جامعة مولاي إسماعيل
UNIVERSITÉ MOULAY ISMAÏL

كلية العلوم والتقنيات
FACULTÉ DES SCIENCES ET TECHNIQUES

Département d'Informatique
Licence Sciences et Techniques en Informatique
Option : Génie Logiciel

MÉMOIRE DE PROJET DE FIN D'ETUDES

Intitulé :

UNE ÉTUDE DE RECHERCHE EMPIRIQUE SUR L'EFFICACITÉ DES ALGORITHMES DE DÉTECTION DE PLAGIAT

Préparé par :

AFZID Omayma

ELAZZAOUI Mohamed

EL BAHMADI Abdelkabir

Soutenu le 01 juillet 2022 devant le Jury :

Pr. FARHAOUI Yousef	Faculté des Sciences et Techniques Errachidia	Président
Pr. EL ALLAOUI Ahmad	Faculté des Sciences et Techniques Errachidia	Examinateur
Pr. Imad Zeroual	Faculté des Sciences et Techniques Errachidia	Encadrant

Année Universitaire : 2021-2022

Remerciements

Louange à Allah, le tout-puissant de nos avoir donné le courage, la volonté et la patience de mener à terme ce modeste travail.

Tout d'abord, nous tenons à exprimer notre gratitude à notre encadrant Monsieur Imad ZEROUAL, qui a dirigé nos travaux de recherches avec beaucoup de patience et de gentillesse.

Nous remercions également tous les membres de jury pour l'honneur qu'ils expriment en acceptant de présider et examiner ce travail.

Nous devons inévitablement exprimer notre gratitude à nos collègues, nos familles et tous ceux qui nous ont apporté leur contribution.

Merci à tous les enseignants de notre Faculté des Sciences et Techniques Errachidia, pour leurs efforts pour fournir une formation adéquate et améliorer nos connaissances.

Enfin, nous voudrions exprimer nos reconnaissances à tous ceux qui nous ont aidés à mener à bien ce projet.

Dédicace

Nous dédions ce modeste travail :

A nos chers parents pour tous leurs sacrifices, leurs amours, leurs tendresses, leurs soutiens et leurs prières tout au long de nos études.

A toute nos familles pour leurs soutiens tout au long de nos parcours universitaires.

A tous les cousins, les voisins et les amis que j'ai connu jusqu'à maintenant. Merci pour leurs amours et leurs encouragements.

Résumé

Le plagiat peut être un grand problème pour les entreprises, les écoles, les collèges et ceux qui distribuent leurs archives en ligne. Les étudiants des écoles et des collèges compilent leurs devoirs et leurs expériences avec des copies d'autres archives.

À l'aide de cette structure, les enseignants et les analystes peuvent identifier les documents et les articles créés par un étudiant en particulier ou copiés de quelqu'un d'autre. Pour vérifier le plagiat, la plate-forme prend deux ou plusieurs rapports en entrée et après avoir utilisé des algorithmes de traitement de chaînes et les technologies de NLP, elle produit un résultat. Dans ce résultat, le système renvoie plusieurs valeurs entre 0 et 1. Où 1 indique qu'il est strictement plagié. Si 0, ça signifie que le rapport est libre de plagiat. Au cas où il y aurait un score entre 0 et 1 à ce moment-là, il apparaît que seules quelques parties du document qui sont similaires. Le but principal de cette étude est de trouver une méthode de détection de plagiat plus précis.

Table des matières

Liste des figures

Liste des tableaux

Liste des formulaires

Liste des accro-names

- IA : Intelligence artificielle.
- NLP : Natural Language Processing.
- NLG : Natural Language Generation.
- TF-IDF : Term Frequency-Inverse Document Frequency.
- TCV : Token Count Vectoriser.
- Lev : Levenshtein.
- J : Jacard.
- LCS : Longest Common Subsequence.
- DSC : Dice Similarity Coefficient.
- HTML : HyperText Markup Language.
- CSS : Cascade Style Sheet.
- JS : JavaScript.
- PHP : Personal Home Page.
- MYSQL : My Structured Query Language.

Chapitre 1 :
Introduction et contexte générale

I.Introduction :

La facilité de partager du matériel en ligne a encouragé la recherche de littérature sur Internet depuis que nous sommes entrés dans l'ère des communications numériques. Cette évolution augmente le potentiel de nouvelles inconduites académiques et de vols de propriété intellectuelle.

La détection automatique du plagiat attire de plus en plus l'attention en raison de préoccupations croissantes par le plagiat. Il s'agit d'une méthode informatique permettant de déterminer si un texte est volé ou non. Cependant, la plupart des méthodes de détection de plagiat actuelles reposent sur des méthodes de correspondance de chaînes insensées et de force brute.

Il est proposé de détecter le plagiat. Au lieu de s'appuyer entièrement sur les procédures traditionnelles de mise en correspondance des chaînes, il utilise des techniques de traitement du langage naturel. À l'aide d'une approche de groupe, l'objectif est d'explorer et d'évaluer l'impact du prétraitement de texte, des méthodes linguistiques statistiques, superficielles et profondes.

De cette étude donnent des idées pour de nouvelles directions de recherche et des applications potentielles pour relever les défis qui se présentent dans la détection de la réutilisation des textes.

II.Le Plagiat :

1. Définition du plagiat :

Le plagiat, c'est quand quelqu'un vole l'idée ou le travail de quelqu'un d'autre et le transmet comme le sien. Le plagiat est classé comme une atteinte au droit moral dans un certain nombre de pays. Le plagiat devient de plus en plus courant dans l'évolution technologique d'aujourd'hui et l'utilisation croissante d'Internet. Il peut être trouvé dans divers établissements d'enseignement, y compris des documents de recherche, des blogs, des articles et des devoirs.

2. Types de plagiat :

Le plagiat, ce n’est pas seulement « copier-coller » le travail de quelqu’un d’autre. Le plagiat peut prendre de nombreuses formes différentes. Cela peut aller de la réutilisation d'un document entier à la modification d'un paragraphe entier. Le plagiat est défini comme l'acte de transmettre les idées ou les mots de quelqu'un d'autre en tant qu'idées ou mots individuels.

On distingue donc les différents cas suivants :

a. Copier-Coller :

Le plagiat par copier-coller, souvent appelé plagiat pur et simple, se produit lorsqu'une personne copie et colle du texte d'une autre source sans s'y référer. Vous devez apprendre à citer un texte d'une autre source si vous voulez vraiment l'inclure mot pour mot.

b. Mosaïque :

Copier et coller ensemble différents morceaux de texte pour créer une sorte de « mosaïque » ou de « patchwork » des idées d'autres chercheurs est un plagiat.
Bien que le résultat soit un morceau de texte complètement nouveau, les mots et les idées ne sont pas nouveaux.

c. Auto-Plagiat :

Lorsque vous utilisez des parties d'un travail antérieur (comme un article, une analyse documentaire ou un ensemble de données) sans une citation appropriée, vous vous engagez dans ce que l'on appelle l'autosuffisance.
Bien que la pénalité pour avoir plagié votre propre travail puisse sembler étrange, vous devez comprendre que c'est parce qu'elle va à l'encontre des attentes des lecteurs de votre article. Ils s'attendent à ce que l'œuvre soit unique.

d. Traduction :

Lorsque vous utilisez un outil de traduction de texte que vous copiez et collez simplement, cela ne signifie pas que vous l'avez écrit.

Le plagiat se produit lorsqu'une personne copie et colle le travail d'une autre personne dans une langue étrangère, puis le traduit sans mentionner la source.

e. Paraphrase :

La paraphrase, aussi appelée reformulation paraphrastique, est le processus d'extraction d'une phrase d'un texte et de la modifier pour la rendre plus précise ou explicite. C'est un processus qui préserve le sens et, dans la plupart des cas, l'ordre des concepts proposés dans le texte, mais permet de modifier le vocabulaire, ainsi que l'ajout, la suppression ou le remplacement de certains mots pour le compléter ou l'ajouter Information.

III. L'Intelligence artificielle et le traitement automatique du langage naturel :

1. L'intelligence artificielle :

L'intelligence artificielle (IA) est une méthode d'imitation de l'intelligence humaine en concevant et en mettant en œuvre des algorithmes dans un environnement informatique dynamique. Son objectif est de permettre aux ordinateurs de penser et d'agir de la même manière que les humains.

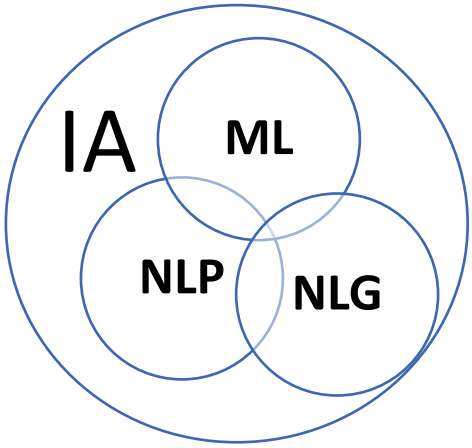


Figure 1 : Panorama de quelques domaines de l'IA

Notre objectif est d'étudier les moyens d'améliorer l'exactitude des approches de détection existantes en utilisant les technologies de traitement du langage naturel (NLP). De meilleures approches de détection pourraient également être utilisées pour établir une compréhension plus large de l'importance d'un référencement correct et encourager le déploiement d'approches de prévention du plagiat.

