuations	69%	69%	69%	69%	69%	68%	69%	69%
	Suppression des mots vides+Elimination des ponctuations	69%	69%	69%	69%	69%	69%	69%	69%
	-	69%	69%	69%	69%	69%	69%	69%	69%

Dans le tableau 3.26, nous remarquons que le meilleur résultat est obtenu en appliquant le ***Lemmatizer*** avec ***l'élimination des mots vides***.

- **Data 2.2**

Nous avons travaillé avec les mêmes classifieurs et les commandes de data 1.

Les tableaux 3.27 et 3.28 montrent les résultats obtenus :

Tableau 3.27: Résultats de LSVM expérience 2 data 2.2

LSVM	Prétraitement	Vec_char
		Stemmer+Lemmatizer+Postagger
Accuracy	Suppression des mots vides+Elimination des ponctuations	81,59%
F1-score	Suppression des mots vides+Elimination des ponctuations	82%

Tableau 3.28: Résultats de DT expérience 2 data 2.2

DT	Prétraitement	Vec_char		
		Stemmer	Lemmatizer	Postagger
Accuracy	Suppression des mots vides	65,54%	65,92%	65,88%
	Elimination des ponctuations	66,01%	65,65%	65.57%
	Suppression des mots vides+Elimination des ponctuations	65,74%	65,65%	65,46%
	-	65,73%	65,68%	65,68%
F1-score	Suppression des mots vides	66%	66%	66%
	Elimination des ponctuations	66%	66%	66%
	Suppression des mots vides+Elimination des ponctuations	66%	66%	66%
	-	66%	65%	65%

Dans le tableau 3.28, nous remarquons que le meilleur résultat est obtenu en appliquant le ***Stemmer*** avec ***Elimination des ponctuations***. En plus, dans le tableau 3.27 nous remarquons que le LSVM donne des meilleures performances que DT. Ce résultat est justifiable car le modèle LSVM est bien connu par sa robustesse [31]. Nous remarquons que les résultats de l'expérience 1 est meilleur que l'expérience 2, tandis que dans l'expérience 1 nous avons travaillé avec deux classes et le LSVM nous donne **99,98%** comme meilleur résultat (tableau 3.23), par contre dans l'expérience 2 quand on a travaillé avec 20 classes de symptôme le LSVM nous donne **84,87%** (tableau 3.25), quand on a travaillé avec 30 classes de symptôme nous donne **81,59%** (tableau 3.27), donc nous concluons que quand on a plus des classes quand les résultats vont diminuer, c'est l'une des raisons rend le travail plus compliquées.

Dans l'expérience1 les données étaient simples, dans cette expérience les erreurs ont commencé à apparaître à cause de la complexité du problème d'identification des symptômes (Expérience 2) par rapport à la classification des tweet (Expérience 1).

- **Expérience 3**

La troisième expérience est l'identification des médicaments seulement pour les tweets médicaux. Nous avons choisi les meilleures commandes qui ont les bons résultats à partir l'expérience 2. Nous avons travaillé avec les mêmes classifieurs de la deuxième expérience.

- **Data3.1**

Les tableaux 3.29 et 3.30 montrent les résultats obtenus :

Tableau 3.29: Résultats de LSVM expérience 3 data 3.1

LSVM	**Prétraitement**	**Vec_char**
		(Stemmer+Lemmatizer+Postagger)
Accuracy	**Suppression des mots vides+ Elimination des ponctuations**	53,26%
F1-score		54%

Tableau 3.30: Résultats de DT expérience 3 data3.1

DT	Prétraitement	Vec_char							
		Stemmer	Lemmatizer	Postagger	-	Stemmer+ Lemmatizer	Stemmer+ Postagger	Postagger+ Lemmatizer	Stemmer+Postagger+ Lemmatizer
Accuracy	Suppression des mots vides		40,95%	40,90%				41,14%	
	Elimination des ponctuations	41,28%			40,92%	40,54%	41,01%	41,28%	41,09%
	Suppression des mots vides+ Elimination des ponctuations	40,46%						41,28%	
	-				41,01%	41,17%	41,17%		40,76%
F1-score	Suppression des mots vides		41%	41%				41%	
	Elimination des ponctuations	41%			41%	41%	41%	41%	41%
	Suppression des mots vides+ Elimination des ponctuations	40%						41%	
	-				41%	41%	54%		41%

Dans le tableau 3.30 nous remarquons que le meilleur résultat est obtenu en appliquant le ***Stemmer*** avec ***l'élimination des ponctuations*** et aussi en appliquant le ***Lemmatizer+Postagger*** avec ***l'élimination des ponctuations+l'élimination des mots vides*** tandis qu'on a obtenus de **41,28%**.

- **Data3.2**

Les tableaux 3.31 et 3.32 montrent les résultats obtenus :

Tableau 3.31: Résultats de LSVM expérience 3 data 3.2

LSVM	**Prétraitement**	**Vec_char**
		Stemmer+Lemmatizer+ Postagger
Accuracy	**Suppression des mots vides+Elimination des ponctuations**	48,35%
F1-score		49%

Tableau 3.32: Résultats de DT expérience 3 data 3.2

DT	Prétraitement	Vec_char							
		Stemmer	**Lemmatizer**	**Postagger**	**-**	**Stemmer+ Lemmatizer**	**Stemmer+ Postagger**	**Postagger+ Lemmatizer**	**Stemmer+ Postagger+ Lemmatizer**
Accuracy	**Suppression des mots vides**		34,84%	35,11%				35,27%	
	Elimination des ponctuations	35,60%			35,64%	36,22%	35,31%	35,31%	35,19%
	Suppression des mots vides+ Elimination des ponctuations	35,37%					35,31%		
	-				34,96%	35,17%	35,52%	34,90%	35,03%
F1-score	**Suppression des mots vides**		35%	35%				35%	
	Elimination des ponctuations	35%			36%	35%	35%		35%
	Suppression des mots vides+ Elimination des ponctuations	35%					35%	35%	
	-				35%	35%	35%	35%	35%

Dans le tableau 3.32 nous remarquons que le meilleur résultat est obtenu en appliquant le ***Lemmatizer+ Stemmer*** avec ***l'élimination des ponctuations*** tandis qu'on a obtenue de **36,22%**.

Quand on a travaillé avec 30 classes de médicaments le LSVM nous donne **53,26%** (tableau 3.29), quand on a travaillé avec 50 classes de médicaments le LSVM nous donne **48,35%** (tableau 3.31). Donc nous pouvons déduire que lorsque la complexité du problème augmente (dans ce cas le nombre des classes), les performances diminuent.

- **Expérience4**

La quatrième expérience est pour faire une reconnaissance multilabels (médicale ou non, symptômes et médicaments) à la fois. Nous avons travaillé avec le meilleur classifieurs de l'expérience 3 (LSVM). Nous avons choisi les commandes qui ont réalisé les meilleurs résultats durant l'expérience 1 et 2 et qui sont similaires.

Le tableau 3.33 montre les résultats obtenus :

Tableau 3.33: Résultats de LSVM expérience 4

LSVM	**Prétraitement**	**Suppression des mots vides+ Elimination des ponctuations**
		(Stemmer+Lemmatizer+ Postagger)
Review-label (exp1)	**F1-score**	100,00%
Review-condition (exp2)	**F1-score**	70%
Review-drugname (exp3)	**F1-score**	52%
Review-Label-Condition-Drug	**F1-score = (F1exp1+F1exp2+F1exp3)/3**	74%

Dans le tableau 3.33, nous remarquons que quand on a travaillé avec 2 classes le LSVM nous donne **100%** pour l'identification des tweets, avec 30 classes de symptômes le LSVM nous donne **70%** et avec 50 classes de médicaments le LSVM nous donne **52%**, donc nous concluons que quand on a plus des classes quand les résultats vont diminuer, c'est l'une des raisons rend le travail plus compliqué.

- **Expérience 5**

La cinquième expérience est pour faire une reconnaissance multilabels (médicale ou non, symptômes et médicaments) à la fois, nous avons utilisés le même dataset de l'expérience 4. Nous avons travaillé avec le meilleur classifieur de l'expérience 3 (LSVM). Nous avons choisi les commandes qui ont réalisé les meilleurs résultats durant l'expérience 1 et 2 et qui sont distincts.

Le tableau 3.34 montre les résultats obtenus :

Tableau 3.34: Résultats de LSVM expérience 5

LSVM	Prétraitement		Review-Label (Exp 1)	Review-Condition (Exp 2)	Review-Drug (Exp 3)	Review-Label-Condition-Drug (Exp 5)
			F1-score	F1-score	F1-score	F1-score = (F1Exp1+F1Exp2+F1Exp)/3
Vec_char	Elimination des ponctuations	Stemmer + Lemmatizer + PosTagger	100%	85%	50%	78.33%
Vec Char_wb	Suppression des mots vides		100%	84%	53%	79%
	Suppression des mots vides+ Elimination des ponctuations	PosTagger+ Stemmer	100%	82%	53%	78.33%
		Stemmer + Lemmatizer	100%	83%	55%	79.33%
		PosTagger+ Lemmatizer	100%	82%	53%	78.33%
		Stemmer + Lemmatizer	100%	83%	54%	79%

Dans le tableau 3.34, nous remarquons que le meilleur résultat obtenu est **79.33%** quand on a appliqué le ***vec char_wb*** avec le ***Stemmer+Lemmatizer*** et ***suppression des mots vides+élimination des ponctuations***. Nous avons obtenu des résultats identiques **78.33 %** pour ***Vec_char*** avec ***Stemmer + Lemmatizer + Postagger*** et ***élimination des ponctuations*** et ***Vec Char_wb*** avec (***Postagger+Stemmer ou Postagger+Lemmatizer***) et ***Suppression des mots vides + Elimination des ponctuations***.

3.2.2. Corpus arabe

Dans le cas de la langue arabe, malheureusement, les corpus sont vraiment rares, et on n'a pas pu trouver un disponible gratuitement. Cela peut être dû au fait que le nombre de travaux portant sur la classification automatique des documents écrits en caractères arabes est beaucoup moins inférieur à celui des travaux réalisés sur des documents écrits en caractères latins. Pour cela, nous étions obligés de concevoir notre propre corpus (dont une partie destinée pour l'apprentissage et une autre pour les tests) en utilisant twitter comme ressources. Pour mesurer les performances d'un classifieur nous avons utilisé la fonction train_test_split pour diviser aléatoirement le corpus en deux : Corpus d'entrainement 80%, Corpus de test 20%.

