République Algérienne Démocratique et Populaire
Ministère de l'Enseignement supérieur et de la Recherche Scientifique
Université ABBES LAGHROUR– KHENCHELA
Faculté des Science et de la Technologie
Département MI

Mémoire de Fin d'études

Pour l'obtention du diplôme Master (L.M.D)

Filière : Informatique

Option : STW

Thème :

Résumé Automatique d'un Texte Scientifique

Présenté par :

ZEROUAL OUSSAMA

Encadré par :

Mr. BOUSSALEM MOHAMED

Année universitaire 2021/2022

Remerciement

Avant tout je remercie Allah qui m'a donné le courage et la force pour continuer. C'est grâce à lui que mon chemin est éclairé pour finir ce modeste travail.

Je tiens à remercier mon encadreur de ce mémoire, Mr. Boussalem Mohamed, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

J'adresse mes plus sincères remerciements aux membres du jury

Je remercie tous les enseignants du département Math et informatique que je respecte beaucoup.

Enfin, je souhaiterais adresser des remerciements plus particuliers à toute ma famille.

Dédicaces

Je dédie ce travail à mes parents, mes sœurs et mes frères, leur amour et leur soutien inconditionnels ont rendu ce travail possible.

Zeroual Oussama

بسم الله الرحمن الرحيم

Résumé

Le processus de recherche scientifique commence généralement par un examen de l'état de l'art, qui peut impliquer de nombreuses publications. Le résumé automatique des articles scientifiques aiderait les chercheurs dans leur investigation en accélérant le processus de recherche.

Notre système proposé est : un système de résumer automatique d'un texte scientifique, il se base sur l'approche extractive, dans ce système, nous avons construit deux modules : un module de préparation, il s'occupe de l'extraction des sections (de l'article) nécessaires pour le résumé et prétraiter ces sections. Le deuxième module est : la génération de résumé, ce module sélectionne les phrases les plus pertinentes, après avoir calculé le score de chaque phrase, pour construire le résumé. Pour l'évaluation de notre système, nous l'avons comparé avec deux algorithmes de résumé automatique à savoir, LSA et TextRank en utilisant les métriques ROUGE avec le corpus cmp-lg (*Computation and Language*) (fournit par SUMMAC) composé de 183 articles scientifiques sous forme xml. Les résultats montrent que notre méthode a surpassé certains autres systèmes, et c'est un résultat satisfaisant.

Mot clés : Information pertinente, Résumé automatique de texte, Article scientifique, Evaluation de système de résumé automatique de texte, ROUGE.

Abstract

The scientific research process generally begins with a review of the state of the art, which may involve numerous publications. The automatic summary of scientific articles would help researchers in their investigation by speeding up the research process.

Our proposed system is: a system for automatically summarizing a scientific text, it is based on the extractive approach, in this system, we have built two modules: a preparation module, it deals with the extraction of sections (of the article) needed for the abstract and pre-process these sections. The second module is: summary generation, this module selects the most relevant sentences, after calculating the score of each sentence, to build the summary. For the evaluation of our system, we compared it with two automatic text summarization algorithms: LSA and TextRank using ROUGE metrics with cmp-lg (*Computation and Language*) corpus (offered by SUMMAC) composed of 183 scientific articles in xml format. The results show that our method has outperformed some other systems, and this is a satisfactory result.

Keywords: Relevant information, Automatic text summarization, scientific article, Evaluation of automatic text summarization system, ROUGE.

Table des Matières

Liste des Figures

Liste des Tableaux

Liste des Abbreviations

TALN	Traitement Automatique du Langage Naturel
IA	Intelligence Artificielle
NLTK	Natural Language Toolkit
TF-IDF	Term Frequency-Inverse Document Frequency
VSM	Vector Space Model
LSA	Latent Semantic Analysis
RI	Recherche d'Information
SCU	Semantic Content Unit
ROUGE	Recall Oriented Understudy for Gisting Evaluation
ROUGE-N	N-gram Co-Occurrence Statistics
ROUGE-L	Longest Common Subséquence
ROUGE-W	Weighted Longest Common Subsequence
ROUGE-S	Skip-Bigram Co-Occurrence Statistics
LCS	Longest Common Subsequence
SUMMAC	Summarization Conference
NIST	National Institute of Standards and Technology
TSC	Text Summarization Challenge
DUC	Document Understanding Campagnes de conférence
TAC	Text Analysis Conference
cmp-lg	Computation and Language

Introduction Générale

L’évolution des nouvelles technologies de l'information et de la communication, l'avènement d'Internet et la multiplication des moteurs de recherche permettent de disposer de quantités énormes d'information sous différentes formes (visuelle, audio, multimédia, textuelle). La compression de ces informations ou leur résumé s’avèrent être des moyens nécessaires sinon indispensables pour ne garder que l’information pertinente pouvant servir de lien pour la recherche de sa globalité, au besoin.

L'information textuelle transmet beaucoup de connaissances courantes sous forme de documents. Les résumés de ces documents ont constitué depuis le début de l’informatique des domaines de recherches et d’investigations investis de façon intensive par les chercheurs. Le résumé d'un texte consiste en la production d'un deuxième texte composé des idées essentielles et clairement plus court et plus concis.

Contribution :

La diffusion des recherches originales par la publication d'articles scientifiques est essentielle pour le développement des connaissances, l’amélioration des pratiques et l’émergence de débats. La lecture de plusieurs articles scientifiques est une tâche très couteuse en matière de temps et d’effort, surtout avec la grande masse de documents sur l’internet. L’objectif de ce travail est d’appliquer les techniques de résumer automatique de texte sur un texte (document) scientifique afin de produire un texte qui permet aux chercheurs d’avoir une idée claire sur la contribution de cet article sans le lire entièrement.

Plan du mémoire :

Notre travail sera organisé en quatre chapitres comporte une introduction et une conclusion générales :

Le premier chapitre est intitulé généralités sur le Traitement Automatique de Langage Naturel, où certaines notions de base du traitement automatique de langage naturel sont abordées.

Résumé Automatique de Texte est l'intitulé du deuxième chapitre où un bref état de l'art sur le résumé automatique de texte est présenté, à la fin de ce chapitre les méthodes de résumer d'un article scientifique et ses structures sont présentés.

Dans le troisième chapitre : Evaluation de résumé automatique de texte, nous avons conclu que pour connaître la qualité du résumé, il faut l'évaluer. Cette évaluation peut être manuelle par un humain ou automatiquement par une machine. Nous savons également qu'il existe deux approches de l'évaluation, à savoir : l'évaluation intrinsèque et l'évaluation extrinsèque.

Le dernier chapitre : conception et réalisation du système, ce chapitre présente l'implémentation de l'application, nous avons expliqué les différentes étapes de l'implémentation et de l'évaluation de notre système ainsi que les résultats obtenus.

Chapitre 1 : Généralités sur le Traitement Automatique de Langage Naturel

1. Introduction

Le traitement automatique du langage naturel (Natural Language Processing en anglais) est un domaine de recherche et d'application qui explore les ordinateurs de démonstration qui peuvent être utilisés pour comprendre et manipuler du texte ou de la parole en langage naturel pour faire des choses utiles.

Les chercheurs en TALN visent à rassembler des connaissances sur la façon dont les êtres humains comprennent et utilisent le langage afin que des outils et des techniques appropriés puissent être développés pour permettre aux systèmes informatiques de comprendre et de manipuler les langages naturels pour effectuer les tâches souhaitées. Les applications de le TALN comprennent un certain nombre de domaines d'études, tels que la traduction automatique, le traitement et le résumé de texte en langage naturel, l'analyse de sentiment, les interfaces utilisateur et le multilingue, la reconnaissance vocale et les systèmes experts.

2. Traitement automatiquedu langage naturel dans l'intelligence artificielle

L'intelligence artificielle est un vaste domaine de l'informatique qui s'intéresse au comportement « intelligent » des programmes informatiques. Le traitement du langage naturel fait partie intégrante d'applications d'IA telles que la compréhension du langage naturel, y compris la réponse aux questions et la reconnaissance vocale, la traduction automatique, l'apprentissage automatique et les interfaces intelligentes des systèmes de gestion de bases de données.[1]

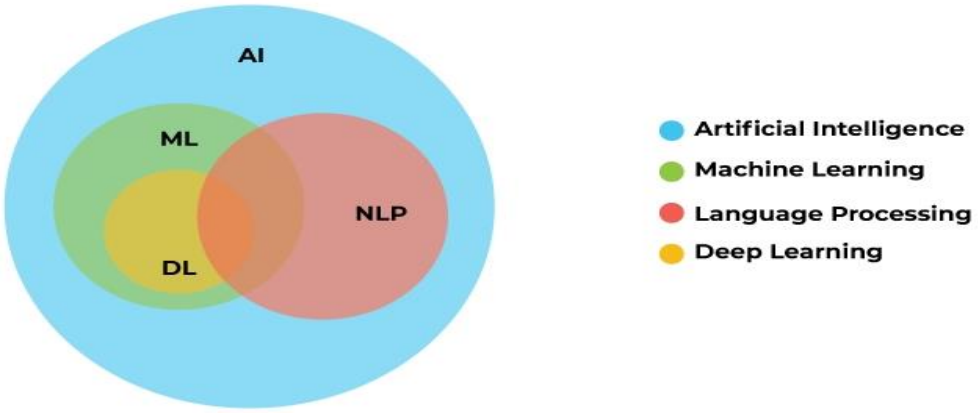


Figure 1.1 : TALN dans l'intelligence artificielle.[2]

3. Les Applications de TALN

Nous mentionnons plusieurs fois le terme traitement automatique du langage naturel (TALN) dans ce chapitre.

Le TALN est définie comme un domaine spécialisé de l'informatique et de l'ingénierie et de l'intelligence artificielle avec des racines dans la linguistique informatique. Il s'intéresse principalement à la conception et à la construction d'applications et de systèmes permettant l'interaction entre les machines et les langages naturels créés par les humains. Cela rend également le TALN liée au domaine de l'interaction homme-machine. Les techniques de TALN permettent aux ordinateurs de traiter et de comprendre le langage naturel humain et de l'utiliser davantage pour fournir une sortie utile. Nous parlons de certaines des principales applications de le TALN.[3]

3.1 Traduction automatique :

La traduction automatique est peut-être l'une des applications les plus recherchées de la TALN. Il est défini comme la technique qui aide à fournir des traductions syntaxiques, grammaticales et sémantiquement correctes entre deux paires de langues. C'était peut-être le premier grand domaine de recherche et de développement en TALN.

À un niveau simple, la traduction automatique est la traduction du langage naturel effectuée par une machine. Par défaut, les blocs de construction de base du processus de traduction automatique impliquent une simple substitution de mots d'une langue à une autre, mais dans ce cas, nous ignorons des éléments tels que la grammaire et la cohérence de la structure des phrases. Par conséquent, des techniques plus sophistiquées ont évolué au fil du temps, notamment en combinant de grandes ressources de corpus de textes avec des techniques statistiques et linguistiques. L'un des systèmes de traduction automatique les plus populaires est Google Traduction.

3.2 Systèmes de reconnaissance vocale :

C'est peut-être l'application la plus difficile pour le TALN. Le test de Turing l'un est des tests principaux et peut-être les plus difficiles de l'intelligence réelle dans les systèmes d'intelligence artificielle.

Ce test indique que si une question est donnée par l'utilisateur à l'ordinateur et à un humain, il serait incapable de distinguer les réponses obtenues. Au fil du temps, de nombreux progrès ont été réalisés dans ce domaine en utilisant des techniques telles que la synthèse vocale, l'analyse syntaxique et le raisonnement contextuel. Cependant, l'une des principales limites des systèmes de reconnaissance vocale reste qu'ils sont très spécifiques à un domaine et ne fonctionneront pas si l'utilisateur s'écarte même un peu des entrées de script attendues nécessaires au système. Les systèmes de reconnaissance vocale se trouvent désormais dans une grande variété d'endroits, des ordinateurs aux téléphones mobiles, en passant par les systèmes d'assistance virtuelle comme l'assistant de Google.

3.3 Résumé du texte :

L'objectif principal du résumé de texte est de prendre un ensemble de documents texte, qui peut être un ensemble de textes ou de paragraphes et de réduire le contenu de manière appropriée pour créer un résumé qui conserve les points les plus pertinents de l'ensemble de documents.

Le résumé du texte peut être effectué en examinant les divers documents et en essayant de trouver les mots-clés, les expressions et les phrases qui ont une importance dans la collection. Deux principaux types de techniques de résumé de texte comprennent le résumé basé sur l'extraction et le résumé basé sur l'abstraction. Avec l'avènement d'énormes quantités de texte et de données non structurées, le besoin de résumé de texte pour obtenir rapidement des informations précieuses est en forte demande.

3.4 Catégorisation du texte :

L'objectif principal de la catégorisation de texte est d'identifier dans quelle catégorie ou classe un document spécifique doit être placé en fonction du contenu du document. C'est l'une des applications les plus populaires de le TALN et de l'apprentissage automatique, car avec les bonnes données, il est extrêmement simple de comprendre les principes qui sous-tendent ses composants internes et de mettre en œuvre un système de catégorisation de texte fonctionnel. Des techniques d'apprentissage automatique supervisées et non supervisées peuvent être utilisées pour résoudre ce problème et parfois une combinaison des deux est utilisée. Cela a aidé à créer de nombreuses applications réussies et pratiques, notamment des filtres anti-spam et des catégorisations d'articles.

3.5 Analyse de texte :

L'analyse de texte, également connue sous le nom d'exploration de texte, est définie comme la méthodologie et le processus suivis pour obtenir des informations de qualité et exploitables à partir de données textuelles. Cela implique l'utilisation de techniques de traitement du langage naturel, de récupération d'informations et d'apprentissage automatique pour analyser les données textuelles non structurées dans des formes plus structurées et en déduire des modèles et des informations à partir de ces données qui seraient utiles à l'utilisateur final. L'analyse de texte comprend un ensemble de techniques d'apprentissage automatique, linguistiques et statistiques qui sont utilisées pour modéliser et extraire des informations du texte principalement pour les besoins d'analyse, y compris l'informatique décisionnelle, l'analyse exploratoire, descriptive et prédictive. Certaines des principales techniques et opérations d'analyse de texte sont mentionnées ci-dessous.

- Classement de texte
- Regroupement de texte
- Analyse des sentiments
- Extraction et reconnaissance d'entités
- Analyse de similarité et modélisation des relations

4. Les niveaux d'analyse linguistique

Actuellement, le TALN est définie comme une gamme théoriquement motivée de techniques de calcul pour analyser et représenter des textes naturels à un ou plusieurs niveaux d'analyse linguistique dans le but de réaliser un traitement du langage de type humain pour une gamme de tâches ou d'applications. Concrètement, il s'agit d'un domaine d'étude axé sur la compréhension du sens intégral d'un texte, qu'il soit écrit ou parlé, qui intègre des concepts et des méthodes tirés de divers domaines, notamment l'informatique, la linguistique, la psychologie, la théorie de l'information, les mathématiques et les statistiques.

Un système TALN est généralement composé de plusieurs composants, mieux illustrés dans le contexte des différents niveaux d'analyse du langage, décrits ci-dessous.