2. Traitement du Langage Naturel :

Le Langage Naturel est le moyen de communication employés par les êtres humains de façon inné dans la vie quotidienne. L'anglais, l'arabe et le français sont des exemples de langages naturels. Ils sont construits selon une syntaxe, une grammaire et peuvent contenir beaucoup d'ambiguïtés.

Le traitement du langage naturel (NLP) est un ensemble d'approches théoriquement informatisées pour évaluer et modéliser des textes naturels à un ou plusieurs niveaux d'analyse linguistique afin d'obtenir un traitement du langage de type humain pour une variété d'activités et d'applications.

La capacité de la technologie à reconnaître et à interpréter d'énormes quantités de données textuelles dans le monde numérique, y compris les plateformes de médias sociaux, les critiques en ligne, les bulletins d'information, etc. C'est l'un des avantages les plus importants de la NLP pour les organisations.

Tout cela est possible car Google a appris aux machines à comprendre le langage humain plus rapidement, plus précisément et de manière cohérente que les humains. La technologie peut suivre et traiter les données de manière cohérente

3. L'intelligence artificielle et la détection du plagiat :

La détection manuelle et la détection du plagiat ne sont plus possibles. Comme il y a beaucoup de contenu sur le Web, la détection du plagiat prend beaucoup de temps et n'est pas aussi précise que les solutions basées sur l'IA. De plus, les centres de données qui vérifient automatiquement les informations pour le plagiat sont la nouvelle norme à l'ère des systèmes d'exploitation.

L'application a été créée spécifiquement pour détecter le plagiat dans n'importe quel travail, d'une thèse de maîtrise à un e-book commercial. Malgré cela, en raison de la facilité d'accès au matériel sur Internet, les cas de plagiat ont considérablement augmenté à l'ère d'Internet.

De plus, pour éviter la détection de plagiat, certaines personnes utilisent des éditeurs de texte avancés. Par conséquent, le moyen le plus précis de détecter le plagiat est d'utiliser l'intelligence artificielle. Le plagiat est devenu un problème sérieux et le plagiat n'a jamais été aussi simple. Pour réduire la propagation du plagiat, des solutions logicielles hautement intelligentes sont crées pour supprimer le contenu volé des archives Web et fournir un environnement sans plagiat.

IV.Conclusion :

Nous avons examiné le plagiat et les principaux types de plagiat ainsi que le traitement du langage naturel, l'intelligence artificielle et le rôle de l'intelligence artificielle dans la détection du plagiat.

Chapitre 2 :
Prétraitement des données textuelles et extraction des caractéristiques

I.Introduction :

Le prétraitement est une tâche critique et une étape essentielle dans le traitement du langage naturel, l'extraction de texte et la recherche d'informations.

En fait, les données brutes sont souvent insuffisantes, confuses et difficiles à interpréter pour déterminer les tendances d'utilisation. Il existe également de nombreuses erreurs telles que d'autres langues dans le texte et des alphabets incorrects. La préparation des données s'est avérée être un moyen efficace de résoudre ces problèmes.

Le but du prétraitement de texte est de changer le texte dans un format facile à lire pour les machines. Le traitement de texte est très important car il nous aide à mieux comprendre nos données et à en tirer des enseignements.

Le prétraitement de texte dans notre recherche comprend les étapes suivantes :

Figure 2 : Les étapes du prétraitement

II.Prétraitement :

1. Normalisation :

Avant de passer à toute étape du traitement initial des complexes de texte, nous commençons toujours par l'étape de normalisation.

La normalisation est une méthode de prétraitement des données qui permet de réduire la complexité des modèles. C'est également un préalable à l'application de certains algorithmes.

Par exemple Lettres majuscules à minuscules Éliminez les symboles, les chiffres et les ponctuations etc.

- Exemple :

Tableau 1 : Exemple normalisation

Avant	Après
Hello, I'm 21 years old. what about you?	hello I m years old what about you
Le sorcier habite dans un château 21.	Le sorcier habite dans un château
بحث الانسان على مر التاريخ، على اختراع يمكنه أن يحاكي العقل البشري في نمط تفكيره أبريل أربعة33.	بحث الانسان على مر التاريخ على اختراع يمكنه أن يحاكي العقل البشري في نمط تفكيره أبريل أربعة

2. Suppression du Tashkil de la langue arabe :

Le sens littéral de tashkīl est « former ». Comme le texte arabe normal ne fournit pas suffisamment d'informations sur la prononciation correcte, le but principal du tashkīl est de fournir un guide phonétique ou une aide phonétique ; c'est-à-dire montrer la prononciation correcte.

La suppression des tashkil peut être considérée comme une contribution au nettoyage des textes arabes avant de les traiter.

- Exemple :

Avant : "بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ محمد"
Après : "بسم الله الرحمـــن الرحيم محمد"

3. Suppression des mots vides :

L'une des principales formes de prétraitement consiste à filtrer les données inutiles. Dans le traitement du langage naturel, les mots inutiles sont appelés mots vides.

Un mot vide est un mot couramment utilisé (comme « le », « a », « an », « dans » …).

- Exemple :

Tableau 2 : Exemple suppression des mots vides

Avant	Après
Research shows that sustainable loss of weight can only be achieved on a diet	sustainable loss weight achieved diet
Le sorcier habite dans un château	Sorcier habite château
بحث الانسان على مر التاريخ على اختراع يمكنه أن يحاكي العقل البشري في نمط تفكيره أبريل أربعة	بحث الانسان مر التاريخ اختراع يمكنه يحاكي العقل البشري نمط تفكيره

4. Lemmatisation :

Le processus de la lemmatisation consiste à représenter les mots sous leur forme canonique. Par exemple pour un verbe, ce sera son infinitif. Pour un nom, son masculin singulier. L'idée étant encore une fois de ne conserver que le sens des mots utilisés dans le texte.

Le but de l'utilisation de Lemmatisation dans la détection de plagiat est de détecter la paraphrase.

- Exemple :

Tableau 3 : Exemple Lemmatisation

Avant	Après
plays, playing, played, play	play, play, play, play
Voudrais, non, animaux, yeux, dors, couvre	Vouloir, non, animal, œil, dor, couvrir
سيعرفونها خلال العمل بالحاسوبين المستعملين	عرف خلال عمل حاسوب مستعمل

5. Stemming :

Le stemming consiste à réduire un mot dans sa forme « racine » en supprimant les suffixes et les préfixes. Le but du stemming est de regrouper de nombreuses variantes d'un mot comme un seul et même mot.

- Exemple :

Tableau 4 : Exemple Stemming

Avant	Après
plays, playing, played, play	plai plai plai plai
sorciere habiter châteaux	sorci habit château
سيعرفونها خلال العمل بالحاسوبين المستعملين	يعرفون خلال عمل الحاسوبين مستعملين

6. Tokenisation en N-grammes :

Les N-grammes de textes sont largement utilisés dans les tâches d'exploration de texte et de traitement du langage naturel. Un N-gramme signifie une suite de N mots. Les N-grammes sont essentiellement un ensemble de mots simultanés dans une fenêtre donnée et lors du calcul des n-grammes, vous avancez généralement d'un mot (bien que vous puissiez avancer X mots dans des scénarios plus avancés).

a. Exemple :

Tableau 5 : Exemple tokenisation en n-grammes

Phrase	Uni-grams	Bi-grams	Tri-grams
My father is Ahmed	▪ My ▪ father ▪ is ▪ ahmed	▪ My Father ▪ father is ▪ is Ahmed	▪ My Father is ▪ father is Ahmed
Mon pére est Ahmed	▪ Mon ▪ pére ▪ est ▪ Ahmed	▪ Mon pére ▪ pére est ▪ est Ahmed	▪ Mon pére est ▪ pére est Ahmed

Phrase	Uni-grams	Bi-grams	Tri-grams
بحث الانسان عن اختراع	▪ بحث ▪ الانسان ▪ عن ▪ اختراع	▪ بحث الانسان ▪ الانسان عن ▪ عن اختراع	▪ بحث الانسان عن ▪ الانسان عن اختراع

III. Extraction des caractéristiques :

La représentation vectoriel ou l'extraction des caractéristiques est une méthode qui consiste à rendre les données exploitables pour quelques algorithmes mathématiques.

Dans notre étude, nous allons tester les deux méthodes «Token Count Vectorizer » et « TF-IDF ».

1. Token Count Vectoriser :

TCV est utilisé pour transformer un corpus de texte en un vecteur de nombres de termes / jetons.