Le corpus se compose de 309 644 tweets depuis le compte Twitter entre le 4 avril 2020 (premier message) et le 13 avril 2020, répartis-en 4 catégories, ces statistiques sont représentées dans le tableau 3.35 :

Tableau 3.35 : Statistique de corpus arabe

Catégorie	Nombre de tweets
Covid	90 812
E-learning	3 914
Cuisine	208 068
Télémédecine	6 850
	309 644

Les statistiques de corpus utilisé sont représentées dans le tableau 3.36:

Tableau 3.36 : Statistiques de corpus arabe

	Data1		
	Train	**Test**	**Totale**
Nombre de tweets	247 715	61 929	309 644
Nombre de mots	3 694 832	925 237	4 620 069
Nombre de caractères	24 583 079	6 159 669	30 742 748
Nombre de mots par tweet (moyenne)	14.91	14.94	14.92
Nombre de caractère par tweet (moyenne)	99.23	99.46	99.28

Nous avons fait 2 expériences :

- **Expérience 1**

La première expérience est pour détecter si les tweets sont médicaux ou non. Nous avons utilisé le dataset qui nous avons créée, nous avons choisi les 16 meilleures commandes qui ont les bons résultats à partir l'expérience 2 de corpus anglais et nous avons travaillé avec les deux meilleurs classifieurs que nous les avons obtenus durant la première expérience de corpus anglais LSVM DT. Les résultats obtenus sont donnés par les tableaux 3.37 et 3.38:

Tableau 3.37: Résultats de LSVM expérience 1 Arabe

LSVM	**Prétraitement**	**Vec_char**
		(Stemmer+Lemmatizer+ Postagger)
Accuracy	**Suppression des mots vides+Élimination des ponctuations**	99,89%
F1-score		100%

Tableau 3.38 : Résultats de DT expérience 1 Arabe

DT	Prétraitement	Vec_char							
		Stemmer	Lemmatizer	Postagger	-	Stemmer+ Lemmatizer	Stemmer+ Postagger	Postagger+ Lemmatizer	Stemmer+ Postagger+ Lemmatizer
Accuracy	Suppression des mots vides		99,53%	99,54%				99,56%	
	Elimination des ponctuations	99,48%				100.0%	99,48%	99,49%	99,46%
	Suppression des mots vides+ Elimination des ponctuations	99,54%					99,53%	99,53%	
	-	99,47%			99,45%	99,49%	99,48%		99,47%
F1-score	Suppression des mots vides		100%	100%				100%	
	Elimination des ponctuations	99%				100%	99%	99%	99%
	Suppression des mots vides+ Elimination des ponctuations	100%				99%	99%	100%	
	-	99%			99%		99%		99%

Dans le tableau 3.38, nous remarquons que le meilleur résultat est obtenu en appliquant le ***Lemmatizer+Stemmer*** avec ***l'élimination des ponctuations*** tandis qu'on a obtenue de **100%**. Le DT donne des meilleurs résultats que le LSVM.

- **Expérience 2**

La deuxième expérience est pour détecter les topiques soit : Covid, Cuisine, Télémédecine, E-learning. Nous avons travaillé avec les mêmes commandes et les mêmes classifieurs de l'expérience précédente. Les résultats obtenus sont donnés par les tableaux 3.39 et 3.40 :

Tableau 3.39: Résultats de LSVM expérience 2 Arabe

LSVM	**Prétraitement**	**Vec_char**
		(Stemmer+Lemmatizer+Postagger)
Accuracy	**Suppression des mots vides+Elimination des ponctuations**	99,81%
F1-score		100%

Tableau 3.40: Résultats de DT expérience 2 Arabe

DT	Prétraitement	Vec_char							
		Stemmer	Lemmatizer	Postagger	-	Stemmer+ Lemmatizer	Stemmer+ Postagger	Postagger+ Lemmatizer	Stemmer+Postagger +Lemmatizer
Accuracy	Suppression des mots vides		99,27%	99,31%				99,29%	
	Elimination des ponctuations	99,25%				99,29%	99,26%	99,28%	99,25%
	Suppression des mots vides+Elimination des ponctuations	99,31%					99,30%	99,32%	
	-	99,25%			99,29%	99,30%	99,26%		99,29%
F1-score	Suppression des mots vides		99%	99%				99%	
	Elimination des ponctuations	99%				99%	99%	99%	99%
	Suppression des mots vides+Elimination des ponctuations	99%					99%	99%	
	-	99%			99%	99%	99%		99%

Dans le tableau 3.40, nous remarquons que le meilleur résultat est obtenu en appliquant le ***Postagger+Stemmer*** avec ***Suppression des mots vides*** et ***l'élimination des ponctuations*** tandis qu'on a obtenue de **99,32%**.

Le LSVM réalise des meilleurs résultats que le DT, plus précisément, quand on a travaillé avec 2 classes le LSVM nous donne **99,89%** (tableau 3.37), quand on a travaillé avec 4 classes le LSVM nous donne **99,81%** (tableau 3.39). De ce fait, nous concluons qu'il n'y a pas une différence remarquable entre un problème de 4 classes et un problème binaire.

Conclusion

Dans ce chapitre, plusieurs tests expérimentaux ont été effectués sur deux dataset Anglais et Arabe. Ces tests ont été réalisés avant et après le prétraitement.

Nous constatons que les résultats de l'expérience 1 est meilleur que l'expérience 2, tandis que dans l'expérience 1 nous avons travaillé avec deux classes**,** par contre dans l'expérience 2 quand on a travaillé avec 20 classes de symptôme, les performances sont diminuées. Donc nous concluons que quand on a plus des classes quand les résultats vont diminuer.

Après un prétraitement, les algorithmes de classification ont présenté une amélioration d'accuracy et de F1-score.

Parmi tous les classifieurs, nous concluons que LSVM est le meilleur dans les deux corpus.

Chapitre 4 : Implémentation et réalisation

4.1. Introduction

Dans ce chapitre nous allons présenter les outils et le langage utilisés ; puis nous illustrerons quelques interfaces de notre application.

4.2. Outils et langage utilisé

Nous avons utilisé les outils suivants :

- **Flask**

Flask est un framework open-source de développement web en Python. Son but principal est d'être léger, afin de garder la souplesse de la programmation Python, associé à un système de templates [32].

- **WampServer**

WampServer (acronyme de Windows exploitant le serveur MySQL PHP) est une plate-forme de développement Web pour Windows exploitant le serveur Web Apache, le langage de scripts PHP et le SGBD MySQL [33]. Elle intègre également, entre autres, Phpmyadmin pour gérer facilement les bases de données.

Avec l'installation réalisée de WampServer, le serveur Web Apache et le serveur de base de données MySQL sont sur une même machine. Les commandes SQL contenues dans les scripts PHP seront exécutées, pour le serveur MySQL, par l'hôte « local host ».

Nous avons utilisé le langage python :

- **Python**

Nous avons utilisé Python 3.7 pour réaliser nos expériences. Ce langage a été conçu par Guido van Rossum à la fin des années 1980 en tant que membre de l'Institut national de recherche en mathématiques et en informatique [34]. Initialement, il a été conçu comme une réponse au langage de programmation ABC qui a également été mis en avant aux Pays-Bas. Parmi les principales caractéristiques de Python par rapport au langage ABC, Python avait une gestion des exceptions et était ciblé pour le système d'exploitation Amoeba.

- **Bibliothèques utilisées**

- **Nltk 3.5**

Est un ensemble d'outils TALN en langage Python. L'outil propose un accès à plus de 100 corpus de textes, parmi lesquels des textes en anglais, portugais, polonais, néerlandais, catalan et basque [6]. De plus, le kit peut effectuer le traitement de différents textes, comme l'étiquetage morpho-syntaxique, l'arbre syntaxique, la segmentation (tokenisation en anglais, ce qui constitue souvent la première étape du TALN) et la

synthèse de texte. Le kit d'outils TALN comporte également une introduction à la programmation et une documentation détaillée. Il est ainsi bien adapté aux étudiants, doctorants et chercheurs.

- **Scikit-Learn 0.23.2**

C'est une bibliothèque d'apprentissage statistique dans le langage de programmation python, basée sur d'autres bibliothèques python : NumPy, SciPy et matplotlib. Au début Scikit-learn était un projet « Google summer of code » de David Cournapeau en 2007.

En 2010, l'INRIA, l'institut français de recherche en informatique et en automatisme, a commencé à développer ce projet et a publié la première version le 1er février 2010. Le projet repose aujourd'hui sur un effort mondial en code source ouvert rassemblant plus de 200 contributeurs. Scikit-learn fournit des algorithmes pour les tâches d'apprentissage statistique, notamment la classification, la régression, la réduction de dimension et le clustering [35]. L'application des algorithmes d'apprentissage automatique n'était pas à la portée des utilisateurs qui pourraient en tirer le plus grand profit : chercheurs biologistes, climatologues, physiciens expérimentaux d'où vient l'intérêt de Scikit-learn grâce à sa simple documentation. De nombreux algorithmes de Scikit-learn sont rapides et adaptables à tous les ensembles de données, les développeurs peuvent donc les adapter aux méthodes choisies en ne modifiant que quelques lignes de code.