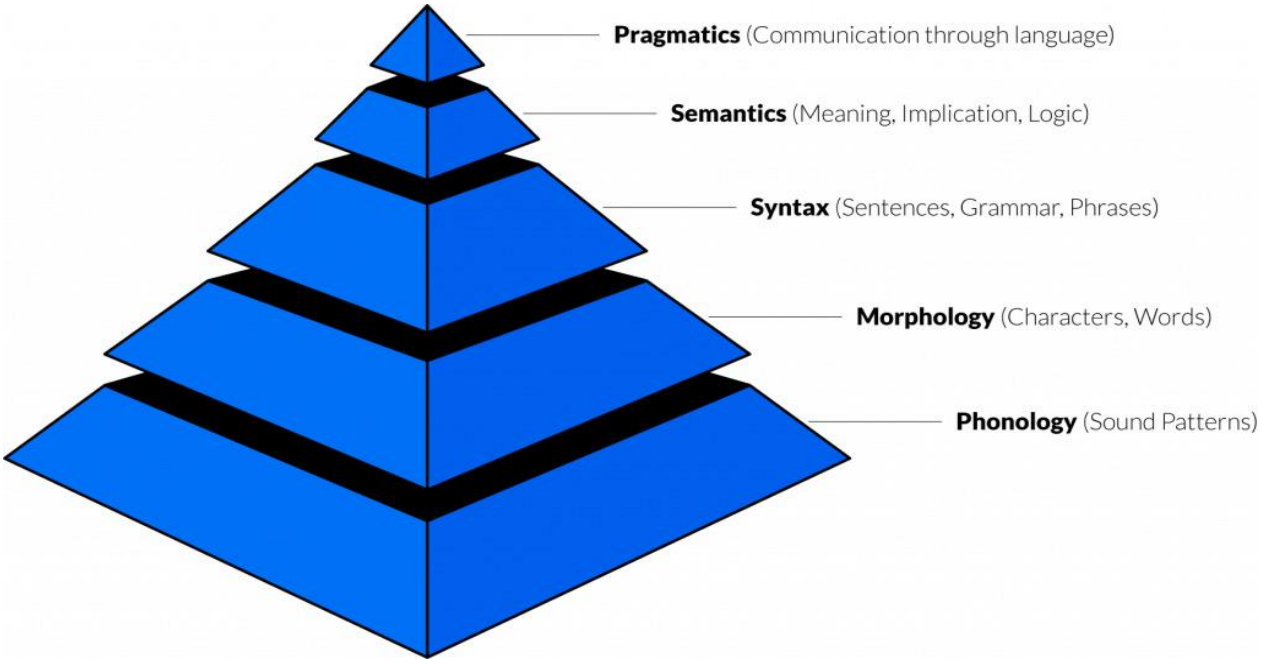


Figure 1.2 : différents niveaux d'analyse du langage**.**[4]

4.1 Phonétique et phonologique :

Les connaissances phonétiques et phonologiques font référence à la compréhension des mots basée sur les sons de la parole (tels que les phonèmes et les affichages vocaux) qu'ils produisent. Le traitement automatique des langages naturel (TALN) diffère par la reconnaissance de la parole, couramment utilisée dans la transcription des rapports de radiologie, et qui identifie les mots à partir d'un signal audio. Comprendre le sens de ces mots nécessite l'utilisation d'un système de TALN parlé.[5]

4.2 Morphologique :

Ce niveau traite de la nature composante des mots, qui sont composés de morphèmes - les plus petites unités de sens. Par exemple, le mot international peut être analysé morphologiquement en trois morphèmes distincts : le préfixe inter, la racine nation et le suffixe al. Étant donné que la signification de chaque morphème reste la même d'un mot à l'autre, les humains peuvent décomposer un mot inconnu en ses morphèmes constitutifs afin de comprendre sa signification. De même, un système TALN peut reconnaître le sens véhiculé par chaque morphème afin d'acquérir et de représenter le sens.[5]

4.3 Syntaxique :

Ce niveau se concentre sur l'analyse des mots dans une phrase afin de découvrir la structure grammaticale de la phrase. Cela nécessite à la fois une grammaire et un analyseur. La sortie de ce niveau de traitement est une représentation de la phrase qui révèle les relations de dépendance structurelle entre les mots. Il existe différentes grammaires qui peuvent être utilisées et qui, à leur tour, auront un impact sur le choix d'un analyseur. Toutes les applications TALN ne nécessitent pas une analyse complète des phrases, par conséquent, les défis restants dans l'analyse de l'attachement de la phrase prépositionnelle et de la portée de la conjonction ne bloquent plus les applications pour lesquelles les dépendances de phrase et de clause sont suffisantes. La syntaxe transmet le sens dans la plupart des langues parce que l'ordre et la dépendance contribuent au sens. Par exemple, les deux phrases : « Le chien a chassé le chat. » et « Le chat a chassé le chien. » ne diffèrent que par la syntaxe, mais véhiculent des significations assez différentes.[5]

4.4 Sémantique :

Le traitement sémantique détermine les significations possibles d'une phrase en se concentrant sur les interactions entre les significations au niveau des mots dans la phrase. Ce niveau de traitement peut inclure la désambiguïsation sémantique des mots aux sens multiples d'une manière analogue à la façon dont la désambiguïsation syntaxique des mots qui peuvent fonctionner comme plusieurs parties du discours est accomplie au niveau syntaxique. La désambiguïsation sémantique permet de sélectionner et d'inclure un et un seul sens de mots polysémiques dans la représentation sémantique de la phrase.[5]

4.5 Pragmatique :

Ce niveau concerne l'utilisation délibérée du langage dans des situations et utilise le contexte au-delà du contenu du texte pour comprendre le but est d'expliquer comment une signification supplémentaire est lue dans les textes sans y être encodée. Cela nécessite une grande connaissance du monde, y compris la compréhension des intentions, des plans et des objectifs. Certaines applications TALN peuvent utiliser des bases de connaissances et des modules d'inférence.[5]

5. Prétraitement du texte

Le prétraitement du texte est une méthode pour nettoyer les données textuelles et les préparer à alimenter le modèle en données. Les données textuelles contiennent du bruit sous diverses formes comme les mots vides, les ponctuations, les espaces supplémentaires. Lorsque nous parlons de langage humain, il existe différentes façons de dire la même chose, et ce n'est que le principal problème auquel nous devons faire face car les machines ne comprendront pas les mots, elles ont besoin de chiffres, nous devons donc convertir le texte en nombres dans un manière efficace.[6]

5.1 Minuscule :

Si le texte est dans la même casse, il est facile pour une machine d'interpréter les mots car les minuscules et les majuscules sont traitées différemment par la machine. Par exemple, des mots comme Boule et boule sont traités différemment par la machine. Donc, nous devons faire le texte dans le même cas et le cas le plus préféré est une minuscule pour éviter de tels problèmes.

5.2 Supprimer les ponctuations :

L'une des autres techniques de traitement de texte consiste à supprimer les ponctuations. il y a au total 32 ponctuations principales qui doivent être prises en compte. Nous pouvons directement utiliser le module string avec une expression régulière pour remplacer toute ponctuation dans le texte par une chaîne vide.

5.3 Segmentation :

C'est le processus de division du texte en phrases où apparaissent des points et des points d'interrogation. Cependant, il peut y avoir des abréviations, des formes de titre (Dr., Mr., etc.), ou des points de suspension (...) qui peuvent gâcher la violation de la règle simple.

5.4 Tokenisation des mots :

Semblable à la segmentation des phrases, pour segmenter une phrase en mots, nous pouvons commencer par une règle simple pour diviser le texte en mots en fonction de la présence de signes de ponctuation. La bibliothèque NLTK nous permet de le faire.

5.5 Supprimer les mots vides :

Les mots vides sont les mots les plus courants dans un texte qui ne fournissent aucune information valable. Les mots vides comme ils, le, la, ceci, ou, etc. sont quelques-uns des mots vides. La bibliothèque NLTK est une bibliothèque commune utilisée pour supprimer les mots vides. Si nous voulons ajouter un nouveau mot vide à un ensemble de mots, il est facile d'utiliser la méthode d'addition.

5.6 Normalisation (Stemming et lemmatisation) :

La radicalisation est un processus pour réduire le mot à sa racine, par exemple Définir, Définition, Définitions, Définissons dériver du même mot que Définir. Fondamentalement, la racine do consiste à supprimer le préfixe ou le suffixe d'un mot comme, s, es, etc. La bibliothèque NLTK est utilisée pour radicaliser les mots. La technique de radicalisation n'est pas utilisée à des fins de production car ce n'est pas une technique aussi efficace et la plupart du temps, elle supprime les mots indésirables. Ainsi, pour résoudre le problème, une autre technique est apparue sur le marché sous le nom de lemmatisation. il existe différents types d'algorithmes de stemming comme porter stemmer, snowball stemmer. Porter stemmer est largement utilisé dans la bibliothèque NLTK.

5.7 Supprimer les espaces supplémentaires :

La plupart du temps, les données textuelles contiennent des espaces supplémentaires ou lors de l'exécution des techniques de prétraitement ci-dessus, il reste plus d'un espace entre le texte, nous devons donc contrôler ce problème. La bibliothèque d'expressions régulières fonctionne bien pour résoudre ce problème.

6. Représentations textuelles

Comment les ordinateurs comprennent-ils et interprètent-ils le langage ?

Les ordinateurs sont brillants lorsqu'il s'agit de chiffres. Ils sont plus rapides que les humains dans les calculs et les schémas de décodage de plusieurs ordres de grandeur. Mais que se passe-t-il si les données ne sont pas numériques ? Et si c'était la langue ? Que se passe-t-il lorsque les données sont en caractères, mots et phrases ? Comment faire en sorte que les ordinateurs traitent notre langage ? Comment Alexa, Google Home et de nombreux autres assistants intelligents comprennent-ils et répondent-ils à notre discours ?

Le traitement du langage naturel est un sous-domaine de l'intelligence artificielle qui permet aux machines de comprendre et de traiter le langage humain. L'étape la plus élémentaire pour la majorité des tâches de traitement du langage naturel (TALN) consiste à convertir les mots en nombres pour que les machines comprennent et décodent les modèles dans une langue. Nous appelons cette étape la représentation textuelle. Cette étape, bien qu'itérative, joue un rôle important dans le choix des fonctionnalités de votre modèle/algorithme d'apprentissage automatique. Parmi les modèles de représentation textuelle, nous mentionnons les suivants.[3]

- **One-Hot Encoding :**

C'est un type de représentation qui attribue 0 à tous les éléments d'un vecteur sauf un, qui a la valeur 1. Cette valeur représente une catégorie d'un élément.

Par exemple :

Si j'avais une phrase, "I love my cat", chaque mot de la phrase serait représenté comme ci-dessous :

I → [1 0 0 0] love → [0 1 0 0] my → [0 0 1 0] cat →[0 0 0 1]

Il présente de nombreux inconvénients, notamment une consommation de mémoire et de temps.

- **VSM (Vector Space Model)**

Pour que les algorithmes ML (Machin Learning) fonctionnent avec des données textuelles, les données textuelles doivent être converties en une forme mathématique. Nous représenterons les unités de texte (caractères, phonèmes, mots, expressions, phrases, paragraphes et documents) avec des vecteurs de nombres. C'est ce qu'on appelle le modèle d'espace vectoriel (VSM). C'est un modèle algébrique simple largement utilisé pour représenter n'importe quel texte.

- **Bag of Words:**

Représentation par sac de mots (Bag of Words), comme son nom l'indique intuitivement, place les mots dans un "sac" et calcule la fréquence d'occurrence de chaque mot. Il ne prend pas en compte l'ordre des mots ou les informations lexicales pour la représentation du texte.

Par exemple:

- La Phrase (1): John likes to watch movies. Mary likes movies too.

- La Phrase (2): Mary also likes to watch football games.

Donc le sac de mots de chaque phrase est :

$BoW1 = \{"John": 1, "likes": 2, "to": 1, "watch": 1, "movies": 2, "Mary": 1, "too": 1\}$

$BoW2 = \{"Mary": 1, "also": 1, "likes": 1, "to": 1, "watch": 1, "football": 1, "games": 1\}$

- **TF-IDF**

Pour supprimer les mots à très haute fréquence et ignorer les mots à basse fréquence, il est nécessaire de normaliser les "poids" des mots en conséquence.

La représentation TF-IDF : la forme complète de TF-IDF est : Fréquence du terme-Fréquence inverse de document est un produit de 2 facteurs :

$$TF - IDF = TF(w, d) * IDF(w)$$

- TF (w, d) est la fréquence du mot ‘w’ dans le document ‘d’.
- IDF(w) peut être décomposé en :

$$IDF = \log\left(\frac{N}{df(w)}\right)$$

- N est le nombre total de documents.
- df(w) est la fréquence des documents contenant le mot ‘w’.

7. Conclusion :

Dans ce chapitre, nous avons donné un aperçu de le TALN, une discipline jeune, mais qui a atteint un niveau de maturité appliquée important, surtout au cours de cette décennie. Le succès récent de le TALN, en particulier dans les domaines appliqués comme nous l'avons mentionné dans ce chapitre, est dû à la fois aux avancées technologiques et aux conditions historiques actuelles.

Dans le chapitre suivant, nous commencerons notre spécialisation dans l'étude, le résumé automatique de texte, l'un des domaines du traitement automatique du langage naturel.

Chapitre 2 : Résumé Automatique de Texte

1. Introduction

Le résumé automatique de texte est considéré comme l'une des applications les plus importantes du traitement du langage naturel. Il fonctionne pour produire un bref résumé qui contient des informations suffisantes et importantes à partir d'un long texte. Il vise à aider les humains pour traiter des textes volumineux en peu de temps. Nous lisons habituellement des articles et des documents et nous rédigeons nos résumés manuellement, et cela prend beaucoup de temps, alors nous devons passer au résumé automatique.

Il existe des applications importantes pour le résumé de texte dans diverses tâches liées à le TALN telles que la classification de texte, la réponse aux questions, le résumé de textes juridiques, le résumé de nouvelles et la génération de titres. De plus, la génération de résumés peut être intégrée à ces systèmes comme une étape intermédiaire qui permet de réduire la longueur du document.

2. Définition (C'est quoi un résumé du texte ?)

Un résumé est un texte produit à partir d'un ou plusieurs textes, qui transmet les informations importantes dans les textes originaux, Le résumé ne doit pas représenter plus de la moitié du texte original et généralement beaucoup moins que cela (jusqu'à un tiers ou un quart du texte original texte).[7]

3. Classification des systèmes de résumé automatique

Les auteurs ont proposé différentes classifications pour classer les systèmes de résumé automatique. Chaque auteur propose sa propre classification. Selon Jones [8] les systèmes de résumé automatique sont classés selon trois critères :

- La source : le document d'entrée
- L’objectif : le but de résumé
- La sortie : le document de sortie

Les systèmes de résumé automatique peuvent être classés selon les critères suivants (**voir la figure 2.1**).

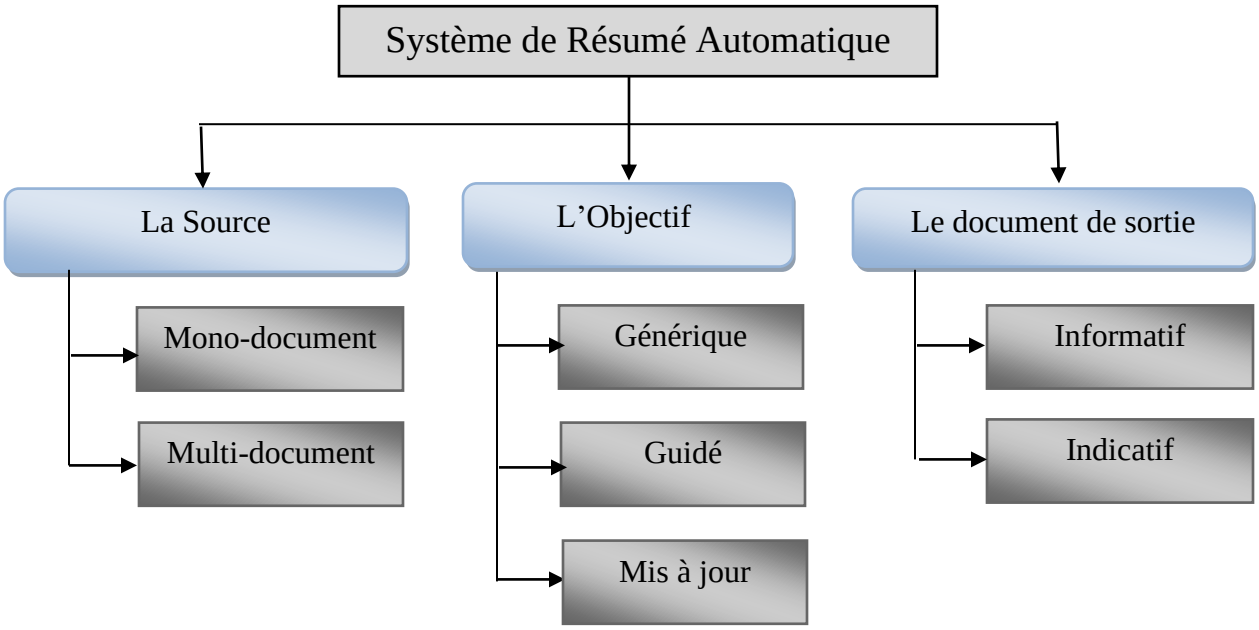


Figure 2.1 : Taxonomie des Systèmes de Résumé Automatique

3.1 La Source :

L'entrée dans le système de résumé automatique peut être un document unique ou un groupe de documents. Ainsi, le système de résumé automatique peut être mono-document ou multi-document.

a) Le Résumé Mono document :

C'est le processus de représentation d'un document unique dans un texte court en capturant les informations pertinentes et en filtrant les informations redondantes.[9]

b) Le Résumé Multi document :

Est un processus de représentation d'un ensemble de documents avec un court morceau de texte en capturant les informations pertinentes et en filtrant les informations redondantes. Deux approches importantes du résumé multi-documents sont le résumé extractif et le résumé abstrait.[9]

3.2 L'Objectif :

Selon la cible visée, le système de résumé automatique peut être soit générique, soit orienté et soit mis à jour.

a) Résumé Générique :

Consiste à produire des résumés en conservant les sujets principaux et sans tenir compte des besoins d'information du lecteur. Cette tâche, qui semble simple, mais présente plusieurs difficultés. Parmi eux se trouve le type de document que nous devons résumer.[10]

b) Résumé Guidé :

Comprend la génération de résumés qui répondent aux besoins d'information de l'utilisateur. Ces besoins sont généralement exprimés par le biais de requêtes et doivent permettre au système d'isoler les parties du document qui traitent de sujets spécifiques. L'objectif est de générer un résumé de document qui ne contient que des unités directement liées à l'objet de la demande.[11]

c) Résumé de la Mise à jour :

Dans ce type de système, le lecteur est supposé avoir lu le document et son résumé sur un sujet particulier. Le résumé mis à jour montrer seulement les nouveautés importantes, tout en évitant la redondance des informations des documents que l'utilisateur a déjà lus.[11]

3.3 Le document de sortie :

Selon le type de sortie, il existe deux types de système : indicatif ou informatif.

a) Le résumé informatif :

Il contient une partie informative du texte original. Après l'avoir lu, vous pouvez apprendre les idées principales du document original. Il donne un aperçu du contenu du texte, tout en conservant l'ordre des sujets. Les résumés informatifs peuvent être trouvés dans les articles de recherche où l'auteur tente de présenter l'essentiel de sa recherche. En affichant tous les grands thèmes.[7]

b) Le résumé indicatif :

Les résumés indicatifs ne décrivent que les métadonnées du document, qui comprend des éléments de recherche clés tels que l'objectif, la portée et la méthodologie de recherche. Il fournit au lecteur suffisamment d'informations pour décider de revenir ou non au texte original. Les résumés indicatifs décrivent simplement le type de recherche ou de rédaction du document et ne contiennent aucun contenu physique du document tel que des conclusions.[7]

4. Les approches de résumé automatique

Il existe deux approches fondamentales de résumé de texte : extractive et abstractive. Le premier extrait des mots et des phrases à travers le texte original pour créer un résumé. La deuxième réécrit le sujet principal du document dans différentes phrases pour générer des résumés similaires aux résumés humains. [12]

4.1 Approche de résumé par abstraction

Lorsqu'un humain reçoit un corpus de texte à résumer, il peut réécrire les points principaux dans ses propres mots. C'est ce qu'on appelle le résumé abstrait et cela nécessite des compétences humaines de haut niveau comme la capacité de combiner plusieurs perspectives dans un langage naturel cohérent. L'état de l'art du résumé abstraite n'est pas encore à la hauteur, c'est pourquoi de nombreux systèmes de résumé automatique choisir une technique appelée résumé extractive.[10]

4.2 Approche de résumé par extraction

Les résumés extractifs sont des extraits tirés directement des documents d'entrée et présentés de manière lisible. Le résumé ne contient aucune reformulation des idées présentées dans le texte original. Les méthodes de résumés extractives utilisent des techniques basées sur l'intelligence artificielle en général et le TALN en particulier pour identifier les déclarations les plus importantes directement à partir de la source.[10]

4.3 La différence entre le résumé par extraction et par abstraction

Le résumé par abstraction peut diffère avec le résumé par extraction, en ce sens qu'il cherche à créer de nouvelles phrases à partir de zéro tout en préservant l'essence, plutôt que de les extraire du document source. Le développement des méthodes abstraites est généralement plus difficile, car elles nécessitent des techniques performantes de génération de langage naturel, et sont elles-mêmes un domaine de recherche actif.