Tout d'abord, nous définissons un dictionnaire sur la base duquel les dimensions du vecteur qui contiendra les nombres sont déterminées.
Ensuite, nous calculons le nombre d'occurrence de chaque mot du dictionnaire dans le texte.

- Exemple :

On a deux phrases : « Je m'appelle Mohamed » et « Je m'appelle Abdelkbir » On extraire le dictionnaire du premier phrase :
Dictionnaire = {Je,m'appelle,Mohamed}
Maintenant on compte l'occurrence de chaque term dans les deux phrase, le tableau suivant montre le processus :

Tableau 6 : Exemple token count voctoriser

	Je	M'appelle	Mohamed
Je m'appelle Mohamed	1	1	1
Je m'appelle Abdelkbir	1	1	0

2. Term Frequency-Inverse Document Frequency:

TF-IDF signifie Term Frequency-Inverse Document Frequency. Et c'est une mesure qui quantifie l'importance ou la pertinence des représentations de chaînes (mots, phrases, lemmes, etc.) dans un document parmi une collection de documents.

Les étapes de TF-IDF sont comme suite :

1. Prétraitement, tokenisation et trouver le nombre d'occurrence pour chaque mot

2. Trouver TF pour les mots
   TF = (Nombre de répétitions de mot dans le document) / (nombre de mots dans le document)
3. Trouver IDF pour les mots
   IDF = Log [(Nombre de documents) / (Nombre de documents contenant le mot)]
4. Vectoriser le vocabulaire
   Calculer pour chaque mot la valeur TF * IDF

- Exemple :

On prend par exemple les trois documents suivants :
Document 1 : Je suis Mohamed.
Document 2 : Je suis un étudiant.
Document 3 : Mon ami est un étudiant.

1. Étape1 :

Tableau 7 : Vocabulaire utilisé

Mot	Nb d'occurence
Je	2
Suis	2
Mohamed	1
Un	2
Etudiant	2
Mon	1
Ami	1
Est	1

2. Etape 2 :

Tableau 8 : Calcule de TF pour les differents documents

Mot	TF dans Doc 1	TF dans Doc 2	TF dans Doc 3
Je	0.33	0.25	0
Suis	0.33	0.25	0
Mohamed	0.33	0	0
Un	0	0.25	0.20
Etudiant	0	0.25	0.20
Mon	0	0	0.20
Ami	0	0	0.20
Est	0	0	0.20

3. Etape 3 :

Tableau 9 : Calcule de IDF pour chaque mot

Mot	IDF
Je	Log(3/2)=0.18
Suis	0.18
Mohamed	0.48
Un	0.18
Etudiant	0.48
Mon	0.48
Ami	0.48
Est	0.48

4. Etape 4 :

*Tableau 10 : Calcule de TF*IDF pour chaque mot dans chaque document*

Mots/Documents	Je	Suis	Mohamed	Un	Etudiant	Mon	Ami	Est
Document 1	0.06	0.06	0.16	0	0	0	0	0
Document 2	0.05	0.05	0	0.05	0.12	0	0	0
Document 3	0	0	0	0.04	0.01	0.01	0.01	0.01

IV.Les bibliothèques que nous avons utilisées pour le prétraitement :

- NLP Tools

NlpTools est un ensemble de classes php 5.3+ pour les travaux de traitement du langage naturel débutants à semi-avancés.

- PHP AI

Bibliothèque d'apprentissage automatique en PHP pleine d'algorithmes, de validation croisée, de réseau de neurones, de prétraitement, d'extraction de fonctionnalités et bien plus encore.

- PHP Lemmatizer

Une bibliothèque PHP pour obtenir un lemme à partir d'un mot donné et obtenir une liste de mots correspondant à un lemme.

- LCS DEVELOP

Implémentation PHP d'un algorithme pour résoudre le problème de "la plus longue commune sous-séquence".

- PHP Stemmer

Implémentation native PHP de Snowball stemmer (Snowball est un langage de traitement de petites chaînes permettant de créer des algorithmes de radicalisation à utiliser dans la recherche d'informations).

V.Conclusion :

Nous avons examiné les principales étapes du prétraitement à effectuer lorsque vous travaillez avec des données textuelles. En suivant ces procédures de traitement de base, des modèles tels que Bag Of Word et Word2Vec peuvent être utilisés pour traiter davantage les données.

Chapitre 3 :
Les algorithmes de calcule de similarité

I.Introduction :

Dans le traitement du langage naturel, différents types d'algorithmes ont été développés pour détecter la similitude entre les textes, dans le but d'atteindre une précision et une efficacité maximales, chacun avec ses propres avantages et inconvénients.

Dans ce chapitre, nous nous intéresserons à définir ces algorithmes, à en donner un aperçu, à étudier le fonctionnement de chaque algorithme et comment il analyse les données et enfin quelques exemples.

Les algorithmes que nous avons ciblés dans notre étude sont :

- Les algorithmes traditionnels comme STRCMP
- Algorithme de Similar Text
- Distance de Levenshtein
- Coefficient de Jaccard
- Similarité en Cosinus
- La plus longue sous-séquence commune
- Coefficient de Dés

II.Les algorithmes de calcule de similarité :

1. Comparaison traditionnelle :

• Aperçu :

Les langages de programmation nous fournissent un ensemble de fonctions déjà défini, elles nous permettent de comparer des chaînes de caractères tels que les fonctions« strncmp »et« strcmp ».
La fonction « strcmp » est basée sur la fonction « strncmp », qui ne compare que deux caractères de la chaîne.
« strcmp » compare lexicalement deux chaînes caractère par caractère et renvoie une valeur positive si la première chaîne inférieure à la deuxième chaîne, négative si le contraire, et 0 s'ils sont égaux.

• Exemple :

Entrée : « hello » et « hello ». ***Sortie :*** 0.

Entrée : « hello » et « Hello ». ***Sortie :*** 1. Attention, Strcmp est sensitive à la casse.

Entrée : « hello » et « hell ». ***Sortie :*** 1.

Entrée : « hello » et « hello students ». ***Sortie :*** -9.

2. Algorithme de Similar_Text :

- Aperçu :

Similar_text est une fonction intégrée à PHP. Cette fonction calcule la similarité de deux chaînes et renvoie le nombre de caractères identiques dans les deux chaînes. La fonction fonctionne en trouvant la première sous-chaîne commune la plus longue et en la répétant pour les préfixes et les suffixes, de manière récursive. La somme des longueurs de toutes les sous-chaînes communes est la valeur renvoyée par la fonction.

- Exemple :

Entrée : « Salut » et « Salut ». ***Sortie :*** 5 (100.00%).

Entrée : « Salut » et « salut ». ***Sortie :*** 4 (80%) (Similar_Text est sensitive à la casse).

Entrée : « bafoobar » et « barfoo ». ***Sortie :*** 5 (71.42%).

Entrée : « barfoo » et « bafoobar ». ***Sortie :*** 3 (42.85%).

3. Algorithme distance de Levenshtein (Minimum Edit Distance) :

- Aperçu :

La distance de Levenshtein (également appelée distance d'édition) est une mesure de la similarité entre deux chaînes. Supposons-nous que vous avez deux chaînes : la chaîne source et la chaîne cible. La distance Levenshtein est le nombre de suppressions, d'insertions ou de substitutions nécessaires pour transformer la chaîne source en chaîne cible.

- Formule mathématique :

Formellement, on définit cette distance avec deux chaînes a et b, où $|a|$ le cardinal de a (ou son nombre de lettres), et $a - 1$ la chaîne a tronquée de sa 1re lettre $a[0]$:

$$lev(a,b) = \begin{cases} \max(|a|,|b|), & si\ \min(|a|,|b|) = 0 \\ lev(a-1,b-1), & si\ a[0] = b[0] \\ 1 + \min \begin{cases} lev(a-1,b) \\ lev(a,b-1) \\ lev(a-1,b-1) \end{cases}, & sinon \end{cases}$$

Formulaire 1: Distance de Levenshtein

- Exemple :

Soit la chaîne source **« panorama »** et la chaine cible **« paronomase »**.

Pour être transformée de la chaîne source vers la chaîne cible, on doit passer par cinq opérations :

i. → parorama (substitution de "n" en "r")
ii. → paronama (substitution de "r" en "n")
iii. → paronoma (substitution de "a" en "o")
iv. → paronomas (ajout d'un "s")
v. → paronomase (ajout d'un "e")

Alors la distance de Levenshtein est le nombre d'opérations effectuées. Dans notre cas la distance égale à cinq.

4. Algorithme coefficient de Jaccard (Jaccard Index):

- Aperçu :

L'algorithme de similarité Jaccard (également appelé indice Jaccard, coefficient Jaccard, dissemblance Jaccard et distance Jaccard) est une mesure de proximité utilisée pour déterminer la similarité de deux objets.
L'algorithme de similarité de Jaccard peut être utilisé pour trouver la similarité entre deux vecteurs binaires asymétriques ou deux ensembles.

