REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ DES MATHEMATIQUES ET DE l'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et télécommunications

Par :

MESSAOUDI Fatima Zohra Sarra
MOSTEFAOUI Imen

Sur le thème

WSC_ACTION : Une approche Deep Learning pour un meilleur processus de découverte des services web sémantique

---

Soutenu publiquement le 29/06/ 2022 à Tiaret devant le jury composé de :

Mr MOSTEFAOUI Sid Ahmed	Grade	Université MCB	Président
Mr MEGHAZI Hadj Madani	Grade	Université MAA	Encadrant
Mr AID Lahcene	Grade	Université MCB	Examinateur

2021-2022

Remerciement

Avant tout, nous remercions Dieu Tout-Puissant de nous donner du courage et de nous guider pour pouvoir mener à bien cet humble travail.

A notre promoteur Mr MEGHAZI Hadj Madani

Nous adressons nos sincères remerciements et notre appréciation, pour tous les conseils et orientations qu'il nous a donnés, afin de s'assurer que cette recherche soit complétée et présentée de la manière requise.

A Mr. MOSTEFAOUI Sid Ahmed

Nous sommes ravis de vous avoir dans le jury de ce travail, nous tenons à vous exprimer notre profonde gratitude pour avoir bien voulu accepter de présider le jury de cette thèse. Veuillez compter sur notre gratitude et notre considération respectueuse.

A Mr. AID Lahcene

Nous sommes reconnaissantes de l'honneur que vous nous en acceptiez d'examiner notre modeste travail, Veuillez trouver dans le travail l'expression de notre attention, et le témoignage de profonde et sincère considération.

Un grand Merci aux enseignants ainsi que l'administration de faculté informatique qui ont veillé sur notre formation et suivi durant tout le cursus d'étude

Enfin, nous tenons à remercier tous ceux qui ont contribué par leurs conseils ou leurs encouragements à ce travail.

Dédicace

Mes parents pour sa gentillesse, leur savoir-faire, leur disponibilité, leur tendresse, leurs encouragements et son soutien moral dans les moments les plus difficiles. Je vous dois mon éducation, ma réussite dans ma vie et mes études. Merci très chère parents je suis très reconnaissante.

Mes sœurs très chères, Sarah, Halima et Fatima, pour leur compréhension, leur soutien affectif et leur disponibilité dans les moments les plus difficiles.

Un hommage particulier aux personnes les plus chères à mon cœur, Mostefa, Tonton tayeb, djihed et fatma pour leurs encouragements constants, leurs conseils que j'appliquerai toujours, par amour sincère, pour leur bienveillance et je remercie surtout ma tente Amina qui est loin mais proche du cœur, qui essaie toujours de trouver un moyen de se débarrasser de mon stress, merci pour tous les fous rires, que Dieu vous protège tous les deux pour moi.

Tous les membres de ma grande famille pour leur encouragement et leur patience et particulièrement

Oh Dieu, aie pitié d'une âme qui est montée jusqu'à toi, et il n'y a rien entre nous et elle sauf la supplication. Que Dieu ait pitié de ma grand-mère et lui pardonne et la regarde avec les yeux de ta bonté et de ta générosité, Seigneur de Pardonne-lui et rassemble-moi avec elle dans ton paradis.

Au propriétaire de l'excellence et des idées brillantes, les salutations les plus pures, les plus belles, les plus belles et les plus gentilles, je vous les envoie avec toute l'affection, l'amour et la sincérité, les lettres ne peuvent pas écrire ce que mon cœur porte d'appréciation et de respect, et à décris ce qui a rempli mon cœur de louanges et d'admiration, combien il est beau d'être un être humain Une bougie qui éclaire les chemins des perplexes.

A mon chère binôme Sarra avec laquelle J'ai vécu des souvenirs inoubliables durant les cinq ans d'études à l'université. Avec elle j'ai partagé le stress et la pression des études, la saveur de la réussite et beaucoup de fou rire.

Un mot de remerciement et de gratitude. À ceux qui ont bon cœur, à ceux qui ont un sourire unique à mes amis (es) proches Chaima Kheira Nessrine Amel Karima et Fatima

IMEN

Dédicace

Mes très chers parents qui m'ont comblée par leur amour, leur sacrifice et leurs précieux conseils. Des personnes qui m'ont soutenu moralement et financièrement jusqu'à ce jour, qui m'ont encouragé tout au long de mes études. Que ce travail soit pour eux un humble témoignage de ma profonde gratitude. Qu'Allah les préserve et leur accorde santé, bonheur et longue vie.

Mon cher frère unique, Mourad, est mon exemple, sa générosité et son soutien moral qui ont été pour moi une source de courage et de confiance, et mes adorables Amina, Cherifa qui m'ont soutenue tout au long de mes études et ont toujours été à mes côtés.

Une dédicace particulière aux personnes les plus chers à mon cœur, Tonton Mostefa, Tonton Ameur, tonton Ahmed et mon chère frère Mostafa tonton Hmida, tonton Mohamed et bien sûr Tata hassina pour leurs encouragement permanemt, leurs conseils que j'appliquerai toujours, pour leurs amour sincère, pour leurs gentillesse, et je remercie surtout tata Yamina qui m'a vraiment soutenu et qui essaye toujours de trouver un moyen pour me déstresser.

À mon cher frère, Khaled, la meilleure personne qui était à mes côtés dans les mauvais moments avant les bons, pour tous nos bons souvenirs ensemble.

Tous les membres de ma grande famille pour leur encouragement et leur patience et particulièrement mes sœurs Yasmine Fatima Lina chamsse el khir lydya Inès imen rihab marwa chahinez sidra et mes frère amine Mohamed Ibrahim Ahmed et Ismail

Mes cousins et spécialement ma très chère Zohra.

Comme tu me manques, source de tendresse dans ma vie. Toi qui m'as toujours fait sentir que j'étais le meilleur et le plus aimé de toi, je prie Dieu d'avoir pitié de toi ma grand-mère et de te pardonner autant que tu nous as donné d'amour et de tendresse.

Que Dieu ait pitié de son cœur, de ses cheveux gris et de ses rires, que Dieu refroidisse et paix sur la tombe de ma tante Fatma, que Dieu ait pitié de ceux qui sont morts dans le monde et ne sont pas morts dans nos cœurs.

Il y a beaucoup de mots qui se bousculent pour obtenir des phrases organisées que vous méritez peut-être, mais aucune phrase ne peut suffire à décrire ce que je ressens, vous êtes la bonté et la subsistance, vous êtes aussi la grâce et la miséricorde.

A mon chère binôme Imen, Peu importe combien j'ai cherché dans le dictionnaire des mots et des expressions de remerciement dispersées, je ne trouverais pas de mots qui accompliraient votre droit et votre destin.

Tous mes ami (es) proches Chaima, Nessrine, Karima, Kheira, Fatima et Amel qui mon soutenu et qui ont été des personnes inoubliables et formidables

« Pour Khaled Al-Zikr, qui est décédé et qui était le meilleur exemple de patience et d'enseignants qui n'ont pas négligé de fournir les chemins du bonheur et de la bonté. Mère et Mme Belkacem Fatima, que Dieu ait pitié d'elle. »

Sarra

Résumé

Le nombre d'API Web a augmenté de façon exponentielle, car de plus en plus d'entreprises et d'organisations regroupent et publient leurs données ou ressources commerciales sur Internet sous forme d'API. Pour cette raison, il devient difficile de trouver rapidement et efficacement des API Web à partir d'un tel ensemble. La classification de services a été utilisée pour faciliter la découverte dans un large tas des services. Des méthodes classiques ont été proposées pour regrouper des services web à l'aide de caractéristiques sémantiques mais elles manquaient de précision. Le présent travail vise à proposer une nouvelle approche qui met l'accent sur les actions extraites à partir des descriptions textuelles des SW et intègre des solutions issue du domaine du Deep Learning qui permettent d'améliorer la découverte des services web.

Mots clés : Services Web, Services Web Sémantiques, Classification, Deep Learning, Système de découverte.

Abstract

The number of Web APIs has grown exponentially as more and more companies and organizations aggregate and publish their business data or resources on the Internet in the form of APIs. Because of this, it becomes difficult to quickly and efficiently find Web APIs from such a pool. Service classification has been used to facilitate discovery from a large pool of services. Conventional methods have been proposed to group web services using semantic features but they lacked precision. The present work aims at proposing a new approach that focuses on actions extracted from textual descriptions of SWs and integrates solutions from the field of Deep Learning that improve web services discovery.

Keywords: Web Services, Semantic Web Services, classification, Deep Learning, Discovery system.

ملخص

زاد عدد واجهات برمجة التطبيقات الويب بشكل كبير مع قيام المزيد والمزيد من الشركات والمؤسسات بتجميع ونشر بيانات أو موارد أعمالهم على الإنترنت في شكل واجهات برمجة التطبيقات. لهذا السبب، يصبح من الصعب العثور بسرعة وكفاءة على واجهات برمجة تطبيقات الويب من هذه المجموعة. تم استخدام تصنيف الخدمات لتسهيل الاكتشاف في مجموعة واسعة من الخدمات. تم اقتراح الطرق التقليدية لتجميع خدمات الويب باستخدام الخصائص الدلالية لكنها تفتقر إلى الدقة. يهدف هذا العمل إلى اقتراح نهج جديد يركز على الإجراءات المستخرجة من الأوصاف النصية لخدمات الويب ويدمج الحلول من مجال التعلم العميق التي تعمل على تحسين اكتشاف خدمات الويب.

الكلمات المفتاحية: خدمات الويب، خدمات الويب الدلالية، التصنيف، التعلم العميق، نظام الاكتشاف.

Table des matières

Liste des Figures

Liste des Tableaux

Introduction générale

Plusieurs technologies innovantes ont été introduites ces dernières années pour améliorer la communication, le partage des ressources et l'interopérabilité des systèmes. Les problèmes liés à ces tâches ont amené les experts du domaine à inventer et découvrir des nouvelles technologies. Parmi les technologies apparues au début du 21ème siècle et largement adoptées par les entreprises, on trouve la technologie des services Web.

Les services Web sont des composants logiciels qui n'ont aucune contrainte de compatibilité logicielle ou matérielle. Les services Web présentent de nombreux avantages, ils peuvent être utilisés à distance via n'importe quel type de plate-forme, ils peuvent être utilisés pour développer des applications distribuées et sont accessibles depuis n'importe quel type de machines. Les services Web appartiennent à des applications capables de collaborer entre elles de manière transparente pour les utilisateurs et leur implémentation étant basée sur une architecture distribuée. L'architecture orientée services (Service Oriented ArchitectureSOA) est un style architectural basé sur la description des services et leurs interactions. Les services sont publiés dans des annuaires par les fournisseurs qui les hébergent. Puis, ils sont accessibles via un réseau pour que les utilisateurs les découvrent, les sélectionnent, les invoquent et les utilisent. Cependant ils se retrouvent souvent confrontés au défi d'élaborer et exécuter combinaison de services qui répondent le mieux à leurs besoins. Une solution serait d'aider intelligemment les utilisateurs à trouver leurs services grâce à une découverte pertinente. Ce processus est appelé « *La découverte de service Web* ».

Dans ce sens, nous avons proposé une approche de découverte de services web basée sur les techniques de classification de texte issues du domaine de Deep Learning. Ce choix est justifiable vu la nature textuelle des données sur lesquelles nous avons travaillé.

Pour illustrer cela, nous avons adopté le plan de travail suivant:

Nous commençons par la définition, caractéristiques et architecture des Services Web, en effet nous présentons les services web sémantiques en évoquant leur évolution de la syntaxe à la sémantique, ainsi que les standards utilisés dans la description sémantique des services web.

Ensuite, nous donnons la définition de SWSoc et nous parlons de service web social des modèles. En fin, nous présentons la problématique et la démarche pour découvrir les services web.

Dans le deuxième chapitre nous allons aborder la notion de la classification de texte où nous allons pouvoir appliquer les techniques de l'apprentissage automatique.

Dans le troisième chapitre nous introduisons notre approche appliquée pour améliorer la découverte des services Web sémantiques et les outils utilisés pour implémenter leur propre solution, nous expliquons toutes les étapes de mise en œuvre et nous présentons les résultats obtenus à partir de la mise en œuvre de notre solution.

Chapitre I

Les Services Web

Sommaire

I.1 Introduction

Aujourd'hui, il y a un certain nombre de plateformes pour construire les applications. En général, elles utilisent leurs propres protocoles. Par conséquent, les applications fonctionnant sur différentes plateformes ont peu de possibilités d'échange de données. La prise de connaissance de ces limitations a conduit à un effort majeur pour normaliser les formats de données et l'échange de données. En fait, de plus en plus d'attention se tourne vers un nouveau paradigme informatique : l'intégration transparente des services Web qui contourne les barrières logicielles et matérielles traditionnelles.

Au cœur de cette vision se trouve le concept d'interopérabilité, qui est la capacité de différents systèmes à communiquer et à partager des données de manière transparente. C'est le but des services web.

Un service Web est une logique d'application programmable accessible à l'aide de protocoles Internet standard, qui peuvent également être décrits comme la mise en œuvre de normes Web pour une communication transparente entre les appareils et entre applications.

I.2 Les services web

Selon la définition du W3C (World Wide Web Consortium), un service web est un composant logiciel identifié par un URI[1], dont les interfaces publiques sont définies et appelées en XML[2]. Sa définition peut être découverte par d'autres systèmes logiciels. Les services web peuvent interagir les uns avec les autres d'une manière spécifiée par leurs définitions, en utilisant des messages XML véhiculés par des protocoles internet.

La figure I.1 montre les aspects généraux d'un service web

D'abord, le service web est décrit par une interface XML appelée WSDL, qui peut échanger des documents XML avec d'autres services qui utilisent SOAP, peut être recherché dans un fichier un manuel comme UDDI.

[1]URI : Uniform Resource Identifier
[2] XML : eXtended Markup Language

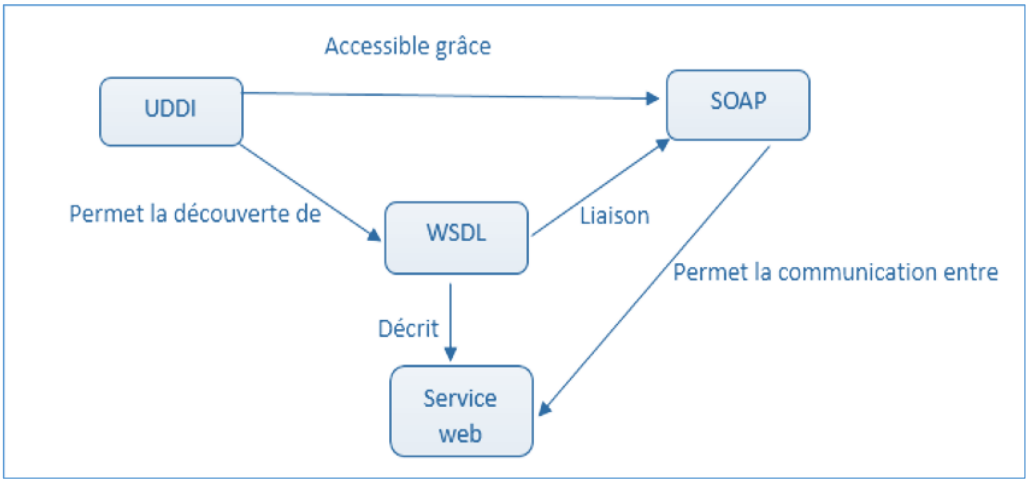

Figure I.1-*Les différentes relations entre les spécifications des services web.*

I.2.1 L'infrastructure des services web

Si le web est constitué de plateformes totalement hétérogènes où s'entremêlent les intérêts des différents acteurs du marché, cela ne l'a pas empêché de se développer et de devenir populaire. Ce succès est principalement dû à la publication d'un ensemble de standards ouverts, dont le plus célèbre qui est le protocole HTTP. HTTP fournit un mécanisme d'échange de données de toute nature, quelle que soit la nature des plates-formes impliquées.

Le cycle de vie d'un web service se déroule ainsi : une fois créé, le service est déployé sur le réseau (local ou Internet). Un utilisateur ayant un besoin spécifique recherchera alors un service correspondant à ses besoins à l'aide d'un annuaire spécialisé. Enfin, une fois le service trouvé, l'utilisateur invoquera le service : une communication sera établie entre l'utilisateur et le Web Service. Illustré à la Figure I.2, repose sur trois technologies principales : SOAP, WSDL et UDDI [1].

- **SOAP (Simple Object Access Protocol) :** assure la communication entre les services web.
- **WSDL (Web Services Description Langage) :** offre un schéma formel de description services web.
- **UDDI (Universel Description Discovery and Integration):** offre une manière uniforme de définir les registres des services web et un schéma simplifié des descriptions extensibles des services web [1].

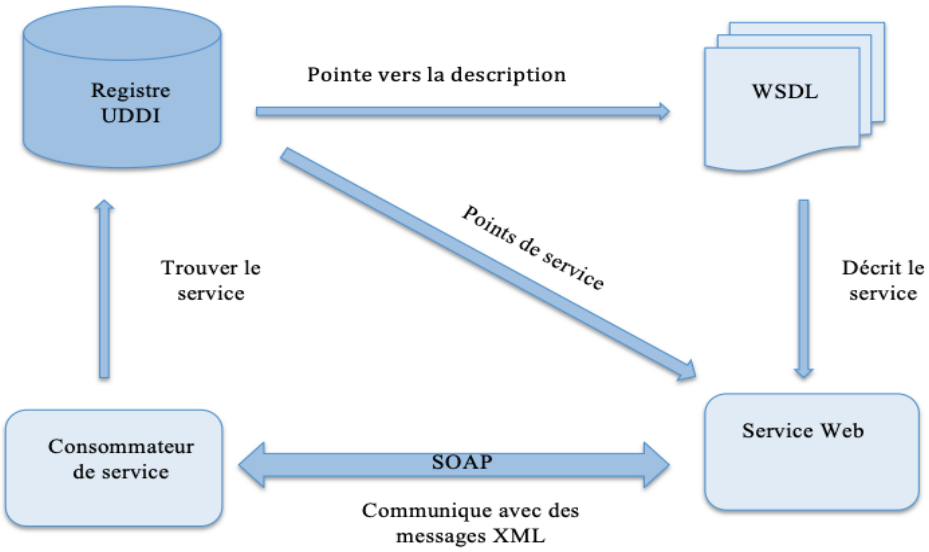

Figure I.2-*l'infrastructure des services web [1].*

I.2.2 Les caractéristique des services web

Un service web a les caractéristiques suivantes [2] :

- Il est accessible sur le réseau.
- Il possède une interface générique décrite en XML.
- Sa description est stockée dans un répertoire.
- Il communique à l'aide de messages XML, qui sont transmis par des protocoles Internet.

I.2.3 L'Architecture générale d'un service Web

Dans la figure (I.3) l'architecture de référence des services web repose sur les trois concepts suivants [3] :

- Le fournisseur de service : définit le service publie la description dans l'annuaire réalise les opérations.
- L'annuaire : obtenez les descriptions des services publiés par le fournisseur et répondez aux recherches de services fournis par les clients.
- Le client : instantané la description du service grâce l'annuaire utilise le service.

