République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahleb Blida - Blida 1

Faculté des Sciences

Département d'Informatique

La Génération d'un Résumé Extractif à partir d'un document

MEMOIRE DE MASTER

Option: Ingénierie des Logiciels

Réalisé par:

Ayoub ZEROUAL

Ibtissam BOULGHITI

Date : 04/10/2021

Membres de jury :

Mme. Nesrine LAHIANI	Président
Mr. Abdallah Hicham KAMECHE	Examinateur
Mme. Fatima BOUMAHDI	Promoteur
Mr. Hamza HENTABLI	Encadreur

Année : 2020-2021

Résumé

La forte augmentation des textes au format numérique met en évidence la nécessité de concevoir et de développer des outils de résumé efficaces pour localiser et extraire les informations pertinentes sous forme abrégée. Le résumé de texte dans le traitement du langage naturel est principalement traité par les méthodes d'extraction, qui consistent à sélectionner une partie du document original pour capturer l'idée principale du sujet. Contrairement à la méthode d'extraction, la méthode d'abstraction cela dépend de l'étude du sujet et de la formulation d'un résumé pour celui-ci. Dans notre travail, nous appliquons le type d'extraction.

Dans notre travail, nous avons utilisé Word2Vec qu'est un ensemble de modèles d'intégration lexicale de type Word Embedding. Word2Vec convertit le mot en un vecteur, dans notre cas un vecteur de 50 segments, ces segments sont des dérivés du mot. Nous avons également utilisé Auto-encoder et Multi-layer perceptron ensuite BLEU score pour l'aprentissage. Nous appliquons notre modèle sur l'ensemble de données Amazon Food_Reviews, et nous utilisons des métriques standard (telles que ROUGE) pour évaluer les paragraphes reconstruits.

- L'objectif de cette étude est d'Améliorer le résumé automatique extractif du texte à l'aide des nouvelles méthodologies.

Mots-clés: Résumé automatique, Extractive, Word Embidding, Word2Vec, Auto-Encoder, Multi-layer perceptron, BLEU score, Amazon Food_Reviews.

Abstract

The sharp increase of texts in digital format highlights the need to design and develop efficient summarization tools to locate and extract relevant information in abbreviated form. Text summarization in natural language processing is mainly handled by extraction methods, which consist in selecting a part of the original document to capture the main idea of the topic. Unlike the extraction method, the abstraction method depends on the study of the topic and the formulation of a summary for it. In our work, we apply the extraction type.

In our work, we have used Word2Vec which is a set of lexical integration models of the Word Embedding type. Word2Vec converts the word into a vector, in our case a vector of 50 segments; these segments are derivatives of the word. We also used Auto-encoder and Multi-layer perceptron then BLEU score for learning. We apply our model on the Amazon Food_Reviews dataset, and use standard metrics (such as ROUGE) to evaluate the reconstructed paragraphs.

- The objective of this study is to improve automatic extractive text summarization using new methodologies.

Keywords: automatic summarization, Extractive, Word Embidding, Word2Vec, Auto-Encoder, Multi-layer perceptron, BLUE score, Amazon Food_Reviews.

الملخص

تبرز الزيادة الحادة للنصوص في التنسيق الرقمي الحاجة إلى تصميم وتطوير أدوات تلخيص فعالة لتحديد واستخراج المعلومات ذات الصلة في شكل مختصر. يتم التعامل مع تلخيص النص في معالجة اللغة الطبيعية بشكل أساسي من خلال طرق الاستخراج، والتي تعتمد على تحديد اجزاء من المستند الأصلي لالتقاط الفكرة الرئيسية للموضوع. على عكس طريقة الاستخراج، تعتمد طريقة التجريد على دراسة الموضوع وصياغة ملخص له. في عملنا هذا نطبق طريقة الاستخراج.

بحيث استخدمنا Word2Vec وهي مجموعة من نماذج التكامل المعجمي Word Embedding. Word2Vec يحول الكلمة إلى شعاع، في نموذجنا هذا يحول الكلمة الى شعاع من 50 مقطعًا ، و هذه الأجزاء عبارة عن مشتقات للكلمة. استخدمنا أيضًا التشفير التلقائي Auto-Encodeur و الإدراك متعدد الطبقات Multi-layer perceptron ثم BLUE score من أجل التدريب. قمنا بتدريب نموذجنا على مجموعة بياناتAmazon Food_Reviews و استخدمنا مقاييس لتقييم الفقرات التي أعيد بناؤها

(مثل ROUGE ، BLEU-Score).

- الهدف من هذه الدراسة هو تحسين الملخص التلقائي الاستخراجي للنص باستخدام تقنيات جديدة.

الكلمات المفتاحية: التلخيص التلقائي ، الاستخراجي ، تضمين الكلمات Word Embidding ، Word2Vec، التشفير التلقائي ، الإدراك متعدد الطبقات Multi-layer perceptron ، BLUE score، Amazon Food_Reviews.

Remerciements

Tout d'abord, nous remercions Dieu de nous avoir donné l'opportunité de terminer cet humble travail. Nous aimerions également profiter cette occasion pour exprimer notre gratitude à notre estimé promotrice, Dr Fatima BOUMAHDI, pour nous avoir donné l'opportunité de travailler avec elle, et la remercier pour la quantité généreuse de conseils et de soutien fournis au cours de cette recherche. Nous lui sommes redevable pour son aide inestimable, sa patience à chaque étape de ce travail, elle a été et sera toujours une inspiration et une motivation pour nous.

Nous tenons également à exprimer notre sincère gratitude à notre promoteur adjoint, Dr Hamza HENTABLI, pour son aide et son soutien continus, ainsi que pour sa patience et ses conseils jusqu'à la dernière minute de notre travail.

Remerciements particuliers aux membres du jury d'avoir accepté de revoir ce modeste mémoire.

Nous remercions le chef du département d'informatique, Dr Abdelkader GUESMIA, pour son aide et son soutien précieux.

Nous ne serions pas là où nous en sommes aujourd'hui sans l'énorme quantité de prières, d'encouragements et de soutien de nos familles, pour eux tous remerciements, respect et gratitude.

Nos sincères remerciements à tous ceux qui ont contribué de près ou de loin à ce travail.

Contenu

Liste des Figures

Liste des Figures

Liste des Tables

Introduction Générale

Contexte

Un résumé textuel est un extrait d'informations importantes du texte original, puis présenté à l'utilisateur sous forme de résumé. Donc le résumé est utile car il permet de gagner du temps pour extraire les informations importantes des documents volumineux. Auparavant, les gens lisaient des articles et des documents, puis comprenaient et écrivaient leur propre résumé, qui peut varier d'une personne à l'autre et prend beaucoup du temps. Dans la dernière période, avec l'augmentation de la publication sur Internet, et le nombre de ses utilisateurs, un grand nombre de documents électroniques sont disponibles en ligne où les utilisateurs ont du mal à trouver des informations pertinentes, ils se sentent très fatigués de lire une grande quantité de texte, donc ils peuvent sauter la lecture de nombreux documents importants. Pour cela, il y avait un besoin urgent d'un puissant système de résumé de texte, c'est le résumé automatique de texte, ce dernier est une solution pour préserver le contenu du texte principal, et aide l'utilisateur à comprendre et interpréter un grand volume de texte, ainsi réduire le temps d'utilisateur pour trouver les informations de base dans le document. De nos jours, Il existe plusieurs domaines nécessitent un résumé de texte automatique, tels que le résumé des articles de presse, le résumé des e-mails, les actualités SMS sur téléphone mobile, les résumés d'informations pour les hommes d'affaires et les fonctionnaires, etc.

Problématique

Le résumé automatique de texte consiste à produire un résumé Bref et lâche sans aucune intervention humaine tout en préservant le sens du texte original, afin d'aider le lecteur à déterminer si le document en question contient les informations qu'il recherche ou non. La production d'un résumé peut être une tâche difficile pour les humains, encore plus difficile pour les machines. Nous lisons généralement l'intégrité du texte pour développer notre compréhension avant d'écrire un résumé qui met en évidence ses idées importantes, et comme les ordinateurs manquent de connaissances humaines et de compréhension du langage, le résumé automatique de texte devient une tâche très complexe. La plupart des travaux dans le domaine du résumé automatique reposent sur l'approche

extractive qu'est l'extraction des parties du texte, et de phrases en général, et leur concaténation pour produire un résumé.

Alors que la deuxième approche est le résumé abstractif, qui est la production du résumé en étudiant le texte et en créant un nouveau texte plus court, dont certaines parties peuvent ne pas apparaître dans le document original et il ne suffit pas de produire des mots et des phrases qui capturent l'essence de la source, puisque le résumé doit être précis et compréhensible, Cette approche est plus difficile mais finalement c'est l'approche qui celle que les humains utilisent. Malgré l'existence des systèmes de résumé automatique, mais l'idée d'établir d'autres systèmes dans ce domaine est toujours valable afin d'atteindre des techniques qui produisent un résumé idéal et aussi proche que possible de résumé d'homme.

Objectifs

Dans notre projet, nous chercherons à créer un outil de résumé automatique de texte de type extractif utiliserons des mécanismes différents que les mécanismes utiliser dans les travaux précédents afin d'améliorer la qualité du résumé automatique. Où nous allons utiliser dans notre travail le Word Embedding exactement Word2Vec. Ensuite nous allons utiliser l'Auto-Encodeur pour encoder et décoder les informations et aussi, Multi-layer perceptron, BLEU score. De plus, nous entraînerons notre travail sur un grand ensemble de données Amazon Food_Reviews afin d'obtenir des bons résultats. Pour l'évaluation nous allons évaluer notre système avec la méthode ROUGE.

Organisation du mémoire

Pour faciliter la lecture de ce mémoire, nous présenterons brièvement les chapitres qui les composent. Notre travail est organisé comme suit:

- **Chapitre 1 : L'Etat de l'art :** dans ce chapitre, nous allons arborer le résumé automatique de façon générale. Nous allons presenter quelques définitions ainsi que les différentes approches et les différents types du résumé automatique. Ensuite, nous allons expliquer les étapes du résumé automatique ainsi que leurs méthodes et les méthodes d'évaluation. Enfin, nous allons présenter quelques travaux connexes dans le domaine de résumé automatique de texte.
- **Chapitre 2 : Apprentissage en profondeur :** Dans ce chapitre nous expliquons les concepts utilisé dans le cadre d'apprentissage en profondeur.

- **Chapitre 3 : La solution proposée :** Dans ce chapitre, nous allons parler sur les techniques que nous allons utiliser dans notre travail, ainsi que notre architecture proposée.
- **Chapitre 4 : Testes et Résultats :** Dans ce dernier chapitre, nous allons d'abord présenter l'environnement de développement, en définissons les différents outils utilisés, puis nous allons expliquer les différentes étapes de mise en œuvre et de test de notre système ainsi que les résultats obtenus.

Chapitre 1 État de l'art

1.1 Introduction

Un résumé de texte est : La reformulation du texte original de façon plus brève, respecter un nombre spécifié de mots, avec la nécessité de garder les informations de base. Le résumé automatique est un domaine de recherche depuis les années 1950[ARIES 2013], les chercheurs étudiés le développement des outils de résumé de texte, ils ont utilisé déférents méthodes pour but d'améliorer le résumé. En consequence, plusieurs façons de résumer texte automatique ont émergé. Dans ce chapitre, nous verrons ce qu'est un résumé automatique de texte. Aussi nous discuterons également de leurs approches, types, étapes, méthodes, évaluation et ensembles des données.

1.2 Définiton

Le résumé automatique de texte est une version condensée d'un document texte obtenu grâce à la technologie informatique. Le résumé est l'abréviation et la représentation exacte du contenu de document. Cependant, la production des résumés pertinents et de haute qualité nécessite au résumeur (système manuel ou automatisé) pour faire des efforts pour sélectionner, évaluer, organiser et combiner des segments des informations en fonction de leur pertinence. Afin de produire des synthèses automatisées fiables, la compréhension et la gestion de la redondance, de la cohérence et de la cohésion sont essentielles. [Moreno 2014]

Et d'après [Radev 2002] un résumé est défini comme "un texte produit d'un ou plusieurs textes, qui contient les informations importantes du (des) texte(s) original (originaux), et qui est moins que la moitié du (des) texte(s) original (originaux) et habituellement trop moins que çà".

Dans cette définition, nous pouvons voir trois aspects importants caractérisent le résumé automatique:

1. nous pouvons faire des résumés à partir d'un ou plusieurs documents dans le même domaine.
2. Le résumé doit contenir des informations importantes.
3. Le résumé doit être court.

1.3 Les approches de résumé automatique de texte

Afin de générer des résumés des textes automatiques, plusieurs méthodes et techniques peuvent être utilisées. Essentiellement, les deux principaux types de méthodes sont: méthode par extraction et méthode par abstraction. Récemment, une nouvelle méthode connue sous le nom de semi-extraction utilisant des techniques de compression, de fusion et de segmentation des phrases. Dans ce titre, nous présentons brièvement ces méthodes clarifie dans la figure suivante.

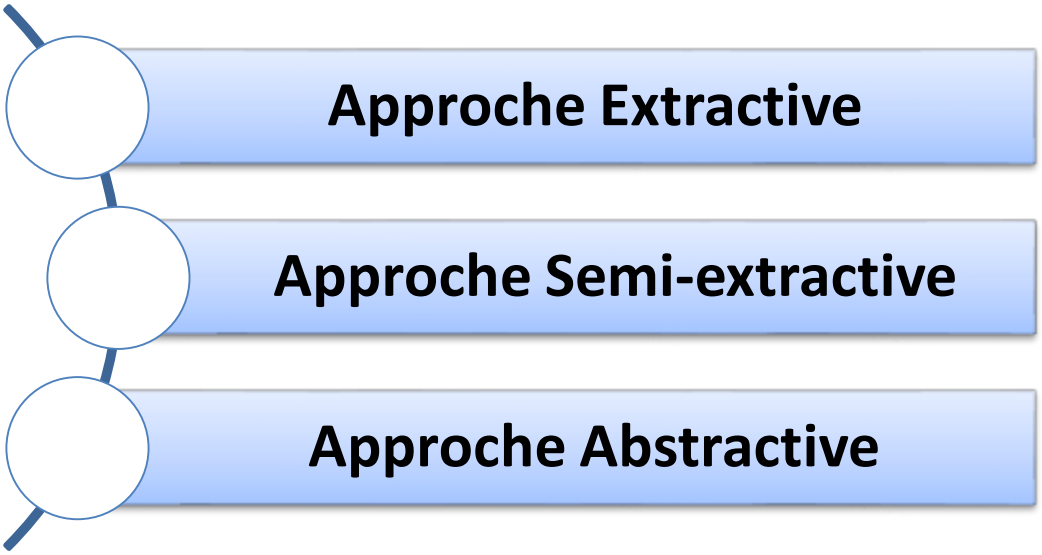


Figure 1-1 : Les approaches de résumé automatique de texte [Lamsiyah 2020]

1.3.1 Approche Extractive

La méthode d'extraction tente d'identifier et d'extraire les segments de texte les plus pertinents pour former un résumé. Généralement, le processus de génération d'un résumé comprend 4 étapes (voire la figure ci-dessous).PolyCom [ZHANG 2011] utilise cette méthode.

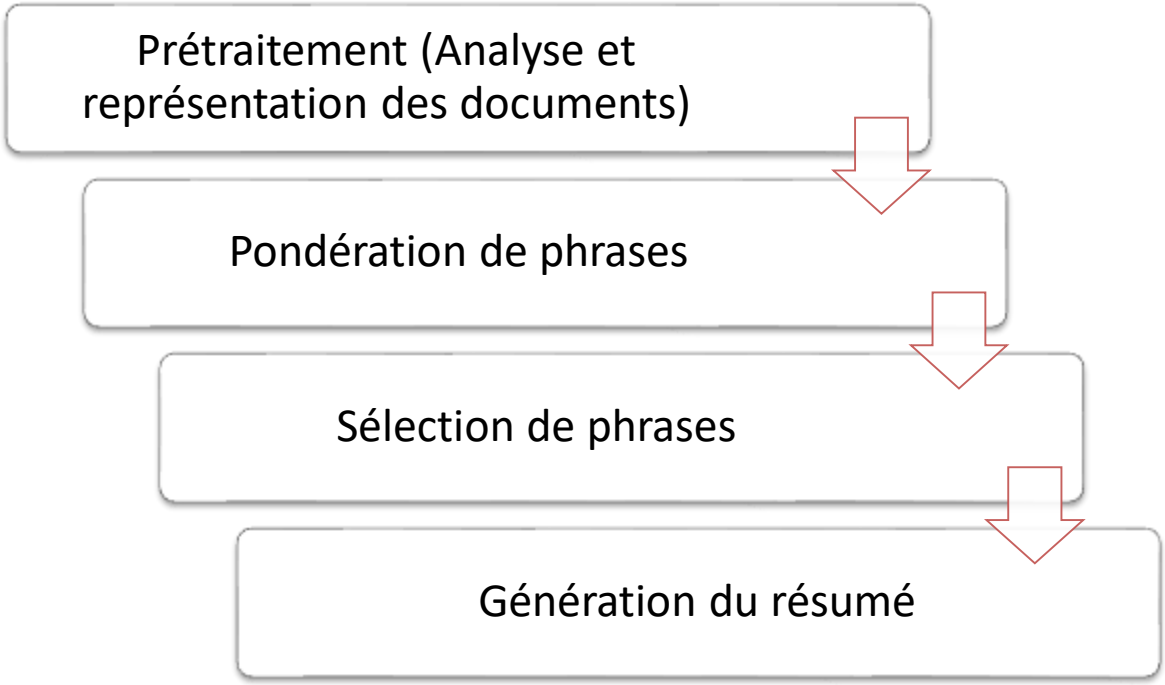


Figure1-2 : Les étapes de l'approche extractive [Lamsiyah 2020]

- Prétraitement (Analyse et représentation des documents) :

Le fichier source est dans formulaire non structuré; cette étape permet de prétraiter ces documents pour les représenter, le prétraitement de manière structurée nécessite généralement une technologie TALN[1], y compris la segmentation, la tokenisation, la lemmatisation /stemming, la reconnaissance terminé, le document source doit être représenté, et chaque document doit représenter par un vecteur afin que l'algorithme puisse l'utiliser. [Lamsiyah 2020]

- Pondération de phrases :

Cette étape est essentielle pour SRAT[2]par extraction. Sur la base des caractéristiques de représentation et de phrase déjà créées lors de la première étape, cette étape comprend l'attribution d'un score à chaque phrase indiquant sa pertinence. Plusieurs méthodes ont été développées pour cette tâche. Les méthodes les plus complètes de la littérature sont: méthodes statistiques, méthodes basées sur des grapheset, méthodes d'apprentissage automatique, méthodes utilisant des réseaux de neurones et d'autres méthodes récentes. [Lamsiyah 2020]

- Sélection des phrases :

[1] Traitement Automatique de Langage Natural
[2] Système de Résumé Automatique de Texte

Après avoir calculé le score de la phrase, nous sélectionnons la phrase avec le score le plus élevé pour générer un résumé. L'un des plus gros problèmes de cette étape est d'éviter la redondance, en particulier lors de la combinaison de plusieurs documents. A cet effet, plusieurs méthodes ont été introduites, comme MMR (Maximum Marginal Relevance) [CARBONELL 1998] et ILP (Integer Linear Programming). [OLIVEIRA 2016]

- Génération du résumé :

Habituellement, le système combinera les phrases affichées à l'étape précédente pour générer un résumé. [Lamsiyah 2020]

Ce type d'approche sera utilisé dans notre solution proposée pour savoir si cela donne de bons résultats en appliquant des nouvelles techniques

1.3.2 Approche Semi-Extractive

La compression, la fusion et la division de phrases sont des directions de recherche relativement nouvelles dans le résumé automatique de texte c'est ce qu'on appelle méthodes de semi-extraction. Ces tâches de traitement de phrases permettent de nombreuses améliorations, notamment la réduction de la fréquence et la production des bons résumés. Les méthodes de compression incluent la conversion des phrases pertinentes en phrases grammaticales plus courtes et leur conservation comme informations importantes.

La fusion phrase est une phrase simple et grammaticalement correcte générée à partir d'un ensemble de phrases apparentées, qui conserve les informations importantes de l'ensemble. Cette phrase n'est pas nécessairement incluse dans cet ensemble de phrases. La segmentation des phrases est une nouvelle méthode de semi-extraction proposée par [GENEST 2011] et aussi utilisé par [LI 2011]. Cette méthode consiste à rechercher d'abord un élément d'information (InIts), qui est défini comme la plus petite information cohérente dans une phrase ou un texte. Ensuite, la Sélection de ceux qui répondent aux besoins d'informations de l'utilisateur. Et finalement, un résumé contenant les InIts les plus pertinents est généré. [Lamsiyah 2020]

1.3.3 Approche Abstractive

Cette méthode, apparue à la fin des années 1970, s'inspirait principalement des domaines de l'intelligence artificielle et de la psychologie cognitive, cherchant à produire des résumés de qualité langagière comparable à ceux produits par l'homme. ABSUM [GENEST 2012] travail avec cette approche.Généralement, il existe trois techniques pour les méthodes abstractives [Lamsiyah 2020]:

(i) la technique de transformation des informations importantes contenues dans le document source en modèles cognitifs (tels que l'ontologie, les motifs, les graphiques);

(ii) la technique de séparation basée sur la représentation sémantique documentée ;

(iii) la technique d'utilisation des réseaux de neurones en apprentissage profond.