5. Les Méthodes utilisées dans le résumé automatique par extraction

Dans cette section de chapitre, nous présentons brièvement différentes méthodes d'extraction de phrases clés, Ces méthodes reposent principalement sur le calcul du score attribué à chaque phrase pour estimer son importance dans le texte. Le résumé final ne conserve que les phrases avec les scores les plus élevés.[13]

5.1 Méthodes à base de mots clés :

Cette méthode est basée sur le fait que l'auteur se sert (pour exprimer ses idées principales) de quelques mots-clés qui ont tendance à être récurrents dans le texte. Le résumé automatique est alors produit en recherchant dans le texte source les unités de texte minimales réunissant ses mots-clés. Ce principe est souvent utilisé dans diverses variantes présentées dans les sous-sections qui suivent.[13]

5.1.1 Mots-clés prédéfinis :

Pour calculer le score de chaque phrase S selon les mots-clés qu'elle contient, on peut calculer le score suivant :

$$Score_{mot-clé}(S) = \sum_{w \in S} a(w) * F(w)$$

$$a(w) = \begin{cases} A \; si \; w \in Liste \; des \; mots - clés (A > 1) \\ 1 \; sinon \end{cases}$$

F(w) est la fréquence du terme w dans la phrase S

La liste de mots-clés peut être introduite par l'utilisateur (domaine d'intérêt) ou composée des mots-clés établis par l'auteur. L'importance du poids du terme w est donné par A×F(w), avec A>1.

5.1.2 Les mots de titre :

Étant donné que le titre est l'expression la plus significative et qui résume le mieux un document en quelques mots, on peut dire que la phrase qui ressemble le plus au titre est la plus marquante du document. Par conséquent, chaque phrase peut être pondérée selon sa similarité au titre. Dans ce cas on considère les mots du titre du texte comme des mots-clés et on produit le résumé en sélectionnant les phrases qui couvrent certains mots apparaissant dans un titre.

$$Score_{mot-clé}(S) = \sum_{w \in S} b(w) * F(w)$$

$$b(w) = \begin{cases} A \; si \; w \in Liste \; des \; mots \; tits (A > 1) \\ 1 \; sinon \end{cases}$$

F(w) est la fréquence du terme w dans la phrase S

5.2 Méthode à base de position :

Cette méthode part du principe que la position d'une phrase dans un texte indique à quel point elle est importante dans le contexte. Les premières phrases du paragraphe, par exemple, peuvent transmettre l'idée principale et devraient donc être incluses dans le résumé. Comme variante de cette méthode on peut citer la méthode Lead : c'est une méthode qui détermine les phrases importantes en extrayant celles qui sont en tête.

Cette méthode est efficace pour résumer les articles de journaux, puisque les phrases importantes ont tendance à apparaître dans les premières phrases de l'article. On définit le score d'une phrase S à la position i comme suit :

$$Score_{\text{Lead}}(Si) = \beta i$$

$$\beta i = \begin{cases} B > 0 \; si \;\; i < N \\ 0 \; si \;\; i \geq N \end{cases}$$

βi Est une fonction rectangulaire qui modélise la distribution de phrases importantes selon leur position dans l'article.

Dans le cas où les dernières phrases auraient une certaine importance, il suffit d'introduire un nouvel intervalle pour la valeur de i.

L'inconvénient de cette méthode est qu′elle dépend de la nature du texte à résumer ainsi que du style de l'auteur. [13]

5.3 Méthode dépendant de la longueur de phrase :

Cette méthode donne un poids à une phrase selon le nombre de mots dans la phrase.[13] Deux techniques peuvent être employées pour le calcul du score :

- longueur de chaque phrase (Li) par rapport à la longueur maximale de la phrase.

$$Score_{long}(Si) = Li/Lmax$$

- affecte un score nul à une phrase plus courte qu'une certaine longueur (L minimale).

$$Score_{long}(Si) = \begin{cases} 0 \; si \; Li \leq Lmin \\ \dfrac{Li - Lmin}{Lmax} \; si \; Li > Lmin \end{cases}$$

5.4 Méthode à base d'expressions indicatives (cue-phrases) :

Cette méthode choisit des unités de texte avec des indications spécifiques ou des expressions spécifiques. Par exemple, pour les textes scientifiques, on a comme expressions le but de ce travail …, ce papier présente …, les résultats et des conclusions sont de bons candidats pour indiquer les phrases à inclure dans un résumé. Des textes de types différents peuvent avoir des expressions indicatives différentes.[13]

On peut déduire un score pour une phrase d'un texte quelconque à analyser en fonction de la ressemblance qu'elle présente, pour le trait donné.

On pourrait définir le score d'une phrase S correspondant à un certain motif comme :

$$Score_{cue}(S) = \begin{cases} 1 \; si \; S \; correspond \text{ à } un \; motif \\ 0 \; sinon \end{cases}$$

5.5 Méthode hybride :

Les méthodes présentées dans les sections précédentes utilisent des traits (fréquence, position, expression indicative, etc.) qui ne peuvent isolément garantir des résultats optimaux. On combine souvent ces traits par exemple avec l'équation suivante.

$$Score_{hbride}(S) = a_1 * Score_{tf-idf} + a_2 * Score_{lead} + a_3 * Score_{cue} + a_4 * Score_{titre}$$

Les poids a peuvent être fixés arbitrairement ou déterminés de manière expérimentale (par apprentissage par exemple).

Certaines expériences sur un corpus hétérogène de 200 documents ont montré que si on combine les méthodes cue, titre et position (poids zéro pour la méthode mot-clés), on obtient de meilleurs résultats que si on les combine avec la méthode mot-clés.[13]

6. Applications de résumé automatique de texte :

6.1 TextRank :

Est un algorithme basé sur *PageRank*, souvent utilisé dans l'extraction de phrases et le résumé de texte. Ce système est considéré comme le système de résumé le plus connu. Qui utilise l'approche extractive et contient quater étapes de base.[14]

- Extraire toutes les phrases du document (texte).
- Un graphique est créé à partir des phrases extraites à l'étape 1. Les nœuds représentent les phrases, tandis que le poids sur les arêtes entre deux nœuds est trouvé en utilisant une fonction de similarité, comme la similarité cosinus.
- Cette étape consiste à trouver l'importance de chaque nœud en itérant l'algorithme jusqu'à la convergence, jusqu'à l'obtention de scores cohérents.
- Les phrases sont triées par ordre décroissant en fonction de leurs scores. Les k premières phrases sont sélectionnées pour faire partie du résumé.

6.2 Latent Semantic Analysis (LSA) :

Est une méthode algébrique-statistique qui extrait les structures sémantiques cachées des mots et des phrases, c'est-à-dire qu'elle extrait les caractéristiques qui ne peuvent pas être directement mentionnées.[15] La méthode de LSA se compose de trois étapes principales :

- **Création d'une matrice d'entrée :** Le document d'entrée est représenté sous forme d'une matrice pour le comprendre et effectuer des calculs dessus. Ainsi, une matrice de termes de document est générée. Les cellules sont utilisées pour représenter l'importance des mots dans les phrases.
- **Décomposition en valeurs singulières (SVD) :** dans cette étape, nous effectuons la décomposition en valeurs singulières sur la matrice de termes du document généré. SVD est une méthode algébrique qui peut modéliser les relations entre les phrases. L'idée de base derrière SVD est que la matrice de termes de document peut être représentée sous forme de points dans l'espace vectoriel.
- **Sélection de phrases :** en utilisant les résultats de SVD, différents algorithmes sont utilisés pour sélectionner les phrases importantes.

7. La Structure d’un article scientifique

Il n'y a pas de structure unique qui soit complètement acceptée pour les articles scientifiques, mais une structure générique peut être partiellement sinon totalement trouvée dans les articles scientifiques. Cette structure est principalement importante pour faciliter la communication entre les scientifiques au sujet de leurs résultats et/ou découvertes. Ce format rend également le document facile à lire à différents niveaux. Ainsi, cela aide le lecteur à trouver rapidement ce dont il a besoin.[16]

Un article scientifique peut être contient les éléments suivants :

- **Le résumé (l’abstract) :**

Un résumé est la première section d'un article scientifique. Il contient généralement 150 à 250 mots ou moins et contient un résumé informatif des principaux aspects de l'article sans citations. Elle doit répondre aux trois questions principales : Pourquoi l'étude a-t-elle été menée ? Comment a-t-il été fabriqué ? Quelle est la conclusion atteinte ? Il couvre les objectifs de l'article, les matériaux, les méthodes utilisées, les résultats et les conclusions.

- **Les mots clés :**

Les résumés sont généralement accompagnés d'une série de mots clés. Cela rend les articles plus faciles à citer et à trouver en ligne. La plupart du temps, les revues scientifiques nécessitent 3 à 10 mots-clés par article.

- **L'introduction :**

L'introduction présente au lecteur les bases du sujet, ainsi que le "pourquoi" de l'étude. Son rôle dans la construction de l'information définit l'axe de recherche et conduit à la problématisation. il peut contenir plusieurs parties :le contexte, La problématique etc.

- **La méthodologie :**

La méthodologie répond au "comment" des questions de recherche scientifique. Cette section constitue le cœur de l'article. Il explique en détail les principaux éléments de l'étude, les étapes de réalisation et les méthodes expérimentales utilisées pour tester l'hypothèse.

- **Les Résultats :**

Cette section présente les résultats. Ceux-ci sont parfois présentés sous forme de tableaux, schémas ou graphiques afin de mieux les analyser. La partie des résultats peut être divisée en sous-parties présentées de manière logique (par exemple, une partie par expérience menée) afin de répondre au mieux à la question de recherche.

- **La Discussion :**

Une discussion doit suivre les résultats, afin de les analyser et de leur donner une signification scientifique. Dans certaines revues, il est possible de trouver la discussion dans la section des résultats.

- **La conclusion :**

La conclusion permet de dresser un bilan, établir un résumé des résultats et des principales interprétations de la recherche. C'est un lieu où le contexte peut être rappelé et comparé aux résultats obtenus.

8. Résumé automatique de texte scientifique

Les trois sous-sections suivantes passent en revue les approches de pointe en matière de résumé d'articles scientifiques selon la classification proposée à la Figure (2.2). Il existe principalement deux classes d'approches pour le résumé d'articles scientifiques : les approches basées sur l'extraction et les approches basées sur les citations. Il existe également d'autres approches pour le résumé d'articles scientifiques qui se concentrent sur des problèmes spécifiques tels que le résumé de tableaux, de figures et de sections de travail connexes.[16]

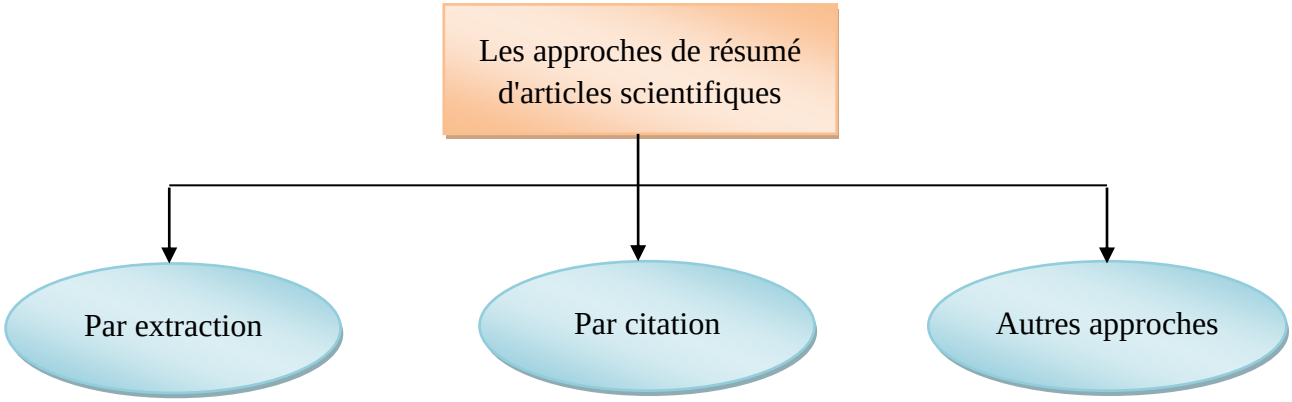


Figure 2.2 : Les approches de résumé d'articles scientifiques

8.1 Résumé par extraction :

L'un des aspects intéressants de nombreuses applications de résumé de texte est la création automatique d'un résumé de recherche. Ceci est un résumé du sujet principal et des résultats présentés dans l'article correspondant. Les chercheurs peuvent utiliser ce résumé pour obtenir un aperçu de l'article. Il peut également aider d'autres systèmes automatisés à indexer, rechercher et récupérer des informations sans accéder à l'intégralité du document. Bien qu'ils puissent être générés automatiquement à l'aide de techniques de synthèse de texte, le processus est assez difficile.

8.2 Résumé en utilisant les citations :

Les phrases de citation traitent généralement des informations les plus importantes de l'article cité. Ils mettent en évidence le problème de recherche, la méthode proposée, les résultats rapportés, les contributions et même les inconvénients et les limites. L'ensemble des phrases de citation vers un article cible pourrait être considéré comme un court résumé rédigé par les chercheurs cités et présentant l'impact de l'article cible sur la communauté de recherche. Une façon d'utiliser ces phrases est de créer un résumé d'article. Cela diffère du résumé de l'article, puisque le résumé de la citation représente les points de vue de plusieurs chercheurs, alors que le résumé ne reflète que les points de vue des auteurs. L'utilisation de phrases de citation dans la tâche de résumé automatique d'un article scientifique génère un résumé basé sur les citations.[16]

9. Conclusion

Dans ce chapitre, nous avons présenté un portrait global de résumé automatique. Tout d'abord, nous avons donné quelques définitions du résumé, pour comprendre ce domaine. Un résumé automatique peut appartenir à plusieurs classes ou types, car il existe plusieurs facteurs de classification. Donc, Nous avons vu que l'on peut décomposer ces facteurs selon la source (le document d'entrée), l'objectif et le document de sortie. Nous avons vu que le résumé consiste en deux approches de base, le résumé par abstraction et le résumé par extraction. Ensuite, nous avons fourni quelques méthodes pour le résumé. Enfin, nous avons expliqué le résumé automatique du texte scientifique et ses types.

Dans le chapitre suivant nous allons présenter les méthodes d’évaluation de résumé et les métriques utilisées.

Chapitre 3 : Evaluation de résumé automatique de texte

1. Introduction

L'évaluation de la qualité d'un résumé est un problème difficile auquel il n'existe que des solutions partielles. Plusieurs facteurs sont derrière cette problématique. En effet, il n'y a pas de résumé idéal ou parfait. Plusieurs résumés peuvent convenir au même document, et ils ne sont pas forcément similaires ou convergeant au niveau du contenu. Par conséquent, deux résumés pour le même document peuvent être produits en utilisant un vocabulaire totalement différent. Voire même, la même personne peut résumer le même texte de manière différente au fil du temps.

L'évaluation est une étape très importante dans le développement d'une application informatique, et en particulier les systèmes de résumé automatique. Elle concerne le contenu et la qualité des résumés produits, afin d'estimer la capacité d'une application à effectuer des tâches qu'on lui soumet. L'évaluation de la qualité des résumés produits teste la lisibilité, la grammaticalité et la cohérence du résumé. Toutefois, ces évaluations peuvent être faites d'une façon intrinsèque ou extrinsèque ce dont nous parlerons plus loin dans ce chapitre.

2. Les approches d'évaluation de résumé automatique

Évaluer l'efficacité du système de résumé automatique est une tâche très ambitieuse. Il n'y a pas de méthode spécifique mais il existe différentes méthodes. Nous pouvons comparer le résumé généré par le système avec le document original, ou un résumé préparé par un humain, ou un résumé d'un autre système. Les approches d'évaluation des résumés automatisés peuvent être classées en deux grandes catégories : l'évaluation intrinsèque et l'évaluation extrinsèque.[17]

Dans l'évaluation extrinsèque, la qualité du résumé est jugée sur la base de l'utilité des résumés pour une tâche particulière, et dans l'évaluation intrinsèque, elle est directement basée sur l'analyse du résumé.