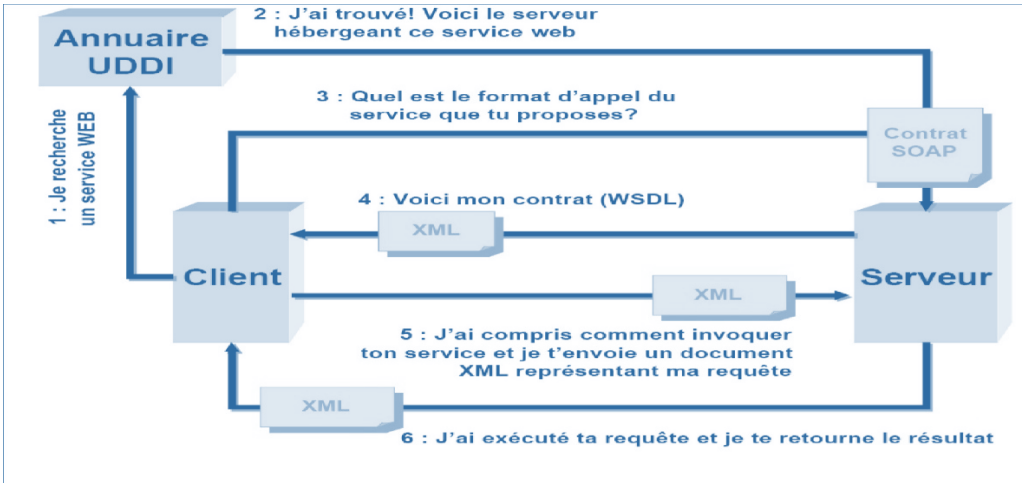

Figure I.3-*structure des services web [3]*.

I.2.4 Les type des services web

Il existe deux types de services Web :

- Les services de type SOAP (Simple Object Access Protocol) : utilisent les standards UDDI, WSDL et SOAP.
- Les services web de type REST (Representational State Transfer) : utilise le protocole HTTP pour faire des appels mètres et vous pouvez obtenir le résultat sous forme JSON, ou d'autres formats.

I.2.4.1 Les services web de type SOAP

I.2.4.1.1 Architecture

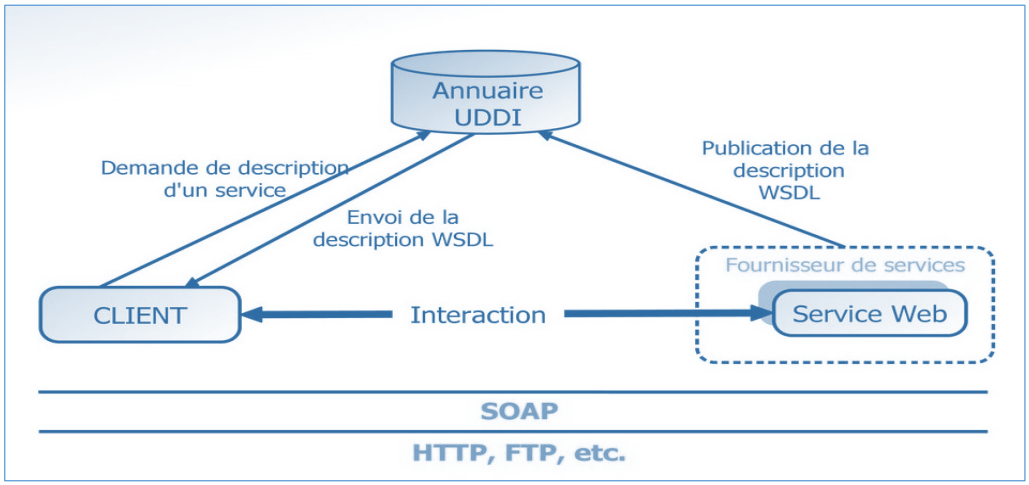

Figure I.4-*Architecture des services web de type SOAP [4]*.

I.2.4.1.2 Le protocole SOAP

Se compose de deux parties :

- Un modèle de données, qui définit le format du message, c'est-à-dire les informations à transmettre.
- Une enveloppe, contenant des informations sur le message lui-même pour permettre le routage et des informations contenant des informations sur le message lui-même afin de permettre le transport et son traitement [5].

I.2.4.1.3 Structure d'un message SOAP

La structure des messages SOAP se divise en quatre parties [6] :

<Envelope> est l'élément racine de chaque message SOAP et contient deux éléments enfants, un élément facultatif <Header> et un élément obligatoire <Body>.

<Header> est un sous-élément facultatif de l'enveloppe SOAP, est utilisé pour transmettre des informations relatives à l'application qui doivent être traitées par les nœuds SOAP le long du chemin du message ; voir L'en-tête SOAP.

<Body> est un sous-élément obligatoire de l'enveloppe SOAP, qui contient des informations destinées au destinataire final du message, voir Le corps SOAP.

<Fault> est un sous-élément du corps SOAP, qui est utilisé pour signaler les erreurs.

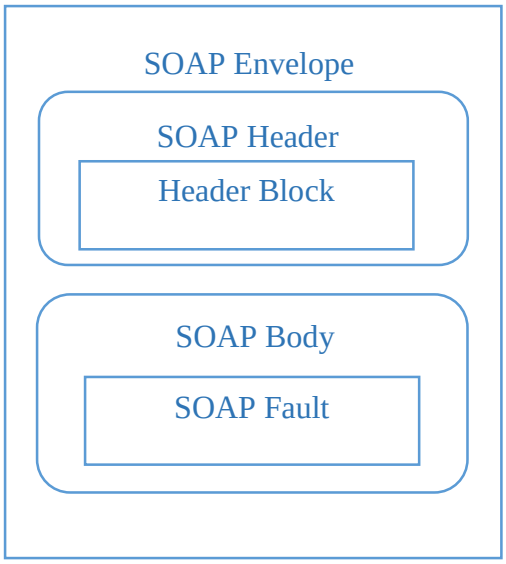

Figure I. 5- *format d'un message SOAP [6].*

I.2.4.2 Les services web de type REST

I.2.4.2.1 Définition

C'est une architecture pour construire des applications, il s'agit d'un ensemble de conventions et de meilleures pratiques, pas d'une technologie autonome.

I.2.4.2.2 Architecture

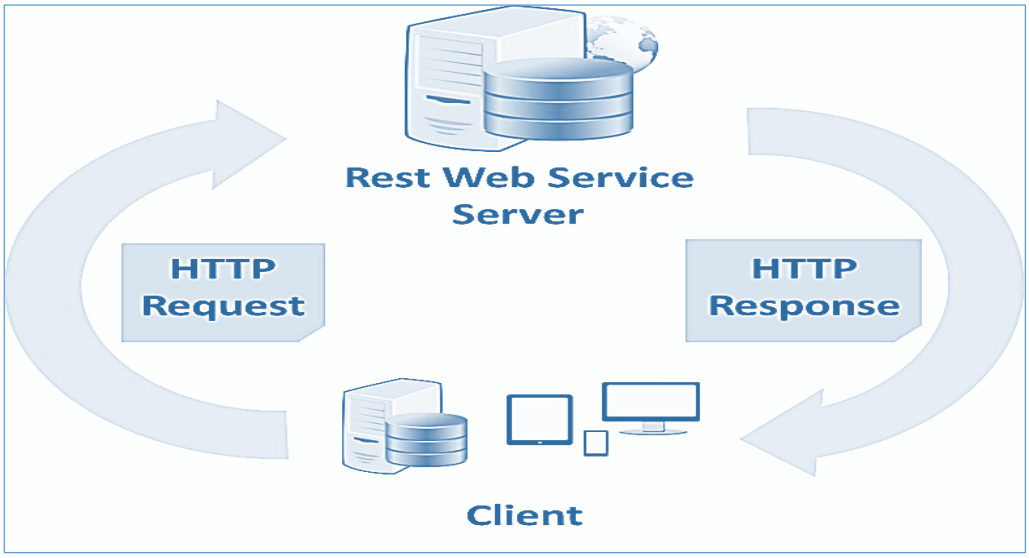

Figure I.6-*Architecture des services web de type REST [7].*

I.2.4.2.2 Caractéristiques

- Les services REST sont sans états
  - Chaque requête envoyée au serveur doit contenir tout ce qui est lié à son état et être traitée indépendamment des autres requêtes.
  - Minimiser les ressources système
- Interface unifiée basée sur les méthodes http
- Les architectures RESTful construites à partir d'URI identifiés de manière unique [8].

I.3 Les services web sémantiques

Les standards utilisés par les services Web offrent des définitions syntaxiques qui ne permettent pas de décrire complètement les fonctionnalités des services et ne peuvent pas être comprises par

les programmes. Il faut l'intervention des humains pour interpréter la signification des entrées, des sorties, des contraintes et du contexte dans lequel les services peuvent être employés.

La technologie du Web sémantique a permis de doter les services Web par une couche sémantique. La combinaison entre le Web sémantique et les services Web a permis de créer une nouvelle technologie dite les services Web sémantiques.

Les services Web sémantiques sont nécessaires dans le but d'automatiser les fonctionnalités suivantes :

− **La découverte automatique des services Web :** ça consiste à localiser automatiquement des services Web qui satisfont des contraintes exigées. Grâce aux SWS, les informations nécessaires pour la découverte des services Web peuvent être spécifiées formellement.

− **L'invocation automatique des services Web :** ça consiste à exécuter automatiquement un service Web identifié par un utilisateur ou un agent logiciel. Après une requête, un service

Web est exécuté comme un ensemble d'appels de fonction grâce à une API fournie par les SWS.

− **La composition et l'interopérabilité automatique des services Web :** cette tâche implique la sélection, la composition et l'interopérabilité automatique des services Web pour effectuer d'autres tâches. Grâce aux SWS, les informations nécessaires pour sélectionner et composer des services vont être encodées avec ces services Web.

− **La surveillance de l'exécution automatique d'un service Web :** cette tâche permet de savoir l'état de la requête d'un client pendant l'exécution d'un service Web. Ainsi, les SWS fournissent des descriptions de l'état d'exécution des services [10].

I.3.1 L'infrastructure des services Web sémantiques

Le développement des services Web sémantiques est effectué en dimensions : les activités d'utilisation, l'architecture et l'ontologie de service

Ces trois dimensions sont relatives aux besoins des SWS niveau application, physique et conceptuel :

− **Les activités d'utilisation** : remplissent les besoins fonctionnels qu'un Framework les SWS devrait supporter.

− **L'architecture des SWS :** décrit les composants nécessaires pour accomplir des activités.

− **L'ontologie de service** : intègre tous les concepts utilisés pour la description des services Web sémantiques [10].

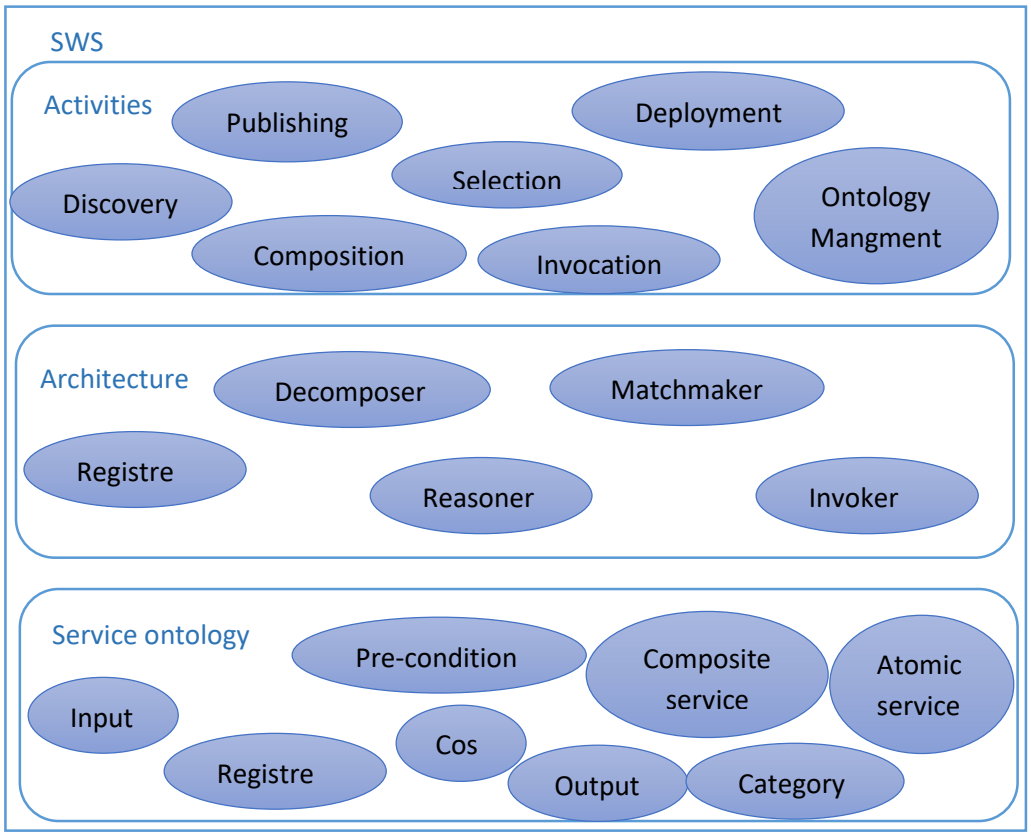

Figure I.7-*Infrastructure des services Web sémantiques [10]*.

I.3.2 Approches des services web sémantiques

Il existe deux approches pour décrire la sémantique des services web [11] :

- L'annotation avec l'information sémantique des standards existants que nous avons déjà présentés (WSDL, UDDI, SOAP)
- Le développement des langages qui décrit les services Web ainsi que leur sémantique.

I.3.2.1 Annotation des standards déjà existants

Les annotations sémantiques incluent l'enrichissement et la complétion des services. Elle établit une correspondance entre les éléments d'un concept et celle d'un ensemble d'ontologies référentielles. Les principaux modèles de cette approche :

- **WSDLS :** est un langage de description sémantique pour les services, cette sémantique est ajoutée sur deux étapes :
  - La première étape consiste à se référer à la définition du WSDL, à une ontologie spécifique à la publication.
  - La deuxième étape consiste à annoter les opérations qui définissent la sémantique WSDL en ajoutant deux nouvelles balises, la balise Action et la balise Contrainte.
    - Balise Action : permet la représentation par action de l'activité
    - Balise Contrainte : représente les conditions de fonctionnement avant et après [11].

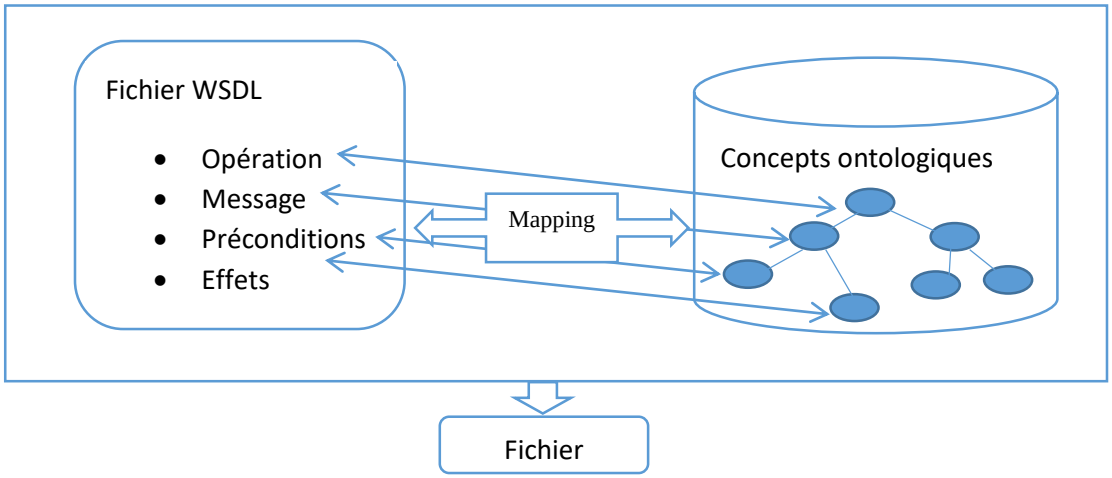

Figure I.8 *Mapping entre le fichier WSDL et les concepts ontologiques [11].*

- **SA-WSDL (Semantic Annotation of WSDL and XML Schema)**

SAWSDL est un complément à WSDL-S (Web Service Description Language - Semantic). Le SAWSDL définit un mécanisme d'annotation pour identifier les éléments WSDL à l'aide de l'ontologie. Cette annotation est basée sur la définition des attributs étendus du critère de description. Les annotations sémantiques font référence à une ontologie préexistante. Le mécanisme d'annotation SAWSDL est indépendant de tout langage de représentation d'ontologie.

Le SAWSDL fournit deux types d'annotations sémantiques : la première pour définir le concept sémantique (représenté par l'attribut model référence) et la deuxième pour faire le lien entre le concept et le document WSDL (représenté par les attributs up-and-down du schéma) [12].

I.3.2.2 Langages de description sémantique

- **OWL-S**

Appelé DAML-S (DARPA Proxy Markup Language Service) dans les versions antérieures, est une ontologie de haut niveau pour décrire les services Web, dont les objectifs sont de résoudre l'ambiguïté et de rendre les descriptions de service compréhensibles par l'appareil. Les auteurs présentent une ontologie de services Web dans le but d'automatiser la découverte, l'appel, la configuration et la surveillance de l'exécution des services. Ces auteurs adoptent l'idée des catégories OWL et suggèrent l'ontologie OWL-S. OWL-S fournit trois informations de base sur les services Web. La figure I.9.1 décrit ces trois concepts :

- **Le « service profile »** pour objectif d'automatiser la découverte et la sélection, il fournit une description de haut niveau d'un service et de son fournisseur.
- **Le « service model »** définit le flux de contrôle et données d'une composition de services.
- **Le « service grounding »** décrit les moyens d'accès au service en spécifiant le protocole de communication, le format des messages, l'encodage des paramètres [11].

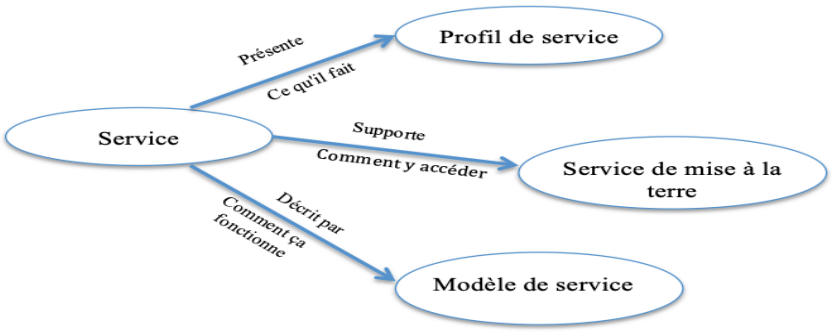

Figure I.9-*l'ontologie d'OWL-S [11].*

- **Web Service Modeling Ontology (WSMO)**

WSMO (Web Service Modeling Ontology), initié par le groupe de travail WSMO d'ESSI, il s'appuie sur le WSMF, WSMO Web Service Modeling Framework partagé avec OWLS dans le but d'automatiser ces services liés aux services.

La Séparation des descriptions des services et des technologies du web sémantique possibles est un aspect essentiel de WSMO. En effet, WSMO fournit un modèle de description d'ontologies qui est indépendant du langage utilisé pour les décrire. Le diagramme de classes UML, illustré à la Figure I.12**,** il représente les principaux éléments de l'ontologie WSMO et leurs interactions. Le concept central est l'élément WSMO spécialisé dans l'ontologie ou le web Service ou un goal ou un mediator.

•Un élément ontologie définit une ontologie de référence importée par un élément à partir de l'ontologie WSMO afin de décrire ses caractéristiques.