- Formule mathématique :

Soit J la fonction qui implémente l'algorithme de coefficient Jaccard. On définit J avec deux ensembles A et B.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Formulaire 2: Coefficient de Jacard

Nous devons d'abord compter le nombre de membres qui sont partagés entre les deux ensembles A et B. Ensuite, comptons le nombre total de membres dans les deux ensembles (partagés et non partagés).
Enfin nous divisons le nombre de membres partagés par le nombre total de membres.

- Exemple :

Soit $A = \{\text{Salut}, Je, suis, Ahmed\}$ et $B = \{\text{Bonsoir}, Je, suis, Mohamed\}$.

On calcule $A \cap B$ et $A \cup B$:

$A \cap B = \{Je, suis\} \Rightarrow |A \cap B| = 2$

$A \cup B = \{Salut, Je, suis, Ahmed, Bonsoir, Mohamed\} \Rightarrow |A \cup B| = 6$

Alors, le taux de similarité est $J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{2}{6} \approx 0.34 = 34\%$

5. Algorithme de similarité en cosinus (Cosine Similarity) :

- Aperçu :

La similarité en cosinus est une métrique qui mesure la similarité de deux vecteurs ou plus. Le cosinus de l'angle entre les vecteurs est le cosinus de similarité. Les vecteurs sont généralement non nuls et appartiennent à un espace produit interne.

- Formule mathématique :

La similarité cosinus est décrite mathématiquement comme la division entre le produit scalaire des vecteurs et le produit des normes euclidiennes ou de l'amplitude de chaque vecteur.

Soit A et B deux vecteurs non nuls, et θ l'angle entre eux :

$$Similarité = \cos(\theta) = \frac{A.B}{\|A\|.\|B\|}$$

Formulaire 3: Cosine Similarity

Plus la valeur de $\cos(\theta)$ est proche de 1, plus la similarité entre A et B est grande

- Exemple :

Soit A, B deux vecteurs de mêmes dimensions :

$A = \begin{pmatrix}1\\2\\3\end{pmatrix}$ et $B = \begin{pmatrix}1\\2\\3\end{pmatrix}$, $\cos(\theta) = 1$: Les vecteurs sont identiques.

$A = \begin{pmatrix}1\\2\\3\end{pmatrix}$ et $B = \begin{pmatrix}1\\2\\12\end{pmatrix}$, $\cos(\theta) \approx 0.89$: Le taux de similarité est élevé.

$A = \begin{pmatrix}33\\0\\0\end{pmatrix}$ et $B = \begin{pmatrix}5\\28\\1\end{pmatrix}$, $\cos(\theta) \approx 0.17$: Les vecteurs sont très différents.

6. Algorithme de la plus longue sous-séquence commune (LCS) :

- Aperçu :

Le problème de la plus longue sous-séquence commune (Longest Common Subsequence) est le problème de trouver la plus longue sous-séquence commune à toutes les séquences dans un ensemble de séquences (souvent seulement deux séquences).

Soient X et Y deux suites données
Soient n la longueur de X et m la longueur de Y
Initialiser une table LCS de dimension n*m
Pour k de 0 à n : LCS [0][k] = 0

```
Pour k de 0 à m : LCS [0][k] = 0
Pour i de 1 à n :
        Pour j de 1 à m :
                Comparer X [i] et Y [j]
                  Si X [i] = Y [j] :
                    LCS [i][j] = 1 + LCS [i-1, j-1]
                Autre :
                    LCS [i][j] = max(LCS [i-1][j], LCS [i][j-1])
LCS_longueur = LCS [n][m]
```

- Exemple :

Soit les deux chaines S1= « AATGGCCATA » et S2= « ATATAATTCTAT » :

S1: **A** _ **A** **T** _ G G C **C** _ **A** **T** A n=10
S2: **A** T **A** **T** A A T T **C** T **A** **T** _ m=12

LCS entre ces deux chaines est **« AATCAT »**, sa longueur est 6.

7. Algorithme Coefficient de Dés (DCS):

- Aperçu :

Le coefficient de similitude des dés, également connu sous le nom d'indice Sørensen-Dice ou simplement le coefficient de dés, est un outil statistique pour déterminer à quel point deux ensembles de données sont similaires. Cet index est probablement l'outil le plus utilisé pour valider les algorithmes de segmentation d'images basés sur l'IA, mais c'est une idée beaucoup plus large qui peut être utilisée aux collections de données pour une gamme d'applications, y compris la NLP.

- Formule mathématique :

Soit D la fonction qui implémente l'algorithme de coefficient de dés. On définit D avec deux ensembles A et B.

$$DSC = \frac{2 \cdot |A \cap B|}{|A| + |B|}$$

Formulaire 4: Coefficient de Dés (Formule Générale)

Où $|A|$ et $|B|$ sont les cardinalités des deux ensembles A et B.

Lorsque nous voulons appliquer l'algorithme pour la similarité des textes, Le coefficient peut être calculé pour deux chaînes, en utilisant des bigrams comme suit :

$$DSC = \frac{2 \cdot n_T}{n_A + n_B}$$

Formulaire 5: Coefficient de Dés pour deux chaines

Où $n_A$représente le nombre de bigrams dans la chaîne A, $n_B$représente le nombre de bigrams dans la chaîne B, et n_T représente le nombre total de bigrams commun entre les deux chaînes.

- Exemple :

Soit deux chaînes : « père » et « mère », l'ensemble des bigrammes dans chaque mot est { pè , èr , re } et { mè , èr , re }.
Chaque ensemble a trois éléments, et l'intersection de ces deux ensembles a deux éléments : { èr , re }.
En insérant ces nombres dans la formule, nous trouvons le taux de similarité suivant :

$$DSC = \frac{2 \cdot n_T}{n_A + n_B} = \frac{2 \cdot 2}{3 + 3} \approx 0.67$$

III.Conclusion :

Il existe de nombreux autres algorithmes qui mesurent la similarité entre les textes, y compris Boyer Moore, Rabin Karp et Knuth Morris Pratt …

Dans le dernier chapitre, nous essaierons tous les algorithmes que nous avons définis dans ce chapitre.

Chapitre 4 :
Conception d'Interface Utilisateur

I.Introduction:

La plate-forme que nous avons conçue sera l'interface que nous utiliserons pour saisir les textes, afin que le logiciel se charge ensuite de son rôle d'analyse et de recherche du texte le plus proche du texte saisi.
La réalisation de cette interface nous oblige à avoir des connaissances sur plusieurs domaines, logiciels, outils et langages de programmation.

II.Conception:

Ci-dessous, nous détaillons la conception en utilisant la méthodologie UML.

1. Diagramme de cas d'utilisation :

Le cas d'utilisation est une séquence d'activités organisées en étapes distinctes.
Le diagramme de cas d'utilisation permet de déterminer les interactions entre le système et les acteurs.

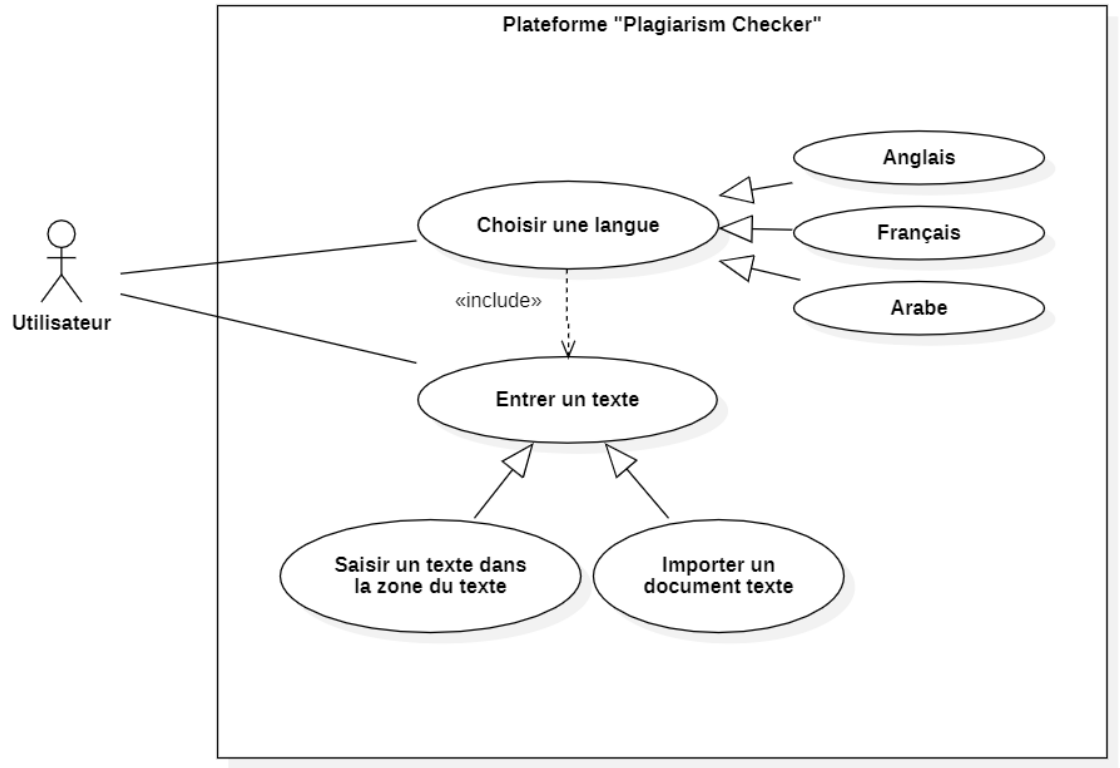


Figure 3 : Diagramme de cas d'utilisation - Générale

Ce diagramme montre que chaque utilisateur doit effectuer certaines fonctions:

- Choisissez la langue avant de saisir les données.
- Saisie des données soit en saisissant dans la zone de texte ou télécharger le fichier

2. Diagramme de séquence :

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système.