4.3. Interfaces d'application

Nous avons présenté dans cette partie quelques interfaces de notre application. Pour mieux comprendre nos interfaces prenant les exemples suivants :

4.3.1. Connexion et inscription (Login and Register)

La première interface vise l'authentification des visiteurs, Ils doivent s'authentifier par le nom d'utilisateur et le mot de passe (Voir la figure 4.1) :

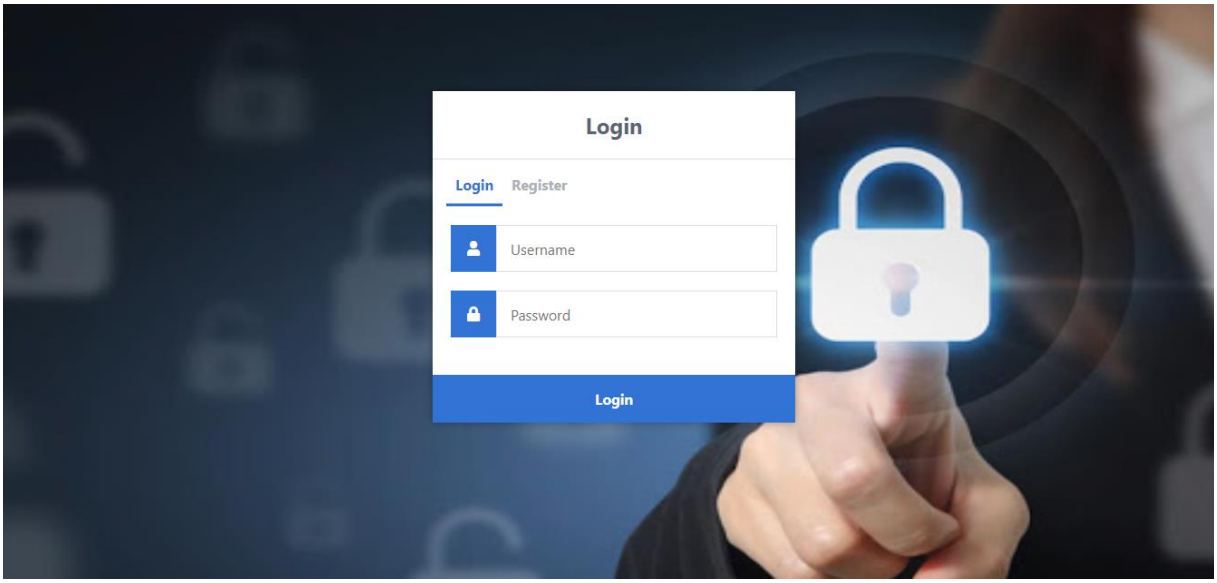


Figure 4.1: Interface login

Si l'utilisateur n'a pas un compte il faut qu'il créer le à partir de l'interface suivant (Voir la figure 4.2) :

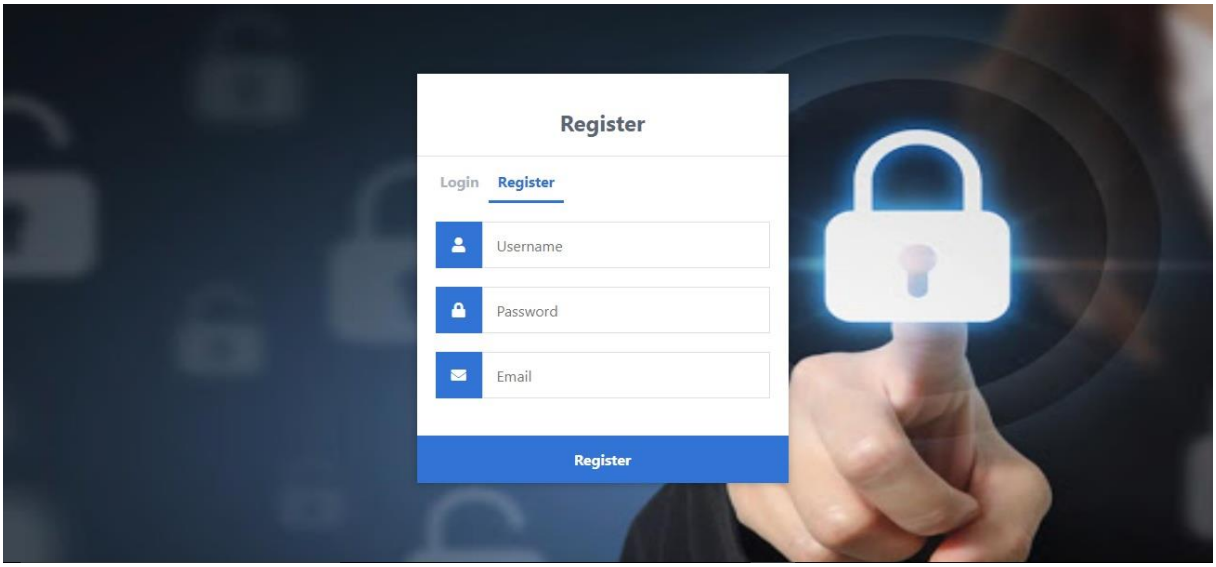


Figure 4.2: Interface register

Quand il connecté, une deuxième interface apparaît.

4.3.2. Accueil (Home)

Cette interface contient une page "à propos de nous (about us)" et deux boutons pour accéder aux deux applications. Pour déconnecter cliquer sur « log out » (Voir figure 4.3) :

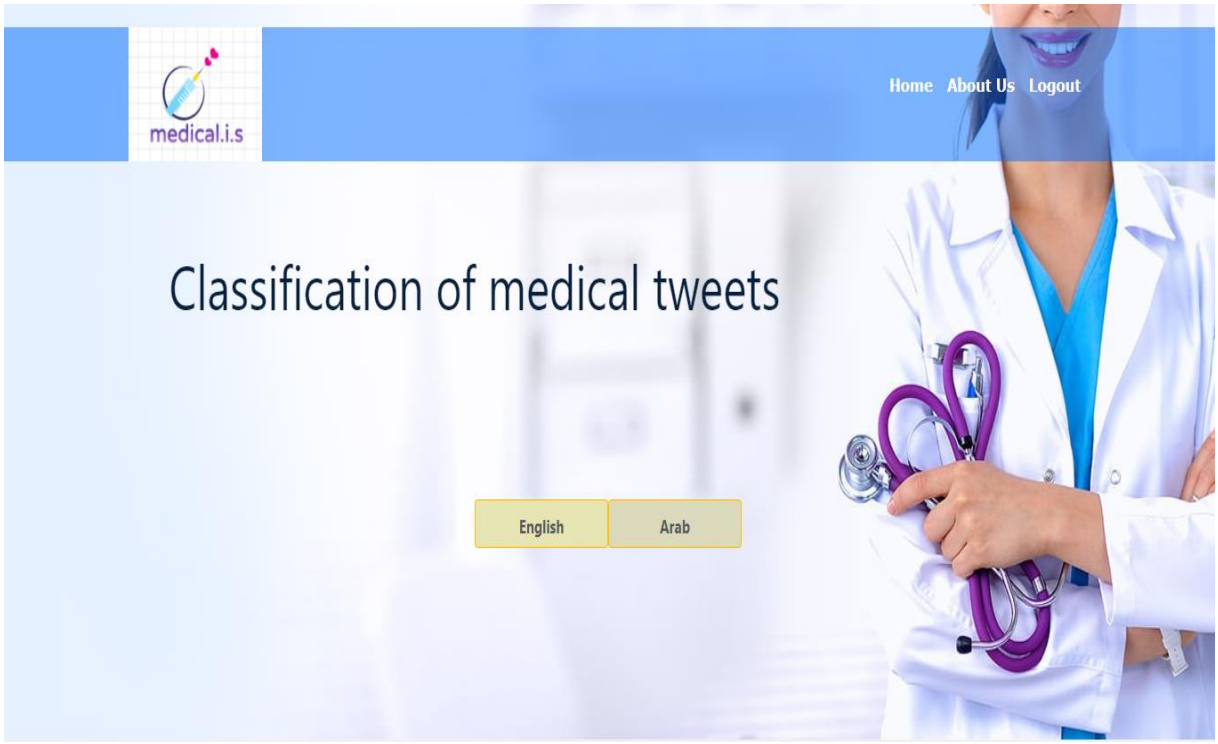


Figure 4.3: Interface home

En cliquant sur le premier bouton « English » qui est pour la langue anglaise, une interface qui s'affiche qui contient trois sous applications et un tableau avec des exemples pour faciliter l'opération de recherche. Nous avons appliqué cet exemple (Voir figure 4.4) :

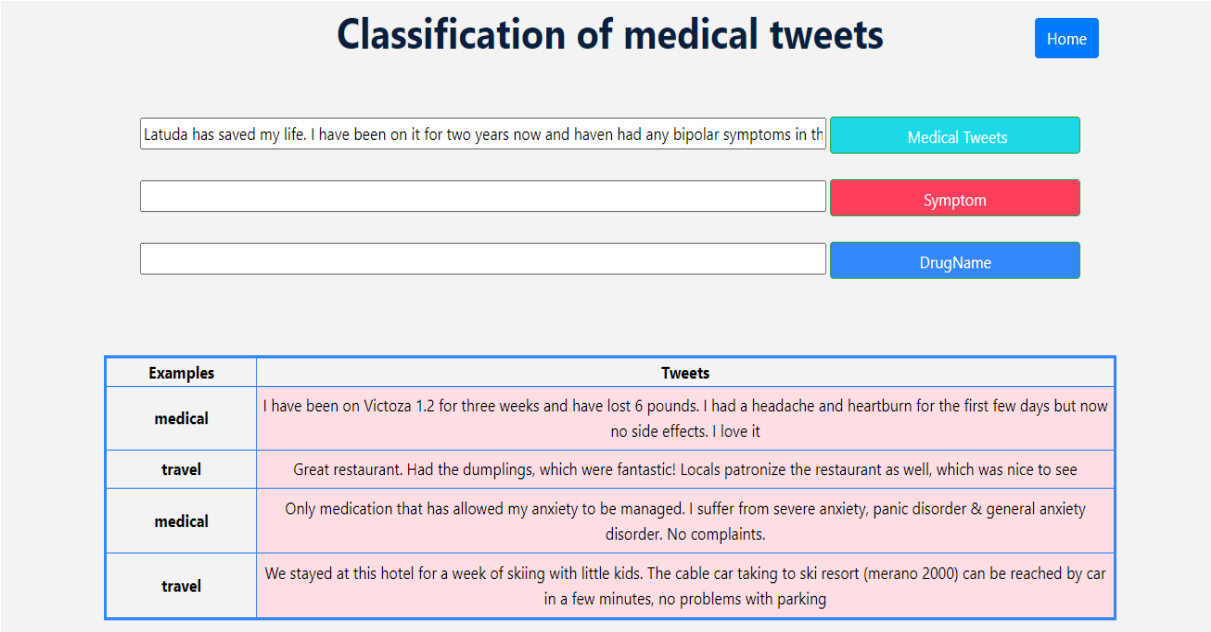


Figure 4.4: Exemple de tweet médicale pour l'identification de tweet

En cliquant sur le bouton « médical tweets », le type de ce tweet sera affiché (Voir figure 4.5) :