• Un élément web Service est une « unité de calcul capable de répondre à requête de l'utilisateur » services web WSMO définis de manière uniforme comprenant les éléments suivants : fonction service, interfaces de description, propriétés non fonctionnelles, ontologie importée et intermédiaires utilisateurs.

- Les fonctions de service, désignées par capacité, sont décrites en termes de forme processus associés aux conditions préalables, aux hypothèses, aux post-conditions et aux effets. Les conditions préalables et les suffixes décrivent exigences en matière de données avant la mise en œuvre du service et des modifications qu'ils souffrent après l'exécution, Décrire les hypothèses et les influences exigences et changements liés aux services en interaction avec service décrit.
- Les interfaces de description décrivent le comportement du service au format. En d'autres termes, l'ordonnancement des opérations.
- Les propriétés non fonctionnelles d'un service Web décrivent les fonctionnalités qui ne sont pas directement liés aux données traitées, telles qu'un exemple de la durée de l'opération.

- Les ontologies importées sont utilisées pour faire référence aux éléments de description du service. Des intermédiaires sont utilisés si deux services web en interaction sont importants deux ontologies différentes.

• Un élément goal est utilisé pour décrire les désirs des utilisateurs en termes de fonctionnalité requise. Les objectifs sont le point de vue de l'utilisateur sur le processus utilisant des services Web, ils constituent une entité à part entière dans le modèle WSMO. L'objectif décrit la fonctionnalité, les entrées/sorties, les préconditions et les post-conditions d'un service Web.

• Un élément médiator est utilisé pour résoudre de nombreux problèmes d'incompatibilité, tels que l'incompatibilité des données dans le cas des services web l'utilisation de termes différents, le processus d'incompatibilité dans le cas une combinaison de services Web et d'incompatibilités de protocole lorsque créer des connexions [13].

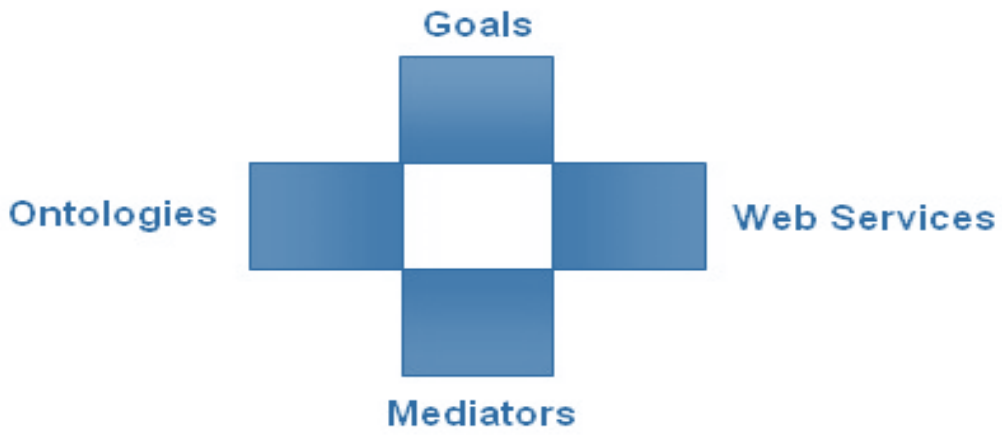

Figure I.10-*Composants de WSMO [13]*.

I.4 Les services Web sociaux

Les services web sociaux [14] sont nés de l'intersection entre les services web et les réseaux sociaux. Sur la base de l'interaction des services web, des qualités sociales sont attribuées aux services Web.

I.4.1 Les modèles sociaux des services Web

Le modèle social est composé de quatre étapes :

I.4.1.1 Relations liants les services Web

L'objectif de cette étape est d'établir les relations qui existent entre les services Web, trois types de relation pu être établis selon, suivant les cas :

1. Les services Web qui offrent des fonctionnalités similaires
    - Son en compétition, vu qu'un seul service sera sélectionné à la fois.
    - Substitut les uns autres, dans le cas d'un
2. Les services Web qui offrent des fonctionnalités différentes, collaborent à la création de nouveaux services composites [14].

I.4.1.2 Réseaux sociaux correspondants aux relations

Le but de cette étape est d'identifier les réseaux potentiels qui peuvent mettre les services web en contact. Chaque relation constitue une base sur laquelle un réseau est développé [14].

I.4.1.3 Construction du réseau social de services Web

Le but de cette étape est de définir les composants sur lesquels le réseau social sera construit. Ces composants sont des nœuds et des bords qui prennent en charge les services Web et les relations, respectivement [14].

I.4.1.4 Comportement des services Web

Le but de cette étape est d'identifier le comportement des services Web, différents types de comportements sociaux existant la vie réelle [14].

I.4.2 Services Web sociaux en action

Lorsque les services Web sont « sociaux », ils peuvent fournir des informations sur comportement et utilisation passée. D'un point de vue architectural, les services ont été déployés dans un nuage, on peut facilement exploiter ses aspects sociaux .Les réseaux sociaux pour les services Web peuvent être créés et maintenus à partir de Trois façons:

- **Collaboration :** En combinant leurs fonctions respectives, SWSoc est Capacité à travailler ensemble sur des demandes complexes d'utilisateurs. Ainsi, SWSoc exploite son propre

réseau de collaborateurs afin de prendre la décision s'il aime collaborer avec des pairs sur la base d'expériences passées. Pouvez comme le recommandent les pairs.

- **Compétition :** les SWSocs s'affrontent lorsqu'ils présentent caractéristiques similaires. Ils se distinguent par leurs caractéristiques non fonctionnelles lorsque des besoins non fonctionnels des utilisateurs doivent être satisfaits. Ainsi, SWSoc découvre son propre réseau de concurrents, qui il permet de tenter d'améliorer ses propriétés non fonctionnelles par rapport à d'autres pairs.
- **Substitution :** Bien que SWSoc soit en concurrence, ils peuvent toujours s 'entraider en cas d'échec, s'ils proposent des emplois similaires. Par conséquent, SWSoc gère ses propres réseaux d'agents afin de pouvoir respect des accords de niveau de service (SLA) en cas d'échec potentiel. Il peut alors sélectionner ses meilleurs Alternatives en réponse aux besoins non fonctionnels des utilisateurs.

Ces trois façons d'entretenir les réseaux sociaux peuvent être envisagées indépendamment en tant que réseau de comportements sociaux. Ils pourraient être la cible de commencer à construire plusieurs réseaux en fonction des interactions entre services Web tels que la délégation et supervision [15].

I.4.3 Systèmes de recommandation et plates-formes sociales

De nos jours, l'utilisation généralisée d'Internet partout dans le monde permet à de nombreuses personnes de se connecter à Internet. Cette explosion du Web 2.0 a créé un besoin croissant de systèmes de recommandation basés sur des méthodes d'exploration des réseaux et des informations sociales. Pour de tels systèmes, une infrastructure sociale peut être exploitée, également appelée réseau social ou communauté virtuelle.

La croissance exponentielle du réseau social présente de nouveaux défis opportunités de recherche en RS[3]. La principale raison est le fait que le réseau social se transforme consommateurs d'informations en contributeurs actifs, leur permettant de partager Statut, commenter ou évaluer le contenu Web. Trouvez du contenu connexe et Intéressant au bon moment et dans le bon contexte présent des approches stimulantes recommandations actuelles. Dans le même temps, la principale

[3]RS : Réseaux sociaux.

valeur ajoutée des plateformes sociales est d'encourager Interaction entre utilisateurs. Chaque interaction peut être extraite et utilisée entrer dans le système de recommandation, car cela aide à mieux comprendre les intérêts utilisateurs et besoins d'information. De plus, l'architecture de réseau social de base de la plate-forme sociale peut aider à générer plus de recommandations, fiable (par exemple, en tenant compte de la distanciation sociale dans le processus Recommandation, nous faisons généralement confiance à plus de recommandations à partir de liens plus étroits). On peut donc conclure que le réseau Le social présente une excellente opportunité d'améliorer les systèmes de recommandation [16].

D'autre part, les systèmes de recommandation peuvent clairement aider à améliorer participation des utilisateurs aux systèmes sociaux, où ils peuvent recommander de nouveaux amis, service ou contenu intéressant. Ainsi, l'utilisateur sera plus motivé à continuer participation à la plateforme sociale, car plus les utilisateurs partagent de contenu, plus le système peut recommander des connexions pertinentes, avec un profil précis à leur sujet [16].

Figure I.11-*Apport réciproque entre le système de recommandation et le réseau social [16]*.

I.5 Découverte des services web

I.5.1 Problématique de découverte de service web

De nos jours, le nombre de services web augmente avec l'émergence du web sémantique et l'augmentation des demandes des clients pour trouver les services recherchés, la découverte est plus importante et nécessaire pour un partage, un échange et une réutilisation efficaces des informations web entre les utilisateurs [17].

Car cet objectif de la SOA est d'offrir un environnement dynamique et hétérogène de découverte et de sélection du meilleur service parmi eux et exploité de manière plus "intelligente" par les machines. Ils peuvent "comprendre" les contenus décrits dans les ressources et faciliter les tâches de traitement automatique et efficace de l'information.

Le service de découvert est comparant la demande client avec l'ensemble des services disponibles et publiés sur le web, il utilise des descriptions d'interfaces, des descriptions de qualité de service comportementales (fonctionnelles) ou (non fonctionnelles).

La découverte de services Web est vue comme un problème de recherche et de sélection de services pour répondre aux besoins de l'utilisateur, et l'importance de ce choix doit être adapté aux préférences de l'utilisateur [18].

I.5.2 Approches de découverte de services Web sémantiques

La découverte de services Web a révélé que les méthodes de confiance existantes sont hétérogènes dans la méthode de correspondance utilisée. Elle peut être classée en trois catégories : l'approche algébrique, l'approche déductive et l'approche hybride.

I.5.2.1 Approche algébrique

L'approche de découverte algébrique a ses origines dans l'algèbre et les mécanismes de recherche d'informations. Elle repose sur le calcul du degré de similarité textuelle à partir de graphes structurés conçus à cet effet, ou sur le calcul de la distance (chemin) entre des concepts identiques. Cette approche utilise des mécanismes d'appariement structurels, numériques et syntaxiques en couplant des graphes structurés et en calculant des distances scalaires pour vérifier

la similarité grammaticale. Pour exploiter la sémantique, les mécanismes d'appariement utilisent des fréquences de termes et de sous-graphes.

I.5.2.2 Approche déductive

L'approche déductive est basée sur la logique. Les entreprises qui choisissent cette classe de méthodes, ils utilisent des descriptions de services et des requêtes définies dans des langages dérivés de formalismes booléens, comme la logique de description et la logique du premier ordre. Ils utilisent également des booléens pour découvrir des services Web et exploitent l'ontologie pour couvrir l'aspect sémantique. Pour calculer le degré de conformité, ils recourent à diverses méthodes et ciblent divers éléments de la description du service en tenant compte de leur sémantique.

I.5.2.2 Approche hybride

L'approche hybride utilise des mécanismes inférentiels qui intègrent des méthodes de calcul de distance. De nombreuses entreprises adoptent cette approche. Nous notons que d'autres travaux basés sur la CBR entrent également dans la catégorie des méthodes hybrides [19].

I.6 Conclusion

Dans ce premier chapitre, nous avons présenté la notion des services Web ainsi que les grands défis liés à leur cycle de consommation. Entre autres, nous avons montré l'importance du processus de découverte dans la prestation d'un service de qualité, la découverte des services web est principalement syntaxique, mais avec le développement de la technologie du Web sémantique, les techniques proposées pour la découverte de services Web sont devenues essentiellement sémantiques qui ont les besoins d'ontologie. Cette ontologie est difficile à mettre en œuvre et le système ne contient pas de mémoire donc il ne peut pas être l'historique si bien que les services du web social sont apparus. Dans le chapitre suivant, nous présenterons les techniques de machine Learning basées sur la classification de texte.

Chapitre II

Les Techniques De Machine Learning Pour La Classification De Texte

Sommaire

II.1 Introduction

Au cours des dernières années, le développement rapide d'Internet et des technologies de l'information, en particulier avec l'ère des méga données, engendre une énorme quantité de données sur tous les domaines de notre vie. La quantité croissante d'informations enregistrées empêche les gens de trouver ce dont ils ont besoin. Dans le passé, les gens choisissaient de catégoriser manuellement les informations textuelles, ce qui prend du temps, demande beaucoup de travail et coûte cher.

La classification de texte est le processus d'étiquetage du texte en fonction de son contenu, c'est l'un des principes de base du traitement du langage naturel.

La classification de texte, également connue sous le nom de classification de texte, est un problème classique du traitement du langage naturel, qui vise à attribuer des étiquettes de balises à des unités de texte telles que des requêtes, des paragraphes et des documents. Il dispose d'un large éventail d'applications, y compris la réponse aux questions, la détection de spam, l'analyse des sentiments, l'analyse des actualités, la classification des intentions des utilisateurs, l'analyse du contenu, etc.

Les données textuelles peuvent provenir de diverses sources, notamment des données Web, des e-mails, des chats, des médias sociaux, des tickets, des réclamations d'assurance, des avis d'utilisateurs et des commentaires du service client, pour n'en nommer que quelques-unes. Le texte est une source d'informations incroyablement riche. Mais extraire des informations d'un texte peut être difficile et prendre du temps, en raison de son manque de structure. La classification du texte peut être effectuée manuellement ou par étiquetage automatique. Avec le niveau croissant de données textuelles dans les applications, la classification automatique du texte devient de plus en plus importante.

Vu la nature de description des services web de notre Dataset, voir la section (II.3), on va essayer de se base sur certaines techniques de machine Learning pour la classification de texte.

II.2 Word Embedding

L'incorporation de mots (Word Embedding) est l'un des concepts les plus puissants de deep Learning appliqué au traitement du langage naturel. Il peut capturer le contexte d'un mot dans le document, les similitudes sémantiques et syntaxiques, les relations avec d'autres mots, etc.

Le concept de base de l'incorporation de mots est que chaque mot utilisé dans une langue peut être représenté par un ensemble de nombres réels (un vecteur). Ils ont appris des façons de représenter le texte dans un espace à n dimensions où les mots ayant le même sens ont des représentations similaires. Cela signifie que deux mots similaires placés très près l'un de l'autre dans l'espace vectoriel ont des représentations vectorielles presque similaires. Ainsi, lors de la construction d'un mot qui incorpore un espace, le but est de capturer une sorte de relation dans cet espace, que ce soit le sens, la morphologie, le contexte ou une autre sorte de relation.

Certaines des principales caractéristiques de la fonction d'incorporation de mots sont répertoriées ci-dessous :

- Chaque mot a un mot intégré unique (ou "vecteur"), qui est juste une liste de nombres pour chaque mot.
- L'incorporation de mots est multidimensionnelle ; généralement, pour un bon modèle, la longueur de l'incorporation est comprise entre 50 et 500.
- Pour chaque mot, l'ensemble capture le "sens" du mot.
- Les mots similaires se terminent par des valeurs intégrées similaires [20].

II.2.1 One Hot Encoding

L'une des techniques les plus élémentaires utilisées pour représenter les données numériquement est One Hot Encoding. Dans cette méthode, un vecteur est créé avec une taille égale au nombre total de mots uniques. Les valeurs des vecteurs sont affectées de sorte que la valeur de chaque mot sur son index soit « **1** » et les autres mots soient « **0** ». Par exemple, voir la figure II.1 [22].

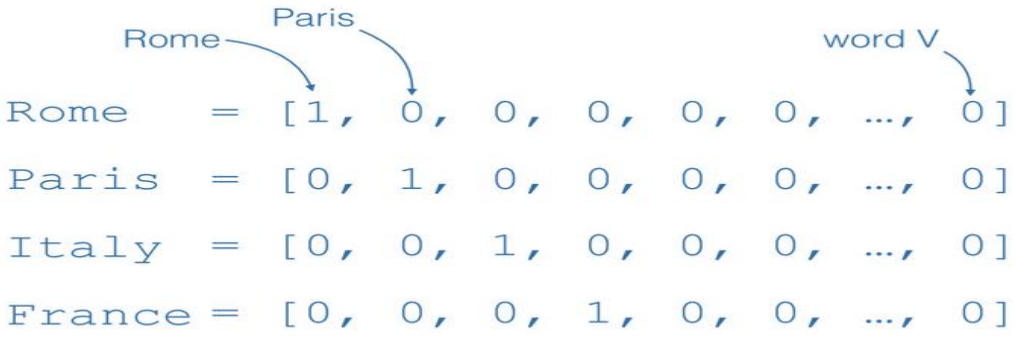

Figure II.1-*one Hot Encoding.*

II.2.2 Bag of Word

Le modèle du sac de mots est un modèle simplifié utilisé dans le traitement du langage naturel et la recherche d'informations. Dans ce modèle, un texte (tel qu'une phrase ou un document) est représenté comme un sac (plusieurs ensembles de) de ses mots, indépendamment de la grammaire et même de l'ordre des mots, mais en conservant toujours son intégrité multi-sens.

Le modèle de sac de mots est couramment utilisé dans les méthodes de classification de documents où la (fréquence de) l'occurrence de chaque mot est utilisée comme caractéristique pour former un classificateur.

Le modèle du sac de mots est facile à comprendre et à mettre en œuvre et a montré un grand succès dans des problèmes tels que la modélisation du langage et la classification des documents.

Cela implique deux choses :

- Un vocabulaire de mots connus : Cette étape s'articule autour de la construction d'un corpus documentaire constitué de tous les mots uniques de l'ensemble du texte présents dans les données fournies. C'est un peu comme un dictionnaire où chaque index correspond à un mot et chaque mot est une dimension différente
- Une mesure de la présence de mots connus : Maintenant, si nous prenons le premier examen et décompte de chaque mot dans le tableau ci-dessous, nous aurons où la ligne 1 correspond à l'index des mots uniques et la ligne 2 correspond au nombre de fois un mot apparaît dans une critique [22].

II.2.3 TF-IDF

TFIDF, abréviation « **Term Frequency-Inverse Document Frequency** », est une statistique numérique destinée à refléter l'importance d'un mot pour un document dans une collection ou un corpus. La valeur Tf–Idf augmente proportionnellement au nombre de fois qu'un mot apparaît dans le document et est compensée par le nombre de documents dans le corpus contenant le mot, ce qui permet de corriger le fait que certains mots apparaissent plus souvent qu'en général. Tf–Idf est l'un des schémas de pondération des termes les plus populaires aujourd'hui [22].

- **Term Frequency (TF)** est utilisé dans la recherche d'informations et indique la fréquence à laquelle un terme, mot apparaît dans un document. La fréquence des termes indique l'importance d'un terme particulier dans le document. C'est le nombre de fois qu'un mot W_i apparaît dans la revue r_j par rapport au nombre total de mots dans la revue r_j [22].

$$TF\left(w_{i,r_j}\right) = \frac{No.of\ times\ w_i\ \ Occurs\ in\ r_j}{Total\ no.of\ words\ in\ r_j}$$

- **Inverse document frequency (IDF)** est une mesure de la quantité d'informations qu'un mot fournit, c'est-à-dire est-il commun ou rare dans tous les documents. Il permet de calculer le poids des mots rares sur l'ensemble des documents du corpus. Les mots qui apparaissent rarement dans le corpus ont des scores IDF élevés. C'est la fraction inverse sur l'échelle logarithmique des documents contenant le mot (obtenez-la en divisant le nombre total de documents par le nombre de documents contenant le mot, puis en prenant le logarithme de ce quotient) [22] :

$$Idf(t,D) = \log\frac{N}{|\{d \in D: t \in d\}|}$$

- **TF–IDF** est calculé comme :

$$TfIdf(t,d,D) = Tf(t,d)\ .Idf(t,D)$$

II.2.4 Word2vec

Word2Vec est un algorithme NLP qui encode la signification des mots dans des vecteurs courts et denses (Word Embedding) qui peuvent être utilisés pour des tâches NLP de base telles que la réponse aux questions, la recherche d'informations, la traduction automatique, la programmation, modèles de langage, etc. Ce ou ces vecteurs contextualisent le sens des mots dans un corpus donné en examinant les mots entourant le mot dans le corpus. Cet algorithme a été introduit en 2013 par Mikolov et al [23].