Une interaction est un ensemble d'objet qui interagissent en s'échangeant des messages.

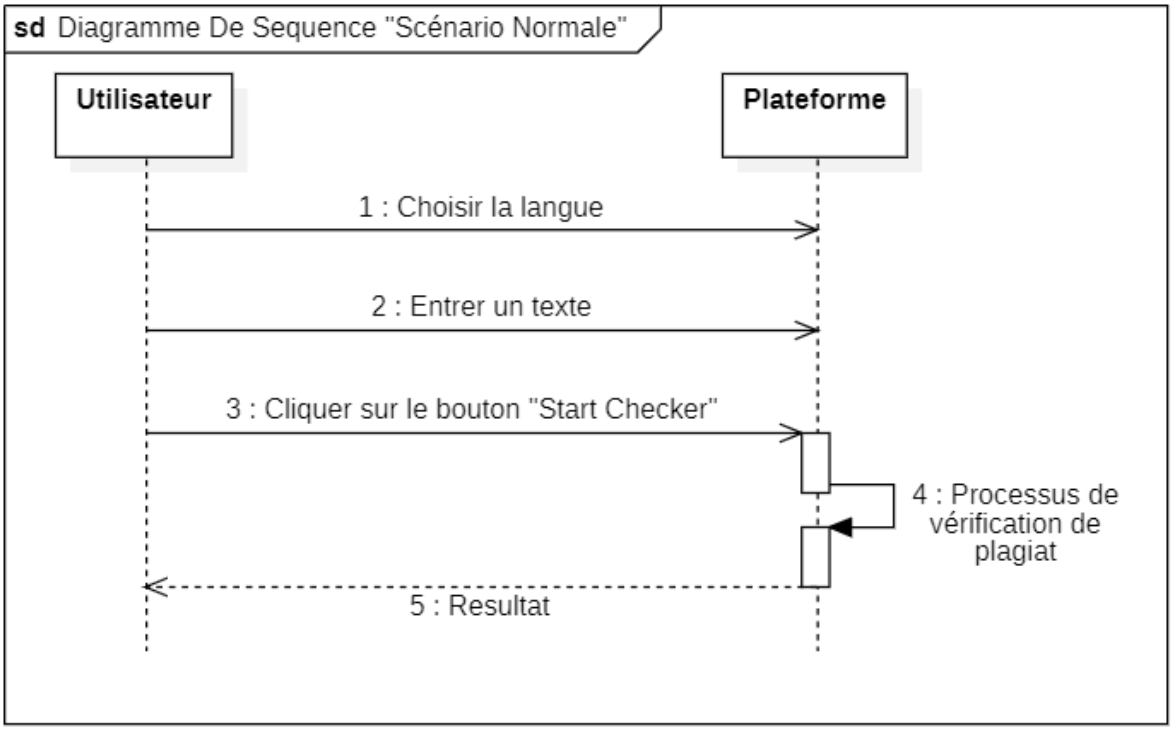


Figure 4 : diagramme de séquence- Scénario normale

Ce diagramme illustre le scénario effectué par le saisi des données :

L'utilisateur sélectionne d'abord la langue, puis il saisit le texte a testé soit en tapant directement dans la zone de texte, ou en important le fichier.

L'utilisateur clique sur le bouton « Start Checker ». Par suite, l'application commence le processus de vérification de plagiat et enfin le résultat qui est le taux du plagiat est affiché.

III. Développement et Réalisation :

1. Outils utilisés pour créer le site Web de projet :

a. Environnement matériel :

Pour la réalisation de ce projet, nous avons disposé d'un ordinateur LENOVO caractérisé par :

- Processeur : Intel(R) Core (TM) I5 2.5 GHz.
- Mémoire : 8 Go de RAM.
- Disque dur : 256Go.
- Système d'exploitation : Windows 10.

b. Environnement logiciel :

Dans ce qui suit, nous présentons l'environnement logiciel utilisé pour mener à terme ce projet :

- **Les Logiciels :**
  - XAMPP :

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.

  - Visual Studio Code :

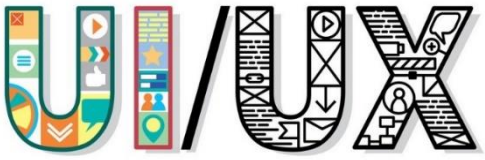
Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

  - PHPMyAdmin :

PHPMyAdmin (PMA) : est une application web de gestion pour les systèmes de gestion de base de données MySQL et Maria DB, réalisée principalement en PHP et distribuée sous licence GNU GPL. Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, gratuits comme payants, le proposent ce qui évite à l'utilisateur d'avoir à l'installer.

- **Design :**
  - UI/UX :

La conception de l'interface utilisateur nécessite une bonne compréhension des besoins de l'utilisateur. Il se concentre principalement sur les besoins de la plateforme et les attentes des utilisateurs.



La conception de l'interface utilisateur (UI) est le processus de création d'interfaces esthétiques. L'objectif de la conception de l'interface utilisateur est de rendre l'interaction utilisateur aussi simple et efficace que possible, en termes d'atteinte des objectifs de l'utilisateur.

L'UX Design se définit comme l'ensemble des moyens mis en œuvre pour concevoir une interface répondant pleinement aux besoins d'usage de chaque utilisateur. L'objectif est de fournir la meilleure expérience utilisateur possible.

- **Les langages de programmation :**
  - HTML5 :

HTML5 est une combinaison de nouvelles balises HTML, de propriété CSS3, de JavaScript et de plusieurs technologies associées mais structurellement séparées de la spécification HTML5.

  - CSS3 :

Le rôle du CSS est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...). Ce langage est le complément du langage HTML pour obtenir une page web avec du style. Le navigateur parcourt le document HTML. Lorsqu'il rencontre une balise, il demande à la CSS de quelle manière il doit l'afficher.

  - JavaScript :

Programmation très récent, créé par les sociétés Netscape et Sun Microsystems vers la fin de l'année 1995.Son objectif principal est d'introduire de l'interactivité avec les pages HTML et effectuer des traitements simples sur le poste de travail de l'utilisateur.

  - PHP :

PHP est un langage de script qui est principalement utilisé pour être exécuté par un serveur HTTP, mais il peut fonctionner comme n'importe quel langage interprété en utilisant les scripts et son interpréteur sur un ordinateur.

  - MySQL :

Le terme MySQL, pour My Structured Query Language, désigne un serveur de base de données distribué sous licence libre GNU (General Public License). Il est, la plupart du temps, intégré dans la suite de logiciels LAMP

qui comprend un système d'exploitation, un serveur web (Apache) et un langage de script (PHP)

- **Technologies et Framework adoptées :**
  - Bootstrap :

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation, et interaction avec la page dans le navigateur ...etc.) des sites et d'application web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres élément interactifs, ainsi que des extensions JavaScript en option.

2. Exposition du travail réalisé :

Dans cette partie, nous présentons notre site web en exposant des captures d'écran.

- Base de données :

La base de données se compose d'une table d'articles qui se compose des attributs id, path et lang.