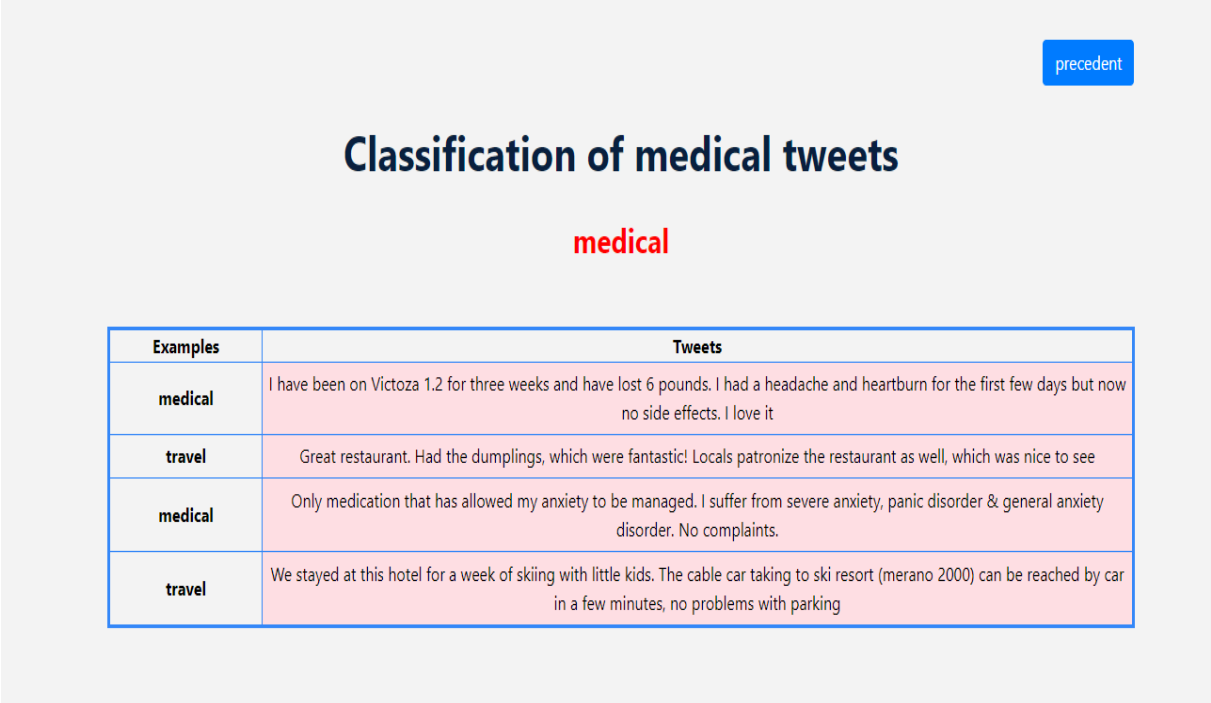


Figure 4.5: Identification de tweet médicale

En cliquant sur le bouton « precedent » pour revenir à l'interface précédente (Voir figure 4.6) :

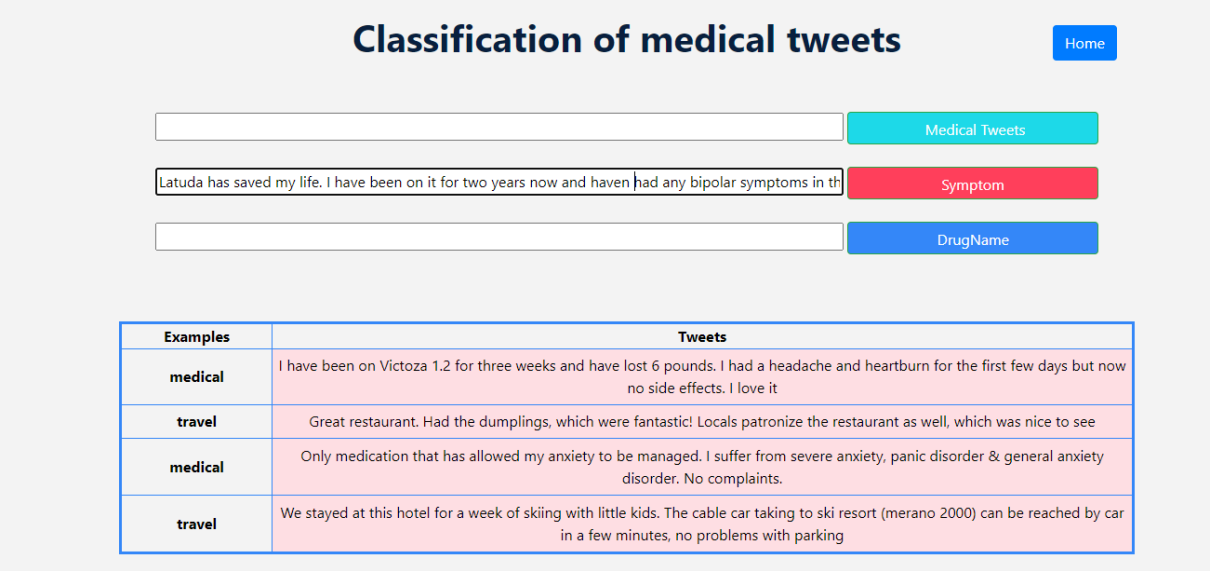


Figure 4.6: Exemple de tweet médicale pour l'identification de symptôme

En cliquant sur le bouton « symptom », le symptôme de ce tweet sera affiché (Voir figure 4.7) :

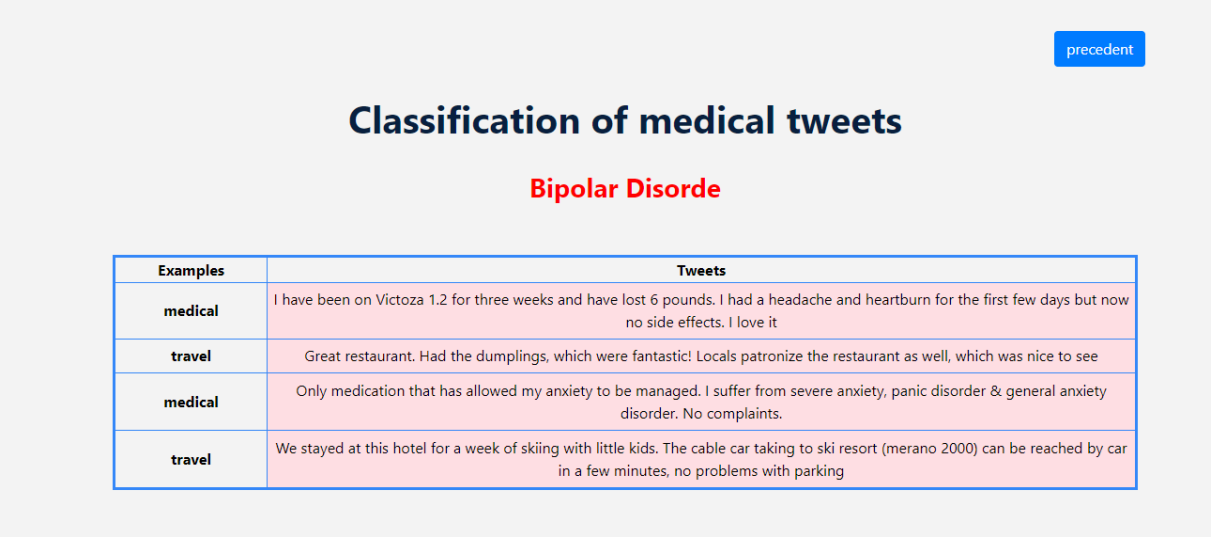


Figure 4.7: Identification de symptôme

En cliquant sur le bouton « precedent » pour revenir à l'interface précédente (Voir figure 4.8) :

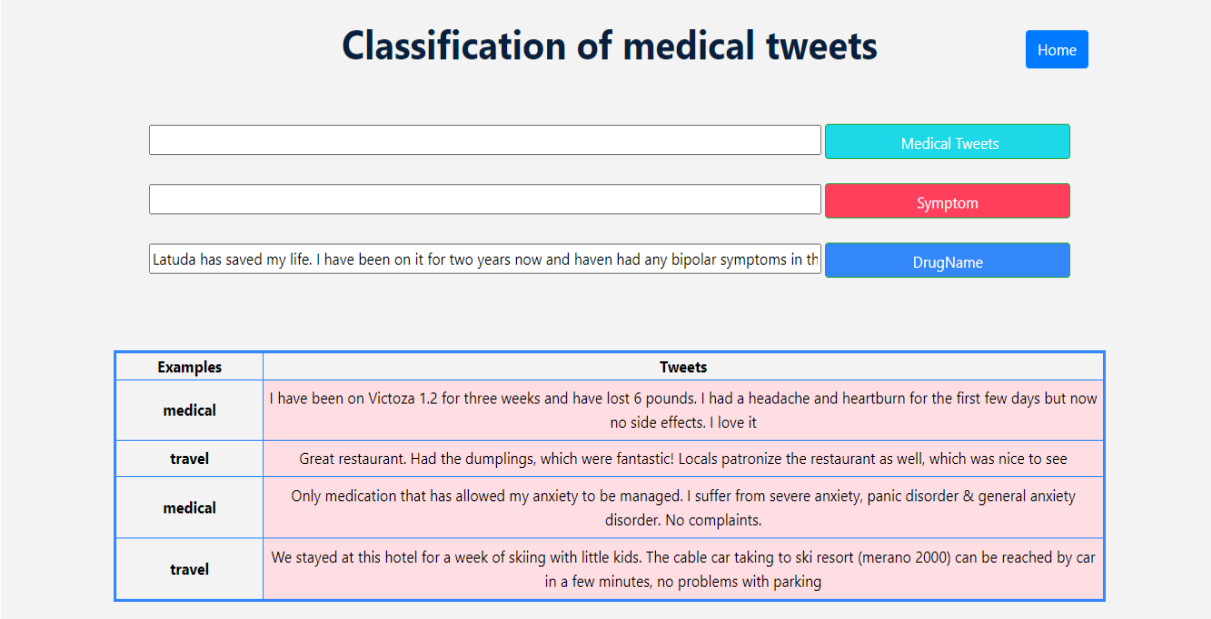


Figure 4.8 : Exemple de tweet médicale pour l'identification de médicament

Pour voir le nom de médicament en cliquant sur le bouton « DrugName », le nom sera affiché (Voir figure 4.9) :