Il existe deux variantes de Word2vec, Continuous Bag of Word (CBOW) et Skip-Gram, sont les deux architectures utilisées pour la formation des intégrations word2vec.

- **Continuous Bag of Word (CBOW)** les représentations de mots en prédisant un mot central dans la fenêtre des mots de contexte sélectionnés. Dans CBOW, nous échantillonnons une fenêtre de mots de contexte autour d'un mot particulier, l'introduisons dans le modèle et prédisons le mot central. Dans cette architecture particulière, la matrice de poids entre la couche d'entrée et la couche de projection est partagée entre tous les mots. On mappe des vecteurs one-hot de mots d'entrée (du contexte) à la couche de projection (couche d'intégration). La couche d'intégration à n dimensions est multipliée par une autre matrice de poids pour obtenir la couche de sortie. On effectue une opération ***softmax*** sur la couche de sortie pour obtenir une distribution de probabilité sur les vocabulaires.
- **Skip gram** est une autre variante de word2vec. Contrairement à CBOW, où on prédit un mot central basé sur des mots de contexte dans le vocabulaire, ici on essaye d'apprendre la représentation vectorielle des mots en prédisant des mots de contexte autour d'un mot. Le modèle essaie de maximiser la classification d'un mot en fonction d'un autre mot dans la phrase. Idéalement, plus la fenêtre de dépendance est longue, meilleure est la qualité des vecteurs de mots. Les auteurs ont également constaté que cette complexité augmentait et que des mots parfois éloignés avaient peu de rapport avec le mot modélisé actuel. Les auteurs ont utilisé une taille de fenêtre de 10 dans leur article original pour la formation, et les résultats ont montré que le modèle de saut de gramme fonctionnait mieux que CBOW dans certains tests [23].

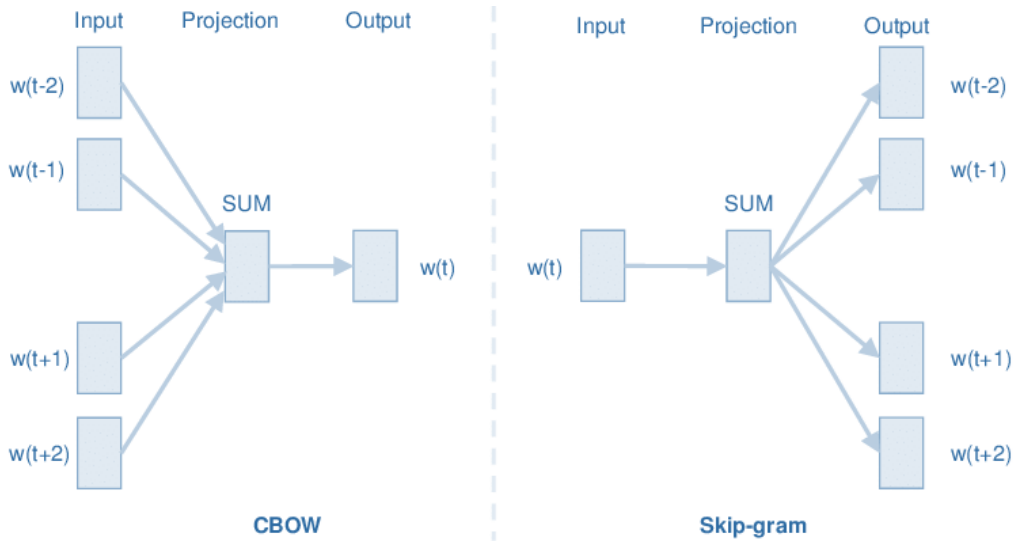

Figure II.2-*différentes dans l'algorithme Word2Vec : Sac continu de mots et Skip-Gram [23].*

II.2.5 GloVe (Global Vectors for Word Representation)

L'algorithme GloVe est une extension de la méthode word2vec pour apprendre efficacement les vecteurs de mots. Il a été développé par Pennington, et al. A Stanford. GloVe capture les statistiques globales et locales d'un fichier. Il s'agit d'un algorithme d'apprentissage non supervisé pour dériver des représentations vectorielles de mots. L'apprentissage est effectué sur la base des statistiques de cooccurrence du mot-clé global synthétisé à partir d'un corpus, et les représentations résultantes représentent des sous-structures linéaires de l'espace vectoriel de mots [24].

Figure II.3-*Architecture de Glove.*

II.2.6 FastText

FastText est assez différent avec 2 combinaisons Word2Vec et GLOVE considèrent chaque mot comme la plus petite unité à pratiquer, FastText utilise le caractère ngram comme la plus petite unité. Par exemple, le mot vectoriel "Apple" peut être décomposé en unités vectorielles de mots séparées comme "AP", "App", "ple". Le plus grand avantage de l'utilisation de FastText est qu'il génère de meilleurs incorporations de mots pour des mots rares ou même des mots qui ne sont pas vus pendant l'entraînement, car les vecteurs de caractères ngram sont partagés avec d'autres mots. C'est quelque chose que Word2Vec et GLOVE ne peuvent pas réaliser [25].

II.3 les techniques de machine Learning pour la classification de texte

II.3.1 Classification de texte

Le traitement du langage naturel, l'analyse des sentiments, la détection de spam et d'intention et d'autres applications utilisent la classification de texte comme technique d'apprentissage automatique de base.

Cette fonctionnalité importante est particulièrement utile pour la reconnaissance de la langue, permettant aux organisations et aux individus de mieux comprendre des éléments tels que les commentaires des consommateurs et les efforts futurs. Le classificateur de texte classe le texte non structuré de catégories de texte prédéfinies. Au lieu que les utilisateurs aient à passer au crible et à analyser de grandes quantités d'informations pour comprendre le contexte, la classification de texte aide à obtenir des informations pertinentes.

Par exemple, les entreprises peuvent avoir besoin de catégoriser les tickets d'assistance entrants afin qu'ils soient envoyés au service d'assistance client approprié [26].

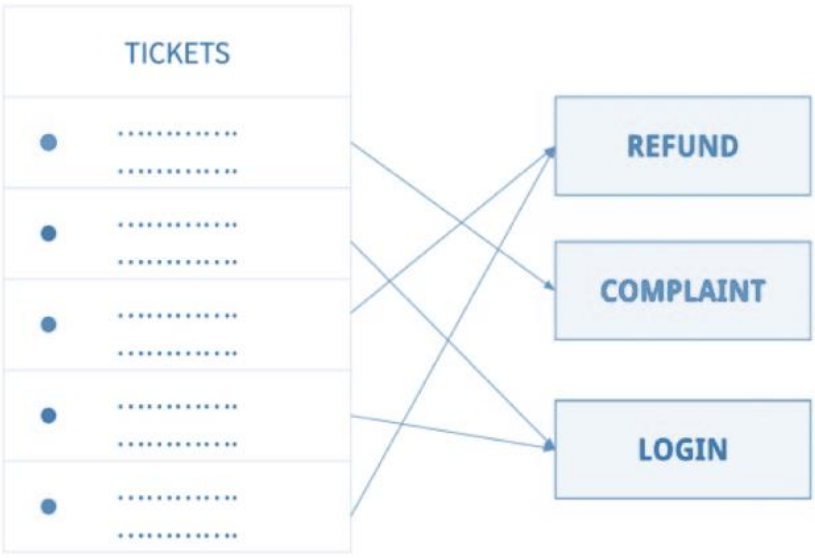

Figure II.4-*Exemple d'étiquettes de classification de texte pour les tickets de support client.*

Le système de classification de texte d'apprentissage automatique n'est pas basé sur des règles définies manuellement. Il apprend à classer les textes en fonction des textes précédents, en utilisant généralement des données d'apprentissage pour des exemples pré-balisés. Les

algorithmes de classification de texte peuvent détecter des corrélations entre des parties de texte distinctes et attendues pour un texte ou une entrée donnés. Dans des tâches très complexes, le résultat est plus que les règles humaines et les algorithmes peuvent apprendre de nouvelles données [26].

II.3.2 Apprentissage Automatique

L'apprentissage automatique est un sous-domaine de l'intelligence artificielle qui donne aux systèmes la capacité de comprendre grâce à des algorithmes. Il est basé sur l'idée d'apprendre à partir de données et de faire des prédictions à partir de ces données et de cette façon, les ordinateurs apprennent des tâches spécifiques sans les programmer. Le but du Machine Learning est de reconnaître entre des structures souvent trop difficiles à détecter ou manuellement. A partir de ces construits, on peut chercher à identifier des individus, des objets, à prédire des valeurs variables à un certain horizon, à expliquer l'occurrence ou d'une caractéristique [27].

Les algorithmes d'apprentissage automatique peuvent être classés en : supervisés, non supervisés, selon le type d'expérience qu'ils ont dans le processus d'apprentissage.

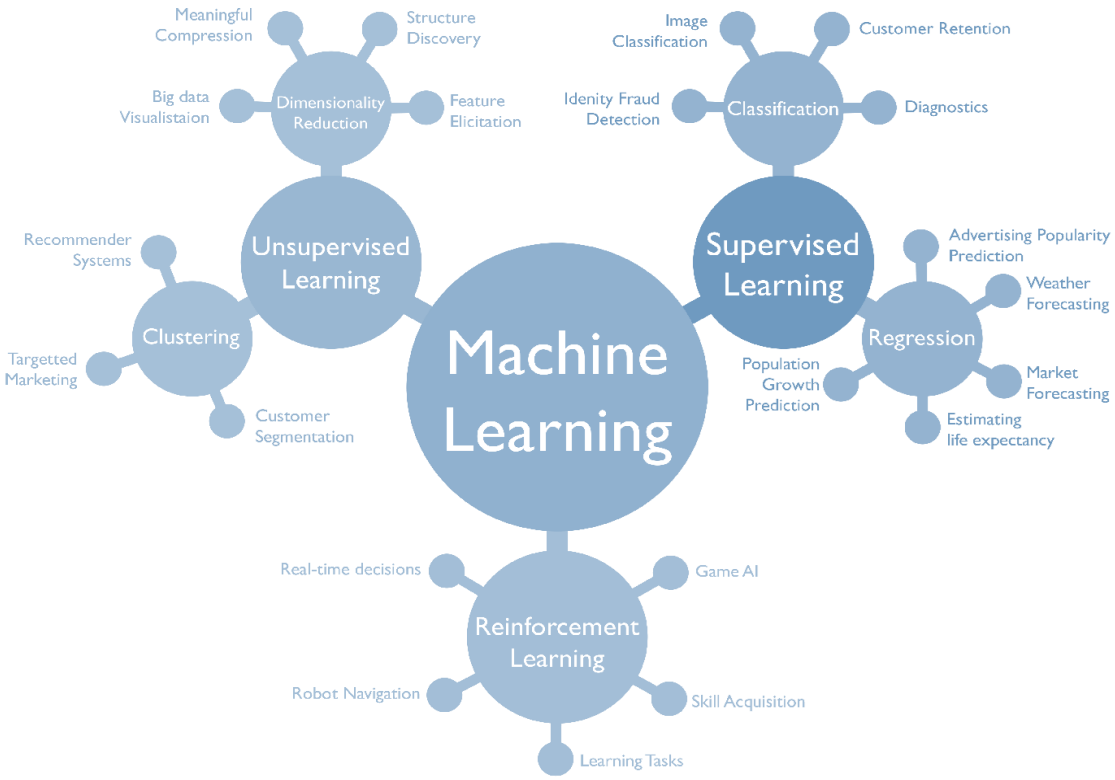

Figure II.5-*Les différentes catégories d'apprentissage automatique.*

II.3.2.1 L'apprentissage Supervisé

L'apprentissage supervisé est une méthode d'apprentissage automatique, qui se caractérise par la création d'un algorithme qui apprend une fonction prédictive. Ceci est rendu possible par la formation à partir d'exemples annotés, constitués d'un groupe de variables d'entrée, ainsi que de leurs variables de sortie respectives. Ce processus d'entraînement est répété jusqu'à ce que la performance souhaitée soit atteinte. Lors de chaque itération, la machine génère un certain nombre de règles, liant les variables d'entrée aux variables de sortie. Ce processus permet au modèle d'apprendre à partir des données et d'appliquer des règles pour prédire avec précision la valeur de sortie lorsque la valeur d'entrée est donnée [28].

Les problèmes d'apprentissage supervisé sont catégorisés en problèmes de « régression » et de « classification ».

II.3.2.1.1 La Régression

Contrairement aux classificateurs qui prédisent les catégories ou les étiquettes, les modèles de régression prédisent les valeurs de sortie en continu, en fonction de la variable indépendante d'entrée. Cette technique est utilisée lorsque la variable de sortie à prédire doit être une valeur continue, par exemple des prévisions météorologiques ou des tendances de marché. Différents modèles de régression existent et varient en fonction de la relation entre les variables dépendantes et indépendantes considérées, ainsi que du nombre de variables indépendantes utilisées pour le modèle. Certains des principaux modèles de régression sont la régression linéaire et multilinéaire simple, la régression de Poisson et la régression vectorielle de support (SVR).

Ainsi, l'analyse de régression est une forme de statistiques inférentielles, utilisée pour identifier les tendances dans les données. Il existe différents algorithmes de régression qui partagent le même objectif de déterminer une valeur de sortie continue en fonction d'une variable d'entrée indépendante [28].

II.3.2.1.2 Classification

La méthode de classification s'applique lorsque l'ensemble de valeurs est discret. Cela équivaut à attribuer une classe à chaque valeur. Les techniques de classification peuvent être basées sur des probabilités, des concepts de proximité ou la recherche d'espaces d'hypothèses. Le choix de la technique convenable est important, il est nécessaire de pouvoir choisir la méthode la plus adaptée qui pourra séparer au mieux les données d'entraînement [29].

- **Naïve bayes**

La classification Naïve Bayes est basée sur l'hypothèse que toutes les caractéristiques sont conditionnellement indépendantes les unes des autres. Cette méthode est basée sur le théorème de Bayes qui calcule la probabilité d'un événement en utilisant les conditions précédentes pertinentes. Ce théorème a été découvert par un statisticien anglais, Bayes, au 18ème siècle mais il n'a jamais publié l'ouvrage. Après sa mort, ses notes ont été éditées par et par le mathématicien Richard Price. Le théorème est donné par la formule suivante :

$$P(A\backslash B) = \frac{p(B\backslash A)p(A)}{p(B)}$$

- A et B sont des événements.
- P (A) est la probabilité d'observer l'événement A
- P (B) est la probabilité d'observer l'événement B.
- P (A | B) est la probabilité conditionnelle d'observer A, sachant qu'un autre événement B de probabilité non nulle s'est réalisé.

Dans un problème de classification, notre tâche est de trouver l'étiquette la plus probable A, étant donné les caractéristiques B, le théorème de Bayes devient :

$$P(y|x_1, \dots., x_n) = \frac{P(x1,\dots.,x_n\backslash y)P(y)}{P(x_1,\dots.,x_n)}$$

Où n représente le nombre de caractéristiques, y est l'évènement qu'on cherche à classer.

Par conséquent, en tenant compte de l'hypothèse d'indépendance, Bayes est la classe qui constitue la probabilité la plus élevée [29].

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i\backslash y)$$

- **K-Nearest Neighbor(KNN)**

Le principe de la méthode KNN est de trouver les k plus proches voisins, à partir de l'échantillon d'apprentissage, d'une nouvelle instance que l'on cherche à classer. La classe de la nouvelle instance est la classe majoritaire (la plus représentative) de ces k voisins. Dans le cas d'une régression, la valeur de sortie est une valeur continue, peut-être, par exemple, la moyenne des valeurs des k voisins. Il existe plusieurs fonctions pour calculer la distance entre deux voisins, notamment la distance euclidienne, la distance de Manhattan, la distance de Minkowski, la distance de Jaccard, etc.

Il convient de noter que, KNN n'a pas la phase d'apprentissage pour laquelle le modèle d'exploration de données a été généré. Les exemples d'apprentissage sont des vecteurs dans l'espace multidimensionnel avec l'étiquette de la classe à laquelle ils appartiennent, qui sont stockés en permanence dans la mémoire pendant la phase de classification [30].

Figure II.6-*K-Nearest Neighbor (KNN).*

- **Support Vector Machine (SVM)**

Le classificateur SVM [30], développé par Vladimir Vapnik en 1995, est un classificateur puissant, il a fait ses preuves dans plusieurs domaines. Le principe est de projeter des données non linéairement séparables dans un autre espace de dimension supérieure, là où elles peuvent se trouver, en utilisant des noyaux différents. Le but du SVM binaire est de trouver un hyperplan optimal qui sépare les deux couches en maximisant la distance. Cette distance s'appelle la

marge. Dans le cas de la classification binaire, l'hyperplan est une droite. Les points les plus proches, utilisés uniquement pour déterminer l'amplitude, sont appelés les vecteurs de support**.**

L'hyperplan divisé est représenté par l'équation :

$$H(x) = w^T + b$$

W est un vecteur à m dimensions et b est le terme. La fonction de décision, pour un exemple x, peut être exprimée comme suit :

$$\begin{cases} Classe = 1 \; Si \; H(x) > 1 \\ Classe = -1 \; Si \; H(x) < -1 \end{cases}$$

Maximiser la marge équivaut à maximiser $\frac{2}{\|W\|}$ et qui est le plus petit $\|W\|$ [30].

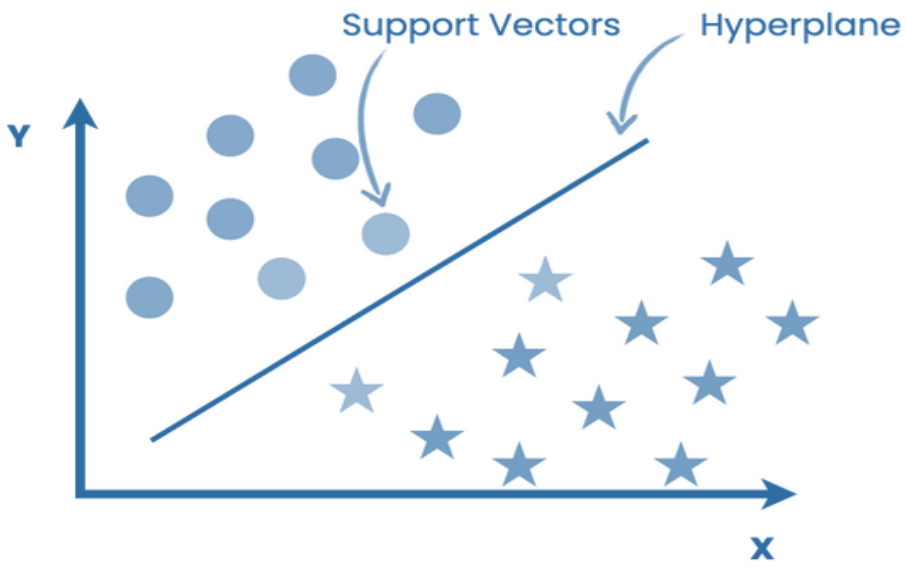

Figure II.7-Support Vector Machine (SVM).

