République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

---

Université Saad Dahleb, Blida1

Faculté des Sciences

Département Informatique

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master en *informatique*

Option intitulée : *Traitement Automatique de la Langues*

Utilisation des transformes pour le résumé texte abstractifs

Réalisé Par :

Belmehdi Abdelkader

Hamidou Yasmine

Présenté devant :

Présidente : Mme Djeddar.

Examinateur : Monsieur Cherif Zehar.

Encadreur : KAMECHE Abdellah Hicham.

2021 / 2022

Remerciments

En rédigeant cette page du manuscrit, on est obligé de reconnaître que cette thèse est le fruit d'un peu de recherche et de beaucoup d'aide reçue de nombreuses personnes.

On tient à remercier avant tout, Dieu de nous avoir prodigué la force morale et physique et nous a permis d'achever ce travail.

On adresse nos remerciements aux personnes qui nous ont aidés dans la réalisation de ce mémoire.

En premier lieu, on remercie Monsieur Kameche. En tant qu'encadrant de mémoire, il nous a aidé et encadrer tout au long de la réalisation de ce travail malgré ses tâches très lourdes.

On adresse nos plus sincères remerciements aux membres du jury pour l'intérêt qu'ils ont exprimé pour ce modeste travail et accepté de l'examiner.

Nos remerciements vont à tous les enseignants du département d'informatique que nous respectons beaucoup.

Enfin, on souhaiterait adresser des remerciements très particuliers à toute notre famille.

Mille fois merci !

Dédicace

Nos dédicaces ne sont que l'expression de nos profondes gratitudes, de nos salutations chaleureuses et de nos sincères reconnaissances à tous ceux qui comblent nos vies et y confèrent son goût et sa saveur.

Il nous sommes agréable de profiter de cette occasion, pour rendre un hommage particulièrement sincère à travers ce modeste travail, à tous ceux qui nous sont chers, à tous ceux qui nous ont soutenus moralement et matériellement.

On dédie donc ce modeste travail:

A nos très chers et honorables parents ainsi que toutes nos familles.

À ma femme Fatima.

À tous nos enseignants de la faculté.

À tous nos chers amis.

Et à toutes personnes qui ont contribué à la réussite de ce projet que ce soit de près ou de loin.

Résumé

Tout le monde veut trouver les bonnes informations rapidement et efficacement maintenant qu'ils vivent à l'ère de l'information. Cependant, en raison de la quantité massive de données déjà disponibles, la tâche est devenue de plus en plus difficile au fil du temps ! En conséquence, les scientifiques ont conçu la technique de résumé multi-documents, qui utilise des innovations récemment développées telles que l'apprentissage automatique et les réseaux neuronaux.

Afin de mettre en évidence seulement les événements pertinents, le résumé automatique de texte revêt une grande importance car il permet d'extraire automatiquement les informations considérées comme essentielles pour former un résumé automatique bref et informatif.

Les études précédentes se focalisent sur la génération d'un résumé automatique mono-document. Dernièrement plus d'études ont commencé à se centraliser sur les problèmes de construction d'un résumé automatique multi-documents à cause des différentes contraintes et éléments qui s'imposent, tel que la redondance des mêmes phrases et paragraphe dans différents documents.

Dans notre travail, nous proposons une solution qui consiste à développer une application pour la génération de résumé automatique multi-documents basés sur l'apprentissage profond en utilisant différentes variantes d'architectures neuronales à base de Transformers appelé Pegasus et BART. Nous avons construit et mis au point un modèle de Pegasus. En utilisant des algorithmes de clustering, nous avons prétraité nos ensembles de données et obtenu des résultats pertinents. L'évaluation a été faite automatiquement en utilisant les scores ROUGE (est une mesure d'evaluation utilisée pour évaluer les résumés).

Mots clés : Résumé automatique, Multi-documents, Apprentissage profond, Abstractif, Transformers, Pegasus, BART, ROUGE (ROUGE-N, ROUGE-L).

Abstract

Everyone wants to find the right information quickly and efficiently now that they live in the information age. However, due to the massive amount of data already available, the task has become increasingly difficult over time! As a result, scientists devised the Multi-Document Summarization technique, which uses recently developed innovations such as machine learning and neural networks.

In order to highlight only the relevant events, the automatic text summarization is of great importance because it allows to automatically extract the information considered essential to form a brief and informative automatic summary.

Previous studies have focused on the generation of a single-document automatic summary. Lately, more studies have started to focus on the problems of building an automatic multi-document summary because of the different constraints and elements that are imposed, such as the redundancy of the same sentences and paragraphs in different documents.

In our work, we propose a solution which consists in developing an application for the generation of automatic multi-document summary based on deep learning using different variants of neural architectures based on Transformers called Pegasus and BART. We have built and developed a model of Pegasus. Using clustering algorithms, we pre-processed our datasets and obtained relevant results. The evaluation was done automatically using ROUGE scores (is an evaluation measure used to evaluate a summarization).

Key word : Automatic Summary, Multi-Document, Deep Learning, Abstract, Transformers, Pegasus, ROUGE (ROUGE-N, ROUGE-L).

الملخص

يريد الجميع العثور على المعلومات الصحيحة بسرعة وكفاءة الآن بعد أن أصبحوا في عصر المعلومات. ومع ذلك، نظرًا للكمية الهائلة من البيانات المتاحة بالفعل، أصبحت المهمة صعبة بشكل متزايد بمرور الوقت! نتيجة لذلك، صمم العلماء الأجزاء التقنية المجردة المتعددة، والتي تستخدم الابتكارات المطورة حديثًا مثل التعلم الآلي والشبكات العصبية.

ولإبراز الأحداث ذات الصلة فقط، يتسم الموجز التلقائي للنص بأهمية كبيرة لأنه يتيح الاستخراج التلقائي للمعلومات التي تعتبر ضرورية لتشكيل موجز تلقائي موجز وغني بالمعلومات.

ركزت الدراسات السابقة على إنشاء ملخص تلقائي من وثيقة واحدة. و في الآونة الأخيرة، بدأ المزيد من الدراسات في التركيز على مشاكل وضع موجز تلقائي متعدد الوثائق بسبب القيود والعناصر المختلفة المطلوبة، مثل التكرار نفس الجمل والفقرات في وثائق مختلفة.

في عملنا، نقترح حلاً يتمثل في تطوير تطبيق لإنشاء ملخص تلقائي متعدد المستندات يعتمد على التعلم العميق باستخدام متغيرات مختلفة من البنى العصبية بناءً على محولات تسمى Pegasus و BART. لقد بنينا وطورنا نموذجًا لـ Pegasus. باستخدام خوارزميات التجميع، قمنا بمعالجة مجموعات البيانات الخاصة بنا مسبقًا وحصلنا على النتائج ذات الصلة. تم إجراء التقييم تلقائيًا باستخدام درجات (هي مقياس تقييم يستخدم لتقييم التلخيص)ROUGE .

كلمات مفتاحية : الملخص التلقائي ، متعدد المستندات ، التعلم العميق ، تجريدي ، شبكة الخلايا العصبية

Transformers, Pegasus, ROUGE (ROUGE-N, ROUGE-L) .

Table des matières

Chapitre II : Concepts de base du résumé automatique de texte

Chapitre III : Approche proposée

Chapitre IV : Tests et résultats

Liste des figures

Liste des abréviations

TAL	Traitement Automatique du Langage
ML	Machine Learning
IA	Intelligence Artificiel
DL	Deep Learning
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory Neural
GRU	Gated Recurrent Unit
LCS	Longest Common Subsequence
BE	Basic Element
ROUGE	Recall-Oriented Understudy for Gisting Evaluation

Liste des tableaux

Introduction générale

Contexte de travail

Aujourd'hui Internet est devenu un espace de recherche immense ou l'on trouve tout et rien à la fois ! Apres une recherche simple dans une page Google on se retrouve toujours face à des milliers et des milliers de résultats à cause de la grande richesse informationnelle qui nous entoure.

Grâce aux : médias, blogs, livres, revues… on voit bien que l'information textuelle s'accumule rapidement et en très grande quantité de ce fait il est intéressant d'offrir des outils informatiques tels que les résumés automatique afin de facilité la tâche, car lire chaque document trouver des informations fiables et créer manuellement des résumés est cependant impossible.

Ce travail s'intéresse au résumé automatique de texte abstractif tel que les approches abstractives visent à générer des résumés comme le font les humains en paraphrasant les phrases les plus cruciales et en générant éventuellement de nouveaux mots.

Problématique

De nos jours le temps est devenu si précieux que les gens ne peuvent plus le sacrifier, C'est pourquoi les gens cherchent à résumé leurs documents textuels de manière automatique et sans aucune aide humaine. Le plus grand problème du résumé automatique peut être défini par la sélection des informations les plus pertinents d'un ou de plusieurs document tout en minimisant la redondance. Mais comment savoir si un document est pertinent ou non et comment faire pour en avoir le bon résumé ?

Objectifs de travail

Il existe deux approches principales pour aborder la tâche de résumé automatique : une approche par abstraction et une approche pas extraction.

Nous allons nous concentrer au cours de ce projet aux résumés automatiques abstractifs en exploitant différentes variantes d'architectures neuronales à base de Transformers.

Notre travail consistera à :

- ✓ Etudier les étapes et les techniques utilisées dans le résumé automatique.

- ✓ Présenter l'environnement de développement en détaillant les différents outils utilisés et expliquer l'approche proposée.
- ✓ Construire et présenter l'architecture de notre système.

Organisation de mémoire

Afin de simplifier et organiser la lecture de ce mémoire, nous avons décidé de diviser le travail de la façon suivante :

CHAPITRE I : L'apprentissage automatique pour le TAL.

CHAPITRE II : Concepts de base du résumé automatique de texte.

CHAPITRE III : Approche proposée.

CHAPITRE IV : Tests et résultats.

Chapitre I : L’apprentissage automatique pour le TAL

I.1 Introduction

Le résumé est un art vital pour l'humanité, car il permet d'obtenir rapidement les informations les plus importantes tout en gagnant du temps. Mais quelle que soit son importance, cela pourrait prendre beaucoup de temps pour acquérir un bon résumé de nos jours en raison des données massives que nous avons à cette époque.

Donc, pour le rendre plus rapide et plus pertinent, les scientifiques ont mis en place certaines technologies capables de le faire, telles que l'apprentissage automatique et les réseaux de neurones.

Ce chapitre est consacré pour la définition de l'apprentissage automatique et l'utilité de l'apprentissage automatique pour la génération du résumé automatique multi-documents.

I.2 Apprentissage automatique (machine learning)

En terme court l'apprentissage automatique « Machine Learning » est une technique de modélisation qui fait appel à des données. Si on veut plus de détails, cet apprentissage consiste à découvrir un ensemble de données et apprendre comment ce dernier fonctionne et donne à la fin un modèle général qui peut résoudre un problème similaire à la base de données, ce modèle est conclu après un entrainement avec les données d'entrée. [1]

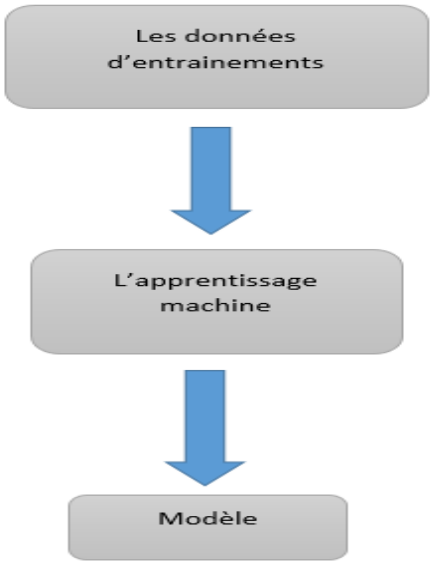


Figure 1 : Processus de l'apprentissage automatique

I.3 les types d’apprentissage automatique

Nous allons présenter les différents types majeurs de l’apprentissage automatique :

I.3.1 Apprentissage supervisé :

L'apprentissage supervisé commence généralement par un ensemble de données bien défini et une certaine compréhension de la façon dont ces données sont classifiées. L'apprentissage supervisé a pour but de déceler des modèles au sein des données et de les appliquer à un processus analytique. Ces données comportent des caractéristiques associées à des libellés qui définissent leur signification (par exemple, créer une application d'apprentissage automatique capable de faire la distinction entre plusieurs millions d'animaux, en se basant sur des images et des descriptions écrites. [2]

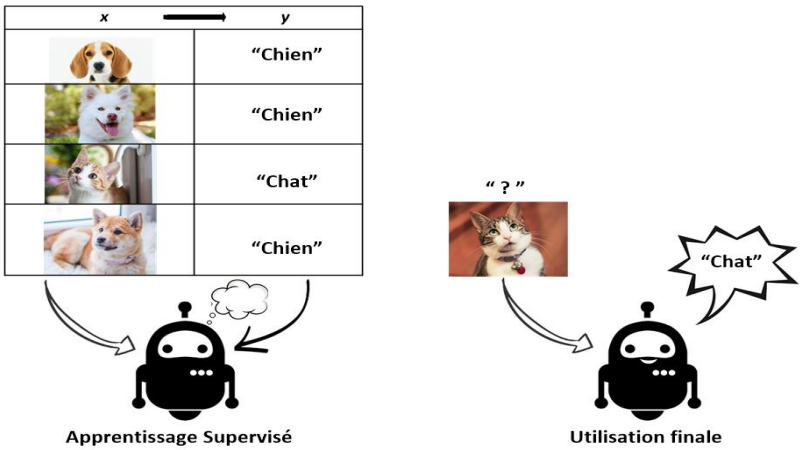


Figure 2 : Schéma explicatif d’apprentissage supervisé

I.3.2 Apprentissage non supervisé :

L'apprentissage non supervisé est utilisé lorsque le problème nécessite une quantité massive de données non étiquetées. Par exemple, les applications de réseaux sociaux, telles que Twitter, Instagram et Snapchat, exploitent toutes de très grandes quantités de données non étiquetées.

L'apprentissage non supervisé mène un processus itératif, analysant les données sans intervention humaine. Il est utilisé avec la technologie de détection de spam

envoyé par e-mail.Les e-mails normaux et les spams comportent un nombre de variables beaucoup trop élevé pour qu'un analyste puisse étiqueter les e-mails indésirables envoyés en masse. [2]

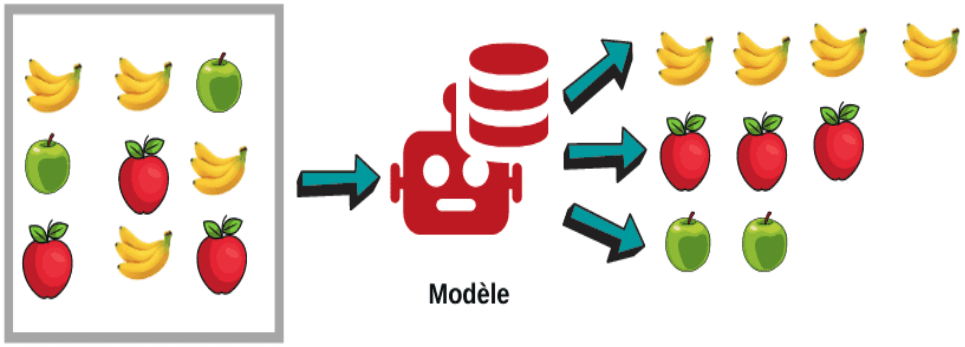


Figure 3 : Schéma d'apprentissage non supervisé

I.3.3 Apprentissage semi supervisé :

Il s'agit d'un mixe entre l'apprentissage supervisé et non supervisé en utilisant des données étiquetées et non-étiquetées pour le même ensemble de données.

L'avantage d'utiliser cette approche réside dans le fait que l'étiquetage de données peut être couteux et prend souvent beaucoup de temps. En plus, il pourra entrainer un biais humain dans les données étiquetées. Dans ce cas, l'apprentissage semi-supervisé, qui ne nécessite que quelques étiquettes, est très pratique. Et le fait d'inclure un grand nombre de données non étiquetées au cours du processus d'entrainement a tendance à améliorer la performance du modèle final tout en réduisant le temps et les coûts consacrés à sa construction. [3]

I.3.4 Apprentissage par renforcement :

L'apprentissage par renforcement est un modèle d'apprentissage comportemental. L'algorithme reçoit les commentaires de l'analyse des données et guide l'utilisateur vers les meilleurs résultats. L'apprentissage par renforcement est différent des autres types d'apprentissage supervisé, Parce que le système n'est pas formé avec un exemple d'ensemble de données. Au lieu de cela, le système préfère apprendre par essais et erreurs. Ainsi, une série de décisions réussies conduit au renforcement du processus car c'est celui qui résout le plus efficacement le problème posé.

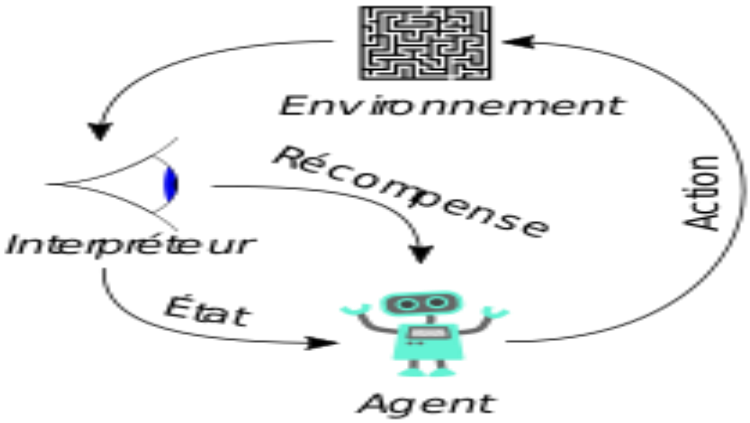


Figure 4 : Schéma d’apprentissage par renforcement.

I.3.5 Apprentissage en profondeur :

L'apprentissage en profondeur (deep learning) est un domaine de recherche sur l'apprentissage automatique basé sur un type particulier de mécanisme d'apprentissage.

Il est caractérisé par l'effort de créer un modèle d'apprentissage à plusieurs niveaux, dans lequel les niveaux les plus profonds prennent en compte les résultats des niveaux précédents, les transformant et en faisant toujours plus d’abstraction. Cet aperçu des niveaux d'apprentissage est inspiré par la façon dont le cerveau traite l'information et apprend en réagissant aux stimuli externes. Chaque niveau d'apprentissage correspond, par hypothèse, à l'une des différentes zones qui composent le cortex cérébral. [4]

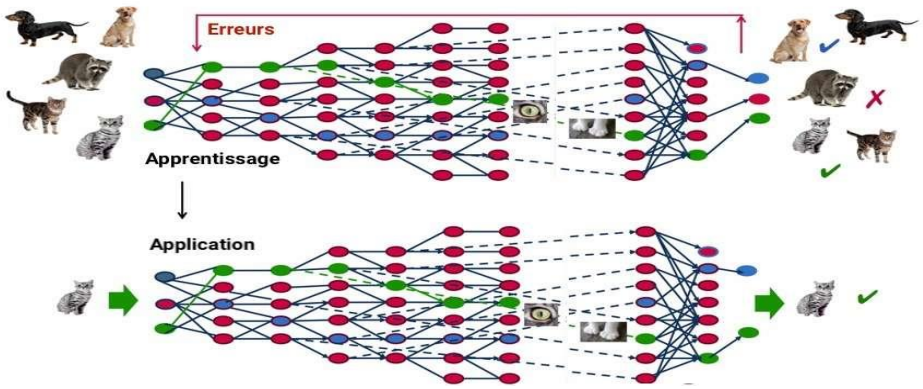


Figure 5 : Schéma explicatif d’apprentissage en profondeur

I.4 les réseaux de neurones

I.4.1. Définition :

Un réseau de neurones artificiel est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement de vrais neurones.

Un réseau de neurones est constitué d'un très grand nombre de petites unités de traitement identiques appelées neurones artificiels.

Chacun de ces neurones est par ailleurs fort complexe. Essentiellement, il s'agit de tissu vivant et de chimie. Les spécialistes des neurones biologiques commencent à peine à comprendre quelques-uns de leurs mécanismes internes. On croit en général que leurs différentes fonctions neuronales, y compris celles de la mémoire, sont stockées au niveau des connexions (synapses) entre les neurones. C'est ce genre de théories qui a inspiré la plupart des architectures de réseaux de neurones. L'apprentissage consiste alors soit à établir de nouvelles connexions, soit à en modifiant des existantes.

L'origine des réseaux de neurones vient de l'essai de modélisation du neurone biologique par Warren McCulloch et Walter Pitts [Lot-99]. Ils supposent que l'impulsion nerveuse est le résultat d'un calcul simple effectué par chaque neurone et que la pensée née grâce à l'effet collectif d'un réseau de neurones interconnectés. [1]

Le réseau de neurones est inspiré du cerveau humain, il a plusieurs nœuds connectés entre eux qui transmettent des informations.

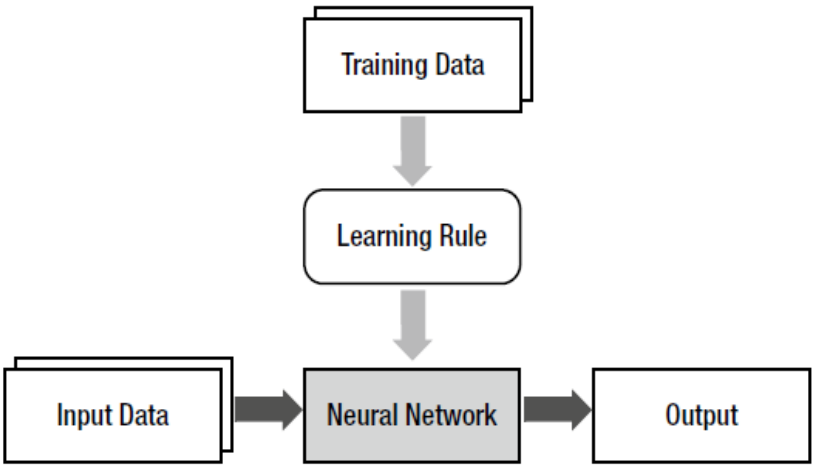


Figure 6 : Schéma explicatif d'un réseau de neurone

I.4.2. Modèle mathématique :

Les réseaux de neurones biologiques réalisent facilement un certain nombre d'applications telles que la reconnaissance de formes, le traitement du signal, l'apprentissage par l'exemple, la mémorisation, la généralisation. Ces applications sont pourtant, malgré tous les efforts déployés en algorithmique et en intelligence artificielle, à la limite des possibilités actuelles. C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau que les réseaux de neurones artificiels se sont développés. Les réseaux de neurones artificiels sont des modèles, à ce titre ils peuvent être décrits par leurs composants, leurs variables descriptives et les interactions des composants.

A. Composant (le neurone artificiel)

Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amont. A chacune de ces entrées est associé un poids w abréviation de weight (poids en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associé un poids. [5]

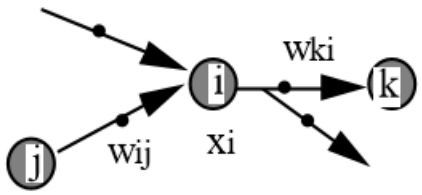


Figure 7 : Structure d'un neurone artificiel

Pour le neurone d'indice i, les entrées sur celui-ci sont de poids wij alors que les connexions avals sont de poids wki.

B. Variables descriptives

Ces variables représentent l'état du système. Dans un réseau de neurones, qui est un système non autonome, un sous-ensemble de variables descriptives est formé par des variables d'entrée. La valeur de la variable d'entrée est déterminée en dehors du modèle.

C. Structure d'interconnexion

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle. Elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité.

Réseau multicouche (au singulier) : les neurones sont arragés par couche. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches avalent. Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurone d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelés couches cachées. [3]

- **Les fonctions d'activation**

La fonction d'activation est une composante essentielle du réseau neuronal. Ce que cette fonction a décidé est si le neurone est activé ou non. Il calcule la somme pondérée des entrées et ajoute le seuil. Il existe de nombreux types de fonctions d'activation. [2]

I.4.3. Descente de gradient :

La méthode du gradient est la méthode d'optimisation la plus ancienne et la plus courante. Le but de cette méthode est de trouver les extrémums en déterminant un vecteur de direction basé sur le gradient avec une mise à jour des paramètres qui est effectuée pour la convergence progressive vers une valeur optimale de la fonction objective. Donc le gradient de descente minimise une fonction objective $j(\theta)$ avec son modèle de paramètreθ. On rappelle que le gradient est un vecteur de dérivé partiel d'une fonction par rapport à ces entrées.

I.5 l'architecture neuronales pour le TAL

Un réseau de neurones peut prendre des formes différentes selon l'objet de la donnée qu'il traite et selon sa complexité et la méthode de traitement de la donnée. Les architectures ont leurs forces et faiblesses et peuvent être combinées pour optimiser les résultats. Le choix de l'architecture s'avère ainsi crucial et il est déterminé principalement par l'objectif.

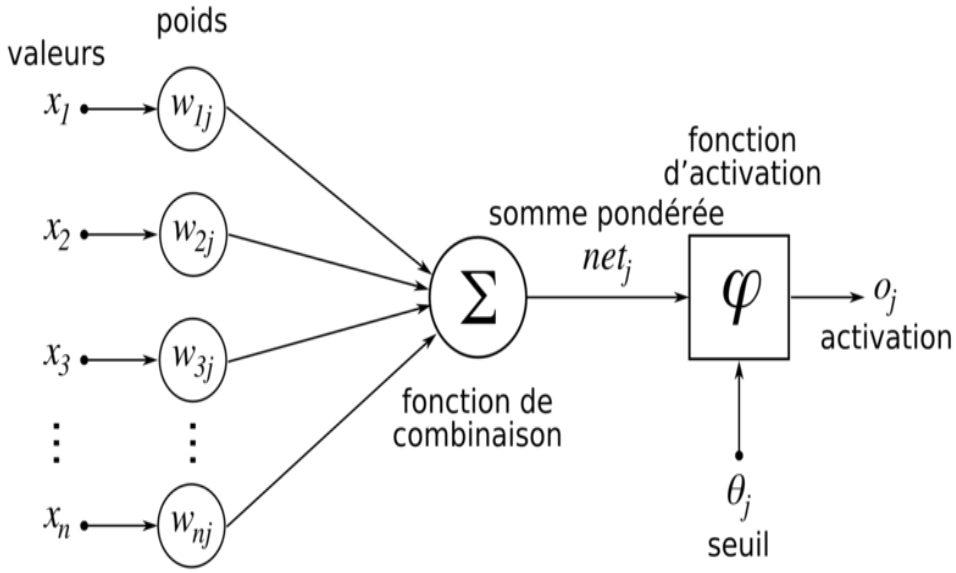


Figure 8 : Architecture des réseaux neurones

I.6 Les modèles de réseaux de neurones

Il existe beaucoup des modèles pour les réseaux de neurones et parmi ces modèles nous citions

a. Récurrent Neural Network (RNN) :

Les Réseaux de Neurones récurrents traitent l'information en cycle. Ces cycles permettent au réseau de traiter l'information plusieurs fois en la renvoyant à chaque fois au sein du réseau.

La force des Réseaux de neurones récurrents réside dans leur capacité de prendre en compte des informations contextuelles suite à la récurrence du traitement de la même information. Cette dynamique auto-entretient le réseau.

Les Réseaux de neurones récurrents se composent d'une ou plusieurs couches. Le modèle de Hopfield (réseau temporel) est le réseau de neurones récurrent d'une seule couche le plus connu.

Les Réseaux de neurones récurrents à couches multiples revendiquent quant à eux la particularité de posséder des couples (entrée/sortie) comme les perceptrons entre lesquels la donnée véhicule à la fois en propagation en avant et en rétro propagation.

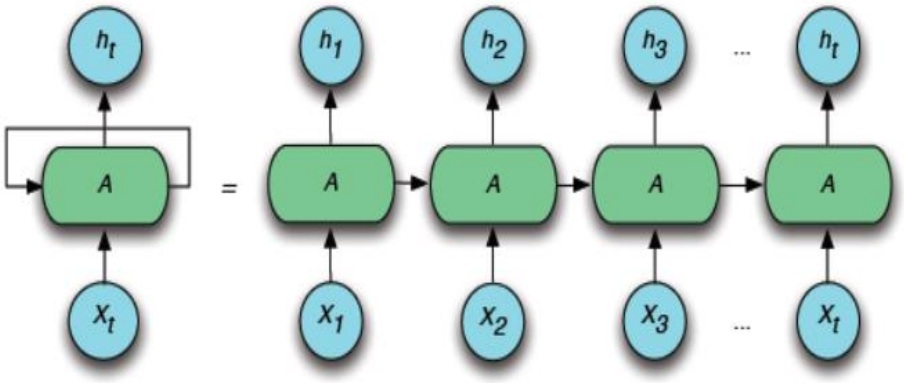


Figure 9 : Description de la récurrence dans un réseau récurrent

Cette illustration représente une couche récurrente d'un réseau de neurone. On voit sur la partie de gauche de cette figure que cette couche prend en entrée une observation \(x_t\) (où \(t\) est l'indice de l'observation dans la séquence) et retourne un vecteur \ (h_t\). On remarque surtout, ce qui n'était pas le cas pour les couches que vous avez étudiées jusqu'à présent, qu'il existe une boucle de rétroaction.

b. Réseau de neurone récurrent à mémoire court et long terme LSTM :

Les réseaux de mémoire à long terme, souvent appelés simplement "LSTM", sont un type spécial de RNN qui sont capables d'apprendre des dépendances à long terme. Ils ont été introduits par Hochreiter et Schmidhuber (1997), et ont été raffinés et popularisés par de nombreux travaux ultérieurs.

Ils fonctionnent extrêmement bien sur une grande variété de problèmes et sont maintenant largement utilisés.

Le concept de Lstm est similaire à celui de RNN, consistant en un ensemble de composants connectés de manière récurrente appelés blocs de mémoire. Chaque bloc de mémoire contient généralement une cellule de mémoire auto-connectée, des portes d'entrée, de sortie et d'oubli qui permettent de mettre à jour un bloc donné. [2]

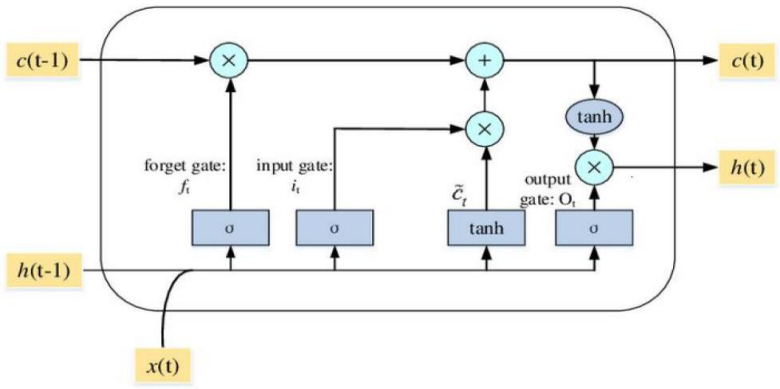


Figure 10 : Illustration d'un bloc de mémoire LSTM avec une cellule

c. Gated Recurrent Unit (GRU) :

Les Gated Recurrent Unit (GRU) soit Unité Récurrentes à Portes (Cho et al, 2014), sont une alternative aux LSTM. Elles sont plus simples, composées de moins de portes. [5]

L'équation 1.6 et la figure 6 présentent le fonctionnement d'une cellule GRU.

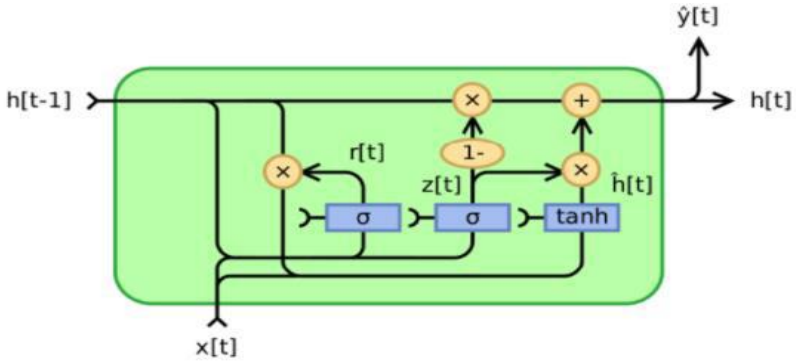


Figure 11 : Unité de base GRU.

zt = ‡ (Wz · [ht-1,Xt])
rt = ‡ (Wr · [ht-1,Xt]) (6.1)
˜ht= tanh (Wh · [rt ú ht-1,Xt])
ht = (1 -zt) * ht-1 + zt * ˜ht

La porte de mise à jour (update gate) zt et la porte de remise à zero (reset gate) rt sont calculées par activation sigmoïde suivant les poids des neurones W, l'entrée de la cellule X, la valeur de la couche précédente ht≠1 et le biais b. L'état candidat˜ht fonctionne de façon similaire utilisant cette fois une fonction d'activation tangente hyperbolique (tanh). Cependant, il prend aussi en compte la porte de remise à zéro. [5]

L'état de la cellule ht est obtenu avec la multiplication de la porte de mise à jour zt et l'état de la couche précédente ht≠1 ainsi que la multiplication de la porte de mise à jour zt et l'état candidat˜ht. En traduction automatique les GRU obtiennent des résultats similaires aux LSTM. Cependant, leur simplicité requiert moins de calculs et offre donc de meilleures performances en temps de calcul. [5]

d. Transformers :

Transformer est un modèle d'apprentissage en profondeur de type seq2seq (donc un réseau de neurones) qui a la particularité de n'utiliser qu'un mécanisme d'attention, et non un réseau récurrent ou convolutif.

Les réseaux de neurones transformers ont pour but de prévoir une séquence de longueur variable en fonction d'une autre séquence de longueur variable.

Le modèle seq2seq est un modèle qui prend une séquence (séquence d'éléments de même type) en entrée et renvoie une séquence en sortie. Un excellent exemple d'un tel modèle est la traduction de texte.

Pour faire des modèles séquence à séquence (seq2seq) avant la venue du Transfomer, il fallait faire recours au fameux LSTM (ou GRU) qu'on utilisait dans une architecture Encoder-Decoder.

Il se différencie par le fait de n'utiliser que le mécanisme d'attention et aucun réseau récurrent. Contrairement aux RNN, les Transformers n'exigent pas que les données séquentielles soient traitées dans l'ordre. C'est grâce à cette fonctionnalité, que le Transformer permet une parallélisation. [2]

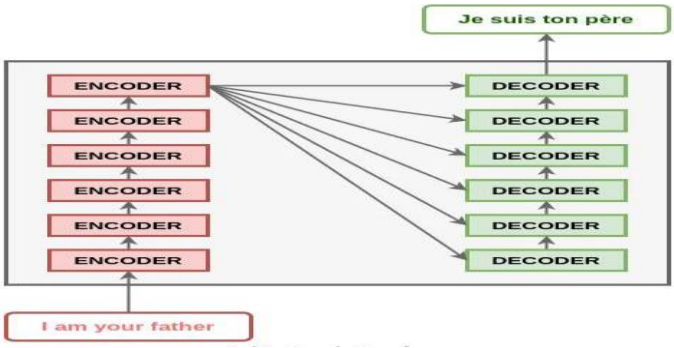


Figure 12 : Architecture du Transformers

. **Encodeur** : L'encodeur est composé d'un empilement de N = 6 couches identiques. Chaque couche comporte deux sous-couches. Le premier est un mécanisme d'auto-attention à plusieurs têtes, et le second est un réseau d'anticipation simple, entièrement connecté en fonction de la position. Nous utilisons une connexion résiduelle autour de chacune des deux sous-couches, suivie d'une normalisation de couche. Autrement dit, la sortie de chaque sous-couche est LayerNorm(x + Sublayer(x)), où Sublayer(x) est la fonction mise en œuvre par la sous-couche elle-même. Pour faciliter ces connexions résiduelles, toutes les sous-couches du modèle, ainsi que les couches d'intégration, produisent des sorties de dimension d modèle = 512. [6]

. Décodeur : Le décodeur est également composé d'un empilement de N = 6 couches identiques. En plus des deux sous-couches dans chaque couche d'encodeur, le décodeur insère une troisième sous-couche, qui effectue une attention multi-tête sur la sortie de la pile d'encodeur. Semblable au codeur, nous utilisons des connexions résiduelles autour de chacune des sous-couches, suivies d'une normalisation de couche. Nous modifions également la sous-couche d'auto-attention dans la pile du décodeur pour empêcher les positions de s'occuper des positions suivantes. Ce masquage, combiné au fait que les plongements de sortie sont décalés d'une position, garantit que les prédictions pour la position i ne peuvent dépendre que des sorties connues à des positions inférieures à i. [7]

L'encoder est une pile de N petits encoders, le decoder une pile de N petits decoder.

Les petits encoders et decoders ont tous la même architecture (mais ils ne partagent pas leur paramètres).

L'encodeur consiste en deux blocs (qui sont tous deux réseaux de neurones) : Une couche dite de « Self-attention » et un réseau à propagation avant (ou Feed-forward Neural Network). Si le second est un réseau de neurones assez connu, le premier l'est un peu moins. La couche de Self-attention est l'élément central de l'architecture du Transformer. Son rôle est de faire garder l'interdependance des mots dans la représentation des séquences. Nous verrons le mécanisme d'attention plus en détail un plus bas. Le décodeur est également composé d'un bloc de Self-attention et d'un Feed-forward mais il contient en plus une couche « EncoderDecoder Attention » qui a pour but de permettre au décodeur de réaliser le mécanisme d'attention entre la séquence d'entrée (encodée) et la séquence de sortie (en train d'être décodée).

- **BERT (modèle de transformers):**

La représentation bidirectionnelle de l'encodeur à partir des transformateurs, également connue sous le nom de BERT, est un modèle de langage pré-formé construit au-dessus des blocs Transformer. BERT peut avoir 12 ou 24 couches, et chaque couche est un encodeur Transformer. BERT est unique car il lit les mots dans les deux sens à la fois. Ce qui lui permettra d'utiliser différentes stratégies telles que : Masked Langage Model (MLM), et Next Sentence Prediction (NSP). Le BERT est composé de deux phases : Pre-Training et Fine Tuning.

. Pre-Training: BERT est déjà pré-formé, tout ce que nous avons à faire est de mettre en œuvre et d'affiner notre modèle dans ce que nous voulons qu'il fasse. En l'occurrence : Résumer.

. Fine Tuning : BERT peut être utilisé pour plusieurs tâches NLP en aval. BERT a déjà une capacité de compréhension de la langue, il est plus facile d'ajuster les poids. Le BERT peut être utilisé pour la classification, la réponse aux questions et toute autre tâche où la prédiction à une certaine position est autorisée à examiner d'autres informations dans les deux sens.

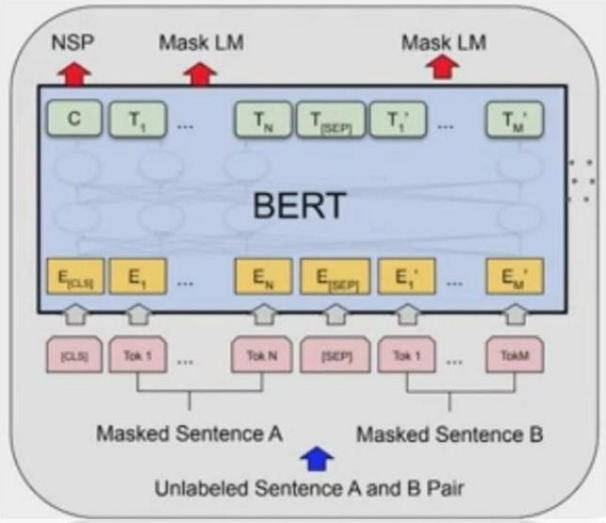


Figure 13 : Structure du BERT

e. Generative Pre-Trained Transformrer : GPT-2

GPT-2 est un modèle basé sur GPT, un grand modèle de langage basé sur un transformateur composé uniquement de blocs décodeurs empilés dans

l'architecture du transformateur. Comme tout modèle de langage traditionnel, GPT2 génère un jeton à la fois. Une fois chaque jeton généré, il est ajouté à la séquence d'entrée, qui est ensuite transmise en tant qu'entrée au modèle, et ainsi de suite. Il existe de nombreuses variantes de GPT-2, chacune avec plus de couches et plus de têtes d'attention que le décodeur Transformer d'origine.

Figure 14 : Représentation de GPT-2

f. Generative Pre-Trained Transformrer: GPT-3

GPT-3 ou Generative Pre-trained Transformer est un modèle de langage autorégressif qui utilise l'apprentissage en profondeur pour produire un texte de type humain. GPT-3 a été annoncé le 28 mai 2020 par OPEN-AI. GPT-3 est le plus grand modèle de langage formé, avec près de 175 milliards de paramètres.

Il y a peu de choses qui sont "défaites" avec GPT-3, c'est tellement puissant que c'est presque dangereux. Qu'il s'agisse de résumer ou de créer des sites Web à partir de rien en comprenant un simple mot, GPT-3 est de loin l'une des plus grandes inventions de l'humanité.

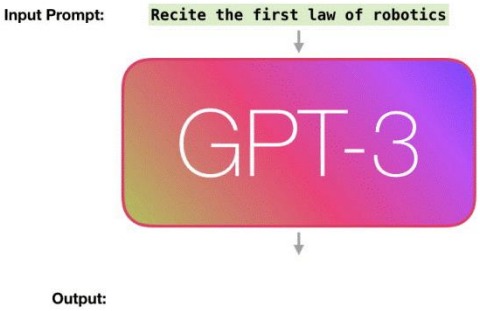


Figure 15 : Représentation de GPT-3

g. Gopher :

Le nouveau leader de l'IA linguistique Gopher, comme GPT-3, est un LLM dense basé sur un transformateur autorégressif - en gros, il prédit le mot suivant en fonction d'un historique de texte. Le modèle a été formé sur MassiveText qui comprend diverses sources comme MassiveWeb (une compilation de pages Web), Wikipedia, GitHub, des livres et des articles de presse.

I.7 conclusion

Dans ce chapitre, nous avons cité les définitions de l’apprentissage automatique, ainsi que sur ses différents type (apprentissage supervisé, apprentissage non supervisé, apprentissage semi-supervisé et apprentissage par renforcement). Nous avons réalisé aussi une étude détaillée sur les réseaux de neurones RNN (LSTM, GRU).

Chapitre II : Concepts de base du résumé automatique de texte

II.1 Introduction

Depuis les années 1950, la recherche en résumé automatique s'est concentrée sur l'extraction de phrases importantes. Les phrases extraites doivent constituer un texte cohérent, fidèle aux idées/informations exprimées dans les documents d'origine.

Dans ce chapitre, nous allons introduire le résumé automatique d'une manière générale. Premièrement, nous allons voir quelques définitions pour la fonction de résumé. Ensuite, nous allons présenter les différents types du résumé automatique. Après, nous allons présenter les étapes du résumé automatique ainsi que les travaux liés au résumé automatique.

II.2 Définition

Le résumé automatique de textes, apparu vers la fin des années 1950, a connu un fort renouveau ces dernières années. Produire automatiquement un résumé pertinent et de qualité nécessite de condenser le ou les documents originaux tout en minimisant la redondance, et en maximisant la cohérence et la cohésion.

Le résumé automatique de texte est une problématique difficile, fortement dépendante de la langue et qui peut nécessiter un ensemble de données d'apprentissage conséquent. L'approche par extraction peut aider à surmonter ces difficultés. (Mihalcea, 2004) a démontré l'intérêt des approches à base de graphes pour l'extraction de segments de texte importants. [9]

Le but du résumé automatique de texte est de générer des représentations abrégées d'un ou plusieurs documents.

Le résumé de texte automatique peut être classifié en deux approches abstraction et extraction.

II.3 Les types de résumé automatique

Divers critères ont été adoptés pour classifier les résumés automatiques. Une classification selon la fonction du résumé regroupe les résumés en indicatif et informatif.

Nous citons les plus importants et les plus fréquemment utilisés dans la littérature.

II.3.1 Résumé dynamique

Le résumé dynamique est une variante du résumé automatique multi-document incluant la dimension supplémentaire du temps. Alors que dans le problème du résumé multi-document les données d'entrée sont statiques, le résumé dynamique introduit une difficulté supplémentaire en faisant varier les données d'entrée sur l'axe du temps. Les travaux sur ce type de résumé peuvent être classés en deux catégories. Les systèmes de résumé dynamiques incrémentaux produisent quant à eux des mises à jour d'un résumé initial à chaque fois que des informations nouvelles apparaissent concernant l'objet du résumé initial. [13]

II.3.2 Résumé mono-document

Dans le résumé mono-document, le système prend un document à la fois et produit un résumé pour chaque entrée qu'il prend [29]. Un même document peut être composé de plusieurs sous-documents avec plusieurs paragraphes. Le contenu décrit de chacun de ces sous-documents met l'accent sur différents aspects entourant tous le même sujet.

II.3.3 Résumé multi-documents

Dans la synthèse multi-documents, nous résumons plusieurs documents qui ont le même sujet [30]. L'utilisation de la synthèse multi-documents sur des documents obtenus après une recherche Google est un autre cas d'utilisation courant. Cependant, c'est un grand défi d'éviter la redondance puisque tous sont plus susceptibles d'inclure un certain degré d'informations similaires. Il existe 2 type de résumé multi-documents :

II.3.3.1 Résumé abstractif

Dans l'abstraction, le texte résumé est une interprétation du texte original, un processus de production qui réécrit le texte source dans une version plus courte en substituant certains concepts. Sa mise en œuvre nécessite l'utilisation de grammaires et de dictionnaires pour l'analyse et la génération, en plus de modélisation de la compréhension humaine du texte. Ce processus est difficile à réaliser. [12]

Les méthodes de résumé abstractives imitent, jusqu'à un certain degré, le processus naturel accompli par l'homme pour résumer un document. Par conséquent, elles produisent des résumés plus similaires aux résumés manuels. Ce processus peut être décrit par deux étapes majeures : la compréhension du texte source et la génération du résumé.

II.3.3.2 Résumé extractif

L'extraction est le processus de sélection d'extraits appropriés (phrases, paragraphes, etc.) du texte original et de les enchaîner dans une forme plus courte. Le texte résumé est extrait du texte sur une base statistique ou en employant des méthodes heuristiques ou une combinaison des deux.

En règle générale, les phrases complètes considérées comme les plus importantes sont extraites du texte source. L'avantage de cette approche est qu'elle est facile à mettre en œuvre, mais il y a des risques Introduisez quelques incohérences dans le résumé.

Le point fort du résumé par extraction est qu'il évite la génération de texte. Ceci permet d'une part, de se concentrer sur la sélection du contenu pertinent et d'autre part, d'obtenir un résumé lisible et linguistiquement correct.

II.4 Processus du résumé automatique

Tous les résumés suivent généralement ces étapes pour générer un résumé :

- Prétraitement
- Traitement
- Post-traitement
- Evaluation

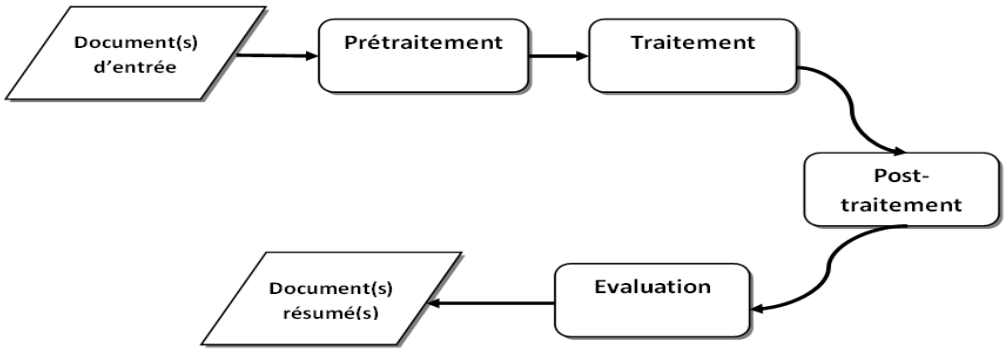


Figure 16 : Procédure de résumé automatique.

Dans cette section, nous décrirons chacune de ces étapes.

II.4.1 prétraitement

Le prétraitement de texte est un ensemble d'étapes pour créer un document dans un format prévisible et analysable. Dans tout texte d'entrée, certains mots et symboles sont liés au sujet et n'ont aucune signification significative et sont souvent utilisés pour lier des mots entre eux. Les occurrences répétées de ces mots peuvent détruire des scores de mots importants. Pour résoudre ce problème, plusieurs méthodes ont été utilisées telles que la tokénisation, la normalisation, etc.

- **Normalisation :**

  Il existe plusieurs techniques pour réduire la taille du vocabulaire. La principale est la normalisation des corpus. La normalisation est une somme de règles de segmentation et de transformation des phrases. Une fois celle-ci définie, elle est appliquée aux corpus.Elle permet de formater et d'orthographier les mots de la même façon, sur l'ensemble du corpus. [5]

  Le formatage et l'orthographe des mots font apparaitre parfois un même mot sous de multiples représentations dans les corpus. Ils sont parfois orthographiés de façons différentes, avec ou sans accent, avec ou sans tiret pour les mots composés, avec ou sans apostrophe et avec ou sans espace etc. Chaque écriture possible devient une entrée de notre vocabulaire si laissée telle quelle. La normalisation permet d'uniformiser leurs représentations sur l'ensemble du corpus et donc de réduire le nombre d'entrées du vocabulaire, qui sont appelées token. [5]

Ex 1 *Exemple : « J'ai un problème de normalisation. »*

Phrase normalisée : « J' ai un problème de normalisation . »

Dans l'exemple 1, la normalisation de «j'ai» est sujette à débat. La plupart du temps, on choisi de normaliser en séparant l'apostrophe du mot qui la suit mais c'est une convention. Le point est séparé du dernier mot de la phrase afin d'éviter d'avoir une entrée du vocabulaire pour « normalisation » et « normalisation . ».

Ex 2 *Exemple : «:'(»*

Version normalisée : «Visage triste»

- **Tokénisation :**

  La Tokénisation est une étape essentielle dans tout système de traitement automatique des langues, d'autant plus que de nombreux outils dépendent du

découpage obtenu [14]. Le but de la tokénisation est de diviser un texte volumineux en un ensemble de phrases, puis de diviser ces phrases pour explorer les mots qu'il contient, donc la liste des jetons devient une entrée pour de nombreux algorithmes d'approche statistique en résumé extractif.

Segmentation en phrases

Dans la plupart des langues écrites, les phrases sont délimitées par des marques de ponctuation comme le point, le point d'exclamation, le point d'interrogation. La phrase est définie par une clause qui commence par une lettre majuscule et se termine par un des trois marques de ponctuation précédentes. Mais, il existe des cas où cette définition ne peut pas être appliquée [8] :

- Le point peut être utilisé dans les abréviations, par exemple "Mr.", "Dr.", "etc.", etc., qui se trouvent dans la plupart des cas au milieu d'une phrase.

- Les phrases peuvent être délimitées par plusieurs autres marques de ponctuation. Il existe des cas où des phrases sont délimitées par des marques autres que les points, comme par exemple les virgules utilisées par les séquences d'actions.

- Dans des langues, comme le Thaï, il n'existe pas de ponctuation pour différencier les limites de phrases.

Plusieurs facteurs contextuels ont été proposés pour aider à la segmentation en phrases, comme :

- Distinction de la casse : les mots commençant par une lettre majuscule donnent une information sur les limites de phrases. Les phrases commencent toujours par une lettre en majuscule.
- Longueur du mot: La longueur de mots avant et après un point, est utilisée par, comme un critère contextuel.
- Préfixes et suffixes: [15] utilisent les préfixes et les suffixes des mots entourant la marque de ponctuation, comme un critère contextuel.

- **Radicalisation :**

La radicalisation tente de réduire un mot vers son radical. L'effet n'est pas seulement celui de réduire les différentes variantes d'un terme vers une seule forme représentative, mais aussi de réduire la taille du vocabulaire utilisée par le système pour stocker les représentations. Dans la plupart des cas, la petite taille du dictionnaire nous permet de préserver l'espace du stockage et le temps de traitement, ainsi de rendre le document moins bruyant, plus compact, et plus souple. [16]

Le résultat d'une radicalisation peut être un mot qui n'a aucun sens, mais qui est commun entre les mots ayant le même sens. Le rendement d'une radicalisation dépend sur la racine résultat; il est bon si les différentes mots avec le même sens de base ont la même racine, et si les mots qui n'ont pas le même sens sont séparés. Selon ces conditions, on peut avoir deux types de problèmes [16] :

— Sur-radicalisation : se passe lorsque deux mots sont donnés la même racine, mais en réalité ils ne l'ont pas.

— Sous-radicalisation : se passe lorsque les mots qui doivent avoir la même forme de base, ne l'ont pas. Par exemple, les deux mots "running" et "ran" doivent avoir la même racine "run", mais le système nous donne "run" et "ran" en ordre.

II.4.2 Traitement

Le processus de génération d'un résumé dépend de nombreux facteurs, tels que le format de l'entrée, le but du résumé, le type de résumé, etc. Par conséquent, il n'existe pas de méthode fixe pour générer des résumés. Plusieurs solutions existent et suivent une certaine approche. Les approches les plus populaires qui existent sont :

L'approche statistique, l'approche basée sur les graphes, l'approche linguistique et l'approche d'apprentissage automatique.

Méthode de résumé automatique :

- **Approche statistique**

Les approches statistiques dépendent des caractéristiques statistiques pour extraire les unités pertinentes du ou des documents d'entrée.

Le principe de la méthode statistique consiste à sélectionner les unités (phrases) saillantes et les combiner pour avoir un résumé. L'approche statistique comprend les trois phases :

La première étape consiste à faire des statistiques sur des critères pour chaque unité (pour un document texte, elle peut être une phrase, un paragraphe, etc.). Pour le résumé d'un document, texte, ces critères peuvent être : la fréquence des mots, la position des mots, les mots de titre, etc.

La deuxième phase consiste à sélectionner les unités saillantes dans le texte, en se basant sur les statistiques précédentes, et en attribuant à chaque unité un score selon ces critères. Enfin, la phase d'extraction qui consiste à éliminer les unités ayant un score très faible, et donc qui ne sont pas pertinentes dans le document.

Les méthodes statistiques sont des méthodes simples, qui attribuent de score en se basant sur certains critères dans le document. Il est plus aisé de combiner un ensemble de critères de natures différentes pour exprimer la valeur de pertinence globale d'une phrase (mais aussi d'un paragraphe, etc.). [8]

L'inconvénient majeur de cette méthode est l'incohérence. Il existe deux grandes origines de l'incohérence : les anaphores et la structure de phrases. Pour les anaphores, on peut trouver dans le résumé un lien anaphorique et qui n'a pas de sens, puisque la phrase qui l'exprime a été éliminée.

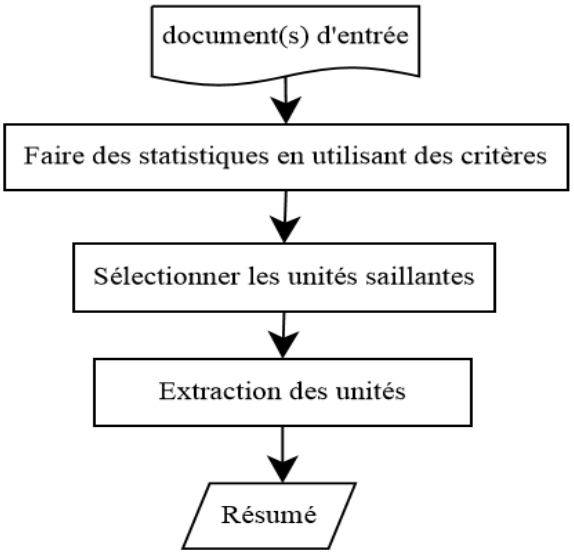


Figure 17 : Résumé par approche statistique.

- **Approche graphique**

Les nœuds dans les approches de résumé basées sur les graphes représentent les phrases et les arêtes représentent la similitude entre les phrases. Les valeurs de similarité sont calculées à l'aide des mots ou des phrases qui se chevauchent. Les phrases les plus similaires aux autres phrases sont choisies dans le cadre du résumé résultant. TextRank et Cluster LexRank sont deux méthodes qui utilisent une approche basée sur des graphiques pour la synthèse de documents.

- **Approche linguistique**

L'approche linguistique utilise des techniques sophistiquées de traitement du langage naturel (TAL) pour générer des résumés[17]. Certaines de ces techniques sont la partie du discours, les relations rhétoriques, la sémantique, les chaînes lexicales, etc. Elle est plus puissante que l'approche statistique car elle intègre un traitement plus élaboré du texte d'entrée. Nenkova et McKeown [18] a suggéré de l'utiliser comme une tâche de post-traitement pour améliorer la qualité linguistique du résumé généré plutôt que comme une tâche de traitement.

Pour un document (ou plusieurs) d'entrée, le système utilise les informations linguistiques pour créer une représentation de l'entrée. Ensuite, cette représentation va être réduite en utilisant des règles de réduction, soit en gardant les phases les plus importantes ou en créant une nouvelle représentation. Enfin, l'étape de génération du résumé, qui sert à fusionner les phrases extraites pour avoir un résumé extractif, ou transformer la représentation réduite à un résumé par abstraction. La figure 13 illustre les différentes étapes suivies dans l'approche linguistique. [8]

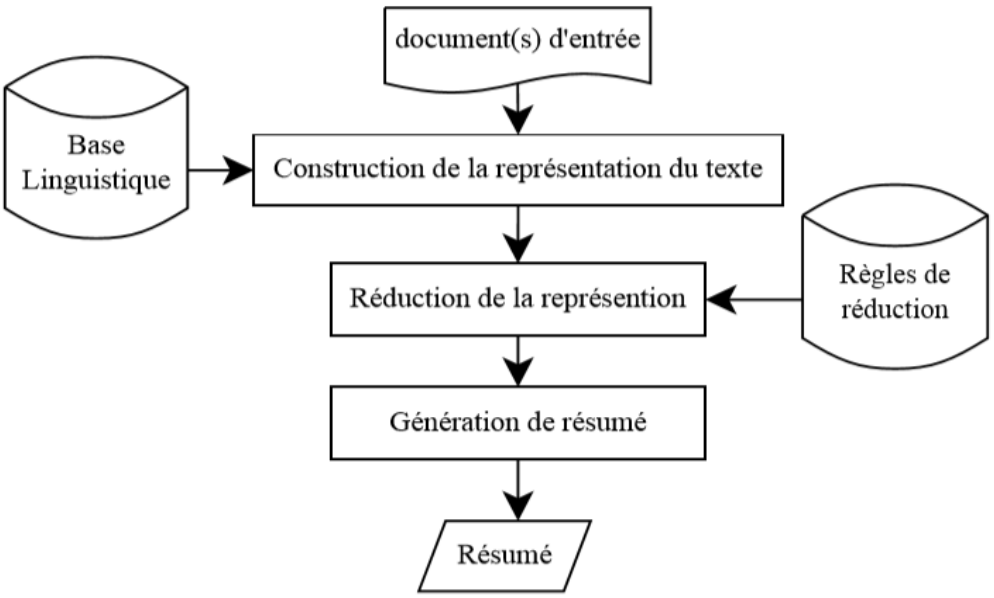


Figure 18 : Un modèle général pour l'approche linguistique de résumé automatique.

L'utilisation de cette approche nous garantit une analyse plus profonde du texte d'entrée; on peut examiner les liens sémantiques entre les mots (les synonymes, les antonymes), le sens des phrases, les différentes concepts existants et les relations entre eux, etc. Un autre point fort de cette approche, est le respect de l'acheminement de l'auteur; on peut extraire les différentes idées existantes dans le texte et générer un résumé qui respecte leur évolution.

- **Approche d'apprentissage automatique**

L'apprentissage automatique consiste à amener les machines à effectuer des tâches que les humains peuvent faire mieux et plus rapidement. En ce qui concerne les humains, cela est réalisé en faisant apprendre à la machine à l'aide de données. Il peut être appliqué à un large éventail de domaines tels que la reconnaissance d'images, la reconnaissance

vocale, le résumé automatique de texte, etc. Le processus d'utilisation des données fait référence à la formation, tandis que l'exécution des tâches fait référence à la prédiction ou à l'inférence. Les techniques d'apprentissage automatique utilisent des algorithmes complexes qui s'améliorent d'eux-mêmes en observant les données sur lesquelles ils sont formés, puis ils sont utilisés pour prédire sur la base de données invisibles [19].

II.4.3 Post-Traitement

Le post-traitement est une étape généralement utilisée pour améliorer la qualité et la lisibilité du résumé généré. Il peut supprimer la redondance, résoudre d'éventuelles incohérences concernant les pronoms, etc. La résolution d'anaphores est une méthode utilisée pour remplacer les pronoms dans une phrase par le sujet auquel ils se réfèrent dans la phrase précédente, elle est considérée comme un pré-traitement ainsi que une technique de post-traitement [20]. L'élimination de la redondance concerne soit la suppression des phrases répétitives en termes de sens, soit la suppression des mots répétitifs consécutifs dans la même phrase, résultant en un résumé plus compressé et clair. Les techniques de post-traitement ne sont pas toujours utilisées par rapport à celles qui sont généralement cruciales pour cette tâche.

- **Résolution des anaphores**

  Dans le résumé par extraction, les phrases extraites pourraient contenir des références anaphoriques non résolues. Par exemple, supposant qu'on a le texte suivant :

  "Cosette était à sa place ordinaire, assise sur la traverse de la table de cuisine près de la cheminée. Elle était en haillons, elle avait ses pieds nus dans des sabots, et elle tricotait à la lueur du feu des bas de laine destinés aux petites Thénardier." [21]

  Si le résumé contient uniquement la deuxième phrase, on perd la référence "Elle", et donc ça va diminuer la qualité du résumé. C'est pourquoi, il faut récupérer la référence de ce pronom, et le remplacer par cette référence, mais le deuxième pronom dans la phrase doit rester tel qu'il est. [8]

II.4.4 Evaluation

L'évaluation d'un résumé est une tâche difficile car il n'existe pas de résumé idéal pour un document ou un ensemble de documents donné. L'objectif de tout système de synthèse est d'optimiser la couverture et la lisibilité des sujets.

Les méthodes d'évaluation de résumé de textes peuvent être classées en deux catégories. La première est l'évaluation intrinsèque, qui consiste à évaluer le système de résumé en interne. Elle s'occupe surtout de l'évaluation de cohérence et le contenu informatif des résumés produits. La deuxième est l'évaluation extrinsèque, qui consiste à tester l'impact de résumé sur les tâches comme l'évaluation de pertinence, la compréhension en lecture, etc. La figure 14 représente les différentes catégories de l'évaluation d'un résumé automatique. [8]

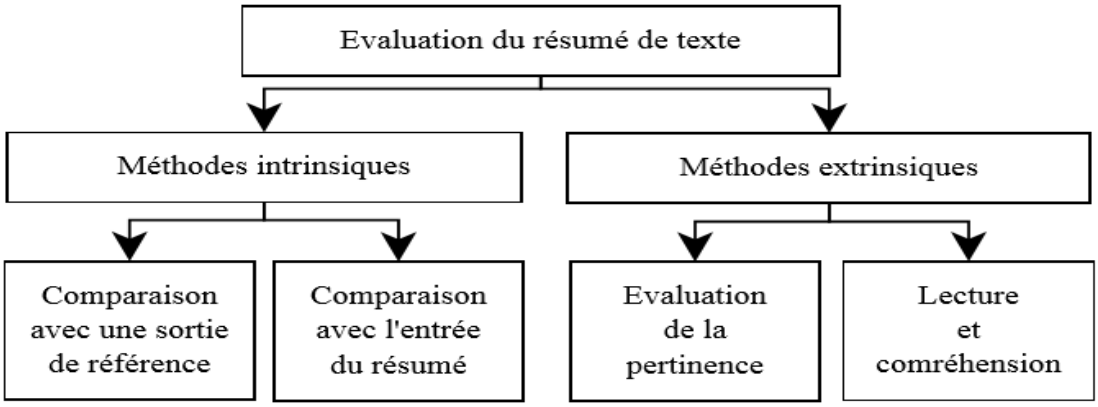


Figure 19 : Les approches d'évaluation des systèmes de résumé automatique.

On distingue deux types d'évaluation : les techniques intrinsèques et les techniques extrinsèques :

- **Évaluation intrinsèque**

L'évaluation intrinsèque vise à évaluer le système en lui-même. Celui-ci détermine la qualité du résumé en se basant sur une comparaison entre le résumé généré automatiquement et le document original, ou à un résumé manuel créé par l'homme. On compare le résumé produit par rapport au résumé de référence ou carrément avec le document source selon différentes critère. Le premier est la cohérence, qui consiste à la vérification de la lisibilité du résumé. Pour les résumés par extraction, les problèmes sont dus à la présence des anaphores et des brèches dans leurs structures rhétoriques. L'autre critère est le contenu informatif du résumé, qui vise à estimer les informations que le résumé contient. [8]

L'évaluation intrinsèque mesure les propriétés concernant la nature du sujet à évaluer et son objectif, alors que l'évaluation extrinsèque mesure les aspects concernant les impacts et les effets de sa fonction.

Les approches de l'évaluation intrinsèque

- Comparaison avec une sortie de référence :

L'idée est de comparer le résumé automatique avec celui fait par un humain d'un même texte source. L'évaluation classique de [22] a été réalisée par humain en comparant le résumé par machine avec celui crée par un humain. Le problème pour l'évaluation humain est que, même si on peut trouver plusieurs résumés de référence pour un même document, le système peut générer un résumé qui est différent de tous ces résumés de référence, mais qui est informatif et cohérent [23]. On peut aussi évaluer un résumé de manière automatique, en utilisant des différentes mesures, parmi elles ROUGE, Pyramides, BE.

- **ROUGE**

ROUGE évalue les résumés en le comparant à des résumés modèles. Cette comparaison est automatique et ne nécessite pas de prétraitement particulier. Elle est déduite à partir du recouvrement entre les N-grammes des deux textes. Elle utilise trois métriques pour quantifier la comparaison [13]. Il existe plusieurs variantes de ROUGE, telles que ROUGE-N, ROUGE-L, ROUGE-S :

- ROUGE-N mesure le chevauchement des unigrammes, des bigrammes, des trigrammes et des n-grammes d'ordre supérieur.

- ROUGE-L mesure la séquence de mots correspondante la plus longue en utilisant les sous-séquences communes les plus longues (LCS). L'un des avantages de l'utilisation de LCS est qu'il ne nécessite pas de correspondances consécutives, mais des correspondances en séquence qui reflètent le niveau de la phrase.

- ROUGE-S est n'importe quelle paire de mots dans une phrase dans l'ordre, permettant des espaces arbitraires.

- **PYRAMID**

Cette méthode permet de comparer un résumé candidat à un ensemble de résumés de référence. Une pyramide est une représentation du résumé de référence. Il représente également les opinions de plusieurs auteurs, dont chacun a écrit un modèle de résumé. [13]

La principale caractéristique d'une pyramide est qu'elle quantifie l'accord entre les abstraits humains. Puisque nous l'utilisons pour évaluer le contenu du résumé, les unités de comparaison dans une pyramide correspondent aux unités de sens. Par conséquent, une SCU est un ensemble d'unités textuelles d'abrégés de référence exprimant la même information. Il a un poids égal au nombre de résumés de référence qui l'instancient. Ces SCU sont organisées en pyramide où chaque niveau regroupe des SCU de même poids. [24]

- **BE**

Les éléments basiques (Basic Elements : BEs) sont des unités sémantiques minimales qu'on peut obtenir d'une phrase. D'après [26], le problème d'évaluation du contenu d'un résumé peut être résolu en utilisant trois différentes modules : découpeur de BE, comparateur de BE, et évaluateur de BE. Le premier sert à créer les unités BE d'un texte d'entrée, le deuxième sert à évaluer la similarité entre deux BE, et le troisième sert à donner un score pour chaque BE.

Le système d'évaluation utilisant les BEs prend le résumé et un ensemble des résumés de référence pour avoir un score. Il applique les trois modules précédents deux fois, en deux étapes : la préparation et la notation. Dans la phase de préparation, le premier module décompose les résumés de référence en une liste de BEs de référence; le second module prend en compte tous BEs de référence et fusionne ceux sémantiquement identiques et le troisième module attribue un score à chacun des BEs de référence. Dans la deuxième étape (notation), le premier module décompose le résumé en une liste séparée de BEs, le second compare chaque BE à la liste des BEs de référence, le troisième attribue un score à chaque BE et calcule le score global de tous les BEs contenus dans le résumé candidat. [8]

- Comparaison avec l'entrée de résumé :

Dans ce type d'évaluation, le résumé et la source sont donnés à des personnes, en leurs demandant d'évaluer le contenu informatif du résumé dans le contexte de la source. Selon [23] il existe deux types de méthodes pour la comparaison entre le résumé et la source : les méthodes sémantiques et les méthodes de surface. Les méthodes sémantiques consistent à comparer le sens dans le texte source par rapport à celui du résumé. Une méthode est de marquer le sens de chaque phrase dans le résumé, ensuite voir combien de propositions existantes dans la source ce résumé couvre.

Évaluation extrinsèque

L'évaluation extrinsèque est une méthode qui détermine l'impact de la synthèse sur d'autres tâches [25].Si la tâche peut être effectuée à l'aide du résumé au lieu du document d'origine, cela signifie que le résumé inclut toutes les informations pertinentes dans le document d'origine. Certaines de ces tâches sont la catégorisation de documents, la récupération d'informations et la réponse aux questions. [24]

L'idée d'une évaluation extrinsèque d'un résumé est de déterminer l'effet de résumé sur d'autres tâches. Il existe plusieurs tâches sur lesquels un résumé peut être appliqué, parmi ces tâches on peut citer celles mentionnées dans [23] :

— Si le résumé affecte le comportement d'autres tâches, il est possible de mesurer l'efficacité en exécutant ces tâches. Par exemple, si on a un système de décision qui se base sur notre système de résumé automatique, on peut mesurer l'efficacité de notre système de résumé en examinant son effet sur le système de décision.
— On peut examiner l'utilité de résumé avec respect des informations de besoin ou d'objectif, comme trouver des documents pertinentes au besoin d'une personne issus d'une large collection.
— On peut évaluer l'impact d'un résumé sur le système qu'il le contient, par exemple, comment un outil de résumé peut aider dans un système de question-réponse?

- **Les domaines d'applications des résumés automatiques :**

Nombreux sont les domaines d'outils du résumé automatique, en effet, on peut recenser les usages suivants [27] :

- Pour résumer les nouvelles au SMS pour les téléphones portables.
- Pour laisser un ordinateur synthétique lu le texte résumé. Le texte écrit peut être long et ennuyeux pour le lire.
- Dans des moteurs de recherche pour présenter les descriptions compressées des résultats de recherche.
- Pour chercher dans des langues étrangères et obtiennent un résumé automatiquement traduit du texte automatiquement résumé.

Sans oublié les domaines de l'archivage, des bibliothèques et du journalisme.

II.5 les travaux liés au résumé automatique

Plusieurs travaux ont été réalisés dans le but de développer des systèmes de résumé automatique. Le premier objectif est d'obtenir des textes plus clairs et précis, faciles à comprendre et bien structurés. Tandis que le deuxième objectif est d'aider à éviter la difficulté de lire des textes trop longs.

Des chercheurs du RALI, sous la direction de Guy Lapalme, travaillent dans le domaine du résumé automatique depuis plusieurs années.

D'ailleurs depuis 2002, le RALI a participé à plusieurs des compétitions de Document Understanding Conference (DUC) et plus récemment de Text Analysis Conference (TAC). [28]

Les principaux travaux au domaine du résumé automatique [28]:

▪ SumUM :

SumUM a été développé par Horacio Saggion dans le cadre de sa thèse de doctorat (1997-2000). SumUM génère de courts résumés (10-15 lignes) de longs documents (15-20 pages) scientifiques et techniques. SumUM produit le résumé en deux étapes: l'utilisateur reçoit d'abord un résumé *indicatif,* qui identifie les sujets importants du document et le système génère ensuite un résumé *informatif* qui élabore quelques sujets choisis par l'utilisateur.

L'entrée du système est un article scientifique en anglais, contenant les éléments structuraux suivants: titre de l'article, auteur et affiliation, introduction, sections principales, conclusion, bibliographie et remerciement. La sortie du système est un court résumé indicatif composé de phrases complètes. Ce résumé n'est pas qu'un simple extrait de phrases du texte original, il est régénéré à partir des informations trouvées. Il est de qualité comparable à celle des résumés d'auteur. Il est ensuite possible d'obtenir des informations supplémentaires sur des sujets identifiés par l'usager.

▪ Lakhas :

Fouad Douzidia a développé Lakhas (signifiant résumé en arabe), un système de résumé de textes journalistiques arabes basé sur la combinaison de méthodes d'extractions utilisées jusqu'ici en anglais.

Lakhas a également été utilisé pour produire de très courts résumés en arabe lors de l'évaluation à DUC2004. Ces résumés étaient ensuite traduits en anglais avec un système de traduction automatique. Malgré le fait d'avoir suivi un chemin différent des autres compétiteurs, les résultats de l'évaluation ont été excellents et même les meilleurs lorsqu'on disposait du même système de traduction automatique que celui qui avait été utilisé pour traduire les autres textes.

▪ LetSUM (Legal text Summarizer) :

En collaboration avec le groupe LexUM, qui faisait alors partie du Centre de recherche en droit public de la Faculté de Droit de l'Université de Montréal, Atefeh Farzindar a étudié la problématique des résumés de textes juridiques, plus particulièrement les jugements. La méthodologie repose sur l'exploitation de la structure thématique des décisions juridiques afin de constituer automatiquement une fiche de résumé augmentant la cohérence et la lisibilité du résumé. LetSUM permet aux juristes de consulter rapidement les idées clés d'un jugement pour trouver les jurisprudences pertinentes.

▪ Résumé par abstraction :

Pierre-Étienne Genest a participé à plusieurs compétitions TAC (2008 à 2011), notamment en explorant une approche symbolique au résumé développée dans son mémoire de maîtrise en 2009.

Il a par la suite élaboré un système automatique de rédaction de résumés entièrement par abstraction, dans le domaine journalistique. Il extrait d'abord les éléments d'information importants au résumé, pour une catégorie de documents à la fois. Des heuristiques sont utilisées pour filtrer et sélectionner le contenu retenu pour le résumé. Enfin, un plan de génération et des patrons de génération permettent de réaliser le texte du résumé en langue naturelle.

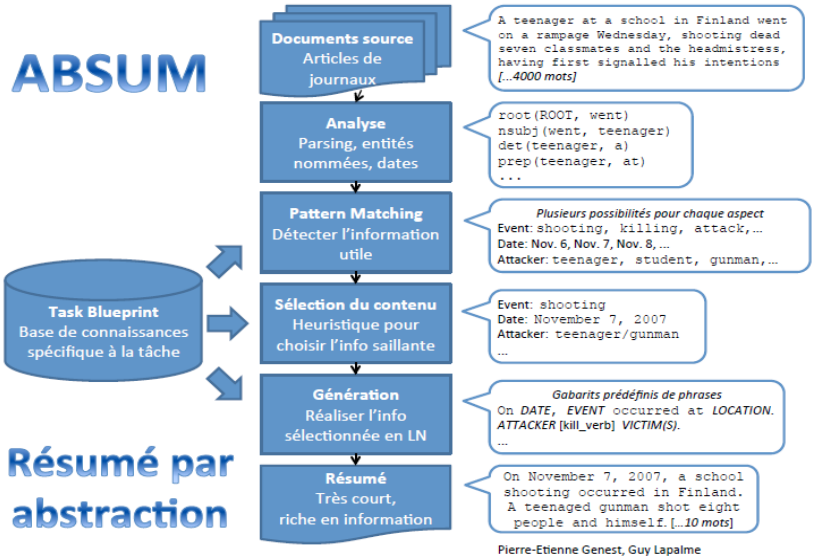


Figure 20 : Schéma explicatif de résumé par abstraction.

▪ Résumé par citation :

Le projet de doctorat de Bruno Malenfant combine et modifie des techniques de résumé automatique pour construire un résumé d'un article de référence (RP) à partir de l'information que d'autres chercheurs ont retenue en analysant le texte des citations vers RP pour constituer la base du résumé. Le résumé de RP sera donc construit à partir de l'analyse des contextes de citation dans les articles citant (CP) qui décrivent le type de lien avec RP. Comme cette information n'était pas disponible lors de l'écriture de l'article, elle ajoute un niveau d'interprétation de RP et fournit un indice sur son apport à la communauté scientifique.

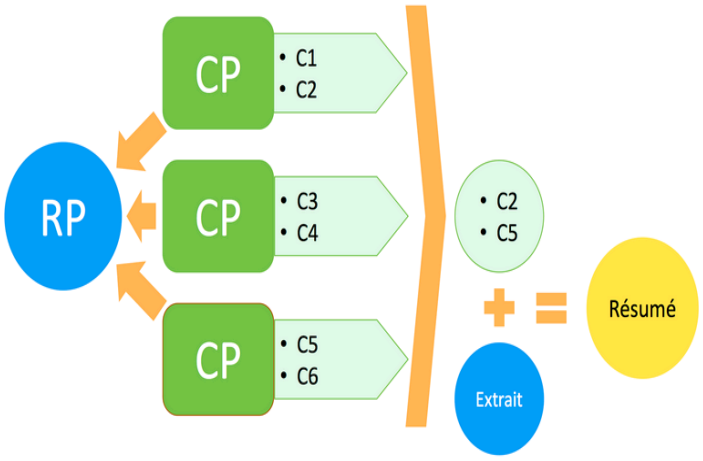


Figure 21 : Schéma explicatif de résumé par citation.

Dans notre modèle on utilise l'algorithme TextRank :

- **Algorithme TextRank :**

  Parmi les différentes méthodes de résumés basées sur des graphiques, l'algorithme TextRank, très cité, c'est un algorithme extractif non supervisé typique, qui a été inspiré à l'origine par le célèbre PageRank de Google en prenant les similitudes entre les phrases comme un type de recommandation ou en votant pour construire les graphiques correspondants. Le résumé de sortie sera alors composé des phrases sélectionnées qui sont classées en haut en fonction de leurs poids convergés après un certain nombre d'étapes de calcul itératives. Sur la base de la proposition originale de TextRank, certains algorithmes améliorés pour le résumé de document unique ont été proposés.

La figure ci-dessous illustre les étapes prises dans le système utilisé :

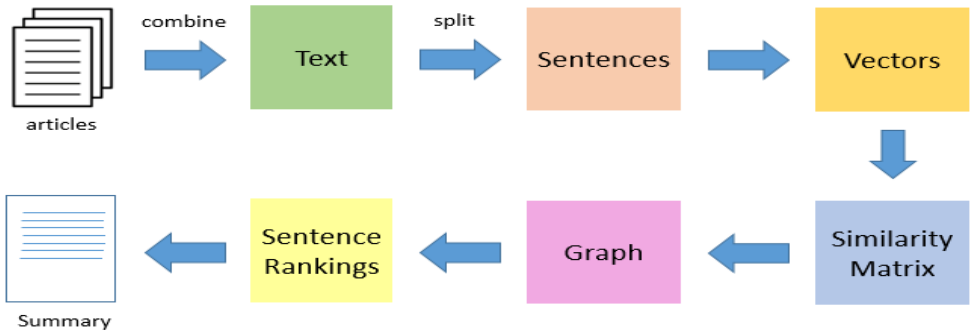


Figure 26 : illustration explicatif d'algorithme TextRank.

1. La première étape consiste à relier tout le texte de l'article.
2. Ensuite, divisez le texte en phrases individuelles.
3. Dans l'étape suivante, nous trouverons une représentation vectorielle (mot inclus) pour chaque phrase.
4. Ensuite, calculez la similarité entre les vecteurs de phrases et stockez-les dans un tableau.
5. Transformez ensuite la matrice de similarité en un graphique, avec les phrases comme sommets et les degrés de similarité comme arêtes pour calculer les niveaux de phrase.
6. Enfin, un certain nombre de phrases d'ordre supérieur constitue le résumé final.

Conclusion

Le résumé automatique de texte consiste à générer une version plus courte d'un ou de plusieurs documents. Dans ce chapitre, nous avons présenté un état de l'art sur le résumé automatique. Dans un premier temps, nous avons présenté quelques notions pour le résumé automatique, afin de comprendre ce domaine. Il peut appartient à plusieurs classes ou types passant par des étapes, et utilisant des différentes méthodes. Donc le système de résumé automatique est un domaine très important pour faciliter l'accès à un résumé bien compris et bien précis rapidement.

Chapitre III : Approche proposée

III.1 Introduction

Au cours des dernières années, grâce à des progrès importants réalisés dans le développement des méthodes neuronales, la majorité des approches de résumé automatique est progressivement passée des techniques d'extraction aux techniques abstractives. Les méthodes extractives sélectionnent les phrases les plus pertinentes du texte d'entrée et les concatènent pour obtenir le résumé. D'autre part, les approches abstractives visent à générer des résumés comme le font les humains en paraphrasant les phrases les plus cruciales et en générant éventuellement de nouveaux mots.

Le travail de Rush, Chopra, and Weston, (2015) [29] sur l'application de la traduction automatique neurale au résumé est le premier à susciter un nouveau moyen de construire des systèmes de résumé abstractifs. Depuis, ce modèle neuronal séquence-à-séquence est devenue la technologie de base de la plupart des systèmes abstractifs modernes. [30]

Ce chapitre est consacré pour la présentation de l'approche proposée et nous discuterons en détail sur les principes de fonctionnement de notre solution basée sur les réseaux de neurone pour créer automatiquement des résumés multi-documents.

III.2 Vue globale de l'approche

Pour la génération du résumé multi-documents un aperçu global de notre approche est illustre dans la figure ci-dessous. (Figure 22)

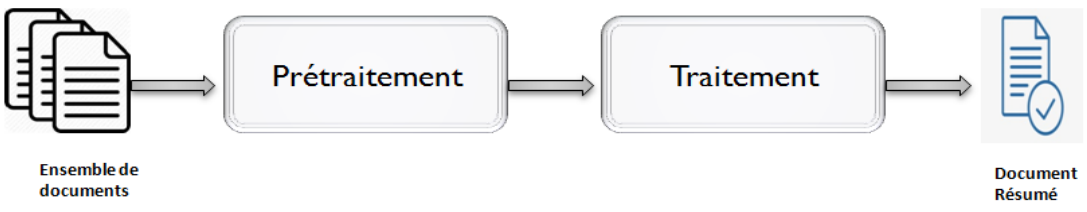


Figure 22 : Illustration de schéma globale de notre approche

Les différentes phases que constitue notre approche vont être définies par la suite :

1. Phase de prétraitement.
2. Phase de traitement (Fine-Tuning/ Clustering).

III.3 Pré-traitement

Le prétraitement est une phase primordiale dans le traitement automatique de texte. Dans notre cas, nous allons exploiter deux modèles préconçus par Google et huggingface pour générer le résumé d'un document. Dans la phase de prétraitement, nous allons préentrainer ces deux modèles sur plusieurs datasets.

III.3.1 Approche avec Bart

BART (Bidirectional and Auto-Regressive Transformers) est un auto-encodeur qui mappe un document résumé au document d'origine dont il est dérivé. Il est implémenté comme un modèle séquence à séquence avec un encodeur bidirectionnel sur le texte résumé et un décodeur autorégressif de gauche à droite.

Il utilise une architecture seq2seq/NMT standard avec un encodeur bidirectionnel (comme BERT) et un décodeur de gauche à droite (comme GPT).

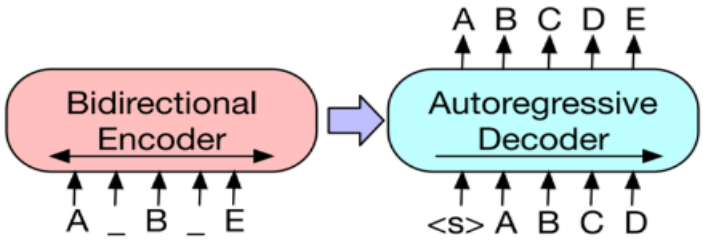


Figure 23 : L'architecture de BART

Pré entraînement

les entrées du codeur n'ont pas besoin d'être alignées sur les sorties du décodeur, ce qui permet d'avoir une sortie de taille différente par rapport à l'entrée. Idéal pour des systèmes de traduction ou de résumé.

Pour entraîner l'encodeur, le document original est corrompu en remplaçant certaines parties du texte par des vides (symboles de masque).

Le décodeur essaiera par la suite de compléter le texte et retrouver le document original en calculant les différentes propababiltés d'apparition des mots. Il utilisera pour cela les différents contextes possibles pris sur de gros datasets. On parle ici de pré-entrainement.

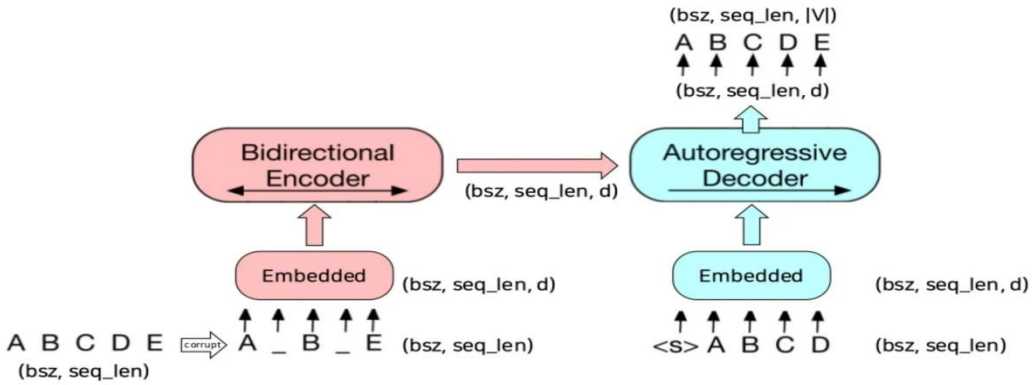


Figure 24 : illustration explicatif de pré-entrainement BART

Afin d'améliorer les résultats du décodeur, une phase de fine-tuning est nécessaire. Ici, un document non corrompu est entré à la fois dans l'encodeur et le décodeur. Les résultats du décodeur seront améliorés à travers les deux documents.

III.3.2 Approche avec Pegasus

PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive summarization Sequence-to-sequence) a été spécialement conçu pour le résumé abstractif et est pré-enteainé avec un objectif de génération de phrases vide. Dans cette tâche, des phrases entières sont masquées du document source, concaténées et utilisées comme « résumés » cible.

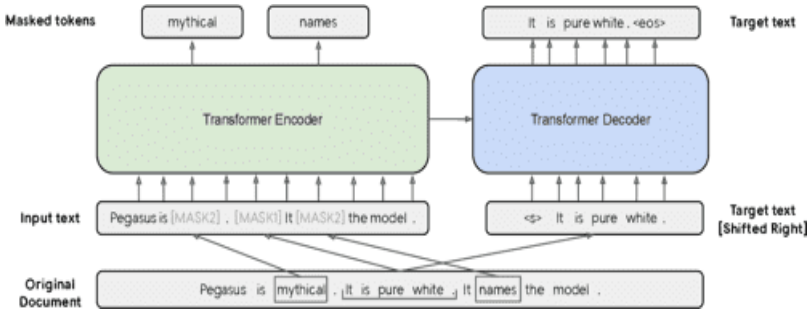


Figure 25 : Structure de Pegasus

- **Pré entraînement**

Au lieu de masquer quelques mots comme dans BART, Pegasus utilisé la technique GSG (The Gap Sentences Generation) où on masque une phrase entière. On sélectionne et on masque des phrases entières à partir de documents, et on concaténe les phrases vides dans un pseudo-résumé. La position correspondante de chaque phrase vide sélectionnée est remplacée par un jeton de masque [MASK1] pour informer le modèle. Pour se rapprocher encore plus d'un résumé, les phrases qui semblent être importantes/principales pour le document sont masquées. Celà permettra de se focaliser sur les phrases les plus importantes du texte.

Nous considérons 3 stratégies principales pour sélectionner m phrases vides sans remplacement à partir d'un document, D = {xi} n, composé de n phrases :

Random Sélectionnez uniformément m phrases au hasard.

Lead Sélectionnez les m premières phrases**.**

Principal Sélectionnez les phrases les mieux notées en fonction de leur importance.

Pour le pré-entrainement nous avons considéré deux grands corpus de textes :

- ✓ **C4 :** la version Colossal and Cleaned of Common Crawl, introduit dans Raffel et al. consiste en texte de 350 millions de pages Web (750 Go).

- ✓ **HugeNews :** un ensemble de données de 1,5 milliard d'articles (3,8 To) collectés à partir de sites Web d'actualités et similaires de 2013 à 2019. Une liste blanche de domaines allant des éditeurs d'actualités de haute qualité aux sites de moindre qualité telle que les journaux de lycées et les blogs a été organisée et utilisée pour lancer un robot d'exploration Web. Des heuristiques ont été utilisées pour identifier les articles de type actualité, et seul le texte principal de l'article a été extrait en texte brut.

Elimination des phrases redondantes et non pertinentes :

III.3.3 segmentation du document

La segmentation du document est une étape nécessaire pour la tâche du résumé automatique. Cette étape consiste à hiérarchiser et à structurer le texte source en différentes unités (titres, sections, paragraphes et phrases).

Depuis une dizaine d'années, de nombreuses méthodes ont été proposées pour segmenter automatiquement des textes. Elles se distinguent principalement par le type d'indices employés. Certaines se basent exclusivement sur une analyse de la cohésion lexicale alors que d'autres prennent également en compte des dispositifs linguistiques qui ont pour fonction de signaler la présence de changements de thèmes. Une autre distinction importante oppose les approches qui s'appuient exclusivement sur les informations contenues dans le texte à segmenter et celles qui ont recours à des connaissances acquises par ailleurs. La méthode proposée par Choi se base exclusivement sur la cohésion lexicale, mais existe dans deux versions correspondant à ce second critère de différenciation.

La procédure de Choi est composée de trois étapes. Tout d'abord, le document à segmenter est découpé en unités textuelles minimales, habituellement les phrases. Les mots composant ces phrases sont soumises à différents traitements comme la suppression de mots peu informatifs sur le thème du texte (article, pronom, verbes très fréquents, ...) et une lemmatisation. Ensuite une mesure de similarité entre toutes les paires d'unités prises deux à deux est calculée. Enfin, le document est segmenté de façon récursive en fonction des frontières entre les unités textuelles qui maximisent la somme des similarités moyennes à l'intérieur des segments ainsi constitués.

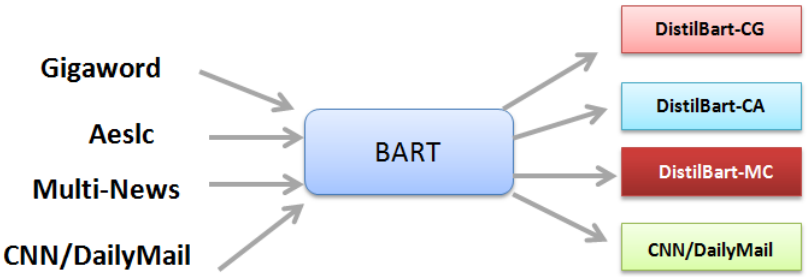
III.4 Traitement (Intégration)

III.4.1 Mono-document

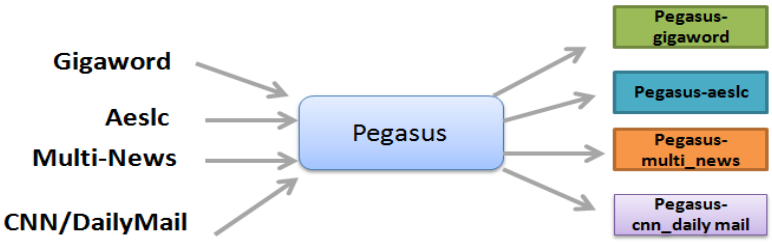
- **Fine-Tuning**

Fine-tuning est un moyen d'application ou d'utilisation de l'apprentissage par transfert. Plus précisément, Fine-tuning est un processus qui prend un modèle qui a déjà été formé pour une tâche donnée, puis ajuste ou modifie le modèle pour lui faire effectuer une deuxième tâche similaire. Pour faire simple : Fine-Tuning prend un modèle pré-formé et le peaufine pour effectuer une deuxième tâche similaire, dans ce cas : Résumer. Nous avons affiné notre modèle pré-entraîné (BART, Pegasus) avec plusieurs Datasets, qui sont : (Aeslc, Grand_brevet, Somme d'argent, CNN/DailyMail).

✓ **BART :** On génère avec BART 4 modèles différents par rapport au datasets utilisé.



✓ **Pegasus :** On génère avec Pegasus 4 modèles différents par rapport au datasets utilisé.



III.4.2 Multi-document

● Clustering

L'analyse de clusters (clustering) est classé comme une technique d'apprentissage automatique non supervisée qui regroupe des points de données dans la création de partitions basées sur la similarité.

Il existe un grand nombre d'algorithmes de clustering. Ces algorithmes visent à explorer la structure interne des données et de les partitionner en groupes plus ou moins homogènes dans nos cas nous avons choisi de travailler avec quatre algorithmes.

Pour rassembler les phrases, on doit d'aborde les représenter sous forme numérique. Pour cette raison, on utilise le modèle *all-MiniLM-L6-v2* basé sur BERT pour trouver une représentation vectorielle adéquate de chaque phrase.

✓ Modèle all-MiniLM-L6-v2

Il s'agit d'un modèle de transformation de phrases : il mappe les phrases et les paragraphes dans un espace vectoriel dense de 384 dimensions et peut être utilisé pour des tâches telles que le regroupement ou la recherche sémantique.

Les modèles all-* ont été entraînés sur toutes les données d'entraînement disponibles (plus d'un milliard de paires d'entraînement) et sont conçus comme des modèles à usage général. Le modèle all-mpnet-base-v2 offre la meilleure qualité, tandis que all-MiniLM-L6-v2 est 5 fois plus rapide et offre toujours une bonne qualité. [46]

L'utilisation de ce modèle devient facile lorsque vous avez installé des sentence-transformers.

✓ Approche de clustering Pairwise

Les méthodes de regroupement pairwise partitionnent un ensemble de données en utilisant la similarité par paires entre les points de données. La matrice de similarité pairwise peut être utilisée pour définir une marche aléatoire de Markov sur les points de données. Ce point de vue forme une interprétation probabiliste des méthodes de regroupement spectral.

✓ **Approche de clustering HCA**

Le clustering hiérarchique, également connu sous le nom d'analyse de cluster hiérarchique, est un algorithme qui regroupe des objets similaires en groupes appelés clusters. Le point final est un ensemble de clusters, où chaque cluster est distinct de l'autre cluster, et les objets au sein de chaque clusters ont globalement similaires les uns aux autres.

La classification ascendante hiérarchique(HCA) est une méthode de classification automatique utilisée en analyse des données. La HCA procède par classification itérative des données du minimum de critères de regroupements au maximum des critères de regroupements jusqu'à ce que tous les objets soient regroupés en classe.

Les regroupements successifs produisent un arbre de classification appelée Dendrogramme. Ce qui, en fin de compte, donne une idée sur les classes et la manière de regrouper.

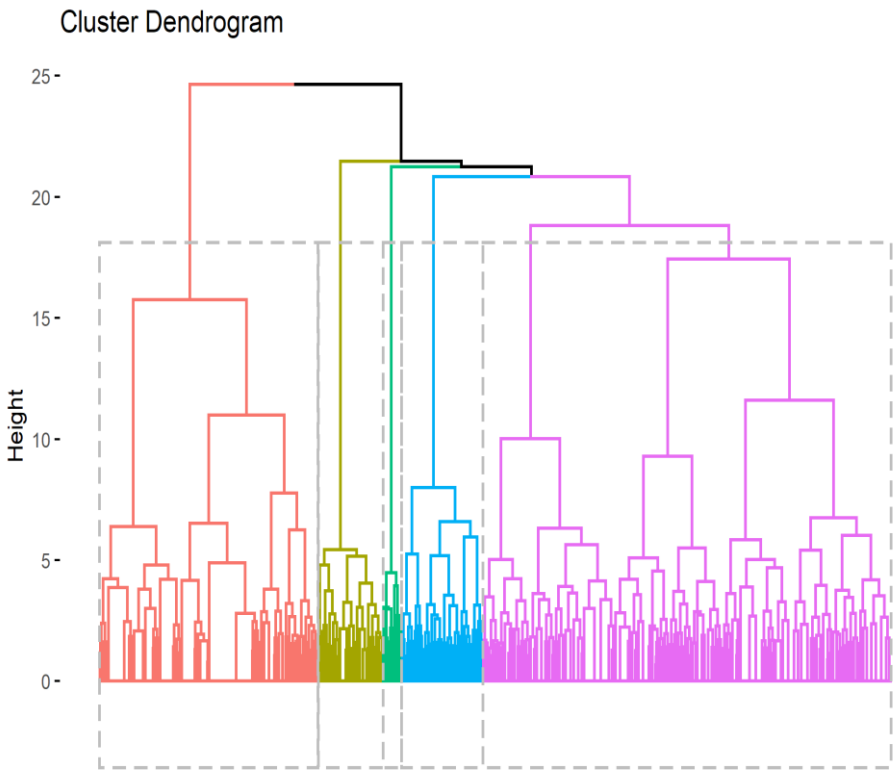


Figure 27 : Analyse de cluster hiérarchique

- **DUC Document Understanding Conferences 2004 (**revoir dans chapitre 4**)**

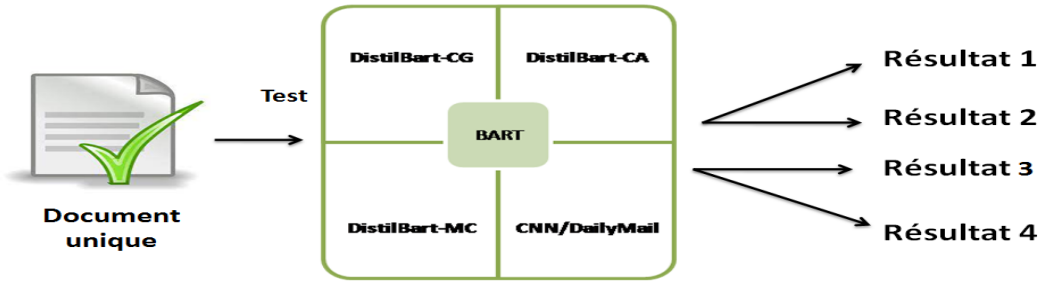
On peut choisir les phrases les plus importantes (pertinents) des documents via l'algorithme ***TextRank*** utilisés dans les moteurs de recherche , pour cela on applique ce algorithme sur notre dataset ***DUC 2004*** pour évaluer nos modèles, suivie les phases suivantes :

✓ **Phase 1 :** Eliminer les phrases non pertinentes de tous les documents avec l'algorithme ***TextRank :***

✓ **Phase 2 :** On utilise les modèles précédentes (résultant de BART et Pegasus (voir dans prétraitement)) sur le nouveau document ***Document unique*** :

Document unique : C'est un document qui contient que les phrases les plus importants (pertinentes) de tous les documents.

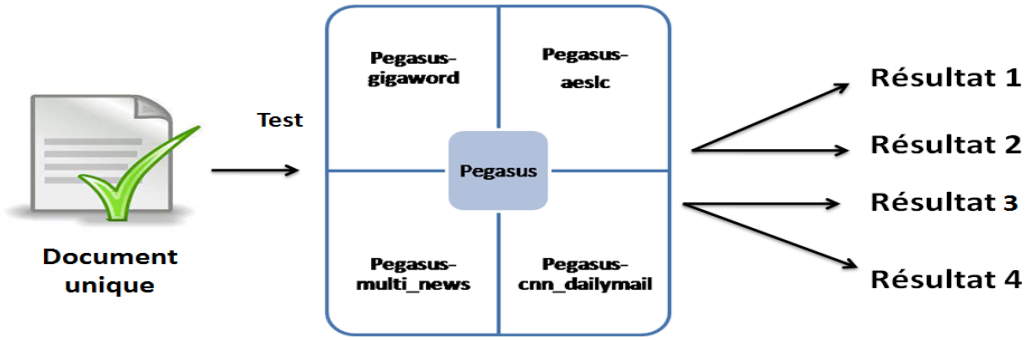
- **Avec BART :**



Tant que:

- DistilBart-CG le modèle résultant de dataset « Gigaword ».
- DistilBart-CA le modèle résultant de « dataset Aesl ».
- DistilBart-MC le modèle résultant de dataset « Multi-news ».
- CNN/DailyMail le modèle résultant de dataset « CNN/DailyMail ».

- **Avec Pegasus :**



Tant que:

- pegasus-gigaword le modèle résultant de dataset « Gigaword ».
- pegasus-aeslc le modèle résultant de « dataset Aesl ».
- pegasus-multi_news le modèle résultant de dataset « Multi-news ».
- pegasus-cnn/dailymail le modèle résultant de dataset « CNN/DailyMail ».

III.5 Conclusion

Dans ce chapitre, nous avons abordé tout ce qui est en rapport avec les étapes nécessaires pour construire et créer un bon modèle de résumé. Nous avons brièvement décrit chaque étape et proposé des méthodes qui ont été utilisées lors du prétraitement dans la génération de résumés multi-documents.

Chapitre IV : Tests et résultats

IV.1 Introduction

Nous avons défini notre approche de la création automatique de résumé multi-documents, ainsi que tous les concepts qui l'accompagnent, et nous sommes maintenant prêts à la mettre à l'épreuve. Ce chapitre décrit l'environnement matériel et logiciel dans lequel nous avons travaillé, ainsi que le jeu de test sur lequel nous avons travaillé et les métriques que nous avons utilisées. Enfin, nous terminerons ce chapitre par un résumé des résultats des tests.

IV.2 Environnement matériel

Notre application va être réalisée sur une machine qui comporte les caractéristiques suivantes :

- **Marque :** Dell Latitude E7440
- **Processeur :** Intel(R) Core (TM) i5-6300U CPU @ 2.40GHz 2.50 GHz
- **Carte graphique :** Intégrée - Intel HD Graphics 520
- **Mémoire :** 8,00 Go
- **System d'exploitation :** Windows 10 Professionnel, 64 bits

IV.3 Environnement logiciel

Python 

Python est un langage de programmation interprété multi-paradigme. Il favorise la programmation impérative structurée, et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions, il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Ce langage de programmation présente de nombreuses caractéristiques intéressantes :

- Il est multiplateforme. C'est-à-dire qu'il fonctionne sur de nombreux systèmes d'exploitation : Windows, Mac OS X, Linux, Android, iOS, depuis les mini-ordinateurs Raspberry Pi jusqu'aux supercalculateurs.

- Il est gratuit. Vous pouvez l'installer sur autant d'ordinateurs que vous voulez (même sur votre téléphone !).
- C'est un langage de haut niveau. Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utilisé.
- C'est un langage interprété. Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à des langages comme le C ou le C++.
- Il est orienté objet. C'est-à-dire qu'il est possible de concevoir en Python des entités qui miment celles du monde réel (une cellule, une protéine, un atome, etc.) avec un certain nombre de règles de fonctionnement et d'interactions.
- Il est relativement simple à prendre en main 2.
- Enfin, il est très utilisé en bioinformatique et plus généralement en analyse de données.
- Toutes ces caractéristiques font que Python est désormais enseigné dans de nombreuses formations, depuis l'enseignement secondaire jusqu'à l'enseignement supérieur. [31]

Google Colab

Google Colaboratory largement connu sous le nom de Google Colab est un service open source fourni par Google à toute personne possédant un compte Gmail. Google Colab fournit un GPU pour la recherche aux personnes qui n'ont pas assez de ressources ou qui ne peuvent pas se le permettre. Le service Google Colab fournit 12,72 Go de RAM et 358,27 Go d'espace disque en une seule exécution. Chaque exécution dure 12 heures, après quoi l'exécution est réinitialisée et l'utilisateur doit établir à nouveau une connexion. Il s'agit de s'assurer que les gens n'utilisent pas le service GPU pour l'extraction de crypto-monnaie et à d'autres fins illégales. Différents cas d'exécution sont présentés ci-dessous.

Une fois que l'utilisateur ouvre un fichier Google Colab, il doit sélectionner un type d'exécution. Il y a 3 options disponibles :

1. None (qui utilisera le processeur de l'ordinateur que l'utilisateur utilise)

2. GPU

3. TPU (en particulier pour le traitement des tenseurs)

La boîte de sélection se trouve dans Exécution -> Modifier le type d'exécution tapez et ressemble à l'image ci-dessous [32] :

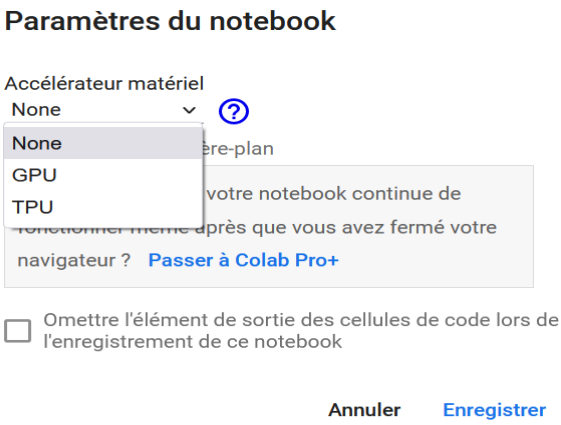


Figure 28 : paramètre du notebook Google Colab

Streamlit

Streamlit est une librairie open source Python créée en 2018. Open Source signifie que le code source est accessible à tous. Cela implique qu'il est réutilisable par tous les utilisateurs pour créer son propre logiciel et permet ainsi une grande flexibilité sur les besoins de chacun. L'open source qui est devenu un véritable mouvement dans le développement logiciel, encourage la production collaboratrice et permet, ainsi, d'améliorer la qualité des logiciels.

Streamlit permet de créer une application en écrivant simplement un code Python. Ainsi, elle en devient un moyen pratique et très accessible pour toute personne ayant des connaissances en python et souhaitant réaliser une web application. C'est un outil facile et rapide pour intégrer de la visualisation de data dans une application. [33]

TensorFlow

TensorFlow est une bibliothèque open source de Machine Learning, créée par Google, permettant de développer et d'exécuter des applications de Machine Learning et de Deep Learning. Découvrez tout ce que vous devez savoir à son sujet.

Transformers

Transformers fournit des architectures à usage général (BERT, GPT-3, Pegasus pour la compréhension du langage naturel (NLU) et la génération de langage naturel (NLG) avec plus de 32 modèles pré-entraînés dans plus de 100 langues et une interopérabilité approfondie entre Jax, PyTorch et TensorFlow. [31]

IO

Le module IO fournit les principales fonctionnalités de Python pour traiter différents types d'E/S. Il existe trois principaux types d'E/S : les E/S textuelles, les E/S binaires et les E/S brutes. Ce sont des catégories génériques et divers magasins de sauvegarde peuvent être utilisés pour chacune d'entre elles. Un objet concret appartenant à l'une de ces catégories est appelé un objet fichier. Les autres termes courants sont flux et objet de type fichier.

NumPy

NumPy est un package utiliser pour les calculs scientifiques en Python. Il est idéal pour les opérations liées à l'algèbre linéaire, aux transformations de Fourier, ou au crunching de nombres aléatoires. Il peut être utilisé en guise de container multi-dimensionnel de données génériques. De plus, il s'intègre facilement avec de nombreuses bases de données différentes.

Nltk

Natural Language Toolkit (NLTK) est une bibliothèque logicielle en Python permettant un traitement automatique des langues, développée par Steven Bird et Edward Loper du département d'informatique de l'université de Pennsylvanie. En plus de la bibliothèque, NLTK fournit des démonstrations graphiques, des données-échantillon, des tutoriels, ainsi que la documentation de l'interface de programmation (API).

NetworkX

NetworkX est une bibliothèque Python pour l'étude des graphes et des réseaux. NetworkX est un logiciel libre distribué sous la nouvelle licence BSD.

Matplotlib.pyplot matplotlib

Pyplot est un module Matplotlib proposant plusieurs fonctions simples pour ajouter des éléments tels que des lignes, des images ou des textes aux axes d'un graphique. Son interface est très confortable, et c'est pourquoi ce module est très utilisé. Matplotlib est distribuée librement et gratuitement sous une licence de style BSD.

IV.4 Ensemble de donnée (Dataset)

Dans ce qui suit, nous présentons nos cinq jeux de données multi-documents de notre dataset (Gigaword, Aeslc, Multi-News, CNN/DailyMail, DUC):

IV.4.1 Aeslc :

Créé par Zhang et Tetreault, AESLC [34] se compose de 18 000 corps d'e-mails et de leurs sujets issus du corpus Enron, C'est une collection de messages électroniques d'employés d'Enron Corporation.

Les sujets des e-mails sont généralement beaucoup plus courts que les résumés générés à partir d'autres ensembles de données. Bien que nos ensembles de données soient principalement liés à l'actualité, nous étions intéressés de voir comment cela fonctionnerait sur notre modèle, si le modèle apprenait plus d'abstraction.

Nous avons utilisé le jeu de données fourni par TensorFlow avec deux fonctionnalités :

- Email_body : texte du corps de l'email.
- Subject_line : texte de l'objet de l'e-mail.

IV.4.2 Gigaword:

Génération de titres sur un corpus de paires d'articles de Gigaword composé d'environ 4 millions d'articles. Utilisez le « org_data » fourni. La tâche consiste à générer le titre à partir de la première phrase. [35]

Il existe deux fonctionnalités :

- Document : article.
- Résumé : Titre.

IV.4.3 Multi-News:

Multi-News, se compose d'articles de presse et de résumés écrits par des humains de ces articles du site newser.com. Chaque résumé est rédigé de manière professionnelle par des éditeurs et comprend des liens vers les articles originaux cités.

Il existe plusieurs fonctionnalités [36] :

- Document : texte des articles de presse séparés par un jeton spécial "|||||".
- Récapitulatif : récapitulatif des factures.
- Résumé : résumé de l'actualité.

IV.4.4 CNN/DailyMail :

Ils représentent deux ensembles de données créés par Hermann et al contenant plus de 300 000 articles au total (93 000 pour CNN et 220 000 du journal Dailymail). Cet ensemble de données associe chaque article à un court ensemble de points résumés qui représentent les faits saillants significatifs de l'article.

Il existe deux fonctionnalités :

- Article : texte de l'article d'actualité, utilisé comme document à résumer.
- Faits saillants : texte joint des faits saillants avec et autour de chaque fait saillant, qui est le résumé cible.

IV.4.5 DUC Document Understanding Conferences 2004

Le Document Understanding Conference (DUC) est organisé chaque année depuis 2001 et constitue le principal forum comparant les systèmes de résumé sur un ensemble de tests partagé. Quatre principaux groupes de tâches ont été traités et évalués manuellement pour la couverture (chevauchement entre le résumé produit par le système et un modèle humain unique) : résumé générique d'un seul document, résumé générique multi-documents, génération de titres et résumé ciblé par question ou sujet. [37]

Le DUC est devenu un événement annuel auquel chaque organisation intéressée par le résumé de textes peut participer à une série d'expériences coordonnées. Les résultats sont présentés dans l'atelier annuel où une comparaison est faite pour préserver le caractère compétitif de la conférence. Le personnel du NIST a élaboré

une série de lignes directrices qui protègent la valeur de la conférence ainsi que les participants.

DUC 2004 utilisera des documents des collections TDT et TREC et incorporera des focus de différentes sortes pour réduire la variabilité et mieux modéliser les tâches réelles. Il examinera la création automatique de résumés courts et très courts. Ce qui suit est une brève description des données et des tâches - une version plus détaillée de ce qui a été développé lors de l'atelier DUC 2002. [38]

Les lois strictes de la conférence et le système d'évaluation ont donné aux données la crédibilité d'être utilisées dans les systèmes de résumés. Nous avons utilisé le DUC 2004 pour tester nos modèles et comparer les résultats, il contenait 30 groupes, chaque groupe représentait un sujet et avait 10 documents, et chaque groupe avait une moyenne de 3 à 4 résumés de référence.

IV.5 Les mesures d'évaluation

IV.5.1 ROUGE

Recall-Oriented Understudy for Gisting Evaluation ou ROUGE est une mesure d'évaluation utilisée essentiellement pour évaluer les résumés. Il fonctionne en comparant les résumés générés à un ou à un ensemble d'autres résumés généralement écrits par des humains. En général, il compare les mots qui se chevauchent dans les n-grammes entre deux textes; cependant, il ne suffit pas de comparer les mots qui se chevauchent. C'est pourquoi nous calculons également la précision et la mesure F. C'était notre choix pour évaluer les résumés générés par notre système.

- ✓ ***Rappel :*** Le rappel mesure la capacité du système à retrouver tous les documents pertinents répondant à une requête. Autrement dit, il mesure la proportion de documents pertinents restitués par le système relativement à l'ensemble des documents pertinents contenus dans la collection. Il est exprimé par [39]:

$$Rappel = \frac{\boldsymbol{documents\ pertinents\ retrouvés}}{\boldsymbol{documents\ pertinents\ dans\ la\ collection}}$$

- ✓ ***Précision:*** La précision mesure la capacité du système à rejeter tous les documents non pertinents à une requête. Autrement dit, elle mesure la proportion de

documents pertinents restitués relativement à l'ensemble des documents restitués par le système. [39]

Elle mesure la capacité du système à trouver exclusivement des documents pertinents et donc à éliminer le bruit. Elle est exprimée par :

$$Précision = \frac{documents\ pertinents\ retrouvés}{documents\ retrouvés}$$

✓ ***F-Mesure :*** Mesure qui combine le rappel et la précision. En effet, le rappel et la précision ont tendance à varier en sens inverse. Elle est définie comme suit :

$$F\text{-}Mesure = \frac{(1+\beta 2)\ Rappel * Précision}{\beta 2\ Rappel + Précision}$$

Où Fβ est le score F. Si β est supérieur à un, le rappel est favorisé. S'il est inférieur à un, la précision est favorisée.

✓ ***ROUGE-N :*** C'est un rappel de n-grammes entre le résumé candidat et un ensemble de résumés de référence.

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

✓ ***ROUGE-L:*** Pour pallier la faiblesse du ROUGE-N, ROUGE-L utilise le concept des sous-séquences communes les plus longues (LCS). Le motif est le suivant : plus le LCS entre deux phrases sommaires est long, plus elles sont similaires. Pour appliquer LCS, les phrases du résumé représentent une séquence de mots. Le problème avec ROUGE-L est qu'il ne calcule que la séquence principale ; par conséquent, d'autres LCS alternatifs ou plus courts ne seront pas pris en compte dans le calcul du score.

$$R_{LCS} = \frac{\sum_{i=1}^{u} LCS_U(s_i, C)}{m}, P_{LCS} = \frac{\sum_{i=1}^{u} LCS_U(s_i, C)}{n}$$

IV.6 Expérimentations

IV.6.1 Mono-Document

Les résultats de finetuning sont présentés dans le tableau 1 qui montre les scores ROUGE sur chaque ensemble de données.

Résultats avec les modèles de PEGASUS :

Le tableau 1 résume les résultats obtenus avec le modèle PEGASUS une fois fine-tuné avec les 4 différents datasets pour le résumé mono-document. Toutes ces mesures sont exprimées en termes de F-Score.

	ROUGE 1	ROUGE 2	ROUGE L	Taille moyenne des résumés obtenus
Gigaword	40.42	17.62	36.67	52 words
AESLC	36.44	15.66	33.42	63 words
Multi-News	39.53	17.28	36.68	148 words
CNN/DailyMAIL	**42.13**	**21.22**	**40.82**	**86 words**

Tableau 1 : Résultats rouge pour les résumés mono-document obtenus avec PEGASUS.

Ces résultats montrent que ***CNN/DailyMail*** obtient le score le plus élevé, tandis que le score le plus élevé entre nos modèles est attribué à celui affiné avec Multi-News avec un petit écart de 2.60/3.94/4.14 uniquement pour ROUGE1, ROUGE2 et ROUGEL respectivement.

Résultats avec les modèles de BART :

Le tableau 2 résume les résultats obtenus avec le modèle BART une fois fine-tuné avec les 4 différents datasets pour le résumé mono-document. Toutes ces mesures sont exprimées en termes de F-Score

	ROUGE 1	ROUGE 2	ROUGE L	Taille moyenne des résumés obtenus

Gigaword	21.23	8.13	18.25	61words
AESLC	12.33	5.51	11.45	57words
Multi-News	41.96	15.35	23.40	140words
CNN/DailyMAIL	**43.30**	**20.5**	**30.29**	**78words**

Tableau 2 : Résultats rouge pour les résumés mono-document obtenus avec BART

Ces résultats montrent encore une fois que ***CNN/DailyMail*** obtient le score le plus élevé. Les autres modèles obtiennent de moins bons scores. On constate également que les scores obtenus par BART sont moins bons que ceux obtenus avec Pegasus. Cela va de soi, la technique de pré-entrainement de PEGASUS permet au modèle une meilleure compréhension et reconnaissances du vocabulaire d'une langue et de ce fait, une meilleure génération de résumés.

Comme on peut le voir, les meilleurs scores sont attribués au modèle ***CNN/DailyMail*** avec Bart et Pegasus.

IV.6.2 Multi-Documents

Pour évaluer les modèles du résumé multi-documents, nous menons une série d'expérimentations en utilisant le jeu de données DUC 2004.

Vu que les résultats sont nettement bas comparativement au résumé mono-document, nous avons détaillé l'ensemble des mesures ROUGE pour mieux discuter nos résultats. Tous les modèles présentés exploitent la technique TextRank pour éliminer les phrases redondantes et non pertinentes des documents.

- **Résultats avec les modèles de BART :**

Ces scores représentent respectivement le rappel, la précision et la F-mesure. Les meilleurs résultats ROUGE sont en gras.

	ROUGE 1			ROUGE 2			ROUGE L		
	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
DistilBart-CG	1.29	25	2.46	0	0	0	1.29	25	2.46
DistilBart-CA	1.29	14.92	13.8	0	0	0	11.68	13.43	12.24
DistilBart-MC	**7.79**	**28.57**	**12.24**	**0**	**0**	**0**	**7.79**	**28.57**	**12.49**
DistilBart-CNN	5.19	13.33	7.47	0	0	0	5.19	13.33	7.47

Tableau 3 : Résultats de nos modèles BART sur DUC 2004.

Notre modèle ***DistilBart-MC*** surpasse le modèle ***DistilBart-CA*** avec des valeurs plus élevées de 13,63/0/15.14 compte tenu la valeur de précision en ROUGE 1/ROUGE 2/ROUGE L respectivement.

Les pires résultats concernent le ***DistilBart-CG***, parfois le décodeur prédit des résumés avec les bons sujets et parfois non. Les résumés n'étaient pas bien formés syntaxiquement et démontraient parfois un manque de compréhension sémantique de l'article d'entrée.

- **Résultats avec les modèles de Pegasus :**

Ces scores représentent respectivement le rappel, la précision et la F-mesure. Les meilleurs résultats ROUGE sont en gras.

	ROUGE 1			ROUGE 2			ROUGE L		
	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
pegasus-gigaword	5.19	25	8.6	0	0	0	5.19	25	8.6
pegasus-aeslc	0	0	0	0	0	0	0	0	0
pegasus-multi_news	**19.48**	**13.27**	**15.78**	**0**	**0**	**0**	**16.88**	**17.78**	**13.68**
pegasus-cnn/dailymail	14.28	17.18	15.6	0	0	0	14.28	17.18	15.6

Tableau 4 : Résultats de nos modèles Pegasus sur DUC 2004.

Notre modèle ***pegasus-multi_news*** surpasse le modèle ***pegasus-cnn/dailymail*** avec des valeurs plus élevées de 5.2/0/0.6 compte tenu la valeur de rappel en ROUGE 1/ROUGE 2/ROUGE L respectivement.

Les pires résultats concernent le ***pegasus-aeslc***, parfois le décodeur prédit des résumés avec les bons sujets et parfois non. Les résumés n'étaient pas bien formés syntaxiquement et démontraient parfois un manque de compréhension sémantique de l'article d'entrée.

Comme on peut le voir, les meilleurs scores sont attribués aux modèles DistilBart-MC et pegasus-multi_news.

IV.7 Discussion

Globalement, on constate que les modèles obtiennent d'assez bons résultats dans le contexte mono-document, cela s'explique par les corpus utilisés. Vu que les corpus sont de taille réduite et sont focalisés sur le résumé abstractif pour la génération de headline (titres) ou bien d'objet de message, il était évident que les résultats obtenus sur ces mêmes datasets ont de bons résultats vu la taille réduite que doit gérer le modèle.

Lors du passage à l'échelle, il est difficile pour nos modèles d'assimiler un gros volume d'information surtout qu'on est limité en terme de mémoire. De plus, la nature du dataset utilisé sur le résumé multi document diffère de celle du mono-document. Là, il est question d'un vrai résumé abstractif.

La qualité des résultats obtenus dans le contexte multi-documents peut s'expliquer aussi par la nature de la mesure d'évaluation utilisée. Lorsqu'on résume un texte de façon abstractive, parfois, on a tendance à utiliser des synonymes ou des reformulations, or celles-ci ne sont pas considérées par les mesures d'évaluation. On a plus de chance de retrouver des titres de textes similaires ou que des résumés similaires.

Néanmoins, ce travail nous a ouvert une voie pour le résumé multi-documents, surtout que nos modèles étaient initialement limités par la taille du document en entrée, mais on a pu contourner cette contrainte via l'utilisation de l'algorithme TextRank.

IV.8 Conclusion

A travers ce chapitre, nous avons décrit le côté technique de notre projet, y compris ses informations logicielles et matérielles, ainsi que les différents résultats obtenus pour nos jeux de données **«Gigaword, Aeslc, Multi-News, CNN/DailyMail et DUC»**.

Nous avons pu comparer les résultats des approches utilisées et avons constaté qu'ils peuvent déboucher sur d'autres projets de recherche.

Conclusion générale

Face à l'arrivée d'Internet et des moteurs de recherche, la masse de données et la quantité de textes disponibles en format Électronique s'accumule rapidement et en très grandes quantités. Les informations peuvent provenir de n'importe quelle source d'ou la difficulté de trouver des informations pertinentes. Il est donc indispensable d'offrir des outils de visualisation rapide des textes, en particulier des résumés automatiques, afin que l'utilisateur puisse produire un résumé concis et fluide sans aucune aide humaine tout en préservant le sens du document du texte original.

Notre objectif initial durant cette thèse était de proposer une approche de résumé automatique abstractif en exploitant différentes variantes d'architectures neuronales à base de Transformers capable de traiter une entrée dépassant 1024 mots.

Dans ce travail, on a commencé par citer quelque définitions on a parlé sur l'apprentissage automatique. Nous avons fait une petite présentation afin de comprendre ce domaine d'où on a présenté quelque notions du Résumé Automatique (types, étapes, domaines …).

Ensuite nous avons discuté sur la création des résumés automatiques multi-documents qui se basent sur les réseaux de neurones, on a abordé tous les étapes nécessaires pour construire un bon modèle de résumé. Nous avons proposé des méthodes qui ont été utilisées lors du prétraitement dans la génération de résumés multi-documents telle que BART et Pegasus. Une fois que notre approche a été bien définie nous somme prêt à la mettre à l'épreuve. Nous avons présenté les jeux de données de notre dataset (Gigaword, Aeslc, Multi-News, CNN/DailyMail, DUC) sur lequel nous avons travaillé. Et à la fin nous avons terminé ce travail par un résumé des résultats et des tests obtenus par nos jeux de données.

Deux travaux ont déjà été entamés en 2017 et en 2019 par nos camarades, nous avons fait de notre mieux pour être à la hauteur de vos attentes et nous espérons de tout cœur que ce travail honnête apportera un petit changement dans un avenir proche.

Bibliographie

[1] H. A. Baraka, « Étude sur les Méthodes d'Optimisation Utilisée dans l'Apprentissage Automatique », Thesis, M. RIMOUCHE Ali, 2020. Disponible sur: http://thesis.essa-tlemcen.dz:8080/xmlui/handle/STDB_UNAM/173

[2] F. E. zohra Kobbi et T. yasmine Leulmi, « Résumé vidéo multi vues », Thesis, Université Blida 1, 2021.
Disponible sur: http://di.univ-blida.dz:8080/jspui/handle/123456789/14503

[3] R. Mifdal, « Application des techniques d'apprentissage automatique pour la prédiction de la tendance des titres financiers », masters, École de technologie supérieure, Montréal, 2019. Disponible sur: https://espace.etsmtl.ca/id/eprint/2422/

[4] G. Zaccone, M. R. Karim, et A. Menshawy, *Deep Learning with TensorFlow*. Packt Publishing, 2017

[5] A. Bardet, « Architectures neuronales multilingues pour le traitement automatique des langues naturelles », phdthesis, Le Mans Université, 2021.
Disponible sur: https://tel.archives-ouvertes.fr/tel-03199494

[6] Aurélien Géron, Hands on Machine Learning Deep Learning et Sckitlearn, 13 mars 2017

[7] Siméon Kostadinov, Guide de démarrage rapide des réseaux de neurones récurrents avec Pythonÿ: Apprentissage séquentiel et modélisation du langage avec TensorFlow Simeon Kostadinov, 29ÿnovembreÿ2018.

[8] A. Aries, « Résumé automatique de textes », 2013. doi: 10.13140/RG.2.1.4436.5528.

[9] F. Boudin et J.-M. Torres-Moreno, « Résumé automatique multi-document et indépendance de la langue : une première évaluation en français », p. 10.

[10] M. H. Maaloul, « Approche hybride pour le résumé automatique de textes. Application à la langue arabe. », p. 194.

[11] G. Crispino et J. Couto, « Construction de résumés automatiques: une approche dynamique ».

[12] F. S. Douzidia, « Résumé automatique de texte arabe », 2005.

[13] M. Mnasri, « Résumé automatique multi-document dynamique », PhD Thesis, Université Paris-Saclay (ComUE), 2018.

[14] D. Bernhard *et al.* « Problèmes de tokénisation pour deux langues régionales de France, l'alsacien et le picard », in *DiLiTAL 2017*, 2017, p. 14-23.

[15] J. C. Reynar et A. Ratnaparkhi, « A maximum entropy approach to identifying sentence boundaries », *arXiv preprint cmp-lg/9704002*, 1997.

[16] M. Hassel, « Resource lean and portable automatic text summarization », PhD Thesis, KTH, 2007.

[17] A. P. Widyassari, S. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, et A. Affandy, « Review of automatic text summarization techniques & methods », *Journal of King Saud University-Computer and Information Sciences*, 2020.

[18] A. Nenkova et K. McKeown, *Automatic summarization*. Now Publishers Inc, 2011.

[19] Y. LeCun, « L'apprentissage profond, une révolution en intelligence artificielle », *La lettre du Collège de France*, n° 41, p. 13, 2016.

[20] C. Orasan et S. St, « Pronominal anaphora resolution for text summarisation », *of: Proceedings of the Recent Advances in Natural Language Processing*, p. 430-436, 2007.

[21] V. Hugo et R. Journet, *Les Misérables. Tome II*. Classiques Garnier Numérique Paris, 2014.

[22] H. P. Edmundson, « New methods in automatic extracting », *Journal of the ACM (JACM)*, vol. 16, n° 2, p. 264-285, 1969.

[23] I. Mani, « Summarization evaluation: An overview », 2001.

[24] W. Bensidiaissa et R. Bouchetara, « Generative models for automatic multi-document summarization », Thesis, Université Blida 1, 2020. Disponible sur: http://di.univ-blida.dz:8080/jspui/handle/123456789/9420

[25] J. Steinberger, « Evaluation measures for text summarization », *Computing and Informatics*, vol. 28, n° 2, p. 251-275, 2009.

[26] E. H. Hovy, C.-Y. Lin, L. Zhou, et J. Fukumoto, « Automated Summarization Evaluation with Basic Elements. », in *LREC*, 2006, vol. 6, p. 604-611.

[27] A. Hocine, « LES RESUMES ET LES RESUMES AUTOMATIQUES ».

[28] http://rali.iro.umontreal.ca/rali/?q=fr/Resume%20automatique

[29] V. Nyzam, C. Rodrigues, et A. Bossard, « MOTS: un outil modulaire pour le résumé automatique (MOTS: A Modular Framework for Automatic Summarization) », in *Actes de la Conférence TALN. Volume 1-Articles longs, articles courts de TALN*, 2018

[30] T.-H. Le, « Neural Methods for Sentiment Analysis and Text Summarization », p. 116.

[31] P. F. Fuchs et P. Poulain, « Introduction à la programmation Python pour la biologie », PhD Thesis, Université de Paris, 2020.

[32] P. Kanani et M. Padole, « Deep learning to detect skin cancer using google colab »,

International Journal of Engineering and Advanced Technology Regular Issue, vol. 8, nº 6, p. 2176-2183, 2019.

[33] « Streamlit ou l'outil pour présenter votre travail de Machine Learning », *Formation Data Science | DataScientest.com*, 5 avril 2022. https://datascientest.com/streamlit-ou-loutil-pour-presenter-votre-travail-de-machine-learning.

[34] R. Zhang et J. Tetreault, « This Email Could Save Your Life: Introducing the Task of Email Subject Line Generation », arXiv, arXiv:1906.03497, juin 2019. doi: 10.48550/arXiv.1906.03497.

[35] « gigamot | TensorFlow Datasets ». https://www.tensorflow.org/datasets/catalog/gigaword?hl=fr

[36] « gigamot | TensorFlow Datasets ». https://www.tensorflow.org/datasets/catalog/gigaword?hl=fr

[37] A. Nenkova, « Automatic Text Summarization of Newswire: Lessons Learned from the Document Understanding Conference. », janv. 2005, p. 1436-1441.

[38] « DUC 2003 Documents for Summarization, Tasks, and Measures ». https://duc.nist.gov/duc2003/tasks.html

[39] S. Chaudiron, « L'évaluation des systèmes de recherche d'information ». Hermès, 2004.