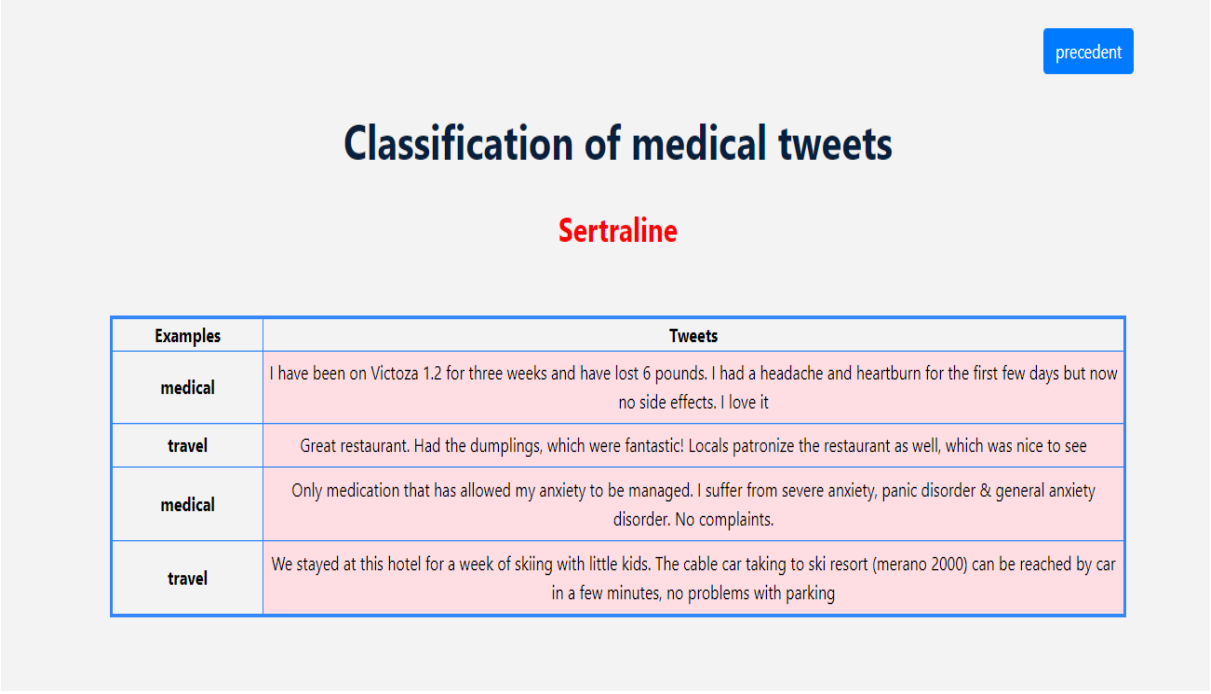


Figure 4.9: Identification de médicament

En cliquant sur le bouton « precedent » pour revenir à l'interface précédente.

Nous avons appliqué autre exemple non médicale (Voir figure 4.10) :

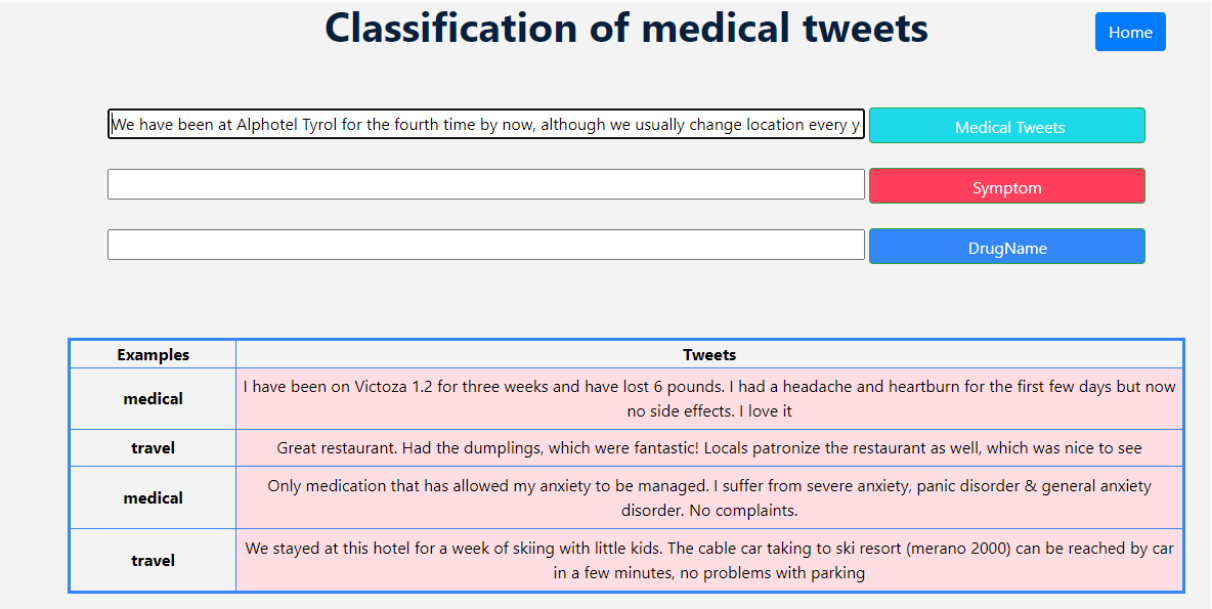


Figure 4.10: Exemple de tweet non médicale

En cliquant sur le bouton « medical tweets ». Le type de ce tweet sera affiché (Voir figure 4.11) :

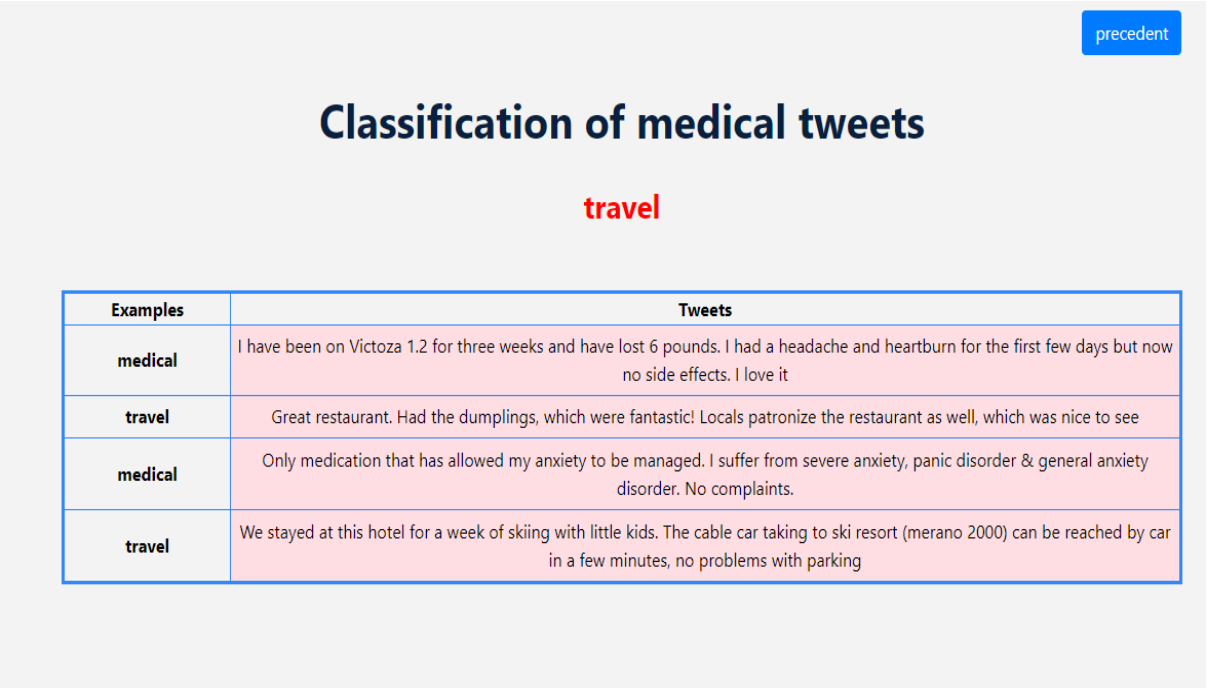


Figure 4.11: Identification de tweet non médicale

Et si nous voulons voir ce symptôme et médicament, il affiche que ce n'est pas une maladie (Voir figure 4.12) :

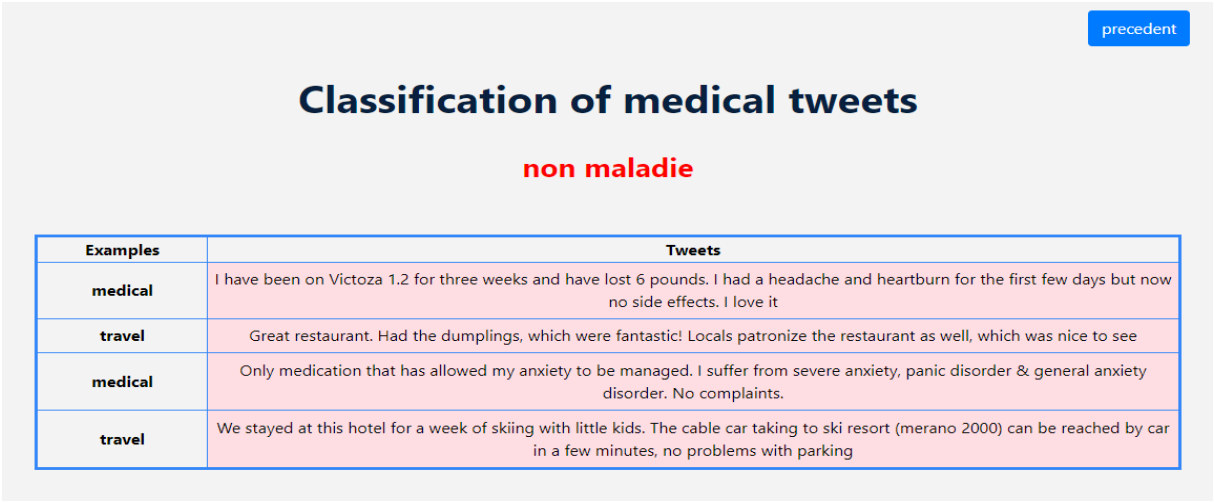


Figure 4.12: Identification de symptôme et médicament de tweet non médicale

On revient à la page d'accueil par une clique sur le bouton « home ».