SVM réduit le problème multicouche à un composant de plusieurs sur ensembles bicouches afin que les limites de décision puissent être tracées entre différentes couches. Il décompose l'ensemble d'exemples en plusieurs sous-ensembles, chacun représentant un problème de classification binaire, chaque hyperplan divisé a été déterminé en utilisant la méthode binaire SVM. La hiérarchie du super plan binaire construit dans le classifieur est passée de la racine à la feuille pour décider du type d'une nouvelle instance [30].

II.3.2.2 L'apprentissage non Supervisé

Dans l'apprentissage automatique, l'apprentissage non supervisé (ou apprentissage non supervisé) est inclus dans les modèles d'apprentissage, sans qu'il soit nécessaire d'étiqueter manuellement ou automatiquement les données au préalable. Les algorithmes regroupent les données en fonction de leur similarité sans aucune intervention humaine.

L'apprentissage non supervisé détecte des données ou des individus ayant des caractéristiques ou des modèles communs. En règle générale, l'apprentissage non supervisé peut être utilisé pour développer un moteur de recommandation de produits, conçu pour fournir des produits aux clients en fonction des préférences des clients présentant des caractéristiques similaires [30].

II.3.2.2.1 Clustring

Un problème de clustering est un problème où la machine est censée assembler en groupe (cluster) les objets présents dans des groupes de données et ceci de la manière la plus juste et la plus efficace possible. Par exemple, cette technique, bien que parfois difficile à comprendre pour les humains, est largement utilisée dans le domaine du marketing pour organiser différents clients en groupes. Un exemple d'algorithme très couramment utilisé dans le clustering est K-means [31].

II.3.2.2.2 Association

Le système d'association permet de trier et de regrouper des données qui peuvent être liées grâce à certaines caractéristiques. Par conséquent, le but est de trouver des objets connexes qui ne sont pas identiques. Par exemple, en alimentant l'algorithme de multiples images de chats et d'accessoires pour chats, l'algorithme d'apprentissage non supervisé ne regroupera pas tous les chats, mais par exemple une pelote de laine avec un chat. Un exemple d'algorithme très couramment utilisé en association est l'algorithme à priori [31].

II.3.4 Deep Learning

L'apprentissage en profondeur est une méthode d'IA dérivée du concept d'apprentissage automatique. Cette méthode dite d'apprentissage en profondeur repose sur un concept plus spécifique de réseau de neurones artificiels.

De structure non linéaire, un réseau de neurones artificiels se présente sous la forme d'un réseau d'ensembles d'unités d'exécution d'informations (représentant des neurones) empilées les unes sur les autres et associées entre elles par des connexions (synapses). À partir de là, il traite

les informations à travers les schémas de propagation d'activité de ces unités, qui fonctionnent au-delà d'un certain seuil.

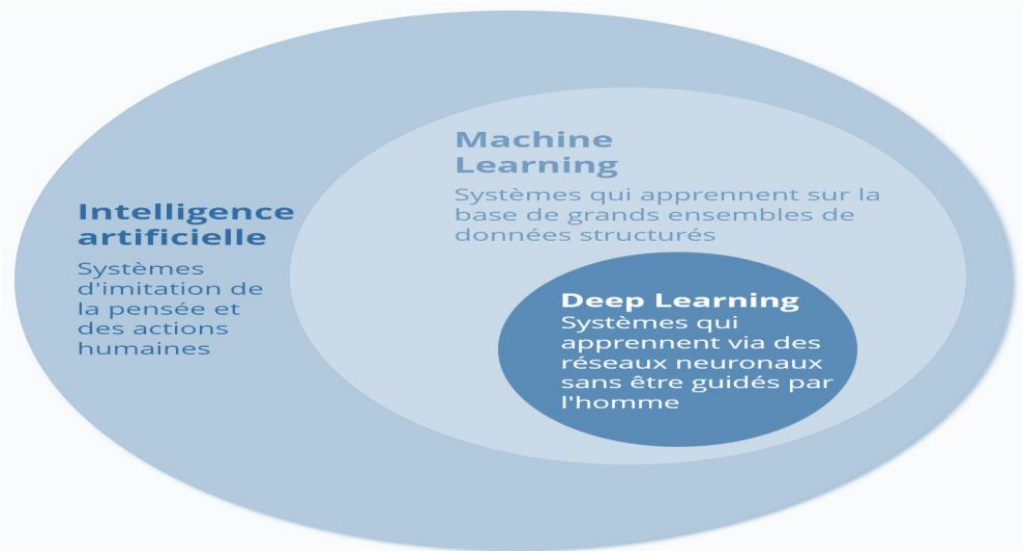

Figure II.8-*Deep Learning.*

L'apprentissage en profondeur peut être considéré comme une nouvelle étape dans le développement de l'intelligence artificielle. Concernant son origine, ce dernier se contente de respecter des règles prédéterminées basées sur un modèle cognitif. L'intervention d'un programmeur est alors encore nécessaire pour parfaire le système ou intégrer d'autres fonctions ou de nouvelles règles.

À l'instar de la machine Learning statistique, le deep Learning rend l'IA autonome en lui permettant d'intégrer seule de nouvelles règles. L'amélioration exponentielle de la puissance de calcul et le développement d'applications associées permettent au deep Learning de générer des couches de neurones de plus en plus complexes et denses [32].

II.3.4.2 modèle deep Learning pour la classification de texte

II.3.4.2.1 RNN

RNN[4] est une autre architecture de réseau neuronal abordée par les chercheurs en classification et en exploration de texte.

RNN attribue plus de poids aux points de données précédents de la séquence. Par conséquent, cette technique est une méthode puissante pour classer les données textuelles, de chaîne et

[4]RNN : Recurrent Neural Network.

séquentielles. Dans un RNN, le réseau de neurones considère les informations de nœud précédentes d'une manière très complexe qui permet une meilleure analyse sémantique des structures dans l'ensemble de données [33].

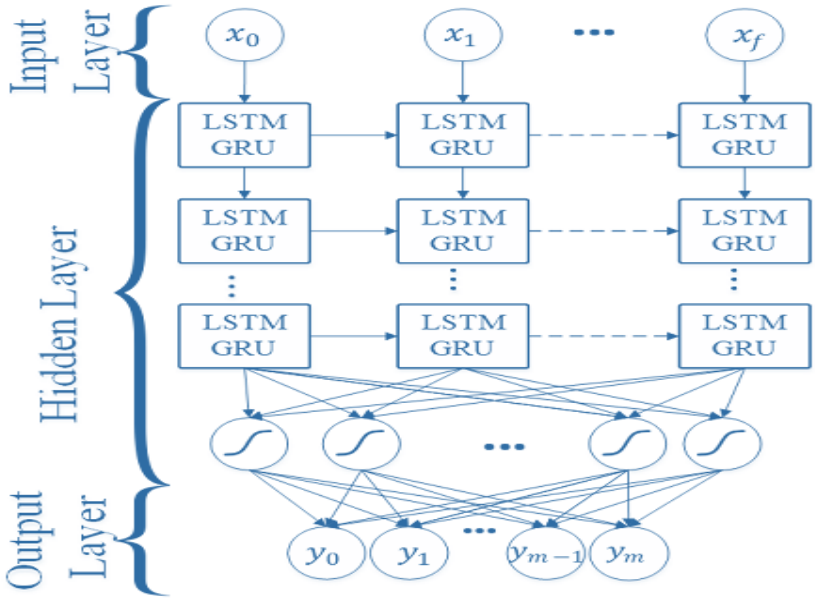

Figure II.9-classification de texte avec RNN [33].

- **Gated Recurrent Unit (GRU)**

  Gated Recurrent Unit (GRU) est un mécanisme d'activation RNN introduit par J. Chung et al. Et K.Cho et al. Le GRU est une variante simplifiée de l'architecture LSTM, mais présente les différences suivantes : Le GRU contient deux portes et pas de mémoire interne [33].

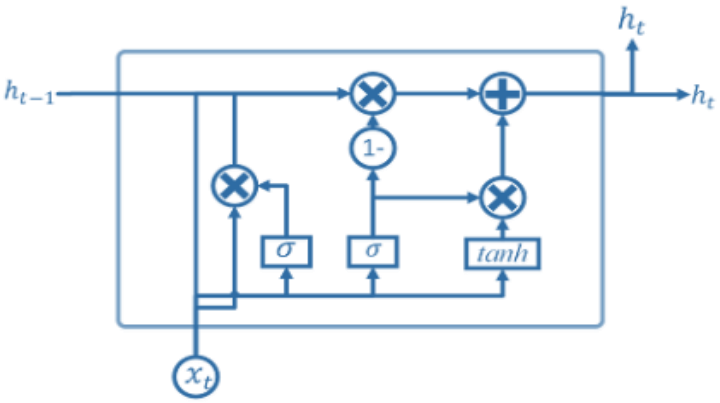

Figure II.10-*Gated Recurrent Unit (GRU) [33]*.

- **Long Short-Term Memory (LSTM)**

  Pour résoudre ces problèmes, la mémoire à long terme à court terme est un type spécial de RNN dont la dépendance à long terme est en quelque sorte plus efficace que les RNN de base. Ceci est particulièrement utile pour dépanner les fuites car les LSTM utilisent plusieurs ports pour réguler la quantité d'informations qui sera autorisée dans chaque état de nœud [33].

  La figure ci-dessous montre la cellule de base d'un modèle

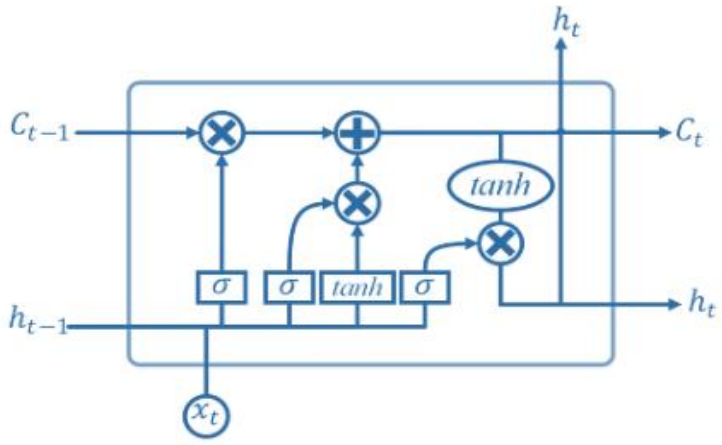

Figure II.11-Long Short-Term Memory (LSTM*) [33]*.

II.3.4.2.2 CNN

CNN[5] est une autre architecture intensive utilisée pour la classification hiérarchique des documents. Bien qu'initialement conçus pour un traitement d'image similaire à celui du cortex visuel, les CNN sont également efficaces dans la classification de texte. Dans CNN de base pour le traitement d'image, l'étireur d'image est transformé avec un ensemble de taille de noyau dx d. Ces couches convolutives sont appelées cartes de caractéristiques qui peuvent être empilées pour fournir plusieurs filtres sur l'entrée. Pour réduire la complexité de calcul, les CNN utilisent une disposition commune pour réduire la taille du mot d'une couche à la suivante dans le réseau. Différentes techniques de mise en commun sont utilisées pour les sorties tout en préservant des fonctionnalités importantes.

La méthode de regroupement la plus courante est le regroupement maximal où l'élément maximal est sélectionné dans la fenêtre de regroupement. Pour fournir la sortie groupée des cartes d'entités empilées pour la couche suivante, les cartes sont aplaties dans une colonne. Les

[5]CNN : Convolution Neural Network

canapés finales d'un CNN sont généralement des canapés entièrement connectés. Habituellement, lors de l'étape de rétro propagation d'un réseau de neurones accumulatif, non seulement les poids sont ajustés, mais également les filtres de détection de caractéristiques. Un problème potentiel avec CNN utilisé pour le texte est le nombre de "canaux", Sigma (taille de l'espace objet). Cela peut être énorme (par exemple 50K), pour le texte, mais pour les images, cela pose moins de problème (par exemple, seulement 3 canaux RVB). Cela signifie que la taille du CNN pour le texte est très élevée.

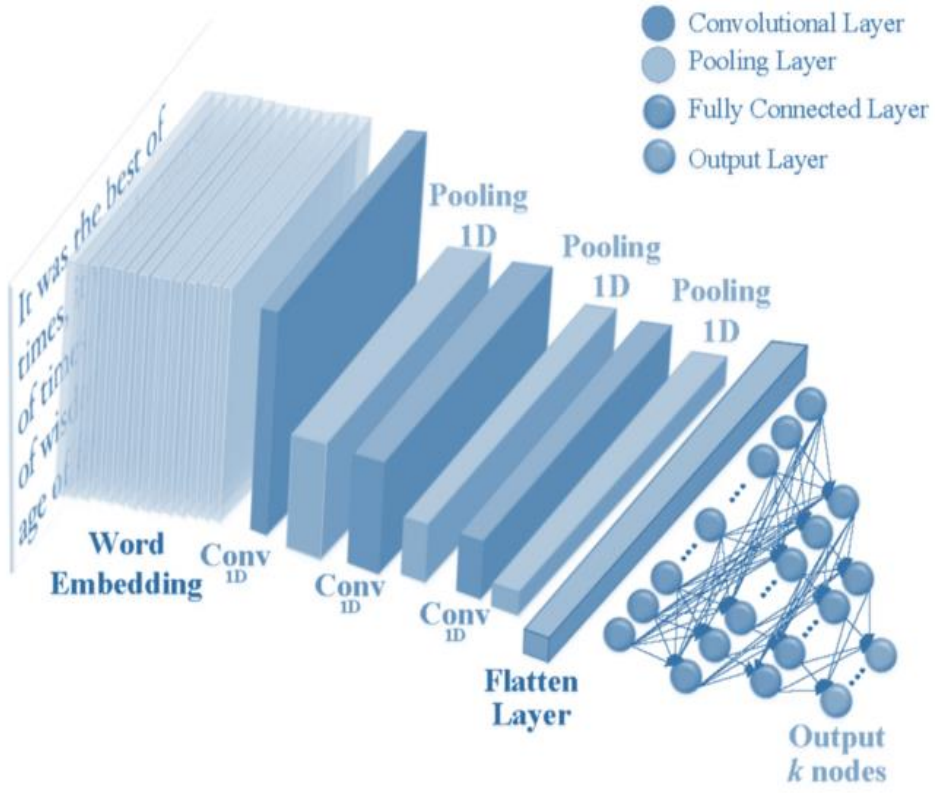

Figure II.12-*classification de texte avec CNN [33]*.

La classification de texte est un problème fondamental dans le traitement du langage naturel. En tant que modèle d'apprentissage en profondeur populaire, le modèle neuronal convolutif s'est avéré être un grand succès dans cette tâche. Cependant, la plupart des modèles CNN existants appliquent des transformations de taille de fenêtre fixe, ce qui empêche une compréhension flexible des caractéristiques de transformation de n-grammes.

Les connexions denses créent des raccourcis entre les blocs d'intégration en amont et en aval, permettant au modèle de générer des caractéristiques à plus grande échelle à partir de caractéristiques à plus petite échelle et ainsi de générer des objets variables n-gram.

De plus, l'attention portée aux caractéristiques multiniveaux est développée pour sélectionner de manière adaptative les caractéristiques multiniveaux pour la classification. Des tests approfondis démontrent que notre modèle fonctionne de manière compétitive par rapport aux meilleures lignes de base sur cinq ensembles de données de référence. La visualisation de l'attention révèle en outre la capacité du modèle à sélectionner les caractéristiques N-gram appropriées pour la classification du texte [33].

II.3.4.2.3 RCNN

RCNN[6] sont également utilisés pour la classification de texte. L'idée principale de cette technique est de capturer des informations contextuelles avec une structure répétitive et de construire une représentation textuelle à l'aide d'un réseau de neurones intégré. Cette architecture est une combinaison de RNN et CNN pour utiliser les avantages des deux techniques dans un seul modèle [33].

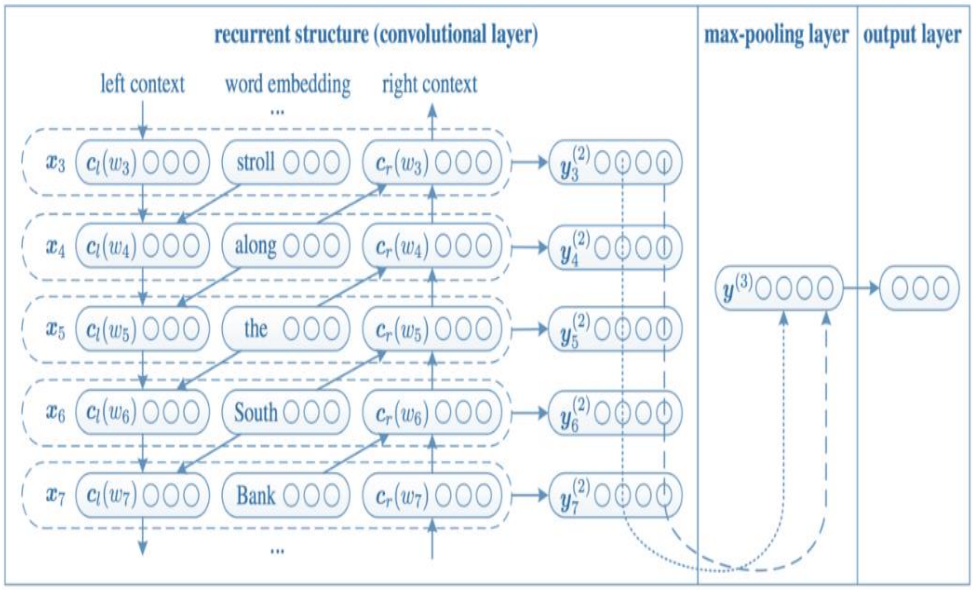

Figure II.13-*classification de texte avec RCNN.*

II.3.4.2.5 Mécanisme d'attention

Un réseau de neurones cyclique fait des prédictions à partir de données séquentielles. Chaque élément fournit des informations que le réseau utilise pour générer une sortie. Cependant, il est possible que certains éléments de la séquence contiennent plus d'informations utiles à la

[6]RCNN : Reccurent Convolution Neural Network

prédiction. Le but du mécanisme d'attention est de déterminer quel élément est le plus utile. Exemple : proposer une architecture RNN de type codeur-décodeur avec mécanisme d'attention. L'encodeur est un modèle à deux composants qui permet le mappage entre les séquences d'entrée et de sortie. La partie codeur est chargée de représenter la chaîne d'entrée sous la forme d'un vecteur de taille fixe, tandis que la partie décodeur utilise cette représentation pour générer la chaîne de sortie. Par conséquent, le rôle du mécanisme d'attention est d'attacher de l'importance aux positions des éléments dans la séquence d'entrée. Pour ce faire, un poids est attribué à chaque élément encodé en entrée. Ces pondérations sont ensuite mises à jour après la génération de chaque sortie. Le décodeur calcule alors un vecteur de contexte en générant une somme, pondérée par les éléments codés. Ce vecteur de contexte permet au décodeur d'accorder plus d'attention à certains éléments pour produire sa sortie. Pour chaque sortie produite par le décodeur, le vecteur de contexte est recalculé [34].