Figure 5 : Les enregistrements de la table "Articles"

id	path	lang
49	files/1yjNB_What_Is_Natural_Skin_Care.txt	en
50	files/pgAxC_Which_Is_The_Best_Skin_Care_Product.tx...	en
51	files/OFd0i_Women_Fitness.txt	en
52	files/AMeSp_Amara_Camara,_porte-parole_de_la_prési...	fr
53	files/iPmVH_Christophe_Lutundula,_chef_de_la_diplo...	fr
54	files/NkoYo_En_RD_Congo,_le_roi_Philippe_renouvell...	fr
55	files/qBVDN_Le_Sahel_face_à_la_menace_jihadiste.tx...	fr
56	files/cuhpb_Mali__la_junte_se_donne_deux_ans_pour_...	fr
57	files/5Libo_RD_Congo__à_Kinshasa,_le_roi_des_Belge...	fr
58	files/rWU5U_RD_Congo__le_roi_de_Belgique_exprime_s...	fr
59	files/eVMxG_RD_Congo__une_visite_historique_du_roi...	fr
60	files/MX0A6_Sahara_occidental__l'Algérie_suspend_l...	fr
61	files/NoQXS_Scandale_de_corruption_en_Afrique_du_S...	fr
62	files/2kzA5_اتفاقية_شراكة_في_كلية_الحقوق_بالمحمدية...	ar
63	files/Boxki_أحمد_التوفيق_يصدر_رواية_واحة_تيلودا.tx...	ar
64	files/aUQBE_المديلي_يتذكر_الخوري_بمعرض_الكتاب.txt	ar

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	id	int(255)			Non	*Aucun(e)*		AUTO_INCREMENT
2	path	varchar(255)	utf8mb4_general_ci		Non	*Aucun(e)*		
3	lang	varchar(255)	utf8mb4_general_ci		Non	*Aucun(e)*		

Figure 6 : La structure de la table "Articles"

Tableau 11 Dictionnaire de la table Articles

Attribut	Type	Taille	Explication
id	int	255	Représente l'identifiant de chaque article est automatiquement incrémenté, en général c'est la clé primaire.
path	varchar	255	Représente le chemin des fichiers dans un dossier.
lang	varchar	255	Signifie la langue de l'article (En : anglais, Fr : français, Ar : arabe).

Le dossier consiste en un corpus textuel de 70 fichiers de divers domaines et langages.

La taille des articles :

- Anglais : entre 414 à 793 mots.
- Français : entre 80 à 346 mots.
- Arabe : entre 67 à 3870 mots.

Un corpus textuel est un ensemble des documents linguistiques, ils sont utilisés largement pour faire des études statistiques, des tests dans le traitement automatique des langues naturels.

- Interface d’accueil :

L'interface principale du site Web contient la zone de texte, les boutons radio et la zone de téléchargement de fichiers.
Elle est aussi caractérisée par la facilité d'utilisation, en plus que l'utilisateur passera une expérience agréable grâce à utilisation d'UI/UX.

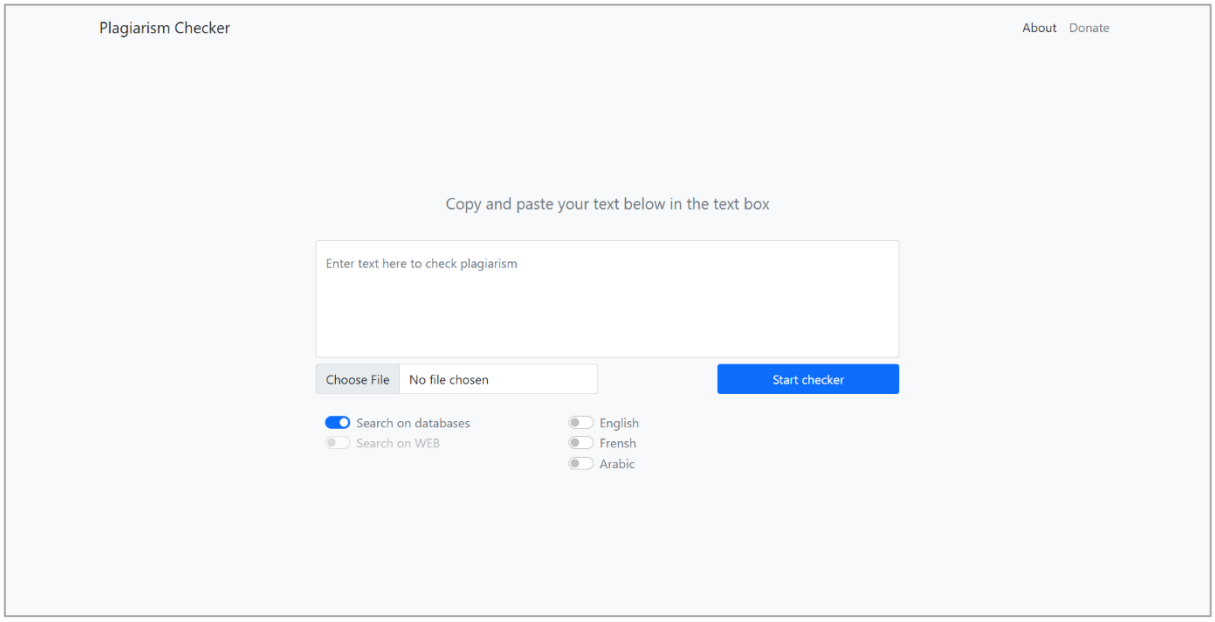
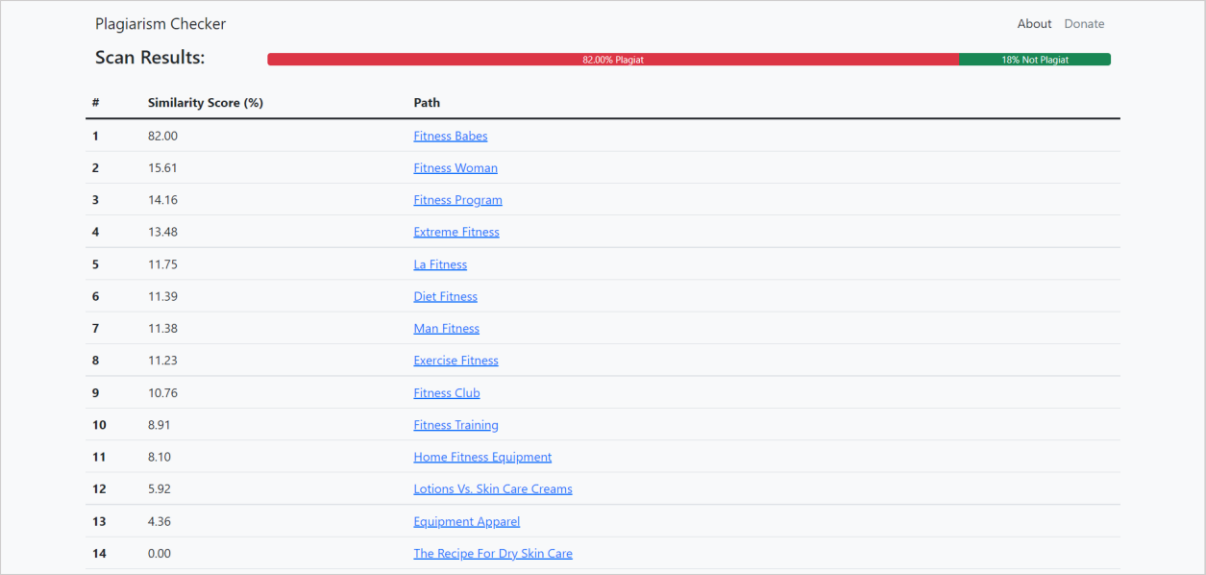


Figure 7 : L'interface d'accueil de la plateforme

- Page de résultat :



#	Similarity Score (%)	Path
1	82.00	Fitness Babes
2	15.61	Fitness Woman
3	14.16	Fitness Program
4	13.48	Extreme Fitness
5	11.75	La Fitness
6	11.39	Diet Fitness
7	11.38	Man Fitness
8	11.23	Exercise Fitness
9	10.76	Fitness Club
10	8.91	Fitness Training
11	8.10	Home Fitness Equipment
12	5.92	Lotions Vs. Skin Care Creams
13	4.36	Equipment Apparel
14	0.00	The Recipe For Dry Skin Care

Figure 8 : Page du resultat du vérificateur

L'affichage sous la forme d'une barre de progression divisée en deux parties, la couleur rouge indique le pourcentage de plagiat et la couleur verte indique le pourcentage de non-plagiat.

Au bas de la barre de progression se trouve un tableau contenant le chemin de tous les fichiers de la base de données, ainsi que le pourcentage de plagiat pour chaque fichier.

Les fichiers sont classés par ordre décroissant, donc au début du tableau se trouvent les fichiers avec le taux de plagiat le plus élevé.

IV.Conclusion :

A travers ce chapitre, nous avons présenté notre proposition de conception de site.
Nous avons présenté tous ce qui est essentiel dans la conception avec des diagrammes UML ; qui nous ont permis de définir une vue statique ainsi qu'une vue dynamique du site.
Aussi, nous décrivons à la fois les environnements matériels et logiciels sur lesquels nous avons construit notre application.
Enfin, nous avons expliqué les fonctionnalités importantes de la plateforme en fournissant des captures d’écrans.

Chapitre 5 :
Implémentation,
Expérimentation et Discussion

I.Introduction:

Dans ce chapitre, Nous intéressons à installer tout ce qu'on a mentionné dans les chapitres précédents.
Nous passerons en revue tous les outils que nous avons utilisés lors de nos expérimentations, et nous comparerons les différents résultats que nous avons obtenus afin de conclure l'efficacité des différents algorithmes et outils utilisés.

II. Implémentation, Expérimentation et Discussion:

1. Implémentation :

Nous avons collecté quelques articles du même domaine, ce n'était pas un choix arbitraire, mais plutôt afin de mieux étudier la précision des algorithmes de détermination de la source du texte que nous avons testés en présence de textes contenant des mots du même domaine.