En cliquant sur le deuxième bouton « Arab » qui est pour la langue arabe, une interface qui s'affiche qui contient deux sous applications et un tableau avec des exemples pour faciliter l'opération de recherche. Nous avons appliqué cet exemple (Voir figure 4.13) :

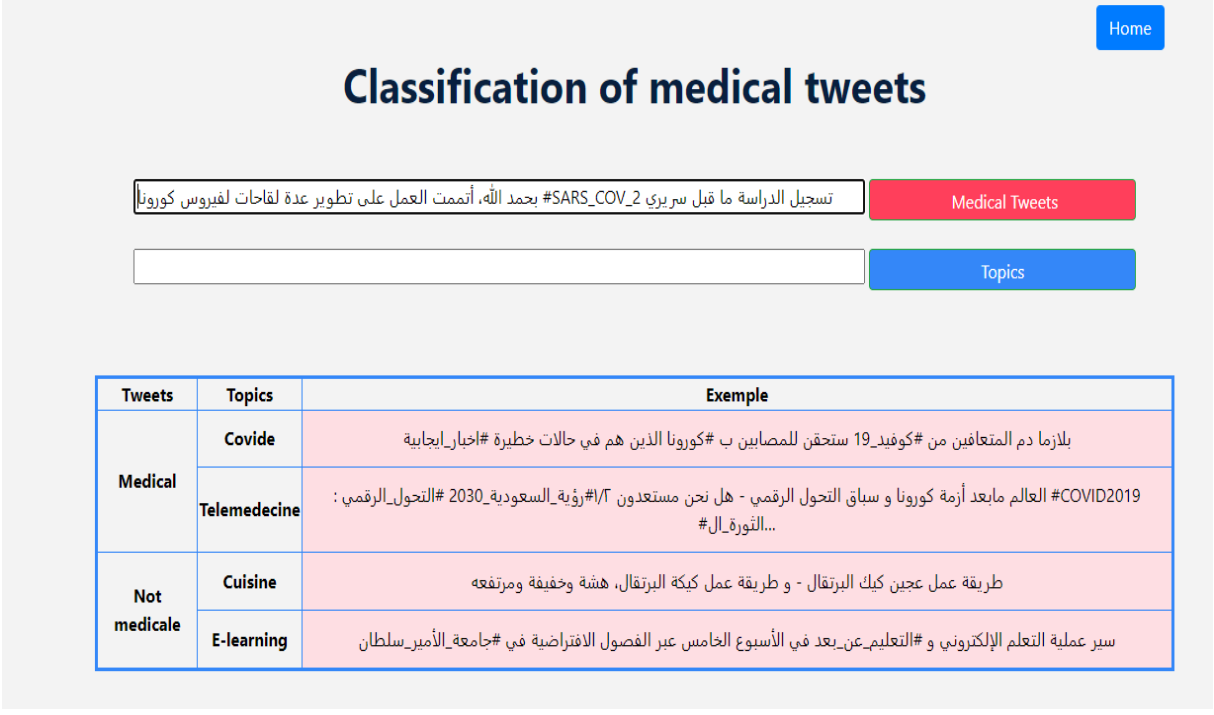


Figure 4.13 : Exemple de tweet médicale en arabe

En cliquant sur le bouton « medical tweets », le type de ce tweet sera affiché (Voir figure 4.14) :

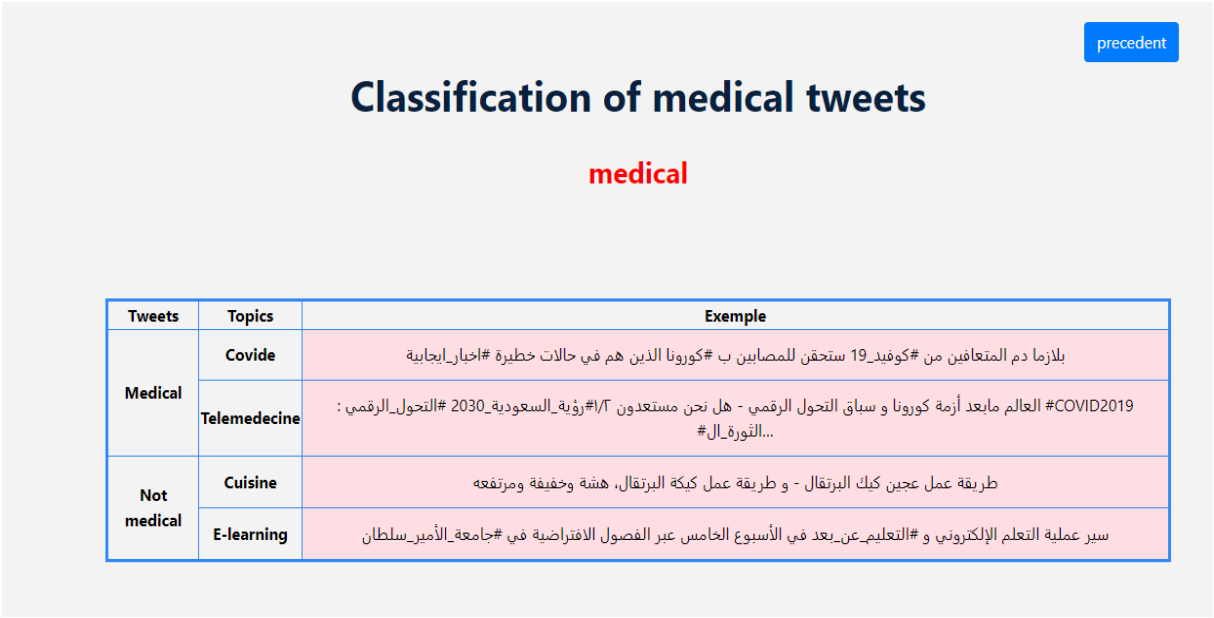


Figure 4.14: Identification de tweet médicale en arabe

En cliquant sur le bouton « topics ».la topique de ce tweet sera affichée (Voir figure 4.15) :

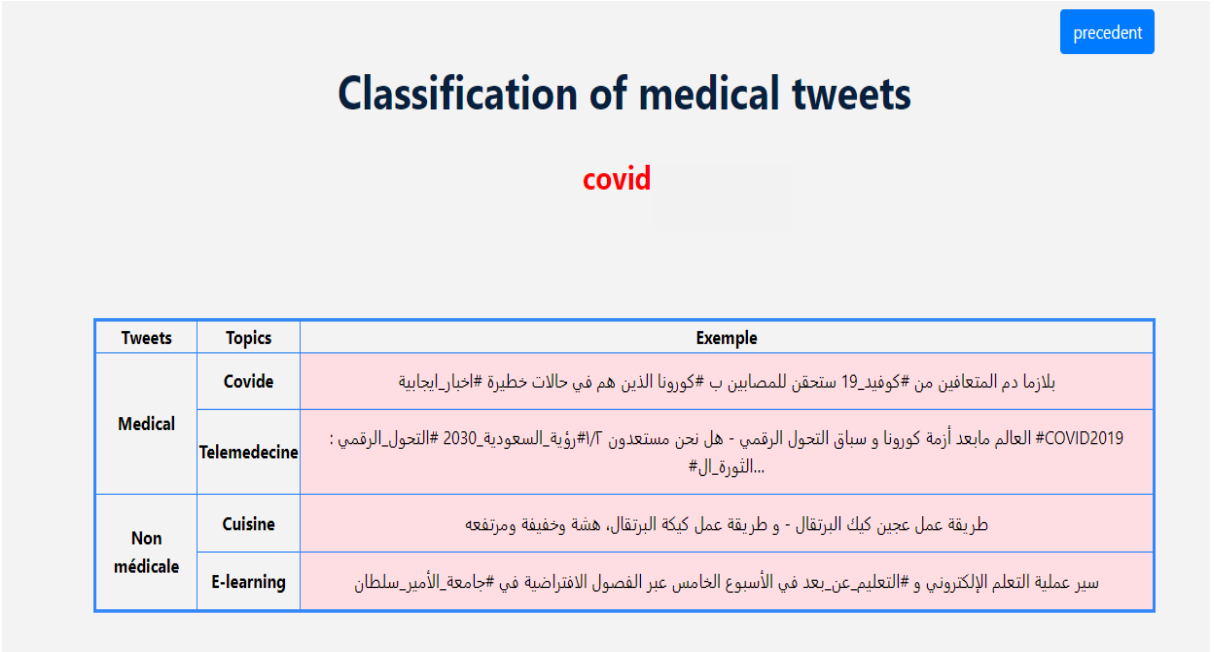


Figure 4.15: Identification de topique de tweet médicale

Nous avons appliqué autre exemple non médicale (Voir figure 4.16) :