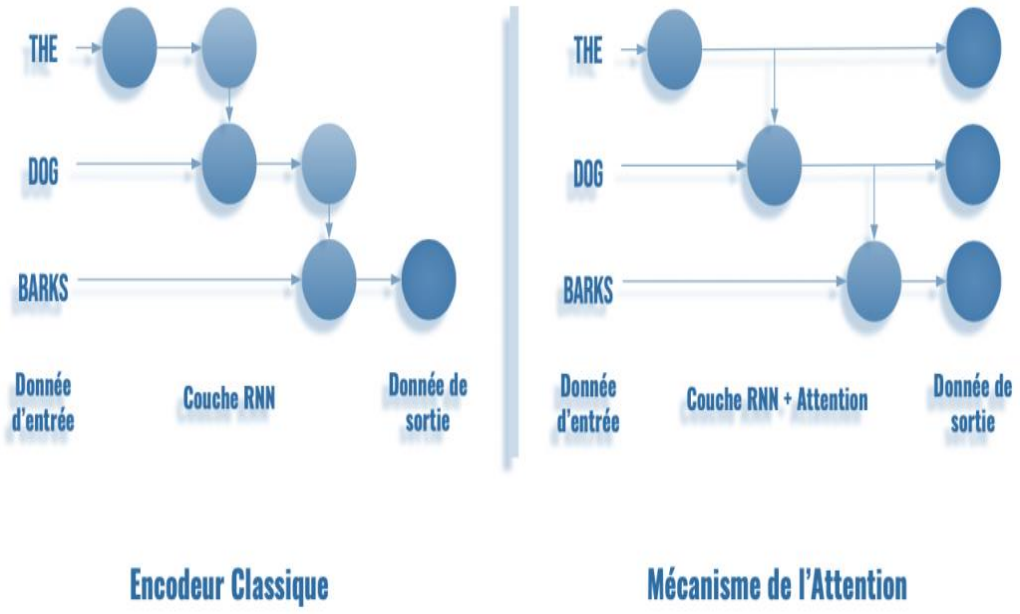

Figure II.14-*Mécanisme d'attention.*

II.3.4.2.5 Transformers

Les Transformers fournissent des milliers des modèles pré-entraînés pour effectuer des tâches de traitement sur des corpus de textes : la classification, l'extraction d'informations, la réponse aux questions, le résumé/la synthèse, la traduction, et ce dans une centaine de langues. Parmi ces modèles, les plus populaires sont BERT (Devlin et al, 2018), [35].

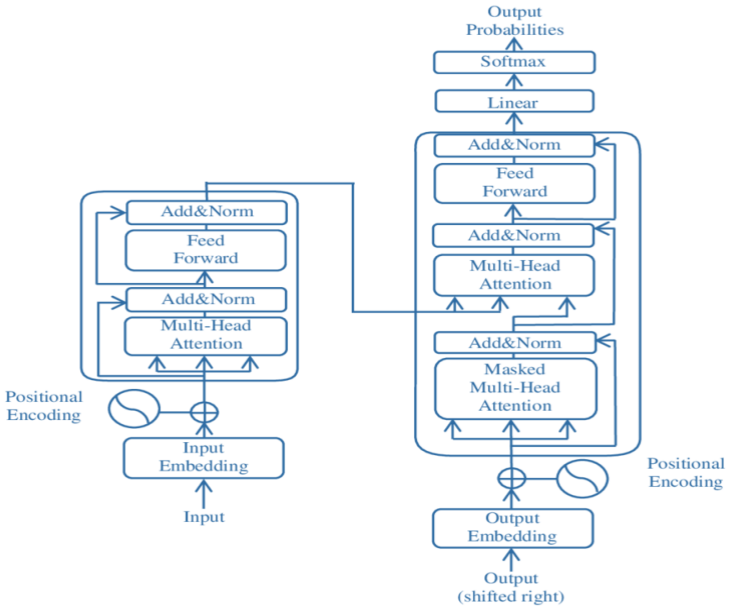

Figure II.15-*Architecture globale des Transformers.*

II.3.4.2.6 Bert

Le développement du BERT[7] [36] s'est multiplié dans divers domaines au fil du temps. Descendant de l'architecture Transformer, BERT est une représentation d'encodeur bidirectionnelle où le modèle apprend à prédire le contexte de différentes manières. Les modèles BERT sont fortement pré-entraine sur des millions et des milliards de textes non annotés, ce qui nous permet d'affiner le modèle sur des tâches personnalisées et avec des ensembles de données spécifiques. La préformation ne nécessite pas d'apprentissage à partir de zéro, BERT peut atteindre une grande précision avec un temps de calcul amélioré grâce à l'apprentissage par transfert. BERT est open source et largement étudié pour faire des prédictions avancées. Depuis sa sortie, plusieurs versions alternatives ont été lancées. La technologie BERT est devenue un cadre révolutionnaire pour de nombreuses tâches de traitement du langage naturel telles que l'analyse sentimentale, la prédiction de phrases, le résumé abstrait, la réponse aux questions, l'inférence en langage naturel et bien d'autres. BERT à différentes configurations de modèles, l'une est BERT Base, le modèle le plus basique avec 12 couches d'encodeur. Puis modèle BERT Large avec un nombre supplémentaire de couches. Au fil du temps, de nombreux nouveaux modèles ont été inspirés par l'architecture BERT mais sont formés dans différents langages ou optimisés sur des ensembles de données spécifiques à un domaine. Il y a des progrès continus et des versions plus optimisées sont souvent introduites

[7] BERT : Bidirectional Encoder Representations from Transformers.

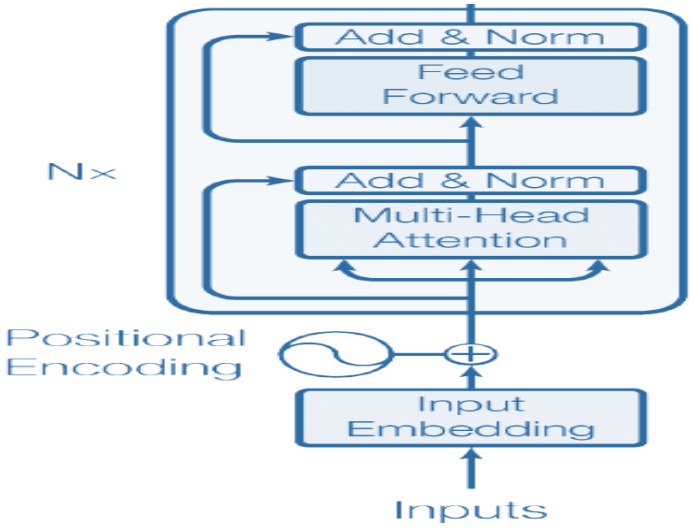

Figure II.16-*Architecture globale de BERT*

- **Variantes BERT**

BERT a inspiré de nombreuses variantes [37] :

Variantes BERT	Formulaire complet	Résume
ALBERT	« A Lite BERT»	Le modèle ALBERT a 12 millions de paramètres avec 768 couches cachées et 128 couches intégrées. Comme prévu, le modèle plus léger a réduit le temps de formation et le temps d'inférence.
RoBERTa	« Robustly Optimized BERT pre-training Approach»	L'une des variantes les plus populaires qui est tout à fait comme BERT. La principale différence étant étapes de pré-formation légèrement différentes.
ELECTRA	« Efficiently Learnning an Encoder thet Classifies Token Replacements Accurately»	Au lieu d'utiliser MLM, ce modèle est pré- formé à l'aide d'une nouvelle tâche appelée "remplacé tâche de détection de jeton".
SpanBERT	\	Principalement utilisé pour prédire une "portée" de texte. Le modèle est largement utilisé pour les questions réponse, extraction de relation, etc
DistiBERT	« Distillation BERT »	DistilBERT est un petit modèle de transformateur rapide, bon marché et léger basé sur l'architecture BERT. La distillation des connaissances est effectuée pendant la phase de pré-apprentissage pour réduire de 40 % la taille d'un modèle BERT.

Tableau II.1-*Les variantes BERT.*

II.4 Conclusion

Dans ce chapitre, nous avons présenté différentes techniques d'apprentissage automatique pour la classification de textes. Nous nous sommes concentrés particulièrement sur le type supervisé. Nous avons ensuite introduit l'aspect révolutionnaire du Deep Learning ainsi que certains modèles qui vont être utiliser dans notre proposition. Dans le chapitre suivant, nous présenterons notre approche pour améliorer la découverte des services web.

Chapitre III

Notre approche pour améliorer la découverte des services web

Sommaire

III.1 Introduction

Comme toute technologie, les services web génèrent beaucoup d'intérêt et promettent de nombreux avantages. Ils fournissent des fonctions accessibles via des protocoles web standards qui permettent aux entreprises d'échanger des données à distance et de naviguer facilement entre les entreprises, leurs partenaires et leurs clients. La classification des services Web est le processus d'organisation d'un ensemble de services Web selon une classification et des critères fonctionnels similaires. Il peut faciliter, optimiser et automatiser l'efficacité et l'efficience des processus de découverte, de composition, d'exécution et de gestion des services Web. Dans ce travail, nous nous concentrons sur les modèles de classification basés sur l'apprentissage profond pour améliorer le processus de découverte des services web.

III.2 Les travaux connexes

III.2.1 DeepWSC: A Novel Framework with Deep Neural Network for Web Service Clustering

Les auteurs de [38] en 2019 proposent un nouveau cadre avec un réseau de neurones profonds, appelé DeepWSC, qui combine les avantages des neurones récurrents réseau et réseau neuronal convolutif pour regrouper les services Web grâce à l'extraction automatique par notre RCNN amélioré qui est un service de réseau neuronal profond extracteur de fonctionnalités formé sur la base d'un modèle de sujet probabiliste. Les résultats démontrent que DeepWSC surpasse les approches de pointe pour le clustering de services Web en termes de multiples métriques d'évaluation.

III.2.2 DeepWSC: Clustering Web Services via Integrating Service Composability into Deep Semantic Features:

Les auteurs de [39] en 2020 proposent un nouveau Framework basé sur la méthode d'analyse DeepWSC pour le clustering de services web. Il intègre des fonctionnalités sémantiques profondes extraites des descriptions de service à l'aide de réseaux de neurones cumulatifs itératifs avancés et des fonctionnalités de faisabilité de service obtenues à partir des relations d'appel de traduction. Service à l'aide d'un réseau intégré de graphes signés, pour créer conjointement des fonctionnalités embarquées pour les services Web en cluster. Tests approfondis ont été effectués sur 8 59 services Web réels. Les résultats du test démontrent que

DeepWSC surpasse les meilleures approches de clustering de services Web dans plusieurs métriques.

III.2.3 Combination of ELMo Representation and CNN Approaches to Enhance Service Discovery

Les auteurs de [40] en 2020 proposent une approche efficace qui combine la représentation des Embeddings from Language Models (ELMo) et le Convolutional Neural Network (CNN) pour obtenir un score de similarité plus précis pour récupérer les services Web cibles. Plus précisément, premièrement, l'étude adopte le modèle ELMo pour générer des représentations de mots efficaces pour capturer les informations suffisantes à partir des services et des requêtes. Ensuite, les représentations de mots sont utilisées pour composer une matrice de similarité, qui sera considérée comme l'entrée du CNN pour apprendre les relations de correspondance. Enfin, la combinaison de la représentation ELMo et du CNN est utilisée pour traiter les processus de représentation et d'interaction au sein de la tâche d'appariement afin d'améliorer les performances de découverte de service.

III.2.4 Web Services Classification Based on Wide & Bi-LSTM Model

Les auteurs de [41] en 2019 proposent une méthode de classification des services Web basée sur le modèle Wide & Bi-LSTM, tout d'abord, toutes les caractéristiques discrètes dans les documents de description des services Web sont combinées pour effectuer la prédiction étendue de la catégorie de service Web en exploitant le modèle d'apprentissage large. Deuxièmement, l'ordre des mots et les informations de contexte des mots dans les documents de description des services Web sont extraits en utilisant le modèle Bi-LSTM pour effectuer la prédiction de profondeur de la catégorie de service Web. Troisièmement, il utilise l'algorithme de régression linéaire pour intégrer les résultats de prédiction de largeur et de profondeur des catégories de services Web comme résultat final de la classification des services. Enfin, par rapport à six méthodes de classification des services Web basées sur TF-IDF, LDA, WE-LDA, LSTM, Wide&Deep et Bi-LSTM, respectivement, les résultats expérimentaux montrent que l’approche atteint une meilleure performance dans la précision de la classification des services Web.

III.2.5 A new cloud-based classification methodology (CBCM) for efficient semantic web service discovery

Les auteurs de [42] en 2021 proposent la méthodologie de classification basée sur le Cloud car elle présente une nouvelle méthodologie basée sur la classification des services Web sémantiques. En outre, le Cloud computing est utilisé non seulement pour stocker, mais également pour allouer la grande échelle des services Web avec à la fois une disponibilité et une accessibilité élevées. Les résultats expérimentaux utilisant la méthodologie suggérée montrent une meilleure performance du système proposé en ce qui concerne à la fois la précision et l'exactitude par rapport à la plupart des méthodes discutées dans la littérature de l'étude actuelle.

III.2.6 A Novel Dual-Graph Convolutional Network based Web Service Classification Framework

Les auteurs de [43] en 2020 proposent un cadre à double GCN qui peut supprimer efficacement la propagation du bruit des contenus textuels en distinguant les documents de description fonctionnelle et d'autres sources d'informations (en particulier les modèles de Co-invocation Mashup-API par défaut dans cet article) pour la classification des API. Ce cadre est extensible avec la possibilité d'inclure différentes sources d'informations accumulées dans l'écosystème de l'API Web. Des expériences complètes sur un ensemble de données publiques du monde réel démontrent que notre méthode proposée peut surpasser diverses méthodes représentatives pour la classification des API.

III.3 Notre approche

Vu la nature textuelle des descriptions des services web du Dataset, que nous allons utiliser pour notre étude, nous allons essayer de faire rentabiliser des modèles Deep Learning de classification de texte, à savoir *RCNN* et *BERT*.

III.3.1 Les étapes de notre travail

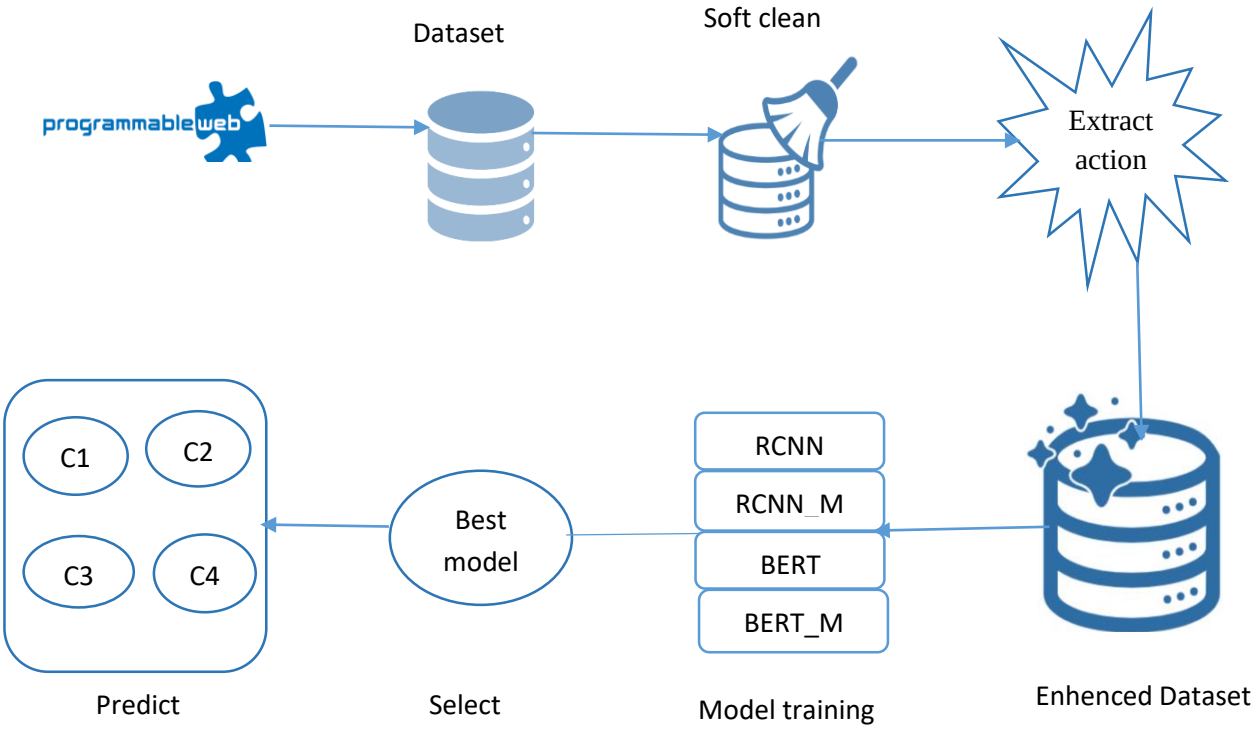

Figure III.1-*Etapes de notre travail.*

III.3.1.1 Extraire des actions

Dans notre approche proposée qui concerne la découverte des services Web, nous avons utilisé des services Web réels issus du répertoire programmableWeb.com, nous avons accentué l'apprentissage, on intègre les actions pour améliorer le résultat de la découverte des services web.

Algorithme : Extraire des actions

```
Paramètres : description textuelle du service Web
Résultat : liste d'actions [ ]
  function SGet Action(Sentence)
   if (Item is V erb) and (has dobj) then
     Concatenate the verb with dobj ;
     return as Action;
   else
     if (Item is V erb) and (has conjuction) then
        go to the conjunction item;
        once you reach the Noun: concatenate the V erb with the Noun;
        if (this Noun has conjuction) then
          concatenate also the V erb with the next Noun;
        end if
     end if
     return Actions;
   end if
  end function
```

```
  procedure Get Action(Sentence)
   if (Item is V erb) and (has xcomp) then
      go to the xcom node;
      concatenate the V erb with each action of SGet_Action (xcomp);
      Add all as Actions to the list of Actions[];
   else
      Add all Actions of SGet Action (Sentence) to the list of Actions [];
   end if
  end procedure
  for each sentence in Web service textual description do
     Get Action(Sentence) ;
  End for
```

Xcomp: An open clausal complement.

Dobj: The direct object.

Examples:

Case 1:

- Description: Service X translates documents.
- Actions: [translates documents].

Case 2:

- Description: Service X translates paragraphs and documents.
- Actions: [translates paragraphs, translates documents].

Case 3:

- Description: Service X uses FacebookApi to subscribe users.
- Actions: [uses facebookApi, uses subscribe users].

III.3.1.2 Préparation des données

Notre Dataset est extrait du site «programmable-Web». Ensuite, nous effectuons un nettoyage superficielle sur Dataset pour supprimer les caractères spéciaux et le bruit, après avoir effectué l'algorithme de Extraire des actions, à la fin, nous extrayons une Dataset amélioré.

Après avoir extrait une Dataset optimisé, nous avons développé des modèles de Deep Learning basés sur *RCNN* et *BERT* pour avoir le test, à la fin nous avons sélectionné le meilleur modèle pour faire la prédiction sur 20 classes des services web.

III.3 Discussion des résultats

III.3.1 Dataset

Notre ensemble des données issu du monde réel, dont les données ont été fournies d'un annuaire bien connu « programmableweb.com », qui contient 8459 services Web. Nous avons mené les expérimentations sur les 20 meilleures catégories. Le nombre moyen de mots dans la description de service est de 38,68 mots. Le nombre de services Web dans chaque catégorie sont répertoriés dans Tableau III.2.

Nom de la classe	#services	Nom de la classe	#services
Tools	887	*Telephony*	342
Financial	757	*Security*	312
Messaging	591	*Reference*	304
ECommerce	553	*Email*	299
Payments	553	*Search*	290
Social	510	*Travel*	294
Enterprise	509	*Video*	281
Mapping	429	*Education*	277
Government	371	*Advertising*	274
Science	357	*Transportation*	269

Tableau III.1-*Répartition du nombre de services Web dans 20 catégories.*

Dataset que nous avons utilisé est devisé en deux parties (train, test) :

- train : pour entrainer les modèles.
- test : pour testes les modèles.

api_id	api_name	api_prim_cate	api_desc	api_pc_name	api_sc_name	api_orignal_desc	actions	d_verbs	av_comb
66271	Freshmeat	12	freshmeat maintain web' largest unix cross-pla...	Reference	['Open Source', 'Application Development']	freshmeat maintains the webs largest index of ...	['added deals', 'including', 'screenshots', 'u...	['maintains', 'released', 'are', 'cataloged', ...	['maintains', 'released', 'are', 'cataloged', ...
69387	Heello	5	heello social microblog servic user share shor...	Social	['Messaging', 'Blogging']	heello is a social microblogging service that ...	['allows', 'share', 're', 'doing what', 'calle...	['is', 'allows', 'theyre', 'doing', 'called', ...	['is', 'allows', 'theyre', 'doing', 'called', ...
65336	Cue	14	name greplin cue user search data store web - -...	Search	NaN	previously named greplin cue lets users search...	['named', 'added', 'added', 'allowing', 'becom...	['named', 'lets', 'search', 'store', 'blog', '...	['named', 'lets', 'search', 'store', 'blog', '...
193207	US County Boundary	7	counti boundari api border counti unit develop...	Mapping	['Geography', 'Images']	the us county boundary api shows the borders o...	['shows borders', 'allows', 'shade', 'shade', ...	['county', 'shows', 'allows', 'select', 'shade...	['county', 'shows', 'allows', 'select', 'shade...
150765	PinnacleCart RPC	3	pinnaclecart rpc api develop integr shop cart ...	eCommerce	NaN	the pinnaclecart rpc api allows developers to ...	['allows', 'integrate services', 'includes', '...	['allows', 'integrate', 'shopping', 'includes'...	['allows', 'integrate', 'shopping', 'includes'...

Figure III.2-*Dataset utilisées de notre approche.*

III.3.2 Métriques d'évaluation

Nous évaluons les performances de nos modèles à l'aide de quatre (04) mesures d'évaluation largement utilisées : purity, NMI, Précision, F1-Score.

III.3.2.1 Purity

Calcule la proportion de services correctement regroupés le nombre total de services et est calculé comme suit [44] :

$$Purity(B, A) = \frac{1}{n}\sum_{k} max|B_k \cap A_r|$$

III.3.2.2 NMI

L'information mutuelle normalisée (NMI) est une normalisation du score d'information mutuelle (MI) pour mettre les résultats à l'échelle entre 0 (aucune information mutuelle) et 1 (corrélation parfaite) [45].

III.3.2.3 Recall

Il s'agit du nombre exact de résultats positifs divisé par le nombre de tous les échantillons pertinents [46].

$$\text{Recall} = \frac{positifs\ corrects}{positifs\ corrects + negative\ incorect}$$

III.3.2.4 F1Score

Le score F1 est une moyenne harmonisée de précision et de recall. La plage du score F1 est [0, 1]. Il vous indique la précision de votre classificateur (combien de cas il classe correctement), ainsi que sa robustesse (il ne manque pas un nombre significatif de cas).Une précision élevée mais une récupération plus faible vous donne une précision extrême, mais manque alors beaucoup de cas difficiles à classer. Plus le score F1 est élevé, meilleures sont les performances de notre modèle. Mathématiquement, il peut être exprimé comme suit [46] :

$$F1 = 2 * \frac{1}{\frac{1}{précision} + \frac{1}{reccal}}$$

III.3.2 Implémentation

Tous les tests ont été réalisées su Saturn cloud avec (2XLarge – 8 cores -64 GB RAM –40Gi Disk)

III.3.2.1 Saturn cloud

Saturn Cloud est une plate-forme de science des données qui aide les gens à se mettre au travail rapidement en utilisant la technologie dont ils ont besoin, y compris l'informatique à haute mémoire, les processeurs GPU et les clusters Dask. Vous pouvez utiliser Saturn Cloud avec Python, R ou presque n'importe quel autre langage de programmation [47].

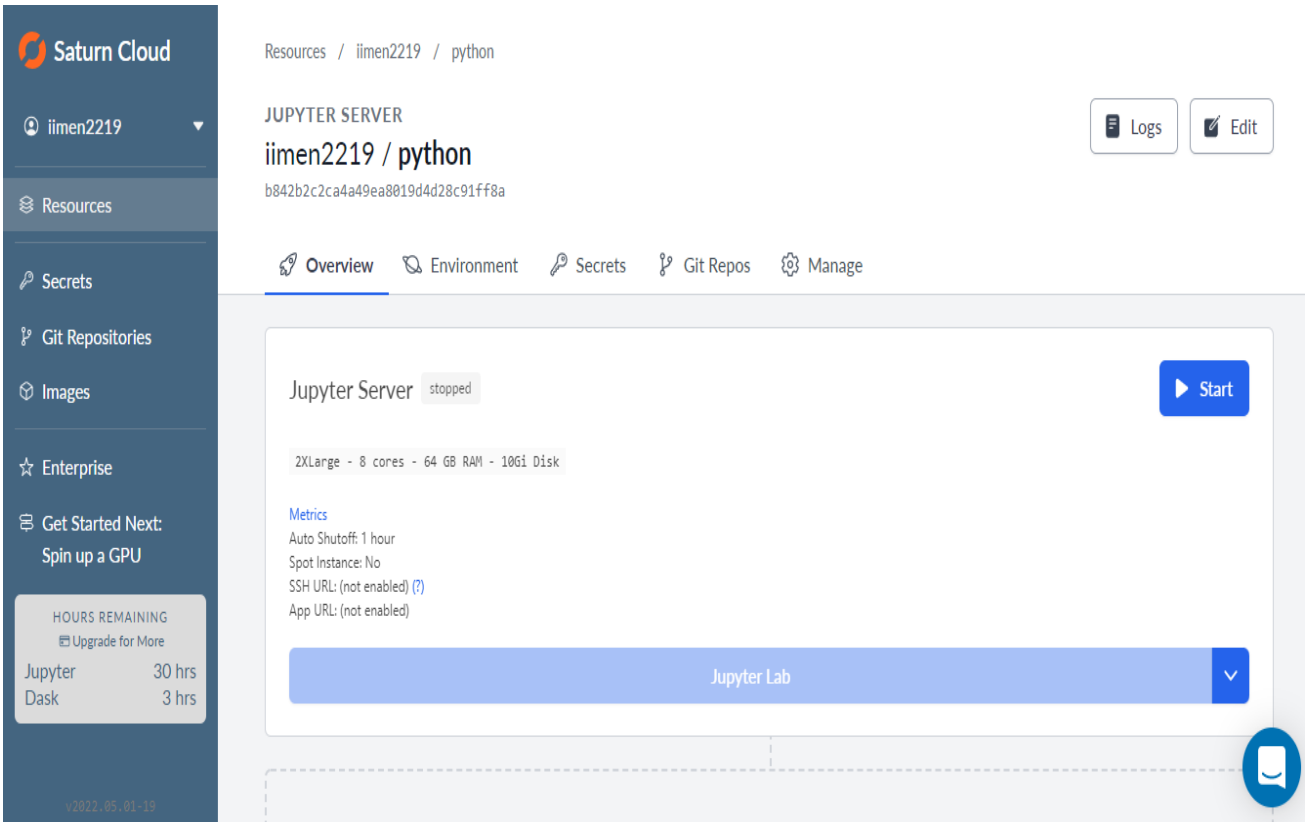

Figure III.3-*Plate-forme de Saturn Cloud.*

III.3.3 Résultats

Après plusieurs exécutions, nous avons identifié les paramètres qui retournent la meilleure prédiction pour notre modèle de classification « multi-class». Dans ce qui suit, nous allons présenter et argumenter le choix de ces paramètres.

III.3.3.1 RCNN

Nous avons utilisé l'algorithme de RCNN pour obtenir un modèle de classification, ce modèle a utilisé les descriptions simples de service Web et pour faire les prédictions, nous avons utilisé la colonne (Api_prim_cate), puis nous utilisons les paramètres suivants pour obtenir le résultat :

- Max_length =433
- Batch_size=32
- Learning_rate= 0.0009
- Nombre epoch=5
- Glove= glove.6B.100d.txt

- **Le code de l'algorithme RCNN**

```
history=model_RCNN.fit(X_train_Glove, y_train,
                       validation_data=(X_test_Glove, y_test),
                       epochs=5,
                       batch_size=32,
                       verbose=2

                   )
```

- **Résultats d'exécution**
  - **NMI et Purity**

```
cluster_nmi(predicted, y_true)
```
Last executed at 2022-06-11 11:45:31 in 37ms

```
0.5704676261819446
```

```
purity(y_true, predicted)
```
Last executed at 2022-06-11 11:45:31 in 33ms

```
0.6438140267927502
```

Figure III.4- NMI et Purity de modèle RCNN.

- **Rapport de classification**

	precision	recall	f1-score	support
0	0.5031	0.6090	0.5510	133
1	0.6923	0.8684	0.7704	114
2	0.7684	0.8202	0.7935	89
3	0.8542	0.4940	0.6260	83
4	0.6881	0.9036	0.7813	83
5	0.5464	0.6974	0.6127	76
6	0.4432	0.5132	0.4756	76
7	0.6984	0.6875	0.6929	64
8	0.6087	0.7500	0.6720	56
9	0.7544	0.7963	0.7748	54
10	0.5682	0.4902	0.5263	51
11	0.4545	0.1064	0.1724	47
12	0.3333	0.0652	0.1091	46
13	0.7391	0.7556	0.7473	45
14	0.5652	0.3023	0.3939	43
15	0.6538	0.7727	0.7083	44
16	0.7556	0.8095	0.7816	42
17	0.7907	0.8095	0.8000	42
18	0.7188	0.5610	0.6301	41
19	0.6471	0.5500	0.5946	40
accuracy			0.6438	1269
macro avg	0.6392	0.6181	0.6107	1269
weighted avg	0.6377	0.6438	0.6247	1269

Figure III.5-rapport de classification de modèle RCNN.

- **Les graphes (Accuracy et loss)**

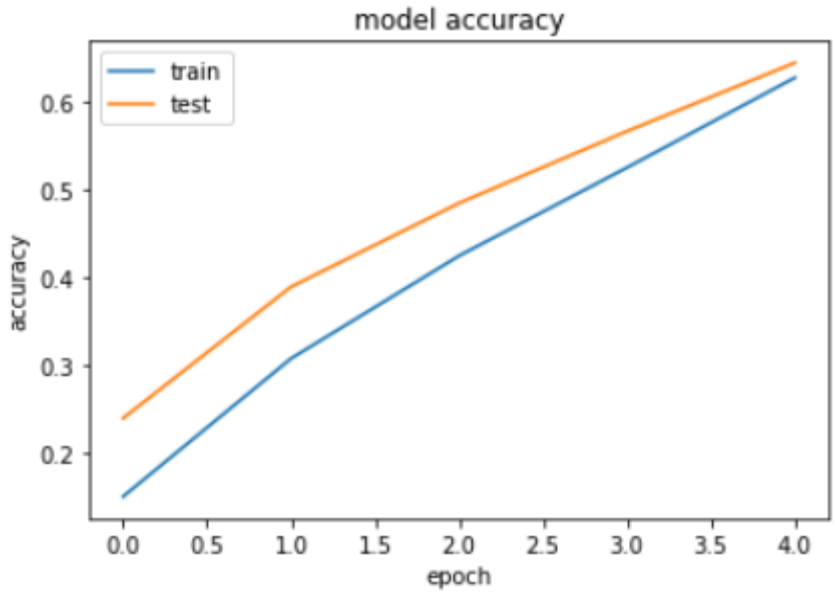

Figure III.6-*Graphe accuracy de modèle RCNN.*

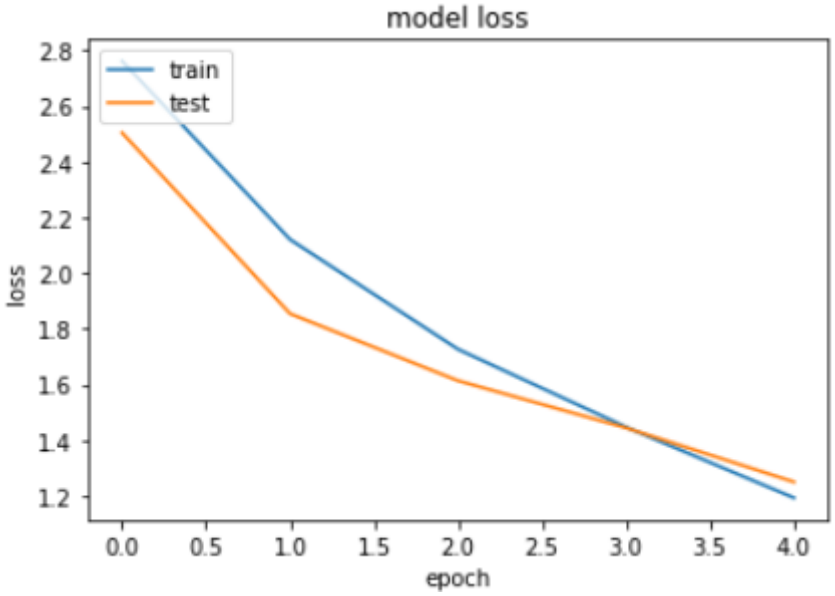

Figure III.7-*Graphe loss de modèle RCNN.*

III.3.3.2 RCNN Merged

Nous avons mis en œuvre RCNN_Merged pour obtenir un modèle de classification, ce modèle nous avons combine les descriptions des services web avec les actions extraire on utilise l'algorithme (III.3.1.1). puis nous avons utilisé les paramètres suivant pour obtenir le résultat :

- Max_length =433
- Batch_size=32
- Learning_rate= 0.0009
- Nombre epoch=6
- Glove= glove.6B.100d.txt

- **Le code de l'algorithme RCNN_Merged**

```
history = model_RCNN.fit([X_train_Glove1, X_train_Glove2], y_train,
                         validation_data=([X_test_Glove1, X_test_Glove2], y_test),
                         epochs=6,
                         batch_size=32,
                         verbose=2
                          )
```

- **Résultats d'exécution**
  - **NMI et Purity**

```
cluster_nmi(predicted, y_true)
```
Last executed at 2022-06-11 13:29:46 in 26ms

```
0.5795975235914327
```

```
purity(y_true, predicted)
```
Last executed at 2022-06-11 13:29:46 in 23ms

```
0.6595744680851063
```

Figure III. 8-*NMI et Purity de modèle RCNN Merged.*

  - **Rapport de classification**

	precision	recall	f1-score	support
0	0.5283	0.6316	0.5753	133
1	0.8198	0.7982	0.8089	114
2	0.8316	0.8876	0.8587	89
3	0.8846	0.5542	0.6815	83
4	0.6903	0.9398	0.7959	83
5	0.6400	0.6316	0.6358	76
6	0.3587	0.4342	0.3929	76
7	0.8293	0.5312	0.6476	64
8	0.6418	0.7679	0.6992	56
9	0.6833	0.7593	0.7193	54
10	0.6000	0.6471	0.6226	51
11	0.4688	0.3191	0.3797	47
12	0.1667	0.0870	0.1143	46
13	0.8095	0.7556	0.7816	45
14	0.5172	0.3488	0.4167	43
15	0.7400	0.8409	0.7872	44
16	0.6727	0.8810	0.7629	42
17	0.6591	0.6905	0.6744	42
18	0.8667	0.6341	0.7324	41
19	0.6744	0.7250	0.6988	40
accuracy			0.6588	1269
macro avg	0.6541	0.6432	0.6393	1269
weighted avg	0.6621	0.6588	0.6512	1269

Figure III.9-*Rapport de classification de modèle RCNN Merged.*

- **Les graphes (Accuracy et loss)**

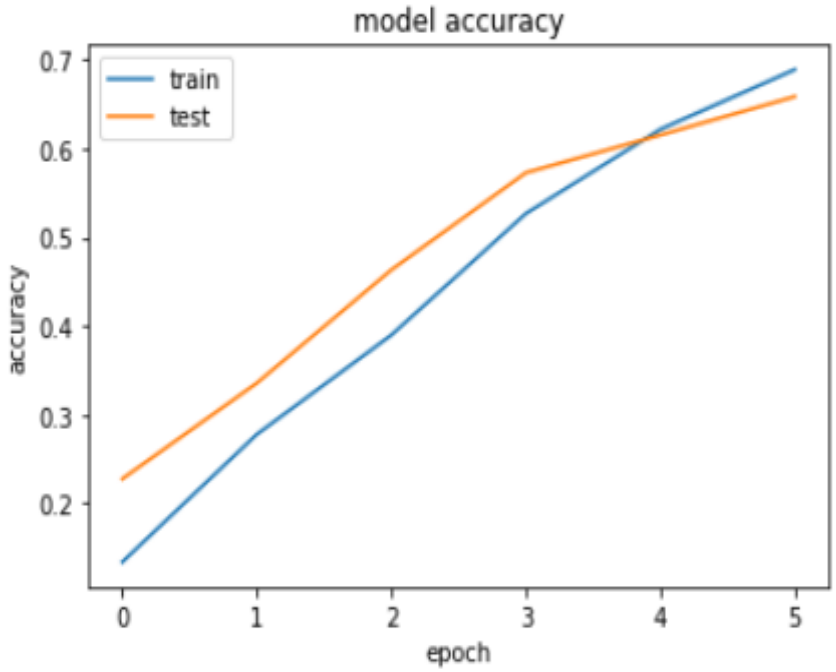

Figure III.10-*Graphe accuracy de modèle RCNN Merged.*

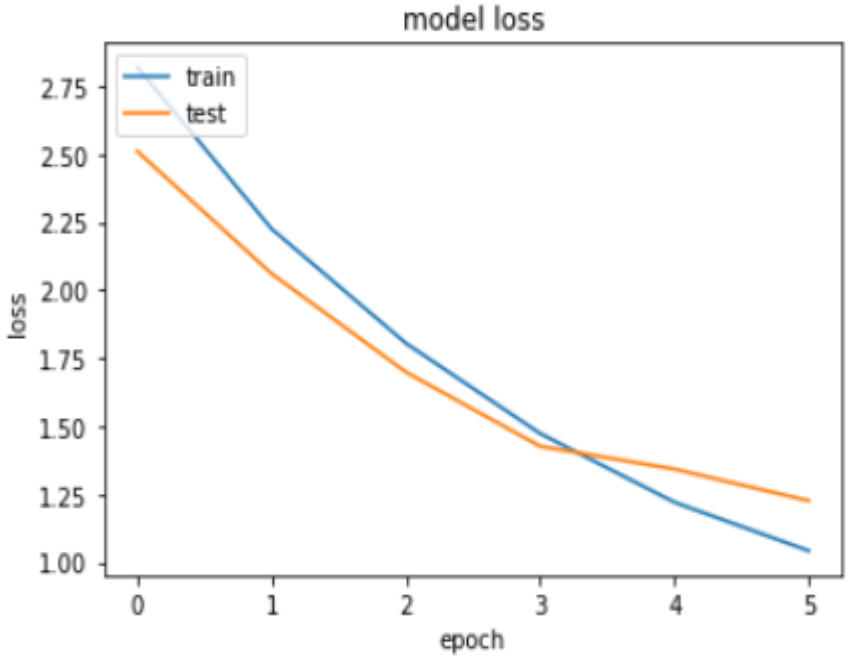

Figure III.11-*Graphe accuracy de modèle RCNN Merged.*

III.3.3.3 BERT

Nous avons utilisé l'algorithme de BERT pour obtenir un modèle de classification, ce modèle utilise les descriptions simples des services web et pour faire les prédictions, nous avons utilisé la colonne (Api_prim_cate), puis nous utilisons les paramètres suivants pour obtenir le résultat :