1.4 Les types de résumé automatique de texte

IL existe plusieurs types de résumé automatiques, dans ce titre on va distinguer les plus importants et couramment utilisé dans la littérature (voire la figure suivante).

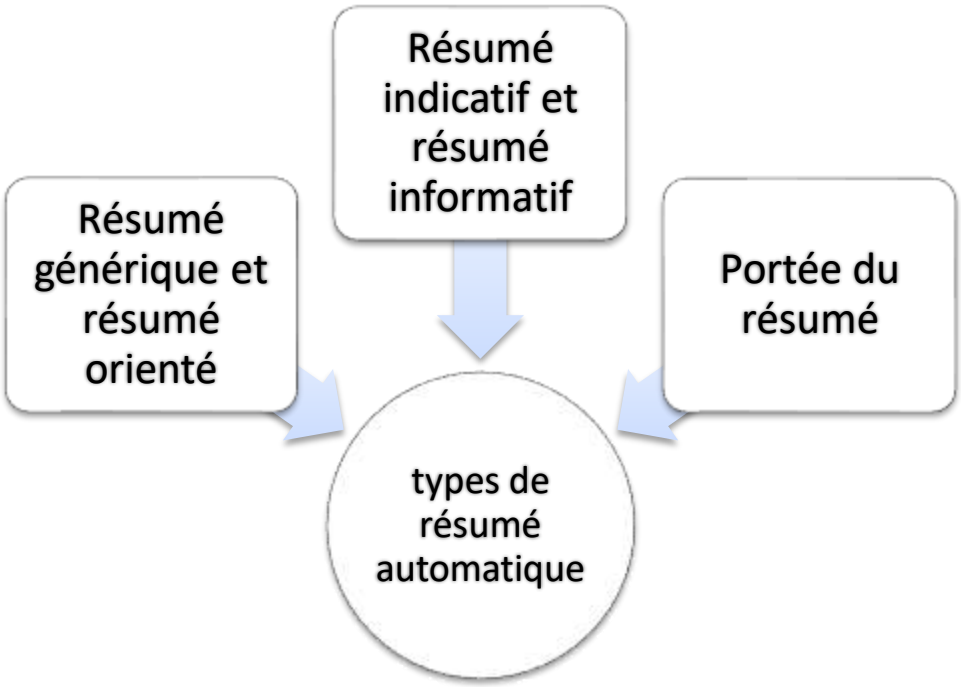


Figure 1-3: Les types de résumé automatique de texte [Mnasri 2018]

1.4.1 Résumé générique et résumé orienté

Si le résumé textuel est général ou orienté. Le résumé général est généré en ne citant que le contenu du texte source, quel que soit son contexte. À l'inverse, les résumés orientés sont guidés par des tâches ou des tâches demandent. Dans ce cas, sélectionnez uniquement les informations liées à la tâche ou à la demande. Par conséquent, ce type d'abrégé dépend largement du contexte. Ce dernier peut être défini comme un ensemble de facteurs d'entrée pour le système de résumé automatique [Spärck Jones 2007]. Il couvre l'audience, l'utilisation, la durée, etc. [Mnasri 2018]

1.4.2 Résumé indicatif et résumé informatif

Le résumé est informatif ou indicatif. Un résumé informatif est un modèle réduisez le texte d'origine, Préserver au maximum les informations des fichiers. D'autre part, le résumé indicatif énumère les sujets les plus importants causé par du texte. Quelques systèmes de résumé d'orientation [Saggion 2002] générer un résumé indicatif du texte dans un premier temps. L'utilisateur choisisse des sujets d'intérêt parmi les sujets présentés dans le résumé. Système générez ensuite un résumé des informations du texte guidé par la requête de l'utilisateur. Dans ce cas, la requête concerne tous les sujets sélectionnés dans le résumé indicatif. [Mnasri 2018]

1.4.3 Portée du résumé

Le système de résumé automatique peut être un document unique ou plusieurs documents; le premier peut générer des résumés pour un seul document, et peut être plus ou moins adapté à des documents de différentes tailles: résumé d'article ce n'est pas exactement la même chose que les problèmes causés par le résumé de rapport scientifique. Par conséquent, le système CHORAL[3] basé sur l'analyseur de langage LIMA [Chalendar 2014] est très efficace dans le traitement de longs documents. Il fournit un résumé de 1 à 5 pages pour le rapport de thèse. Les résumés des articles système multi-documents les plus récents génèrent un résumé évolutif d'un ensemble de documents. [Mnasri 2018]

1.5 Les étapes de résumé automatique de texte

Dans le résumé automatique de documents, trois étapes différentes peuvent être identifiées. Ces étapes sont: l'identification du sujet, l'interprétation et la génération abstraite. Aujourd'hui, la plupart des systèmes n'utilisent que la première étape.

L'identification du sujet produit un résumé simple; une fois que le système identifie les unités importantes, il sera présenté sous la forme d'un résumé. Ensuite, l'interprétation comprend incorporer des concepts, des évaluations ou d'autres procédures qui utilisent des connaissances autres que le document d'entrée. Le résultat de l'interprétation est que l'abrégé est illisible ou que le contenu extrait est incohérent. Par conséquent, l'étape de génération est utilisée pour produire un texte (document) lisible par l'homme, et dans le cas de l'extraction, cette étape peut être considérée comme une étape de "lissage" pour rendre le résumé plus cohérent, la figure suivante représente ses étapes. [ARIES 2013]

[3]CHaine d'Outils pour le Résumé Automatique du LVIC: pipeline of automaticsummarizationtools of the LVIC laboratory

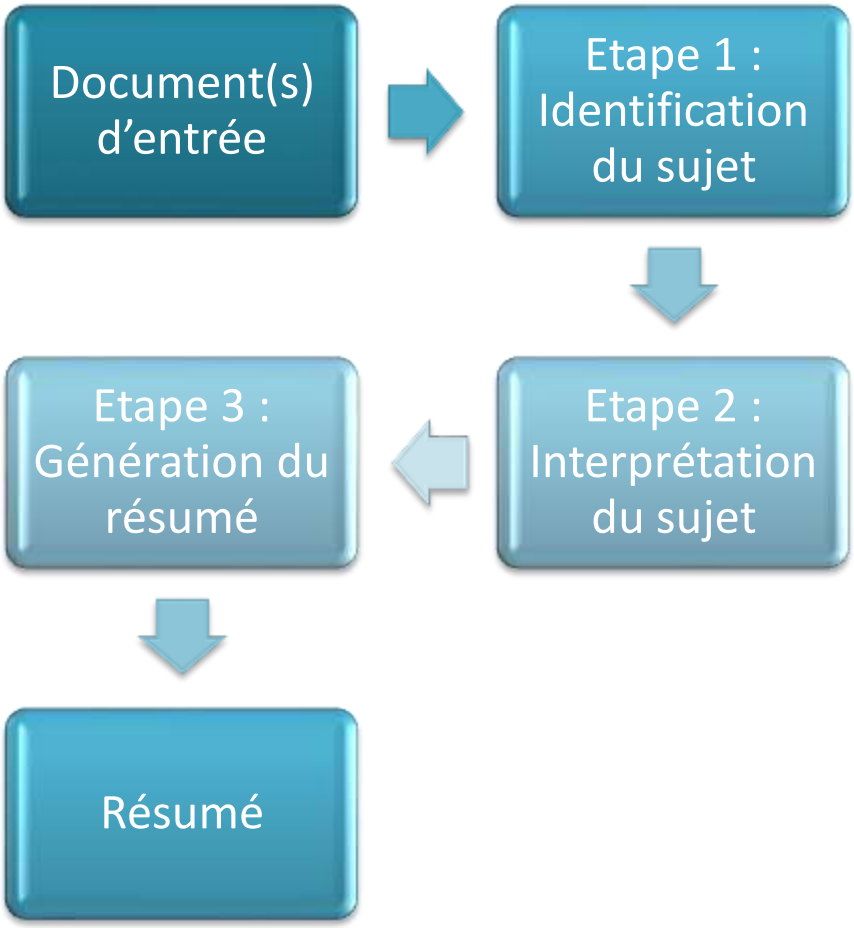


Figure 1-4: Les étapes du résumé automatique [ARIES 2013]

Etape 1 : Identification des thèmes

Il génère un résumé simple (extrait) en détectant les unités importantes (mots, phrases, paragraphes, etc.) dans le document. Un système de résumé qui n'utilise que des étapes d'identification du sujet produira un résumé. Cela se fait en filtrant les fichiers d'entrée pour obtenir uniquement les sujets les plus importants. Une fois ces thèmes déterminés, ils seront présentés sous forme d'extraits. Pour effectuer cette étape, presque tous les systèmes utilisent plusieurs modules indépendants. Chaque module note les unités d'entrée (mots, phrases ou paragraphes plus longs), puis un module de combinaison combine les scores de chaque unité pour attribuer un score unique. Enfin, le système renvoie l'unité avec le score le plus élevé en fonction de la demande de l'utilisateur ou de la longueur de résumé préalablement définie par le système. [ARIES 2013]

Etape 2 : Interprétation des thèmes

Dans l'interprétation, le but est de compresser en réinterprétant et en fusionnant les sujets extraits pour les faire avoir des sujets plus courts. Ceci est essentiel car les résumés sont générale-

ment plus courts que les extraits équivalents. La deuxième étape du résumé automatique (passage de l'extrait au résumé) est naturellement plus compliquée que la première étape. Pour mener à bien cette étape, le système a besoin de connaissances sur le monde, car sans connaissance, aucun système ne peut fusionner les sujets extraits pour produire moins de sujets pour former des abstractions.

Au cours du processus d'interprétation, les sujets identifiés comme importants seront fusionnés, exprimés en de nouveaux termes et concepts ou mots qui n'existaient pas dans le document original. [ARIES 2013]

Etape 3 : Génération du résumé

Le résultat de l'interprétation est un ensemble des représentations qui sont généralement illisibles, telles que des abstractions. Pour les extraits, en raison de références coupées, de l'ignorance des liens entre les phrases et de la redondance ou de l'omission de certains matériaux, le résultat est rarement un extrait cohérent. Par conséquent, le système comprend l'étape de génération de résumés pour produire un texte cohérent et lisible par l'homme. [ARIES 2013]

1.6 Les méthodes de résumé automatique

Dans cette partie, nous présentons brièvement différentes méthodes d'extraction des phrases clés (voire la Prochain figure), qui reposent principalement sur le calcul du score associé à chaque phrase afin d'estimer son importance dans le texte. Seules les phrases avec les scores les plus élevés seront conservées dans le résumé final. [Douzidia 2004]

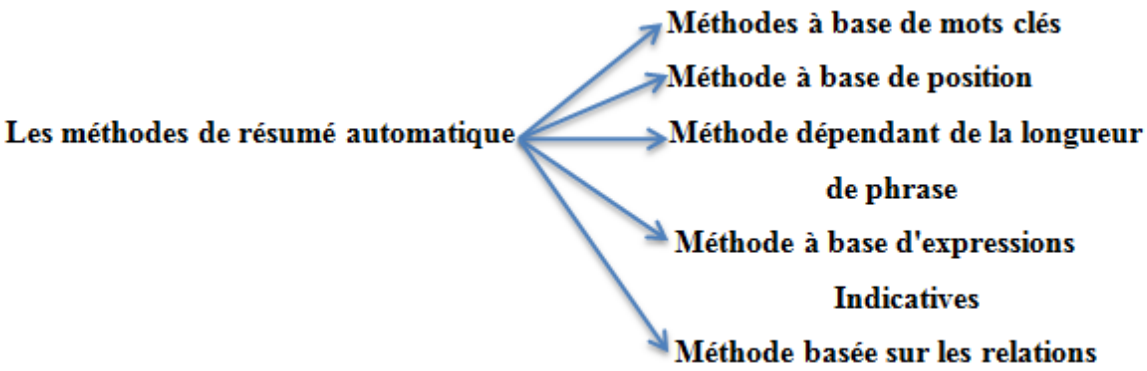


Figure 1-5: Les méthodes de résumé automatique [Douzidia 2004]

1.6.1 Méthodes à base de mots clés

Cette méthode repose sur le fait que l'auteur utilise (pour exprimer son idée principale) des mots-clés, qui apparaissent souvent dans le texte. Ensuite, un résumé automatique est généré en recherchant dans le texte source la plus petite unité de texte correspondant à ses mots-clés. Ce principe est généralement appliqué aux différentes variantes décrites dans les sous-sections suivantes. [Douzidia 2004]

a. Mots-clés prédéfinis

Pour calculer le score en fonction des mots-clés contenus dans chaque phrase S, nous pouvons calculer le score suivant:

Score $_{\text{mot-clé}}$(S) = a (t) * F (t)

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

$$a(t) = \begin{cases} \text{A si } t \in \text{liste de mots - clés (A > 1)} \\ \text{1 sinon} \end{cases}$$

La liste de mots-clés peut être constituée d'une entrée utilisateur (zone d'intérêt) ou de mots-clés créés par l'auteur. L'importance du poids de t est donnée par A × F (t), où A> 1.

b. Titre

Comme le titre est la phrase la plus significative et résume le mieux d'un document en quelques mots, on peut dire que la phrase la plus similaire au titre est la plus visible dans le document. Par conséquent, chaque phrase peut être pondérée en fonction de sa similitude avec le titre. Dans ce cas, nous traitons les mots du titre du texte comme des mots clés et générons un résumé en sélectionnant des phrases qui couvrent certains mots qui apparaissent dans le titre.

Score $_{\text{titre}}$(S) = b (t) * F (t)

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

$$b(t) = \begin{cases} \text{A si } t \in \text{liste de mots - clés (A > 1)} \\ \text{1 sinon} \end{cases}$$

c. Distribution des termes

L'idée de cette méthode est de traiter les phrases contenant des mots importants dans le texte comme des phrases importants. Si un mot est utilisé assez fréquemment dans le texte, il est considéré comme important.

1.6.2 Méthode à base de position

Cette méthode suppose que la position de la phrase dans le texte indique son importance dans le contexte. Par exemple, la première et la dernière phrase d'un paragraphe peuvent véhiculer l'idée principale et doivent donc faire partie du résumé. En variante de cette méthode, on peut citer la méthode Lead, qu'est une méthode d'identification des phrases importantes en extrayant des phrases principales. Comme des phrases importantes ont tendance à apparaître dans les premières phrases d'un article, cette méthode est très efficace pour résumer des articles de journaux. [Douzidia 2004]

Nous définissons le score de la phrase S à la position i comme indiqué ci-dessous:

Score $_{lead}$ (S_i) = β_i

$$\beta_i = \begin{cases} \mathbf{B > 0 \text{ si } i > 0} \\ \mathbf{0 \text{ si } i >= N} \end{cases}$$

β_i est une fonction rectangulaire qui simule la distribution des phrases importantes en fonction de leurs positions dans l'article. Si la dernière phrase est importante, vous pouvez 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 et du style de l'auteur.

1.6.3 Méthode dépendant de la longueur de phrase

Cette méthode attribue des poids aux phrases en fonction du nombre de mots de la phrase. Deux techniques peuvent être utilisées pour calculer les scores [Douzidia 2004]:

- La longueur de chaque phrase (L_i) est relative à la longueur maximale de la phrase. Fraction longue (S_i) = L_i / L_{max}.
- Attribuez zéro point aux phrases plus courtes qu'une certaine longueur (L minimum).

1.6.4 Méthode à base d'expressions indicatives (cuemethods)

Cette méthode sélectionne des unités de texte avec des instructions spécifiques ou des expressions spécifiques. Par exemple, pour les manuels de sciences, nous exprimons le but de ce travail ..., cet article propose ..., les résultats et les conclusions sont de bons candidats pour montrer les phrases contenues dans le résumé. Différents types de texte peuvent avoir différentes expressions indicatives. Pour une caractéristique donnée, nous pouvons déduire le score de toute phrase de texte à analyser en fonction de la similitude présentée. [Douzidia 2004]

Nous pouvons définir le score de la phrase S correspondant à un certain modèle comme:

$$\text{Score-}_{cue}(S) = \begin{cases} \textbf{1 si S correspond à un motif} \\ \textbf{0 sinon} \end{cases}$$

1.6.5 Méthode basée sur les relations (cohésion lexicale)

L'utilisation de la fréquence des mots est un bon moyen de mettre en évidence des termes clés dans un texte, mais elle ne tient pas compte de la relation entre les mots dans différentes parties du texte. L'exploration de phrases basée sur la fréquence des mots conduit généralement à un manque de cohésion. Pour surmonter ce problème, les personnes déployées sur le terrain ont développé une méthode basée sur la cohésion grammaticale (ie, citation, substitution, conjonction) et la cohésion lexicale (ie, mots sémantiquement liés).Cette méthode indique que plus une phrase a de connexions avec une autre phrase du texte, plus elle est appropriée dans ce cas, c'est-à-dire qu'elle exprime le même sujet. Par conséquent, le contenu des phrases connexe doit être sélectionné ensemble pour former un résumé. L'omission de certaines phrases très pertinentes peut produire un texte incohérent. Cette corrélation est généralement identifiée à partir d'un vocabulaire ou d'un dictionnaire informatisé, ce qui permet de déterminer la relation entre les mots. Les chaînes de vocabulaire sont composées de mots de texte candidats, et ces chaînes combinent des mots liés entre eux via le thesaurus et l'extrait des phrases les plus pertinentes pour la chaîne de vocabulaire. [Douzidia 2004]

1.7 Evaluation du résumé automatique

Pour évaluer l'efficacité du système de résumé automatique, il n'y a pas un seul protocole, mais nous présenterons ici diverses méthodes. Par conséquent, nous allons introduire deux catégo-

ries d'évaluation intrinsèque et extrinsèque. Ensuite, nous nommerons deux méthodes d'évaluation: les méthodes PYRAMID et ROUGE, ce sont toutes des méthodes d'évaluation intrinsèque.

1.7.1 Classification des méthodes d'évaluation

D'après [Mani 2001], les méthodes d'évaluation des résumés textuels peuvent être divisées en deux types. Le premier est l'évaluation intrinsèque, qui implique une évaluation systématique des aspects suivants: Résumé interne. Elle se concentre sur l'évaluation de la cohérence et le contenu informatif des résumés générés. La seconde est l'évaluation extrinsèque, qui teste l'impact du résumé sur des tâches telles que l'évaluation de la pertinence et la compréhension en lecture. La figure ci-dessous montre les différentes catégories d'évaluation de résumé automatique.

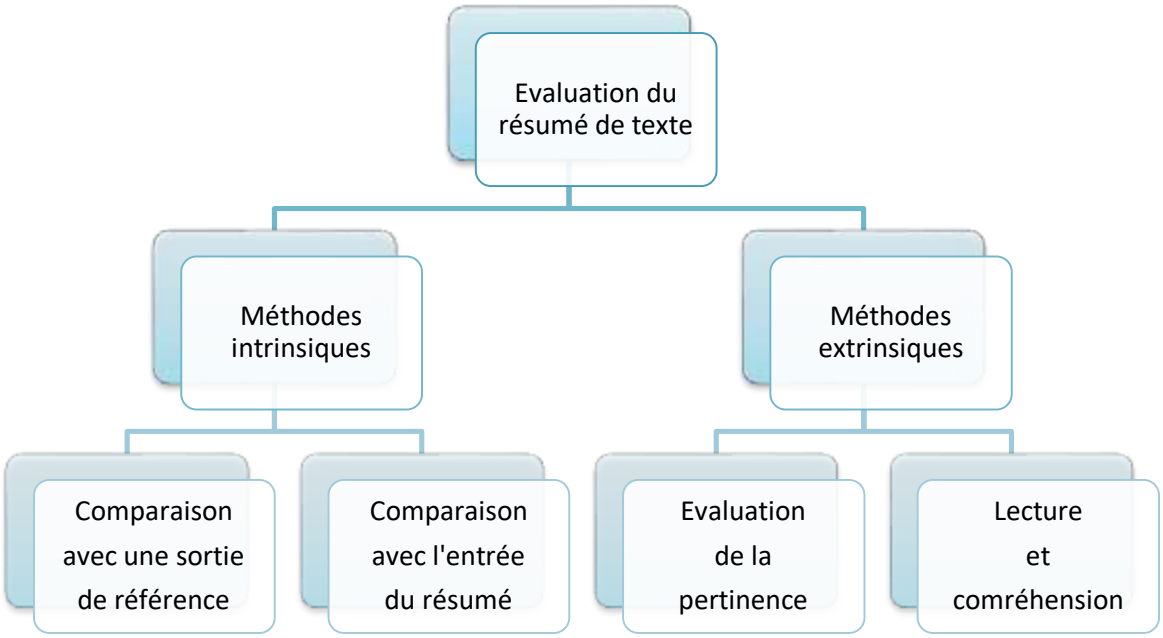


Figure 1-6 : Les approaches d'évaluation des systems de résumé automatique [ARIES 2013]

1.7.1.1 Évaluation intrinsèque

a) <u>Critères utilisés</u>

Afin d'évaluer les résumés générés en intrinsèque, comparez les résumés contenu lié au résumé de référence ou contenu complètement lié au document original selon différentes normes. Le premier est la cohérence, qui consiste à vérifier la lisibilité du résumé. Pour l'extraction abstraite, ces problèmes sont dus à l'existence de lacunes de sa structure rhétorique. Pour les résumés, les vérifications qui peuvent être effectuées selon [Saggion 2000] comprennent: l'orthographe et la grammaire correctes, des instructions claires sur le document original, le style impersonnel, la con-

cision, la facilité de lecture et de compréhension, etc. Mais d'ici là, il est presque impossible d'automatiser cette évaluation.