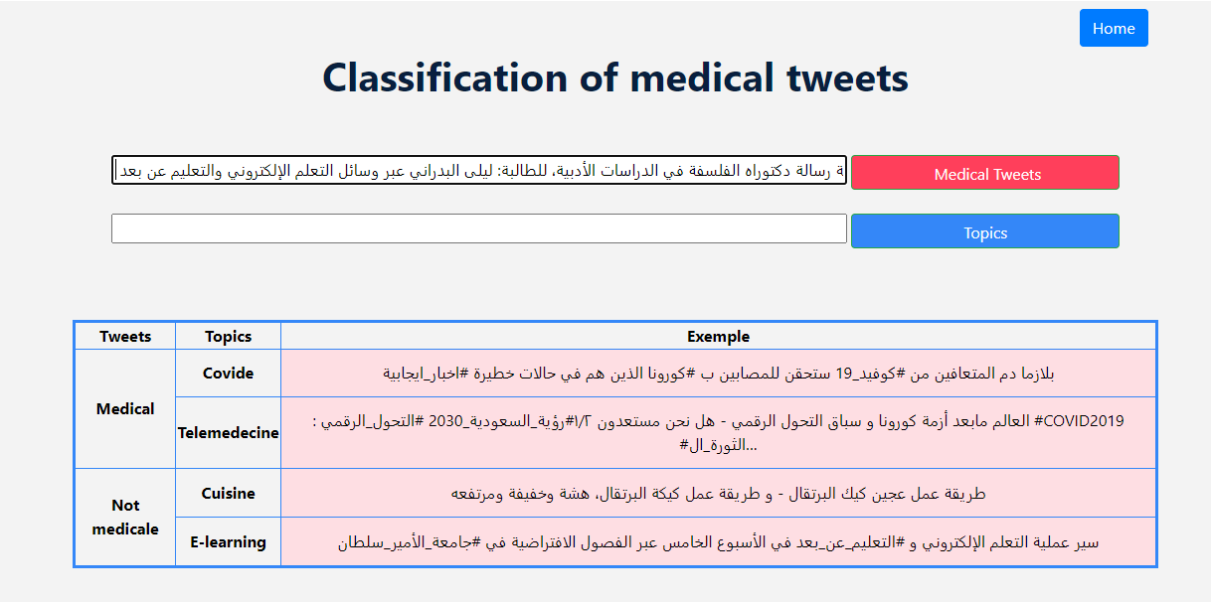


Figure 4.16: Exemple de tweet non médicale en arabe

En cliquant sur le bouton « medical tweets », le type de ce tweet sera affiché (Voir figure 4.17) :

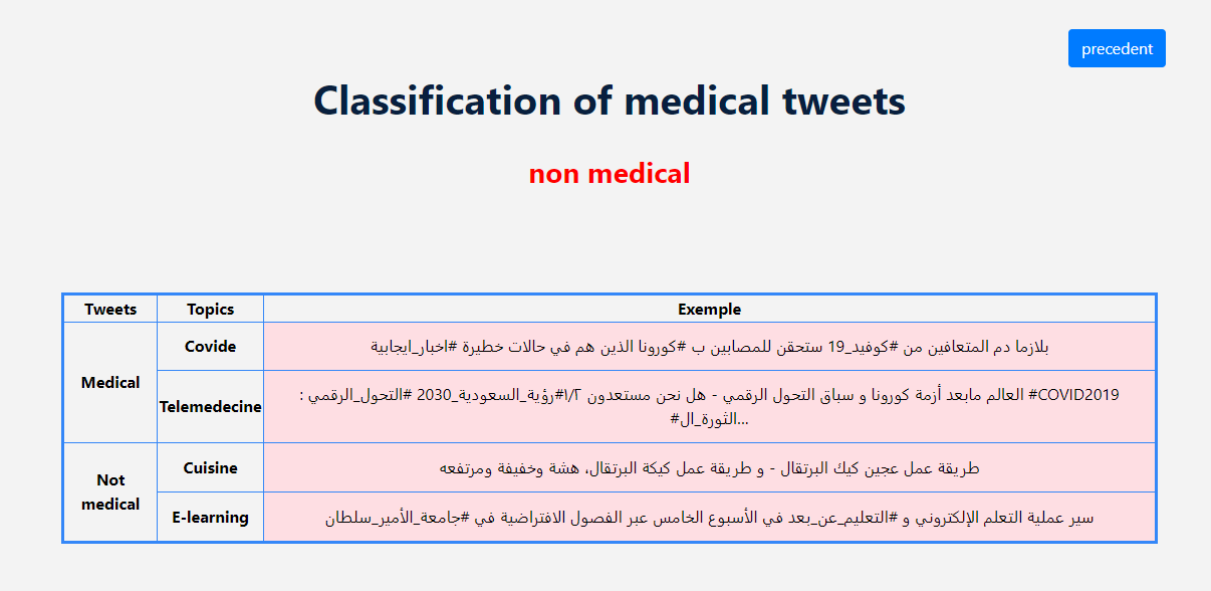


Figure 4.17: Identification de tweet non médicale

Pour voir la topique de ce tweet, on clique sur le bouton « topics » (Voir figure 4.18) :

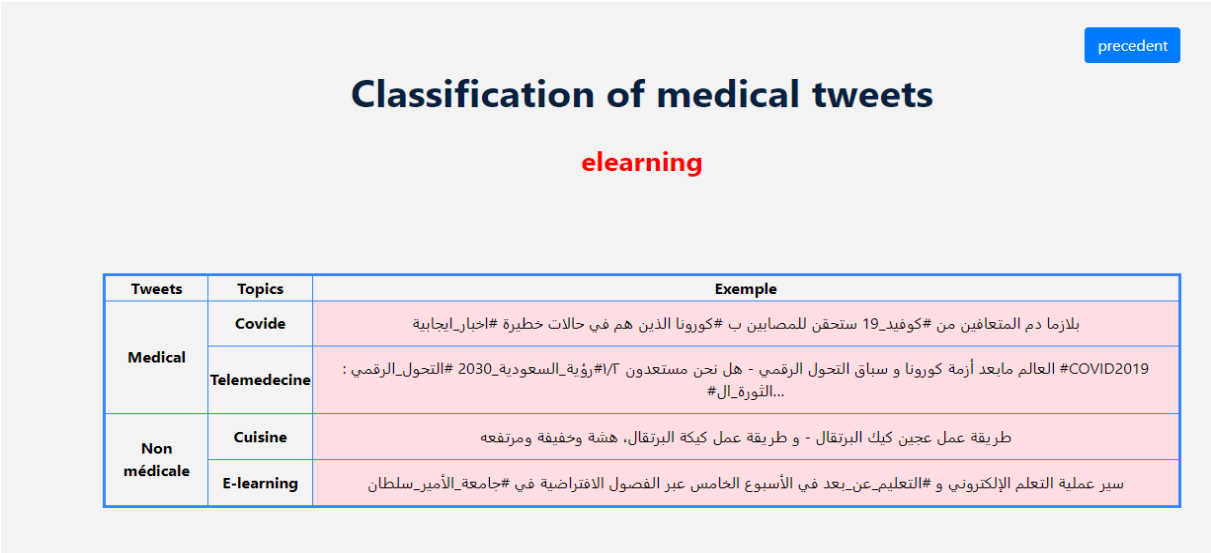


Figure 4.18: Identification de topique de tweet non médicale

Et voici la page de « about us » (Voir figure 4.19) :

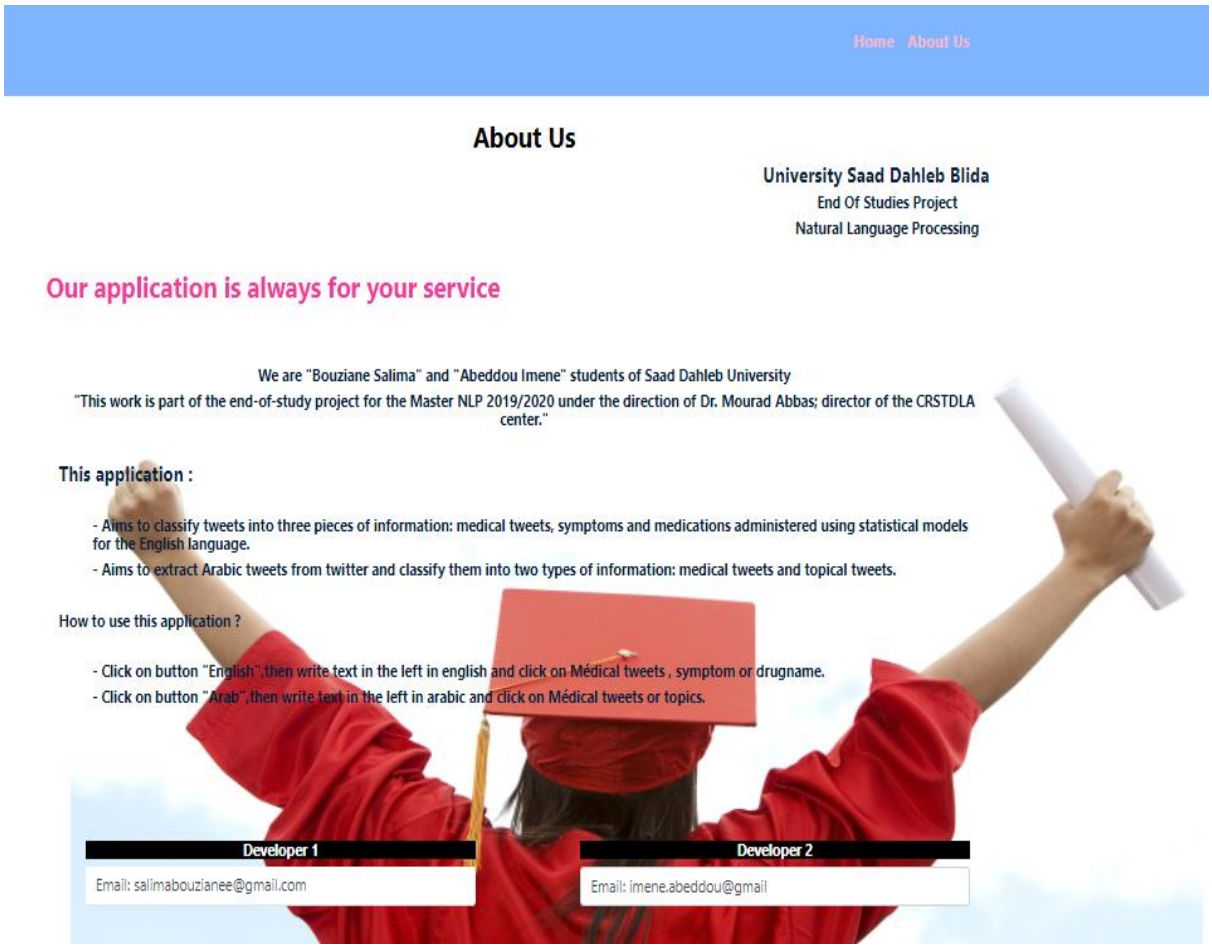


Figure 4.19: Page : « About Us »

Conclusion

Au cours de ce chapitre, nous avons abordé l'implémentation de notre système et les outils utilisés pour le développement de celui-ci, ensuite nous avons entamé la présentation des différentes interfaces et fonctionnalités de notre application à travers les différentes captures.

