الجمـــهوريـــة الجـــزائريـــة الديـمقراطيـــــــة الشـعبيـــــة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعـلـيـم الـعـالـــي و الـبـحـث الـعـلـمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جـامـعـة عمّـــار ثـلـيـــجـــي بالأغـــواط
UNIVERSITE AMAR TELIDJI LAGHOUAT
كـلـيـــــة الـعـلـــــوم
FACULTE DES SCIENCES
DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

Mémoire de MASTER

Domaine : Mathématiques et Informatique
Filière : Informatiques
Option : Systèmes d'Information et de Décision

Par:
Ahmed chaouch Aymen

THEME

Etude comparative entre deux techniques d'extraction automatique des mots-clés

Soutenu publiquement le 12-07-2021 devant le jury composé de:

Mr Mohamed El Habib MAICHA	*M.A.(A)*	*Président*
Mr. Younes GUELLOUMA	*M.C.(A)*	*Examinateur*
Mr Mustapha BOUAKKAZ	*M.C.(A)*	*Encadreur*

N 06 Année Universitaire 2020/2021

Remerciements

Tout d'abord nous rendons grâce à Dieu, lui qui nous a permis d'être bien portant afin d'effectuer ce travail du début jusqu'à la fin.

Nous remercions nos parents pour leurs soutiens durant notre parcours de formation.

Nos remerciements vont, à notre directeur de mémoire, le professeur ***Mr* Mustapha BOUAKKAZ** qui nous a guidé avec ses orientations, ses conseils et ses critiques tout au long de ce travail de recherche en nous laissant la liberté dont on avait besoin.

On ne peut que lui être reconnaissant surtout pour ses qualités intellectuelles et humaines.

Nous remercions également tous mes enseignants.

Nos remerciements vont aussi aux membres du jury, pour l'honneur qu'ils nous ont fait en acceptant d'évaluer ce travail et de participer à la soutenance.

Dédicaces

JE DÉDIE CE MÉMOIRE à :

Mes deux grands-pères Ali et Rachid qui nous ont quitté l'an dernier, que dieu les accueille dans son vaste paradis.

Mes chers parents, pour tous leurs sacrifices, leur soutien tout au long de mes études.

A Mes grands-mères Khadoudja et Safia que dieu les garde.

Ainsi que toute ma famille

Je vous Remercie tous

الملخص

في هذه المذكرة ذكرنا طرق استخراج الكلمات المفتاحية أليا في العقد الماضي، اخترنا طريقتين مختلفتين من اجل تفصيلهما بالإضافة الى أمثلة لكل منهما.

قمنا بتقييم الطريقتين بواسطة استخراج الكلمات المفتاحية من المقالات العلمية من خلال مقارنة الكلمات المفتاحية المستخرجة تلقائيًا

والكلمات المفتاحية المعينة من طرف الكاتب.

الكلمات المفتاحية: الكلمات المفتاحية – المقالات العلمية – الكلمات المفتاحية التلقائية – الكلمات المفتاحية من طرف الكاتب

Resume :

Dans ce memoire on a cité des méthodes d'extarction de mots clés de la dérniere décennie,on a opté pour deux méthodes differentes RAKE et TextRank qu'on a détaillé avec des exemples pour chacune.

On a évalué les deux méthodes sur des articles scientifiques, en comparant les mots clés extraits automatiquement et ceux associés initialement à ces articles.

Mots clés : extarction de mots clés , RAKE , TextRank , articles scientifiques

Abstract :

In this memory we have cited methods of keyword extraction of the last decade, we have opted for two different methods RAKE and TextRank that we have detailed with examples for each one.

The two methods were evaluated on scientific articles, comparing the keywords extracted automatically and those initially associated with these articles.

Keywords : keyword extraction , RAKE, TextRank , scientific articles

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction générale

1.1 Contexte général

La Recherche d'information est une activité dont la fonction est de localiser et de délivrer un ensemble de documents à un utilisateur en fonction de son besoin en informations, parmi le volume important de documents disponibles le défi est de trouver ceux qui correspondent au mieux à l'attente de l'utilisateur.

C'est un domaine qui permet de retrouver des informations dans un corpus, ce dernier est composé de plusieurs documents qui peuvent être relationnels ou non structurés. Le contenu des documents peut être du texte, des sons, des images ou des données.

Historiquement la recherche d'information était liée à la bibliothéconomie qui vise à représenter des documents dans le but d'en récupérer des informations, au moyen de la construction d'index.

La recherche d'information est née aux Etats-Unis, très peu de temps après l'avènement des premiers ordinateurs. Si la période des années 1980 fut très fructueuse, la période actuelle, que nous situons du début des années 1990 à nos jours, est marquée par une véritable explosion du domaine de la recherche d'information.

L'informatique a permis le développement d'outils pour traiter l'information et établir la représentation des documents au moment de leur indexation, ainsi pour faciliter les recherches [Chiaramella 2007] .

1.2 Problématique

Les informations peuvent être des documents non structurés qui sont des données représentées ou stockées sans format prédéfini.
Elles sont typiquement constituées de documents textes ou multimédias, mais peuvent également contenir des dates, des nombres et des faits.
Cette absence de format entraîne des irrégularités et des ambiguïtés qui peuvent rendre difficile la compréhension des données, contrairement au cas des données stockées dans des tableurs ou des bases de données c'est le cas des données structurées.
Les documents non structurés sont difficiles à exploiter pour surmonter ce défi et aller chercher l'information pertinente là où elle se trouve, et de tirer le meilleur parti de ces données, on utilise des méthodes qui règlent ce problème parmi elles le recours à l'extraction des mots clés. L'extraction des mots clés permet de relever ce qui est important d'une grande quantité de données non structurées, il existe plusieurs méthodes qui sont utilisées. L'objectif de ce mémoire est de comparer certaines méthodes d'extraction des mots clés pour déduire leurs performances et leur complémentarité.

Chapitre 2

État de l'art

Sommaire

2.1 Extraction de mots-clés

Dans cette partie, nous présentons des méthodes d'extraction automatique de mots-clés qui sont réparties en deux catégories : les méthodes supervisées et les méthodes non supervisées.

Les méthodes non supervisées d'extraction des mots-clés ne tiennent compte ni du domaine ni de la langue des documents à analyser.

Avant l'extraction des mots clés, il y a tout d'abord une extraction des termes candidats.

Cette étape d'extraction de mots candidats est trés importante car c'est seulement dans ces termes là que nous extrayons les mots clés, c'est-à-dire que si les mots clés ne se sont pas présent dans ces mots candidats alors ils ne seront pas choisis. [Ramiandrisoa]

2.2 Extraction des mots candidats

Les mots candidats sont des unités textuelles pouvant devenir des mots clés (groupes de mots).

C'est la méthode qui utilise les plus longues séquences de noms et d'adjectifs pour l'extraction des mots candidats. [Daille 2017]

2.3 Méthodes d'extraction de mots-clés

2.3.1 Tableau des méthodes

Méthode	Année	Technique
TF-IDF	2013	Graphe
TextRank	**2004**	**Graphe**
SingleRank	2008	Graphe
TopicRank	2013	Graphe
K-core	2015	Graphe
YAKE	2018	N-gramme
RAKE	**2012**	**Statistique des mots**

TABLE 2.1 – Tableau des méthodes

2.3.2 TF-IDF

TF-IDF (ou Term Frequency Inverse Document Frequency) mesure la pertinence et l'importance d'un mot ou d'un groupe de mots dans un document donné.

$$TF - IDF(mot) = TF(mot) \log(\frac{N}{DF(mot)})$$

TF : nombre d'occurrences de mot dans le document analysé.