Un autre critère est le contenu informatif du résumé, dont le but est d'estimer les informations contenues dans le résumé. Le résumé étant plus court que le texte original, il conserve moins d'informations. Par conséquent, mesurer le contenu d'informations d'un résumé implique d'estimer la quantité d'informations conservées par rapport au texte source.

b) Comparaison avec une sortie de référence

L'idée est de comparer les résumés automatiques avec les résumés de personnes de même textes ressources. Problème d'évaluation humaine autrement dit, bien que plusieurs résumés de référence puissent être trouvés pour le même document, le système peut générer des résumés différents de tous ces résumés de base, mais les résumés sont informatifs et cohérents. Dans une expérience menée en [Rath 1961], des humains ont été invités à extraire 20 phrases de 10 articles scientifiques, et il a été constaté que les juges ont fourni des résumés différents pour la même source. De plus, on a découvert que pour le même juge, il pouvait dessiner deux résumés complètement différents dans un intervalle de huit semaines. Nous pouvons également utiliser différentes métriques (y compris ROUGE) pour évaluer automatiquement le résumé.

c) Comparaison avec l'entrée de résumé

Dans ce type d'évaluation, le résumé et la source sont en le contenu informatif du résumé doit être évalué dans le contexte de la source. Selon [Mani 2001] Il existe deux méthodes pour comparer les résumés et les sources: les méthodes sémantiques et les méthodes de surface. La méthode sémantique consiste à comparer le sens dans le texte source avec le sens dans le résumé. Une façon est de marquer le sens de chaque phrase dans l'abstrait, puis de regarder combien de clauses existent dans la source couverte par le résumé. Dans l'approche de surface, nous n'essaions pas de représenter ces concepts à un niveau profond, il suffit donc de juger si les idées principales des idées originales sont couvertes dans le résumé.

1.7.1.2 Évaluation extrinsèque

L'idée d'évaluation extrinsèque d'un résumé est de déterminer les effets abstraits autres tâches. Plusieurs tâches peuvent appliquer des résumés, notamment nous pouvons citer les tâches mentionnées dans [Mani 2001]:

- Si le résumé affecte le comportement d'autres tâches, l'efficacité peut être mesurée en effectuant ces tâches. Par exemple, si nous avons un système de prise de décision basé sur notre système de résumé automatique, nous pouvons mesurer l'efficacité de notre système de résumé en examinant son impact sur le système décisionnel.
- Nous pouvons vérifier l'utilité du résumé en fonction des informations des besoins ou des objectifs, c'est comme trouver des documents liés aux besoins d'une personne sous tous ses aspects collection.
- Il est possible d'évaluer l'impact du résumé sur le système qu'il contient, par exemple, comment l'outil résumé aide-t-il dans le système de questions-réponses?

a) Évaluation de la pertinence

Dans l'évaluation de la pertinence, le document et le sujet sont présentés à une seule personne, et demandez-lui de déterminer la relation entre le document et le sujet. Impact du résumé par conséquent, l'exactitude et le calendrier de la tâche ont été étudiés. Bien que l'évaluation SUMMAC y compris une évaluation interne de questions-réponses et l'objectif principal est l'évaluation externe. Grâce à l'utilisation de documents (qui peuvent être des résumés ou des textes normaux, l'évaluateur ne peut pas comprendre sa nature a priori) et des descriptions de sujets, les évaluateurs humains peuvent déterminer la pertinence du document par rapport au sujet. Par conséquent, il doit sélectionner une catégorie parmi les cinq catégories (chacune avec une description) qui représentent le sujet du document, ou ne pas sélectionner «Aucune catégorie». [ARIES 2013]

b) Lecture et compréhension

Dans la tâche de lecture et de compréhension, l'évaluateur lit d'abord la source et / ou un résumé récapitulatif d'un ou plusieurs documents. Il a répondu à un test à choix multiples sur le contenu du document. Ensuite, le système calcule le pourcentage de réponses correctes. Par conséquent, la compréhension humaine basée sur l'abstrait peut être comparée objectivement à la compréhension humaine basée sur la source. Donc, le raisonnement est le suivant: si la lecture du résumé autoriser une personne à répondre à la question comme si elle lisait le matériel original, le résumé est très utile. [ARIES 2013]

1.7.2 Méthodes d'évaluation

Il existe plusieurs méthodes d'évaluation de résumé automatique de texte. Dans ce qui suit, nous expliquons deux d'entre eux: la méthode ROUGE et la méthode PYRAMID (voire la figure suivante).

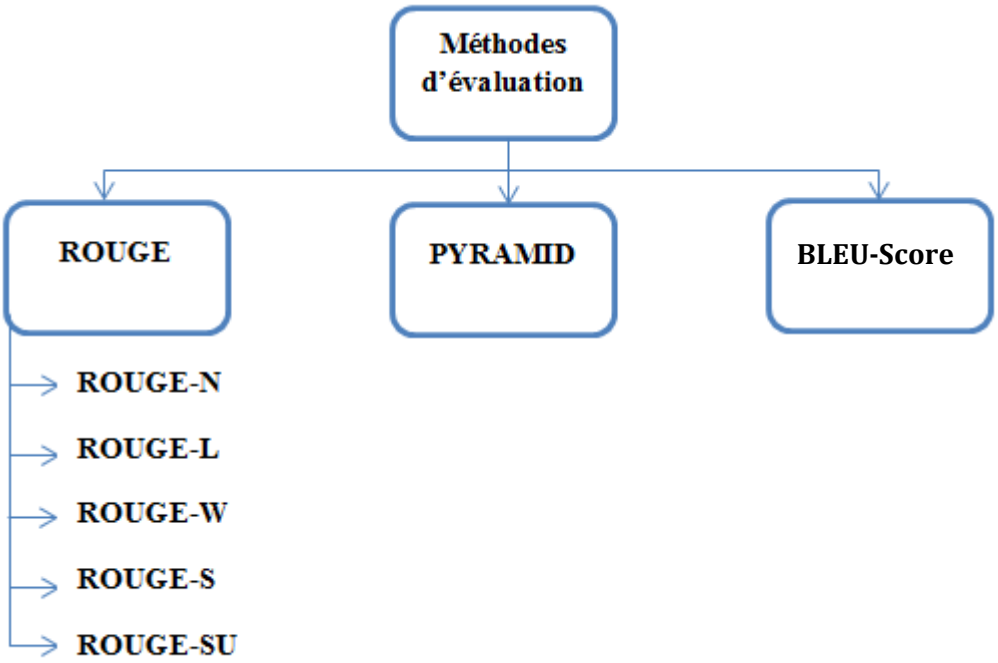


Figure 1-7: Quelques methods d'évaluation [ARIES 2013]

1.7.2.1 La méthode ROUGE

ROUGE[4] développé est l'inspiration pour une autre méthode d'évaluation système de traduction automatique, appelé BLEU[5]. Le but est déterminez automatiquement la qualité du résumé en comparant avec un autre résumé du référence créée par des personnes. Le principe est de calculer le nombre d'unités de récupération tels que les n-grammes, les séquences de mots et les mots combinés entre les résumés générés par ordinateur et résumé de référence. La méthode ROUGE est basée sur comme le dénominateur est le nombre de n-grammes dans le résumé de référence. La Méthode ROUGE contient cinq variantes : ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S et ROUGE-SU. [ARIES 2013]

a) <u>ROUGE-N</u>

[4]ROUGE : Recall-OrientedUnderstudy for Gisting Evaluation
[5]BLEU : BiLingual Evaluation Understudy

Formellement parlant, ROUGE-N est un rappel de n-grammes entre un résumé candidat et un ensemble de résumés de référence. La valeur de ROUGE-N peut être calculée par la formule suivante:

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

Parmi eux, N est la longueur du N-gramme et count match (N-gramme) est le nombre de N-grammes.Résumés des candidats trouvés dans l'un des résumés de référence ($Summ_{ref}$). Count (N -gram) est reportez-vous aux N- grammes dans le résumé.

b) ROUGE-L

ROUGE-L utilise la plus longue sous-séquence commune(LCS[6]) entre deux phrases utilisez X avec m_x mots et Y pour utiliser n_y mots pour estimer la similitude entre ces deux phrases. L'équation suivante représente ça:

$$R_{lcs} = \frac{\sum_{i=1}^{u} LCS(X,Y)}{m_x}, P_{lcs} = \frac{\sum_{i=1}^{u} LCS(X,Y)}{n_y}$$

Pour appliquer LCS à l'évaluation d'un résumé, vous devez regarder les phrases du résumé comme une série de mots. Sachez que la séquence commune la plus longue (LCS) ne aucun appariement continu entre les deux séquences n'est supposé.Afin de mieux comprendre comment calculer ROUGE-L, voici un exemple de phrase de référence (r1) et deux phrases candidates (c1, c2):

r1 [A B C D].

c1 [A E CD].

c2 [CD E A].

c3 [CDAB].

Le ROUGE-L de c1 sera 3/4, et le ROUGE-L de c2 sera 2/4. Par conséquent, selon ROUGE-L, c1 est meilleur que c2. Le problème avec ROUGE-L est qu'il ne calcule que la séquence principale, donc seuls les autres LCS Les longueurs alternatives ou les longueurs plus

[6]LCS : Longest Common Subsequence

courtes ne seront pas prises en compte lors du calcul du score. Dans la phrase c3, LCS ne calcule qu'une séquence au lieu de deux séquences, donc la valeur ROUGE-L de c3 sera égale à la valeur de c2.

L'équation suivante représente la méthode de calcul de ROUGE-L (rappel et précision) entre un résumé de référence ri ϵ R et un résumé candidat C.

$$R_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{m}, P_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{n}$$

Où:

- u est le nombre des phrases dans le résumé de référence R.
- LCSU (ri, C) est le score LCS de l'union des plus longues séquences entre la phrase de référence ri et les phrases du résumé candidat C. Par exemple :

Supposant une phrase de référence ri = w1w2w3w4w5, et C contient deux phrases :
c1 = w1w2w6w7w8 et c2 = w1w3w8w9w5. Donc, le LCS entre ri et c1 est "w1w2",
et le LCS entre ri et c2 est "w1w3w5" ; L'union de ces deux LCS nous donne
"w1w2w3w5" et LCS ∪ (ri, C) = 4.

- m est le nombre des mots dans R, et n est le nombre des mots dans C.

c) ROUGE-W

Bien que le LCS mentionné précédemment ait de bonnes performances, il est toujours confronté au problème de la distinction des sous-séquences communes avec des relations spatiales différentes et leur commande d'origine. Par exemple, si nous avons une séquence de référence r1 et deux candidats séquences c1 et c2 sont les suivants:

r1 [ABCD E F G]
c1 [ABCD H I K]
c2 [A H B K C I D]

c1 et c2 ont le même score ROUGE -L, mais dans ce cas, c1 devrait avoir un score plus élevé vers c2, car il contient des paires consécutives. Afin d'améliorer la méthode LCS, la correspondance continue obtient des scores plus élevés que la correspondance non continue. Ceci peut être réalisé en pénalisant les rayures non continues.

L'équation suivante représente la méthode de calcul ROUGE-W (taux de rappel et précision) entre le résumé ri ϵ R et le résumé candidat C.

$$R_{wlcs} = f^{-1}(\frac{WLCS(r_i, C)}{f(m)}), P_{wlcs} = f^{-1}(\frac{WLCS(r_i, C)}{f(n)})$$

Où:

- f est une fonction satisfaisant la relation f (x; y) > f (x) + f (y). Prenant f (k) = k^2 où k est la taille d'une LCS.
- WLCS (ri;C) = ∑ j f (LCS j(ri,C)), LCS j(ri,C) est la taille de LCS numéro j entre la phrase de référence ri et les phrases du résumé candidat C. Par exemple :

  Dans l'exemple précédent, c1 donne une seule LCS : "A B C D". Donc,

  ROUGE-W (c1) =$\sqrt{(4^2)/7}$ = 0.57143.

  Pour la deuxième séquence c2, nous avons quatre LCS : "A", "B", "C", et "D". Donc,

  ROUGE –W (c2) =$\sqrt{(1^2 + 1^2 + 1^2 + 1^2)/7}$ = 0.28571.

- m est le nombre des mots dans R, et n est le nombre des mots dans C.

d) <u>ROUGE-S</u>

ROUGE-S [7] est une extension de ROUGE-N; il est calculé de la même manière que ROUGE-2[8], sauf qu'il utilise une grammaire binaire trouée au lieu d'une grammaire binaire continue. Le bi-gramme espacé (saut de deux tuples) est n'importe quelle paire de mots disposés dans l'ordre dans une phrase, permettant toutes les lacunes. Si nous avons X textes de référence avec m mots et Y textes candidats avec n mots, ROUGE-S peut être calculé comme suit.

$$R_{SKIP2}(X, Y) = \frac{SKIP2(X, Y)}{C(m, 2)}, P_{SKIP2}(X, Y) = \frac{SKIP2(X, Y)}{C(n, 2)}$$

Où S KIP2(X; Y) est le nombre des bi-grammes à trous similaires entre X et Y. C est la function de combinaison.

Prenons l'exemple utilisé dans ROUGE-L comme exemple, où r1 est la phrase de référence, et trois phrases c1, c2 et c3 sont des phrases candidates:

r1 [A B C D].

c1 [A E C D].

[7]ROUGE-S : skip-bigramco-occurrence

[8]ROUGE-N avec des N-grammes deslongeur N

c2 [C D E A].

c3 [C D A B].

Chaque phrase contient C (4, 2) = 6 Bi-grammes avec des trous. Par exemple la phrase c1 a les bi-grammes suivants : ("A E", "AC", "A D", "EC", "ED", "C D"). Les scores ROUGE-S de c1, c2 et c3 sont respectivement de 3/6, 1/6 et 2/6.

e) ROUGE-SU

Un problème avec ROUGE-S est que si la phrase candidate ne contient pas de paires de mots similaires aux mots de la phrase de référence, le mot candidat ne recevra aucun honneur. Par exemple, la phrase suivante a un score ROUGE-S de zéro:

c4 [D C B A].

Cette phrase est l'opposé de la phrase r1 et ne ressemble pas ceux de r1. Cependant, nous devons faire la distinction entre des phrases comme c4 et ne contient aucun mot identique à r1. Pour cela, nous pouvons étendre ROUGE-S avec ajoutez l'uni-gramme comme unité de calcul, ce qui nous donne ROUGE-SU.

Chaque phrase contient C (4, 2) + 4 = 10 grammes. Par exemple, la phrase c1 a les grammes suivants: ("A E", "A C", "A D", "EC", "ED", "C D", "A", "B", "C", "D")). Score ROUGE-SU

Pour c1, c2, c3 et c4 respectivement: 7/10, 5/10, 6/10 et 4/10.

1.7.2.2 La méthode PYRAMID

Cette méthode permet de comparer les résumés candidats avec un ensemble de résumés de référence [Mnasri 2015]. Comme il n'existe pas de résumé idéal et que le style d'écriture de chacun est différent, l'utilisation d'un seul résumé de référence ne peut pas répondre aux exigences d'équité entre les résumés des candidats. Afin d'assouplir cette restriction, au moins 4 résumés modèles sont fournis dans l'activité d'évaluation. Le principe de la méthode PYRAMID est d'annoter le résumé de référence afin d'identifier l'unité appelée SCU (Summary Content Unit) [ARIES 2013]. SCU est un ensemble d'unités de texte récapitulatif qui expriment les mêmes informations. Le poids qui lui est attribué est égal au nombre de résumés de référence qui l'ont instancié [ARIES 2013].Ces SCUs peuvent être organisés en une pyramide, où chaque couche regroupe le SCU avec le même poids. Pour évaluer le résumé, veuillez l'annoter pour identifier le résumé. Par la suite, chaque SCU candidat héritera du poids du SCU le plus similaire de la pyramide. Le score PYRAMID d'un résumé est finalement le rapport de la somme des poids de tous ses SCUs candidats à la somme des poids du résumé idéal avec le même nombre d' SCU. L'inconvénient de cette méthode est qu'elle nécessite une étape pour annoter le résumé. Le calcul du score PYRAMID est effectué automatiquement en

utilisant une sémantique distribuée. Malheureusement, l'annotation des résumés de modèles est encore difficile à automatiser. [Mnasri 2015]

1.7.2.3 La méthode BLEU-Score

BLEU (bilingual evaluation understudy) est un algorithme d'évaluation de la qualité du texte qui a été traduit mécaniquement d'une langue naturelle à une autre, c'est une valeur entre 0 et 1.

1.8 Les travaux similaires que résumé automatique du texte

Il existe plusieurs travaux dans le domaine de résumé automatique, ces travaux utilisent différentes méthodes et techniques. Dans ce qui suit, nous discuterons certains d'entre eux.

1.8.1 Résumé automatique de textes (mémoire magister de l'Ecole Nationale Supérieure d'Informatique ESI) [ARIES 2013]

Monsieur ARIES Abdelkrime réalise un système de résumé automatique de texte mono-document et multi-document. Il recommande l'utilisation d'une classification et d'un regroupement basés sur les critères du type de nombre pour générer un résumé général indépendamment du genre ou de la langue. Sachant que la classification n'est basée sur aucun corpus de formation, elle est utilisée pour trouver un modèle pour chaque cluster, puis noter les phrases suivant ces modèles.En comparaison avec d'autres systèmes de résumé automatique, ce système ne fait pas exception.Il comprend également trois modules classiques: pré-traitement, traitement et post-traitement, tel que: le pré-traitement nécessite les étapes suivantes :

- Utilisez le framework OpenNLP[9] d'Apache pour la segmentation de phrases.
- Pour détecter les mots, il a implémenté un algorithme simple qui utilise l'espace comme séparateur.
- Pour la radicalisation, il utilise Porter-stemmer[10], la réalisation de l'algorithme de Porter en Java [Porter 1997].

Le traitement contient module d'extraction qui compose de deux étapes : Étape de détermination du sujet dans le fichier et l'étape de classification. Pour détecter différents sujets qui apparaissent dans le document, il utilise le regroupement, ici il suppose que les phrases contient les mots similaires sont des phrases avec le même thème. Dans l'étape de classification, il essaye concevoir un modèle

[9]Site web : http://opennlp.apache.org/
[10]http://www.tartarus.org/~martin/PorterStemmer

pour chaque groupe (cluster), Sera utilisé pour calculer la probabilité d'une phrase appartient à un sujet donné.

- Il utilise un algorithme d'apprentissage (Naïve Bayes), pour entrainer le système sur les différents clusters et les critères.
- Contrairement aux quelques systèmes precedents basés sur l'apprentissage, ce système n'a pas besoin d'un corpus d'entraînement, L'apprentissage n'est utilisé que pour obtenir le score de chaque cluster.
- Pour le regroupement, il a utilisé la similarité Cosinus, est une mesure de calcul de la similarité entre deux segments.

Dans le module de post-traitement, Les étapes suivantes décrivent l'algorithme à suivre pour éliminer la redondance:

- Étant donné toutes les phrases triées par score, ajoutez la première phrase au résumé.
- Pour chaque phrase candidate, vérifiez si le résumé a atteint sa limite. le cas échéant, on s'arrête ; Sinon, on continue.
- Calculez la similitude entre la dernière phrase ajoutée au résumé et la phrase candidate. Si la similitude est inférieure à une certaine valeur (il utilise 0,5), ajoutez-la à Abstrait. Sinon, il passe à la phrase suivante.

Concernant l'analyse de l'anaphore, il choisisse d'utiliser groupe de traitement du langage naturel de l'Université de Stanford "StanfordCoreNLP[11] "[Lee 2013]. Ce système implémente la méthode de résolution d'anaphoreMulti-pass. Dans ce travail, il a tous passé le texte du module puis corrigez le contenu au niveau du résumé. Pour l'évaluation, monsieur ARIES travail avec le corpus Cmp-lg (Computation and language), dans cette évaluation, il espère atteindre les trois objectifs suivants:

- En termes de qualité, comparez se système avec un autre système.
- Testez l'effet de l'ajout de nouvelles conditions au système.
- Modifiez le seuil de regroupement pour voir son effet sur le résumé généré.