Conclusion générale

Dans ce mémoire, nous nous sommes intéressés à l'extraction et la classification des textes médicales sur twitter en utilisant l'apprentissage supervisé. Nous sommes particulièrement concentrés sur la phase de prétraitement des tweets avant d'utiliser un algorithme de classification et ceci en se basant sur les méthodes de classification « 'KNN' KNeighborsClassifier, 'LSVM' LinearSVC, 'LR' LogisticRegression, 'DT' DecisionTreeClassifier, 'MNB' MultinomialNB, 'BNB' BernoulliNB, 'RF' RandomForestClassifier, 'GB' GradientBoostingClassifier». Dans ce cas de travail, nous utilisons deux corpus de tweets de langue anglais et arabe. Nous avons réalisé une application web dans le cas du corpus anglais permet d'identifier les tweets anglais qui contiennent des informations médicales en considérant deux entités : symptômes et médicaments. Dans le cas du corpus arabe, elle vise à identifier les tweets avec ses topiques. Nous avons élaboré une étude comparative entre différents algorithmes de classification. On a pu constater que les résultats obtenus avec LSVM sont très satisfaisants par rapport les autres algorithmes. LSVM est une méthode de classification qui montre de bonnes performances dans la résolution de problèmes variés [36]. Cette méthode a montré son efficacité dans de nombreux domaines d'applications tels que le traitement d'image, la catégorisation de textes ou le diagnostiques médicales et ce même sur des ensembles de données de très grandes dimensions.

Nous avons rencontré quelques difficultés dans notre projet :

- Manque du corpus dédié à la langue arabe.

- Manque d'un dictionnaire médical qui aide à la détection des tweets médicales.

- Rapprochement des symptômes où on peut trouver deux symptômes qui provoque presque le même effet au malade…etc.

On a rencontré aussi un autre problème lié aux limites de la technologie, telles que l'accessibilité aux logiciels existants et au matériel de hautes performances, qui sont nécessaire pour traiter de gros volumes de données et atténuer le problème de temps de traitement.

Quoi que nous ayons réalisé le principal objectif de notre travail. Cependant nous envisagerons quelques perspectives qui permettent l'amélioration et la conformité de notre travail notamment :

-Améliorer le système avec corpus plus large et plus riche et essayer de surmonter la complexité de la langue arabe

- ❖ Collecte de données de haute qualité.
- ❖ Utilisation d'autres techniques de prétraitement.
- ❖ Implémenter d'autres classifieurs pour avoir l'occasion de les comparer avec notre classifieur LSVM pour améliorer les résultats.

-Utiliser les ontologies ou les dictionnaires pour enrichir la représentation des textes.

Bibliographie

[1] François Yvon. Une petite introduction au Traitement Automatique des Langues Naturelles. Article.26/04/2007.

[2] Yash Pershad, Patrick T. Hangge, Hassan Albadawi ,Rahmi Oklu.Social Medicine: Twitter in Healthcare. Journal of clinical medecine. 28 May 2018.

[3] S. Doan, E. W. Yang, S. Tilak and M. Torii, «Using Natural Language Processing to Extract Health-Related Causality from Twitter Messages » . IEEE International Conference on Healthcare Informatics Workshop (ICHI-W), New York, NY, 2018, pp. 84-85, doi: 10.1109/ICHI-W.2018.00031.

[4] Brigitte Bigi. TALN Informatique.document. CLIPS - Equipe GEOD, ITC .Avril 2006.

[5] MouradLoukam. [En ligne].Available : http://www.loukam.net/TALN_Chap1.pdf.(Consulté le 04/04/2020).

[6] Nouadri Zeyneb, « proposition d'un outil d'évaluation automatique par textes libres dans une situation d'apprentissage ». Mémoire Master Université Oum El Bouaghi .13/06/2018.

[7] OUALI Choayb, « Classification automatique de textes ». Mémoire Master Université de Msila. 2014.

[8] GAGAOUA Meriem, « Apprentissage et fouille de données par les algorithmes bio-inspirés : Application à la reconnaissance de caractères arabes manuscrits ». Mémoire Magister. 21/01/2012.

[9] Asma Ben Abacha, Pierre Zweigenbaum, Aurélien Ma, « Extraction d'information automatique en domaine médical par projection inter-langue : vers un passage à l'échelle ». Conference: TALN (Traitement automatique des langues naturelles). At: Grenoble, France.Juin 2012.

[10] Jean-Baptiste Fantun. Les réseaux sociaux et la santé ; un enjeu pour le suivi des patients et la recherche scientifique. International think tank dedicated to big data in healthcare. Article, Septembre 2018.

[11] Elise Bigeard, Natalia Grabar, Frantz Thiessard. Détection de mésusages de médicaments dans les réseaux sociaux. TALN. Communication dans un congrès. Rennes, France. halshs-01968335. May 2018.

[12] Yvon, François. « Des apprentis pour le traitement automatique des langues ». Mémoire d'habilitation à diriger des recherches, Université Pierre et Marie Curie, Paris (2006).

[13] Belainine, Billal. « Classification supervisée de textes courts et bruités: application au domaine des médias sociaux ». Université Québec.Montréal. (2017).

[14] MATALLAH, Hocine, « Classification Automatique de Textes Approche Orientée Agent ». Mémoire Magister. Université Aboubekr Belkaid.Tlemcen.2011.

[15] TERKIA DERDRA Amel, Melle. BENSFIA Fatima Zahra , « La Représentation Conceptuelle pour laCatégorisation des Textes Multilingue ». Mémoire. Université Aboubekr Belkaid.Tlemcen .25 Septembre 2012.

[16]ANTHONY SISTILLI.Twitter Data Mining: A Guide to Big Data Analytics Using Python.[En ligne]Available:https://www.toptal.com/python/twitter-data-mining-using-python.[Accés le 05/04/2020].

[17] Developer. [En ligne]Available:https://apps.twitter.com/.[Accés le 04/02/2020].

[18] ABDELOUAHAB Soumia, « Processus de classification supervisée de textes arabes par la méthode K PPV Application aux articles de presse ». Mémoire Master. Université Aboubekr Belkaid.Tlemcen .2012.

[19] BouhalassaFaiza, « Un outil d'extraction automatique de concepts à partir de données textuelles : Application à l'enrichissement des ontologies ». Mémoire Master. Université Oum El Bouaghi. Juin 2018.

[20] AnsariRazali, SalwaniMohdDaud, Nor Azan Mat Zin ,FaezehsadatShahidi, « Stemming Text based Web Page Classification using Machine Learning Algorithms: A Comparison ». (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 1, 2020.Malaysia.

[21] RadwanJALAM, « Apprentissage automatique et catégorisation de textes multilingues ».Thèse Doctortat. Université Lumière Lyon2. 4 juin 2003.

[22] Bruno Pouliquen., « Indexation de textes médicaux par extraction de concepts, et ses utilisations.Informatique et langage [cs.CL] ». Université Rennes 1.2002. Français.

[23] Hausmane Issarane. « 5 Algorithmes D'Apprentissage Supervisé ».article. 6 février 2019.[En ligne].Available : https://le-datascientist.fr/5-apprentissage-supervise.(Consulté le 21/06/2020).

[24] Nouas Sihem. « Analyse sémantique des publications dans les réseaux sociaux par apprentissage profond ». Mémoire Master.Université Saad Dahleb.Blida. 2019.

[25] Opengenus.Bernoulli Naive Bayes. [En ligne].Available : https://iq.opengenus.org/bernoulli-naive-bayes/.(Consulté le 29/05/2020).

[26] Sébastien Rigaux. « Multi-Label Text Classification of Medical Abstracts ». Mémoire Master. Université de Liège. 2015.

[27] techvidvan. Supervised Learning Algorithm in Machine Learning.[En ligne].Available :https://techvidvan.com/tutorials/supervised-learning/.(Consulté le 25/06/2020).

[28] scikitlearn.sklearn.ensemble.GradientBoostingClassifier.[En ligne].Available :

http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html.(Consulté le 12/06/2020).

[29] Felix Gräßer, Surya Kallumadi, Hagen Malberget Sebastian Zaunseder. «Aspect-Based Sentiment Analysis of Drug Reviews Application Cross-Domain and Cross-Data Learning». Conférence internationale.2018.New York, NY, USA. .[En ligne].Available : https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29.(Accées le 10/12/2019).

[30] Matthias Braunhofer, Francesco Ricci. TripAdvisor Dataset.October 2016. [En ligne]Available:https://www.researchgate.net/publication/308968574_TripAdvisor_Dataset.[Accés le 10/01/2020].

[31] Colas, F., & Brazdil, P. (2006). «Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks». IFIP International Federation for Information Processing, 217, 169-178.

[32] ADDA Ahmed,DJEBABRA , Mohamed Nabil.,« Développement d'une application de télé-opération d'un système chauffage commande par internet ».Mémoire Master . Université Dr.Tahar Moulay.Saïda.2 7/06/2018.

[33] BELMEHDI Ouacila, « Conception et Réalisation d'un site WebTV pour la Fédération des Associations des parents d'Elèves de la wilaya de Bejaia ». Mémoire Master. Université Abderrahmane Mira.Bejaia. 2016/2017.

[34]J. Wolfe, «A Brief History of Python.»05 Mars 2018.[En ligne].Available : https://medium.com/@johnwolfe820/a-brief-history-of-python-ca2fa1f2e99e.[Accès le 24 07 2020].

[35] GUENNINECHE Amel, « Prédiction des propriétés des matériaux par apprentissage automatique ». Mémoire Master. Université Aboubekr Belkaid.Tlemcen. 29/06/2019.

[36] Ardjani Fatima, « Optimisation des SVM multiclasses par des méthodes évolutionnaires (PSO-SVM) ». Mémoire Magister. Université des Sciences et de la Technologie d'Oran - Mohamed Boudiaf.