Les Méthodes d'évaluation de Résumé Automatique peuvent être classées selon les critères suivants (voir la figure 3.1).

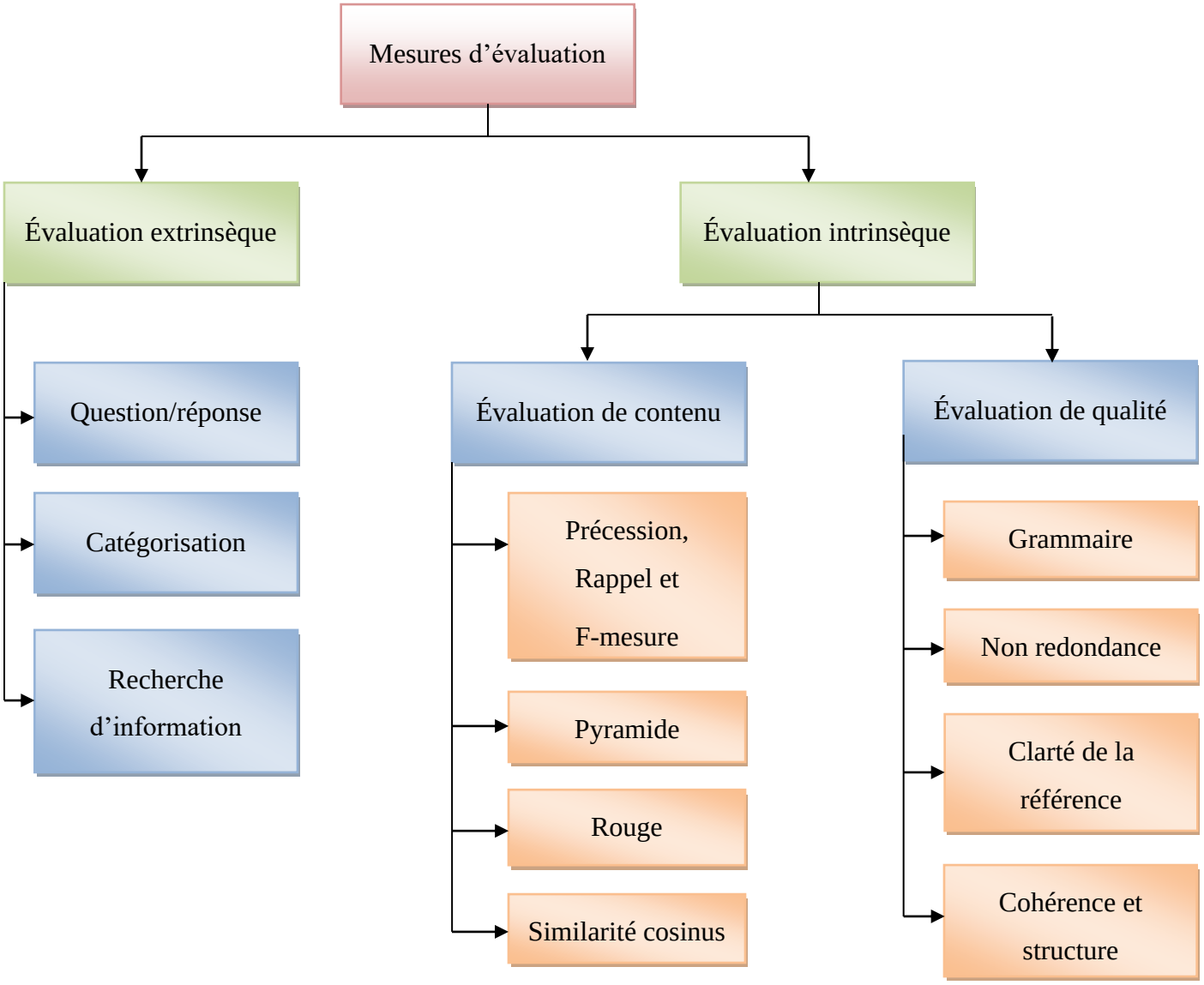


Figure 3.1 : Taxonomie Des Méthodes d'évaluation de Résumé Automatique

3. L'évaluation extrinsèque

Dans l'évaluation extrinsèque, le résumé est évalué selon leur utilité pour une tâche cible. Les méthodes d'évaluation basées sur les tâches n'analysent pas les unités lexicales dans le résumé, mais tentent de mesurer la probabilité que ces résumés soient utilisés dans une tâche particulière. Il existe différentes approches de l'évaluation de résumé basée sur les tâches dans la littérature. Parmi ces approches, nous mentionnons les trois tâches les plus importantes : la catégorisation des documents, la recherche d'informations et la réponse aux questions.[11]

3.1 Catégorisation

La qualité des résumés automatiques peut être mesurée par sa capacité à remplacer des documents entiers pour la catégorisation. Ici, l'évaluation vise à déterminer si le résumé générique est efficace pour saisir les informations dans le document original qui sont nécessaires pour le catégoriser correctement. Un ensemble de documents ainsi que les sujets auxquels ils appartiennent sont nécessaires à cette tâche. Les résultats obtenus en catégorisant les résumés sont généralement comparés à ceux obtenus en catégorisant des documents complets.

La catégorisation peut être effectuée manuellement ou par classificateur. Si nous utilisons la méthode de classification automatique, nous devons garder à l'esprit que le classificateur présente certaines erreurs inhérentes. Il faut donc faire la distinction entre les erreurs générées par le classifieur et celles causées par le système de résumé automatique.[18]

3.2 Recherche d'information

La Recherche d'Information (RI) est le domaine consistant à trouver un objet dans tout média pertinent pour répondre à la requête d'un utilisateur. Jusqu'à récemment, la recherche de techniques efficaces pour la Recherche d'Information ne concernait que des domaines très spécifiques comme la loi, la médecine, le commerce. Avec l'avènement du Web tel nous le connaissons, nous sommes tous devenus des utilisateurs de la Recherche d'Information. Ce changement a poussé la communauté de la Recherche d'Information à développer de nouvelles techniques.

Une tâche de recherche d'informations (RI) peut être utilisée pour évaluer la qualité du résumé. La pertinence est une mesure basée sur RI pour évaluer la baisse relative des performances de recherche (indexation) lors du passage de documents complets à des résumés. Dans cette expérience, les auteurs ont démontré que les résumés courts sont de bonnes alternatives aux documents complets à haute résolution. Ainsi, si les résumés capturent bien les points importants des documents, le système IR indexe ces résumés de la même manière ou presque si un ensemble de documents complets est utilisé.[18]

3.3 Question réponse

Un autre type d'évaluation extrinsèque est la tâche de compréhension de la lecture. Dans cette évaluation, un utilisateur reçoit une source complète et un résumé textuel. Ensuite, le résumé généré par un système est soumis à un test à choix multiple (question a des choix multiples), dont la réponse est dans le document source. Ce cadre d'évaluation est basé sur l'idée qu'un vrai résumé doit pouvoir remplacer la source.[18]

4. L'évaluation intrinsèque

L'évaluation intrinsèque est souvent effectuée en comparant le résumé généré automatiquement avec un résumé idéal généré par un humain ou un autre système de résumé, et peut être divisée en **évaluation de contenu** et **évaluation de la qualité du texte**. Les évaluations de contenu mesurent la capacité à identifier les principaux sujets. En revanche, les évaluations de la qualité des textes évaluent les résumés en termes de lisibilité, de grammaire et de cohérence.[19]

4.1 Evaluation de qualité

Les mesures basées sur la qualité de texte contrôlent les aspects linguistiques, sont les suivantes :

a) Grammaire :

Le texte ne doit pas contenir d'éléments non textuels (par exemple, des balises), d'erreurs de ponctuation, de formatage du système interne, de phrases non grammaticales peu claires ou de mots incorrects. . .

b) Non-redondance :

Le texte ne doit pas contenir des informations redondantes.

c) Clarté de la référence :

Les noms et les pronoms doivent être clairement mentionnés dans le résumé. Par exemple, le pronom « il » doit signifier « quelqu'un » dans le résumé.

d) Cohérence et structure :

Le résumé doit avoir une bonne structure et les phrases doivent être cohérentes.

4.2 Evaluation du contenu

Les métriques basées sur le contenu comparent les mots d'une phrase plutôt que la phrase entière. Ses avantages sont qu'elles peuvent comparer des résumés générés par un système avec des résumés générés par des humains contenant des nouvelles phrases écrites.

4.2.1 Les mesures de rappel et de précision

Le rappel et la précision proposent des mesures de similarité classiques pour la recherche d'informations. Ces mesures de la discipline visent à montrer à quel point les performances obtenues par le système se rapprochent des performances obtenues manuellement par des humains.[20]

Ces mesures sont calculées, pour un système de résumé automatique de texte, tel que :

R*sum* : un ensemble de phrases du résumé de référence.

C*sum* : un ensemble de phrases du résumé de système (candidat).

C*sum* ∩ R*sum* : l'intersection entre le résumé de référence et le résumé du système.

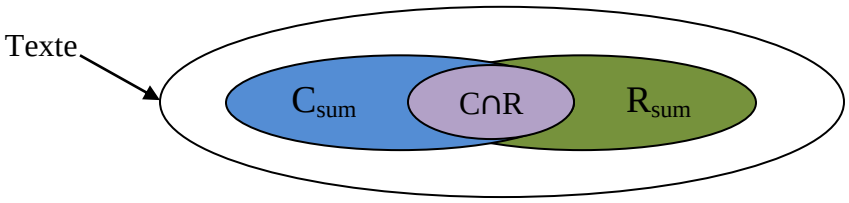


Figure 3.2 : Paramètres de calcul de la précision et du rappel

Dans ce qui suit, nous expliquons les formules pour mesurer le rappel, la précision et la F-mesure.

a) La précision :

La précision fait référence à l'exactitude du système, en d'autres termes, le nombre de phrases correctes données par le système (le résumé généré automatiquement) par rapport au nombre total des phrases du résumé de système. Formellement, la précision est le quotient suivant.

$$Précision = \frac{|C_{sum} \cap R_{sum}|}{|C_{sum}|}$$

b) Le rappel :

Le rappel permet de mesurer le nombre de phrases correctes dans le résumé de système par rapport au nombre total des phrases du résumé de référence.

$$Rappel = \frac{|C_{sum} \cap R_{sum}|}{|R_{sum}|}$$

c) La F-mesure :

Afin de pondérer l'importance de chacun de ces deux paramètres (précision et rappel), un troisième paramètre, appelé La F-mesure, est généralement calculé à partir de ces deux paramètres.

$$F - mesure = \frac{2 * Précission * Rappel}{Précission + Rappel}$$

4.2.2 La méthode Pyramide

Les métriques générées par la méthode PYRAMID sont basées sur la division des résumés de référence en unités d'information sémantique initiales (SCU) "*Semantic Content Unit*", puis sur la vérification de la présence de ces unités dans le résumé du système. Il s'agit d'identifier, à partir des résumés de référence, les unités sémantiques (SCU) qui expriment, de différentes manières, une même idée, dont le poids dépend de leur présence dans chacun des résumés de référence. Ainsi se construit la pyramide dont le sommet est occupé par les unités communes aux abrégés de référence. Il s'agit donc de déterminer maintenant, dans le résumé système à évaluer, quelles unités correspondent à celles de la pyramide.[20]

La somme des scores unitaires standard est donnée par l'équation suivante :

$$Pyramid = \frac{\sum poids(SCU_{système})}{\sum poids(SCU_{référence})}$$

4.2.3 La méthode ROUGE :

ROUGE qui est une abréviation de "*Recall Oriented Understudy for Gisting Evaluation*" est un ensemble de mesures utilisées pour l'évaluation de résumé automatique de texte. Les métriques comparent essentiellement le résumé généré automatiquement (appelé aussi résumé de système) avec le résumé de référence ou plusieurs résumés de référence [21]. Il s'agit d'un package qui comprend diverses mesures de ROUGE :

- ROUGE-N: N-gram Co-Occurrence Statistics
- ROUGE-L: Longest Common Subséquence
- ROUGE-W: Weighted Longest Common Subsequence
- ROUGE-S: Skip-Bigram Co-Occurrence Statistics
- ROUGE-SU: extension de ROUGE-S

ROUGE-N, ROUGE-S et ROUGE-L sont très utiles pour l'évaluation de résumés.

ROUGE-N :

Formellement, ROUGE-N est un rappel de n-grammes entre un résumé du système et un ensemble de résumés de référence (summ-ref). ROUGE-N se calcule comme suit :

$$ROUGE - N = \frac{\sum_{S \in summ-ref} \sum_{N-gram \in S} Count_{Match}(N-gram)}{\sum_{S \in summ-ref} \sum_{N-gram \in S} Count(N-gram)}$$

Où $Count_{Match}(N-gram)$ est le nombre de n-grammes qui existent dans un résumé de système et retrouvé dans un résumé de référence et $Count(N-gram)$ est le nombre de n-grammes dans le résumé de référence.

ROUGE-L :

Mesure la séquence de mots correspondante la plus longue à l'aide de LCS (*Longest Common Subsequence*). Un avantage de l'utilisation de LCS est qu'il ne nécessite pas de

correspondances consécutives mais des correspondances en séquence qui reflètent l'ordre des mots au niveau de la phrase. Puisqu'il inclut automatiquement les n-grammes communs les plus longs dans la séquence, vous n'avez pas besoin d'une longueur de n-gramme prédéfinie.

ROUGE-S :

Est-ce que n'importe quelle paire de mots dans une phrase est dans l'ordre, en tenant compte des lacunes arbitraires. Cela peut aussi être appelé accord de saut de gramme. Par exemple, skip-bigram mesure le chevauchement des paires de mots qui peuvent avoir un maximum de deux espaces entre les mots.

Par exemple, pour l'expression "chat sur le tapi ", les bigrammes de saut seraient " chat sur, chat le, chat tapi, sur le, sur tapi, le tapi ".

ROUGE-SU :

La faiblesse de ROUGE-S est qu'il ne considère que les bigrammes. Si une phrase ne contient aucun chevauchement de bigrammes, cela ne donnera aucun poids à ces phrases. Pour surmonter ce problème de ROUGE-S, ROUGE-SU est une extension qui considère également l'unigramme avec les bigrammes.

4.2.4 La similarité cosinus :

La similarité cosinus calcule la similarité entre deux vecteurs en déterminant l'angle cosinus entre eux comme suit :

$$Cos(X,Y) = \frac{\sum_i X_i * Y_i}{\sqrt{\sum_i (X_i)^2} * \sqrt{\sum_i (Y_i)^2}}$$

Où X et Y sont deux vecteurs d'unités (textes, paragraphes, phrases ou mots) soit résumé de système ou de référence, plus l'angle entre les deux vecteurs est petit plus ils sont similaires.[18]

5. Les compagnes d'évaluation de résumé :

Il existe de plusieurs compagnes de résumé automatique de texte, notamment les suivantes.

5.1 Campagnes d'évaluation TIPSTER SUMMAC :

En 1998, les évaluations de résumé automatique de texte TIPSTER SUMMAC (*Summarization Conference*) (organisées par le *National Institute of Standards and Technology* (NIST)) ont eu lieu dans le Maryland (USA). SUMMAC a été la première évaluation à grande échelle, indépendante du développeur, des systèmes de résumé automatique de texte. Sur la base des activités, qui étaient généralement menées par des analystes de données du gouvernement américain, trois grandes tâches d'évaluation ont été spécifiées :

- La tâche ad hoc (intrinsèque).
- La tâche de catégorisation (intrinsèque).
- La tâche de question-réponse (QA) (extrinsèque).

SUMMAC a organisé une évaluation du résumé automatique de texte qui s'est avérée très efficace pour mesurer la pertinence. Les algorithmes de résumé ont permis d'éliminer le texte (respectivement 83% et 90% pour le résumé guidée et générique), tout en obtenant le même niveau de pertinence que les documents sources et en réduisant de moitié environ le temps d'analyse. La tâche d'assurance qualité a présenté de nouvelles méthodes automatiques pour mesurer le caractère informatif d'un résumé précis guidé par sujet. La notation automatique basée sur le contenu est en corrélation positive avec les notes produites par les juges humains. Les méthodes d'évaluation utilisées dans la campagne SUMMAC avaient deux intérêts : évaluer des résumés et évaluer d'autres résultats liés aux technologies de traitement du langage naturel (TAL).[7]

5.2 Campagnes d'évaluation NTCIR

Le troisième atelier NTCIR (2001-2002) s'est concentré sur l'évaluation des systèmes d'extraction d'informations (tâche RI), des systèmes de questions-réponses (tâche QA) et des systèmes de résumé automatique de texte (tâche de résumé automatique de texte). Organisé par le NTCIR au Japon, le "*Text Summarization Challenge*" (TSC) visait à résumer automatiquement des textes publiés de 1998 à 1999 dans le journal japonais Mainichi.[7] Il y avait trois objectifs :

- Promouvoir la recherche en RI, QA et le résumé de texte en fournir des corpus de test réutilisables.

- Créer un forum d'échange pour les groupes de recherche intéressés par comparer les résultats et les idées dans une atmosphère informelle.
- Améliorer la qualité des corpus de tests basés sur les commentaires.