Ces textes ont été importé sur la base de données associée à la plateforme qui a été développée.

Pour tester l'efficacité des algorithmes, nous avons soigneusement compilé et rédigé certains textes et les avons classés en 4 catégories, un texte qui est exactement le même qu'un texte dans la base de données (100.00% plagiat), un autre qui est complètement différent de tous les textes de la base de données (0% plagiat).
Nous avons combiné les deux textes précédents pour obtenir les deux autres catégories : un texte plagié (presque 75%-85%) et un autre qui contient un peu de texte plagié (presque 15-25%).

Ce travail a été réalisé en 3 langues différentes : Anglais, Français, Arabe.

Diverses méthodes de prétraitement avec divers algorithmes ont été appliqués sur les textes de base de données et le texte qui sera testé.

Tableau 12: Prétraitement utilisé pour chaque algorithme

<table>
<tr><th>Algorithme</th><th>Prétraitement utilisé</th><th>Extraction des caractéristiques utilisé</th></tr>
<tr><td rowspan="2">STRCMP
Similar Text
Levenshtein
Jaccard Index
LCS
Dice Coefficient</td><td rowspan="2">Normalisation
Lemmatisation
Stemming
Suppression des mots vides</td><td>Tokenisation n-grams</td></tr>
<tr><td>TCV
TF-IDF</td></tr>
<tr><td>Cosine Similarity</td><td></td></tr>
</table>

Concernant la tokenisation n-grams, nous avons basé notre choix de la valeur de n sur le débit de mots de la langue sur laquelle nous allons travailler :

- Anglais : 15-20 mots, nous avons choisi 15.
- Français : 15 mots.
- Arabe : 15 mots.

La méthode TF-IDF dans notre étude a été modifié par nous pour avoir un bon résultat. A la place d'utiliser le vocabulaire des deux textes ; on a pris seulement le vocabulaire du premier texte.

Nous allons suivre la procédure suivant :

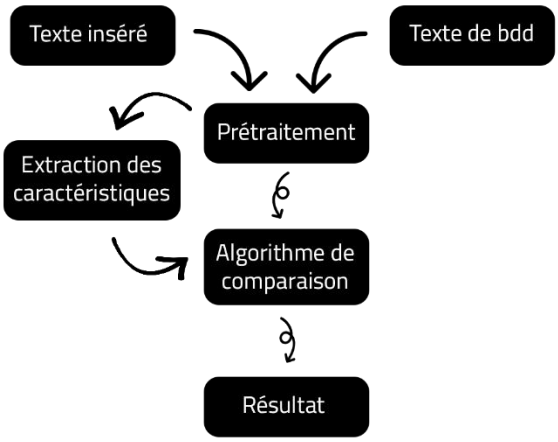


Figure 9: Organigramme d'application

2. Expérimentation :

Nous avons obtenu les résultats suivant afin de tester les 4 textes (pour chaque langue) :

Tableau 13: Résultats du test avec des textes copiés en anglais

Texte en anglais				
Algorithme Utilisé	Taux de plagiat			
	Texte complétem-ent plagié	Texte plagié 1	Texte plagié 2	Texte non plagié
	73 mots	68 mots	72 mots	67 mots
Par visualization (Valeurs approximatives)	100.00% (1)	80.00% (1)	20.00% (1)	0.00% (0)
STRCMP	100.00% (1)	73.68% (1)	0.00% (0)	0.00% (0)
Similar Text	100.00% (1)	91.00% (1)	52.55% (1)	48.46% (0)
LEVENSHTEIN	100.00% (1)	88.98% (1)	44.42% (1)	38.80% (0)
JACCARD	100.00% (1)	82.00% (1)	14.97% (1)	7.54% (0)
LCS	100.00% (1)	91.39% (1)	56.30% (1)	51.94% (0)
Dice	100.00% (1)	85.90% (1)	22.17% (1)	14.99% (0)
Cosine	92.39% (1)	85.51% (1)	22.55% (0)	14.30% (0)
Cosine + TCV	100.00% (1)	78.00% (1)	25.25% (1)	12.47% (0)
Cosine + TF-IDF	100.00% (1)	91.91% (1)	46.08% (1)	19.93% (0)

1 : L'algorithme a trouvé la source du texte testé

0 : L'algorithme n'a pas trouvé la source du texte testé

Tableau 14 : Résultats du test avec des textes copiés en français

Texte en français				
Algorithme Utilisé	Taux de plagiat			
	Texte complétem-ent plagié	Texte plagié 1	Texte plagié 2	Texte non plagié
	97 mots	98 mots	83 mots	73 mots
Par visualization (Valeurs approximatives)	100.00% (1)	80.00% (1)	20.00% (1)	0.00% (0)
STRCMP	100.00% (1)	76.42% (1)	18.48% (1)	0.00% (0)
Similar Text	100.00% (1)	91.95% (1)	60.50% (1)	48.05% (0)
LEVENSHTEIN	100.00% (1)	89.74% (1)	50.26% (1)	37.06% (0)
JACCARD	100.00% (1)	83.15% (1)	33.04% (1)	14.98% (0)
LCS	100.00% (1)	92.36% (1)	62.65% (1)	51.22% (0)
Dice	100.00% (1)	86.62% (1)	32.39% (1)	14.99% (0)
Cosine	100.00% (1)	87.98% (1)	41.45% (1)	29.04% (0)
Cosine + TCV	95.13% (1)	95.13% (1)	38.08% (1)	22.65% (0)
Cosine + TF-IDF	100.00% (1)	96.15% (1)	65.02% (1)	32.17% (0)

1 : L'algorithme a trouvé la source du texte testé

0 : L'algorithme n'a pas trouvé la source du texte testé

Tableau 15: Résultats du test avec des textes copiés en arabe

Texte en arabe				
Algorithme Utilisé	Taux de plagiat			
	Texte complétem-ent plagié	Texte plagié 1	Texte plagié 2	Texte non plagié
	123 mots	117 mots	147 mots	143 mots
Par visualization (Valeurs approximatives)	100.00% (1)	80.00% (1)	20.00% (1)	0.00% (0)
STRCMP	100.00% (1)	68.12% (1)	15.76% (1)	0.00% (0)
Similar Text	100.00% (1)	88.57% (1)	62.62% (1)	54.55 % (0)
LEVENSHTEIN	100.00% (1)	86.88% (1)	58.25% (1)	48.12% (0)
JACCARD	100.00% (1)	75.84% (1)	23.54% (1)	7.69% (0)
LCS	L'algorithme ne fonctionne pas correctement			
Dice	100.00% (1)	75.87% (1)	60.23% (1)	30.26% (0)
Cosine	100.00% (1)	79.13% (1)	27.18% (1)	13.94% (0)
Cosine + TCV	99.44% (1)	95.70% (1)	30.62% (1)	12.43% (0)
Cosine + TF-IDF	100.00% (1)	98.06% (1)	40.40% (1)	22.60% (0)

1 : L'algorithme a trouvé la source du texte testé

0 : L'algorithme n'a pas trouvé la source du texte testé

Après, nous avons pris les méme textes, nous l'avons reformulé et réessayé, et nous avons obtenu les résultats suivants :

Tableau 16: Résultats du test avec des textes paraphrasés en anglais

Texte en anglais				
Algorithme Utilisé	Taux de plagiat			
	Texte complétem-ent plagié	Texte plagié 1	Texte plagié 2	Texte non plagié
	60 mots	62 mots	68 mots	66 mots
Par visualization (Valeurs approximatives)	100.00% (1)	80.00% (1)	20.00% (1)	0.00% (0)
STRCMP	0.00% (1)	0.00% (1)	0.00% (0)	0.00% (0)
Similar Text	69.72% (1)	67.26% (1)	51.99% (1)	49.33% (1)
LEVENSHTEIN	59.31% (1)	58.16% (1)	40.09% (1)	38.86% (1)
JACCARD	42.60 % (1)	36.27% (1)	10.24% (1)	7.25% (0)
LCS	70.43% (1)	68.29% (1)	55.61% (1)	53.04% (1)
Dice	31.29% (1)	29.51% (1)	17.56% (1)	15.92% (0)
Cosine	57.96% (1)	49.87% (1)	16.92% (1)	12.99% (0)
Cosine + TCV	54.14% (1)	46.36% (1)	21.72% (0)	9.30% (0)
Cosine + TF-IDF	84.80% (1)	70.71% (1)	32.85% (1)	16.04% (0)

1 : L'algorithme a trouvé la source du texte testé
0 : L'algorithme n'a pas trouvé la source du texte testé