DF : nombre de documents dans lequel le mot est présent.

N : nombre total de document.

Cette méthode est à base de graphe noté G (N, A) où N est l'ensemble des noeuds et A l'ensemble de ses arcs sortants et entrants.

Chaque sommet du graphe représente un mot-candidat et la constitution des arêtes est propre à chaque méthode. [Labiad 2017]

2.3.3 TextRank

TextRank est un algorithme de classement basé sur des graphe, issu du PageRank de Google utilisé dans l'extraction de mots clés et la synthèse de texte.

PageRank est un algorithme utilisé pour calculer le poids et le classement des pages Web. [Mihalcea 2004]

TextRank est basé sur le calcul du score d'importance des sommets du graphe-of-words en utilisant le principe de vote ou de recommandation entre deux sommets (Mihalcea et al,2004) et inspiré de l'algorithme Pagerank (Page et al, 1999).

TextRank utilise une représentation efficace d'un document, elle peut aussi être utilisée pour faire des résumés automatiques d'un document.

Cette méthode comporte les étapes suivantes :

- Prétraitement du texte
- Construction du graphe
- Calcul les scores des sommets
- Extractions des mots clés.

2.3.3.1 Prétraitement du texte

Transformer tous les caractères du texte en minuscule.

Supprimer les mots bruts du texte, et ne laisser que les mots candidats pour l'extraction.

2.3.3.2 Construction du graphe

Les mots candidats du texte sont considérés comme les sommets du graphe,deux sommets sont reliés par une arête.

2.3.3.3 Calculer les scores des sommets

Au départ, les scores de tous les sommets du graphe sont initialisés a 1.

Un algorithme de classement calcule les scores de chaque sommet à chaque itération et s'arrête lorsque le seuil donné est atteint.

Le score d'un sommet $S(V_i)$ est calculé avec la formule :

$$S(V_i) = (1-d) + d \sum_{j \in ln(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

$S(V_i)$: Le poids du mot (i).

d : facteur d'amortissement qui peut être réglé entre 0 et 1,

Le facteur d est généralement réglé à 0,85 (Brin et Page, 1998).

$ln(V_i)$: Liens entrants de la somme i du graphe.

$Out(V_j)$: Liens sortant de j.

$|Out(V_j)|$: Le nombre de liens sortant

La formule suivante est utilisée dans le cas où les arêtes reliant les sommets sont pondérées :

$$WS(V_i) = (1-d) + d \sum_{V_{j \in ln(V_i)}} \frac{w_{ji}}{\sum_{V_{k \in Out(V_j)}} w_{jk}} WS(V_j)$$

w_{ij} :La force de la connexion entre x_i et x_j (w_{ij} poids qui relie les deux sommets).

2.3.3.4 Extraction des mots clés

C'est à partir des sommets les plus importants (selon leur score par ordre décroissant) que sont choisis les mots-clés. Les séquences des mots (des sommets importants) adjacents dans le texte constituent les mots clés composés de plusieurs mots, les autres mots non adjacents dans le texte qui obtiennent les meilleurs scores sont également retenus comme mots-clés.

2.3.4 SingleRank

SingleRank est une modification de la méthode TextRank, la différence se trouve dans la pondération des arêtes du graphe de mots et dans l'extraction des mots-clés à partir des mots-candidats.

Dans certains cas, cette méthode donne de meilleurs résultats que TextRank.

L'inconvénient de cette méthode est qu'elle favorise les mots-candidats longs qui sont formés par de nombreux mots.

SingleRank a les mêmes étapes que TextRank sauf le calcul des scores des sommets (les score des mots clés) et elle utilise la formule suivante :

$$S(C_i) = (1-\lambda) + \lambda \sum_{C_{j \in A_{entrant(C_i)}}} \frac{S(C_j)P_{j,i}}{\sum_{C_{k \in A_{sortant(C_j)}}} P_{j,k}}$$

λ : facteur d'atténuation.

p_{ij} : poids de l'arc allant du noeud c_j vers le noeud.

c_i, correspondant au nombre de cooccurrences entre les deux mots i et j.

Les scores des mots candidats constitués de plusieurs mots sont calculés par :

$$MotScore(P_i) = \sum S(C_j)$$

p_{ij} :Mot candidat constitué de plusieurs mots.

c_i :Des mots composant le mot candidat pi.

L'extraction des mots clés :

Les mots candidats qui sont les scores les plus importants ce sont des mots clés. [Ramiandrisoa 2016]

2.3.5 TopicRank

TopicRank basé sur TextRank, est différent des autres méthodes qui utilisent les graphes.

Elle fait des recherches sur les sujets importants par rapport à TextRank et SingleRank.

Les avantages de TopicRank : suppression des problèmes de redondance dans les mots-clés extraits, le graphe est plus compact avec des poids des arêtes plus renforcés, un ordonnancement de meilleure qualité, il comporte trois étapes : identification des sujets, ordonnancement des sujets, sélection des mots-clés (Bougouin et al, 2013).

2.3.5.1 Identification des sujets

Un sujet est une information spécifique ou générale transportée au minimum par un texte.

Deux mots clés C_1 et C_2 sont groupés par la similarité de Jaccard :

$$sim(C_1, C_2) = \frac{\|C_1 \cap C_2\|}{\|C_1 \cup C_2\|}$$

$C_1 \cap C_2$:nombre de mots composant C_1 et C_2

$C_1 \cup C_2$:nombre de mots commun à C_1 et C_2

Dès que toutes les paires de mots candidats similaires sont connues, l'algorithme de classification ascendante est appliqué, chaque mot candidat est considéré comme un groupe et puis les deux groupes présentant la plus forte similarité sont réunis en un seul, on continu le regroupement jusqu'à ce que le nombre de groupes soit atteint. [Bougouin 2013]

La similarité entre deux groupes est obtenue en calculant la similarité entre les mots candidats composant chaque groupe.

La valeur de similarité entre deux groupes peut être obtenue en choisissant parmi les trois façons suivantes :

Simple : On retient la plus grande valeur de similarité.

Complète : On retient la plus petite valeur de similarité.

Moyenne : On retient la moyenne de toutes les similarités.

Une fois les sujets connus, un nouveau graphe est défini comme suit : G = (N, A) :

N= ensemble des sujets du document

A= ensemble des liens entre les noeud.

2.3.5.2 Ordonnancement des sujets

La pondération des arêtes est très importante durant cette étape.

C'est la force du lien sémantique entre les noeud du graphe qui est considérée comme poids d'une arête.

$$poids(S_i, S_j) = \sum_{C_i \in S_i} \sum_{C_j \in S_j} dist(C_i, C_j)$$
$$dist(C_i, C_j) = \sum_{p_i \in pos(C_i)} \sum_{p_j \in pos(C_j)} \frac{1}{|p_i - p_j|}$$

$poids(S_i, S_j)$: Le poids de l'arrêt entre les sujets est utilisé.

$dist(C_i, C_j)$: La force sémantique entre les mots candidats C_i et C_j.

$pos(C_i)$:La position de mot candidat C_i dans le document analysé.

L'ordonnancement est basé sur le principe de vote, c'est à dire qu'un noeud (sujet) est très important s'il est fortement connecté avec plusieurs autres noeuds importants.

$$importance(S_i) = (1 - \lambda) + \lambda \sum_{S_j \in V_j} \frac{poids(s_i, s_j) importance(S_j)}{\sum_{s_k \in V_j} poids(s_j, s_k)}$$

V_i : l'ensemble de sujets reliés au sujet S_i.

λ :le facteur d'atténuation.

2.3.5.3 Sélection des mots-clés

Pour choisir le mot clé qui représente le mieux un sujet on a trois méthodes :

- **Première position :** le mot candidat d'un sujet qui apparait le premier dans le document est sélectionné.
- **Fréquence :** le mot candidat d'un sujet le plus fréquent dans le document analysé est sélectionné.

— **Centroide :** le mot candidat d'un sujet le plus similaire aux autres mots candidats du même sujet est sélectionné.

2.3.6 K-core :

K-core est une méthode d'extraction de mots clés basée sur le graphe dont la construction est semblable à celle de TextRank ou SingleRank.

Pour calculer les scores d'importance des sommets on utilise l'algorithme de Batagelj et Zavernik (Batagelj et al, 2011). Le problème est qu'elle dépend de la fenêtre de co-occurrence de mots.

Construction du graphe :

Les mots candidats représentent les sommets du graphe et deux sommets sont reliés si les mots candidats représentant ces sommets co-occurrent dans une fenêtre de N mots.

Ces liens sont pondérés par le nombre de co-occurrences des mots qu'ils relient dans le document, on parle de W-Kcore.

Par contre dans le cas d'un graphe non pondéré ils seront tous pondérés par 1, on parle de K-core. [Tixier 2016]

Une fois le graphe constitué on établit l'algorithme ce dernier attribut un nombre n à chaque sommet du graphe correspondant au K-core graphe auquel il appartient.

Puis l'ordonnancement des sommets se fait par ordre décroissant des nombres n et les top m sont retenus comme mots clés.

2.3.7 Yet Another Keyword Extractor (Yake)

YAKE est une méthode d'extraction automatique de mots-clés légère qui repose sur des caractéristiques de texte statistiques extraites de documents pour extraire les mots-clés. [Campos 2018]

Cette méthode comporte cinq étapes principales :

2.3.7.1 Le prétraitement du texte et l'identification du mot candidat

Le prétraitement divise le texte en mots individuels chaque fois qu'un délimiteur d'espace vide ou de caractère spécial est trouvé.

2.3.7.2 prétraitement du texte et l'identification du mot candidat

On conçoit un ensemble de cinq caractéristiques qui sont :

— **Boîtier** (reflète l'aspect de la case d'un mot.)

— **Mot positionnel** (les mots apparaissant au début d'un document sont les plus pertinents).

— **Fréquence des mots** (marquant plus les mots qui se re produisent plus souvent).

— **Lien entre les mots et le contexte** (calcule le nombre de mot différents qui apparaissent à gauche du mot candidat).

— **Mot DifSentence** (calcule la fréquence à laquelle un mot candidat apparaît dans différentes phrases).

2.3.7.3 Calcul du score du mot

Nous combinons toutes ces caractéristiques en une seule mesure de telle sorte que chaque mot se voit attribuer un score $S(w)$.
Ce poids alimentera le processus de génération des mots-clés.

2.3.7.4 Génération de n-grammes et calcul du score du mot-clé candidat

Ici, nous considérons une fenêtre glissante de 3 grammes, générant ainsi une séquence contiguë de mots-clés candidats de 1, 2 et 3 grammes.
Chaque mot-clé candidat se verra alors attribuer un $S(kw)$ final, tel que plus le score est petit, plus le mot-clé sera significatif.
L'équation 1 formalise ceci :

$$S(Kw) = \frac{\prod_{w \in kw} S(w)}{TF(kw)(1+\sum_{w \in kw} S(w))}$$

$S(kw)$: Le score d'un mot-clé candidat.

$S(w)$: Le score du premier mot du mot-clé candidat.

$TF(kw)$: Le mot fréquence du mot-clé.

2.3.7.5 La déduplication et le classement des données.

Nous éliminons les candidats similaires provenant des étapes précédentes.
Pour cela, nous utilisons la distance de Levenshtein. Enfin, le système affichera une liste de mots-clés pertinents, formée par 1, 2, 3-grammes, de sorte que plus le score $S(kw)$ est bas, plus le mot-clé sera important.

2.3.8 Rapid automatic keyword extraction (RAKE)

L'extraction automatique rapide de mots clés (RAKE) est un algorithme extrêmement efficace qui fonctionne sur plusieurs types de documents. [Rose 2010]
Les paramètres d'entrée de l'algorithme RAKE comprennent une liste de mots vides ainsi qu'un ensemble de délimiteurs de phrases et de délimiteurs de mots.
La liste de mots vides : Une liste de mots prédéfinie par des spécialistes de la langue qui contient des non significatifs figurant dans un texte.
Les délimiteurs de mots : espace, virgule, point, point-virgule, slash ... etc.
Il utilise des mots vides et des délimiteurs de phrases pour découper le document en mots-clés candidats qui sont principalement les mots qui aident un développeur à extraire le mot-clé exact nécessaire pour obtenir des informations du document.
Mots-clés candidats :
Cela se fait essentiellement par les étapes suivantes : Le texte est divisé en mots par les délimiteurs de mots spécifiques et divisé par mot brut,ensuite les mots qui se trouvent dans la même séquence se voient attribuer la même position dans le texte et sont considérés ensemble comme un mot clé candidat.
Scores des mots clés : Après avoir identifier tous les mots-clés candidats à partir des données de texte, un graphique de co-occurrence de mots est généré qui calcule le

score pour chaque mot-clé candidat et défini comme le score de mot membre.
À l'aide de ce graphique, nous évaluons plusieurs métriques pour calculer les scores de mots, en fonction du degré et de la fréquence des sommets du graphique.

2.4 Les mesures d'évaluation

La performance de chaque méthode d'extraction automatique de mots clés est évaluée en comparant les mots clés trouvés manuellement par les auteurs avec ceux extraits automatiquement pour chaque document [Mothe 2018] à travers les mesures suivantes :

2.4.1 Rappel

Défini le nombre de mots clés importants retrouvés par rapport au nombre total de mots clés de référence du document (mots clés auteurs). [Grouin 2011]

$$\frac{VraiPositif}{VraiPositif+FauxNegatif}$$

$VraiPositif$: Le résultat de l'extraction des mots clés automatique est identique au résultat des mots clés extraits manuellement.
$FauxNegatif$: Le résultat de l'extraction des mots clés automatique est contraire au résultat des mots clés extraits manuellement.

2.4.2 Précision

Défini le nombre de mots clés importants retrouvés par rapport au nombre total de mots clés extraits. [Brownlee 2020]

$$\frac{VraiPositif}{VraiPositif+FauxPositif}$$

$fauxPositif$: Le résultat de l'extraction des mots clés automatique est contraire au résultat des mots clés extraits manuellement.

2.4.3 F-Mesure

La moyenne harmonique du Rappel et de la Précision, pondérés de façon égale. [Nakache 2005]

$$\frac{2*Rappel*Prcision}{Rappel+Prcision}$$

Chapitre 3

Implémentation des méthodes

Sommaire

3.1 Implémentation de RAKE

Nous avons le texte suivant :

Compatibility of systems of linear constraints over the set of natural numbers.

Criteria of compatibility of a system of linear Diophantine equations, strict inequations,and nonstrict inequations are considered.

Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given.

These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types. [Kryvyi 2002]

3.1.1 Prétraitement du texte

— Convertir tout le texte en minuscules.

— Utiliser la liste des mots vides pour éliminer les mots bruits.

— Diviser le texte en mots par les délimiteurs de mots spécifiés.

— Déterminer les mots candidats.

Maintenant, à partir du tableau ci-dessus, nous avons tous les mots candidats :

Compatibility - systems - linear constraints - set - natural numbers - Criteria - compatibility - system - linear Diophantine equations - strict inequations - nonstrict inequations - considered - Upper bounds - components - minimal set - solutions - types - algorithms - construction - minimal generating sets - solutions - systems - criteria - corresponding algorithms - constructing - minimal supporting set - solutions - solving - considered types - systems - systems - mixed types

Ces mots-clés candidats ne sont que des mots-clés de sortie par RAKE Mais RAKE fait un pas de plus en calculant le score pour chaque mot-clé.

3.1.2 Tableau de co-occurrence des mots

Pour tracer le tableau de co-occurrence remplir les cases(mot-mot) pour le nombre de répétitions du mot dans le texte(**linear-linear=2(le mot linear Exister deux fois dans le texte**),et remplir les cases qui représentent les mots-clés candidats par 1 (.**mixed types :'mixed-types'et'types-mixed'=1.** la Figure3.1

	compatibility	systems	linear	constraints	set	natural	numbers	criteria	system	diophantine	equations	strict	inequations	nonstrict	considered	upper	bounds	construction	minimal	generating	sets	solutions	corresponding	Algorithms	constructing	Supporting	Solving	types	mixed
compatibility	2																												
systems		4																											
linear			2	1						1	1																		
constraints			1	1																									
set					3														2										
natural						1	1																						
numbers						1	1																						
criteria								2																					
system									1																				
diophantine			1							1	1																		
equations			1							1	1																		
strict												1	1																
inequations												1	2	1															
nonstrict													1	1															
considered															2														
upper																1	1												
bounds																1	1												
construction																		1											
minimal					2														3	1	1				1				
generating																			1	1	1								
sets																			1	1	1								
solutions																						3							
corresponding																							1						
Algorithms																							1	2					
constructing																									1				
Supporting																										1			
Solving																											1		
types																												3	1
mixed																												1	1

FIGURE 3.1 – Tableau de co-occurrence des mots

Maintenant, pour calculer le score de mot, nous devons calculer la fréquence et le degré des mots.

3.1.3 Fréquence des mots (freq(w))

C'est le compte qui indique combien de fois un mot particulier est apparu parmi tous les mots-clés candidats :
Prendre simplement la valeur de cette ligne mot-mot.

3.1.4 Degré de mots (deg (w))

Pour calculer le degré de mot pour un mot particulier dans le tableau ci-dessus, additionner tous les nombres par rangée.

3.1.5 Score des mots clés

Score du mot clé = $\frac{Deg(w)}{freq(w)}$

	compatibility	Systems	linear	constraints	set	natural	numbers	criteria	system	diophantine	equations	strict	inequations	nonstrict	considered	upper	bounds	construction	minimal	generating	sets	solutions	corresponding	algorithms	constructing	supporting	solving	types	mixed
Deg(w)	2	4	5	1	6	2	2	2	1	3	3	2	4	2	2	2	2	1	8	3	3	3	2	3	1	3	1	4	2
Freq(w)	2	4	2	1	3	1	1	2	1	1	1	1	2	1	2	1	1	1	3	1	1	3	1	2	1	1	1	3	1
Score du mot clé	1	1	2.5	1	2	2	2	1	1	3	3	2	2	2	1	2	2	1	2.7	3	3	1	2	1.5	1	3	1	1.3	2

FIGURE 3.2 – Scores de mots calculés à partir du tableau de co-occurrence de mots.

3.1.6 Les scores des mots clés candidats

Pour calculer le score de mots on additionne les scores des mots, par exemple :
Keyword score ('upper bounds') = (2 + 2) = 4

mots clés candidats	Score des mots clés
minimal generating sets	8.7
linear diophantine equations	8.5
minimal supporting set	7.7
minimal set	4.7
linear constraints	4.5
natural numbers	4
strict inequations	4
nonstrict inequations	4
upper bounds	4
corresponding algorithms	3.5
mixed types	2.3
considered types	2.3
set	2
Algorithms	1.5
Types	1.3
Compatibility	1
Systems	1
Criteria	1
System	1
Components	1
Constructing	1
Solving	1
Solutions	1
Construction	1

TABLE 3.1 – Tableau de score de mots clés candidats

3.1.7 Mots-clés attribués par RAKE

Nous avons choisi les mots clés du TOP 10 :

— minimal generating sets

— linear diophantine equations

— minimal supporting set

— minimal set

— linear constraints

— natural numbers

— strict inequations

— nonstrict inequations

— upper bounds

— corresponding algorithms

3.2 Implémentation de TextRank

3.2.1 Implémentation des formules de TextRank

3.2.1.1 La formule 01 (les graphes non pondérés)

Le score d'un sommet $S(V_i)$ est calculé avec la formule :

$$S(V_i) = (1-d) + d \sum_{j \in ln(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

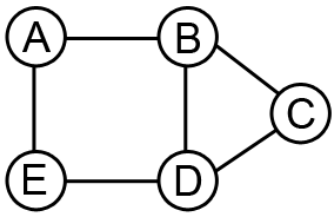

FIGURE 3.3 – Graphe non pondéré

Nous pouvons réécrire la partie sommation de la fonction ci-dessous dans une version plus simple.

$$l_n(v_E) = \{A, D\}$$

$$\sum_{j\in\{A,D\}} \frac{1}{|Out(V_j)|} S(V_j) = \frac{1}{|Out(V_A)|} S(V_A) + \frac{1}{|Out(V_D)|} S(V_D)$$

$$= \frac{1}{|\{B,E\}|} S(V_A) + \frac{1}{|\{B,C,E\}|} S(V_D)$$

$$= \frac{1}{2} S(V_A) + \frac{1}{3} S(V_D)$$

Selon le $\frac{1}{|Out(V_j)|}$ de la fonction, nous devons normaliser chaque colonne :

	A	B	C	D	E
A	0	1/3	0	0	1/2
B	1/2	0	1/2	1/3	0
C	0	1/3	0	1/3	0
D	0	1/3	1/2	0	1/2
E	1/2	0	0	1/3	0

TABLE 3.2 – Matrice représentant les résultats de $\frac{1}{|Out(V_j)|}$ du graphe non pondéré

Pour calculer les poids des noeuds ,nous multiplions le Table 3.4 avec les poids des noeuds (au debut les scores seront 1) comme dans la Figure3.4 .

$$\mathbf{(1-0.85)+0.85} * \begin{bmatrix} \mathbf{0} & \mathbf{0.33} & \mathbf{0} & \mathbf{0} & \mathbf{0.5} \\ \mathbf{0.5} & \mathbf{0} & \mathbf{0.5} & \mathbf{0.33} & \mathbf{0} \\ \mathbf{0} & \mathbf{0.33} & \mathbf{0} & \mathbf{0.33} & \mathbf{0} \\ \mathbf{0} & \mathbf{0.33} & \mathbf{0.5} & \mathbf{0} & \mathbf{0.5} \\ \mathbf{0.5} & \mathbf{0} & \mathbf{0} & \mathbf{0.33} & \mathbf{0} \end{bmatrix} * \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{0.85} \\ \mathbf{1.28} \\ \mathbf{0.71} \\ \mathbf{1.28} \\ \mathbf{0.85} \end{bmatrix}$$

FIGURE 3.4 – La premiere étape pour calculer les poids des noeuds

On remplace les poids des noeuds par les résultats obtenus jusqu'on aura la stabilisation des valeurs. Table 3.3

Fois / Poids	A	B	C	D	E
1	0.8555	1.2805	0.711	1.2805	0.8555
2	0.8727	1.1749	0.8683	1.1749	0.8727
3	0.8504	1.2195	0.8091	1.2195	0.8504
4	0.8535	1.1974	0.8341	1.1947	0.8535
5	**0.8486**	1.2031	**0.8217**	1.2031	**0.8486**
6	0.8481	**1.1974**	0.8249	**1.1974**	0.8481
7	0.8463	1.1969	0.8217	1.1969	0.8463

TABLE 3.3 – Tableau pour calculer les poids des noeuds

— Le poids de A est 0,8486.

— Le poids de B est 1.1974.

— Le poids de C est 0.8217.

— Le poids de D est 1.19.

— Le poids de E est 0.8486.

3.2.1.2 La formule 02 (les graphes pondérés)

$$WS(V_i) = (1-d) + d \sum_{V_{j \in ln(V_i)}} \frac{w_{ji}}{\sum_{V_{k \in Out(V_j)}} w_{jk}} WS(V_j)$$

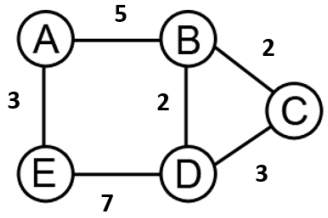

FIGURE 3.5 – Graphe pondéré

Nous pouvons utiliser la Table 3.3 pour représenter les liens entrants et sortants entre A, B, C, D, E dans le graphique.

$$l_n(V_E) = \{A, D\}, j \in \{A, D\}$$

$$Out(V_j) = \{Out(A), Out(D)\}, k \in \{\{B, E\}, \{B, C, E\}\}$$

$$w_{jk} = poidEntreV_jetV_k$$

$$\sum_{V_{k\in\{\{B,E\},\{B,C,E\}\}}} w_{jk=} w_{AB} + w_{AE} + w_{DB} + w_{DC} + w_{DE}$$

$$\sum_{V_{j\in\{A,D\}}} \frac{w_{ji}}{\sum_{V_{k\in\{\{B,E\},\{B,C,E\}\}}} w_{jk}} WS(V_j) = \frac{w_{AE}}{20} WS(V_A) + \frac{w_{DE}}{20} WS(V_D)$$

$$= \frac{3}{20} WS(V_A) + \frac{7}{20} WS(V_D)$$

Nous appliquons cette formule chaque fois jusqu'à ce que la valeur du noeud est fixé.

Fois / Poids	A	B	C	D	E
1	0.5078	0.4560	0.3523	0.3622	0.3858
2	0.3291	0.2994	0.2567	0.2941	0.2997
3	0.2687	0.2480	0.2182	0.2650	0.2717
4	0.2457	0.2305	0.2057	0.2516	0.2601
5	**0.2380**	**0.2237**	0.2008	0.2504	**0.2563**
6	0.2349	0.2215	**0.1991**	**0.2498**	0.2548
7	0.2339	0.2206	0.1984	0.2497	0.2543

TABLE 3.4 – Matrice des poids

— Le poids de A est 0.2380.

— Le poids de B est 0.2237.

— Le poids de C est 0.1991.

— Le poids de D est 0.2498.

— Le poids de E est 0.2563.

3.2.2 Exemple pour Implémentation de TextRank

On a le texte suivant :

Compatibility of systems of linear constraints over the set of natural numbers.

Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered.

Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given.

These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types. [Kryvyi 2002]

3.2.2.1 Prétraitement du texte

- Séparer et transformer tous les caractères du texte en minuscule.
- Lemmatisation de texte :

  Dans la lemmatisation, les différentes contreparties grammaticales d'un mot seront remplacées par un seul lemme de base,

  Par exemple :« glasses » peut être remplacé par « glass ».
- Sélectionnez le type de mots de texte :

  Pour annuler les mots bruts de texte il suffit d'avoir le type de mots on utilise la bibliothèque NLTK (Natural Language Toolkit) pour symboliser une phrase en mots et en ponctuation. [Bal 2009]
- Supprimer les mots bruts

3.2.2.2 Construction du graphe

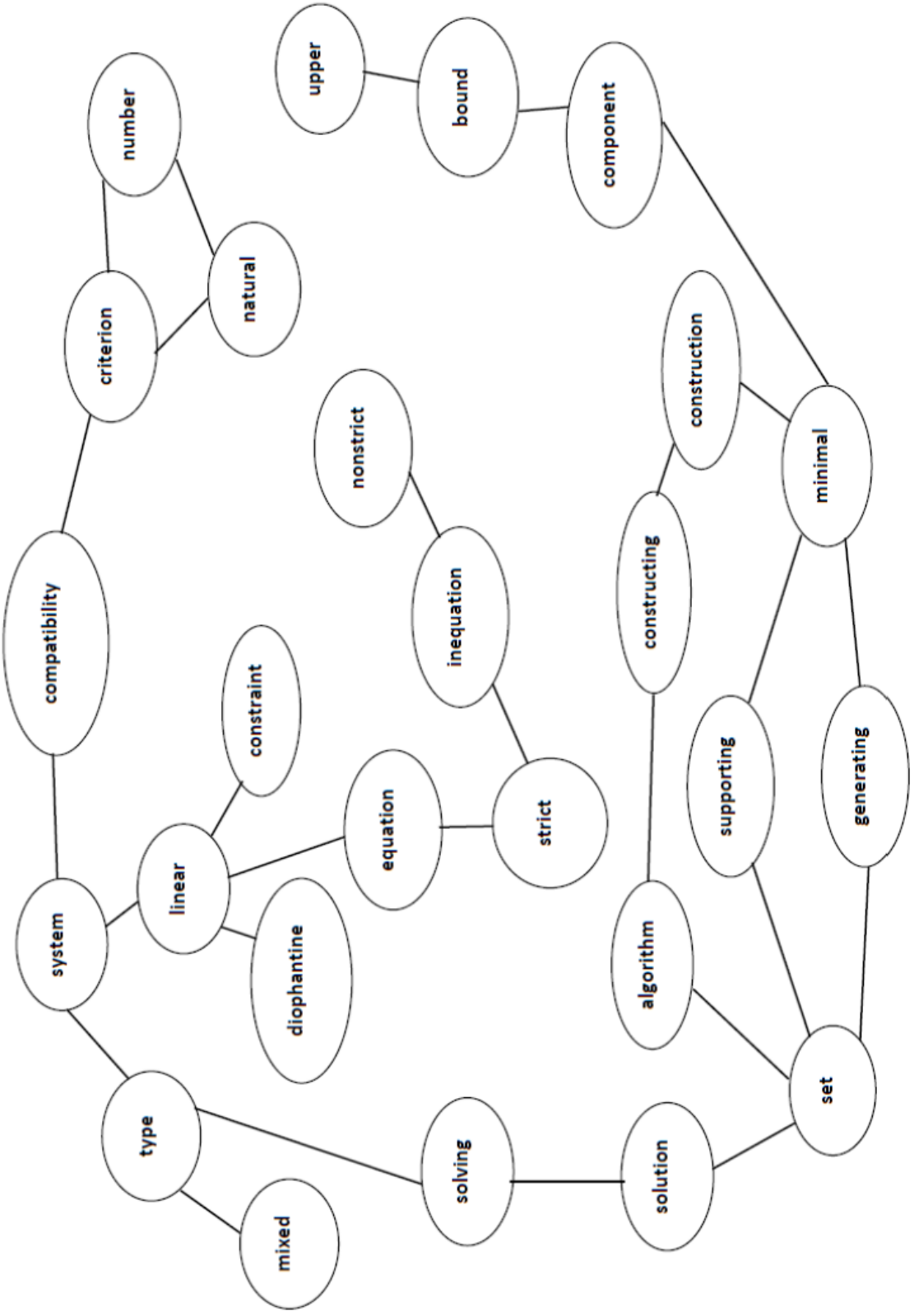

FIGURE 3.6 – Graphe des mots candidats

3.2.2.3 Calculer les scores des noeuds

Mot	Score (poids) de noeud
Set	2.2977562
System	2.1569276
Minimal	1.8108287
Solution	1.70508
Inequations	1.3114841
Linear	1.2844685
Criterion	1.255111
Algorithm	1.2310454
Type	1.0949165
Compatibility	0.95768553
Nonstrict	0.8289814
Strict	0.8260911
Upper	0.8191686
Equation	0.8029257
Bound	0.78934455
Diophantine	0.7637958
Component	0.7420974
Number	0.696681
Natural	0.69546485
Corresponding	0.68675214

TABLE 3.5 – Tableau de score des mots

3.2.2.4 Extractions des mots clés

Pour calculer le score de mots on additionne les scores des noeuds.

Les mots clés	Le score
minimal supporting set	4.7701258063316345
minimal generating set	4.767701208591461
minimal set	4.108584880828857
linear diophantine equation	2.8511900305747986
system	2.1569275856018066
nonstrict inequations	2.140465497970581
strict inequations	2.1375752091407776
linear constraint	1.9650202989578247
corresponding algorithm	1.9177975058555603
solution	1.7050800323486328
upper bound	1.6085131764411926
natural number	1.3921458721160889
mixed type	1.332220420241356
criterion	1.2551109790802002
compatibility	0.9576855301856995
component	0.7420973777770996
constructing	0.6864285469055176
construction	0.6693120002746582
Solving	0.6489318013191223

TABLE 3.6 – Tableau de score finale des mots clés

3.2.2.5 Mots-clés attribués par TextRank

Nous avons choisi les mots clés du TOP 10 :

— minimal supporting set.

— minimal generating set .

— minimal set .

— linear diophantine equation .

— nonstrict inequations.

— strict inequations .

— linear constraint.

— corresponding algorithm .

— solution

— upper bound

Chapitre 4

Expérimentations et résultats

Sommaire

4.1 Introduction

On a utilisé deux articles scientifiques pour comparer les deux méthodes RAKE et TextRank on a obtenu les résultats suivants :

4.2 La premiere étude

4.2.1 Description de l'article

L'article scientifique utilisé pour cette étude est « Integration of scientific research training into undergraduate medical education : a reminder call » .

Le resumé est :

«There is an increasingly growing trend towards integrating scientific research training into undergraduate medical education.

The importance and compulsoriness of this trend has been greatly highlighted at the Boyer Commission on Educating Undergraduates in the Research University. Despite the importance and benefits of undergraduate research, attempts of medical schools to encourage undergraduates to take part in formal research training during undergraduate medical education remain unsatisfactory.

This article serves as a 'reminder call' highlighting the requisite to integrate scientific research training into undergraduate medical curricula.»

Pour plus de détail se referer [Abu-Zaid 2013].

4.2.2 Les résultats des études

4.2.2.1 Diagramme à barre de rappel

Dans la Figure4.1 les deux méthodes ont un rappel maximal au niveau 5 et diminue jusqu'au Top 15 .

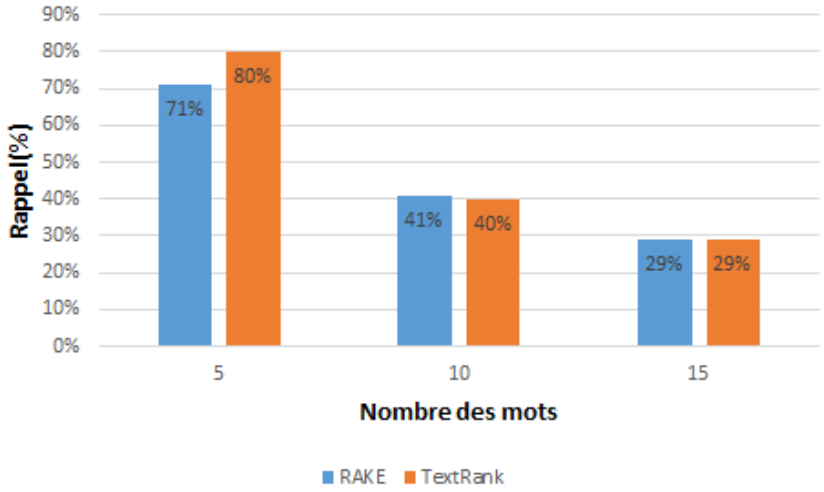

FIGURE 4.1 – Diagramme à barre de rappel

4.2.2.2 Diagramme à barre de précision

Dans la Figure4.2 La précision est stable dans les trois niveaux pour les deux méthodes.

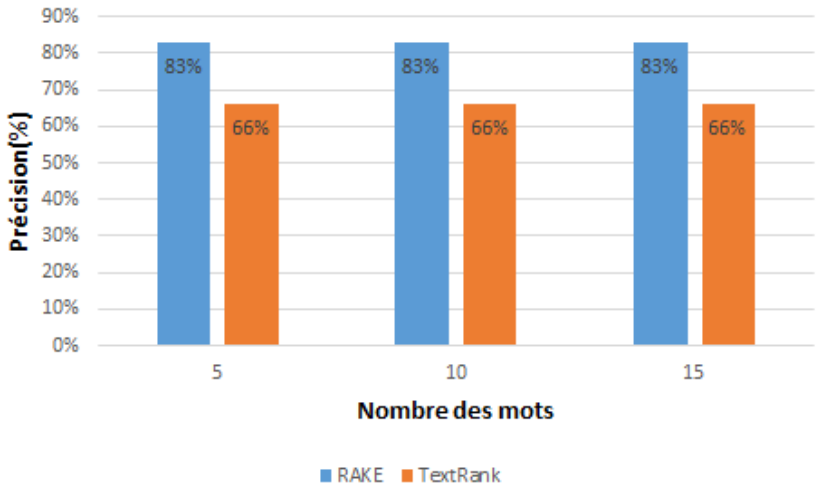

FIGURE 4.2 – Diagramme à barre de Précision

4.2.2.3 Diagramme à barre de F-mesure

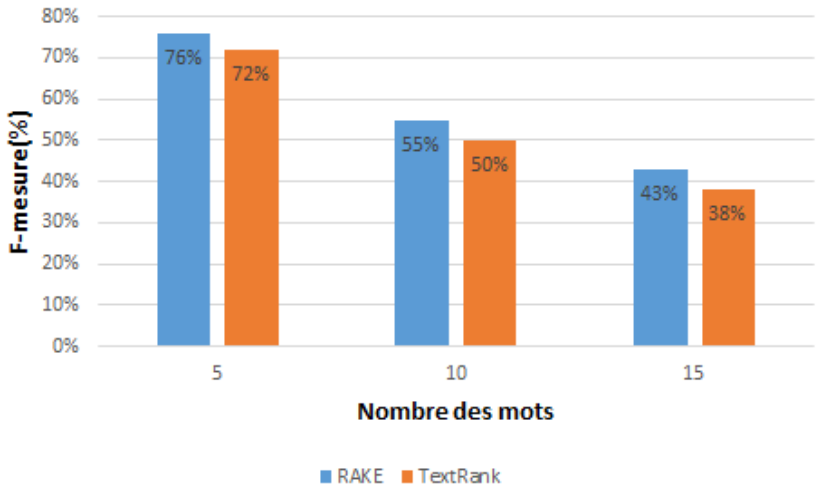

FIGURE 4.3 – Diagramme à barre de F-mesure

On remarque dans cette Figure4.3 que les deux méthodes ont des valeurs maximales au niveau de Top 5 et diminue jusqu'au Top 15 .

4.2.3 Conclusion de La premiére étude

— Les résultats obtenus par la méthode RAKE sont légerement supérieurs à ceux de TextRank.

— Les deux méthodes sont efficaces au niveau des TOP 5 et Top 10 et inefficaces au niveau du Top 15.

4.3 La deuxième étude

4.3.1 Description de l'article

L'article scientifique utilisé pour cette étude est « Scientific Mapping on the Impact of Climate Change on Cultural and Natural Heritage : A Systematic Scientometric Analysis »

Le resumé est :

The world's cultural and natural heritage has been gradually affected by climate change,and although the research agendas of many countries have included this reality since 2003, there is still an incipient approach to it, with analysis techniques used being limited. In addition, there are very few case studies that describe in detail the adaptation processes of spaces to these new conditions. The aim of this research is to identify the scientific production related to the impact of climate change on cultural and natural heritage indexed in the international databases Scopus and Web of Science (WoS), which will enable to establish maturity of the research on this subject. The methodology used for the analysis of the data obtained is bibliometric analysis ; evaluative and relational measures are applied to a set of 78 articles (45 in Scopus and 33 in WoS) and to a joint base of 47 articles after deleting those articles that overlap in both databases. The result is a scientific mapping that enables observing of the evolution of knowledge generation in this field of study. The main findings show that research is incipient, with a large presence of transient authors with a single publication, the research is limited to the geographical scope of Europe and North America, neglecting many other areas, the impact which is measured by the citation of articles is very low, the relational measures corroborate that the thematic approach is new by identifying a high presence of isolated relationships among authors. The results obtained will be very useful for researchers working in this scientific area, as they can find a synthesis of scientific production in this document, allowing them to draw their own conclusions regarding the current gaps in research ; constituting the starting point of their research, with the aim of filling these gaps.

Pour plus de détail se referer [Maldonado-Erazo 2021].

4.3.2 Les résultats des études

4.3.2.1 Diagramme à barre de rappel

On obtient le graphe suivant :

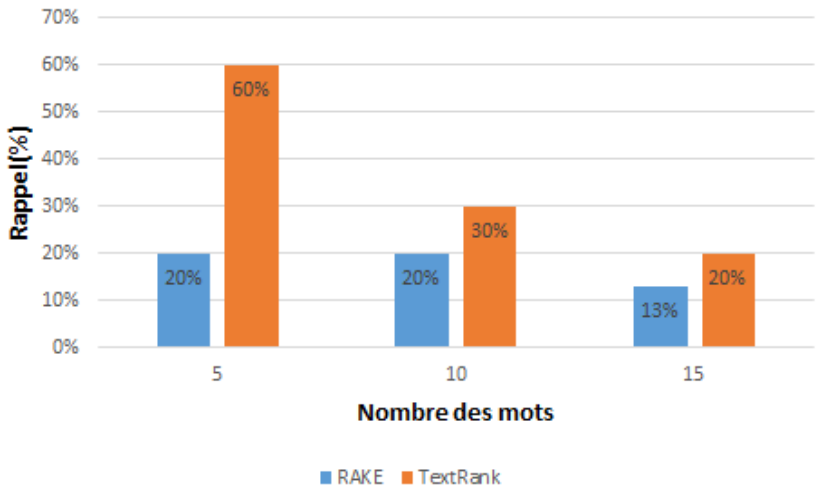

FIGURE 4.4 – Diagramme à barre de rappel

On déduit de La Figure4.4 :

La méthode RAKE : Le rappel des mots clés est stable au niveau des Top 5 et Top 10 avec une diminution de rappel après le Top 10.

La méthode TextRank : Le rappel est maximal au niveau de Top 5 et diminue jusqu'au Top 15.

4.3.2.2 Diagramme à barre de précision

On obtient le graphe suivant :

Dans le Figure4.5 on remarque que :

Pour RAKE La précision est faible au niveau de Top 5 et augmente au niveau Top 10 et stable du Top 10 au Top 15.

Pour TextRank La précision est constante du Top 5 au Top 15.

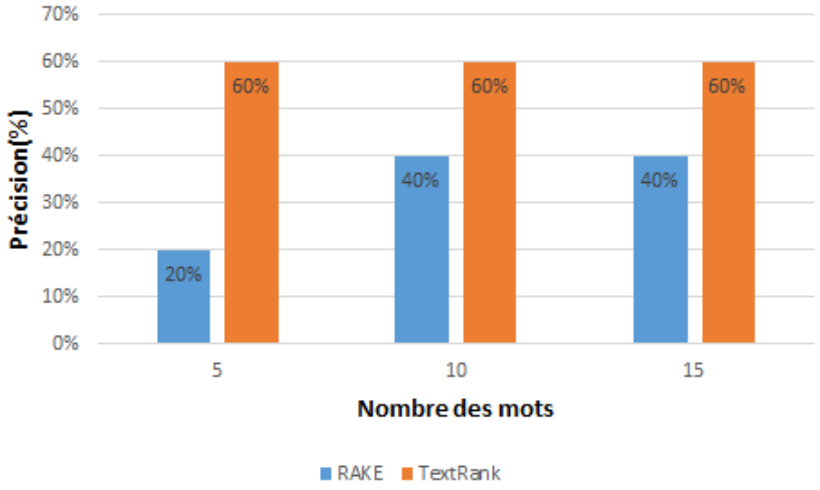

FIGURE 4.5 – Diagramme à barre de précision

4.3.2.3 Diagramme à barre de F-mesure

Pour RAKE le F-mesure augmente du Top 5 au Top 10 et diminue du Top 10 au Top 15.

Pour TextRank Le F-mesure est maximal au Top 5 et diminue au Top 5 jusqu'au Top 15 .

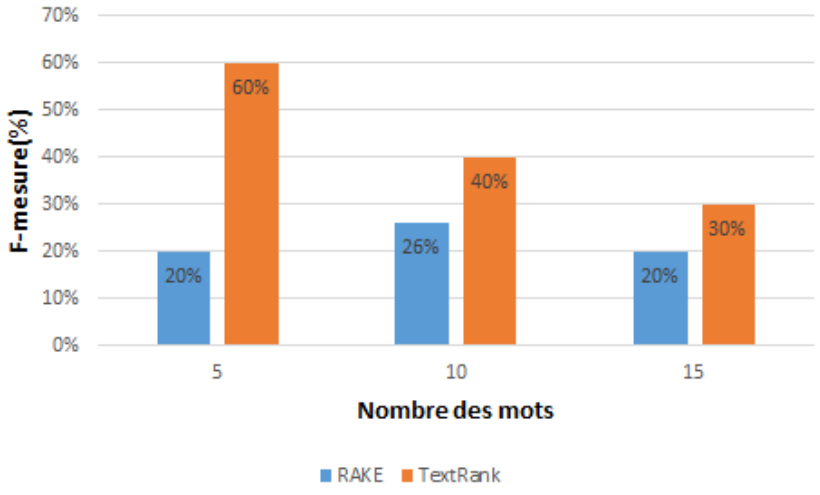

FIGURE 4.6 – Diagramme à barre de F-mesure

4.3.3 Conclusion de La deuxième étude

— les résultats obtenus par la méthode TextRank sont largement supérieurs à ceux de RAKE.

— Les deux méthodes sont efficaces au niveau du Top 10 et inefficaces aux autres niveaux.

4.4 Conclusion

— Dans les expériences précedentes on constate que les résultats obtenus par les deux méthodes sont meilleurs au niveau des Top 5 et 10 par rapport aux autres niveaux donc si on utilise les deux algorithmes, on prend en considération les dix premiers mots-clés et on néglige le reste car leur évaluation est faible.

— On ne peut pas conclure qu'une méthode est meilleure qu'une autre car la comparaison dépend de la mesure d'évaluation F-mesure,qui est influencée par les mots clés définis par l'auteur. La fréquence et le degré de ces derniers peuvent être faibles dans l'article ce qui affecte négativement sur la mesure d'évaluation F-mesure ,de plus, le type d'article affecte directement la F-mesure donc la comparaison en utilisant deux articles n'est pas fiable.

— Si on veut comparer les deux méthodes on doit effectuer des milliers d'expériences afin de déterminer la meilleure.

Chapitre 5

Conclusion

Dans ce mémoire on a comparé les résultats d'extraction des mots clés de deux méthodes RAKE et TextRank on a constaté que pour avoir des mots clés pertinents on optera pour les niveaux Top 5 et Top 10.
Parfois les mots clés extraits par les méthodes donne une idée précise du contenu des documents bien meilleure que celle donnée par les mots clés de l'auteur ainsi la méthode d'extration automatique et plus pérformante que l'extraction manuelle .
Pour confirmer cette étude il faut élargir cette experience sur plusieurs articles pour avoir des résultats plus fiables .

Bibliographie

[Abu-Zaid 2013] Ahmed Abu-Zaid et Khaled Alkattan. *Integration of scientific research training into undergraduate medical education : a reminder call.* Medical education online, vol. 18, no. 1, page 22832, 2013. (Cité en page 30.)

[Bal 2009] Bal Krishna Bal. *Towards building advanced natural language applications-an overview of the existing primary resources and applications in nepali.* In Proceedings of the 7th Workshop on Asian Language Resources (ALR7), pages 165–170, 2009. (Cité en page 24.)

[Bougouin 2013] Adrien Bougouin, Florian Boudin et Béatrice Daille. *Topicrank : Graph-based topic ranking for keyphrase extraction.* In International joint conference on natural language processing (IJCNLP), pages 543–551, 2013. (Cité en page 8.)

[Brownlee 2020] Jason Brownlee. Imbalanced classification with python : better metrics, balance skewed classes, cost-sensitive learning. Machine Learning Mastery, 2020. (Cité en page 13.)

[Campos 2018] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes et Adam Jatowt. *Yake ! collection-independent automatic keyword extractor.* In European Conference on Information Retrieval, pages 806–810. Springer, 2018. (Cité en page 10.)

[Chiaramella 2007] Yves Chiaramella et Philippe Mulhem. *La recherche d'information.* Document numérique, vol. 10, no. 1, pages 11–38, 2007. (Cité en page 1.)

[Daille 2017] Béatrice Daille, Sabine Barreaux, Adrien Bougouin, Florian Boudin, Damien Cram et Amir Hazem. *Indexation d'articles scientifiques Présentation et résultats du défi fouille de textes DEFT 2016.* Information Retrieval, Document and Semantic Web, vol. 17, no. 2, 2017. (Cité en page 4.)

[Grouin 2011] Cyril Grouin, Olivier Galibert, Sophie Rosset, Ludovic Quintard et Pierre Zweigenbaum. *Mesures d'évaluation pour entités nommées structurées.*

Évaluation des méthodes d'Extraction de Connaissances dans les Données, Brest, France, 2011. (Cité en page 13.)

[Kryvyi 2002] SL Kryvyi. *Compatibility of systems of linear constraints over the set of natural numbers.* Cybernetics and Systems Analysis, vol. 38, no. 1, pages 17–29, 2002. (Cité en pages 16 et 24.)

[Labiad 2017] Ali Labiad. *Sélection des mots clés basée sur la classification et l'extraction des règles d'association.* PhD thesis, Université du Québec à Trois-Rivières, 2017. (Cité en page 5.)

[Maldonado-Erazo 2021] Claudia Patricia Maldonado-Erazo, José Álvarez-García, María de la Cruz del Río-Rama, Amador Durán-Sánchez*et al. Scientific mapping on the impact of climate change on cultural and natural heritage : a systematic scientometric analysis.* Land, vol. 10, no. 1, page 76, 2021. (Cité en page 33.)

[Mihalcea 2004] Rada Mihalcea et Paul Tarau. *Textrank : Bringing order into text.* In Proceedings of the 2004 conference on empirical methods in natural language processing, pages 404–411, 2004. (Cité en page 5.)

[Mothe 2018] Josiane Mothe, Rajoelina Michel, Faneva Ramiandrisoa et Razakasoa Hary. *Intégration des plongements de mots dans les méthodes, supervisées et non supervisées, d'extraction automatique de mots clés.* 2018. (Cité en page 13.)

[Nakache 2005] Didier Nakache et Elisabeth Metais. *Evaluation : nouvelle approche avec juges.* In INFORSID, volume 5, pages 555–570, 2005. (Cité en page 14.)

[Ramiandrisoa] Mr Faneva Ramiandrisoa. *Analyse et extraction d'information sur le journal Information processing and Management.* (Cité en page 4.)

[Ramiandrisoa 2016] Faneva Ramiandrisoa et Josiane Mothe. *Extraction automatique de termes-clés : Comparaison de méthodes non supervisées.* 2016. (Cité en page 7.)

[Rose 2010] Stuart Rose, Dave Engel, Nick Cramer et Wendy Cowley. *Automatic keyword extraction from individual documents.* Text mining : applications and theory, vol. 1, pages 1–20, 2010. (Cité en page 12.)

[Sarica 2020] Serhad Sarica et Jianxi Luo. *Stopwords in technical language processing.* arXiv preprint arXiv :2006.02633, 2020. (Non cité.)

[Tixier 2016] Antoine Tixier, Fragkiskos Malliaros et Michalis Vazirgiannis. *A graph degeneracy-based approach to keyword extraction.* In Proceedings of the 2016 conference on empirical methods in natural language processing, pages 1860–1870, 2016. (Cité en page 10.)