5.3 Campagnes d'évaluation DUC/TAC :

Depuis 2001, le NIST organise le *"Document Understanding Campagnes de conférence*" (DUC) pour évaluer les performances des algorithmes NLP. À partir de 2008, ces campagnes ont été rebaptisées *Text Analysis Conference* (TAC). Les campagnes TAC sont beaucoup plus ambitieuses que les campagnes DUC. A partir de 2008, le TAC a organisé des ateliers autour de quatre thématiques : résumé, question-réponse, reconnaissance de l'implication textuelle et population de la base de connaissances. L'objectif des campagnes DUC/TAC est double : d'une part, favoriser les progrès dans le domaine du résumé automatique de textes et, d'autre part, permettre aux chercheurs de participer à des expérimentations à grande échelle, leur permettant à la fois de développer et d'évaluer leurs systèmes.[7]

Les campagnes DUC/TAC ont successivement introduit les tâches suivantes :

- DUC 2001 -DUC 2002 : Résumés génériques mono et multi-documents.
- DUC 2003 : Résume court (*headline*) et multi-document.
- DUC 2004 : Résumés Courts, Résumé multilingues multi-documents et résumé biographiques.
- DUC 2005 : Résumé guidé et résumé biographique guidé par des questions sur le sujet
- DUC 2006 : Résumé guidée de plusieurs documents.
- DUC 2007 : Résumé guidée multi-documents pour des longueurs de ou jusqu'à 250 mots, à partir de groupes d'environ 25 documents, mettre à jour résumé.
- TAC 2008-TAC 2009 : Résumés mise à jour (*Update task summarization*) - Résumé guidé multi-documents.
- TAC 2010 : Résumés orientés multi-documents (*Guided summarization*), Evaluation automatique de résumé (*Automatically Evaluating Summaries Of Peers*)
- TAC 2011 : Résumé guidé (*Guided summarization*), Evaluation automatique de résumé et la nouvelle tâche MultiLing pilot pour favoriser l'utilisation d'algorithmes multilingues pour le résumé.

- TAC 2014 : Résumé de textes biomédicaux (*Biomedical Summarization*)

6. Concusion :

Dans ce chapitre, nous avons couvert les idées de base de l'évaluation automatique du résumé de texte. L'évaluation automatique du résumé ne peut pas être décrite dans un chapitre car c'est un domaine de recherche en soi. Premièrement, nous avons identifié deux approches de l'évaluation de résumé automatique : l'évaluation intrinsèque et l'évaluation extrinsèque. Pour évaluer le résumé, des mesures sont utilisées, parmi celles-ci nous avons cité les plus utilisées dans l'évaluation intrinsèque sont ROUGE et Pyramide, Enfin Nous avons mentionné les différentes campagnes d'évaluation : SUMMAC, NTCIR et DUC/TAC.

Dans le chapitre suivant, nous allons présenter notre système de résumé automatique basé sur l’approche extractive. L'idée principale est de résumer automatiquement un article scientifique par l’extraction.

Chapitre 4 : Conception et réalisation du système

1. Introduction :

Résumer un texte consiste à réduire ce texte en un nombre limité de mots. Le texte ainsi réduit doit rester fidèle aux informations et idées du texte original, et dans la mesure du possible rendre compte du style et de l'intention de l'auteur. L'objectif de la majorité des systèmes de résumé automatique de texte (mono-document) est de résumer un texte complet (fichier, article,), par contre, dans notre travail, l'objectif est de résumer la contribution d'un article scientifique, dans ce chapitre nous allons décrire notre système de résumé automatique d'un article scientifique.

2. Architectur globale de notre système

Comme nous avons vu dans le deuxième chapitre, il existe deux approches de résumé automatique, dans notre système nous allons utiliser l'approche par extraction.

Notre système est un système de résumé automatique d'un article scientifique écrit en anglais. La mise en œuvre fonctionnelle de notre système est représentée dans la Figure (4.1).

Elle repose principalement sur deux modules qui peuvent communiquer entre eux afin de permettre la génération de résumé.

L'entrée de notre système est un article scientifique bien structuré, pour pouvoir résumer sa contribution on doit passer par le module préparation où le système sépare ses sections en utilisant l'architecture particulière de l'article scientifique.

Donc le système de résumé d'un article scientifique comprend deux étapes majeures :

- Préparation de document
- Génération de résumé

Le schéma suivant représenté l'organigramme de notre système.

Dans la section suivante nous allons expliquer en détail les étapes de notre système.

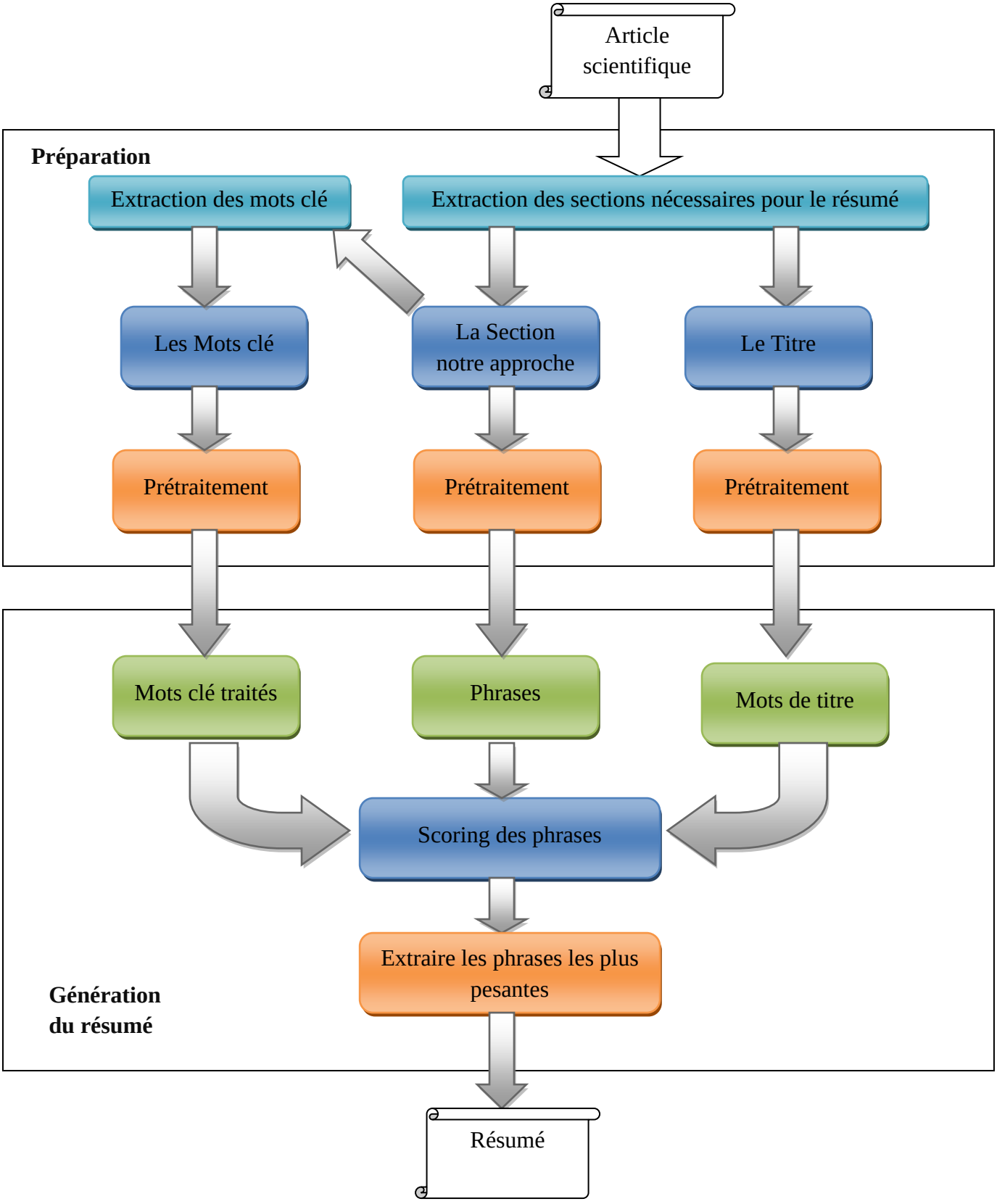


Figure 4.1 : L'architecture globale de notre système

3. Architecure détaillée de notre système

Comme nous l'avons déjà mentionné, afin de générer le résume on va passer par deux modules : le module de préparation et le module de génération de résumé.

3.1 Module de préparation

L'entrée de ce module est l'article scientifique que nous allons résumer, comme nous l'avons mentionné précédemment, le but de notre travail est de résumer la contribution d'un article scientifique, donc, nous n'avons pas besoin de traiter tout le texte. Nous n'avons besoin de traiter que certaines sections, il existe deux étapes dans ce module.

3.1.1 Extraction des sections nécessaires pour le résumé

a) Extraction de la section qui contient la contribution (notre approche) :

Notre objectif est de résumer la contribution de l'article, donc il faut extraire la section qui sera résumée et qui contient la contribution de l'article, nous appelons cette partie la section notre approche.

b) Extraction de titre :

Le titre est l'une des parties les plus importantes de l'article, généralement il reflète la contribution de ce dernier, il doit donc être extrait pour être utilisé dans le résumé.

c) Extraction des mots clé :

Les mots-clés ne sont pas moins importants que le titre, ils doivent donc être extraits pour être utilisés dans le résumé. Nous extrayons les mots avec le poids le plus élevé comme des mots clé, Il est à noter que ces mots-clés sont extraits à partir de la section notre approche, Comme illustré à la figure (4.1).

3.1.2 Le Prétraitement :

a) Prétraitement de la section notre approche :

Tout d'abord, nous devons supprimer les signes de ponctuations, puis segmenté le texte en phrases, ensuit chaque phrase est divisée en mots (Tokenisation), puis supprimer les mots vides(stop-words) de toutes les phrases à la fin, nous appliquons le stemmer aux mots.

b) Prétraitement de titre :

Nous commençons par supprimer les signes de ponctuation, puis nous divisons le titre en tokens (mots séparés), ensuit nous supprimons les mots vides, finalement nous appliquons le stemmer aux mots.

c) Prétraitement des mots clé :

Le prétraitement des mots-clés ne contient que la Tokenisation et l'application du stemmer sur les mots.

3.2 Module de génération de résumé :

Les entrées de ce module sont les mots qui compose le titre après le prétraitement, les mots clé et les phrases de la section notre approche, l'objectif de ce module est de sélectionner les phrases les plus pertinent qui compose le résumé, ces derniers doivent avoir les scores les plus élevés parmi les phrases candidates, dans la section suivante nous allons expliquer comment calculer le score d'une phrase.

3.2.1 Représentation du texte :

La machine ne peut pas utiliser le texte tel qu'il est pour calculer les scores des phrases, donc le texte doit être représenté sous forme compréhensible par la machine.

Donc dans cette étape, nous allons utiliser la méthode TF IDF pour la codification du texte, est une méthode de représentation des textes sous forme numérique. TF-IDF signifie Term Frequency Inverse Document Frequency. TF-IDF effectue une pondération en tenant compte de deux facteurs. D'une part, la fréquence du mot dans un document (dans une phrase) donné, c'est le terme Frequency (TF), d'autre part, la fréquence de ce mot dans tous les autres documents (dans tous les phrases), C'est le document Frequency (DF). Puis, on va multiplie le terme Frequency par l'Inverse Document Frequency, TF multiplié par IDF. Le résultat de la multiplication est TF-IDF. À présent, voyons maintenant comment fonctionne la méthode. L'idée de la transformation TF-IDF est de donner un poids élevé aux mots apparaissant souvent dans un document, tout en prenant soin de diminuer le poids de ce mot s'il apparaît également dans d'autres documents du corpus.

Nous devons appliquer cette méthode pour calculer le score 1 (La similarité des phrases de l'article avec le titre) et le score 2 (La similarité avec les mots-clé), cette étape a pour but de créer

les vecteurs qui représentent les phrases, le titre et les mots clé qui sont les entrées de la fonction qui calcule la similarité.

3.2.2 Scoring des phrases :

Ce module assigne pour chaque phrase un score, le score d'une phrase est calculé comme suit :

$$\boldsymbol{Score(phrase) = S1 + S2 + S3 + S4}$$

a) La similarité avec le titre (S1)

Comme le titre couvre généralement le thème principal abordé dans le texte, les mots de titre sont des mots pertinents et peuvent être considérés comme des termes clés. De ce fait, les phrases qui contiennent les mots du titre sont des phrases pertinentes et doivent être incluses dans le résumé final. Le score de similarité avec le titre attribué à chaque phrase est fonction des occurrences des mots de titre dans la phrase.

On a choisir la similarité cosinus qui permet de calculer la similarité entre deux vecteurs à N dimensions en déterminent le cosinus de l'angle entre eux.

Soit deux vecteurs $\vec{A}$ et$\vec{B}$.Le cosinus de leur angle θ s'obtient par la formule suivante :

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

- θ Est l'angle entre les vecteurs (A et B)
- La similarité obtenue ∈ [0 ; 1]
- A.B = le produit des vecteurs $\vec{A}$ et $\vec{B}$.
- ||A|| et ||B|| = longueur des deux vecteurs $\vec{A}$ et $\vec{B}$.

Exemple :

Soit les deux vecteurs suivants : $\vec{A} = \{3, 2, 0, 5\}$ et $\vec{B} = \{1, 0, 0, 5\}$

Donc, pour calculer la similarité entre les vecteurs A et B on va utiliser la formule de similarité cosinus :

$$\cos(A,B) = \frac{3*1 + 2*0 + 0*0 + 5*5}{\sqrt{3^2+2^2+0^2+5^2} * \sqrt{1^2+0^2+0^2+5^2}} = \frac{28}{6.16 * 5.09} = 0{,}89$$

b) L'existence des mots clés (S2).

Pour le calcul de ce score, on considère les mots clé (keywords) comme une phrase et on calcule la similarité entre cette phrase et chaque phrase de texte extrait à partir de la section qui contient la contribution de l'article.

Exemple:

« Developing the program logic to solve the particular problem. Writing the program logic in a specific programming language (coding the program). Assembling or compiling the program to turn it into machine language. »

Mots-clé: Developing, program, coding, language

La similarité entre les phrases et la phrase que nous avons faite à partir de mots clés :

$Score_{\text{mots-clé}}(phrase\_1) =$ Similarité (phrase_1, phrase de mot-clé) = 0.30

$Score_{\text{mots-clé}}(phrase\_2) =$ Similarité (phrase_2, phrase de mot-clé) = 0.47

$Score_{\text{mots-clé}}(phrase\_3) =$ Similarité (phrase_3, phrase de mot-clé) = 0.22

Si nous voulons créer le résumé à partir de l'extraction des deux premières phrases avec le score la plus élevée, Donc le résumé « Writing the program logic in a specific programming language (coding the program). Developing the program logic to solve the particular problem. »

c) L'existence des cue phrases (S3)

Cette méthode choisit des unités de texte avec des indications spécifiques ou des expressions spécifiques. Par exemple, dans le cas des textes scientifiques, il existe des expressions comme :

« In this study, in this paper we propose, this paper presents, goal of this paper……etc. »

Sont des bons candidats pour indiquer les phrases à inclure dans un résumé. Les textes de différents types peuvent comporter différentes expressions indicatives. On peut définir le score d'une phrase S correspondant à un motif particulier comme suit :

$$Score_{cue}(S) = \begin{cases} 1 \; si \; S \; correspond \text{ à } un \; motif \\ 0 \; sinon \end{cases}$$

Exemple :

« **This paper presents** a study of the runtime, memory usage and energy consumption of twenty-seven well-known software languages**.** We monitor the performance of such languages using ten different programming problems, expressed in each of the languages**. Our results** show interesting findings, such as, slower/faster languages consuming less/more energy, and how memory usage influences energy consumption**.** »

Ensemble de cue-phrases = [this paper presents, in this study, our results… etc.]

L'existence des cue words dans les phrases :

$Score_{cue}(phrase\_1) = 1$

$Score_{cue}(phrase\_2) = 0$

$Score_{cue}(phrase\_3) = 1$

Si nous voulons créer le résumé à partir de l'extraction des deux premières phrases avec le score la plus élevée, Donc le résumé se compose de la phrase 1 et de la phrase 3

d) La position de la phrase (S4)

Cette méthode attribue un poids à une phrase en fonction de la position de cette phrase dans le document. Elle suppose que la position d'une phrase dans un texte montre combien elle est importante dans le contexte. Les premières phrases d'un paragraphe, par exemple, peuvent faire passer l'idée principale et devraient donc être incluses dans le résumé.

$$Score_{\text{position}}\,(Si) = \beta i$$

$$\beta i = \begin{cases} B \in [0\,;1]\ si\ \ i < N \\ 0\ si\ \ i \geq N \end{cases}$$

N : le nombre de premières phrases.

i : la position de la phrase dans le paragraphe.

Exemple:

« This is the first sentence. This is the second sentence. This is the third sentence. This is the fourth sentence. »

Nous avons choisi le nombre de premières phrases N = 3, et le score B = 0.5, donc :

Pour la première phrase i = 1 < N

$$Score_{\text{position}}\,(phrase\_1) = 0.5$$

Pour la deuxième phrase i = 2 < N

$$Score_{\text{position}}\,(phrase\_2) = 0.5$$

Pour la troisième phrase i = 3 = N

$$Score_{\text{position}}\,(phrase\_3) = 0$$

Pour la quatrième phrase i = 4> N

$$Score_{\text{position}}\,(phrase\_4) = 0$$

Si nous voulons créer le résumé à partir de l'extraction des deux premières phrases avec le score la plus élevée, Donc le résumé se compose de la phrase 1 et de la phrase 2

3.2.3 Sélection des phrases de résumé

Cette partie permet de retourner le résultat final suivant le choix du nombre de phrases extraites par rapport au nombre de phrases contenues dans le document.

Après avoir calculer le score de chaque phrase candidate au résumé, dans cette partie nous allons trier les phrases candidates selon leurs scores. Les phrases qui possèdent les scores les plus élevés sont les phrases sélectionnées par ce processus, le nombre de phrases est déterminer par l'utilisateur.

Après avoir sélectionné les phrases avec le score le plus élevé, nous les classons selon leur position dans le texte original, afin d'obtenir le résumé.

4. L'évaluation de résumé

Pour évaluer les performances de notre système nous avons utilisé la mesure ROUGE, elle comprend un ensemble de métriques qui déterminent la qualité d'un résumé, en comptant les unités qui se chevauchent, comme les n-grammes et les séquences de mots entre les résumés générés par notre système et les abstracts des articles. Les principaux avantages de ROUGE sont, sa simplicité et sa forte corrélation avec les jugements humains. Nous notons que nous avons utilisé la version 1.0.1 du package rouge-metric[1] implémenté en langage python, qui contient les cinq principales mesures.

Pour chaque métrique, nous présentons les valeurs obtenues en termes de précision, de rappel et de f-mesure.

4.1 Corpus d'évaluation

Pour évaluer les performances d'un tel système en utilisant les métriques ROUGE, l'utilisation d'un corpus d'évaluation est nécessaire. Ce corpus doit contenir un ensemble de texte à résumer avec ses résumés de référence (généralement produits par des êtres humains), chaque texte peut avoir un ou plusieurs résumés de référence.

Comme nous l'avons mentionné précédemment, et contrairement aux autres systèmes de résumé automatique des articles scientifiques qui résument l'article entièrement, l'objectif de notre système est de produire un résumé décrivant la contribution de l'article scientifique cible.

Donc, pour évaluer notre système nous avons utilisé un corpus de 183 documents de la collection cmp-lg (*Computation and Language*) a été balisé en xml et mis à disposition en tant que ressource générale pour les communautés de recherche. Les documents sont des articles scientifiques écrits en Anglais parus dans des conférences parrainées par ACL (*Association for Computational Linguistics*). Le corpus a été préparé par *The MITRE Corporation* et *The University of Edinburgh*. Fourni par TIPSTER SUMMAC.[2] Nous avons utilisé le résumé (l'abstract)de chaque article comme son résumé de référence.

[1] https://pypi.org/project/rouge-metric/
[2] https://www-nlpir.nist.gov/related_projects/tipster_summac/cmp_lg.html

4.2 L'évaluation de notre système

Pour évaluer notre système, nous allons utiliser le corpus d'évaluation cité auparavant. L'ensemble d'articles scientifiques du corpus sont résumés avec notre système, le système LSA, et le système de TextRank, puis nous utiliserons les métriques ROUGE pour évaluer les trois systèmes, à la fin nous comparerons les résultats obtenus.

Nous avons utilisé le package SUMY[3] implémenté en langage python, qui contient les systèmes : LSA et TextRank.

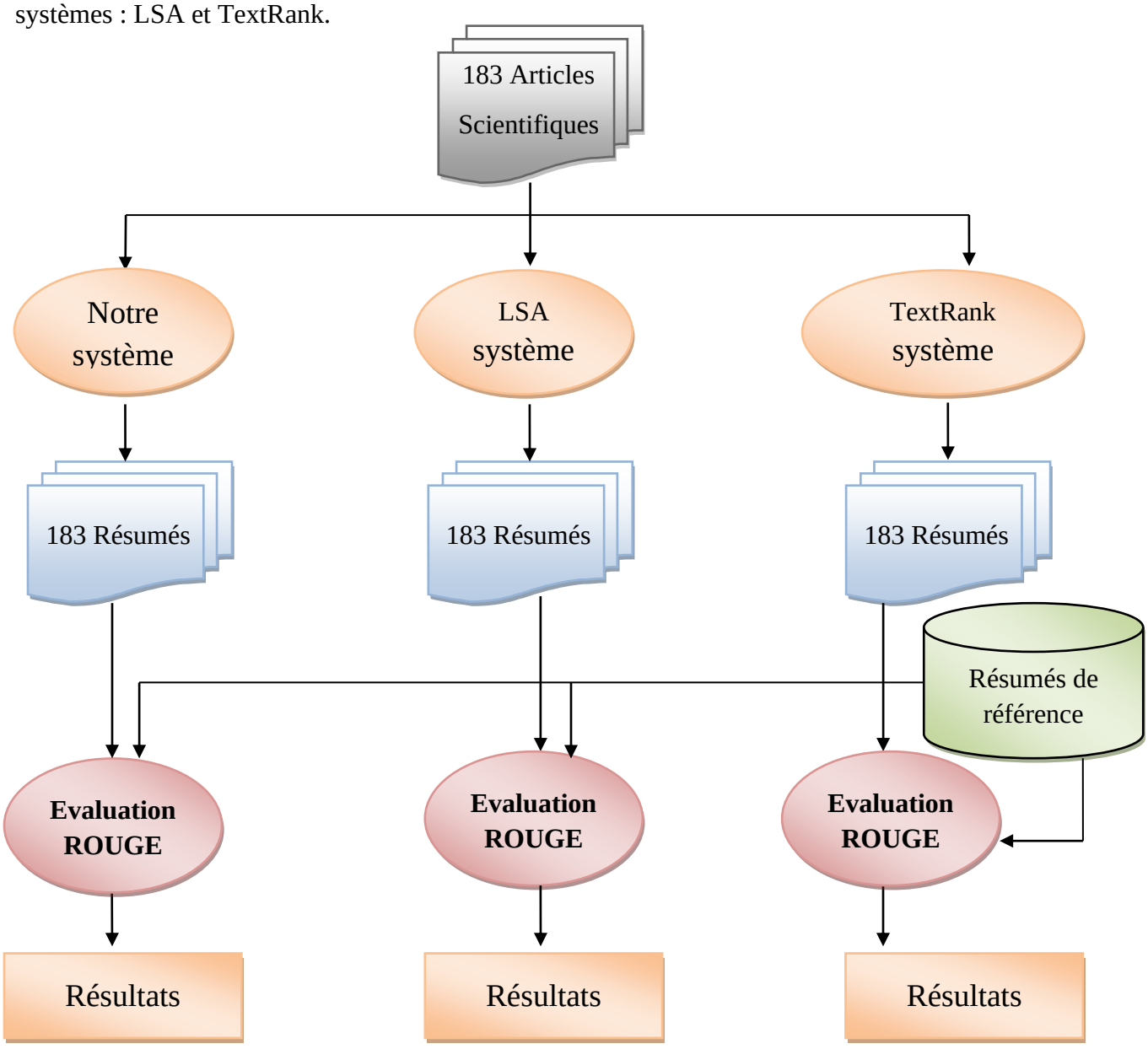


Figure 4.2 : Le processus d'évaluation de notre système

[3] https://pypi.org/project/sumy/

5. Implimentation

L'implémentation est la phase la plus importante après celle de la conception. Le choix des outils de développement influe énormément sur le coût en temps de programmation, ainsi que sur la flexibilité du produit à réaliser.

Cette phase consiste à transformer le modèle conceptuel établi précédemment en des composants logiciels formant notre système.

Dans cette section, nous allons commencer par la description de l'environnement de travail puis à présenter notre système.

5.1 Environnement de travail

L'environnement de travail se compose de deux parties : l'environnement matériel et l'environnement logiciel.

5.1.1 Environnement matériel

La configuration matérielle est décrite comme suit :

- ✓ Système d'exploitation : Windows 10
- ✓ CPU : Intel(R) Atom(TM) x5-Z835 CPU @ 1.44 GHz
- ✓ Mémoire : 2 GO RAM

5.1.2 Environnement logiciel

L'environnement logiciel consiste les composants suivants :

a) Le langage Python

Python est un langage de programmation de haut niveau conçu pour être facile à lire et simple à mettre en œuvre. Il est open source, ce qui signifie qu'il est libre d'utilisation, même pour des applications commerciales. Python peut fonctionner sur les systèmes Mac, Windows et Unix et a également été porté sur des machines virtuelles Java et .NET.

Python est considéré comme un langage de script, comme Ruby ou Perl et est souvent utilisé pour créer des applications Web et du contenu Web dynamique. Il est également pris en charge par un certain nombre de programmes d'imagerie 2D et 3D, permettant aux utilisateurs de créer des plug-ins et des extensions personnalisés avec Python.

Les scripts écrits en Python (fichiers.PY) peuvent être analysés et exécutés immédiatement. Ils peuvent également être enregistrés en tant que programmes compilés (fichiers .PYC), qui sont souvent utilisés comme modules de programmation pouvant être référencés par d'autres programmes Python.[22]

b) Visual Studio Code

Visual studio code ou VS Code est un éditeur de code développé par Microsoft en 2015.Léger mais puissant qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (tels que C++, C#, Java, Python, PHP, Go) et des runtimes (tels que .NET et Unity).

5.2 Présentation de l'application

Comme nous l'avons mentionné précédemment, notre système a besoin d'extraire trois éléments (Titre, mots clés, section notre approche) de l'article scientifique pour les utiliser dans le processus de résumé.

Pour mieux présenter cette étape, nous prenons un article scientifique et extrayons ces éléments (l'extraction automatiquement par notre système), le résultat est ci-dessous :

Le titre de l'article :

```
le titre de l'article:
 Language-independent extractive automatic text summarization based on automatic keyword extraction
```

Figure 4.3 : Le titre de l'article avant le prétraitement

Les mots clé de l'article :

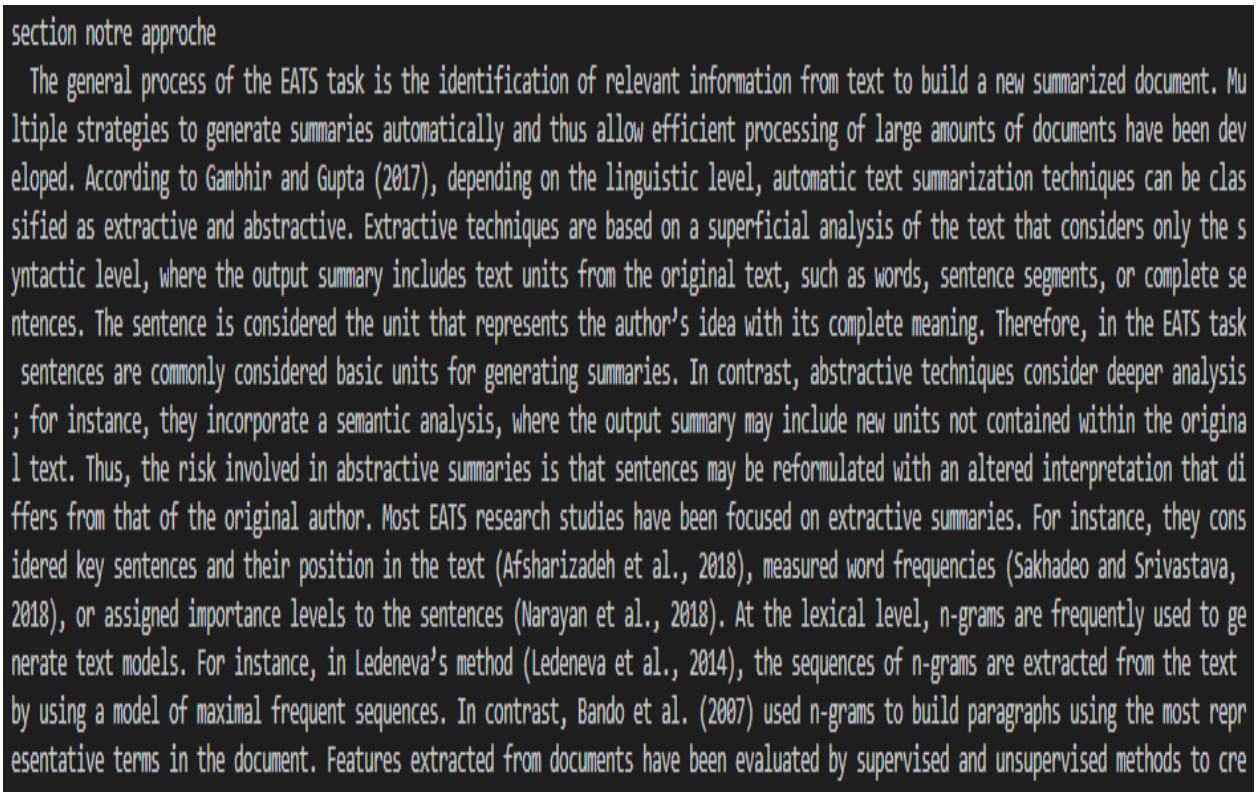
```
les mots clé de l'article:
 Automatic summarization,Genetic algorithm,Topic modeling,Extractive summaries,Keywords
```

Figure 4.4 : Les mots clé de l'article avant le prétraitement

La section notre approche :

L'extraction de cette partie est passée par 3 étapes :

- **La première étape** : extraire la table des matières, qui contient les titres des paragraphes, à l'aide de la bibliothèque ***re (Regular expression)***.
- **La deuxième étape :** Extraire tous les paragraphes avec leurs titres contenus dans l'article en utilisant la même bibliothèque.
- **La troisième étape :** en utilisant la table des matières qui a été extraite lors de la première étape, nous supprimons les paragraphes ou les parties dont nous n'avons pas besoin dans le résumé, tels que : Abstract, Introduction, Conclusion et References…. etc. nous ne gardons que les parties importantes qui contiennent la contribution de l'article, à la fin, ces parties sont fusionnées pour faire la section notre approche.
- Ceci juste une partie :


section notre approche

The general process of the EATS task is the identification of relevant information from text to build a new summarized document. Multiple strategies to generate summaries automatically and thus allow efficient processing of large amounts of documents have been developed. According to Gambhir and Gupta (2017), depending on the linguistic level, automatic text summarization techniques can be classified as extractive and abstractive. Extractive techniques are based on a superficial analysis of the text that considers only the syntactic level, where the output summary includes text units from the original text, such as words, sentence segments, or complete sentences. The sentence is considered the unit that represents the author's idea with its complete meaning. Therefore, in the EATS task sentences are commonly considered basic units for generating summaries. In contrast, abstractive techniques consider deeper analysis; for instance, they incorporate a semantic analysis, where the output summary may include new units not contained within the original text. Thus, the risk involved in abstractive summaries is that sentences may be reformulated with an altered interpretation that differs from that of the original author. Most EATS research studies have been focused on extractive summaries. For instance, they considered key sentences and their position in the text (Afsharizadeh et al., 2018), measured word frequencies (Sakhadeo and Srivastava, 2018), or assigned importance levels to the sentences (Narayan et al., 2018). At the lexical level, n-grams are frequently used to generate text models. For instance, in Ledeneva's method (Ledeneva et al., 2014), the sequences of n-grams are extracted from the text by using a model of maximal frequent sequences. In contrast, Bando et al. (2007) used n-grams to build paragraphs using the most representative terms in the document. Features extracted from documents have been evaluated by supervised and unsupervised methods to cre

Figure 4.5 : La section notre approche avant le prétraitement

5.2.1 Le fonctionnement de l'application en arrière-plan

Le prétraitement de titre de l'article :

```
tokenized_titre:
 ['Language-independent', 'extractive', 'automatic', 'text', 'summarization', 'based', 'on', 'automatic', 'keyword', 'extraction']
token_titre_no_sw:
 ['Language-independent', 'extractive', 'automatic', 'text', 'summarization', 'based', 'automatic', 'keyword', 'extraction']
title_stem:
 ['language-independ', 'extract', 'automat', 'text', 'summar', 'base', 'automat', 'keyword', 'extract']
titre après le prétraitement:
 language-independ extract automat text summar base automat keyword extract
```

Figure 4.6 : Le prétraitement de titre

Le prétraitement des mots clé de l'article :

```
keywords après le prétraitement:
  automat summar genet algorithm topic model extract summari keyword
```

Figure 4.7 : Le prétraitement des mots clé

Le prétraitement de la section notre approche :

On va appliquer les processus suivant sur la section notre approche :

- Segmenté en phrases
- Supprimer les signes de ponctuation
- Supprimer les mots vides (stops words)
- Radicalisation des mots (stemming)

Après avoir appliqué ces processus, nous obtenons ce qui suit (imprimé juste 10 phrases) :

```
les phrases après le Prétritement:
 ['the gener process eat task identif relev inform text build new summar document', 'multipl strategi gener summari automat thu allow effici process la
rg amount document develop', 'accord gambhir gupta 2017 depend linguist level automat text summar techniqu classifi extract abstract', 'extract techniq
u base superfici analysi text consid syntact level output summari includ text unit origin text word sentenc segment complet sentenc', 'the sentenc cons
id unit repres author idea complet mean', 'therefor eat task sentenc commonli consid basic unit gener summari', 'in contrast abstract techniqu consid d
eeper analysi instanc incorpor semant analysi output summari may includ new unit contain within origin text', 'thu risk involv abstract summari sentenc
 may reformul alter interpret differ origin author', 'most eat research studi focus extract summari', 'for instanc consid key sentenc posit text afshar
izadeh et al 2018 measur word frequenc sakhadeo srivastava 2018 assign import level sentenc narayan et al 2018']
```

Figure 4.8: Le prétraitement de la section notre approche

Après le prétraitement des sections nécessaires on va passer à l'étape de Scoring des phrases.

- Similarité avec le titre (Score 1) :

```
Similarité des phrases avec le titre:
 [0.12382973858368108, 0.12810445583249414, 0.3135282960135306, 0.21661996274443387, 0.0, 0.0, 0.03237441047072505, 0.0, 0.1626546880
821312, 0.029137887960935055, 0.06426466412278663, 0.12149102855669217, 0.0, 0.0, 0.10185922401909303, 0.08939535561421565, 0.0, 0.0,
 0.0, 0.15079301258847666, 0.0, 0.0, 0.06441621546514968, 0.0, 0.0, 0.04536738790212239, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.063157975608
85943, 0.0, 0.0, 0.0, 0.19908628462583428, 0.047069721715738116, 0.0, 0.0, 0.15055571802796058, 0.0, 0.0, 0.0, 0.0, 0.030833595253886
81, 0.0, 0.16593461218140151, 0.06308425167401356, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.04076644124793034, 0.0, 0.0, 0
.03465968418487781, 0.0, 0.0, 0.0, 0.0, 0.0, 0.037522510410429595, 0.1592938287209523, 0.0, 0.0, 0.0, 0.0, 0.055381678951888745, 0.0,
 0.042654624039366734, 0.0, 0.08153632964491479, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

Figure 4.9 : Similarité avec le titre

- Similarité avec les mots clé (Score 2) :

```
Similarité des phrases avec les mots clé:
 [0.0894573213066921 5, 0.1308322606143053 6, 0.2127383513185617 2, 0.0862889183396659, 0.0, 0.0605393719989655 7, 0.0349910732949752 6, 0.0470979965548565 6
, 0.1602188191343703, 0.0, 0.062783750657963, 0.09266874661790547, 0.0, 0.0, 0.11320970596858307, 0.09915776096681453, 0.03801870167993543 6, 0.0, 0.118
37987995462843, 0.13351609203381554, 0.0, 0.036890352634943814, 0.128634389828833 1, 0.0, 0.0, 0.0590097716527016 2, 0.078184170237557 7, 0.0586283683435 7
197, 0.0, 0.0, 0.0, 0.0, 0.0822495075656120 2, 0.0, 0.0, 0.032006130319796426, 0.16579608281259417, 0.05021879074193711, 0.07375184458727925, 0.04846209
407630991, 0.12689220351885572, 0.0, 0.0, 0.06727072121019713, 0.06805284351121325, 0.033668992782715686, 0.0, 0.1836284565522201 8, 0.0, 0.0, 0.0, 0.14
01092605701261 3, 0.0811055900675699 4, 0.0933363883717340 5, 0.0647223522644023 3, 0.0, 0.0, 0.0, 0.0, 0.075153921556397 4, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0845010213408442, 0.0966757072186618 5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.04654005420475132 4, 0.1462875
5977457675, 0.0, 0.08960922534737956, 0.10102518700870793, 0.0, 0.0, 0.0, 0.0, 0.14709145300507728, 0.0, 0.0, 0.17218275536934 5, 0.1066749173560342 4, 0
.11981860793740225, 0.16809465563253817, 0.0, 0.0]
```

Figure 4.10 : Similarité avec les mots clé

- L’existence des cue phrases (Score 3) :

```
Le Score de cue words:
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0]
```

Figure 4.11 : L’existence des cue phrases

- Scores de position de la phrase (Score 4) :

```
Score de position de la phrases :
 [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0]
```

Figure 4.12 : Scores de position de la phrase

- La somme des scores :

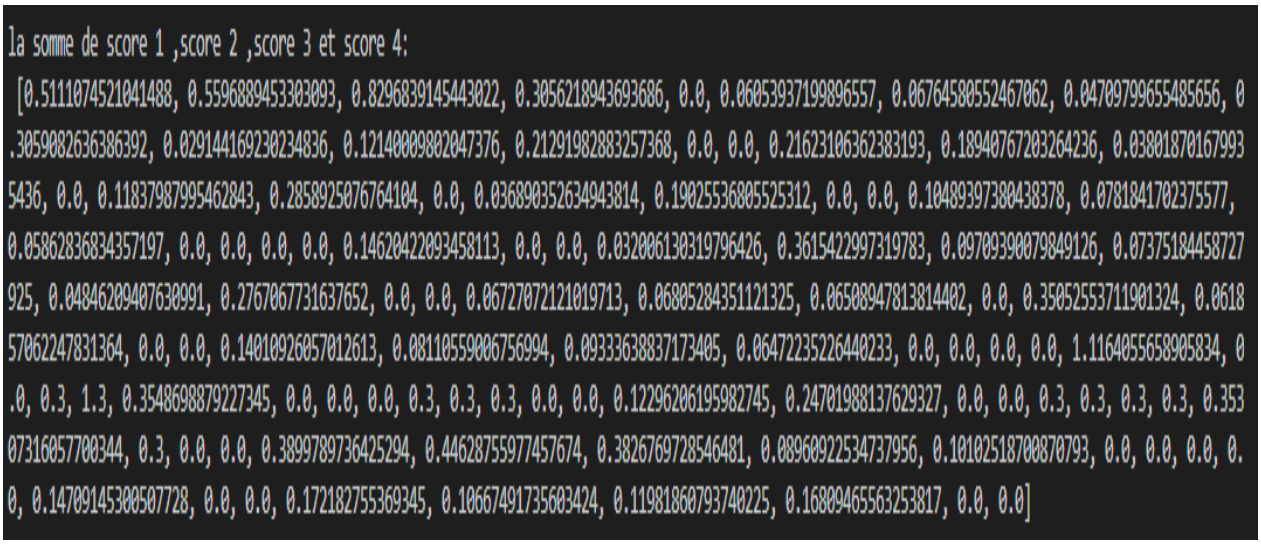


Figure 4.13 : La somme des scores

- Chaque phrase avec son propre score final et sa position (imprimé seulement 10 phrases):

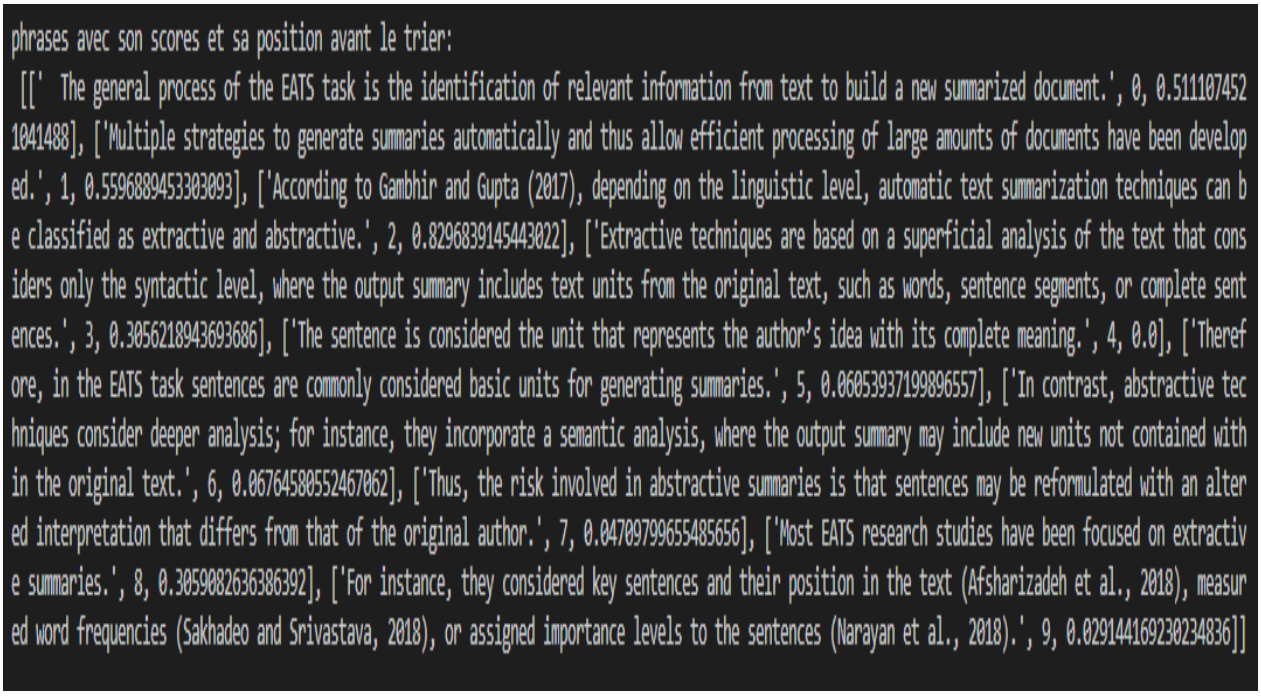


Figure 4.14 : Les phrases avec son propre score final et sa position

Après le Scoring des phrases on va passer à l'étape de sélectionner les phrases qui composent le résumé.

- Premièrement, nous allons classer les phrases par ordre décroissant en utilisant le score comme suit :

```
phrases avec son scores et sa position après le trier:
 [['In this section, we describe the basic concept.', 62, 1.3], ['Therefore, in this paper an evolutionary clustering scheme based on a generative mode
l (LDA) and a context-based model (Doc2vec) that provides substantial information about the latent semantic links among words are proposed.', 59, 1.116
4055658905834], ['According to Gambhir and Gupta (2017), depending on the linguistic level, automatic text summarization techniques can be classified a
s extractive and abstractive.', 2, 0.8296839145443022], ['Multiple strategies to generate summaries automatically and thus allow efficient processing o
f large amounts of documents have been developed.', 1, 0.5596889453303093], ['  The general process of the EATS task is the identification of relevant
information from text to build a new summarized document.', 0, 0.5111074521041488], ['It represents documents as a mixture of different topics, where e
ach topic consists of a set of words that have a link between them.', 85, 0.44628755977457674], ['LDA (Blei et al., 2003) is a probabilistic generative
 model for discrete data collections, such as text collections.', 84, 0.3899789736425294], ['Words, in turn, are chosen based on probability.', 86, 0.3
826769728546481], ['In their study, Soto and García-Hernández (2009) developed an automatic summarization system that uses unsupervised learning.', 36,
 0.3615422997319783], ['Because the proposed approach is based on a clustering scheme, the methods for building the vectorial space are described in Se
ction 3.1.', 63, 0.3548698879227345]]
```

Figure 4.15 : Les phrases avec son score final trié

- Deuxièmement, nous choisirons les phrases les plus importantes qui composent le résumé, dans cet exemple nous avons choisi les 10 meilleures phrases comme suit :

```
Le résumé:
  The general process of the EATS task is the identification of relevant information from text to build a new summarized document. Multiple strategies
 to generate summaries automatically and thus allow efficient processing of large amounts of documents have been developed. According to Gambhir and Gu
pta (2017), depending on the linguistic level, automatic text summarization techniques can be classified as extractive and abstractive. In their study,
 Soto and García-Hernández (2009) developed an automatic summarization system that uses unsupervised learning. Therefore, in this paper an evolutionary
 clustering scheme based on a generative model (LDA) and a context-based model (Doc2vec) that provides substantial information about the latent semanti
c links among words are proposed. In this section, we describe the basic concept. Because the proposed approach is based on a clustering scheme, the me
thods for building the vectorial space are described in Section 3.1. LDA (Blei et al., 2003) is a probabilistic generative model for discrete data coll
ections, such as text collections. It represents documents as a mixture of different topics, where each topic consists of a set of words that have a li
nk between them. Words, in turn, are chosen based on probability.
```

Figure 4.16 : Le résumé

- Dans le résumé résultats les phrases sont triées selon leurs ordres dans l'article.
- Après la génération de résumé on va évaluer ce résumé on utilise les métriques de ROUGE

```
Evauation
     rouge-1   rouge-2   rouge-3   rouge-l  rouge-w-1.2  rouge-s4  rouge-su4
r  0.441176  0.079208  0.030000  0.343137     0.124891  0.082828   0.144295
p  0.236842  0.042328  0.015957  0.184211     0.111756  0.043850   0.076512
f  0.308219  0.055172  0.020833  0.239726     0.117959  0.057343   0.100000
```

Figure 4.17 : L'évaluation de résumé

Tout ce qui a été présenté jusqu'à présent s'exécute en arrière-plan de l'application.

5.2.2 L'interface de l'application

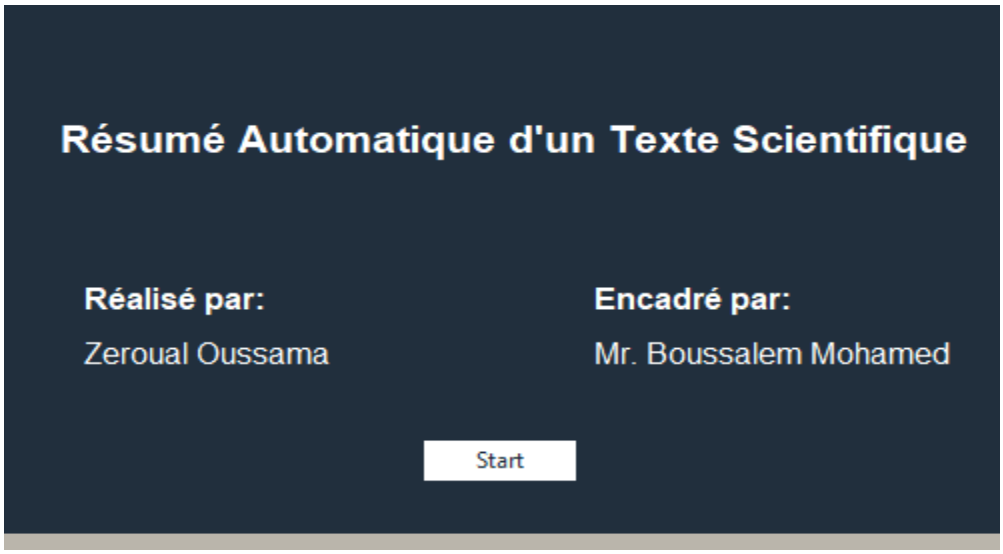


Figure 4.18 : La page d'accueil de l'application

L'application est divisée en deux parties principales :

- La première partie : l'utilisateur doit sélectionner l'article à résumer, déterminer le nombre de phrases de résumé, et cliquer sur le bouton "RESUME" pour générer automatiquement le résumé de cet article scientifique.

Après avoir résumé l'article l'application affiche le nombre des phrases du résumé et ainsi le nombre des mots.

Figure 4.19 : L'interface de résumé

- La partie deuxième contient l'évaluation d'un résumé d'article scientifique, par les mesures du ROUGE.

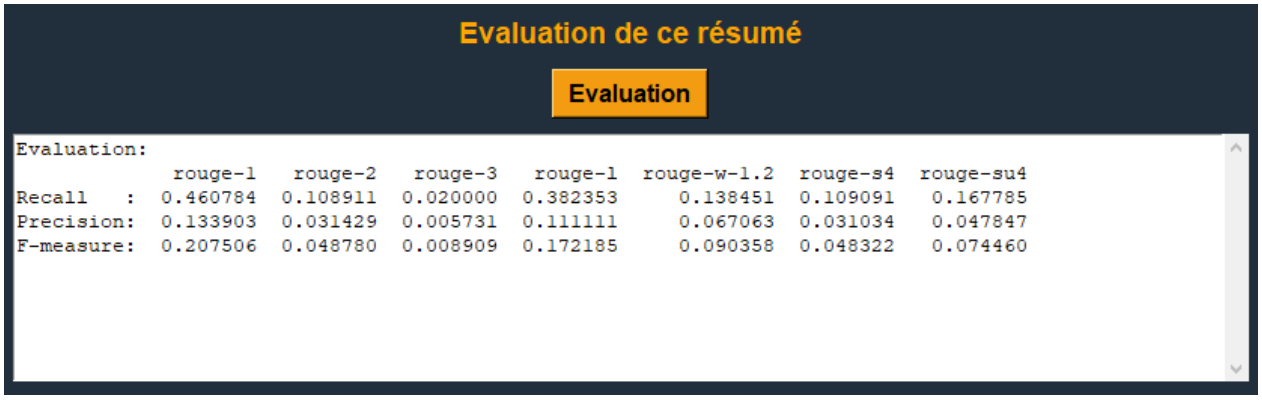


Figure 4.20 : L'interface de l'évaluation de résumé

- ***Copier le résumé*** est une option ajoutée à l'application pour permettre à l'utilisateur de copier le résumé pour l'utiliser dans d'autres tâches. Comme indiqué dans la figure suivante.

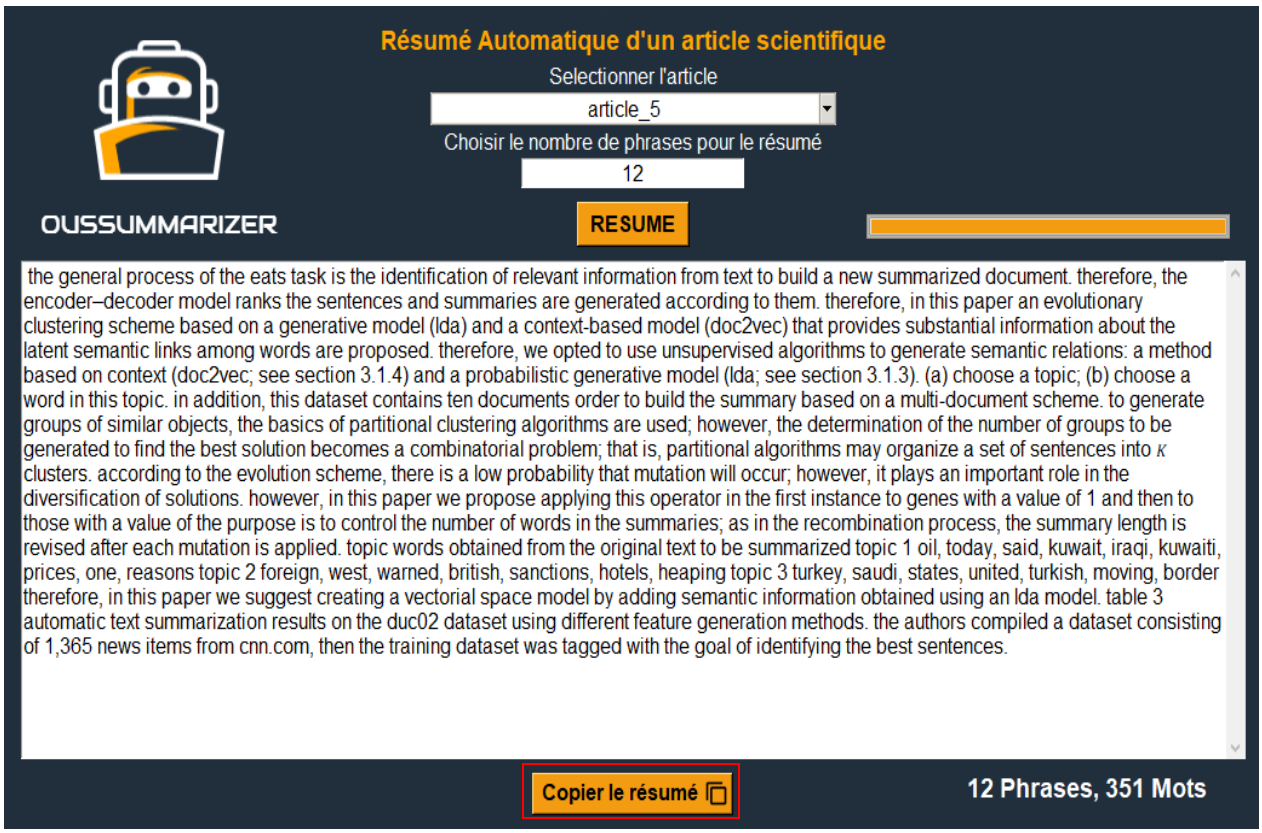


Figure 4.21 : L'option de copier le résumé généré

- ***Télécharger le résumé*** : l'application permet à l'utilisateur de télécharger le résumé sous forme de PDF bien organisé.

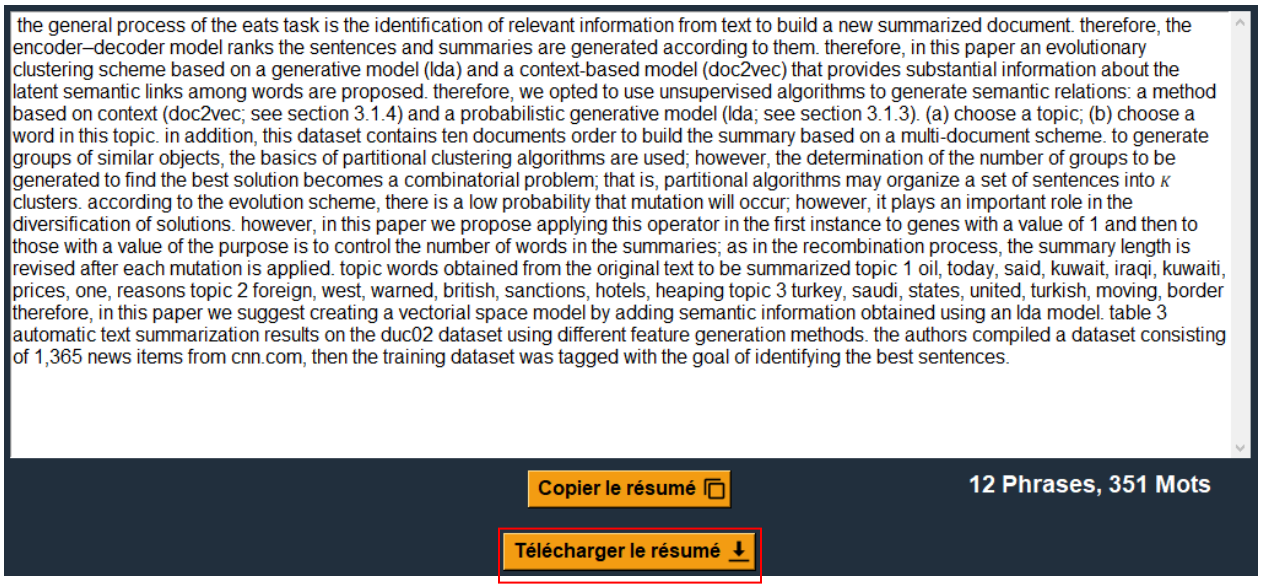
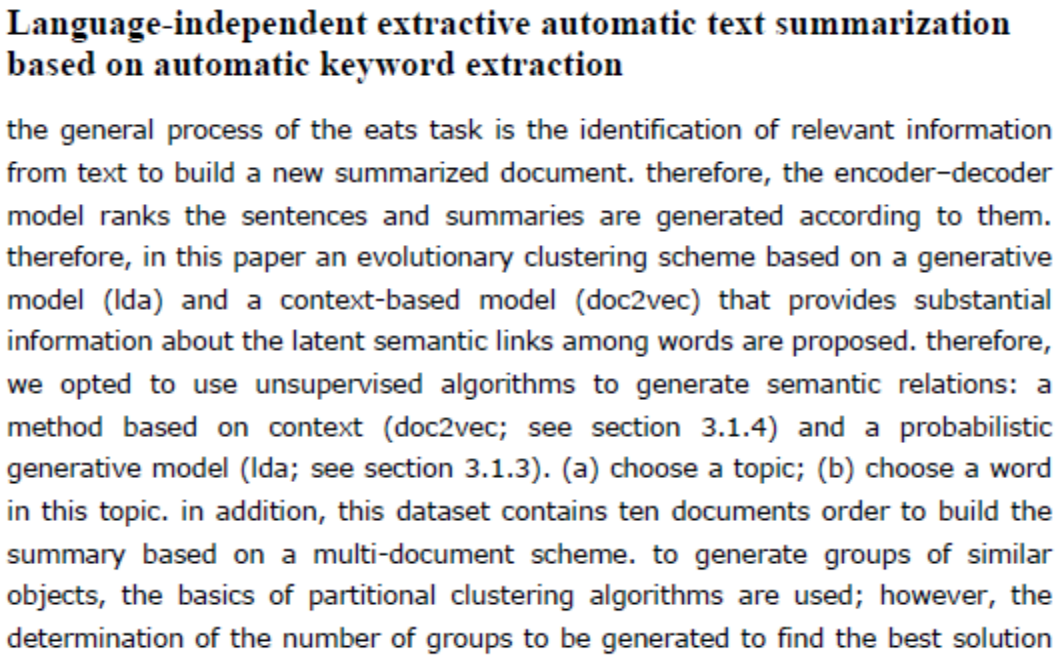


Figure 4.22 : L'option de télécharger le résumé généré

- La figure suivante affiche le format du résumé dans le fichier PDF après le téléchargement



Language-independent extractive automatic text summarization based on automatic keyword extraction

the general process of the eats task is the identification of relevant information from text to build a new summarized document. therefore, the encoder–decoder model ranks the sentences and summaries are generated according to them. therefore, in this paper an evolutionary clustering scheme based on a generative model (lda) and a context-based model (doc2vec) that provides substantial information about the latent semantic links among words are proposed. therefore, we opted to use unsupervised algorithms to generate semantic relations: a method based on context (doc2vec; see section 3.1.4) and a probabilistic generative model (lda; see section 3.1.3). (a) choose a topic; (b) choose a word in this topic. in addition, this dataset contains ten documents order to build the summary based on a multi-document scheme. to generate groups of similar objects, the basics of partitional clustering algorithms are used; however, the determination of the number of groups to be generated to find the best solution

Figure 4.23 : Le format du résumé dans le fichier PDF après le téléchargement

5.3 Evaluation de notre système

Nous avons utilisé le corpus mentionné précédemment, afin d'évaluer notre système et les autres systèmes, en calculant la mesure de Rouge qui est basé sur le calcul des mesures suivantes : Rappel, Précision et F-score.

- Le tableau suivant représente les résultats obtenus par notre système :

	rouge-1	**rouge-2**	**rouge-3**	**rouge-l**	**rouge-w-1**	**rouge-s4**	**rouge-su4**
Rappel	0.508655	**0.126469**	**0.038510**	0.452042	0.450215	0.128033	0.193515
Précision	**0.108259**	**0.026552**	**0.008160**	**0.096364**	**0.095360**	**0.026422**	**0.040108**
F-score	**0.178522**	**0.043890**	**0.013466**	**0.158863**	**0.157384**	**0.043804**	**0.066444**

Tableau 4.1 : Résultats d'évaluation de notre système

- Le tableau suivant représente les résultats obtenus par le système LSA :

	rouge-1	**rouge-2**	**rouge-3**	**rouge-l**	**rouge-w-1**	**rouge-s4**	**rouge-su4**
Rappel	**0.533622**	0.114642	0.033985	**0.476625**	**0.473775**	0.117793	0.189237
Précision	0.096435	0.020477	0.005855	0.086092	0.085201	0.020566	0.033236
F-score	0.163349	0.034747	0.009990	0.145840	0.144428	0.035018	0.056542

Tableau 4.2 : Résultats d'évaluation de système LSA

- Le tableau suivant représente les résultats obtenus par le système TextRank :

	rouge-1	**rouge-2**	**rouge-3**	**rouge-l**	**rouge-w-1**	**rouge-s4**	**rouge-su4**
Rappel	0.519548	0.121202	0.034728	0.475260	0.473589	**0.129205**	**0.196361**
Précision	0.073451	0.017407	0.005106	0.067178	0.066603	0.017943	0.027212
F-score	0.128706	0.030442	0.008902	0.117717	0.116782	0.031511	0.047800

Tableau 4.3 : Résultats d'évaluation de système TextRank

- Le tableau suivant représente les valeurs de F-score de chaque système :

	rouge-1	**rouge-2**	**rouge-3**	**rouge-l**	**rouge-w-1**	**rouge-s4**	**rouge-su4**
Notre système	**0.178522**	**0.043890**	**0.013466**	**0.158863**	**0.157384**	**0.043804**	**0.066444**
Système LSA	0.163349	0.034747	0.009990	0.145840	0.144428	0.035018	0.056542
Système TextRank	0.128706	0.030442	0.008902	0.117717	0.116782	0.031511	0.047800

Tableau 4.4 : les valeurs de F-score de chaque système

Discussion des résultats :

Les résultats obtenus dans les tableaux ci-dessus montrent que notre système proposé peut obtenir des scores rouges plus élevés, c'est-à-dire un meilleur contenu de résumé par rapport les deux autres systèmes « *LSA et TextRank* ».

Où nous avons remarqué que les mesures de notre système sont toujours mieux que celles des systèmes « *LSA et TextRank* » dans La plupart des variantes ROUGE.

Ces résultats nous encouragent bien à travailler et à améliorer notre système.

6. Conclusion

Dans ce chapitre, nous avons présenté notre système de résumé automatique d'un texte scientifique. En premier lieu, nous avons présenté l'architecture globale de notre système, en spécifiant les différents modules, ainsi que la description de chaque module. Ensuite, nous avons déterminé le corpus qu'on va utiliser dans la phase d'évaluation.

Nous avons présenté dans la partie de l'implémentation : l'environnement de travail et l'interface de notre application et comment travailler chaque composant, l'explication des processus qui se déroulent en arrière-plan (de l'application) est faite. Finalement, nous avons expliqué l'évaluation notre système en utilisant le package Rouge et discuté les résultats.

Conclusion Générale

Le concept de résumé automatique devient l'un des principaux thèmes du traitement automatique du langage naturel. Au lieu de diffuser des documents entiers, il est préférable de ne diffuser que des résumés qui contiendraient les informations vraiment pertinentes. Il est plus facile de lire quelques lignes ou quelques pages pour s'apercevoir que ce document est pertinent ou pas pour le lecteur. Un document textuel doit maintenant être géré en même temps que son résumé, ce qui sera aussi un des moyens d'accéder au contenu du document.

L'article scientifique est un type particulier des documents textuels, sa particularité réside dans sa structure et son contenu. On trouve des articles qui présentent des « Survey » sur des axes de recherches particuliers ou des articles de conférences ou de journaux pour publier des nouvelles contributions scientifiques.

L'objectif de notre travail est de produire un résumé automatique à partir d'un article scientifique où le résultat est un ensemble de phrases décrivant la contribution de cet article, bien sur le résumé doit avoir un peu plus de détaille par rapport à l'abstract de l'article.

L'approche proposée était basée sur la division du processus de résumé automatique d'un article scientifique, en deux étapes, et les résultats de la première étape servent d'entrées à la deuxième étape.

Dans la première étape nous avons besoin exploité la structure particulière de l'article scientifique pour extraire les sections nécessaires pour résumer un article scientifique, et de prétraiter ces sections. Puis dans la deuxième étape nous allons utiliser une combinaison des méthodes de résumé pour générer un résumé à partir des sorties de la première étape.

Enfin, nous avons utilisé le corpus cmp-lg (*Computation and Language*) (fournit par SUMMAC) composé de 183 articles scientifiques sous forme xml. Nous avons utilisé leurs abstracts comme des résumés de références. La version 1.0.1 du package ROUGE est utilisé pour calculer les métriques Rouge afin de comparer les métriques de notre système avec celles d'autres systèmes. Les résultats sont prouvés que l'approche proposée fournit une bonne qualité des résumés d'articles scientifiques.

Dans notre travail nous avons utilisé seulement la méthode extractive, où le résumé résultat est un ensemble de phrases extraits de l'article à résumé, ces derniers sont sélectionnés selon leurs

scores. Comme perspective de ce travail, nous voulons combiner la méthode d'extraction avec la méthode d'abstraction pour résumer un article scientifique.

Bibliographiques

[1] T. E. Doszkocs, “Natural Language Processing in Information Retrieval.”

[2] “Natural Language Processing and Machine Learning.” https://www.encora.com/insights/natural-language-processing-and-machine-learning (accessed Apr. 10, 2022).

[3] S. Vajjala, B. Majumder, A. Gupta, and H. Surana, “Practical Natural Language Processing A Comprehensive Guide to Building Real-World NLP Systems.”

[4] “9 Powerful Ways to Use NLP to Improve Customer Service.” https://www.nextiva.com/blog/nlp-in-customer-service.html (accessed Apr. 10, 2022).

[5] R. Lacson and R. Khorasani, “Natural language processing: The basics (Part 1),” *Journal of the American College of Radiology*, vol. 8, no. 6, pp. 436–437, 2011, doi: 10.1016/j.jacr.2011.04.020.

[6] W. Sriyanong, N. Moungmingsuk, and N. Khamphakdee, “A Text Preprocessing Framework for Text Mining on Big Data Infrastructure,” in *2018 2nd International Conference on Imaging, Signal Processing and Communication (ICISPC)*, 2018, pp. 169–173. doi: 10.1109/ICISPC44900.2018.9006718.

[7] J.-M. Torres-Moreno, “Automatic Text Summarization,” 2011.

[8] K. S. Jones, “Automatic summarising: factors and directions,” MIT Press, 1999.

[9] A. Blais, “Résumé automatique de textes scientifiques et construction de fiches de synthèse catégorisées : Approche linguistique par annotations sémantiques et réalisation informatique,” 2008.

[10] M. Mnasri, “Résumé Automatique Multi-Document Dynamique,” 2015.

[11] F. Boudin, “Exploration d’approches statistiques pour le résumé automatique de texte,” 2008. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00419469

[12] M. F. Fakhrezi, M. A. Bijaksana, and A. F. Huda, “Implementation of Automatic Text Summarization with TextRank Method in the Development of Al-Qur’an Vocabulary Encyclopedia,” *Procedia Computer Science*, vol. 179, pp. 391–398, Jan. 2021, doi: 10.1016/J.PROCS.2021.01.021.

[13] Douzidia Fouad Soufiane, “Résumé automatique de Arabe memoire,” 2004.

[14] R. Mihalcea and P. Tarau, “TextRank: Bringing Order into Texts.”

[15] M. G. Ozsoy, F. N. Alpaslan, and I. Cicekli, “Text summarization using latent semantic analysis,” *Journal of Information Science*, vol. 37, no. 4, pp. 405–417, Aug. 2011, doi: 10.1177/0165551511408848.

[16] N. Ibrahim Altmami and M. el Bachir Menai, “Automatic summarization of scientific articles: A survey,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4. King Saud bin Abdulaziz University, pp. 1011–1028, Apr. 01, 2022. doi: 10.1016/j.jksuci.2020.04.020.

[17] “KAREN SPARCK JONES 1995,” 1995.

[18] J. Steinberger and K. Ježek, “EVALUATION MEASURES FOR TEXT SUMMARIZATION,” 2009.

[19] M. El-Haj, U. Kruschwitz, and C. Fox, “Exploring clustering for multi-document Arabic summarisation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 7097 LNCS, pp. 550–561. doi: 10.1007/978-3-642-25631-8_50.

[20] M. H. Maaloul and M. Hedi, “Approche hybride pour le résumé automatique de textes. Application à la langue arabe.,” 2017. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00756111v3

[21] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” 2004.

[22] Per. Christensson, “Python Definition,” *TechTerms. Sharpened Productions*, Jun. 15, 2010.