Tableau 17: Résultats du test avec des textes paraphrasés en français

Texte en français				
Algorithme Utilisé	Taux de plagiat			
	Texte complétem-ent plagié	Texte plagié 1	Texte plagié 2	Texte non plagié
	97 mots	98 mots	83 mots	73 mots
Par visualization (Valeurs approximatives)	100.00% (1)	80.00% (1)	20.00% (1)	0.00% (0)
STRCMP	0.00% (0)	0.00% (0)	0.00% (0)	0.00% (0)
Similar Text	69.46% (1)	62.21% (1)	51.56% (1)	47.00% (1)
LEVENSHTEIN	60.75% (1)	55.22% (1)	41.11% (1)	37.09% (1)
JACCARD	36.02% (1)	28.47% (1)	16.74% (1)	10.68% (0)
LCS	69.81% (1)	64.68% (1)	54.78% (1)	50.44% (1)
Dice	29.8% (1)	23.02% (1)	14.27% (1)	12.08% (0)
Cosine	50.89% (1)	44.95% (1)	24.22% (1)	22.12% (0)
Cosine + TCV	55.08% (1)	34.09% (1)	19.52% (1)	10.42% (0)
Cosine + TF-IDF	73.58% (1)	55.79% (1)	34.87% (1)	19.96% (0)

1 : L'algorithme a trouvé la source du texte testé
0 : L'algorithme n'a pas trouvé la source du texte testé

Tableau 18: Résultats du test avec des textes paraphrasés en arabe

Texte en arabe				
Algorithme Utilisé	Taux de plagiat			
	Texte complétem-ent plagié	Texte plagié 1	Texte plagié 2	Texte non plagié
	124 mots	102 mots	153 mots	119 mots
Par visualization (Valeurs approximatives)	100.00% (1)	80.00% (1)	20.00% (1)	0.00% (0)
STRCMP	4.62% (1)	0.00% (0)	0.00% (0)	0.00% (0)
Similar Text	84.82% (1)	64.58% (1)	57.92% (1)	54.91 % (0)
LEVENSHTEIN	77.00% (1)	57.21% (1)	50.25% (1)	48.22% (0)
JACCARD	44.83% (1)	20.66% (1)	9.68% (1)	6.79% (0)
LCS	L'algorithme ne fonctionne pas correctement			
Dice	36.55% (1)	17.00% (1)	17.91% (1)	16.15% (0)
Cosine	59.36% (1)	33.06% (1)	15.95% (1)	12.27% (0)
Cosine + TCV	67.92% (1)	41.56% (1)	14.63% (1)	15.01% (0)
Cosine + TF-IDF	81.25% (1)	53.33% (1)	21.48% (1)	22.22% (0)

1 : L'algorithme a trouvé la source du texte testé

0 : L'algorithme n'a pas trouvé la source du texte testé

3. Discussion :

Après avoir observé les résultats des expériences et analysé chacune d'entre elles, nous sommes parvenus aux conclusions suivantes :

En commençant par le test du contenu copié sans reformulation :

Pour l'anglais, les résultats de Jaccard étaient satisfaisants, et l'algorithme de Jaccard était le plus précis.

Quant au français, nous avons remarqué une supériorité de précision pour la fonction Strcmp, bien qu'elle soit traditionnelle, et qu'il ne faut pas toujours s'y fier, notamment lors du traitement de textes paraphrasés. Ensuite les algorithmes de Jaccard, Dice avec leurs résultats proches qui sont peu satisfaisants.

Enfin, la langue arabe où l'algorithme LCS a pris plus de temps que d'habitude dans son processus, il a donc été arrêté et il n'y a pas de résultats. La précision était cette fois aussi en faveur des algorithmes Jaccard, Strcmp.

Pour le test avec des textes paraphrasés :

Les résultats ont été désastreux pour la plupart des algorithmes dans toutes les langues, mais il semble que Cosine Similarity avec ses différentes combinaisons (TCV et TF-IDF) ait eu de meilleurs résultats que les autres.

Cette fois, l'algorithme Jaccard n'était pas supérieur, Il a été surmonté par les résultats de l'algorithme précédent. Les résultats de Jaccard ont montré la différence entre les textes testés en valeurs, mais il n'a pas été en mesure d'obtenir des résultats proches des valeurs approximatives.

IV. Conclusion :

Nous avons remarqué que tous les algorithmes peuvent retrouver les textes plagiés et identifier leur source, mais leur faiblesse réside dans l'identification des textes non-plagié, ce qui est normal car ce sont des algorithmes statistiques.

Et selon les résultats obtenus des expérimentations, et après analyse et discussion de ces résultats, il s'avère que s'appuyer sur l'algorithme de Cosine Similarity en combinaison avec l'extraction des caractéristique avec TF-IDF sera meilleur dans la détection des textes paraphrasés. Mais pour les textes copié il sera mieux d'utiliser coefficient de Jaccard.

Conclusion Générale et Perspective

Le plagiat est toute utilisation non autorisée de tout ou partie d'un article sans donner une crédibilité appropriée à l'auteur original. Ainsi, les créateurs de contenu, y compris les chercheurs et les étudiants, doivent connaître l'importance du plagiat. Cela affecte un écrivain qui chérit son écriture et peut également changer négativement la carrière d'un créateur de contenu. Par conséquent, ils doivent faire attention au plagiat.

Indépendamment du fait que le plagiat est une forme de vol et est un crime puni par la loi

De nos jours, les progrès technologiques ont facilité la recherche de contenu en double dans un article. Les lecteurs peuvent utiliser l'outil de vérification du plagiat pour découvrir l'originalité du contenu. Pour d'autres conséquences, cela peut affecter l'intégrité académique, et donc l'université peut prendre des mesures contre les étudiants.

Notre étude a commencé par une étude du sujet et des problèmes soulevés. Ensuite, nous avons pris des cours dans les domaines et les outils nécessaires pour faire l'étude. Puis nous nous sommes lancés dans l'application, qui comprenait à la fois du traitement de texte et de l'extraction de caractéristiques, pour les soumettre à des algorithmes afin de calculer le taux de plagiat en se basant sur la comparaison de textes deux à deux.

De plus, une interface utilisateur Web compatible avec tous les appareils et respectant les règles de conception UI/UX a été conçue pour une utilisation facile.

En conclusion, cette étude a atteint des résultats plutôt satisfaisants dans la détermination du taux de plagiat pour des textes de contenu copier/coller et des textes reformulés. Et pour rappel, nous sommes intéressés à mener d'autres tests pour obtenir un bon résultat dans la détection du plagiat.

Notre étude a également démontré que les algorithmes sont non seulement responsables du taux de précision accru, mais qu'il existe également un chevauchement significatif du prétraitement et de l'extraction des caractéristiques.

Malheureusement, nous n'avons pas eu assez de temps pour l'expérience car la recherche dans ce domaine a demandé plus de temps.

Comme perspectives sur le projet, nous aspirons à utiliser d'autres algorithmes, à utiliser le "Web Scraping", et aussi à découvrir les textes traduits et pourquoi pas utiliser le Machine Learning pour détecter le plagiat.

Bibliographie Et References

Asra, M., Hardik , G., & Mohammed, A. M. (2021, Septembre 09). *Document Plagiarism Detection Tool using Edit Distance Text.* Disponible sur irjet: https://www.irjet.net/archives/V8/i9/IRJET-V8I9201.pdf

Dragut, A. (s.d.). *Data mining : streams et similarités.* Disponible sur lis-lab: https://pageperso.lis-lab.fr/andreea.dragut/enseignementWebMining/r/tp4.html

FABIEN, M. (2019, Novembre 03). *Traitement Automatique du Langage Naturel en français (TAL / NLP)*. Disponible sur stat4decision: https://www.stat4decision.com/fr/traitement-langage-naturel-francais-tal-nlp/

Hiten, C., Mohd, T., Rutuja, K., & Nikita, C. (2021, Avril 4). *Plagiarism Detector Using Machine Learning.* Disponible sur ijresm: https://www.journals.resaim.com/ijresm/article/download/677/650

Indice et distance de Jaccard. (s.d.). Disponible sur wikipedia: https://fr.wikipedia.org/wiki/Indice_et_distance_de_Jaccard

Lina, F. (2020, jully Décembre). *NLP- Natural Language Processing : Introduction*. Disponible sur datascientest: https://datascientest.com/introduction-au-nlp-natural-language-processing

Madan, R. (2019, May 30). *TF-IDF/Term Frequency Technique: Easiest explanation for Text classification in NLP using Python (Chatbot training on words)*. Disponible sur medium: https://medium.com/analytics-vidhya/tf-idf-term-frequency-technique-easiest-explanation-for-text-classification-in-nlp-with-code-8ca3912e58c3

Nishimura, M. (2020, 8 26). *The Best Document Similarity Algorithm: A Beginner's Guide*. Disponible sur towardsdatascience: https://towardsdatascience.com/the-best-document-similarity-algorithm-in-2020-a-beginners-guide-a01b9ef8cf05

Plagiat : qu'est-ce que le plagiat ? (s.d.). Disponible sur scribbr: https://www.scribbr.fr/category/le-plagiat/

Similar Text Function. (s.d.). Disponible sur php documentation: https://www.php.net/manual/en/function.similar-text.php

Sørensen–Dice coefficient. (s.d.). Disponible sur wikipedia: https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient