

الجمهورية الجزائرية الديمقراطية الشعبية  
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
وزارة التعليم العالي و البحث العلمي  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
جامعة عمّار تليجي بالأغواط  
UNIVERSITE AMAR TELIDJI LAGHOUAT



كلية العلوم

Faculté des Sciences

Département : Informatique

Spécialisation : Système d'Information et de Décision

Mémoire de Master

Par: Mohamed Bouzidi

Thème

**Développement d'un système de synthèse d'information  
proactive pour une bibliothèque universitaire**

Les Membres du jury:

Mrs. Zohra ABDELHAFIDI	MCA	Présidente
Mrs. Fatna GUIBADJ	MAA	Examinatrice
Mr. Lakhdar KACHNA	MAA	Examineur
Mrs. Amel BELABACI	MCB	Promotrice

Juin 2025

## *Dédicace*

*A mes chers parents Oum Habiba et Kaddour et à mes deux tantes Ouahiba et Dalila mon frère Ilias et  
ma sœur Asma ainsi que mon oncle Djamel,*

*Pour leurs amours inconditionnels, leurs soutiens indéfectibles et leurs encouragements sincères pleins de  
bonne volonté tout au long dans ma vie*

*A toutes les personnes proches : Hanine, Sarah, Samah, Jaloul, Ali, Acyle, Fazyl, Lina,  
Amine, Ayman, Taher,*

*Pour avoir cru en moi, m'inspirer, soutenu et accompagné dans les moments faciles et difficiles*

*Ce mémoire vous est dédié, avec toute ma gratitude*

# Remerciements

Merci à ma promotrice Dr Amel Belabaci pour avoir cru en moi et m'avoir guidé à travers des conseils avisés qui m'ont aidé à réaliser ce travail.

Je tiens à remercier très promptement madame Dr Abdelhafidi Zohra ,qui m'a fait l'honneur de présider le jury de mémoire.

Je tiens à exprimer ma profonde gratitude à madame Dr Guibadj Fatna et monsieur Dr Kachna Lakhdar pour avoir accepté d'examiner ce mémoire.

Aussi je veux exprimer ma profonde gratitude à ma famille. Mes parents Oum Habiba et Kaddour , mes tantes Dalila et Ouahiba et mon frère et ma sœur Ilias et Asma ainsi que mon oncle Djamel pour leurs amours indéfectibles, leurs soutiens constants et à leurs sacrifices pour mon bien être, sans eux rien de ce que j'ai accompli ne serait possible

Je remercie également mes proches tous sont exception Hanine, Sarah ,Samah ,Jaloul ,Ali , Acyle ,Fazyl ,Lina , Amine ,Ayman ,Taher et mes amis dévoués , je vous adresse toutes ma reconnaissance pour vos paroles d'encouragement , les moments passés ensemble ,et votre bienveillance.

Ce mémoire est autant le fruit de vos prières, de votre foi en moi et mes efforts.

## Résumé :

Dans un contexte d'abondance d'informations scientifiques les chercheurs et étudiants se perdent facilement ce qui complique l'accès rapide et pertinent aux ressources, ce mémoire propose le développement d'un système intelligent de synthèses et de recommandation proactive destiné aux bibliothèques universitaires. Ce système vise à automatiser la génération de synthèses à partir de documents scientifiques en utilisant des techniques de traitement automatique de langage naturel (NLP) et des grands modèles de langage (LLM). Il repose également sur un moteur de recommandation proactive qui s'adapte aux profils utilisateurs, en tenant compte de leur historique de consultation et de leurs diverses interactions avec l'application. Le but est de fournir un accès simple, rapide et synthétique à l'information, tout en améliorant l'efficacité des recherches de documents, en réduisant les efforts de recherche et en proposant des contenus ciblés et pertinents.

Mots clé : système proactive, synthèses automatique, recherche scientifique, NLP, LLM, document scientifique, bibliothèque en ligne

## Abstract :

In a context of overwhelming scientific information, researchers and students can easily become lost, making rapid and relevant access to resources difficult. This thesis proposes the development of an intelligent system for proactive synthesis and recommendation tailored to university libraries. The system aims to automate the generation of summaries from scientific documents using natural language processing (NLP) techniques and large language models (LLM). It also includes a proactive recommendation engine that adapts to users' profiles by considering their consultation history and various interactions with the application. The objective is to provide simplified, fast and synthesized access to information while enhancing the efficiency of document searches, by reducing the research effort and offering more targeted and relevant content, the system helps improve user's academic experience.

keywords : proactive system, automatic summarization, scientific research, NLP, LLM, scientific document, online library

## ملخص:

في عصر وفرة المعلومة، يجد الباحثون والطلبة صعوبة في الوصول السريع والدقيق إلى الموارد مما يزيد من التعقيد عملية البحث. يهدف هذا العمل إلى تطوير نظام ذكي للتلخيص والتوصية الاستباقية موجه للمكتبات الجامعية. يعتمد النظام على تقنية NLP و LLM من أجل توليد ملخصات تلقائية من الوثائق العلمية وتوصيات آخذة بعين الاعتبار سجل التصفح وتفاعلات المختلفة مع التطبيق.

كلمات مفتاحية: نظام استباقي, تلخيص تلقائي, البحث العلمي, LLM, NLP, وثائق علمية, مكتبة رقمية

Liste des figures :.....	9
--------------------------	---

<b>Chapitre 1 Présentation des systèmes proactive .....</b>	<b>10</b>
---	-----------

<b>1 Généralité .....</b>	<b>10</b>
<i>1.1 Recommandations proactives et prédictives .....</i>	<i>10</i>
<i>1.2 La nécessité du développement d'un système proactive .....</i>	<i>14</i>
<i>1.3 objectifs et but du travail: .....</i>	<i>14</i>
<i>1.4 Synthèse automatique de texte .....</i>	<i>15</i>
<i>1.5 Les algorithmes fondés sur le classement par graphe : .....</i>	<i>16</i>
<b>1.5.1 PageRank .....</b>	<b>17</b>
<b>1.5.2 LexRank.....</b>	<b>19</b>
<b>1.5.3 PageRank .....</b>	<b>21</b>
<i>1.6 Comparaison entre TextRank, LexRank et PageRank .....</i>	<i>23</i>
<b>1.6.1 Principe de base: .....</b>	<b>26</b>
<b>1.6.2Qualité du synthèse .....</b>	<b>27</b>
<i>1.7 Les graphes générés .....</i>	<i>28</i>
<i>1.8 Conclusion:.....</i>	<i><b>Erreur ! Signet non défini.</b></i>

<b>Chapitre 2 Traitement du langage naturel (NLP) et grands modèles de langage (LLM) .....</b>	<b>34</b>
--	-----------

<b>2 Généralités .....</b>	<b>Erreur ! Signet non défini.</b>
<i>2.1 Définition.....</i>	<i>34</i>
<i>2.2 Prétraitement et représentation vectorielle .....</i>	<i>34</i>
<i>2.3 Architecture transformer et mécanisme d'attention .....</i>	<i>35</i>
<i>2.4 Entraînement à grande échelle et capacités des LLM.....</i>	<i>36</i>
<i>2.5 Principaux domaines d'application .....</i>	<i>36</i>
2.5.1 La traduction automatique: .....	36
2.5.2 Classification de texte et NLU:.....	37
2.5.3 Synthèse de texte et synthèse de documents: .....	37
<i>2.6 Exemples de modèles et compassion de LLM.....</i>	<i>37</i>
<i>2.7 Conclusion .....</i>	<i>38</i>

<b>Chapitre 3 Conception .....</b>	<b>39</b>
------------------------------------	-----------

<b>3 Diagramme: .....</b>	<b>39</b>
<i>3.1 Diagramme cas d'utilisation: .....</i>	<i>39</i>
<i>3.2 Diagramme de classe .....</i>	<i>39</i>
<i>3.3 Diagramme de sequence:.....</i>	<i>41</i>
<i>3.4 Les scénarios:.....</i>	<i>42</i>

<b>Chapitre 4 Implementation et présentation de l' application.....</b>	<b>44</b>
---	-----------

<b>4 contenus de l'application .....</b>	<b>Erreur ! Signet non défini.</b>
--	------------------------------------

<u>4.1</u>	<u>Outils utilisés.....</u>	<u>44</u>
4.1.1	django	44
4.1.2	Elasticsearch .....	45
4.1.3	API Gemini .....	45
4.1.4	key Bert.....	46
4.1.5	NLTK	47
4.1.6	Spacy	47
<u>4.2</u>	<u>Base de données utilisées.....</u>	<u>48</u>
<u>4.3</u>	<u>Interface homme-machine .....</u>	<u>48</u>
<u>4.4</u>	<u>Etude de cas.....</u>	<u>52</u>
<u>4.5</u>	<u>Analyses.....</u>	<u>53</u>
<u>4.6</u>	<u>Conclusion .....</u>	<u>53</u>
<u>4.7</u>	<u>Conclusion générale.....</u>	<u><b>Erreur ! Signet non défini.</b></u>
	<b>Webographies .....</b>	<b>56</b>

# Liste des figures :

Figure 1 algorithme PagerRank.....	18
Figure 2 algorithme LexRank.....	20
Figure 3 algorithme TextRank.....	22
Figure 4 graphe pagerank .....	30
Figure 5 graphe LexRank .....	31
Figure 6 graphe TextRank .....	32
Figure 7 usecase .....	39
Figure 8 diagramme de classe .....	40
Figure 9 diagramme séquence .....	41
Figure 10 interface principal.....	48
Figure 11 found papers .....	49
Figure 12 historique.....	50
Figure 13 chat contraire sythese ,mots clé,telechargement,recomendation.....	51

# Chapitre 1 Présentation des systèmes proactive

## 1 Introduction :

### 1.1 Recommandations proactives et prédictives :

Les premières expérimentations de systèmes de recommandation datent du début des années 1990. En 1992, Goldberg et al. (PARC Xerox) développèrent Tapestry, l'un des premiers algorithmes recommandant du contenu (filtres collaboratif et basé sur le contenu) pour filtrer les courriers d'un utilisateur selon ses préférences [1]. Quelques années plus tard, le projet GroupLens (Resnick et al., 1994, MIT) adapta ces idées aux forums en ligne (newsgroups) et introduit un système de notation distribué, considéré comme le premier réel système de recommandation moderne [2].

Avec l'explosion du web dans les années 2000, ces techniques ont été progressivement industrialisées. Par exemple, Amazon a mis en production dès 2003 un algorithme de filtrage collaboratif « item-à-item » qui compare l'inventaire de produits aux historiques d'achat et de navigation des clients [cs.umd.edu](http://cs.umd.edu/cs.umd.edu) [3]. Ce type de recommandation personnalisée est devenu un élément clé de la stratégie commerciale d'Amazon pour suggérer des livres (et produits) adaptés à chaque profil d'utilisateur [rickwash.com](http://rickwash.com) [4]. De même, d'autres domaines comme la recommandation de films ont connu des percées majeures : le Netflix Prize (2006-2009) a mobilisé plusieurs milliers de chercheurs pour améliorer les prévisions de notes (« prédiction de préférences ») via des méthodes de factorisation matricielle et d'ensembles de modèles [5].

Au fil des années, les systèmes se sont enrichis de nouvelles approches. Le filtrage collaboratif hybride et deep-learning sont apparus dans les années 2010 pour mieux modéliser les interactions, parallèlement et en compte du contexte utilisateur (heure, lieu, appareil, humeur...) a donné naissance aux systèmes de recommandation contextuels. Des études ont ainsi montré que le comportement et la satisfaction d'un utilisateur dépendent fortement de son contexte (contribution de Riboni & Bettini 2011, Baltrunas et al. 2012), d'où l'émergence des CARS (Context-Aware Recommender Systems) [6]. Cette technique permettra, par exemple, de recommander des livres pour un jeune homme de 20 ans s'il est chez un ami ou seul chez lui, en adaptant automatiquement les recommandations au contexte d'écoute ou de lecture [7].

Maintenant que nous avons vu comment les systèmes proactifs et prédictifs ont évolué avec le temps,

il est important de comprendre pourquoi ils sont devenus si important aujourd'hui. Leurs développements ne servent pas seulement à rendre les technologies plus intelligentes, mais aussi à améliorer l'expérience des utilisateurs. Dans la partie suivante, nous allons voir comment ces systèmes rendent plus pratiques, plus personnalisés et plus satisfaisants pour les personnes qui les utilisent au quotidien. Comme le notent les études marketing, « mettre en avant des articles susceptibles d'intéresser les utilisateurs leur permet de gagner en temps et en effort. Ce faisant, l'internaute est tenté de revenir sur le même site » [8]. En pratique, lorsqu'un utilisateur subit une surcharge d'informations, il a tendance à accepter plus volontiers l'aide d'un système de recommandation : par exemple, Aljukhadar et al. (2012) ont montré que, en situation de surcharge informationnelle, les participants cherchaient davantage à utiliser les recommandations, et la qualité de leur choix s'améliore dans des scénarios complexes lorsque des suggestions personnalisées leur étaient fournies [9].

De plus, au-delà de la simple pertinence, la sérendipité – c'est-à-dire la capacité à proposer des éléments nouveaux mais intéressants – s'avère cruciale pour la satisfaction. Une étude a mesuré qu'un taux de sérendipité élevé dans les recommandations accroît significativement la satisfaction et l'intention d'achat des utilisateurs, plus encore que la nouveauté ou la fraîcheur des suggestions [10]. Enfin, l'aspect humain de la recommandation joue aussi la transparence et la clarté des explications fournies renforcent la confiance. Par exemple, un dispositif fournissant une information minimaliste mais visible sur la provenance de la recommandation (« signal de transparence ») a notablement augmenté le niveau de confiance des utilisateurs [11]. En synthèse, les systèmes proactifs bien conçus (pertinents, surprenants et explicables) élèvent l'engagement et la satisfaction perçue des utilisateurs en rendant la découverte de contenu plus efficace et agréable.

En plus d'apporter une meilleure expérience aux utilisateurs, ces systèmes représentent aussi un véritable atout pour les entreprises en comprenant les besoins des clients et en leur proposant des contenus ou des produits adaptés, les systèmes proactifs et prédictifs peuvent augmenter les ventes, fidéliser des clients et améliorer les performances commerciales. C'est ce que nous allons explorer dans la prochaine partie.

Les recommandations personnalisées ont un impact fort sur les indicateurs commerciaux. Elles améliorent les taux de conversion et les ventes en orientant le client vers des produits susceptibles de l'intéresser. Linden et al. soulignent qu'un bon système de recommandation augmente

considérablement les taux de conversion comparé aux méthodes de promotion large (« broad-scale marketing ») [12]. Amazon a à ce titre observé que son algorithme d'item-to-item conduisait à de meilleurs résultats de vente pour chaque utilisateur, même avec peu d'historique, renforçant l'efficacité des croisements de ventes (« frequently bought together ») [13].

Plusieurs études empiriques confirment cet effet. Par exemple, l'analyse d'un grand site marchand a montré que les liens de recommandation entraînaient une hausse de 9 % des ventes des produits recommandés (compensée en partie par une baisse de 1,9 % des ventes du produit initial), soit un gain net d'environ 11 % des ventes globales dues au système de recommandation [14].

De même, 35 % du chiffre d'affaires d'Amazon proviendrait des recommandations selon un rapport McKinsey [15], même si des travaux plus récents modifient ce chiffre. En proposant de façon proactive des articles pertinents, l'utilisateur gagne en efficacité et est plus enclin à revenir. Comme noté plus haut, le fait de faire gagner temps et effort à l'internaute via des suggestions adaptées le pousse à conserver sa visite ou son abonnement (effet de rétention) [16]. Tous ces éléments (hausse de conversion, d'achat moyen, et de fidélité) font des recommandations prédictives un levier marketing majeur dans le e-commerce.

Si les avantages pour les entreprises sont déjà bien visibles aujourd'hui, le domaine continue d'évoluer rapidement. Les systèmes proactifs et prédictifs ouvrent la voie à de nouvelles questions et pistes de recherche, tant sur le plan technique qu'éthique. Dans la partie suivante, nous verrons quelles sont les tendances actuelles et les perspectives émergentes qui orientent les travaux des chercheurs dans ce domaine en pleine expansion.

Les tendances de recherche actuelles portent notamment sur la personnalisation contextuelle, l'applicabilité des recommandations et des approches plus « cognitives ».

- **Recommandation contextuelle.** Les CARS intègrent explicitement des facteurs de contexte (lieu, moment, dispositif, etc.) pour affiner les suggestions. Une typologie récente regroupe ces facteurs pour mieux exploiter les informations environnementales dans le processus de recommandation. Les systèmes contextuels adaptent ainsi en temps réel les recommandations (par exemple, les livres ou vidéos conseillés diffèrent selon que l'utilisateur est dans son salon, dans un café ou en voiture). Les recherches se concentrent sur la collecte de ces données (GPS, capteurs, calendriers) et sur les modèles prédictifs hybridant filtre collaboratif et logique contextuelle (post-filtrage, pré-filtrage, modélisation 3D) [17].

- Explainable AI et confiance utilisateur. Les travaux de Explainable AI appliqués aux recommandations cherchent à fournir des justifications transparentes. Au-delà de l'algorithme, on étudie comment présenter intelligiblement les raisons des suggestions (par exemple « Vous avez aimé X, donc nous suggérons Y », ou via des visualisations) pour augmenter la confiance et la satisfaction. Les recherches récentes ont développé des indicateurs de qualité explicable (clarté, exhaustivité) et évalué leur impact sur l'acceptation des recommandations. On montre par exemple que même un simple signal de transparence (« info simplifiée ») renforce nettement la confiance de l'utilisateur dans le système. Ce domaine converge avec l'analyse de confiance (« trust-aware RS ») et la protection de la vie privée (explique sans tout révéler).

Recommandation proactive et cognitive. La proactivité vise à anticiper les besoins avant qu'ils ne soient explicitement exprimés. Cela inclut l'envoi d'alertes ou de « push notifications » (par ex. annoncer la sortie d'un nouveau roman selon le profil de lecture de l'utilisateur) avec un timing optimal. Les recherches examinent le bon moment pour recommander (élément when-to-recommend), l'exploration/exploitation pour éviter de « spammer » l'utilisateur, et le compromis entre interruption et valeur ajoutée. Par ailleurs, la notion de système de recommandation cognitif émerge : il s'agit d'algorithmes inspirés des modèles cognitifs humains, capables de détecter l'évolution des préférences et d'apprendre en continu de l'utilisateur. Ces systèmes cognitifs comprennent mieux le profil global du client, adaptent dynamiquement les recommandations (persona, personnalités, émotions) et peuvent même anticiper de nouvelles préférences inconnues [18]. L'avènement des grands modèles de langage (LLM) révolutionne le champ. Ces modèles, dotés de vastes connaissances encodées, ouvrent la voie à des assistants de recommandation conversationnelle : ils comprennent le langage naturel de l'utilisateur, génèrent des recommandations en dialogue et produisent des explications textuelles riches [19].

En synthèse, les recherches actuelles visent à rendre les recommandations toujours plus « humaines » : sensibles au contexte, compréhensibles (explicables), proactives et même interactives (agents virtuels intelligents). L'intégration de l'apprentissage profond multimodal, des graphes de connaissances, de l'IA émotionnelle et des algorithmes transparents trace la feuille de route de la recommandation de demain

## **1.2 La nécessité du développement d'un système proactive :**

Un des problèmes majeurs auxquels sont confrontées les bibliothèques universitaires de nos jours dans le domaine de l'enseignement supérieur et de la recherche est la surcharge informationnelle. Étant donné le grand nombre de publications scientifiques, la diversification des domaines de recherche et des formats des documents, l'activité de recherche d'un étudiant ou d'un chercheur devient de plus en plus difficile. L'approche traditionnelle de la recherche de documents scientifiques est une tâche rigoureuse qui prend du temps. Toutefois, pour s'adapter à la surcharge des données, il convient tout de même d'évoluer vers un autre modèle caractérisé par la logique et par la facilitation : une approche proactive où il est permis de réduire la charge, d'économiser le temps mais d'augmenter l'efficacité de la recherche via la prise en considération des préférences, les habitudes de consultation, le niveau d'étude, le domaine de recherche de l'utilisateur. C'est donc un système qui favorise de manière proactive une recommandation contextualisée et personnalisée des documents en fonction du type de l'utilisateur. À la place d'un simple moteur de recherche, l'application est plutôt un assistant intelligent dont le rôle est de filtrer l'information, recueillir ce qui est essentiel et recommandé à l'utilisateur de manière proactive les documents les plus appropriés à ses intérêts et à son niveau de lecture. En se résumant, le projet présenté dans ce mémoire peut alors être considéré comme une application conforme à cette idée : il s'agit de proposer une application intégrant une bibliothèque universitaire capable de générer automatiquement des synthèses des textes scientifiques à partir des documents disponibles et de mettre en œuvre un système de recommandation proactif, en permettant à chaque utilisateur de se voir recommander des articles ou des ressources supplémentaires en fonction de ses intérêts ou de son sujet de recherche. Par conséquent, une telle combinaison synthèse automatique et recommandation personnalisée résume un cadre concret de solutions nouvelles aux problématiques de se retrouver devant un grand nombre de documents et ne pas savoir par où commencer sa recherche.

## **1.3 Objectifs et but du travail:**

- L'automatisation de la génération de synthèses à partir des documents scientifiques de la bibliothèque numérique à l'aide des techniques de traitement automatique du langage naturel (NLP) et Grand modèle de langage (LLM).

- Un système de recommandation proactif basé sur le profil utilisateur, l'historique des services consultés, les sujets de préférence et les domaines de recherche.
- Un service d'accès à l'information simplifié, rapide, direct et synthétique.
- L'efficacité du temps passé en réduisant les efforts de recherche de l'utilisateur et proposant des services plus riches et plus pertinents.

#### **1.4 Synthèse automatique de texte :**

Les algorithmes de synthèse sont des procédés informatiques conçus pour générer automatiquement du contenu, qu'il soit textuel ou sonore. Ils jouent un rôle central dans des domaines tels que le traitement du langage naturel, la musique assistés par ordinateur et l'intelligence artificielle dans ce mémoire nous allons voir les algorithmes de synthèse de texte automatique

La synthèse automatique de texte est une branche du traitement du langage naturel (TALN) qui vise à condenser un document en un synthèse précis tout en préservant les informations essentielles et importantes

Elle joue un rôle crucial dans un contexte où la masse de données textuelles disponibles dépasse largement la capacité de traitement humain. Grâce à la synthèse automatique, il devient possible d'extraire rapidement les points clés d'un texte sans avoir à le lire intégralement, ce qui permet un gain de temps considérable, notamment dans des domaines à forte densité informationnelle comme la recherche académique, le journalisme, les rapports professionnels ou les documents juridiques, il existe plusieurs approches pour générer des synthèses parmi les différentes méthodes nous avons l'approche extractive.

La synthèse extractive est privilégié dans de nombreux cas en raison de sa simplicité, de sa fiabilité et de son efficacité, qui en font une solution particulièrement intéressante dans divers contextes. Ce type de synthèse consiste à extraire directement les phrases du texte les plus pertinentes d'un document source pour créer un synthèse, sans modifier le contenu original. Cette approche présente un grand avantage : elle garantit une fidélité totale au contenu source ,ce qui permet de conserver l'intégralité des informations sans risque de distorsion ou de perte de sens .au contraire de la synthèse générative ,qui peut parfois réécrire des concepts ou reformuler des informations de manière approximative ,l'extraction se limite à sélectionner les phases les plus significatives présente dans le texte ,évitant ainsi toute altération du contenu original ,elle élimine le risque de mauvaise

interprétation d'information .cette méthode est particulièrement utile dans des contextes où la précision des détails est crucial surtout dans le domaine de la recherche scientifique de plus , les méthodes extractives sont relativement simples à implémenté par rapport aux approches génératives , qui nécessitent des modèles de langage complexes et des phases d'entraînement basées sur des ensembles de données massifs ce qui fait d'elle une méthode incessante et pertinente.

Le développement des technologies informatiques a permis l'émergence de plusieurs familles d'algorithmes de synthèse automatique basés sur l'extraction. Parmi ces familles, on distingue notamment les algorithmes fondés sur le classement par graphe (Graph-Based Ranking), les méthodes basées sur la fréquence statistique (Statistical Frequency-Based), et dans ce mémoire nous allons voir une version hybride entre la famille des Basés sur des graphes de similarité et Basés sur la fréquence brute des mots graphe.

### **1.5 Les algorithmes fondés sur le classement par graphe :**

Les algorithmes de synthèse automatique fondés sur le classement par graphe de texte tels que PageRank, TextRank ou LexRank s'appuient tous sur un principe commun issu du domaine de la théorie des graphes l'idée selon laquelle l'importance d'un élément peut être déterminée à partir des liens qu'il entretient avec d'autres éléments dans un réseau. Initialement conçu pour classer les pages web, ce principe a été adapté avec succès au traitement automatique de langage naturel ,notamment dans le domaine de la synthèse de texte ,le fonctionnement général de ces algorithmes repose sur une modélisation du texte sous forme de graphe .dans ce graphe ,les nœuds représentent des unités textuelles ,qui peuvent être des mots , phrases ou même des paragraphe ,en fonction de la tâche à accomplir les arêtes entre les nœuds traduisent une relation de similarité , par exemple deux phrase seront connectées dans le graphe si elles partagent un certain nombre de mots en commun ,ou si elles traitent de concepts proches une fois ce graphe construit l'algorithme attribue en score l'importance à chaque nœud à l'aide d'un mécanisme de propagation : un nœud est jugé important s'il est fortement relié à d'autres nœuds eux même important .Ce mécanisme est itératif les scores sont recalculés à plusieurs reprises jusqu'à atteindre une certaine stabilité. C'est sur cette base que les algorithmes comme Page Rank, TextRank et LexRank opèrent. Ils appliquent une formule mathématique qui simule une forme de vote entre les nœuds : un nœud reçoit du poids des autres nœuds qui lui sont liés, et plus ces nœuds sont eux-mêmes influents, plus le poids reçu est grand ;ce système permet de faire émerger les éléments les plus centraux du graphe ,c'est-à-dire ceux qui occupent une place

stratégique dans le réseau des relations .dans le contexte de la synthèse de texte .ces éléments centraux correspondre souvent aux phrases qui résument le mieux le contenu du document .Un des grands intérêts de ce type d'approche est qu'elle ne repose pas nécessairement sur une compréhension du texte ni sur une analyse linguistique complexe ,elle s'appuie sur la structure même du document est sur la manière dont ses composantes interagissent entre elles ,ce qui la rend robuste et adaptable a de nombreux types de textes .De plus , ces algorithmes ne nécessitent généralement pas d'apprentissage supervisé ,ce qui facilite leurs déploiement dans des contextes variés

### 1.5.1 PageRank

(Brin & Page, 1998) [20] : bien que conçu pour classer les pages Web, l'algorithme de PageRank original sert de référence. TextRank et LexRank en sont des dérivés, appliquant PageRank sur un graphe de phrases [21]. Son but principal est de déterminer l'importance relative des pages web sur Internet en fonction de la structure de leurs liens. Contrairement aux méthodes traditionnelles qui évaluent la pertinence d'une page en fonction de son contenu, PageRank s'intéresse à la manière dont les pages sont interconnectées, en supposant que les pages les plus importantes sont celles qui sont liées à d'autres pages importantes

Le principe de PageRank repose sur l'idée que si une page A pointe vers une page B via un lien, cela peut être interprété comme un vote en faveur de la page B. Cependant, tous les votes ne sont pas égaux : un lien provenant d'une page déjà bien classée ou ayant un PageRank élevé est considéré comme plus précieux qu'un lien d'une page moins importante. En d'autres termes, la qualité du lien (c'est-à-dire la réputation de la page qui émet le lien) est tout aussi significative que la quantité de liens entrants. Le calcul de PageRank se fait de manière itérative. Chaque page se voit attribuer un score initial, généralement égal. Ensuite à chaque étape , ce score est redistribué entre les pages pointées par la page en question .par exemple si une page A un PageRank de 1 et fait un lien vers trois pages B ,C et D ,chaque page B ,C et D reçoit une fraction de PageRank de A ,à savoir  $1/3$  de ce score ce processus est répété de manière récursive pour toutes les pages web jusqu'à ce que les scores convergent vers une valeur stable .Il est important de noter que PageRank ne se limite pas à un simple calcul de liens .il prend également en compte le dumping factor ,un paramètre qui modélise la probabilité qu'un utilisateur clique sur un liens dans une page donnée .Ce facteur est important pour éviter un cycle infini de liens ou une page pourrait accumuler indéfiniment du PageRank ,

Ce dumping factor est généralement fixé à 0.85, ce qui signifie qu'il y a une probabilité de 15% que l'utilisateur décide de ne pas suivre un lien et de sauter vers une autre page de manière aléatoire. L'algorithme PageRank permet ainsi de donner une mesure d'importance à chaque page web, indépendamment de son contenu textuel.

```
1  fonction pageRank(G,d,N,ε) :PR[V]
2      G = (V, E) // Graphe orienté des pages (V = ensemble de pages, E = ensemble des liens)
3      d = 0.85 // Facteur d'amortissement (souvent 0.85)
4      N = |V| // Nombre total de pages
5      ε = 0.0001 // Seuil de convergence
6
7
8  début :
9      Pour chaque page v dans V :
10         PR[v] ← 1 / N // Valeur initiale de PR égale pour toutes les pages
11
12  Itération :
13      Répéter
14         Pour chaque page v dans V :
15             tempPR[v] ← (1 - d) / N
16             Pour chaque page u ∈ V telle que (u → v) ∈ E :
17                 tempPR[v] ← tempPR[v] + d × (PR[u] / L[u])
18                 // L[u] est le nombre de liens sortants de la page u
19
20         delta ← 0
21         Pour chaque page v dans V :
22             delta ← delta + |tempPR[v] - PR[v]|
23             PR[v] ← tempPR[v]
24
25     Jusqu'à ce que delta < ε
26
27  fin
28  Retourner PR // Vecteur des scores PageRank
```

Figure 1. Algorithme PagerRank

### 1.5.2 LexRank

Similaire à TextRank, LexRank représente chaque phrase par un vecteur TF-IDF et mesure la similarité cosinus pour construire le graphe. L'importance d'une phrase est alors calculée par la centralité propre (eigenvector centrality) dans ce graphe [22]. L'algorithme de synthèse de texte basé sur la théorie des graphes, qui utilise une approche de graphes de similarité pour extraire les phrases les plus pertinentes d'un document afin de générer un synthèse.

Son fonctionnement repose sur l'idée que les phrases importantes dans un texte sont celles qui sont les plus liées aux autres, dans le sens où elles partagent des informations significatives et

sont liées par des similarités sémantiques. L'algorithme est inspiré du PageRank ,mais au lieu de classer des pages web en fonction de la structure des liens ,Lexrank s'intéresse aux relations de similarité entre les phrases dans un document ,le principe de Lexrank repose sur la création d'un graphe de similarité où chaque nœud représente une phrase du document et chaque arête relie deux phrases similaires .La similarité entre deux phrase est calculée généralement en utilisant des mesures telles que le cosinus de similarité , qui mesure la similarité entre les vecteurs de représentation des phrases ,souvent obtenus à partir de technique comme TF-IDF . plus des phrases sont similaires , plus l'arête entre elles ont une forte pondération .une fois ce graphe de similarité construit ,LexRank applique un algorithme de centralité pour déterminer quelles phrases sont les plus importantes .en d'autres termes ,il s'agit d'identifier les phases qui sont au centre du graphe c'est-à-dire celles qui partagent les liens les plus forts avec d'autres ,ces phrase sont considérées comme les plus représentatives du document et , par conséquent ,seront sélectionnées pour constituer le synthèse .

LexRank applique un algorithme de centralité pour déterminer quelles phrases sont les plus importantes. En d'autres termes, il s'agit d'identifier les phrases qui sont au centre du graphe, c'est-à-dire celles qui partagent les liens les plus forts avec d'autres phrases. Ces phrases sont considérées comme les plus représentatives du document et, par conséquent, seront sélectionnées pour constituer le synthèse.

```

1  fonction Lexrank(D,threshold,d,ε) :PR[V
2
3      D = {S1, S2, ..., Sn} // Ensemble de phrases (documents ou textes à résumer)
4      threshold = 0.1 // Seuil de similarité pour établir les arêtes
5      d = 0.85 // Facteur d'amortissement
6
7      ε = 0.0001 // Seuil de convergence
8  Début
9
10 Étapes :
11
12  1. Calcul des vecteurs TF-IDF :
13     Pour chaque phrase Si dans D :
14         Calculer TF-IDF(Si) // Représenter chaque phrase comme un vecteur
15
16  2. Construction du graphe :
17     Pour chaque paire (Si, Sj) avec i ≠ j :
18         sim ← cosinus_similaire(TF-IDF(Si), TF-IDF(Sj))
19         Si sim ≥ threshold :
20             Ajouter une arête pondérée w(i, j) = sim au graphe
21
22  3. Normalisation des poids :
23     Pour chaque phrase Si :
24         Normaliser les poids des arêtes sortantes à partir de Si
25
26  4. Initialisation du score :
27     Pour chaque phrase Si :
28         Score[Si] ← 1 / n // Score initial égal
29
30  5. Itération de type PageRank :
31     Répéter
32         Pour chaque phrase Si :
33             tempScore[Si] ← (1 - d) / n
34             Pour chaque phrase Sj liée à Si :
35                 tempScore[Si] += d × (Score[Sj] × w(j, i)) // pondéré
36
37         delta ← 0
38         Pour chaque phrase Si :
39             delta ← delta + |tempScore[Si] - Score[Si]|
40             Score[Si] ← tempScore[Si]
41
42     Jusqu'à ce que delta < ε
43
44  Retourner Score
45

```

Figure 2. Algorithme LexRank

### 1.5.3 PageRank

(Mihalcea & Tarau, 2004) [23]: Ce modèle de classement textuel dérivé de PageRank applique l'algorithme de PageRank sur un graphe où les nœuds sont les phrases du document et les arêtes leurs similarités. TextRank peut aussi extraire des mots-clés selon la même idée.

Transposé au domaine du traitement automatique du langage naturel (TALN), TextRank sert à identifier les unités textuelles les plus significatives – généralement des phrases – dans un document, en vue de générer un synthèse extractif cohérent et informatif. Contrairement aux approches purement statistiques ou aux modèles complexes de génération de texte, TextRank propose une méthode non supervisée, simple à mettre en œuvre, qui ne nécessite pas de données annotées ni de phase d'entraînement préalable. Le principe de TextRank repose sur la modélisation du texte sous forme de graphe : chaque nœud du graphe représente une phrase (ou parfois un mot, selon l'usage), et les arêtes entre les nœuds représentent les relations de similarité entre ces phrases. Ces similarités sont souvent calculées à l'aide de la mesure de cosinus sur les représentations vectorielles des phrases, utilisant des pondérations TF-IDF. Deux phrases seront liées dans le graphe si elles partagent un certain degré de contenu lexical ou sémantique, et plus cette similarité est forte, plus l'arête entre elles aura un poids élevé. Une fois ce graphe construit, TextRank applique un algorithme itératif de classement des nœuds, sur le modèle de PageRank. Autrement dit, une phrase est considérée comme importante si elle est liée à d'autres phrases importantes. Ce mécanisme de vote mutuel entre phrases permet de faire émerger celles qui jouent un rôle central dans le document, c'est-à-dire les phrases qui capturent le cœur du contenu. Après convergence du processus, les phrases ayant obtenu les scores les plus élevés sont sélectionnées pour constituer le synthèse final, dans l'ordre d'apparition ou selon une stratégie de réorganisation. Un des atouts majeurs de TextRank est sa robustesse et son adaptabilité. Il peut être appliqué à différents types de textes et dans plusieurs langues, tant que les similarités lexicales peuvent être mesurées. Il est également utilisé dans d'autres tâches du TALN, comme l'extraction de phrases ou éléments clé ou la segmentation de texte. Sa capacité à identifier les éléments centraux d'un document sans avoir recours à une compréhension sémantique profonde en fait un outil particulièrement utile dans les systèmes de synthèse automatique, notamment pour traiter de grandes quantités d'informations de manière rapide et efficace. En synthèse, TextRank offre une solution élégante et performante pour la synthèse de texte en exploitant la structure relationnelle entre les unités textuelles. Grâce à son approche non supervisée et basée sur les graphes, il permet de condenser un document tout en préservant l'essentiel de son contenu, sans nécessiter d'apprentissage complexe ou de ressources coûteuses.

```

1 fonction PageRankrank(G,N,d,ε) :PR[V]
2
3     G = (V, E)           // Graphe orienté (pages, phrases, ou nœuds quelconques)
4     d = 0.85             // Facteur d'amortissement
5     ε = 0.0001          // Seuil de convergence
6     N = |V|              // Nombre total de nœuds
7 Début
8
9 Étapes :
10
11 1. Initialisation :
12   Pour chaque nœud v dans V :
13     PR[v] ← 1 / N      // Score initial uniforme
14
15 2. Répéter jusqu'à convergence :
16   Répéter
17     Pour chaque nœud v dans V :
18       tempPR[v] ← (1 - d) / N // Saut aléatoire
19
20     Pour chaque nœud u dans V :
21       Pour chaque voisin v tel que (u → v) ∈ E :
22         tempPR[v] ← tempPR[v] + d × (PR[u] / L[u])
23         // L[u] est le nombre de liens sortants de u
24
25     delta ← 0
26     Pour chaque nœud v dans V :
27       delta ← delta + |tempPR[v] - PR[v]|
28       PR[v] ← tempPR[v]
29
30   Jusqu'à ce que delta < ε
31
32 Retourner PR

```

**Figure 3.** Algorithme TextRank

## 1.6 Comparaison entre TextRank, LexRank et PageRank :

Ce travail se penchera sur une comparaison approfondie des algorithmes TextRank, LexRanks et PageRank, déterminant les similitudes et différences entre eux à différents degrés. Premièrement, on référera à leur principe. Bien que tous soient basés sur un score de propagation dans le graphe, les méthodes sélectionnant les phrases ou les mots-clés sont différentes, selon la spécificité des liens formés entre les nœuds. De plus, le type de graphe et la façon dont les phrases ou les mots sont connectés seront analysés, spécifiant dans quels cas les graphes sont orientés ou non, pondérés ou non. Un autre critère sera la manière de former des arêtes entre les nœuds et la mesure de simplicité utilisée pour cela : cette partie reflète si la similarité cosinus, le TF-IDF ou d'autres mesures statistiques ou sémantiques sont utilisées.

Ensuite, la complexité algorithmique et le temps d'exécution seront analysés, montrant quelle proposition est la meilleure par ses performances. De plus, il existe une sorte de comparaison – la qualité des synthèses. L'évaluation ROUGE s'attaque à l'objectivité de sa mesure, car il est extrêmement difficile de juger de la pertinence, de la cohérence et de la représentation en même temps tout en qualifiant le test. De cette manière, la comparaison vise à déterminer les atouts et les défis de textes pour chaque algorithme de synthèse automatique, lançant le meilleur compromis entre le meilleur temps de travail et la meilleure qualité de synthèse. Le texte qui va être utilisé pour réaliser la comparaison est le suivant « L'histoire de l'informatique est l'histoire de la science du traitement rationnel, notamment par machines automatiques, de l'information. Celle-ci est considérée comme le support des connaissances humaines et des communications dans les domaines techniques, économiques, sociaux ainsi que professionnels.

L'histoire de l'informatique a commencé bien avant la discipline moderne des sciences informatiques, généralement par les mathématiques ou la physique. Les développements des siècles précédents ont évolué vers la discipline que nous connaissons aujourd'hui sous le nom d'informatique [1]. Cette progression, des inventions mécaniques et des théories mathématiques vers les concepts et les machines informatiques modernes, a conduit au développement d'un domaine académique majeur, à un progrès technologique spectaculaire à travers le monde occidental et à la base d'un commerce et d'une culture mondiale massive [2].

En 1966, l'informatique a été définie par l'Académie française comme la « science du traitement rationnel, notamment par machines automatiques, de l'information considérée comme le support des connaissances humaines et des communications dans les domaines techniques, économiques et sociaux »[3].

L'histoire de l'informatique résulte de la conjonction entre des découvertes scientifiques et des transformations techniques et sociales :

Découvertes physiques sur les semi-conducteurs et la miniaturisation des transistors ;

Découvertes mathématiques sur la calculabilité et les propriétés des algorithmes ;

Invention de la théorie de l'information, de la sémiotique et de la cybernétique ;

Transformations techniques, l'introduction de machines ou de composants informatiques, des machines à laver aux automobiles et aux avions, des banques à la santé, de l'imprimerie à la documentation en ligne ;

Transformations sociales avec l'organisation des entreprises et des administrations autour de leur système d'information automatisé, et avec la circulation de l'information dans les réseaux sous forme numérique.

Si la plupart des ordinateurs ont été conçus au départ pour exécuter des calculs numériques trop longs ou trop compliqués pour être effectués à la main, des machines similaires ont été construites pour traiter des informations non numériques (par exemple, reconnaître une chaîne de caractères dans un texte, ce que faisait dès 1943 le Colossus du service de cryptanalyse britannique [4]). Les calculateurs devenaient des machines universelles de traitement de l'information, d'où le mot ordinateur, retenu en 1956 pour trouver un équivalent français à l'expression anglaise data processing machine. Ce terme a progressivement remplacé en français celui de calculateur, au sens trop restreint.

Les premiers ordinateurs datent de 1949. C'est la notion de programme enregistré, due à John von Neumann et à ses collaborateurs, en 1945, qui transforme les machines à calculer en ordinateurs. La machine est composée des éléments suivants :

L'unité arithmétique et logique, un organe de calcul, susceptible d'exécuter les opérations arithmétiques et logiques ;

Une mémoire, ou mémoire centrale, servant à la fois à contenir les programmes décrivant la façon d'arriver aux résultats et les données à traiter ;

Des organes d'entrée-sortie, ou périphériques, servant d'organes de communication avec l'environnement et avec l'homme ;

Une unité de commande (control unit) permettant d'assurer un fonctionnement cohérent des éléments précédents.

L'ensemble formé par l'unité arithmétique et logique, d'une part, et l'organe de commande, d'autre part, constitue l'unité centrale ou processeur. L'ensemble des composants physiques, appelé matériel (hardware), est commandé par un logiciel (software).

Selon l'architecture de von Neumann, les programmes sont enregistrés dans la mémoire de la machine. Ils peuvent comporter des boucles de calcul et des alternatives, contrairement aux programmes exécutés à partir de bandes perforées. Alors qu'on connaît depuis longtemps le codage des procédures par des trous sur des bandes perforées (machines à tisser, orgue de barbarie) lorsque le programme est enregistré en mémoire, il n'y a pas de différence fondamentale entre coder des données et coder des procédures.

On peut donc calculer sur les instructions d'une procédure, par exemple pour faire un saut en arrière dans les instructions, ou pour compter le nombre de tours d'une boucle de calcul sur les éléments d'une liste. Ce calcul sur les programmes a permis le développement du logiciel : langages de programmation, systèmes d'exploitation, applications.

L'histoire de l'informatique porte donc seulement sur une soixantaine d'années. Elle peut se décliner selon quatre points de vue :

La dimension physique qui décrit les moyens matériels du traitement, de la conservation et du transport de l'information : les machines et les réseaux ;

La dimension logicielle qui définit d'une part les méthodes de calcul, de mémorisation et de commande qui caractérisent les fonctions des ordinateurs, et d'autre part les interfaces entre les personnes et les machines depuis l'arrivée des ordinateurs individuels vers 1980 ;

La dimension applicative qui définit les objets et les processus informationnels utilisés, traités, transformés et montrés ;

La dimension sociale qui décrit les usages de l'informatique dans tous les domaines et son insertion dans les pratiques sociales les plus diverses.

L'informatique est une nouvelle technologie intellectuelle, comme l'ont été en leur temps l'écriture et l'imprimerie, un moyen de représenter, de structurer et d'exploiter des informations qui en retour structure la pensée de l'homme, selon Pierre Lévy. »

### 1.6.1 Principe de base

- PageRank: il attribue un score à chaque page en fonction du nombre et de la quantité des liens entrants, modélisant ainsi l'importance relative des pages dans un réseau hypertexte
- TextRank : c'est une adaptation de PageRank pour le traitement de langage naturel, utilisée notamment pour la synthèse automatique et l'extraction de mots-clés. L'algorithme construit un graphe où les nœuds représentent des phrases ou des mots, reliés par des arêtes pondérées selon leur similarité. Le score de chaque nœud est déterminé de manière itérative, à l'image de PageRank, afin de refléter l'importance des phrases dans le texte
- LexRank : c'est un autre algorithme de synthèse basé sur les graphes, qui repose également sur le principe de PageRank. Sa principale différence réside dans l'utilisation d'une mesure de centralité fondée sur la similarité entre phrases. Il introduit souvent un seuil de similarité pour générer un graphe non pondéré, en mettant l'accent sur les phrases les plus centrales dans le graphe.

Ce tableau représente le type de graphe, la mesure de similarité, la complexité temporelle et le temps d'exécution :

**Tableau 1.**

	<b>PageRank</b>	<b>TextRank</b>	<b>LexRank</b>
<b>Type de graphe</b>	Orienté et pondéré	Non orienté et pondéré	Non orienté
<b>Mesure de similarité</b>	Liens hypertextes	Similarité cosinus	Similarité cosinus + seuil de conservation
<b>Complexité temporelle</b>	$O(k \times (n + e))$	$O(n \times m + n^2 + k \times n^2)$	$O(n \times m + n^2 + k \times n^2)$ .
<b>Temps d'exécution</b>	0.027440309	0.109176158	0.067142009

### 1.6.2 Qualité du synthèse

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) est une famille de métriques utilisées pour évaluer automatiquement la qualité des synthèses ou des systèmes de génération de texte en les comparant à un ou plusieurs synthèses de référence humains. Les variantes les plus courantes sont ROUGE-N, qui mesure le chevauchement des n-grammes (par exemple, ROUGE-1 pour les monogrammes, ROUGE-2 pour les bigrammes), ROUGE-L est basé sur la longueur de la plus longue sous séquence commune entre le synthèse généré et un synthèse de référence.

**Tableau 2.**

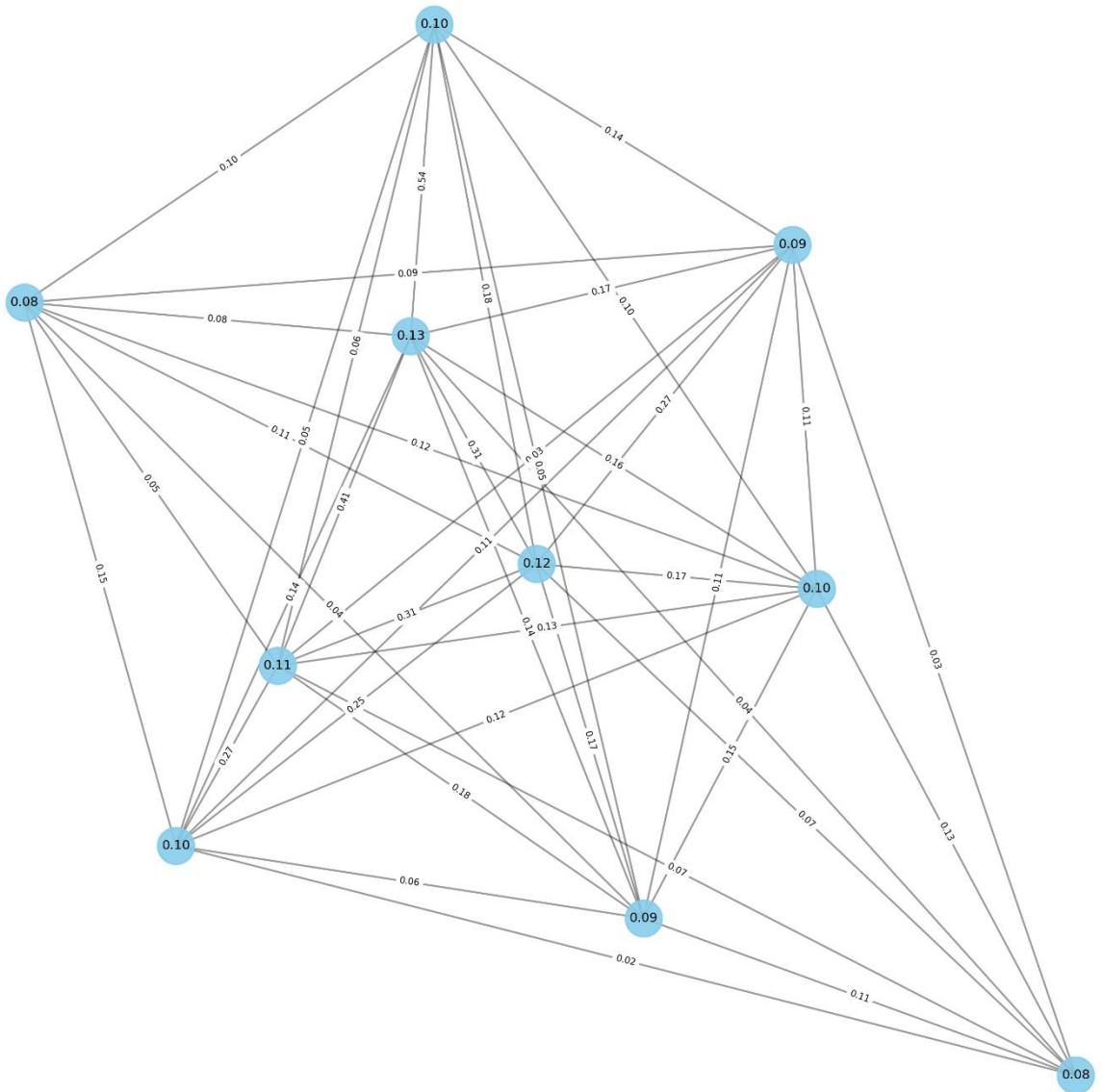
<b>Algorithme</b>	<b>Type rouge</b>	<b>Précision</b>	<b>Rappel</b>	<b>F1 score</b>
<b>PageRank</b>	ROUGE1	1.0000	0.2369	0.3831
	ROUGE2	1.0000	0.2352	0.3808
	ROUGEL	0.8750	0.2073	3352
<b>TextRank</b>	ROUGE1	0.9948	0.4396	0.6098
	ROUGE2	0.9896	0.4361	0.6054
	ROUGEL	0.9948	0.4396	0.6098
<b>LexRank</b>	ROUGE1	0.9948	0.4396	0.6098
	ROUGE2	0.9896	0.4361	0.6054
	ROUGEL	0.9948	0.4396	0.6098

### **1.7 Les graphes générés :**

Dans la phase de test voici les graphes générés par les scripts python contenu dans ce mémoire appliqué sur le paragraphe les valeurs des nœuds et arrêtes sont représenté dans ce tableau.

**Tableau 3.**

	<b>PageRank</b>	<b>Textrank</b>	<b>LexRank</b>
<b>Nœud</b>	Phrase	Phrase	Phrase
<b>Arête</b>	Liens hypertextes	Similarité cosinus	Similarité cosinus et seuil



**Figure 4.** Graphe pagerank

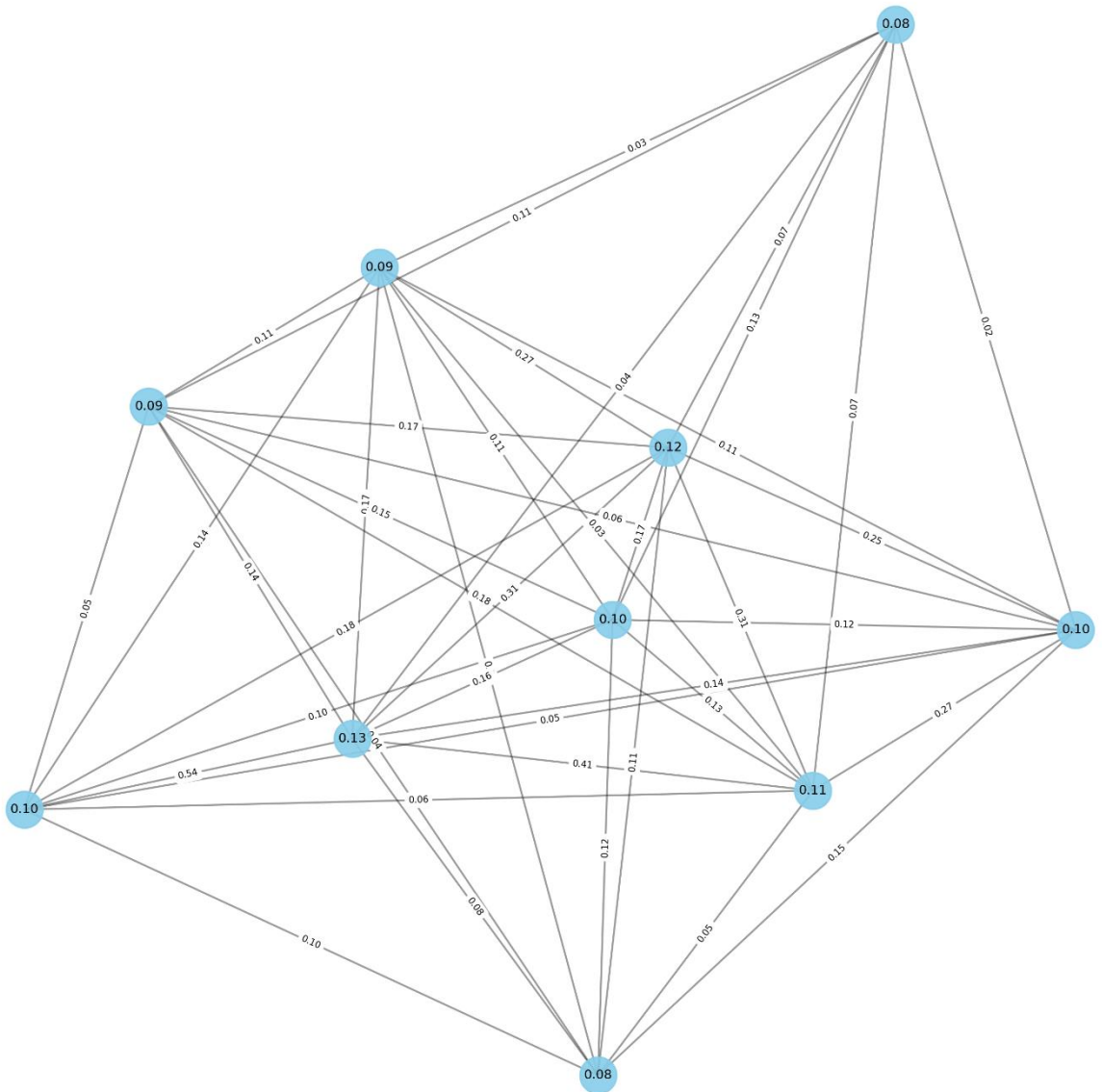
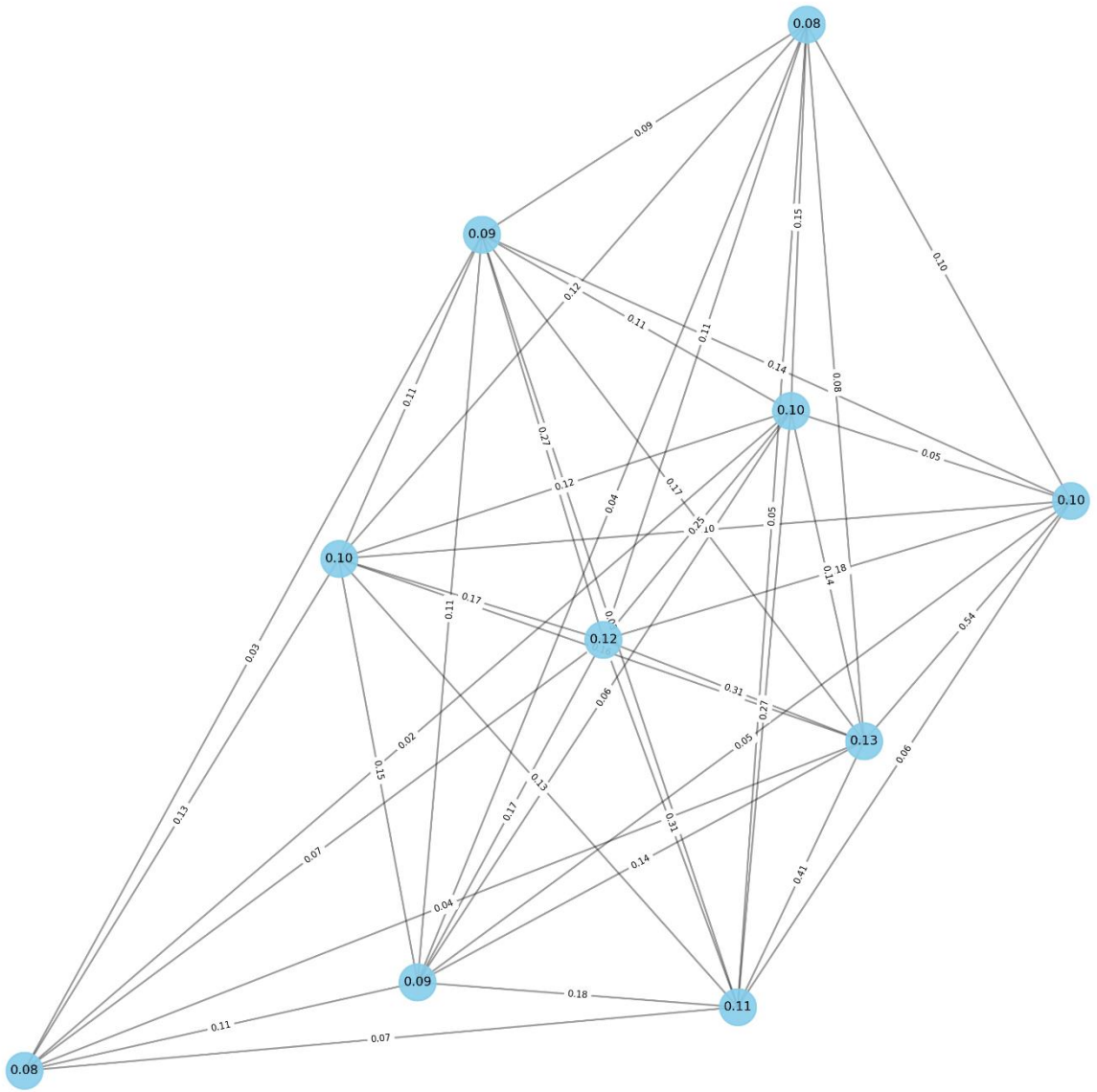


Figure 5. Graphe LexRank



**Figure 6.** Graphe TextRank

## 1.8 Conclusion :

À partir de l'analyse comparative des trois algorithmes TextRank, LexRank et PageRank plusieurs éléments résultant pour juger de leurs performances en synthèse automatique. Ces méthodes, bien qu'inspirées de algorithmes PageRank initialement conçus pour évaluer l'importance des pages web à travers un graphe dirigé, s'en distinguent par leurs adaptations au traitement du langage naturel. TextRank et LexRank reposent sur des graphes non dirigés où chaque nœud représente une phrase, connectée aux autres par des liens pondérés selon une similarité lexicale ou sémantique, généralement mesurée à l'aide de la similarité cosinus appliquée sur des vecteurs TF-IDF.

LexRank ajoute toutefois un seuil de similarité, ce qui a pour effet d'alléger le graphe et de mieux faire ressortir les phrases les plus importantes et représentatives

Du point de vue complexité algorithmique, TextRank et LexRank ont une structure comparable, avec un coût évalué à  $O(n \times m + n^2 + k \times n^2)$ , tandis que PageRank, bien qu'il utilise un graphe dans, peut présenter une complexité équivalente, voire légèrement inférieure dans certains cas

Concernant les performances, les scores ROUGE révèlent que TextRank et LexRank offrent des résultats quasi identiques, avec une précision très élevée (près de 0,995) et un bon équilibre entre rappel et F1-score (avec un F1 de 0,61). À l'inverse, même si PageRank affiche une précision parfaite (1,000), ses faibles scores en rappel et en F1 indiquent qu'il omet une quantité importante d'information pertinentes dans les résumés produits. Sur le plan du temps d'exécution, PageRank est le plus rapide (0,027s), suivie de LexRank (0,067s), puis de TextRank (0,109), néanmoins, cette différence reste minime par rapport à l'importance de la qualité des résultats.

En conclusion, le choix de l'algorithme repose sur la qualité du synthèse généré et l'efficacité en termes de ressources. Même si PageRank est le plus rapide, ses performances en qualité et richesse d'information sont insuffisantes surtout dans le cas de la recherche scientifique. TextRank, pour sa part, se distingue par sa précision, sa robustesse et capacité à générer des synthèses pertinentes, tout en restant rapide pour une utilisation en conditions réelles. Ainsi, l'algorithme TextRank apparaît comme le meilleur compromis entre qualité, efficacité, simplicité et mise en œuvre pour des tâches de synthèse automatique

# Chapitre 2 Traitement du langage naturel (NLP) et grands modèles de langage (LLM)

## 2 Introduction :

### 2.1 Définition :

Le traitement du langage naturel (NLP) est une branche de l'intelligence artificielle qui a pour but de permettre aux ordinateurs de comprendre, interpréter et générer le langage humain [24]. Il couvre un grand nombre de tâches : traduction automatique, classification de texte, synthèse de document, analyse de sentiments, question réponse ect, par exemple , NLP est utilisé dans les assistants virtuel comme siri ou alexa ,les filtres anti-spam , la traduction automatique et la génération de synthèses de document [25] .

Un grand modèle de langage (LLM) est un modèle de deep learning pré-entraîné sur d'énormes corpus de texte capable de réaliser plusieurs tache de NLP [26].

Le LLM utilisent des architectures à base de tranformers et sont entraînées sur des ensemble de données gigantesques. Après pré-entraînement , ils peuvent être affinées (fine-tuning) sur des tâches spécifiques comme la classification de texte, réponse à des questions ,le synthèse de documents ou la génération de texte [27].

### 2.2 Prétraitement et représentation vectorielle :

Les données textuelles brutes doivent être prétraitées avant d'être fournies à un modèle. le prétraitement classique inclut de nettoyer le texte (minuscule suppression de balise HTML, ponctuation etc.).la tokenization et la suppression de mots vides, la lemmatisation.

ma tokenization consiste à découper le texte en unités (tokens)élémentaire (mots ou sous mots) elle est souvent effectuée par des algorithmes déterministes basés sur des expressions régulières optimisés [28]. Par exemple on peut séparer la ponctuation des mots ou scinder les mots composés, tout en traitant les nombres ou les abréviations [29] découpage est critique car il condéennes toutes les étapes suivantes

chaque token est ensuite converti en une représentation numérique .les techniques simple incluent les modèles bag of words ou TF-IDF , qui ne capturent pas la sémantique ,les embeddings de mots apprennent au contraire des vecteur dense ou des mots sémantiquement proches et non loins de l'espace vectoriel .ces embeddings statiques sont souvent remplacées par des représentation contextuelle ou chaque mot prend un vecteur différent selon son contexte , les LLM modernes (comme BERT ) génèrent de tels embeddings contextuels permettant de mieux encadrer la signification des tokens dans la phrase

Les modèles basés sur des architectures récurrentes (RNN/LSTM/GRU) étaient historiquement dominants pour modéliser les séquences de mots. Les meilleurs modèles de traduction de séquences reposaient sur des réseau récurrent ou convolutionnelles avec encodeur décodeur , parfois avec l'aide d'un mécanisme d'attention [papers.neurips.cc](https://papers.neurips.cc) [30] .ces architectures consommaient beaucoup de temps de calcul pour de longues séquences

### **2.3 Architecture transformer et mécanisme d'attention :**

En 2017 ,vaswani et al . ont introduit le transformer , une architransformerement fondée sur l'attention, éliminant les récurrences et convolution [31] Le transformer se compose d'un empilement de blocs d'encodeur et décodeur ,chaque bloc d'encodeur contient notamment un mécanisme d'auto-attention multi-têtes suivie d'un réseau feed-forward,le tout avec une normalisation et connexion résiduelles.

Dans le décodeur, on ajoute une couche d'attention croisée qui permet d'intégrer les informations de l'encodeur ainsi , dans l'encodeur du transformer (chaque couche a deux sous couches la première est une attention multi-têtes, et la seconde un réseau feed-forward )\_[32].

Les mécanismes d'attention permettent au modèle de pondérer d'une façon dynamique les différentes positions d'une séquence lors du calcul de la représentation d'un mot. L'auto-attention relie chaque mot a tous les autres de la séquence pour extraire un contexte global. Cette approche facilite la modélisation des dépendances à longue distance. vaswani et al .notent que l'attention rend possibles des dépendances arbitraires sans tenir compte de la distance [33]

de plus, l'attention multi-têtes(multi-head) divise le vecteur de requête/clé/valeur en plusieurs sous espaces et effectue plusieurs opérations d'attention en parallèle, ce qui permet au modèle de prendre en compte simultanément des informations provenant de différents sous espaces de représentation »t [34],cela renforce la capacité du modèle à capter différents aspects sémantiques simultanément

L'apprentissage des LLM se fait en deux étapes clés. D'abord, un pré-entraînement auto supervisé sur d'immenses corpus de textes apprend des représentations linguistiques générales. Les grandes tranches de données sont cruciales : par exemple gpt-2 (2019) a été pré-entraîné sur webtext; un corpus de millions de pages web non filtrées [35] ensuite, le modèle peut être affiné sur des tâches cibles spécifique via un ajustement supervisé. Dans ce contexte de nombreux LLM sont d'abord entraînés sur une tâche générique (comme la prédiction du mot manquant ou l'auto-régression), puis adaptés à des tâches diverses. OVH cloud rappelle que les grands modèles de langage doivent être pré entraînés puis affiné pour résoudre des problèmes tels que la classification, la réponse à des questions, la synthèse de documents et la génération de texte [36] cette approche pré-entraînement et fine tuning a permis d'atteindre d'excellents résultats sur les benchmarks

## **2.4 Entraînement à grande échelle et capacités des LLM :**

La taille des modèles est un facteur important. Les LLM se comptent souvent en plusieurs paramètres. Par exemple, le transformer original de Vaswani atteignant déjà un nouveau sommet de performance en traduction automatique en comptant 65m de paramètres. Aujourd'hui, les modèles vont bien au-delà : GPT-2 est un transformer autorégressif de 1,5 milliard de paramètres, tandis que GPT-3 en compte 175 milliards. Les chercheurs ont constaté que l'augmentation linéaire du nombre de paramètres (et de la quantité de données d'entraînement) améliore les performances sur les tâches NLP de manière remarquable. En pratique, cela nécessite des ressources de calcul colossales et des architectures distribuées (gpu)

De plus, les LLM modernes disposent souvent d'un contexte long, leur permettant de prendre en compte un très grand nombre de mots. Cette capacité est essentielle pour résumer ou générer du texte long (fenêtre de tokens étendue), leur permettant de prendre en compte un très grand nombre de mots. Cette capacité est importante pour résumer ou générer du texte long, comme des articles scientifiques, en utilisant l'apprentissage auto supervisé à grande échelle, les LLM accumulent des connaissances linguistiques issues de leur corpus, ce qui leur donne une sorte de banque de mémoires sur le langage

## **2.5 Principaux domaines d'application :**

Les LLM sont polyvalents et couvrent plusieurs domaines comme :

### **2.5.1 La traduction automatique**

Les Transformers ont révolutionné la traduction, le modèle transformer original a atteint 41,0 BLEU sur la traduction anglais français [37], battant les meilleurs modèles antérieurs. Les LLM pré

entraînés fournissent des bases solides qui peuvent être adaptées à différentes paires de langues, souvent avec moins de données de traduction spécifique

### **2.5.2 Classification de texte et NLU**

Des tâches comme la classification thématique, détection de sentiment, la reconnaissance des entités nommées ou analyse de sentiment bénéficient grandement des représentations apprises par les LLM. Par exemple, BERT (2018) a permis d'obtenir de nouveaux records sur de nombreux jeux de données de NLP grâce à son modèle pré-entraînement bidirectionnel [38].

Ainsi, un modèle BERT pré-entraîné s'améliore facilement sur une tâche comme la classification d'email en spam ou l'identification d'intention dans un dialogue

### **2.5.3 Synthèse de texte et synthèse de documents**

Les synthèses automatiques visent à produire une synthèse d'un long document. Les approches traditionnelles étaient souvent extractives (sélection de phrases clés). Aujourd'hui, les LLM permettent des synthèses abstraites plus fluides. En particulier, la structure complexe des articles scientifiques rend cette tâche difficile, mais les modèles basés sur transformers atteignent des performances de pointe en synthèse automatique de publication scientifique [39]. Aswani et al. (2024) soulignent que l'architecture transformer a créé un grand changement dans le NLP: les modèles basés sur les transformers sont capables d'atteindre des résultats à l'état de l'art en synthèse de textes, y compris les articles scientifiques [40]. Les LLM peuvent ainsi générer des synthèses de document scientifique en traitant de grands documents et en identifiant automatiquement les points clés.

Génération de texte : les LLM sont également utilisés pour produire du nouveau texte (NLG). Selon Brown et al. (2020) : nous entraînons GPT-3, un modèle de langage autorégressif de 175 milliards de paramètres ... nous trouvons que GPT-3 peut générer des échantillons d'articles de presse que des évaluateurs humains ont du mal à distinguer de textes humains [arxiv.org](https://arxiv.org) [41]. Ainsi les LLM excellent dans la génération de textes cohérents sur de longues séquences et peuvent être sollicités en mode prompt pour répondre à des questions, rédiger du contenu ou même écrire du code, souvent sans entraînement supplémentaire

## **2.6 Exemples de modèles et compétences de LLM :**

Plusieurs LLM ont été proposés ces dernières années. Voici quelques exemples :

BERT(2018) : Un modèle pré-entraîné par Google, c'est un encodeur-uniquement (pas d'auto-régression) basé sur l'architecture transformer, pré-entraîné à la prédiction de tokens masquées et

d'ordres de phrase .BERT-base compte environ 110 millions de paramètre [\[42\]](#). Après pré-entraînement, Bert s'améliore aisément pour des tâches comme la classification ou la réponse des questions [\[43\]](#).

GPT-2 (2019) un modèle décodeur-seulement (transformer autorégressif d'OpenAI.GPT-2 utilise l'apprentissage non supervisé sur le corpus webtext de million de pages web [\[44\]](#). Sa plus grande version comprend 1.5 milliard de paramètres [\[45\]](#). GPT-2 réalise des générations puissantes en zéro-shot sur des taches variées (traduction, question réponse ect) grâce à ses capacités de génération de texte fluide

LLaMA(2023) famille de modèle par Meta , allant de 7B a 65B de paramètre [\[46\]](#) .ces modèles ont montré qu'il est possible de rivaliser avec GPT-2 et GPT-3 tout en s'appuyant uniquement sur des jeux de données accessible publiquement per exemple LLaMA-13B surpasse GPT-3 (175B) sur plusieurs benchmarks [\[47\]](#). Comme GPT, ce sont des modèles décodeur capable de générer du texte et adaptation multi-tâches

## **2.7 Conclusion :**

Le traitement du langage naturel (NLP) et les grands modèles de langage (LLM) ont radicalement transformé la manière d'interagir avec le langage et l'information .Grâce aux avancées apportées par les différentes architectures transformer et un entraînement à grande échelle, ces modèles sont désormais capable de comprendre ,résumer ,traduire et générer du texte avec cohérence et pertinence .des taches autrefois complexes , comme la synthèse de documents scientifiques ou les réponses à des questions posées en langage naturel ,sont aujourd'hui réalisées avec une bonne efficacité .

Cette évolution repose sur l'utilisation de représentation vectorielle contextuelles, un apprentissage auto-supervisé massif, ainsi que des ressources de calcul. Les LLM ont permis l'émergence de nombreuses applications dans des domaines variés comme l'éducation, la recherche, la santé ou encore la justice.

En s'appuyant sur les principes clés du traitement du langage naturel et sur les capacités avancées des grands modèles de langage, il est désormais possible de concevoir des solutions concrètes répondant à des besoins ciblées. C'est dans cette optique que s'inscrit notre démarche. En utilisant les techniques de LLM et de NLP dans le développement d'un algorithme puissant et performant pour synthèse de document.

# Chapitre 3 Conception

## 3 Introduction:

Dans cette partie nous allons voir les différents diagramme UML du système : Digramme use case, classe, séquence et les scénarios

### 3.1 Diagramme cas d'utilisation

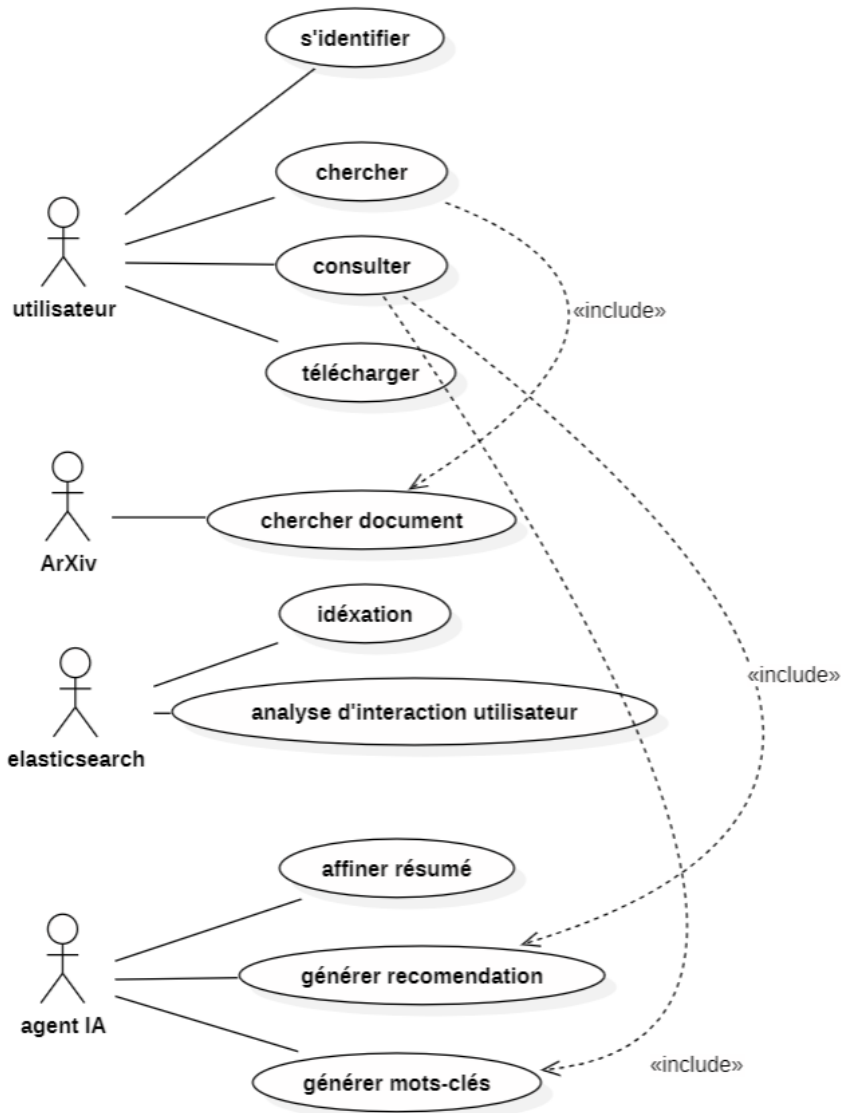


Figure 7. Diagramme Usecase

### 3.2 Diagramme de classe

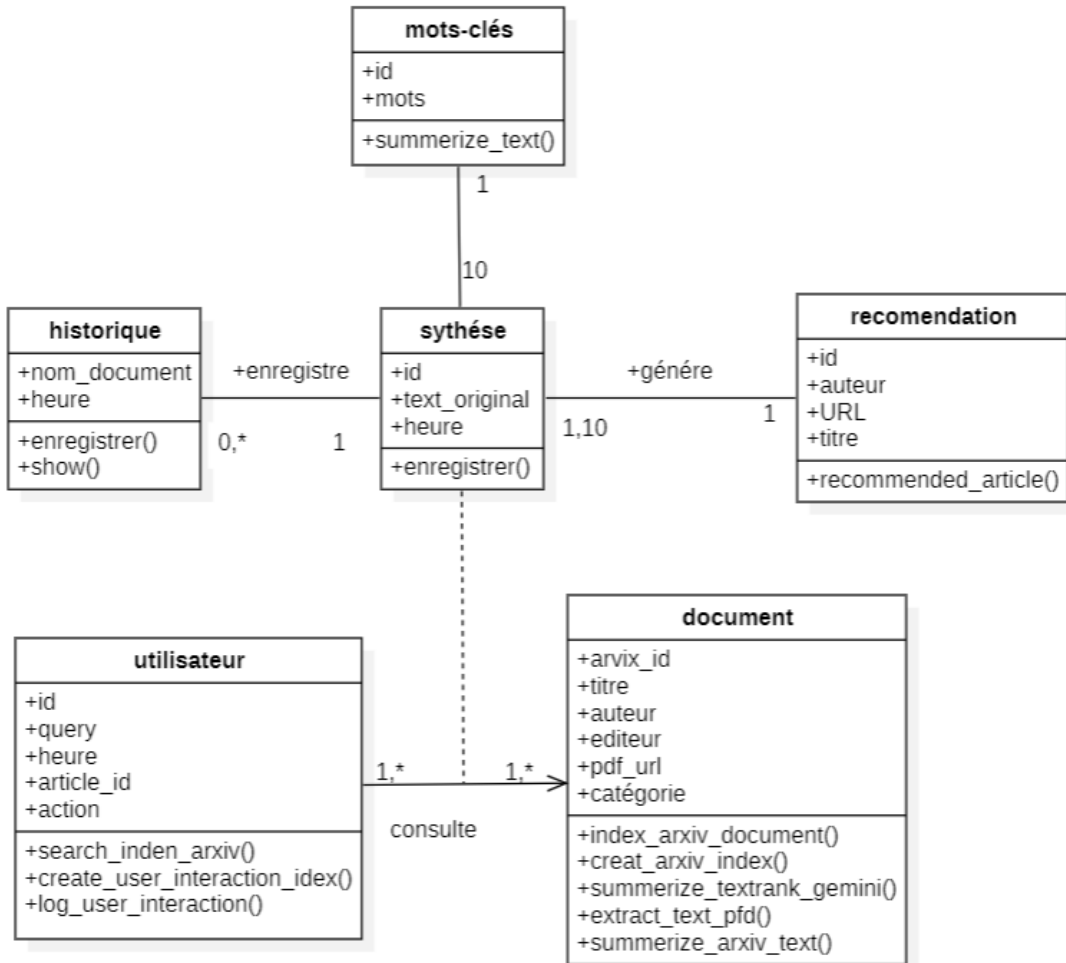


Figure 8. Diagramme De Classe

### 3.3 Diagramme de séquence :

Ceci est le diagramme du fonctionnement global de l'application

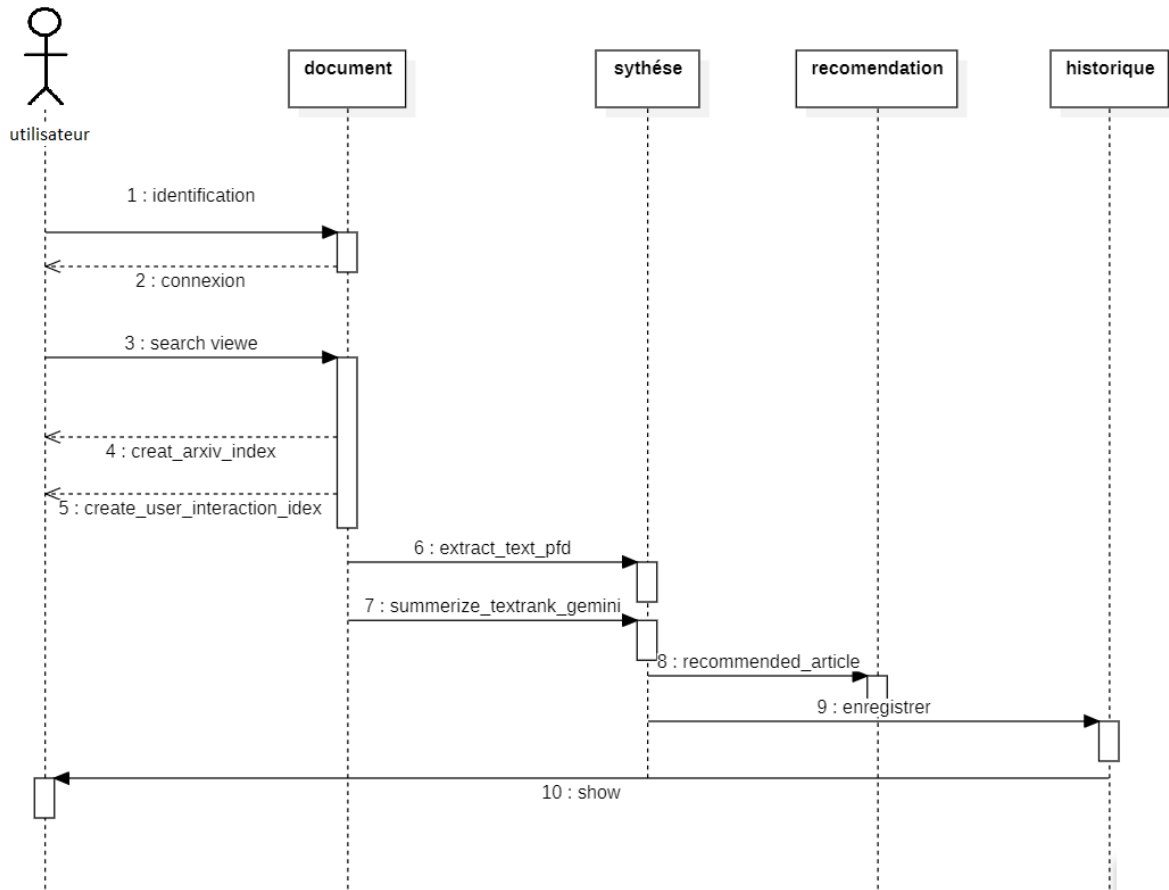


Figure 9. Diagramme séquence

### 3.4 Les scénarios :

Cette partie va expliquer le fonctionnement de certains scénarios de l'application

Scénario : rechercher un document

Acteur principal : utilisateur et système

Un champ de saisie lui est présenté pour entrer sa requête.

- L'utilisateur y saisit un thème de recherche
- La recherche est lancée après validation,
- Enregistre l'interaction (journalisation de la requête)
- Elle effectue la recherche via une arXiv
- Affiche des documents scientifiques

Cas d'utilisation : Sélectionner document

Acteur principal : Utilisateur et application

L'utilisateur consulte la liste des documents proposé

- Il clique sur le titre,
- Application génère comme sortie la synthèse du synthèse, les mots clés et les recommandation
- L'utilisateur click sur une recommandation
- Application donne un synthèse mots et plus de nouvelle recommandation

Cas d'utilisation : Collecte les documents

Acteur principal : Système

- L'application reçoit les données brutes des documents puis le système extrait le titre, url et PDF dès 100 documents les plus pertinents
- Elle les index en utilisant elasticsearch ensuite l'index est enregistré

Scénario : Générer synthèse

Acteur principal : Système

- Le système reçoit le texte complet du document
- Il utilise le textrank pour identifier les phrase clés ensuite
- Il utilise google Gemini pour condenser et affiner le synthèse avec extraction de mots clé

# Chapitre 4 Implémentation et présentation de l'application

## 4 Introduction :

### 4.1 Outils utilisés

#### 4.1.1 Django

Django est Framework web open source pour le développement rapide d'applications web performantes, sécurisées et évolutives en Python. Créé en 2005 par Adrian Holovaty et Simon Willison, le frame-work insiste sur le principe du développement rapide et du principe commun de « ne vous répétez pas » (DRY) et du développement propre [48]. Son architecture MTV (Model-Template-View), équivalent de MCV, consiste à séparer les données (modèles), la logique de traitement de ces données (vues) et la présentation (templates) [49].

Un des nombreux atouts de Django est le "batteries included", ce qui signifie qu'il propose plusieurs fonctionnalités prêtes à l'emploi [50]. On y retrouve notamment un panneau d'administration tout fait, un ORM puissant pour interfacer avec la base de données, un système de gestion d'utilisateurs complet, un moteur de template ou encore une sécurité rapprochée contre les attaques les plus connues comme les injections SQL, le cross-site scripting (XSS) ou le cross-site request forgery (CSRF).

De plus, Django est reconnu pour sa scalabilité. Grâce à son architecture modulaire et son support de base de données tel que PostgreSQL, MySQL ou SQLite, il est capable de gérer toutes sortes d'application, du blog personnel jusqu'à des sites très visités, comme c'est le cas d'Instagram [51]. Avec une compatibilité avec des technologies telles que Docker, ou le Cloud qui accélère encore davantage le déploiement et la gestion de montée en charge des applications [52].

La communauté Django est une des plus connues et actives des communautés du web. C'est grâce à une documentation très complète, de nombreux tutoriels et énormément de librairies tierces pour enrichir ses applications que l'on confirme cette notion. D'énormes événements, comme le fameux DjangoCon ou autres [1](#) [53], aident à montrer l'engouement dont bénéficie tous les acteurs de la communauté autour d'outil web qu'offre Django.

### **4.1.2 Elasticsearch**

Elasticsearch est un moteur de recherche et d'analyse distribué et open source écrit en Java. Basé sur la bibliothèque Lucene, il est capable de stocker, de rechercher et d'analyser de grandes quantités de données de manière distribuée. Elasticsearch est généralement utilisé comme moteur de recherche, de monitoring des logs, de métriques applicatives et de sécurité [54].

Conçu pour être scalable, elasticsearch peut gérer de grands volumes de données en les répartissant sur plusieurs nœuds dans le même cluster .chaque nœud peut héberger un ou plusieurs fragment d'index avec des mécanismes pour créer une réplique dans le but de la disponibilité et tolérance aux pannes .Les données sont stockées au format Json et accessibles via une API RESTFUL dans le but de faciliter l'intégration de plusieurs langage de programmation et système [55] .

La recherche de textes et des usages métiers variés, Elasticsearch permet de stocker, rechercher et analyser rapidement de grandes quantités de données. Il offre des fonctionnalités permettant de répondre à des cas d'usages reliés à la recherche et à l'analyse. Elasticsearch est très souvent utilisé conjointement avec la librairie de collecte, d'enrichissement et de partage de Logstash et de l'interface de visualisation de données Kibana (Stack ELK) [56].

### **4.1.3 API Gemini**

L'API Gemini de Google, désormais intégrée à Vertex AI, est la dernière technologie d'IA générative. Il permet aux développeurs de créer des applications capables à la fois de comprendre et de générer du contenu à partir de plusieurs modalités telles que le texte, les images, l'audio et même la vidéo. Cette approche multimodale signifie qu'ont disposent désormais d'une flexibilité sans précédent pour développer des applications avec des assistants virtuels, la génération de contenu créatif et des analyses de données avancées.

Les modèles Gemini comprenant Gemini 2.0 Flash et Gemini 2.5 Pro sont des exemples de modèles performants et capables de gérer des tâches complexes. Par exemple, le modèle Gemini 2.5 Pro est parfaitement adapté à la création d'interfaces web interactives, ce qui signifie qu'on peut facilement ajouter des effets pour des éléments de page qui se déplacent par exemple. De plus, Gemini Live API

permet de créer des flux de voix et vidéo en temps réel, pour des applications qui surpassent la simple conversation avec les utilisateurs [1](#) [57].

Pour permettre l'intégration de ces fonctionnalités dans les applications, Google propose la bibliothèque Gen AI, disponible dans plusieurs langages de programmation, comme Python, Go, Node.js et Java. Cette bibliothèque offre un moyen d'accéder aux modèles Gemini avec que ce soit à travers l'API ou à travers l'API Vertex AI, assurant une prise en main facile et une transition du prototypage à la production. Les développeurs peuvent aussi utiliser Vertex AI Studio, un environnement interactif où ils peuvent tester et affiner leurs requêtes ou encore intégrer l'API directement dans leurs applications mobiles ou web grâce aux SDK Firebase [58].

#### 4.1.4 key Bert

Key BERT est une bibliothèque open-source en Python qui a pour but d'extraire les mots-clés et phrases clés d'un texte grâce aux modèles de langage BERT (Bidirectional Encoder Représentations from Transformer) à la différence des méthodes statistiques comme TF-IDF, RAKE ou YAKE, Key BERT s'appuie sur les représentations sémantiques apprises par BERT pour identifier les termes les plus importants d'un document [59].

Il utilise une approche simple et efficace. Dans un premier temps il prend un document et d'extraire des vecteurs embeddings grâce à BERT. Ensuite, il fera la même opération pour les n-grammes (un ensemble contigu de n éléments pris consécutivement dans une série) qui peuvent également être extraits du texte. Ces vecteurs pourront être ensuite confrontés à celui du document grâce à une mesure de similarité cosinus. La grandeur de ce nombre comprend un nombre d'avantages à mesurer la similarité entre deux vecteurs.

La première concerne la proportionnalité de deux vecteurs et leurs dissimilarités à l'origine La seconde, du fait qu'un produit scalaire entre les vecteurs vaut 1 alors ils sont égaux (dira-t-on qu'ils « pointent dans la même direction »). La dernière qualité de la mesure de similarité cosinus, est que par construction, elle ne peut pas être plus grande que 1 La mesure de similarité cosinus est mais pas toujours interprétable, elle est utilisée dans les représentations des documents car elle est rapide et nécessite moins de mémoire que d'autres algorithmes métriques [60].

#### 4.1.5 NLTK

C'est une bibliothèque open source en Python conçue pour le traitement automatique de langage naturel (TALN). Développée initialement par Steven Bird et Edward Loper à l'université de Pennsylvanie [61]. Elle est largement utilisée dans les domaines linguistiques, de l'intelligence artificielle et l'apprentissage automatique.

Cette bibliothèque fournit une suite complète d'outils pour diverses tâches de traitement de langage comme la tokenisation, le stemming, lemmatisation, l'étiquetage, l'analyse syntaxique et la reconnaissance d'entité nommées (comme les noms pays entreprise ect ) [62].

En plus de cela, NLTK offre un accès facile à plus de 50 corpus linguistique et ressources lexicales, comme WordNet, dans le but de faciliter l'analyse et la manipulation de données textuelle. La bibliothèque est également accompagnée de démonstration graphique de jeux de données échantillons de tutoriels et d'une documentation complète [63].

#### 4.1.6 Spacy

SpaCy est une bibliothèque de traitement du langage naturel open-source en Python, créée par l'équipe d'Explosion AI. Automatisée pour le déploiement, elle est rapide et précise les applications de traitement de texte en grandeur nature.

Quelques-unes des fonctionnalités proposées par spaCy sont la tokenization (découpage du texte en mots), le part-of-speech tagging (étiquetage morpho-syntaxique), le dependency parsing (analyse syntaxique), la lemmatisation, la Named Entity Recognition (entités nommées) et le text classification. En outre, spaCy propose des fonctionnalités avancées telles que la phrase boundary detection et le rule-based matching.

SpaCy est conçue pour fournir une performance élevée avec des ressources et une vitesse de traitement minimales. Elle est mise en œuvre par le biais d'une bibliothèque en Python et Cython. Elle intègre des modèles évolués, comme des réseaux de neurones convolutifs, et des modèles de langage pré-entraînés, comme BERT [wikipedia](#) [64] qui permettent de prédire et de catégoriser les parties d'un texte sans difficulté. Grâce à cela, on peut en quelques lignes découper un texte en mots

et les lemmatiser, trouver les noms propres et les groupes nominaux qui y sont mentionnés, extraire les relations ou le sujet des verbes, c'est l'outil de NLP q pour le traitement de texte dans tous ses états et l'avantage de spaCy est qu'il prend en charge plus de 70 langues.

## 4.2 Base de données utilisées :

L'api arvix est une interface RESTful open source qui permet un accès programmatique au metadonnées et au contenu scientifique hébergées sur [65] dans divers domaines scientifiques : La biologie, la finance, les statistiques, les mathématiques, l'informatique ect ect. Grâce à cette API, on peut filtrer et récupérer des articles en utilisant des requêtes HTTP, sans nécessiter la clé de l'api. Les résultats sont généralement retournés au format XML, facilitant leur intégration dans divers application a [66].

## 4.3 Interface homme-machine :

L'interface se compose de plusieurs éléments, le principal est celui de la recherche, ou l'utilisateur saisit sont sujet les résultats s'affichent ensuite dans la section recent searches . Lorsque l'utilisateur clique sur un document, une synthèse mots clés liens de téléchargement et recommandation seront afficher dans seach results.

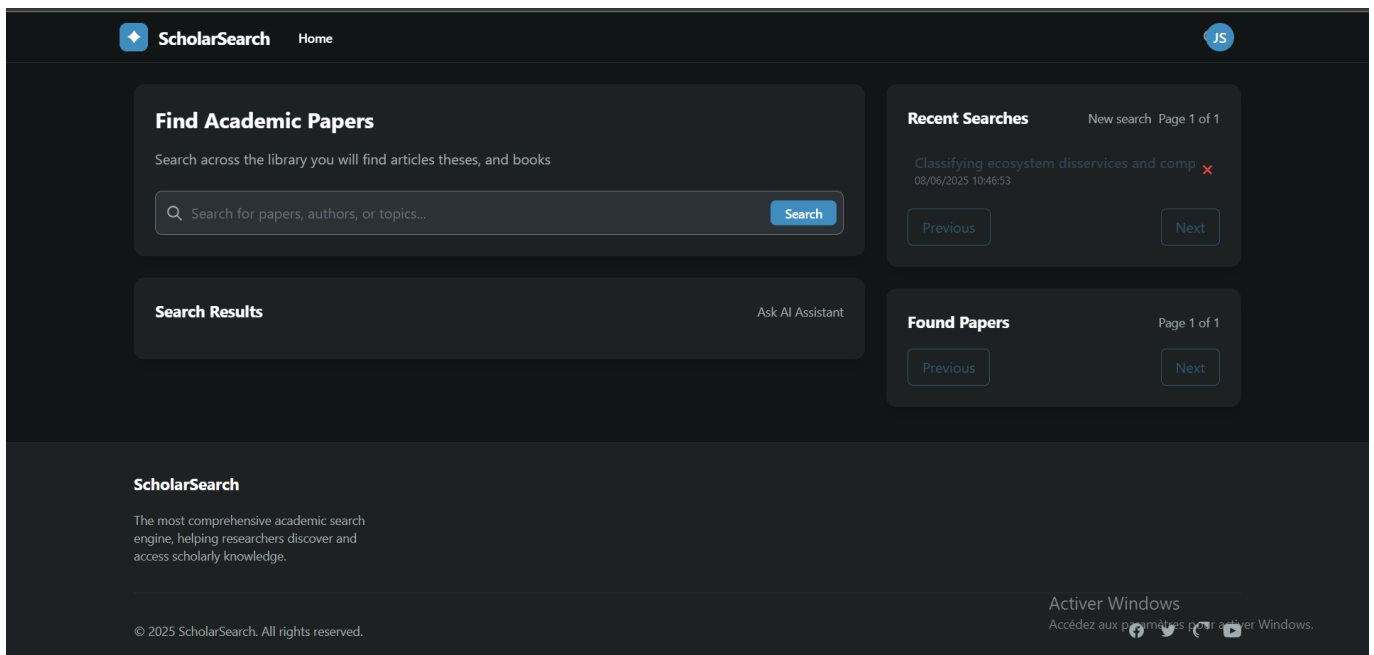


Figure 10. Interface principale

La barre de recherche est utilisée par l'utilisateur afin de mettre son sujet de recherche une fois que c'est fait, les résultats seront affichés dans found papers .

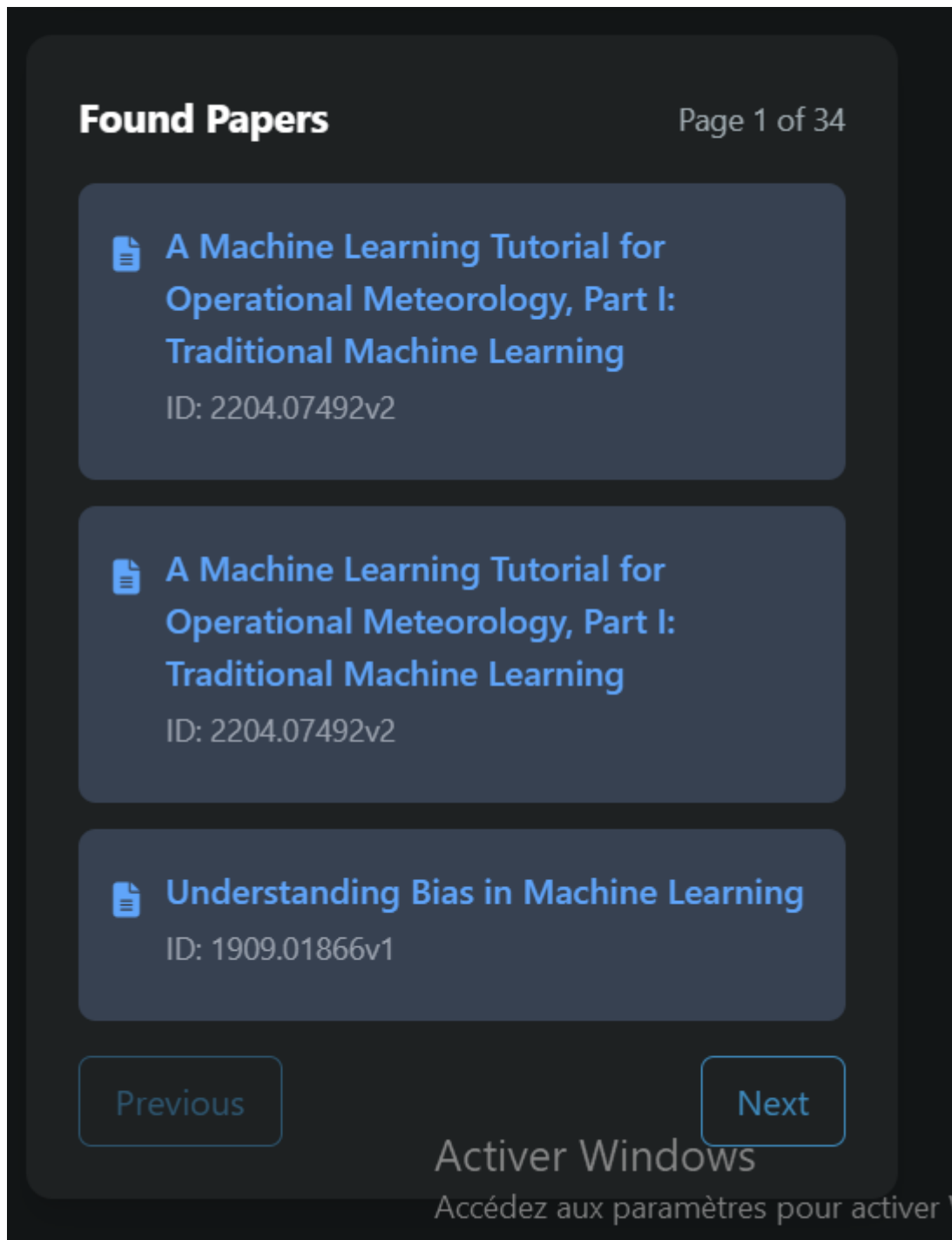


Figure 11. Found papers

Les résultats sont dans des pages ou dans chaque page affiche trois résultats puis lorsque l'utilisateur click sur un résultat l'application va enregistrer l'article dans la base de données et sera affiché dans Recent Searches ensuite l'application va générer une synthèse, les mots clés, liens de téléchargement et les recommandations

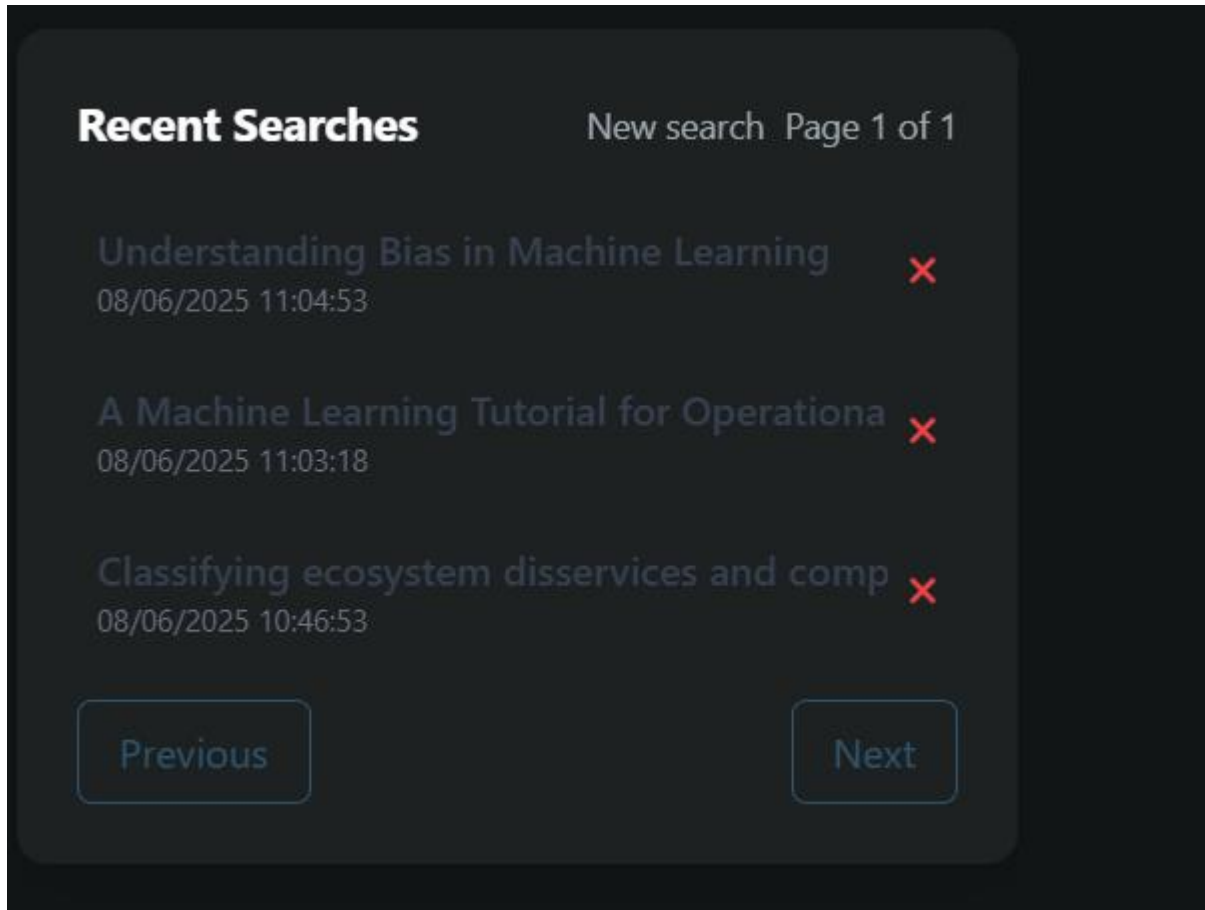


Figure 12 Historique



Hi! I'm your AI research assistant. Ask me anything about the papers you've found or request a summary of any document.

computing science



Requête arXiv utilisée : "computer science"

Veuillez résumer l'article "Computing of Applied Digital Ecosystems" (ID: 0910.0674v1)



Traitement de l'article avec ID : 0910.0674v1...



Résumé de l'article 0910.0674v1 :

This research explores digital ecosystems, inspired by biological ones, to create sustainable regional development through SMEs. It leverages self-organizing principles, measured through simulations of evolving agent populations responding to user requests. The research uses multi-agent systems (MAS) and Service-Oriented Architectures (SOAs), drawing parallels with biological agent migration and evolution. However, direct emulation isn't possible due to fundamental differences in operation. SOAs, evolving from modular programming and distributed computing, facilitate the assembly of applications from loosely coupled services, potentially reducing development costs and promoting standardization. Distributed evolutionary algorithms further enhance application evolution, optimizing software service combinations. Finally, the research experimentally validates the adaptability of the digital ecosystem to various user behaviors, demonstrating its potential to dynamically respond to user needs.



Mots-clés :

ecosystems   ecosys   biosphere   ecology   technologies   environmental  
environmentally   technological   microsystems   technology



Télécharger le document :

[Cliquez ici pour télécharger le PDF](#)



Recommandations :

**Biology of Applied Digital Ecosystems**

ID: 0712.4153v2

**Digital Ecosystems: Ecosystem-Oriented Architectures**

ID: 1112.0204v1

**Computing of Applied Digital Ecosystems**

ID: 0910.0674v1

**The Computing of Digital Ecosystems**

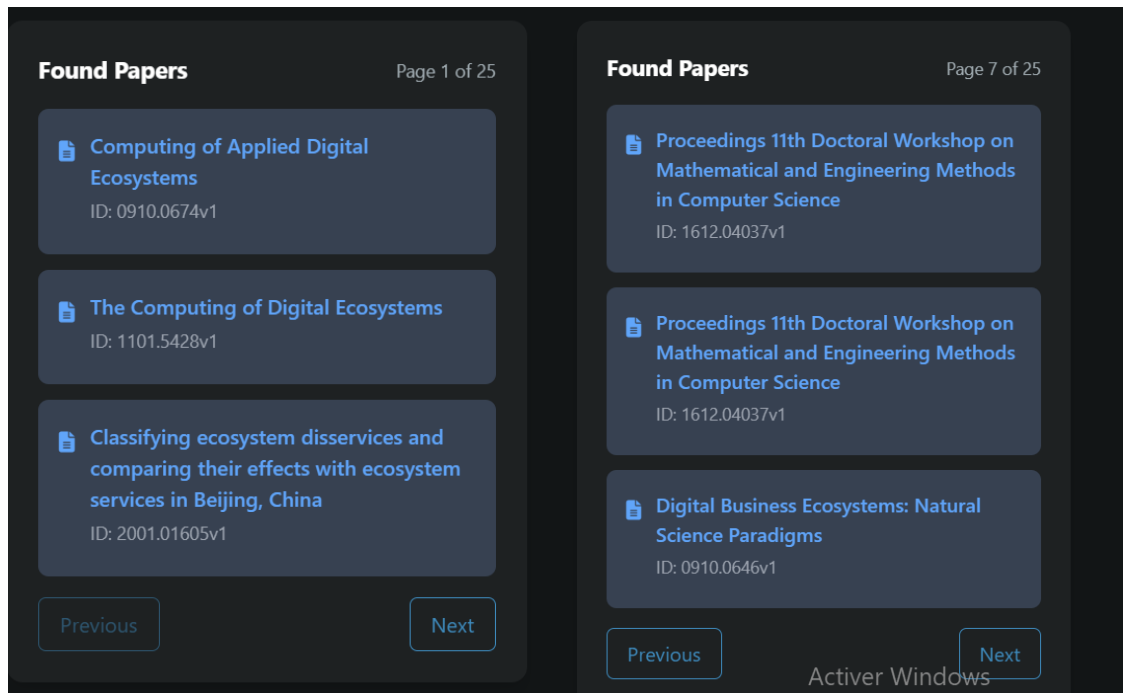
ID: 1101.5428v1

**Digital Ecosystems: Evolving Service-Oriented Architectures**

Figure 13. Chat Contraire Sythese ,Mots Cle,Telechargement,Recomendation

## 4.4 Étude de cas :

L'application est basée sur la proactivité de façon à ce que les divers interactions et recherche de l'utilisateur sont enregistrés. On a fait des recherches sur des thématiques telle que l'écologie et l'informatique maintenant l'application donne comme résultat de recherche des articles depuis la bibliothèque aRxiv entre l'écologie et l'informatique.



## 4.5 Analyses :

L'application « scholar search » est une solution pour accélérer et améliorer l'expérience de recherche de document scientifique à travers un système de recommandation et de synthèse , en combinant des capacité de recherche avec des fonctionnalité de AI à travers l'API Gemini pour améliorer l'expérience utilisateur .il y a une intégration de l'API arXiv pour la récupération de document scientifique , avec l'aide de Elasticsearch pour une indexation et une recherche efficaces des documents scientifiques ,la fonctionnalité de synthèse est basé sur une approche hybride ou l'algorithme TextRank extrait initialement les phrases clés du document ,puis l'API gemini affine et condense ce synthèse ,garantissant une pertinence , logique et précision ,l'extraction de mots-clés enrichit la compréhension rapide de contenu et aussi elle est utilisé pour les futures recommandation ,les synthèses sont générer dans la langue original du document .

la fonctionnalité principal est l'aspect proactive de recommandation qui se manifeste principalement avec que l'utilisateur interagie avec un document , par exemple en sélectionnant pour une synthèse le système fournir des recommandations en même temps que le synthèse d'un article .elle utilise les interactions avec les utilisateurs ainsi que sur la similarité de contenu exploitant TF-IDF (term frequency -inverse document frequency) et la similarité cosinus .l'application extrait le synthèse de l'article et le traite comme document de référence .elle convertie ensuite ce synthèse .ainsi que tous les autres synthèses de l'index .en vecteur numérique en utilisant TF-IDF, qui quantifie l'importance de chaque mots .la similarité cosinus est ensuite calculé entre le vecteur de l'article de référence et ceux de tous les autres articles ,mesurant ainsi leur proximité sémantique .les articles dont le score de similarité cosinus est le plus élevé sont alors sélectionnés et présente comme des recommandations offrant des lectures à l'utilisateur

## 4.6 Conclusion :

En somme , « scholar search » s'inscrit dans une démarche d'automatisation et de rechercher intelligente de document scientifique .grâce à l'intégration elasticsearch pour une indexation efficace .Keybert avec Spacy pour l'extraction de mots-clés , L'API Gemini pour raffiner le synthèse générer par le TextRank ainsi que recommander des documents scientifiques dans le but d'avoir des résultats pertinents , et de la base arXiv pour l'accès à des publications scientifiques actualisés , via le framework Django ,pour offrir une solution fluide et simplifier l'accès a l'information.

## Conclusion Générale

Le travail réalisé avec cette application « scholar Search » représente une avancée dans la simplification et l'enrichissement à l'accès à la littérature scientifique. Le système apporte une valeur ajoutée considérable en combinant habilement, une recherche performante via Elasticsearch, une capacité de synthèse fiable via TextRank et l'API gemini ,ainsi qu'un mécanisme de recommandation proactif basé sur la similarité du contenu TF-IDF et cosinus.

L'interface utilisateur fluide a des fonctions comme la mise à jour en temps réel de l'historique des recherches ,le téléchargement direct des PDF et la gestion des clics sur les recommandation sans redirection ,dans le but d'accélérer le processus de découvert de document scientifique , en fournissant des synthèses précise dans la langue original du document et en suggérant des lectures complémentaires pertinentes , l'application réduit le temps et l'effort nécessaires pour assimiler de nouvelles connaissances , devenant ainsi un outil utile pour les chercheurs et les étudiants .

Cependant, le travail accompli ouvre également la voie à de nouvelles perspectives, tout en révélant certaines limites qui réside, dans la dépendance aux APIs externes, notamment pour le synthèse via Gemini et l'accès à arXiv, ce qui pourrait engendrer des coûts ou des restrictions en cas de volume d'utilisation très élevé ou de changement dans la politique d'API, l'idéal est d'utiliser des données réelles issue de véritable bibliothèques universitaires.

Aussi l'application pourrait avoir des fonctionnalités collaboratives entre superviseurs et étudiants ou même entre chercheurs pour rendre la communication facile et ludique, même ajouter des formats additionnels de document comme EPUB ou DOCS pour une meilleure polyvalence ,et même ajouter un modèle de langue arabe pour des pays comme l'Algérie car même aujourd'hui la langue arabe reste un défi à intégrer dans ce genre d'application.

L'application scholar search se positionne comme un outil essentiel, enrichissant et simplifiant l'accès à la connaissance académique par un mixe de la recherche avancée, de la synthèse par IA et algorithme de synthèse et une recommandation proactive, rendant ainsi la recherche une activité simple.

# Webographies

- [1 Y. J. a. R. W. Jan Boehmer, «rickwash.com,» [En ligne]. Available: <https://rickwash.com/papers/recsys-encyclopedia.pdf#:~:text=The%20invention%20of%20modern%20recommender,The%20Xerox%20Palo%20Alto>.
- [2 Y. J. a. R. W. Jan Boehmer, «rickwash.com,» [En ligne]. Available: <https://rickwash.com/papers/recsys-encyclopedia.pdf#:~:text=on%20item%20similarities%2C%20while%20collaborative,relevant%20items%20in%20the%20large>.
- [3 B. S. a. J. Y. Greg Linden, «cs.umd.edu,» [En ligne]. Available: <https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf#:~:text=form%20of%20targeted%20marketing%20by,Unlike%20other>.
- [4 Y. J. a. R. W. Jan Boehmer, «rickwash.com,» [En ligne]. Available: <https://rickwash.com/papers/recsys-encyclopedia.pdf#:~:text=users%20to%20serve%2C%20for%20many,Their%20goal%20was%20to>.
- [5 Y. J. a. R. W. Jan Boehmer, «rickwash.com,» [En ligne]. Available: <https://rickwash.com/papers/recsys-encyclopedia.pdf#:~:text=users%20to%20serve%2C%20for%20many,Their%20goal%20was%20to>.
- [6 E. Negre, «openscience.fropenscience.fr,» [En ligne]. Available: [https://www.openscience.fr/IMG/pdf/iste\\_roisi20v1n4\\_2.pdf#:~:text=pr%C3%A8s%20de%20l'E2%80%99utilisateur,dans%20le%20processus%20de%20recommandation](https://www.openscience.fr/IMG/pdf/iste_roisi20v1n4_2.pdf#:~:text=pr%C3%A8s%20de%20l'E2%80%99utilisateur,dans%20le%20processus%20de%20recommandation).
- [7 E. Negre, «openscience.fropenscience.fr,» [En ligne]. Available: [https://www.openscience.fr/IMG/pdf/iste\\_roisi20v1n4\\_2.pdf#:~:text=pr%C3%A8s%20de%20l'E2%80%99utilisateur,dans%20le%20processus%20de%20recommandation](https://www.openscience.fr/IMG/pdf/iste_roisi20v1n4_2.pdf#:~:text=pr%C3%A8s%20de%20l'E2%80%99utilisateur,dans%20le%20processus%20de%20recommandation).
- [8 C. B. & P. d. P. Pauline VAUTROT, «management-datascience.org,» [En ligne]. Available: <https://management-datascience.org/articles/3775/#:~:text=Les%20travaux%20en%20marketing%20soulignent,peut%20se%20sentir%20C2%AB%C2%A0manipul%C3%A9%20C2%BB%20ou>.
- [9 M. J. & D. Jannach, «web-ainf.aau.at,» [En ligne]. Available: [https://web-ainf.aau.at/pub/jannach/files/Journal\\_TiiS\\_2017.pdf#:~:text=Aljukhadar%20et%20al.%20,choice%20quality%20in%20complex%20situations](https://web-ainf.aau.at/pub/jannach/files/Journal_TiiS_2017.pdf#:~:text=Aljukhadar%20et%20al.%20,choice%20quality%20in%20complex%20situations).
- [1 «comp.hkbu.edu.hk,» [En ligne]. Available: <https://www.comp.hkbu.edu.hk/~lichen/download/p240-chen.pdf#:~:text=5,and%20novelty%20can%20directly%20affect>.
- [1 C. B. & P. d. P. Pauline VAUTROT, «management-datascience.org,» [En ligne]. Available: <https://management-datascience.org/articles/3775/#:~:text=L'E2%80%99objectif%20de%20la%20recherche%20est,plus%20efficace%20qu'E2%80%99il%20est%20contr%C3%B4lable>.
- [1 B. S. a. J. Y. Greg Linden, «cs.umd.edu,» [En ligne]. Available: <https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf#:~:text=In%20the%20future%2C%20we%20expect,other%20forms%20of%20customer%20communication>.
- [1 B. S. a. J. Y. Greg Linden, «cs.umd.edu,» [En ligne]. Available: <https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf#:~:text=In%20the%20future%2C%20we%20expect,other%20forms%20of%20customer%20communication>.
- [1 A. Clark, «archive.news.ufl.edu,» [En ligne]. Available: <https://archive.news.ufl.edu/articles/2018/09/how-helpful-are-product-recommendations-really-1.html#:~:text=By%20evaluating%20more%20than%20two,you%20were%20initially%20looking%20at>.
- [1 A. Clark, «archive.news.ufl.edu,» [En ligne]. Available: <https://archive.news.ufl.edu/articles/2018/09/how-helpful-are-product-recommendations-really-5>.



- [3 N. S. P. U. J. N. G. K. P. Ashish Vaswani, «papers.neurips.cc,» [En ligne]. Available:  
1] [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf#:~:text=The%20dominant%20sequence%20transduction%20models,dispensing%20with%20recurrence%20and%20convolutions.](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf#:~:text=The%20dominant%20sequence%20transduction%20models,dispensing%20with%20recurrence%20and%20convolutions.)
- [3 N. S. P. U. J. N. G. K. P. Ashish Vaswani, «papers.neurips.cc,» [En ligne]. Available:  
2] [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf#:~:text=The%20dominant%20sequence%20transduction%20models,dispensing%20with%20recurrence%20and%20convolutions.](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf#:~:text=The%20dominant%20sequence%20transduction%20models,dispensing%20with%20recurrence%20and%20convolutions.)
- [3 N. S. P. U. J. N. G. K. P. Ashish Vaswani, «papers.neurips.cc,» [En ligne]. Available:  
3] [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf#:~:text=The%20dominant%20sequence%20transduction%20models,dispensing%20with%20recurrence%20and%20convolutions.](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf#:~:text=The%20dominant%20sequence%20transduction%20models,dispensing%20with%20recurrence%20and%20convolutions.)
- [3 N. S. P. U. J. N. G. K. P. Ashish Vaswani, «papers.neurips.cc,» [En ligne]. Available:  
4] [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf#:~:text=The%20dominant%20sequence%20transduction%20models,dispensing%20with%20recurrence%20and%20convolutions.](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf#:~:text=The%20dominant%20sequence%20transduction%20models,dispensing%20with%20recurrence%20and%20convolutions.)
- [3 J. W. ., R. C. ., L. ., D. A. ., Alec Radford, «cdn.openai.com,» [En ligne]. Available:  
5] [https://cdn.openai.com/better-language-models/language\\_models\\_are\\_unsupervised\\_multitask\\_learners.pdf#:~:text=on%20a%20new%20dataset%20of,the%20language%20model%20is%20essential.](https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf#:~:text=on%20a%20new%20dataset%20of,the%20language%20model%20is%20essential.)
- [3 «ovhcloud.com,» [En ligne]. Available: <https://www.ovhcloud.com/fr/learn/what-is-large-language-model/#:~:text=%C3%80%20part%201%27apprentissage%20des%20langues,et%20la%20g%C3%A9n%C3%A9ration%20de%20textes.>
- [3 N. S. P. U. J. N. G. K. P. Ashish Vaswani, «neurips.cc,» [En ligne]. Available:  
7] [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf#:~:text=less%20time%20to%20train,art%20BLEU%20score%20of%2041.0.](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf#:~:text=less%20time%20to%20train,art%20BLEU%20score%20of%2041.0.)
- [3 M.-W. C. K. L. K. T. Jacob Devlin, «arxiv.org,» [En ligne]. Available:  
8] <https://arxiv.org/abs/1810.04805#:~:text=both%20left%20and%20right%20context,MultiNLI%20accuracy%20to.>
- [3 K. C. S. Seema Aswani, «researchgate.net,» [En ligne]. Available:  
9] [https://www.researchgate.net/publication/378941249\\_Automatic\\_text\\_summarization\\_of\\_scientific\\_articles\\_using\\_transformers-A\\_brief\\_review.](https://www.researchgate.net/publication/378941249_Automatic_text_summarization_of_scientific_articles_using_transformers-A_brief_review.)
- [4 K. C. S. Seema Aswani, «researchgate.net,» [En ligne]. Available:  
0] [https://www.researchgate.net/publication/378941249\\_Automatic\\_text\\_summarization\\_of\\_scientific\\_articles\\_using\\_transformers-A\\_brief\\_review.](https://www.researchgate.net/publication/378941249_Automatic_text_summarization_of_scientific_articles_using_transformers-A_brief_review.)
- [4 B. M. N. R. M. S. J. K. P. D. A. N. P. S. G. S. A. A. S. A. A. H.-V. G. K. T. H. R. C. A. R. D. Tom B.  
1] Brown, «arxiv.org,» [En ligne]. Available:  
<https://arxiv.org/abs/2005.14165#:~:text=well%20as%20some%20datasets%20where,societal%20impact%20of%20this%20finding.>
- [4 «wikipedia,» [En ligne]. Available:  
2] [https://en.wikipedia.org/wiki/BERT\\_\(language\\_model\)#:~:text=BERT%20was%20originally%20implemented%20in,The%20weights%20were%20released%20on.](https://en.wikipedia.org/wiki/BERT_(language_model)#:~:text=BERT%20was%20originally%20implemented%20in,The%20weights%20were%20released%20on.)
- [4 M.-W. C. K. L. K. T. Jacob Devlin, «arxiv.org,» [En ligne]. Available:  
3] <https://arxiv.org/abs/1810.04805#:~:text=both%20left%20and%20right%20context,MultiNLI%20accuracy%20to.>
- [4 J. W. ., R. C. ., L. ., D. A. ., Alec Radford, «cdn.openai.com,» [En ligne]. Available:  
4] [https://cdn.openai.com/better-language-models/language\\_models\\_are\\_unsupervised\\_multitask\\_learners.pdf#:~:text=on%20a%20new%20dataset%20of,the%20language%20model%20is%20essential.](https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf#:~:text=on%20a%20new%20dataset%20of,the%20language%20model%20is%20essential.)
- [4 J. W. ., R. C. ., L. ., D. A. ., Alec Radford, «cdn.openai.com,» [En ligne]. Available:  
5] [https://cdn.openai.com/better-language-models/language\\_models\\_are\\_unsupervised\\_multitask\\_learners.pdf#:~:text=on%20a%20new%20dataset%20of,the%20language%20model%20is%20essential.](https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf#:~:text=on%20a%20new%20dataset%20of,the%20language%20model%20is%20essential.)

models/language\_models\_are\_unsupervised\_multitask\_learners.pdf#:~:text=on%20a%20new%20dataset%20of,the%20language%20model%20is%20essential.

[4 T. L. G. I. X. M. M.-A. L. T. L. B. R. N. G. E. H. F. A. A. R. A. J. E. G. G. L. Hugo Touvron,  
6] «arxiv.org,» [En ligne]. Available:

<https://arxiv.org/abs/2302.13971#:~:text=models%20to%20the%20research%20community>.

[4 T. L. G. I. X. M. M.-A. L. T. L. B. R. N. G. E. H. F. A. A. R. A. J. E. G. G. L. Hugo Touvron,  
7] «arxiv.org,» [En ligne]. Available:

<https://arxiv.org/abs/2302.13971#:~:text=models%20to%20the%20research%20community>.

[4 «wikipedia,» [En ligne]. Available: [https://en.wikipedia.org/wiki/Django\\_%28web\\_framework%29?](https://en.wikipedia.org/wiki/Django_%28web_framework%29?)  
8]

[4 «vpsjungle,» [En ligne]. Available: [https://www.vpsjungle.in/tutorials/what-is-django-features-](https://www.vpsjungle.in/tutorials/what-is-django-features-9)  
9] [advantages/](https://www.vpsjungle.in/tutorials/what-is-django-features-9).

[5 «blog.jetbrains,» [En ligne]. Available: [https://blog.jetbrains.com/pycharm/2023/11/what-is-the-django-](https://blog.jetbrains.com/pycharm/2023/11/what-is-the-django-0)  
0] [web-framework/](https://blog.jetbrains.com/pycharm/2023/11/what-is-the-django-0).

[5 «developer.mozilla,» [En ligne]. Available: [https://developer.mozilla.org/en-](https://developer.mozilla.org/en-1)  
1] [US/docs/Learn\\_web\\_development/Extensions/Server-side/Django/Introduction](https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/Django/Introduction).

[5 «gtechsol,» [En ligne]. Available: <https://gtechsol.com.au/key-features-of-the-django-framework/>.  
2]

[5 «gtechsol,» [En ligne]. Available: <https://gtechsol.com.au/key-features-of-the-django-framework/>.  
3]

[5 «sociumweb,» [En ligne]. Available: <https://www.sociumweb.com/blog/elasticsearch>.  
4]

[5 «wikipedia,» [En ligne]. Available: <https://fr.wikipedia.org/wiki/Elasticsearch?>  
5]

[5 «wikipedia,» [En ligne]. Available: <https://en.wikipedia.org/wiki/Elasticsearch?>  
6]

[5 «androidcentral,» [En ligne]. Available: [https://www.androidcentral.com/news/live/google-i-o-2025-live-](https://www.androidcentral.com/news/live/google-i-o-2025-live-7)  
7] [blog?](https://www.androidcentral.com/news/live/google-i-o-2025-live-7).

[5 «cloud.google,» [En ligne]. Available: <https://cloud.google.com/vertex-ai/generative-ai/docs?hl=fr>.  
8]

[5 «keybert,» [En ligne]. Available: <https://www.geeksforgeeks.org/keyword-extraction-methods-in-nlp/>.  
9]

[6 «readmedium,» [En ligne]. Available: [https://readmedium.com/how-to-extract-relevant-keywords-with-](https://readmedium.com/how-to-extract-relevant-keywords-with-0)  
0] [keybert-6e7b3cf889ae](https://readmedium.com/how-to-extract-relevant-keywords-with-0).

[6 «tutsplus,» [En ligne]. Available: [https://code.tutsplus.com/fr/introducing-the-natural-language-toolkit-](https://code.tutsplus.com/fr/introducing-the-natural-language-toolkit-1)  
1] [nltk--cms-28620t?](https://code.tutsplus.com/fr/introducing-the-natural-language-toolkit-1).

[6 «iana-data,» [En ligne]. Available: <https://www.iana-data.org/fondamentaux/424-cest-quoi-le-nltk.html>.  
2]

[6 «NLTK,» [En ligne]. Available: <https://www.nltk.org/>.  
3]

[6 «wikipedia,» [En ligne]. Available: <https://es.wikipedia.org/wiki/SpaCy?>.  
4]

[6 «arxiv,» [En ligne]. Available: <https://info.arxiv.org/help/api/user-manual.html>.  
5]

[6 «arxiv,» [En ligne]. Available: <https://info.arxiv.org/help/api/user-manual.html>.  
6]

[6 [En ligne].  
7]

[6 Y. J. a. R. W. Jan Boehmer, «rickwash.com,» [En ligne]. Available: <https://rickwash.com/papers/recsys->