Pour cela il exécute les étapes suivantes :

[11]Site web : http://nlp.stanford.edu/software/corenlp.shtml

- Extrayez le résumé et le texte de chaque document dans les 182 documents du corpus cmp-lg. Le résumé extrait sera utilisé comme un résumé de référence et le corps du document sera utilisé comme document d'entrée du système de résumé.

- Utilisez le système UNIS pour générer un résumé [Yatsko 2010]. Le résultat est 174 résumés (les huit documents restants sont filtrés car ils ne sont pas utiliser un résumé UNIS ou aucun résumé de référence).

- Il utilise le critère du uni-gramme (la fréquence des mots dans chaque catégorie), il a généré 4 résumés pour chaque document et modifié le seuil de regroupement: 0,0, 0,1, 0,5 et 0,99. Le nombre de phrases qu'il extraie est égal au nombre de phrases extraites par UNIS.

- Il utilise la norme: uni-gramme et bi-gramme, il a généré 4 résumés utilisez le seuil de regroupement précédent.

- Comparez neuf résumés en tant que résumés candidats avec des résumés de reference utilisé ROUGE.

Il travail avec les méthodes d'évaluation suivantes : ROUGE-1, ROUGE-2, et ROUGE-SU4.

Les déférences entre ce travail et les autres sont :

- ✓ Les autres méthodes de regroupement, essayez de choisir une phrase représentative de chaque cluster. Au lieu de cela, dans ce travail, il essaye de sélectionner des phrases représente le nombre maximal de clusters.

- ✓ La méthode d'apprentissage utilise un algorithme de classification pour trouver le poids de la norme, ou décider si la phrase appartient à l'abstrait. Dans leur méthode, l'algorithme de classification est utilisé comme moyen de notation.

- ✓ Sa méthode n'utilise pas un corpus d'entraînement, elle est donc incompatible avec le type et la langue du document source.

Il s'avère que son système bon performance. L'ajout de conditions peut améliorer la précision du système.

1.8.2 Les Résumés Automatiques des Documents Textuels (université de Abdelhamid Ibn Badis, Mostaganem) [Boudraf 2017]

A été développé par Boudraf Khadidja dans le cadre de sa mémoire master (2016-2017). Ce système s'appliqué à mono et multi-document, utilise l'approche extractive. Il s'appuie sur le système de résumé automatique Intellexer Summarizer[12] pour évaluer le systéme par la comparaison entre le résultat des deux systèmes et le moteur de recherche open source Lucene[13]pour analyser les documents et supprimer les mots inutiles (mots vides). Ensuite le regroupement des phrases C'est que ces phrases sont automatiquement regroupées en groupes en fonction de la similitude du contenu pour éviter de répéter la même phrase. Et le Prétraitement linguistiques des documentes qui passe par la Segmentation (Il s'agit de convertir le document en un ensemble de phrases liées à la ponctuation), Filtrage (suppression des mots vides), Stemming (obtenir la forme originale des mots) et finalement Lemmatisation (obtenir la forme canonique des motes).

Les méthodes utilisées pour générer ce système de résumé sont :

- Méthode dépendant de la longueur de phrase.
- Méthode à base de position de la phrase.
- Méthode à base de mots-clés du document.
- Méthode à base de mots clés de la première phrase du document.

Ce système fait la combinaison entre ces méthodes. Et concernant le Corpus de test, Boudraf utilise deux corpus, l'un est un corpus de test contenant 32 documents, et l'autre est un corpus de jugement, qu'est un corpus de tous les résumés générés par le système integer « Intellexer Summarizer ». Elle calcule la mesure «ROUGE» pour évaluer le systéme.

De ce qui précède, Boudaraf conclu que Les méthodes basées sur la localisation sont meilleures pour générer automatiquement des résumés de texte, car elles peuvent aider à identifier les phrases importantes en extrayant des phrases principales dont les résumés sont plus proches du document original.

1.8.3 Generative models for automatic multi-document summarization (notre université SaadDahleb Blida1) [Bensidiaissa 2020]

A été développé par Bensidiaissa Walid et Bouchetara Rym dans le cadre de son mémoire master (2019-2020). Ils se sont d'abord concentrés sur l'extraction de résumés à partir de miltu-

[12]http://summarizer.intellexer.com/downloads.html
[13] https://archive.apache.org/dist/lucene/java/3.6.2/

documents dans le quel ils utilisent le composant BERT à grande échelle de BERTSUM [Yang 2019] pour extraire les caractéristiques des phrases, puis utilisent l'algorithme de regroupement K-signifieici qui regroupe et génére la phrase la plus proche de chaque centroïde, puis ils ont amélioré le dernier modèle DistilBart qu'est une version allégée de BART, composée de BERT et décodeurs de type GPT-2. DistilBart proposé par l'équipe de Huggingface pour obtenir des résumés abstraits de différents corpus de données, puis ont comparé chaque modèle obtenu avec le modèle de base, puis comparé entre eux.

Ils ont également créé un algorithme pour le prétraitement de la synthèse multi-documents. Le but de cet algorithme est de remplacer des phrases similaires regroupées par une seule phrase appartenant à cette dernière. L'algorithme utilise également un modèle basé sur les Transformaters. Leur méthode est très simple, mais en raison des scores élevés utilisant ce prétraitement, les résultats sont encourageants.

Ils ont utilisés plusieurs ensembles de données sur la même architecture (DistilBart), ces ensembles de données sont : CNN/DailyMail, AESLC, Multi-news, Gigaword, DUC 2004. Ils utilisent le chevauchement uni-gramme et bi-gramme (ROUGE-1et ROUGE-2) comme méthode pour évaluer le caractère informatif, et travaillent avec la sous-séquence commune la plus longue (ROUGE-L) comme méthode pour évaluer la fluidité. Ils ont constaté que les meilleurs résultats peuvent être obtenus après avoir affiné le modèle sur Multi-news.

Les principales contributions de leurs travaux sont les suivantes:

- Ils ont affiné le modèle existant sur le nouvel ensemble de données pour la synthèse d'un seul document, puis ils ont ajusté le modèle résultant pour générer des résumés abstraits de plusieurs documents.
- Ils ont proposé un algorithme pour ajuster leur modèle afin de générer des résumés pour plusieurs documents de type extraction et de type abstraction.

1.8.4 Automatic text summarization for crisis management: Coronavirus (COVID-19) case (université de Akli Mohand Oulhadj , Bouira)[AID 2020]

Dans le cadre de sa mémoire master (2019-2020), AID Aicha a développé un système de résumé automatique qui résume les documents liés au Corona virus. Elle a utilisé un modèle de classement de recherche d'informations appelé Okapi BM25, cette dernier est une méthode de classement probabiliste pour la recherche d'informations, qui peut classer les documents en fonction des scores de pertinence. Le score BM25 est calculé sur la base de deux composantes principales: TF et

IDF. Ce travail basée sur une requête en deux étapes, la première étape est le résumé extractif de mono-document piloté par une requête qui repose sur l'utilisation d'un algorithme d'apprentissage automatique sémantique qui combine la base de connaissances WordNet avec la méthode d'extraction d'informations BM25 OKAPI pour générer un résumé pour chaque document. Le jeu de résumé généré sera utilisé dans la deuxième étape. Il s'agit d'un résumé général extractif multi-document, utilisant l'algorithme TextRank pour créer un résumé global unique. WUP_similarity a choisi pour calculer la similarité sémantique dans sa solution proposée. Elle a utilisé l'ensemble de données CORD-19 qu'est la première version publiée en réponse à la COVID-19. Dans ce travail, l'architecture proposée passe par les étapes suivantes :

PREMIÈRE ÉTAPE : Résumé de mono-document basé sur des requêtes extractives.

- Phase 1: Récupération des documents.

Le but de la recherche des documents est de trouver et de sélectionner le meilleur article correspondant à la requête dans la collection. À cette fin, l'indice OKAPI BM25 est utilisé pour mesurer la similitude entre tous les documents. Avant de pour suivre le traitement, le texte doit être passé par le nettoyage des données fait généralement référence à une série de tâches connexes conçues pour mettre tous les textes sur un pied d'égalité:

- Supprimez les données vides et en double, en particulier dans la colonne "body_text".
- Éliminez les articles non anglais.
- Identifiez les nouveaux articles COVID-19 publiés après 2019.
- Supprimez les crochets qui correspondent aux guillemets, y compris les nombres.
- Supprimez les signes de ponctuation, tels que "?", "!", ";" et les caractères spéciaux.
- Mettez le texte en minuscules.

- Prétraitement des documents et des requêtes : est une phase essentielle, Il s'applique spécifiquement au corps de chaque document, comme suit:
    - Segmentation des phrases
    - Tokenisation
    - Suppression des mots d'arrêt

- La lemmatisation
- Étiquetage POS

- Phase 2 : Extraction des informations

1) Score des phrases: Chaque phrase a un score d'importance, qui reflète la qualité de la phrase et la pertinence de la phrase par rapport à la requête. Ces scores peuvent être utilisés pour classer les phrases et sélectionner les phrases les plus importantes. Pour calculer le score elle a utilisé les mesures suivantes :

   A) Similitude sémantique à l'aide de WordNet.

   B) L'algorithme OKAPI BM25.

2) Combiner les scores et obtenir le score moyen.

3) Sélection des phrases.

DEUXIÈME ÉTAPE : Résumé extractif générique multi-documents passé par les étapes suivantes:

- Prétraitement
- Concaténer des phrases
- Appliquer l'algorithme TextRank
- Sélectionner les phrases

Concernent l'évaluation elle a utilisé les mesures suivante :

Precision, Recall, et F-measure dans le contexte du ROUGE pour la comparaison, et les méthodes ce qui suit : ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-W pour évaluer les deux étapes de système.

Les résultats de ce travail :

- ✓ Un bon résumé est celui avec une grande précision et RAPPEL.
- ✓ Le document numérisé est lié au virus Corona et extrait de sa base de données.
- ✓ Par rapport aux résumés humains, cette méthode (combinant WordNet et BM25) donne de bons résultats. Son algorithme s'est avéré efficace pour la plupart des résumés.

- ✓ Comparé à WordNet uniquement, le système implémenté à l'aide de WordNetet BM25 donne de bons résultats, aussi en termes de précision.

1.8.5 NATS (Neural Abstractive Text Summarization) [Shi 2020]

Les auteurs de cet article ont fourni une documentation complète et étudié différentes séquences de résumé de texte abstrait pour des modèles de séquence du point de vue de la structure du réseau, des stratégies de formation et des algorithmes de génération abstraite. Tout d'abord, plusieurs modèles de tâches de modélisation et de génération de langage (comme la traduction automatique) sont proposés, puis ils sont appliqués à un texte abstrait. Par conséquent, ils ont également présenté brièvement ces modèles. Dans le cadre de cette enquête, ils ont également développé une bibliothèque open source, la boîte à outils NATS basée sur le cadre RNN séquence à séquence, pour le résumé de texte abstractif. Une série d'expériences ont été menées sur le réseau CNN / Daily Mail, largement utilisé pour vérifier l'efficacité de plusieurs composants différents dans les réseaux de neurones, Il existe deux versions principales de cet ensemble de données, la première version rend anonyme le nom de l'entité, tandis que la deuxième version conserve le texte d'origine. Dans cet article, ils ont utilisé la deuxième version. Ils ont évalué deux modèles mis en œuvre dans NATS pour deux ensembles de données récemment publiés à savoir Newsroom qu'est utilisé pour saisir et extraire des données brutes. Ils ont développé des outils de traitement de données pour la saisie de texte et la préparation de NATS. Dans cette enquête, ils ont créé deux ensembles de données pour le résumé de texte et la génération de titres et Bytecup, dans leur expérience, ils ont créé un entraînement et créé un ensemble de tests (0,8 / 0,1 / 0,1) basé sur cet ensemble de données d'entraînement, Ces textes ont été symbolisés à l'aide de StanfordCoreNLP et préparés à l'aide de ses outils informatiques. Ils ont utilisé le modèle d'encodeur-décodeur et les méthodes d'évaluation suivante: ROUGE-1, ROUGE-2, ROUGE-L.

1.8.6 Transformer-based Model for Single Documments Neural Summarization [Elozino 2019]

Ils ont proposé un système qui utilise les ensembles des données CNN / DailyMail et Newsroom pour améliorer les performances des tâches de résumé d'un seul document. Il suit le paradigme populaire encodeur-décodeur, Mais avec un accent particulier sur l'encodeur. L'intuition est que la possibilité de décoder correctement les informations réside principalement dans le modèle, tandis que la précision dépend d'encodeur. C'est pourquoi ils ont introduit le codage-codage-décodage. Utilisez d'abord le convertisseur pour coder le cadre du texte source, puis travailler avec le modèle séquence à séquence (seq2seq) pour coder. Ils ont constaté que le convertisseur et le modèle seq2seq peuvent se compléter complètement, ce qui donne une représentation plus riche du

vecteur codé. Ils ont également constaté que le fait d'accorder plus d'attention au vocabulaire du mot cible dans le processus d'abstraction peut améliorer les performances. Ils ont réalisé des hypothèses et des expériences de cadre sur la tâche d'extraction et de résumé d'un seul document. Pour l'approche extractive, Leur classification de modèle se fait que chaque phrase du document est résumée ou non. Afin d'améliorer ce processus de classification de séquence, ils ont utilisé TRANSFORMEUR pour coder le document d'entrée. Le classificateur logistique apprend alors à étiqueter chaque phrase dans le document converti. Cette approche passe par les trois étapes suivantes : TRANSFORMATEUR Encodeur, Extraction de la peine et Formation extractive. Et concernent l'approche abstractive, l'entrée de leur module abstractif est un sous-ensemble des phrases du document, y compris les résultats de de la partie d'extraction de phrase, pour cela on se passe par Encoder – TRANSFORMER, Encoder – GRU-RNN. Ils ont évalué les deux ensembles des données de la norme par ROUGE-1, ROUGE-2 et ROUGE-L. Il calcule le chevauchement de n-grammes approprié entre la référence et le résumé du système. Leur modèle est facile à former et l'intuition / hypothèse derrière la formule est simple et logique.

1.8.7 Neural Diverse Paraphrastic Compression model (DPC) [Mir Tafseer 2019]

Dans ce travail, ils ont contribué à un nouveau modèle de compression d'endurance qui utilise le modèle de décodeur codeur neuronal séquence à séquence pour générer des phrases compressées avec diverses interprétations. Ils ont généré diverses compressions abstraites au niveau des phrases non traitées dans le passé, certaines opérations ont été effectuées travail de recherche. Leurs modèles améliorent la couverture des informations et l'abstraction des phrases générées. Ils ont exprimé des ensembles de données de compression de phrases abstraites générés artificiellement et évaluent leur système dans des nouvelles recommandations pour les mesures d'évaluation de la traduction automatique (MT). Leur expérience montre qu'à travers différents indicateurs, ces méthodes sont considérablement améliorées par rapport aux méthodes avancées. Leur modèle de compression paraphrastique neuronale diversifiée est basé sur la traduction automatique neuronale (NMT). DPC utilise NMT pour convertir les phrases sources en compression abstraite. Étant donné la phrase source $X = (x_1, x_2, ..., x_N)$, leur modèle apprendra à prédire une cible de compression abstraite diverse $Y = (y_1, y_2, ..., y_M)$, où $M < N$ est dérivé. La source X donnée par la cible Y est un problème d'apprentissage séquence à séquence typique, qui peut être modélisé à l'aide d'un modèle codeur-décodeur basé sur l'attention. Comme son nom l'indique, la forme de base du modèle codeur-décodeur se compose de deux parties. Dans leur cas, ils ont utilisé l'encodeur GRU bidirectionnel (Bi-GRU). Une autre modification importante qu'ils ont appliqué à Bi-GRUs est d'empiler plusieurs couches ensemble. Ensuite, Ils ont utilisé le modèle COPYNET, qui peut intégrer le mode normal

de génération de mots dans le décodeur avec un nouveau mécanisme de copie qui peut sélectionner des mots ou des sous-séquences de la séquence d'entrée et les placer dans la séquence de sortie. Dans autre part, ils ont utilisé l'intégration fastText pour créer une table d'alignement des mots du vocabulaire. Les auteurs de ce travail ont utilisé l'ensemble de données MSR-ATC. Et pour l'évaluation de leur système automatiquement, ils ont travaillé avec diverses mesures automatiques telles que BLEU, SARI et METEOR-E.

- **Comparaison entre les travaux précédents**

Dans ce qui suit, nous allons présenter un tableau comparatif pour les travaux précédents que nous avons déjà expliqués, à partir de certaines caractéristiques telles que le type de document unique ou multiple, le type d'approche et les ensembles des données. Ensuite, les méthodes d'évaluation et autre technique utilisé (voir le tableau 1).

Référence	**Type Document**	**L'approche**	**Data set**	**Méthodes Evaluation**	**Techniques utilisé**
[ARIES 2013]	Mono-document et multi-document	Extractive	Cmp-lg	ROUGE-1 ROUGE-2 ROUGE-SU4	OpenNLP, Porter-stemmer, NaïveBayes, uni-gramme, bi-gramme, UNIS
[Boudraf 2017]	Mono-document et multi-document	Extractive	Lucen, Corpus de test contient 32 documents	le corpus de jugement, ROUGE basé sur : rappel, précision et F-score.	IntellexerSummarizer Les méthodes : - base de mots clés. - base de position. - la longueur de phrase. - base d'expressions indicatives.

					- basée sur les relations
[Bensidiaissa 2020]	Mono-document et multi-document	Extractive et Abstractive	CNN/DailyMail, AESLC, Multi-news, Gigaword, DUC 2004	ROUGE-1 ROUGE-2 ROUGE-L	K-signifie, BERT
[AID 2020]	Mono-document et multi-document	Extractive	CORD-19	ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-W, Les mesures : Precision, Recall, et F-measure	Okapi BM25, WordNet, TextRank, WUP_similarity
[Shi 2020]	Mono-document	Abstractive	CNN / Daily Mail, Newsroom, Bytecup	ROUGE-1, ROUGE-2, ROUGE-L	RNN séquence à séquence, CoreNLP, encodeur-décodeur
[Elozino 2019]	Mono-document	Extractive et Abstractive	CNN / Daily Mail, Newsroom	ROUGE-1, ROUGE-2, ROUGE-L	codage-codage-décodage, séquence à séquence, TRANSFORMEUR, convertisseur
[Mir Tafseer 2019]	Mono-document	Abstractive	MSR-ATC	BLEU, SARI, METEOR-E	Décodeur-codeur Neuronal séquence à séquence, Bi-GRU, COPYNET, fastText

Tableau 1.1: Comparaison entre les travaux connexes

1.9 Conclusion

Dans ce chapitre, nous avons présenté l'état de l'are de résumé automatique de texte. Tout d'abord, pour comprendre ce domaine, nous avons introduit quelques concepts de résumé automatique. Il peut appartenir à plusieurs classes ou types, qui passent par différentes étapes et utilisent différentes méthodes. Par conséquent, le système de résumé automatique est un domaine très important qui peut nous aider pour accéder rapidement à des résumés précis et faciles à comprendre. Et finalement, nous avons évoqué quelques travaux antérieurs dans le domaine de résumé automatique de texte, soit extractif ou bien abstractif, mono ou multiple document. En plus, nous avons fait un tableau comparatif entre ces travaux.

Chapitre 2 Apprentissage en Profondeur

2.1 Introduction

Dans le chapitre précédent nous avons fait un état de l'are, ceci est un aperçu du résumé de texte automatique, Pour que nous puissions le connaître et parler de ses types, ses étapes, l'évaluation et les travaux connexes dans ce domaine. Afin d'atteindre deux objectifs principaux dans le domaine de résumé automatique de texte, le premier objectif est d'obtenir un texte plus clair et plus précis, facile à comprendre et bien structuré, et le deuxième objectif est d'éviter la difficulté de lire trop de texte. Dans ce chapitre nous expliquons les conceptes utilisé dans le cadre d'apprentissage en profondeur.

2.2 L'apprentissage en profondeur

L'apprentissage profond est censé améliorer considérablement les tâches avancées d'intelligence artificielle telles que la détection d'objets, la reconnaissance vocale et la traduction automatique. La nature architecturale profonde de cette technique peut être utilisée pour résoudre des problèmes complexes liés à l'intelligence artificielle. Ainsi, les chercheurs ont utilisé cette méthode dans des domaines modernes pour de nombreuses tâches telles que la détection d'objets et la reconnaissance de visages. L'application de cette méthode à de nombreux modèles de langage a également été réalisée. Par exemple, les réseaux neuronaux récurrents ont été utilisés pour débruiter les signaux vocaux et les autoencodeurs empilés ont été utilisés pour déterminer le modèle de regroupement pendant l'expression des gènes. Une autre étude a utilisé le modèle neuronal pour produire des images de styles différents. En outre, la technologie d'apprentissage profond a été utilisée pour analyser simultanément des sentiments provenant de plusieurs modalités. [Wang 2017]

La technologie d'apprentissage profond a connu des développements massifs au cours des dernières années. Sur la base de résultats empiriques, il a été déterminé que cette technique était meilleure que d'autres algorithmes ML. Cela pourrait être dû au fait que cette technique, comme le modèle céré-

bral, copie le fonctionnement du cerveau et empile plusieurs couches de réseaux neuronaux les unes sur les autres. a déclaré que les machines d'apprentissage profond sont plus performantes que les outils ML conventionnels car elles utilisent également la méthode d'extraction de caractéristiques. Cependant, jusqu'à présent, il n'existe aucun fondement théorique pour la technologie d'apprentissage profond. Les hiérarchies de caractéristiques sont apprises par les techniques d'apprentissage profond en utilisant les caractéristiques obtenues à partir des niveaux hiérarchiques supérieurs, qui ont été formés par l'organisation des caractéristiques de bas niveau. Les caractéristiques d'apprentissage trouvées aux différents niveaux d'abstraction permettent au système de prendre conscience des fonctions complexes qui utilisent les données pour mettre en correspondance l'entrée et la sortie résultante sans dépendre des caractéristiques développées par l'homme. [Wang 2017]

La principale différence entre les technologies d'apprentissage profond et la ML est la différence de leurs performances lorsque le volume de données augmente. Lorsque l'ensemble de données est plus petit, la méthode d'apprentissage profond a une performance inefficace car elle a besoin d'un grand volume de données pour une bonne compréhension. [Wang 2016]

2.3 Réseau Neuronal Convolutionnel

Le réseau neuronal convolutif (CNN) est un type de réseau profond à action directe qui peut être généralisé et formé facilement par rapport à d'autres réseaux qui possèdent une connectivité entre les couches adjacentes. Le CNN a été utilisé avec succès lorsque d'autres réseaux neuronaux n'étaient pas aussi populaires. Actuellement, il est utilisé dans la communauté de la vision par ordinateur. [Sainath 2013]

Les CNN sont formulés pour le traitement de données sous forme de tableaux multiples, comme une image en échelle de gris composée de tableaux 3×2D avec des intensités de pixels variables. Différentes modalités de données sont présentées sous forme de tableaux multiples, comme 1D pour les signaux et les séquences, y compris le langage ; 2D pour les spectrogrammes d'images ou audio ; et 3D pour les images vidéo ou volumétriques. Les quatre idées principales qui permettent aux CNN d'utiliser les caractéristiques des signaux naturels sont le partage des poids, la mise en commun, les connexions locales et l'utilisation des couches multiples. [Sainath 2013]

De nombreuses étapes sont incluses dans une architecture CNN classique (voire la figure suivante). Les étapes initiales consistent en deux types de couches : les couches de mise en commun et les couches convolutives. Dans la couche convolutive, on peut organiser les couches en cartes de carac-

téristiques, où chaque unité est connectée aux patchs locaux des cartes de caractéristiques, qui proviennent des couches précédentes, par des poids appelés banque de filtres. Le résultat de la somme pondérée locale passe par la non-linéarité, comme le ReLU. Toutes les unités trouvées dans la carte de caractéristiques sont observées comme partageant un banc de filtres. Les différentes cartes de caractéristiques trouvées dans la couche utilisent des banques de filtres différentes. Cette architecture a été construite pour deux raisons. Initialement, pour les données de tableau comme les images, il a été considéré que les groupes locaux de valeurs sont fortement corrélés. Ils forment également des motifs locaux uniques et facilement perceptibles. Deuxièmement, les statistiques locales des autres signaux ou images sont considérées comme invariantes par rapport à l'emplacement. Ainsi, si le motif est observé dans une certaine section de l'image, on peut également le trouver ailleurs. Ce réseau repose donc sur le fait que les unités trouvées à divers endroits partagent les mêmes poids et peuvent donc être détectées par l'utilisation de motifs similaires provenant des autres segments du réseau. Mathématiquement, la convolution discrète est considérée comme la principale opération de filtrage mise en œuvre dans les cartes de caractéristiques, d'où son nom. [Nair 2010]

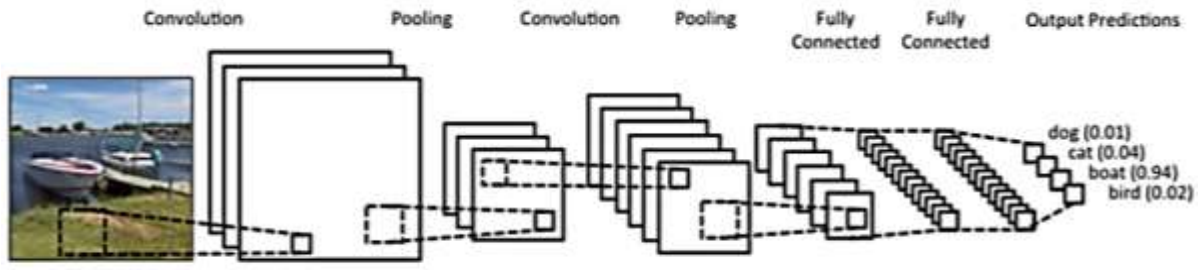
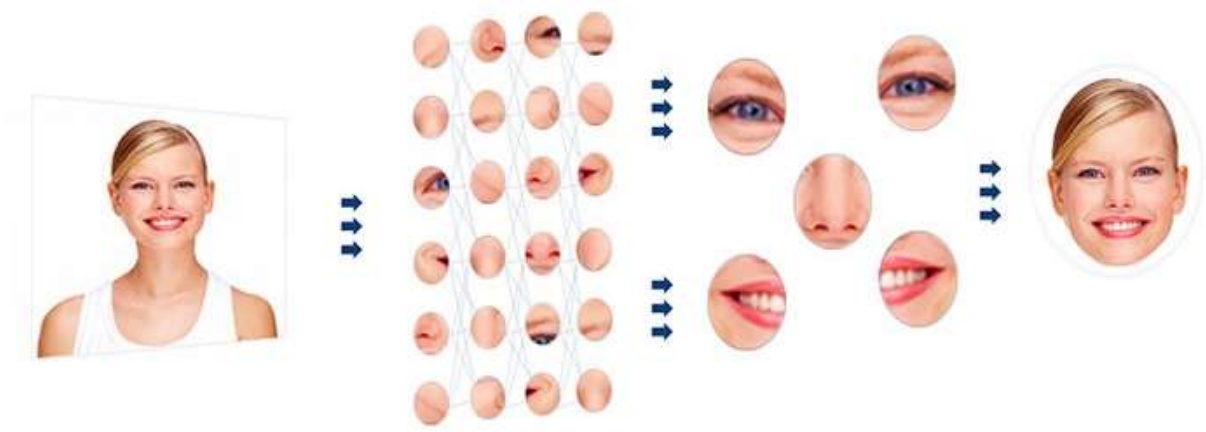


Figure 2.1 : Architecture du CNN pour la classification d'images

Alors que la couche convolutionnelle utilise la couche précédente comme base pour détecter la combinaison locale des caractéristiques, la couche de mise en commun combine les caractéristiques sémantiquement similaires en une seule caractéristique. En raison de la position relative de ces caractéristiques, il peut y avoir des variations dans la formation du motif. En outre, il est possible de détecter le motif de manière fiable en effectuant un grossier travail sur sa position dans chaque caractéristique. L'unité de mise en commun générale a la capacité de calculer une quantité maximale de la parcelle locale des unités en une seule carte de caractéristiques. [Chen 2016]

La figure 2.2 montre que pour classer l'image, la technique CNN détecte les bords à partir des pixels bruts de la couche 1. Ensuite, elle utilise les bords pour détecter les formes simples dans la couche 2. Ensuite, ces formes sont utilisées pour détecter les formes plus simples dans la couche

2. Ces formes sont également utilisées pour déterminer les caractéristiques de haut niveau, telles que la forme du visage dans les couches supérieures. La dernière couche est le classificateur, qui utilise ces caractéristiques de haut niveau. [Chen 2016]



.

Figure 2.2 : Méthode d'extraction de caractéristiques faciales basée sur l'apprentissage profond d'Eyeris (Chen, Yu-Hsin et Krishna, Tushar et Emer, Joel et Sze, Vivienne 2016)

2.4 L'architecture du CNN :

Les CNN sont constitués de couches cachées de sortie, d'entrée et d'autres couches cachées. Comme le montre la figure 4.1, ces couches cachées peuvent être soit des couches de mise en commun, soit des couches convolutionnelles, soit des couches entièrement connectées. [Angermueller 2016]

2.4.1 Couches convolutionnelles

Les couches convolutionnelles utilisent l'opération de convolution pour l'entrée. La sortie est ensuite transmise à la couche suivante. La réponse donnée par les neurones aux stimuli visuels est imitée. La couche convolutive est composée de plusieurs cartes neuronales, appelées cartes de caractéristiques ou filtres. Sa taille est similaire aux dimensions de l'image d'entrée. Deux concepts permettent de diminuer la quantité de paramètres du modèle, à savoir le partage des paramètres et la connectivité locale. Premièrement, contrairement aux réseaux entièrement connectés, chaque neurone de la carte de caractéristiques peut avoir une connexion avec le patch neuronal local de la couche précédente, c'est-à-dire le champ réceptif. Deuxièmement, il a été observé que les neurones de la carte des caractéristiques partagent des paramètres similaires. Ainsi, tous les neurones de la carte de caractéristiques recherchent des caractéristiques similaires dans différentes zones de l'image au sein des couches précédentes. Les différentes cartes de caractéristiques ont la capacité de

détecter des bords avec des orientations variables dans l'image, des motifs de séquence variables dans la séquence génomique. On peut déterminer l'activité neuronale en calculant la convolution discrète du champ réceptif. Cette dernière calcule à son tour la somme pondérée des neurones d'entrée et met en œuvre la fonction d'activation. La prochaine figure élucide la couche de convolution discrète. [Angermueller 2016]

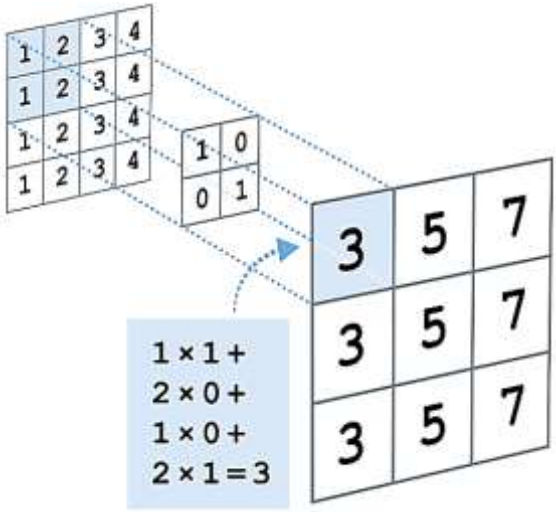


Figure 2.3: La convolution discrète est la première couche CNN.

Trois hyperparamètres sont utilisés pour contrôler la taille de la sortie, c'est-à-dire le pas, la profondeur et le remplissage du zéro.

1. Profondeur : Il s'agit de la quantité de filtres que l'on peut utiliser dans une seule image d'entrée. Ces filtres sont utiles pour détecter des structures telles que des taches, des coins, des bords, etc.
2. Stride : Il s'agit de la quantité de pixels que le filtre saute lorsqu'il glisse sur une image.
3. Remplissage par des zéros : Il s'agit du remplissage de zéros autour du bord d'une image d'entrée afin de préserver sa taille.

2.4.2 Couche d'unités linéaires rectifiées (ReLU)

Conventionnellement, à la fin de chaque couche convolutive, il y a une application immédiate de la couche non linéaire (ou couche d'activation). Cette couche présente une non-linéarité à l'intérieur du système. Cette non-linéarité calcule les opérations linéaires dans les couches convolutionnelles. Auparavant, l'application de fonctions non linéaires comme tanh et sigmoïde était effectuée. Cependant, les chercheurs ont déterminé que les couches ReLU étaient plus efficaces car elles permettaient d'entraîner rapidement le réseau (grâce à l'efficacité du calcul) sans avoir d'influence sur sa précision. Cela a également permis de résoudre le problème de la disparition du gradient, qui fait que les couches inférieures du réseau sont entraînées lentement en raison d'une diminution ex-

ponentielle du gradient dans les différentes couches. Cette couche ReLU a implémenté la fonction f(x) = max (0, x) aux valeurs du volume d'entrée. Par conséquent, cette couche peut être utilisée pour changer l'activation négative en 0. En outre, elle a augmenté les propriétés non linéaires du réseau complet et du modèle appliqué sans affecter les champs réceptifs dans la couche de convolution. Cette fonction d'activation ReLU est représentée à la figure 2.4. [Nair 2010]

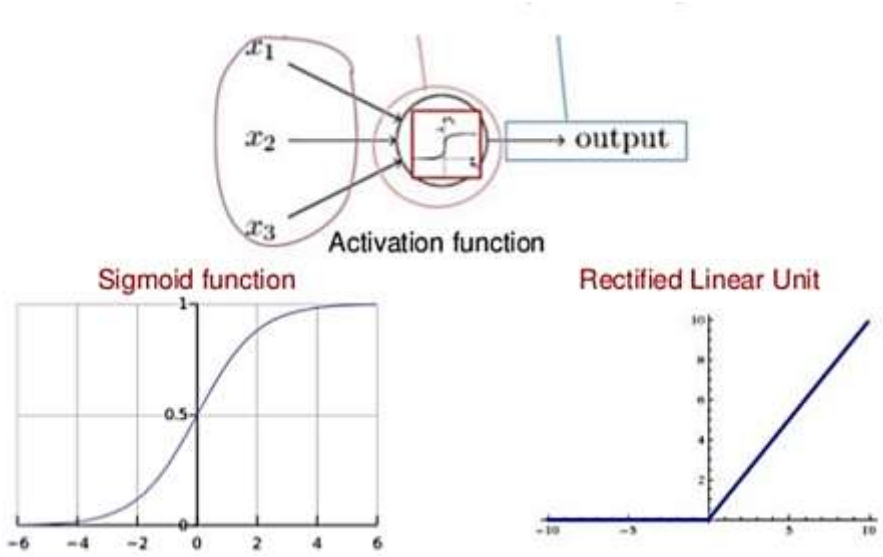


Figure 2.4: Fonction d'activation du ReLU

2.4.3 Couches de mise en commun (Pooling layers)

La couche de mise en commun réduit la taille de l'entrée et permet d'effectuer une analyse multi-échelle. Le max-pooling et l'average-pooling sont inclus dans les opérateurs de pooling standard. Ces opérateurs sont utiles pour calculer une valeur moyenne ou maximale dans un petit bloc spatial. La figure 2.5 décrit l'opération de regroupement max à travers les filtres 2 × 2.

Dans de nombreuses applications, la fréquence et l'emplacement exact des caractéristiques n'ont pas d'importance pour la prédiction finale, par exemple lorsqu'il s'agit de reconnaître des objets dans une image. Compte tenu de ces hypothèses, les couches de mise en commun offrent un résumé des neurones adjacents après le calcul, semblable à la valeur moyenne ou maximale qui est supérieure à leur activité. Cela permet d'obtenir une représentation lisse de leurs activités caractéristiques. Lorsque la mise en œuvre de cette opération de mise en commun est effectuée sur de plus petits patchs d'image qui se décalent de plus d'un pixel, l'image d'entrée subit un sous-échantillonnage efficace, ce qui réduit les paramètres du modèle. [Krizhevsky 2012]

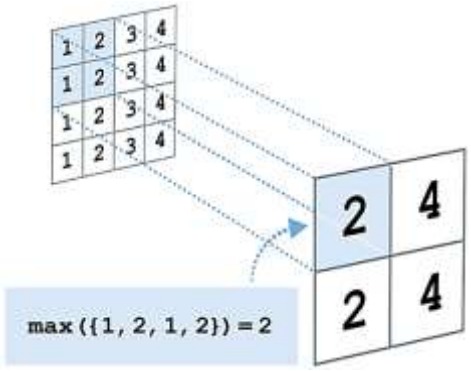


Figure 2.6: Opération de mise en commun maximale à l'aide des filtres 2×2

2.4.4 Couches Entièrement Connectées (Fully-connected layers)

Le CNN est composé de plusieurs couches de mise en commun et de convolution, qui permettent d'identifier les caractéristiques abstraites à une échelle croissante, en commençant par les petits bords, puis les composants de l'objet, jusqu'à l'objet final. Après la couche finale de mise en commun, une ou plusieurs couches entièrement connectées suivent. Les hyperparamètres du modèle, tels que le nombre de couches convolutionnelles, le nombre de cartes de caractéristiques et la taille du champ réceptif, dépendent de l'application. Ils sont également choisis dans l'ensemble de données de validation. [Angermueller 2016]

Les couches entièrement connectées sont reliées aux neurones de la couche précédente. Ces couches entièrement connectées servent de couche finale du réseau et font également partie de la classification. La figure 2.7 présente un exemple de CNN qui élucide les 3 couches. [Angermueller 2016]

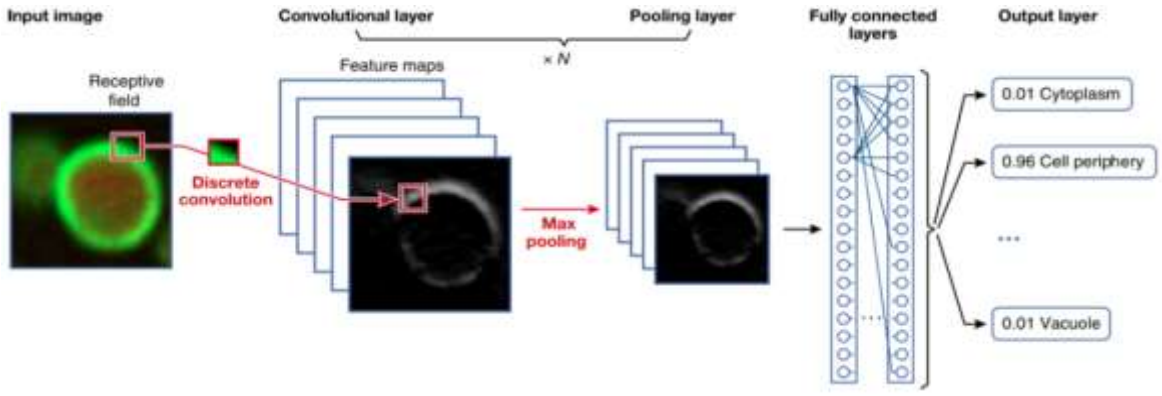


Figure 2.7: Un exemple décrivant l'ensemble de l'architecture CNN

2.5 Conclusion

Au niveau de ce chapitre, nous avons vu ci quoi l'apprentissage en profondeur puis Réseau Neuronal Convolutionnel et son architecture, pour que nous utilisions couches convolutionnelles et max pooling dans notre solution proposé au cours du prochain chapitre.

Chapitre 3 La Solution Proposée

3.1 Intorduction

Dans le chapitre précédent, Nous avons donné un aperçu d'apprentissage en Profondeur et Réseau Neuronal Convolutionnel. Dans ce chapitre, nous presentons notre solution et la nouvelle architecture proposée pour ameliorer la qualite du resumé de text.

3.2 L'architecture proposée

Notre système est un système de résumé automatique de texte en anglais basé principalement sur des techniques d'extraction. L'architecture proposée utilise les concepts de machine et d'apprentissage en profondeur. La figure suivante montre l'organigramme de cette proposition qui passe par certaines étapes pour obtenir un résumé.

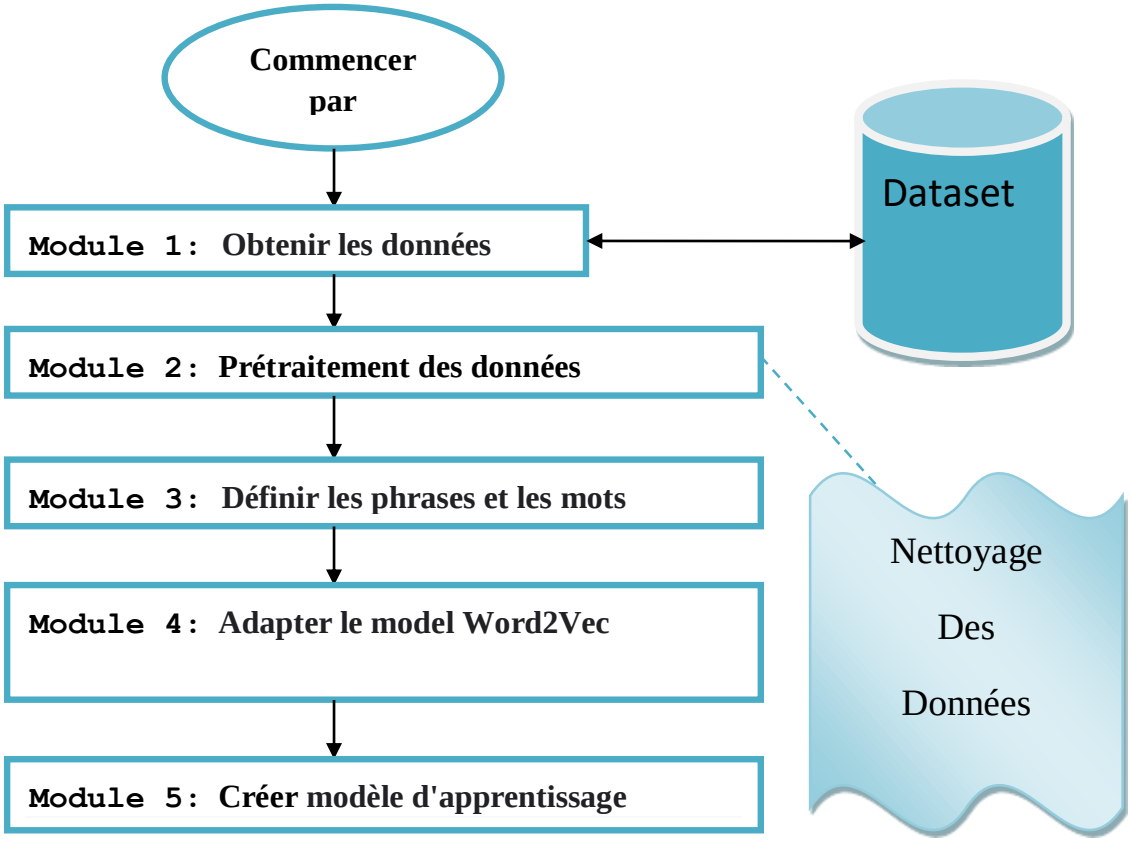


Figure 3-1 : Organigramme de l'architecture proposée

3.2.1 Module 1 : Obtenir les données

La première étape consiste à télécharger la base de données Amazon Food_Reviews qui contient 10 colonnes et 500 000 lignes, Lorsque nous téléchargeons cet ensemble de données, nous obtenons un fichier CSV nous prélevons de ces données 2 colonnes, Colonne de texte en lignes, Colonne de résumé en lignes. L'aperçu d'un exemple de données est présenté dans la figure suivante.

Figure 3-2 : Un exemple des données avant le nettoyage

3.2.2 Module 2 : Prétraitement des données

La phase de nettoyage comprend la suppression des mots vides, la suppression des entrées vides et le remplacement du rétrécissement Grâce à leurs formes plus longues, les contractions doivent être converties dans leurs formes longues d'origine, par exemple, « don't » doit être reconverti en « do not », supprimez les caractères inutiles comme les symboles et les émoticônes, toutes les écritures doivent être en minuscules. La figure suivante représente un exemple de nettoyage des données.

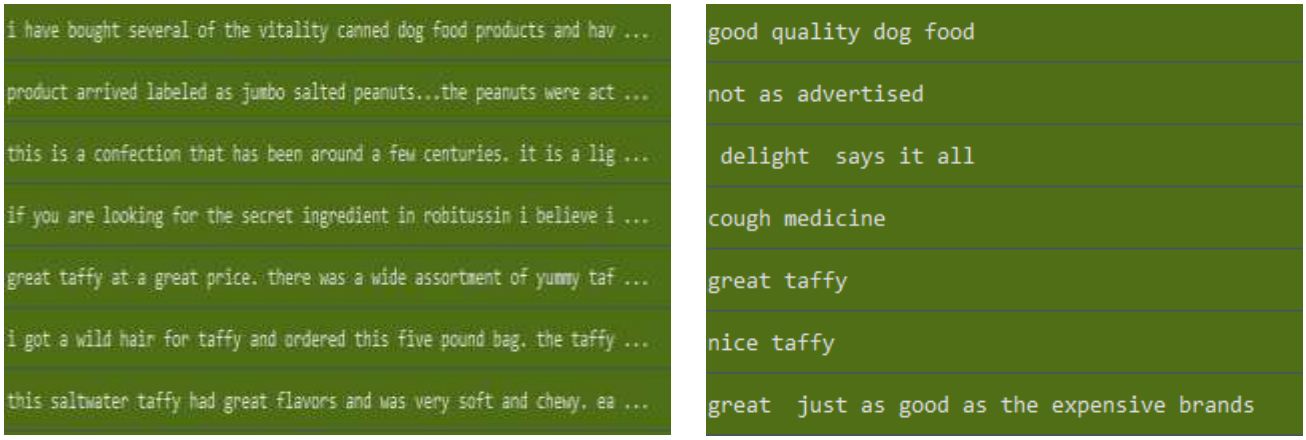


Figure 3-3 : Un exemple de données après le nettoyage

3.2.3 Module 3 : Définir les phrases et les mots

Une fois les données nettoyées, Ensuite, nous divisons chaque texte en phrases, et considérons chaque résumé comme une seule phrase. Ainsi, nous divisons chaque phrase du texte et chaque résumé en mots. La figure suivante est un exemple de ce qui est présenté après la division en phrases et en mots.

Algorithme 1 : Division de texte et résumé en phrases

Entrée: clean_summaries, clean_textsss
Sortie: texte est divisé en phrases, résumé est divisé en phrases
Début :
SSS ← ()
Pour summary dans clean_summaries **faire**
summary ← Remplacer (summary, '.', ' ')
SSS ← Découper (summary)
Fin Pour
T ← ()
TTS ← ()
Pour j dans longueur (clean_textsss) ; incrémenter de 1 **faire**
Pour text dans clean_textsss(j) **faire**
text ← Remplacer (text, '.', ' ')
T ← Découper (text)
Fin Pour
TTS() ← T
T ← ()
Fin Pour
Fin

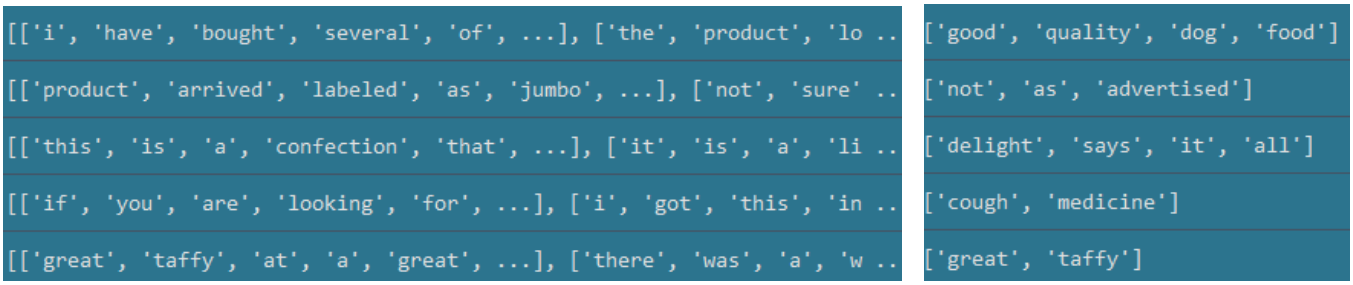


Figure 3-4 : Un exemple après la division en phrases et mots

Après cela, on répète chaque résumé en fonction du nombre de phrases dans chaque texte, pour obtenir au final chaque phrase de chaque texte lui correspondant le résumé du texte, la figure suivante illustre un exemple de cette étape.

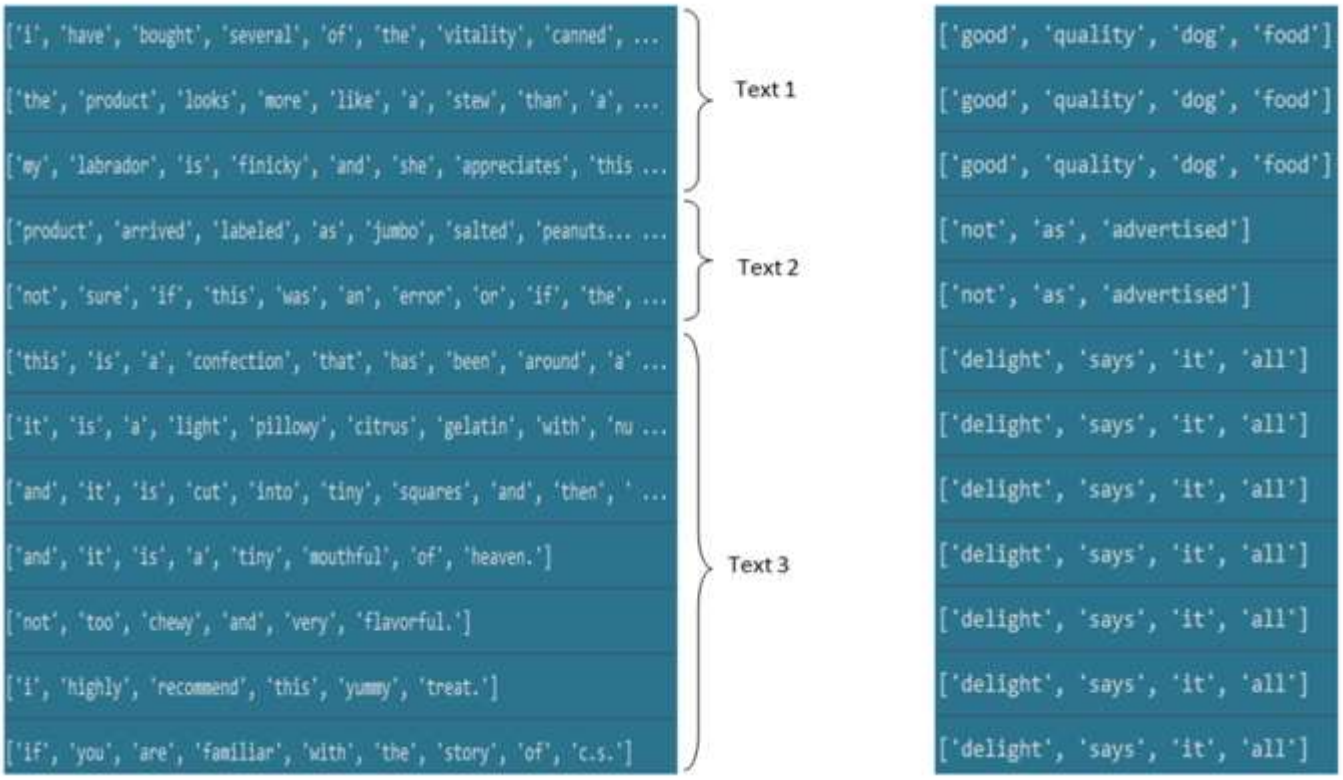


Figure 3-5 : Un exemple de répétition de résumé en fonction de nombre de phrases

3.2.4 Module 4: Adapter le modèle Word2Vec

Dans ce module, nous passons à l'étape de la conversion en vecteur utilisons l'intégration de mots (Word Embbiding) spécialement Word2Vec, pour cela nous avons téléchargé Twitter Glove Word2Vec 50 et avant d'entrer dans les details de notre module, nous allons voir qu'est-ce que l'incorporation de mots Word Embeddings ? Pourquoi l'utiliser? et beaucoup plus.

- Word Embeddings

Voyons quelques exemples d'abord :

1. Il existe de nombreux sites Web qui nous obligent à commenter leurs produits lors de leur utilisation, par exemple: -Amazon, IMDB.
2. nous avons aussi l'habitude de faire des recherches sur Google avec quelques mots et d'obtenir les résultats correspondants.

Comment tout cela est-il fait ? Ces choses sont des applications du traitement de texte. Nous utilisons le texte pour faire l'analyse des sentiments, le regroupement de mots similaires, la classification des documents et le marquage. Lorsque nous lisons un journal, nous pouvons dire de quoi il s'agit, mais comment l'ordinateur va-t-il faire ces choses ? L'ordinateur peut faire correspondre des

chaînes de caractères et nous dire si elles sont identiques ou non, mais comment faire pour que l'ordinateur nous parle de Deep leanrning ou de Neural Network lorsque nous recherchons seq2seq?

Pour des tâches comme la reconnaissance d'objets ou de la parole, nous savons que toutes les informations requises pour la tâche est encodée dans les données (car les humains peuvent effectuer ces tâches à partir des données brutes). Cependant, les systèmes de traitement du langage naturel traitent traditionnellement les mots comme des symboles atomiques discrets. Ainsi, "cat" peut être représenté par Id537 et "dog" par Id143. Ces codages sont arbitraires et ne fournissent aucune information utile au système concernant les relations qui peuvent exister entre les symboles individuels.

Il existe deux types fondamentaux d'incorporations de mots:

1. Incorporation basée sur la fréquence.
    a. Vecteur de comptage.
    b. Vectorisation TF-IDF.
    c. Matrice de cooccurrence avec une fenêtre contextuelle fixe.
2. Embeddings basés sur la prédiction
    a. Sac de mots continu (Continuous Bag of words (CBOW))
    b. Modèle Skip-gram

- Word2Vec

En intelligence artificielle et en apprentissage automatique, Word2vec est un ensemble de modèles d'intégration lexicale Word embedding, cet algorithme est parmi les plus connus. Il a été développé par une équipe de recherche de Google sous la direction de Tomas Mikolov en 2013 [Mikolov et al, 2013]. Il s'appuie sur un réseau de neurones à deux couches et essaie d'apprendre la représentation vectorielle des mots qui composent le texte afin que les mots qui partagent un contexte similaire soient représentés par des vecteurs numériques proches. Cette méthode est appliquée dans la bibliothèque Python Gensim[14]. Word2Vec possède deux architectures neuronales, le modèle de sacs de mots continus (CBOW: Continuous Bag Of Words) et le modèle skip-gram. Ils sont composés de trois couches : une couche d'entrée, une couche cachée et une couche de sortie. La couche d'entrée contient soit un "sac-de-mots" (CBOW) ou bien un mot seul (Skip-gram). La couche cachée correspond à la projection des mots d'entrée dans la matrice des poids. Cette matrice est partagée par tous

[14] Est une bibliothèque logicielle Python pour topic modelling. En apprentissage automatique et en traitement automatique du langage naturel, un topic model (modèle thématique ou « modèle de sujet ») est un modèle probabiliste permettant de déterminer des sujets ou thèmes abstraits dans un document.

les mots (matrice globale). Enfin, la couche de sortie est composée de neurones “softmax”. [Killian et al.. 2015]

Maintenant nous retournons à notre module, pour appliquer Word2Vec dans notre cas nous passons par les étapes suivantes :

- ✓ **Conversion de mot en vecteur**

L'outil Word2Vec prend un corpus de textes en entrée et produit les vecteurs de mots en sortie. Il construit d'abord un vocabulaire à partir des données du texte d'entraînement, puis apprendre la représentation vectorielle des mots.

Donc après avoir passé par Word2Vec, Nous avons obtenu un vecteur de 50 segments pour chaque mot de notre ensemble de données. Dans ce qui suit, un exemple d’une phrase après passé par Word2Vec, (voire la figure suivante).

Algorithme 2 : Vérifier l'existence des données dans Word2Vec et les convertir en Vecteur

```
Entrée: summary SSS, text TTS, devided_textes  matrix, clean_summaries, ensemble
        word2vec ('glove-twitter-50')  wv
Sortie: text_w2v , summary_w2v
Variable : i, j : entier
TS ← ()
TS_str ← ()
SS_str ← ()
SS ← ()
text_w2v ← ()
summary_w2v ← ()
    Début :
        Pour i dans  longueur(TTS); incrémenter de 1 faire
            K← ()
            b← SSS(i)
            Si (longueur (b) < 51) et (longueur (b) > 1) alors
            SS () ← b
            y ← ()
            Pour j dans longueur( b) ; incrémenter de 1 faire
                f← b(j)
                Si f dans  wv alors
                    Y() ← wv(f)
                Fin Si
            Fin Pour
            Pour j dans longueur (TTS(i)) ; incrémenter de 1 faire
                a ← TTS (i,j)
                Si (longueur (a) < 51) et (longueur (a) > 4) alors
                    K() ← a
                    TS_str () ← devided_textes  (i,j)
                    SS_str () ← clean_summaries (i)
                    x ← ()
                    Pour r dans longueur (a) ; incrémenter de 1 faire
                        u ← a(r)
                        Si u dans  wv alors
                            x() ← wv (u)
                        Fin Si
                    text_w2v () ← x
                    summary_w2v () ← y
                    Fin Pour
                Fin Si
            TS () ← k
            Fin Pour
Fin
```

Figure 3-6 : Un exemple d'une phrase après passé par Word2Vec

✓ **Conversion de phrase en matrice**

Nous convertissons la phrase la plus longue dans notre ensemble de données (calcule nombre de mots de chaque phrase et en prend la phrase qui contient le maximum) en une matrice de max sur 50 comme nous avons expliqué par un exemple dans la figure suivante.

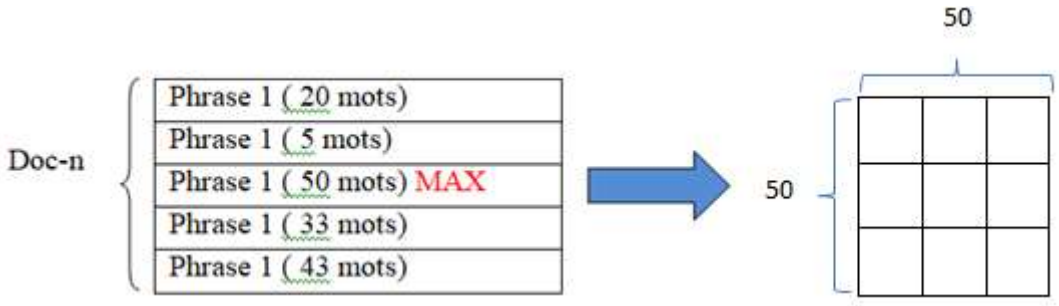


Figure 3-7 : Exemple de Conversion de phrase en matrice.

Après avoir calculé le nombre de mots de chaque phrase et obtenir le nombre maximum, par exemple 50 dans notre cas, donc Nous transformons toutes les phrases de texte et résumé en des matrices de dimension (50, 50) et remplir le vide par des zéro, par exemple si on a matrice de dimension (23,50) on va remplir la matrice par des zéros jusqu'à (50 ,50) (Un exemple dans la figure ci-dessous).

Algorithme 3 : La conversion des phrases en matrices

```
Entrée: textes, résumés
Sortie: matrices de textes, matrices de résumés
Variable : i : entier
texts_w2v ← ()
summarys_w2v ← ()
final_texts_str ← ()
final_ summarys_str ← ()
   Début :
        Pour i dans  longueur(texts_w2v); incrémenter de 1 faire
            Création de matrice s de (summarys_w2v(i))
            Création de matrice a de (texts_w2v (i))
            Si (longueur (a) < 51 et longueur (a) > 0) et (longueur (s) < 51 et longueur (s) > 0)  alors
              o ← pad (s, (50,50)) # pad est une fonction pour construire une matrice des zéros
              w ←pad (a, (50,50))
              texts_w2v () ← w
              summarys_w2v () ← o
              final_text_str ()← TS_str (i) # TS_str les phrases  du textes qui appartient dans le Word2vec
              final_ summary_str ()← SS_str(i)# SS_str les  résumés qui appartient dans le Word2vec
             Fin Si
          Fin Pour
          Ecrire (longueur (final_text_str)
          Ecrire (longueur (final_ summary_str)
          texts_w2v  ← matrice de (texts_w2v)
          Ecrire ( la forme de (texts_w2v))
          summarys_w2v ← matrice de (summarys_w2v)
          Ecrire ( la forme de (summarys_w2v))
      Fin
```

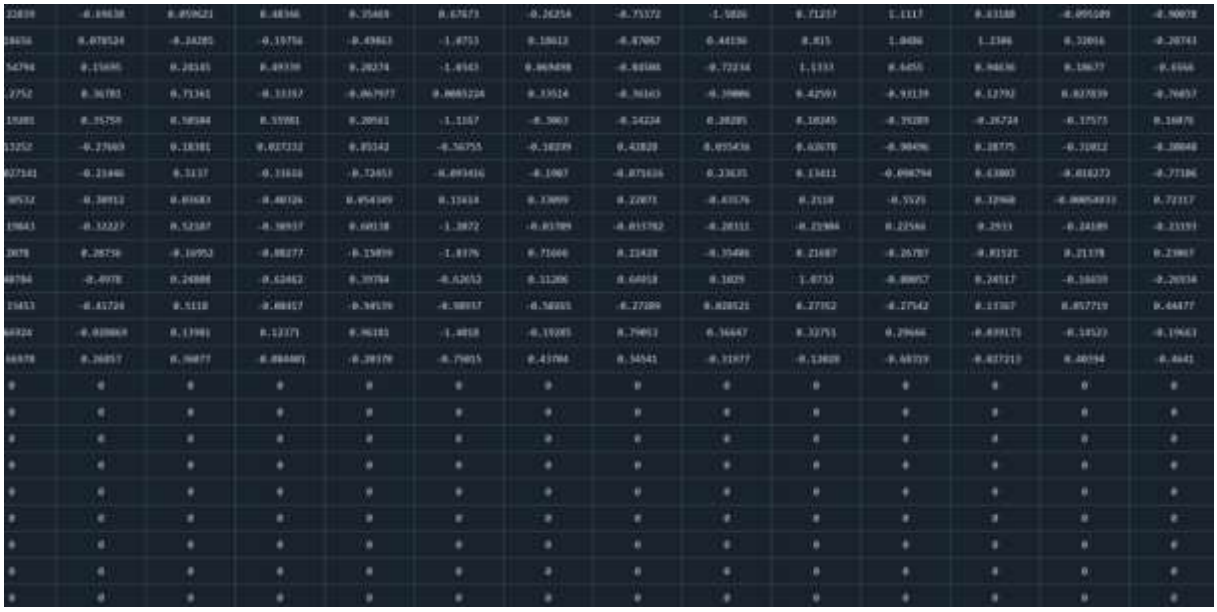


Figure 3-8 : Un exemple de phrase en matrice

Donc Nous avons transformé notre ensemble de données en un vecteur à 3 dimensions :

Nombre de phrase, nombre de mots (fixe 50), vecteur de 50 segments.

D'autre part on répète chaque résumé en fonction du nombre de phrases dans chaque texte, pour obtenir au final chaque phrase de chaque texte lui correspondant le résumé du chaque texte mais son devise chaque phrases en mots, un exemple est illustré à la figure qui suit.

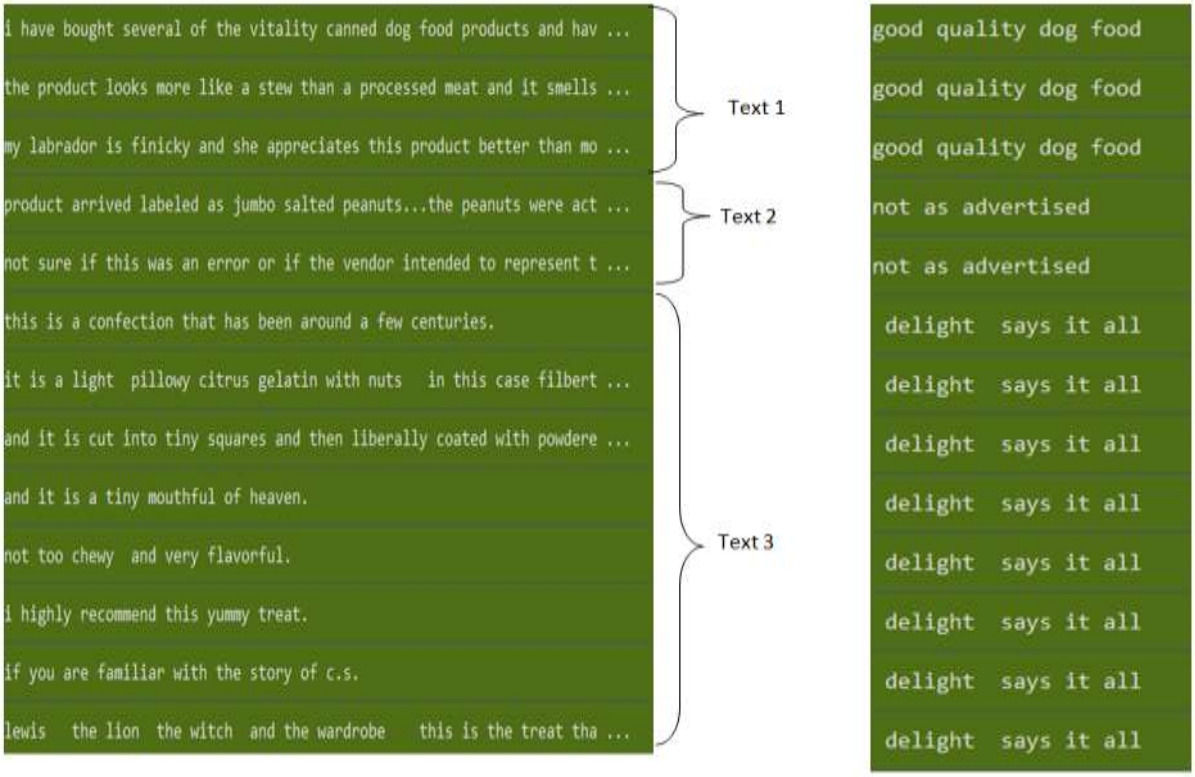


Figure 3-9 : Un exemple de répétition de résumé en fonction de nombre de phrases sans déviser en mots

A la fin, nous obtenons la matrice de phrase de texte, matrice de résumé, phrase de texte, résumé.

3.2.5 Module 5 : Créer Modèle d'apprentissage

Dans ce dernier module, nous avons appliqué deux modèles : auto-encoder et Multi-layer perceptron**,** on prend les matrices textes comme des entrés et les matrices résumés comme des sorties. Pour s'entraîner à produire des résumés, nous passons par les étapes suivantes :

- Etape 1 :

Pour chaque document, on calcule latent convolution entre la matrice de chaque phrase dans le texte et la matrice de résumé de texte et en utilisant le model Auto-Encoder.

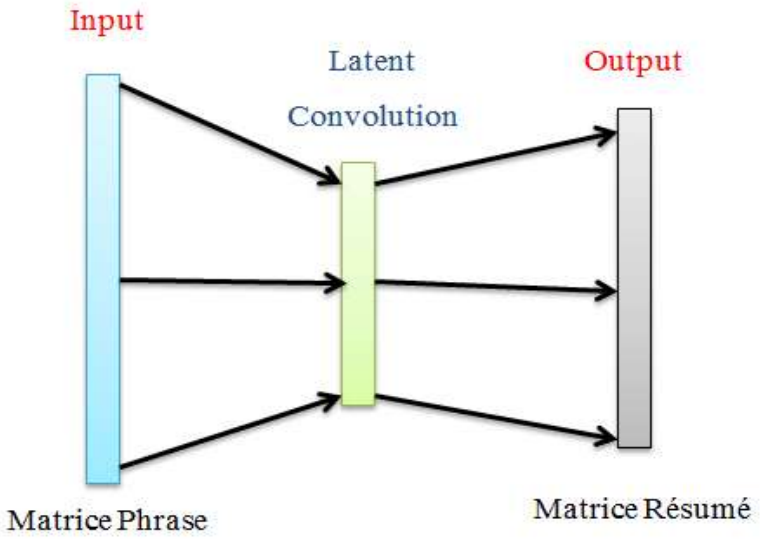


Figure 3-12 : l'étape de Latent Convolution

Voici le résumé d'entrainement de model Auto-Encoder

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 50, 50, 1)]	0
conv2d (Conv2D)	(None, 50, 50, 32)	320
max_pooling2d (MaxPooling2D)	(None, 25, 25, 32)	0
conv2d_1 (Conv2D)	(None, 25, 25, 16)	4624
max_pooling2d_1 (MaxPooling2	(None, 12, 12, 16)	0
conv2d_2 (Conv2D)	(None, 12, 12, 8)	1160
max_pooling2d_2 (MaxPooling2	(None, 6, 6, 8)	0
conv2d_3 (Conv2D)	(None, 6, 6, 8)	584
up_sampling2d (UpSampling2D)	(None, 12, 12, 8)	0
conv2d_4 (Conv2D)	(None, 12, 12, 16)	1168
up_sampling2d_1 (UpSampling2	(None, 24, 24, 16)	0
conv2d_5 (Conv2D)	(None, 24, 24, 32)	4640
up_sampling2d_2 (UpSampling2	(None, 48, 48, 32)	0
conv2d_transpose (Conv2DTran	(None, 50, 50, 1)	289

Total params: 12,785
Trainable params: 12,785

Figure 3-13 : le résumé d'entrainement de model Auto-Encoder

- Etape 2 :

Dans cet étape, nous prenons les textes comme des entrés et leur résumés comme des sorties, Pour connaître le pourcentage de similarité des phrase pour chaque document nous calculons BLEU[15] score qu'est la similarité entre les phrases de texte et le résumé. En obtenir pour chaque similarité un vecteur de 1 segment qui contient la valeur de similarité, (voir la figure suivante).

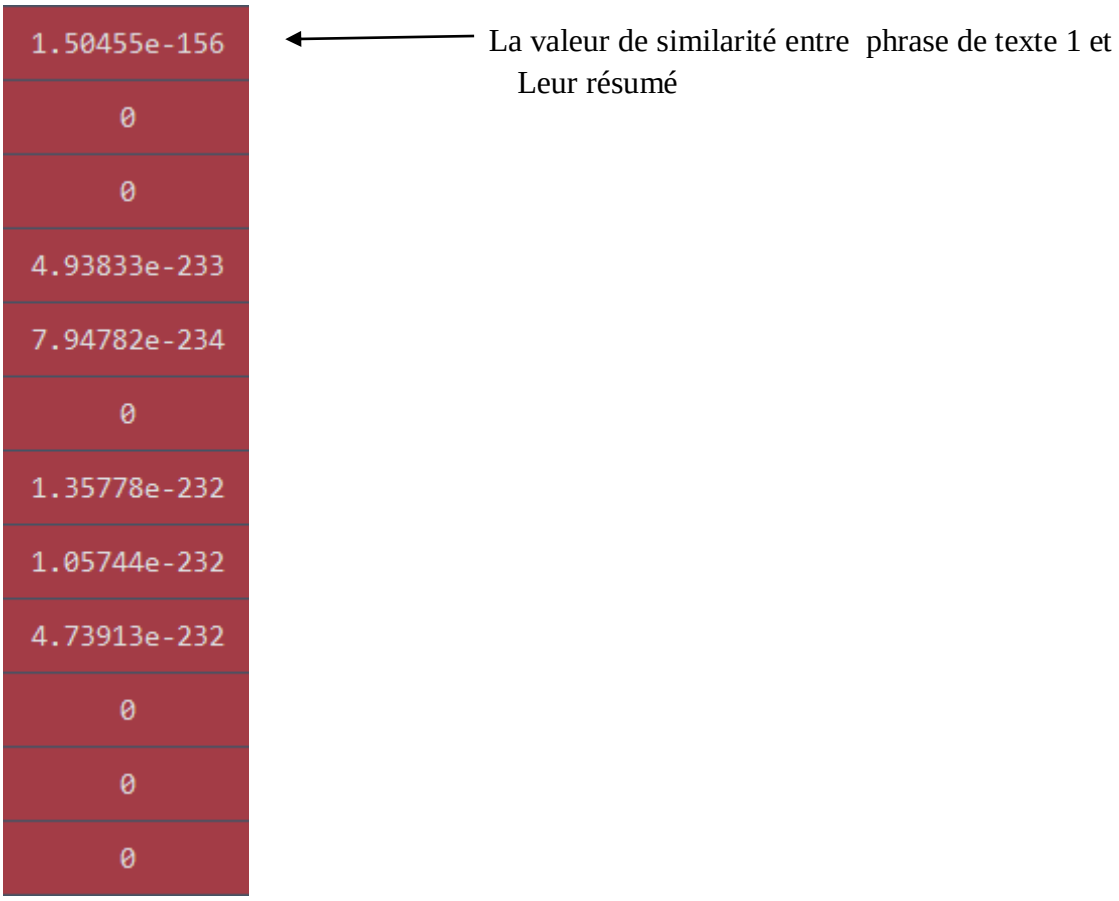


Figure 3-14 : Vecteur de similarité entre une phrase et son résumé

- Etape 3 :

Après avoir terminé le calcul de BLEU score, nous passons au deuxième model Multi-layer perceptron entre latent convolution de chaque phrase comme entré et BLEU score de chaque phrase comme sortie.

[15] Le Bilingue Evaluation Understudy Score, ou BLEU en abrégé, est une mesure permettant d'évaluer une phrase générée par rapport à une phrase de référence.

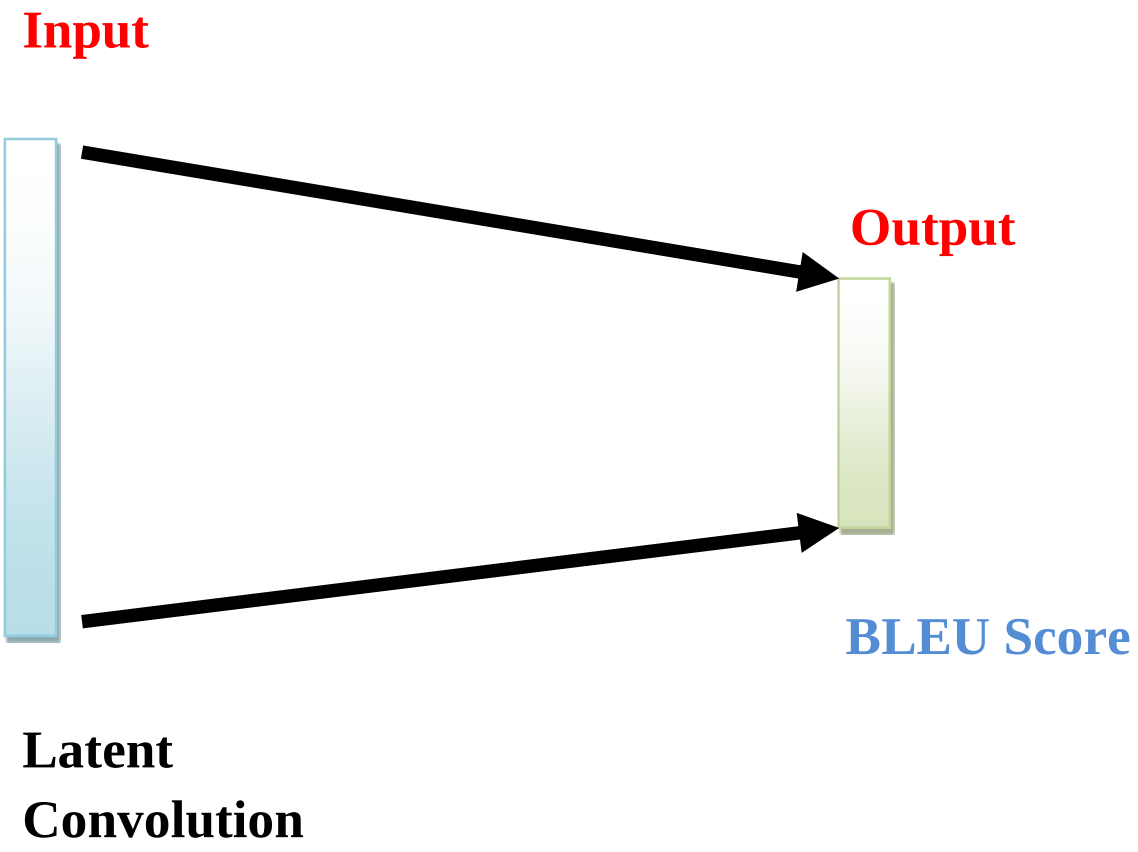


Figure 3-15 : l'étape de la Similarité entre Latent et BLEU score

Voici le résumé d'entrainement de model 2 Multi-layer :

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 288)]	0
dense (Dense)	(None, 500)	144500
dense_1 (Dense)	(None, 300)	150300
dense_2 (Dense)	(None, 100)	30100
dense_3 (Dense)	(None, 40)	4040
dense_4 (Dense)	(None, 1)	41

Total params: 328,981
Trainable params: 328,981

Figure 3-16 : le résumé d'entrainement de model 2 Multi-layer

Et finalement les résultats (Output) de deuxième model nous a permet savoir Quelle sont les phrases plus importante du texte Pour un bon résumé est calculée par prendre la valeur maximum de BLEU score entre les phrases de texte.

3.3 Conclusion

Au cours de ce chapitre, nous avons proposé une solution pour notre projet. En plus, nous avons proposé une architecture générale qui montre les différentes étapes que nous avons traversés et les mécanismes utilisés afin d'obtenir un résumé. Dans le prochain et dernier chapitre, nous allons présenter l'implémentation et les résultats de notre application.

Chapitre 4 Tests et résultats

4.1 Intoduction

Au cours du troisième chapitre, Nous avons illustré notre architecture générale qui explique les différentes étapes par les quelles passées le texte pour obtenir un résumé, à partir de la connexion avec Dataset, la segmentation en phrase,…etc. Dans ce dernier chapitre, nous commençons tout d'abord par la présentation d'environnement de développement, en détaillant les différents outils utilisés, en suite, nous expliquons les différentes étapes d'implémentation et d'expérimentation de notre système ainsi que les résultats obtenus.

4.2 L'environnement de développement

4.2.1 Google colaboratory

Google Colab ou Colaboratory est un service cloud fourni par Google (gratuit), basé sur Jupyter Notebook, pour la formation et la recherche en apprentissage automatique. La plateforme nous permet de former des modèles de machine learning directement dans le cloud. Donc, il n'est pas nécessaire d'installer quoi que ce soit sur notre ordinateur autre que le navigateur. Pour travailler avec Google Colab il suffit d'utiliser Google Drive.

Figure 4-1 : Logo de Google Colaboratory

4.2.2 Spyder (Scientific PYthon Development EnviRonment)

Spyder est un logiciel Python utile et fiable qui propose une édition avancée, des outils d'inspection des données, des tests interactifs et un débogage. Il intègre également des instruments d'assurance qualité et d'introspection du code spécifiques à Python, tels que Pyflakes, Pylint et rope. Cet IDE intègre des outils tels que NumPy, SciPy, Matplotlib et IPython, ainsi que d'autres logiciels open source.

Spyder fait partie de spyderlib, un module Python basé sur PyQt4, pyflakes, rope et sphinx qui fournit des widgets PyQt4 ou PySide robustes comme des éditeurs de code source, des éditeurs de dictionnaires, de listes/tuples et de tableaux NumPy basés sur une console Python ou une interface graphique. Spyder offre une boîte de dialogue de gestion de PYTHONPATH semblable à celle de MATLAB (fonctionne avec toutes les consoles).

Figure 4-2: Logo de Spyder

4.3 L'environnement personnel

L'implémentaion et les tests de notre application ont été réalise dans l'environnement matériel suivant :

- Processeur : Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz 2.71 GHz
- Mémoire installée (RAM) : 32,0 Go (31,9 Go utilisable)
- Windows : Windows 10 Professionnel
- Type de système : sysème d'exploitation 64 bits, processeur x64

4.4 Dataset

Dans notre projet Nous avons utilisé l'ensemble de donnée Amazon food reviews, Ci-dessous, nous définirons brièvement cette Dataset.

Amazon food reviews

Cet ensemble de données se compose de critiques d'aliments d'Amazon. Les données couvrent plus de 10 ans, y compris environ 500 000 commentaires en octobre 2012. Les avis incluent des informations sur les produits et les utilisateurs, des évaluations et des consultations en texte brut. Nous avons également des critiques pour toutes les autres catégories Amazon.

4.5 Tests et resultats

4.5.1 Évaluation Automatique des Résumés à l'aide de ROUGE

ROUGE est un acronyme pour Recall-Oriented Understudy for Gisting Evaluation. Il s'agit essentiellement d'un ensemble de mesures utilisées pour évaluer le résumé automatique de texte et la traduction automatique. Il fonctionne en comparant un résumé ou une traduction générée automatiquement avec un ensemble de résumés de référence ou de modèles (généralement générés manuellement). Le système ROUGE calcule plusieurs métriques pour évaluer le résumé généré par le système. Ces échelles montrent la corrélation entre les résumés des pairs et les résumés des modèles. Dans notre travail, nous utilisons ROUGE-1 (unigramme), ROUGE-2 (bigramme) et ROUGE-L (plus longue sous-séquence commune).

Précision, Rappel et F-mesure dans le contexte de Rouge

Afin d'évaluer l'exactitude du résumé généré par notre machine, nous avons calculé la Précision, Rappel et F- mesure de toute métrique:

- ✓ Précision

Dans le contexte de ROUGE, Précision fait référence au nombre de mots pertinents dans le résumé du candidat. Formule pour calculer la Précision:

$$Précision = \frac{\text{nombre de mots chevauchement}}{\text{nombre total de mots dans le résumé du système}}$$

- ✓ Rappel

Dans le contexte de ROUGE, RAPPEL signifie quelle proportion du résumé de référence est le système récupère-t-il ou capture-t-il ? Si nous ne considérons que les mots individuels, il peut être calculé comme suit:

$$Rappel = \frac{\text{nombre de mots chevauchement}}{\text{nombre total de mots dans le résumé de référence}}$$

✓ F-mesure

La F-mesure est la moyenne harmonique de la précision et du rappel :

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

Afin de mieux comprendre comment calculer ces trois paramètres, un exemple donné sera expliqué. Supposons que :

Le résumé généré par le système est :

"The outstanding student created a new and successful application"

Le résumé généré par l'humain est :

"The student created a new application"

Le nombre de mots qui se chevauchent entre le résumé du système et le résumé de référence est six (The, student, created, a, new, application), le total des mots du résumé de référence est six et le total des mots du résumé du système est de neuf, alors :

Rappel = 6/6 = 1 ; Précision= 6/9 = 0,66 ; F-mesure = 0,79

4.5.2 Comparaison entre résumé de notre système et résumé humain

Nous avons fait des tests sur notre système et nous avons obtenu des valeurs pour les mesures d'évaluation que nous avons utilisées. Dans le tableau suivant nous avons montré la comparaison entre le texte et le résumé de système, ainsi le texte et le résumé humain, aussi entre le résumé de système et le résumé humain. D'après cette comparaison nous avons observé que notre système donne des bons résultats.

	ROUGE-1			ROUGE-2			ROUGE-L			BLEU-Score	
	Rappel	**Précision**	**F-mesure**	**Rappel**	**Précision**	**F-mesure**	**Rappel**	**Précision**	**F-mesure**	**BLEU-1**	**BLEU-2**
Texte S_Résumé	1,0	0,38482	0,51021	1,0	0,33531	0,44965	1,0	0,38482	0,51021	0,21194	0,21194
Texte H_Résumé	0,50762	0,05343	0,09209	0,16700	0,01342	0,02247	0,47984	0,05008	0,08635	0,00435	0,00283
S_Résumé H_Résumé	0,35667	0,12491	0,17044	0,11138	0,03343	0,04605	0,33591	0,11733	0,16018	0,05894	0,01802

Tableau 4-1 : Comparaison de Texte-Résumé de Système et Texte-Résumé Humain et Résumé de Système-Résumé Humain utilisent ROUGE-1, ROUGE-2, ROUGE-L et BLEU- Score-1, BLEU-Score-2

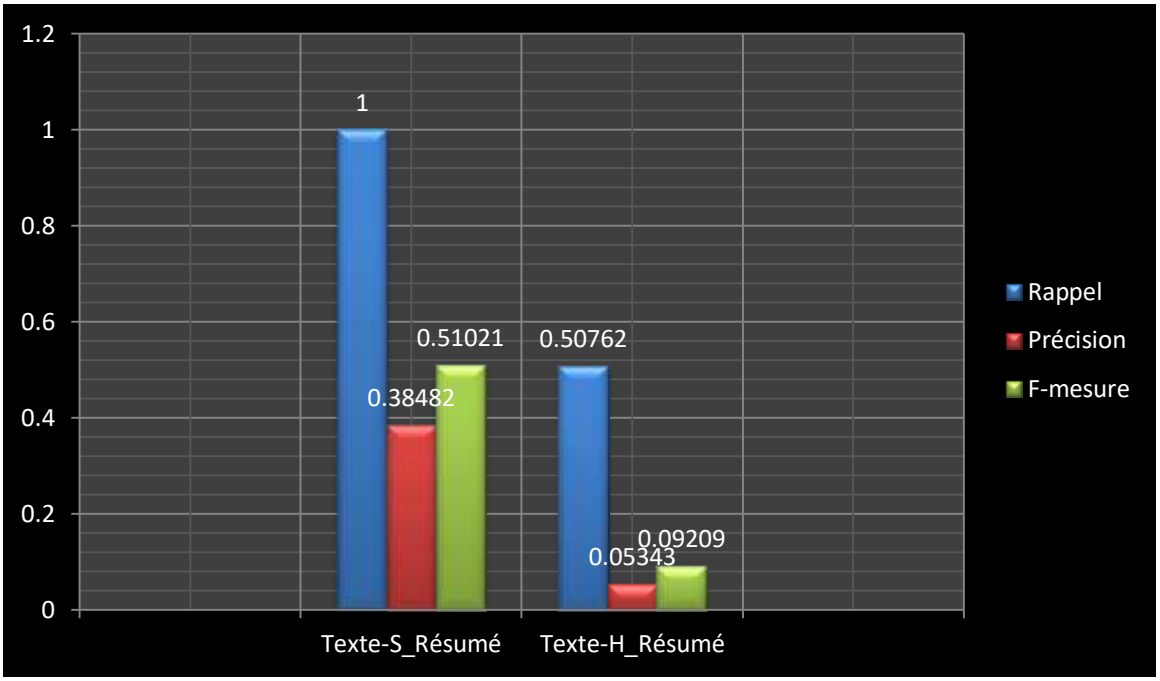


Figure 4-3 : Comparaison entre le Texte-Résumé de Système et le Texte-Résumé Humain utilisent ROUGE-1

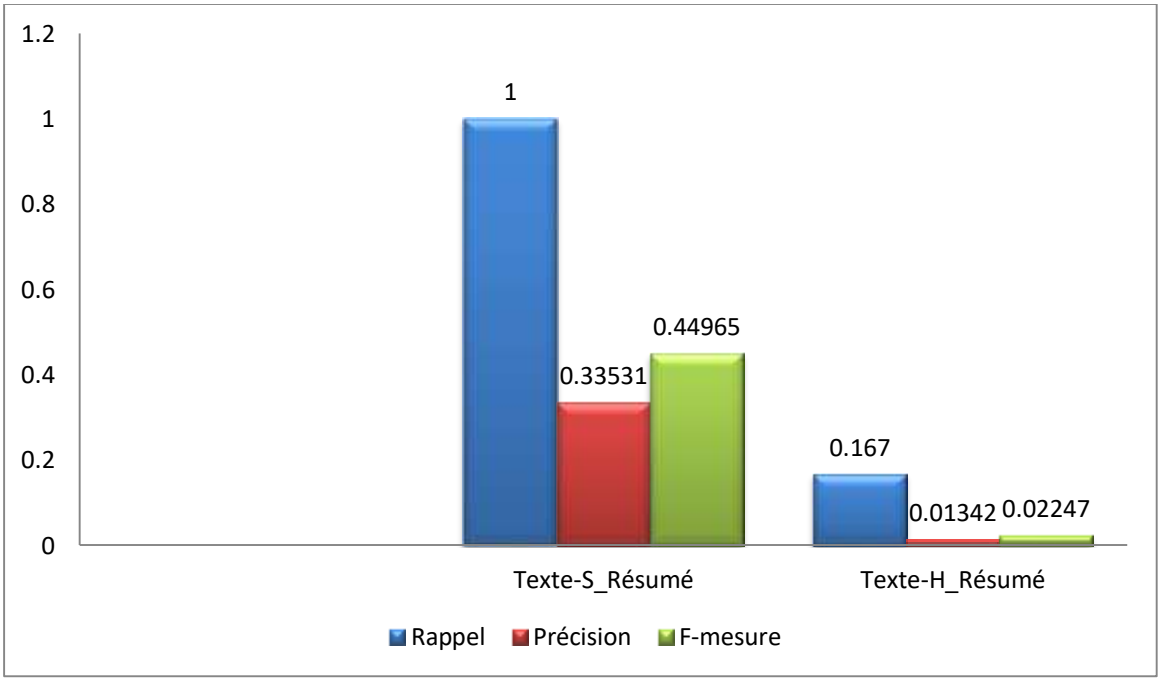


Figure 4-4: Comparaison entre Texte-Résumé de Système et Texte-Résumé Humain utilisent ROUGE-2

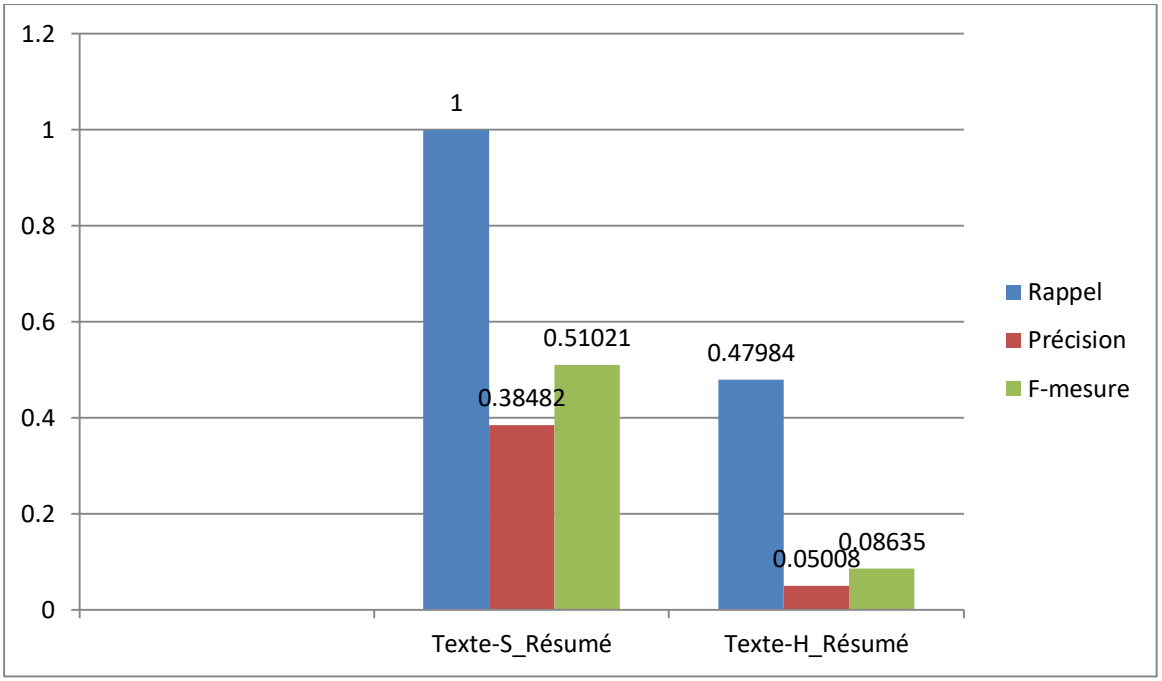


Figure 4-5: Comparaison entre Texte-Résumé de Système et Texte-Résumé Humain utilisent ROUGE-L

A travers cette comparaison, nous concluons que notre système a donné des résultats très satisfaisants, ce qui indique que Word2Vec, Auto-Encoder et Multi-layer perceptron Convient pour la Résumé automatique de texte.

4.6 Conclusion

Au cours de ce chapitre, nous avons d'abord presenté les langages de programmation et de la plateforme de développement utilisés pour implémenter notre système, à savoir, Google Colab et Spyder, ainsi que l'ensemble de données utilisé Amazon Food Reviews, aussi L'environnement de notre travail et le materiel utilisé, puis nous avons évalué notre système à l'aide d'ensemble de mesures ROUGE et BLEU-Score, ensuite nous avons discuté les résultats en démontrant l'efficacité de notre système.

Conclusion Générale

Le résumé automatique de textes est devenue plus que nécessaire dans divers domaines, notamment le domaine de la recherche, et à cause de l'augmentation continue des quantités de données disponibles et du besoin d'informations pertinentes sous une forme rapide et concise, l'utilisateur préfère trouver un bref résumé pour voir si le document contient ce qu'il cherche au lieu de devoir le lire complètement.

Notre travail s'inscrit dans le cadre d'amélioration du résumé automatique, afin d'attendre notre objectif, nous avons adopté la méthode extractive, c'est une méthode facile et donne des bons résultats. Au cours du premier chapitre de ce mémoire, nous avons donné un aperçu du résumé automatique du texte et de ses caractéristiques, tandis que dans le deuxième chapitre nous avons expliqué quelques conceptes utilisé dans le cadre d'. Ensuite, dans le troisième chapitre, nous avons exposé la méthodologie que nous avons suivie dans notre travail, dans lequel nous avons proposé une architecture générale qui explique notre solution, cette derniere montre les mécanismes utilisés tel que : Word2Vec, Auto-Encoder, Multi-layer perceptron, BLEU-score.

Nous avons aussi entrainé notre projet sur un grand ensemble de données Amazon Food_Reviews Dataset. Enfin, dans le dernier chapitre, nous avons présenté nos résultats et nous avons comparé les avec les résultats des travaux précédents.

En conclusion, et après avoir atteint les résultats et les avoir comparés avec les résultats des travaux antérieurs, nous avons conclu que notre projet a donné des résultats satisfaisants et encourageants. Cependant, le domaine de résumé automatique de texte reste sujet à développer afin de résoudre les problèmes des travaux antérieurs et l'ajout des nouvelles techniques qui permettront d'améliorer le niveau de résumé.

Bibliographie

[AID 2020] AID Aicha. Automatic text summarization for crisis management: Coronavirus (COVID-19) case. Mémoire master. Université de Akli Mohand Oulhadj. Bouira. 2020.

[ARIES 2013] ARIES Abdelkrime. Résumé automatique des textes. Mémoire de Magister de l'Ecole Nationale Supérieure d'Informatique (ESI). 26 juin 2013.

[Angermueller 2016] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle, "Deep learning for computational biology," Mol. Syst. Biol., vol. 12, no. 7, pp. 1–16, 2016.

[Barrios 2016] Federico Barrios, Federico L´opez, Luis Argerich, and Rosa Wachenchauzer. Variations of the similarity function of textrank for automated summarization. CoRR abs/1602.03606. http://arxiv.org/abs/1602.03606. 2016.

[Bensidiaissa 2020] Bensidiaissa Walid, Bouchetara Rym. Generative models for automatic multi-document summarization. Mémoire master. université Saad Dahleb Blida1. 2020.

[Bharti 2018] Prerna Bharti, Vivek Gupta, Pegah Nokhiz, Harish Karnick. SumPubMed: Summarization Dataset of PubMed Scientific Articles. 2018.

[Boudraf 2017] Boudraf Khadidja. Les Résumés Automatiques des Documents Textuels. Mémoire master. Université abdelhamid ibn badis – mostaganem. 2017.

[CARBONELL 1998] CARBONELL J, & GOLDSTEIN J. The use of MMR, diversity-based re-ranking for reordering documents and producing summaries. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. pages 335–336: ACM. 1998.

[Chalendar 2014] Gaël de Chalendar. The LIMA Multilingual Analyzer Made Free: FLOSS Resources Adaptation and Correction. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik. Iceland. pages 2932–2937. 2014.

[Chen 2016] Chen, Yu-Hsin and Krishna, Tushar and Emer, Joel and Sze, Vivienne, “Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks,” in IEEE International Solid-State Circuits Conference, ISSCC 2016, Digest of Technical Papers. , pp. 262–263. 2016.

[Chopra 2016] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California. USA. pages 93–98. http://aclweb.org/anthology/N/N16/N16-1012.pdf. 12-17 June 2016.

[Douzidia 2004] Fouad Soufiane Douzidia. Résumé automatique de texte arabe. Mémoire présenté à la Faculté des études supérieures en vue de l’obtention du grade de Magister en informatique. Septembre 2004.

[Elozino 2019] Elozino Egonmwan and Yllias Chali. Transformer-based Model for Single Documents Neural Summarization. In: Proceedings of the 3rd Workshop on Neural Generation and Translation. Hong Kong: Association for Computational Lin- guistics. doi: 10.18653/v1/D19-5607. url: https://www.aclweb.org/anthology/D19-5607. 4 November 2019.

[GENEST 2011] GENEST P.-E, LAPALME G. Framework for abstractive summarization using text-to-text generation. In Proceedings of the Workshop on Monolingual Text-To-Text Generation. pages 64–73: Association for Computational Linguistics. 2011.

[GENEST 2012] GENEST P. & LAPALME G. Fully abstractive approach to guided summarization. In The 50th Annual Meeting of the Association for Computational Linguistics. Proceedings of the Conference. Jeju Island, Korea - Volume 2: Short Papers, pages 354–358. July 8-14 2012.

[Grusky 2018] Max Grusky, Mor Naaman, Yoav Artzi. NEWSROOM: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies. Department of Computer Science. Cornell Tech Cornell University. New York, NY 10044. 2018.

[Kupiec 1995] Julian Kupiec, Jan Pedersen et Francine Chen. A trainable document summarizer.In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR ’95. pages 68–73. New York, NY, USA, ACM. 1995.

[Lamsiyah 2020] Salima Lamsiyah, Saïd Ouatik El Alaoui, Bernard Espinasse. Résumé automatique guidé de textes : État de l’art et perspectives. Paper. 17 février 2020.

[Lee 2013] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu et Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precisionranked rules. In Computational Linguistics. volume 38. MIT Press. 2013.

[LI 2011] LI P, WANG Y., GAO W. & JIANG J. (2011b). Generating aspect-oriented multi-document summarization with event-aspect model. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. EMNLP 2011, 27-31 July 2011. John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1137–1146. 2011.

[Mani 2001] Inderjeet Mani. Summarization Evaluation: An Overview. Paper. USA. 2001.

[Mir Tafseer 2019] Mir Tafseer Nayeem, Tanvir Ahmed Fuad et Yllias Chali.Neural. Diverse Abstractive Sentence Compression Generation. Université de Lethbridge, Lethbridge, AB, Canada. 22 aout 2019.

[Mnasri 2015] Maâli Mnasri. Résumé Automatique Multi-Document Dynamique : État de l'Art. Univ. Paris Sud, Orsay, France. 2015.

[Mnasri 2018] Maâli Mnasri. Résumé automatique Multi-document et dynamique. Thèse de doctorat de l'Université Paris-Saclay. Préparée à l'université Paris-Sud. 20 septembre 2018.

[Moreno 2014] Torres-Moreno, Juan-Manuel, *Automtic Text Summarization*, Wiley. pages 320. 1er octobre 2014.

[Nallapati 2017] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA. pages 3075–3081. http://aaai.org/ocs/index.php/AAAI/ AAAI17/paper/view/14636. 4- 9 Février 2017.

[Nair 2010] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," Proc. 27th Int. Conf. Mach. Learn., no. 3, pp. 807–814, 2010.

[OLIVEIRA 2016] OLIVEIRA H., LIMA R., LINS R. D., FREITAS F., RISS M. & SIMSKE S. J. (2016b). Assessing concept weighting in integer linear programming based single-document summarization. In Proceedings of the 2016 ACM Symposium on Document Engineering, DocEng 2016, Vienna, Austria. pages 205–208. September 13 - 16, 2016.

[Paulus 2017] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. CoRR abs/1705.04304. http://arxiv.org/ abs/1705.04304. 2017.

[Porter 1997] M. F. Porter. Readings in information retrieval. chapitre An algorithm for suffix stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 1997.

[Krizhevsky 2012] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., pp. 1–9, 2012.

[Radev 2002] Dragomir R. Radev, Eduard Hovy et Kathleen McKeown. *Introduction to the special issue on summarization*, Association pour la linguistique informatique, vol. 28, no. 4. pages 399–408. December 2002.

[Radev 2003] D.R., Teufel, S., Saggion, H., Lam, W., Blitzer, J., Qi, H., C¸ elebi, A., Liu, D., Drabek, E.: Evaluation challenges in large-scale document summarization. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, ACL '03. pages 375– 382. 2003.

[Rath 1961] G. J. Rath, A. Resnick et T. R. Savage. Comparisons of four types of lexical indicators of content. American Documentation. Article. vol. 12, no. 2. pages 126–130. 1961.

[Sainath 2013] T. Sainath, A. R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," Proc. IEEE Int. Conf. Acoust. Speech Signal Process., pp. 8614–8618, 2013.

[Saggion 2000] Horacio Saggion et Guy Lapalme. *Concept identification and presentation in the context of technical text summarization*. In Proceedings of the 2000 NAACLANLP Workshop on Automatic summarization - Volume 4, NAACL-ANLP-AutoSum '00. pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics. 2000.

[Saggion 2002] Horacio Saggion and Guy Lapalme. Generating Indicative-Informative Summaries with SumUM. Computational Linguistics, 28(4) :497–526, (Cité en page 9). jan 2002.

[Saggion 2016] Horacio Saggion, Thierry Poibeau. Automatic Text Summarization: Past, Present and Future, HAL archive-ouvertes. France. 27 Juil 2016.

[Spärck Jones 2007] Karen Spärck Jones. Automatic summarising: The state of the art. Inf. Process. Manage., 43(6) :1449–1481. ISSN 0306-4573. doi: 10.1016/j.ipm.2007.03.009. URL http://dx.doi.org/10.1016/j.ipm.2007.03.009. (Cité en page 8). November 2007.

[Shi 2020] Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, et Chandan k. Reddy. *Neural Abstractive Text Summarization with Sequence-to-Sequence Models.* https://doi.org/10.1145/3419106. December 2020.

[Wang 2016] H. Wang, A. Meghawat, L.-P. Morency, and E. P. Xing, "Select-Additive Learning: Improving Cross-individual Generalization in Multimodal Sentiment Analysis," vol. 1, 2016.

[Wang 2017] H. Wang and B. Raj, "On the Origin of Deep Learning," Arxiv, pp. 1–72, 2017.

[Yang 2019] Yang Liu, Fine-tune BERT for Extractive Summarization. In: ArXiv abs/1903.10318. 2019.

[Yatsko 2010] V. A. Yatsko, M. S. Starikov et A. V. Butakov. Automatic genre recognition and adaptive text summarization. volume 44. pages 111–120, Secaucus, NJ, USA. Springer-Verlag New York. June 2010.

[ZHANG 2011] ZHANG R., YOU O. & LI W. *Guided summarization with aspect recognition*. In Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA. 14-15 November 2011.