- Max_length =433
- Batch_size=64
- Learning_rate= 1e-5
- Nombre epoch=5
- MODEL_BERT = ("bert-base-uncased")

- **Le code de l'algorithme BERT**

```
history = model.fit(
    x ={'input_ids':x_train['input_ids'],'attention_mask':x_train['attention_mask']} ,
    y = y_train,
    validation_data = (
    {'input_ids':x_test['input_ids'],'attention_mask':x_test['attention_mask']}, y_test
    ),
  epochs=5,
  batch_size=64
)
```

- **Résultats d'exécution**
  - **NMI et Purity**

```
cluster_nmi(y_predicted, y_true)
```

```
Last executed at 2022-06-01 07:43:06 in 7ms
```

```
0.6924914597666207
```

```
purity(y_true, y_predicted)
```

```
Last executed at 2022-06-01 07:43:14 in 13m
```

```
0.7746256895193065
```

Figure III.12-*NMI et Purity de modèle BERT.*

- **Rapport de classification**

	precision	recall	f1-score	support
0	0.6620	0.7068	0.6836	133
1	0.8547	0.8772	0.8658	114
2	0.8218	0.9326	0.8737	89
3	0.8919	0.7952	0.8408	83
4	0.8280	0.9277	0.8750	83
5	0.8182	0.7105	0.7606	76
6	0.6543	0.6974	0.6752	76
7	0.7656	0.7656	0.7656	64
8	0.7097	0.7857	0.7458	56
9	0.9020	0.8519	0.8762	54
10	0.8611	0.6078	0.7126	51
11	0.6964	0.8298	0.7573	47
12	0.4333	0.2826	0.3421	46
13	0.9091	0.8889	0.8989	45
14	0.6571	0.5349	0.5897	43
15	0.7955	0.7955	0.7955	44
16	0.8636	0.9048	0.8837	42
17	0.9211	0.8333	0.8750	42
18	0.6889	0.7561	0.7209	41
19	0.6957	0.8000	0.7442	40
accuracy			0.7746	1269
macro avg	0.7715	0.7642	0.7641	1269
weighted avg	0.7742	0.7746	0.7712	1269

Figure III.13-*rapport de classification de modèle RCNN Merged.*

- **Les graphes (Accuracy et loss)**

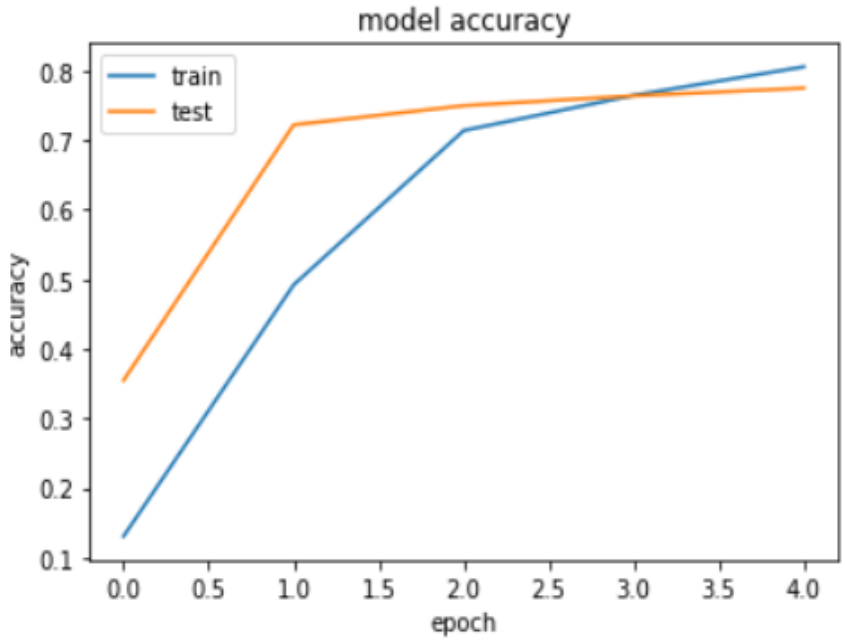

Figure III.14-*Graphe accuracy de modèle BERT.*

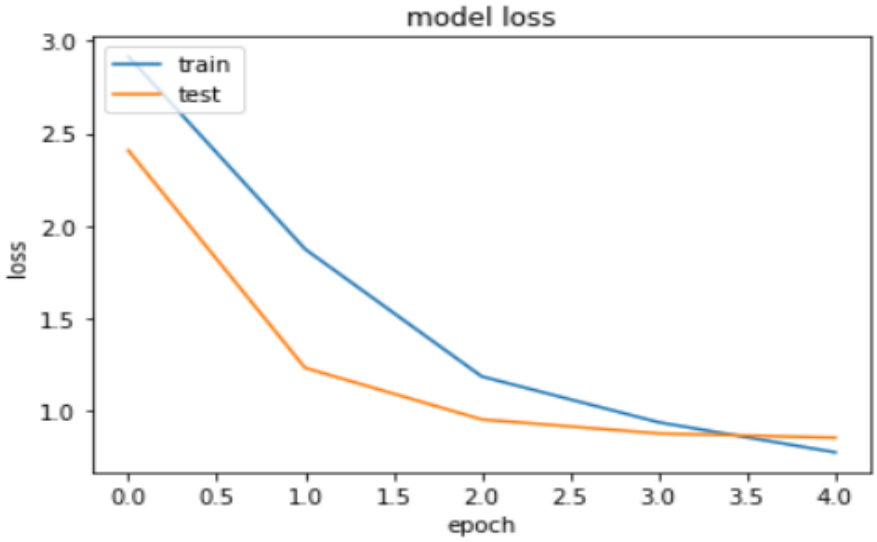

***Figure III.15**-Graphe loss de modèle BERT.*

III.3.3.4 BERT_Merged

Nous avons mis en œuvre BERT_Merged pour obtenir un modèle de classification. Par rapport au modèle BERT nous avons concaténer les descriptions des services web avec les actions extraire on utilise l'algorithme (III.3.1.1). Ensuite, on a mis tout ça comme un entre de notre modèle, puis nous utilisons les paramètres suivants pour obtenir le résultat :

- Max_length =512
- Batch_size=32
- Learning_rate= 30e-6
- Nombre epoch=5
- MODEL_BERT = ("bert-base-uncased")

- **Le code de l'algorithme BERT**

```
train_history = model.fit(
    x ={'input_ids':x_train['input_ids'],'attention_mask':x_train['attention_mask']} ,
    y = y_train,
    validation_data = (
    {'input_ids':x_test['input_ids'],'attention_mask':x_test['attention_mask']}, y_test
    ),
  epochs=5,
  batch_size=32
)
```

- **Résultats d'exécution**
  - **NMI et Purity**

```
cluster_nmi(y_predicted, y_true)
```
Last executed at 2022-06-20 08:12:17 in 7ms

```
0.693162618765053
```

```
purity(y_true, y_predicted)
```
Last executed at 2022-06-20 08:12:17 in 11ms

```
0.7785657998423956
```

Figure III.16-*NMI et Purity de modèle BERT_Merged.*

  - **Rapport de classification**

	precision	recall	f1-score	support
0	0.7458	0.6617	0.7012	133
1	0.8319	0.8684	0.8498	114
2	0.8526	0.9101	0.8804	89
3	0.7551	0.8916	0.8177	83
4	0.8444	0.9157	0.8786	83
5	0.8286	0.7632	0.7945	76
6	0.7231	0.6184	0.6667	76
7	0.7391	0.7969	0.7669	64
8	0.7679	0.7679	0.7679	56
9	0.8980	0.8148	0.8544	54
10	0.7872	0.7255	0.7551	51
11	0.7692	0.8511	0.8081	47
12	0.4130	0.4130	0.4130	46
13	0.9111	0.9111	0.9111	45
14	0.5263	0.4651	0.4938	43
15	0.8140	0.7955	0.8046	44
16	0.8837	0.9048	0.8941	42
17	0.7755	0.9048	0.8352	42
18	0.8387	0.6341	0.7222	41
19	0.7174	0.8250	0.7674	40
accuracy			0.7786	1269
macro avg	0.7711	0.7719	0.7691	1269
weighted avg	0.7777	0.7786	0.7759	1269

Figure III.17-*rapport de classification de modèle BERT_Merged.*

- **Les graphes (Accuracy et loss)**

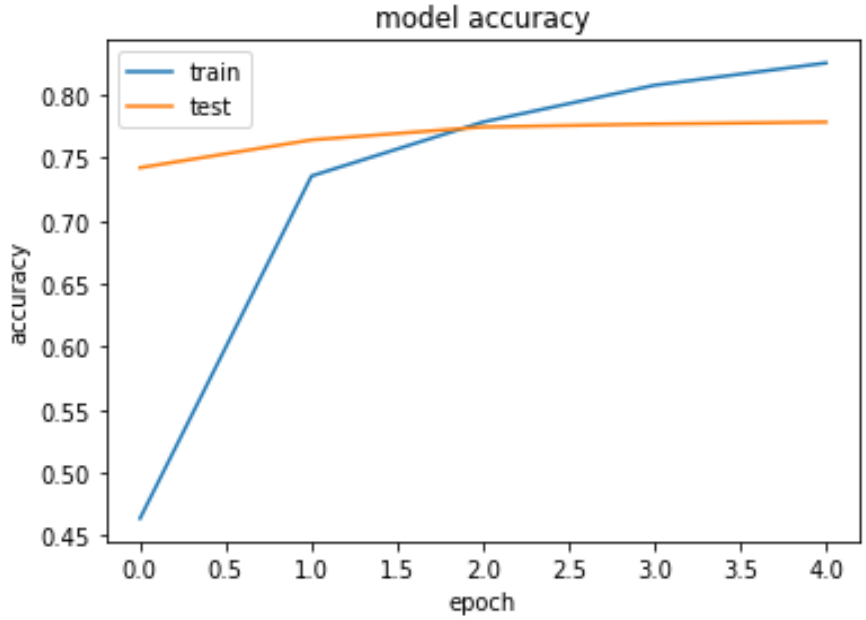

Figure III.18- *Graphe accuracy de modèle BERT_Merged.*

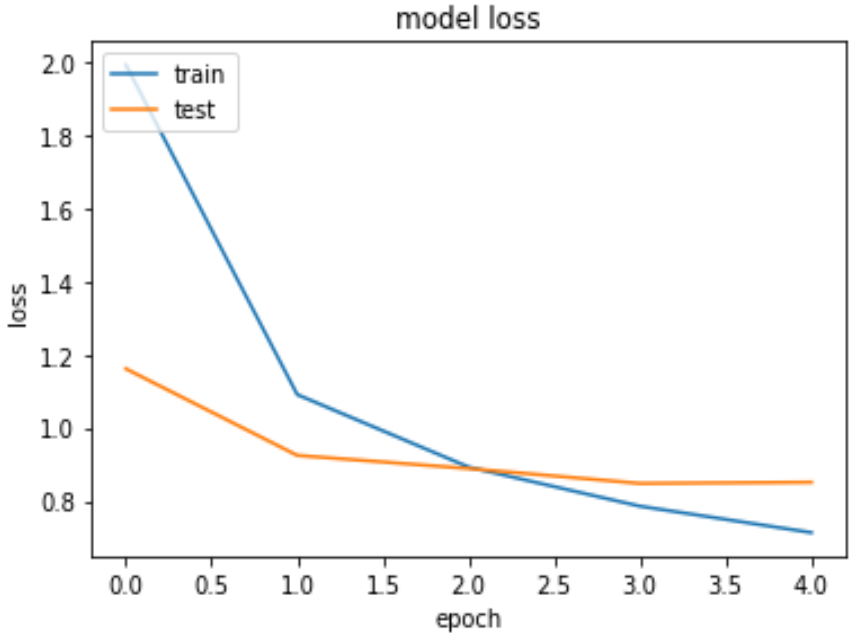

Figure III.19- *Graphe loss de modèle BERT_Merged.*

III.3.3.5 Comparaison entre les algorithmes

Nous avons évalué les résultats de la classification selon quatre mesures largement utilisées (Purity, NMI, Recall, F1-mesaure) le tableau montre les performances de toutes les méthodes concurrentes, on va comparer cette résultats avec les résultats de papier [38] et papier [39]:

Méthodes	Purity	NMI	Recall	F1-mesaure
DeepWSC (RCNN, WE-LDA)	0.5708	0.4856	0.3821	0.3969
DeepWSC (RCNN, WE-LDA, Heuristics)	0.6379	0.5273	0.4186	0.4356
RCNN	0.6438	0.5704	0.6438	0.6247
RCNN_Merged	0.6595	0.5795	0.6588	0.6512
BERT	0.7746	0.6924	0.7746	0.7712
BERT_Merged	**0,7785**	**0,6931**	**0,7786**	**0,7759**

Tableau III.2-*Comparaisons des performances de la classification des services Web.*

Avec notre modèle RCNN, nous avons constaté une amélioration d'une moyenne de 26,58% sur toutes les mesures par rapport aux résultats de classification du modèle DeepWSC (RCNN, WE-LDA, Heuristics) de l'article [39]. Ensuite, une fois que nous avons intégré les actions, notre modèle RCNN_Merged nous a permis d'avoir une amélioration par rapport aux résultats de classification du modèle RCNN 2,65%, le troisième modèle une fois qu'on a intégré le BERT avec simple description nous avons constaté 18,23% par rapport aux résultats de classification du modèle RCNN_Merged, dans le dernier modèle qu'on a intégré les actions le modèle Bert_merged nous avons permis d'avoir une amélioration par rapport aux résultats de classification du modèle BERT 0.41%.

III.4 Conclusion

Nous avons présenté tout au long de ce chapitre la progression générale de la démarche proposée et ses différentes étapes. Dans un premier temps, nous avons cité quelques travaux connexes qui ont été réalisés dans le domaine. Ensuite, nous avons mis en œuvre et illustré quatre (04) modèles de Deep Learning (RCNN, RCNN_Merged, BERT et BERT_Merged) basés sur la classification de texte.

Par la suite, nous avons fait une étude comparative entre nos modèles et ceux des auteurs de [38] et [39] qui sont considérés comme une référence dans le domaine. Les résultats obtenus ont démontré l'efficacité des modèles proposés et l'importance de l'extraction des actions pour les services web.

Conclusion générale

Dans le cadre de ce projet, nous avons essayé de proposer une nouvelle approche pour la classification des services web basée sur la description textuelles des services afin d'améliorer la découverte de ces derniers.

Nous avons présenté la notion des services web, leurs technologies de base ainsi que les grands défis liés à leur cycle de consommation, dont nous avons montré l'importance du processus de la découverte dans la fourniture de services de qualité.

Vue son importance, la découverte a suscité l'intérêt de la communauté de recherche pour la rendre plus pertinente et efficace. Les travaux dans le domaine se résument en trois (03) classes, syntaxique, sémantique et sociale. Mais avec l'avènement des nouvelles technologies issues du domaine du Deep Learning et leur succès, une nouvelle tendance est apparue, c'est l'application des techniques de DL sur les données et les descriptions des services web.

Dans cette perspective, et après avoir effectué un tour d'horizon des différentes méthodes d'apprentissage automatique pour la classification de texte, s'inscrit notre proposition.

Pour commencer, nous avons présenté le déroulement général de la méthode proposée et ses différentes étapes. Comme, nous avons cité quelques travaux connexes dans le domaine, puis nous avons illustré et implémenté certaines méthodes de Deep Learning (RCNN, RCNN_Merged, BERT, et BERT_Merged) basés sur la classification de texte.

Ces modèles ont été entrainés et testés sur un Dataset issu du monde réel, dont les données ont été fournies d'un annuaire bien connu « programmableweb.com ». Les descriptions des services web sont encore utilisées pour extraire, ce que nous qualifions de « ***Actions*** » ces dernières sont employées pour accentuer l'entrainement.

Après avoir comparé nos modèles proposés, avec ceux des travaux récents du domaine et utilisant le même Dataset, les résultats expérimentaux obtenus ont démontré l'efficacité de notre approche, qui a surpassé ceux des travaux les plus récents dans l'état de l'art.

L'approche proposée dans ce mémoire n'est que la première étape dans une démarche qui cherche à ajouter une dimension sociale sur des fondations solides. Cette dimension sert à doter

les Services Web d'une mémoire et ce pour exploiter leurs données d'interactions afin d'améliorer d'avantage leurs processus de découverte.

Bibliographie

[1] JM. Chauvet. Services Web avec SOAP, WSDL, UDDI, ebXML, Eyrolles, Paris, 2002.

[3] Fabrice Rossi « Services Web », Cours, Université Paris-IX Dauphine.

[4] Sylvain Rampacek « Modélisation et vérification d'un ou plusieurs services web », Université de REIMS Laboratoire CReSTIC Équipe SysCom, 19/10/2006.

[9] Bouchiha djelloul « Ré-ingénierie des Applications Web vers les Services Web sémantiques » thèse doctorat, Sidi belabess, Laboratory (EEDIS) 12-01-2011.

[10] Houda EL BOUHISSI « Découverte des Services Web : Approche basée sur les préférences des utilisateurs», thèse doctorat, Laboratory (EEDIS).

[12] Hemam née HIOUALO « Composition sémantique des services web dans un contexte d'ebXML », thèse doctorat en Informatique, Université Mentouri de Constantine, 2011.

[13] MERZOUG Mohammed «La découverte et la sélection des services web», thèse doctorat en Informatique, Université Abou Bekr Belkaid de Alger ,2 /06/2018.

[14] ABDERRAHIM Naziha « Contribution des réseaux sociaux dans l'ingénierie des services Web », thèse doctorat en Informatique, Université Djillali Liabes de Sidi Bel Abbes.

[15] Hakim Hacid Zakaria Maamar and Michael N.Huns, why web services need social networks, IEEE Computer Society, 90-94, PDF: 05731594.

[16] Johann Stan, Fabrice Muhlenbach, Christine Largeron, Recommender Systems using Social Network, Vol 23, page 5,6.

[19] Ibrahim EL BITAR, « CBR4WSD : Une approche de découverte de Services Web par Raisonnement à Partir de Cas », thèse de Doctorat, Université Mohamed v ecole Mohammadia d'ingenieurs rabat, 13/11/2014.

[29] Mokhtar TAFFAR, « INITIATION A L'APPRENTISSAGE AUTOMATIQUE », Support de Cours.

[30] BOUDHEB Tarik, Privacy Preserving Classification of Biomedical Data, these doctorat, UNIVERSITE DJILLALI LIABES FACULTE DES SCIENCES EXACTES SIDI BEL ABBE, 17/07/2019.

[34] Henri Lasselin, Make text look like speech: disfluency generation using sequence-to-sequence neural networks, « Document and Text Processing - Machine Learning », 20/03/2018.

[38] Guobing Zou, Zhen Qin, Qiang He, Pengwei Wang, Bofeng Zhang, Yanglan Gan « DeepWSC: A Novel Framework with Deep Neural Network for Web Service Clustering », 2019 IEEE International Conference on Web Services (ICWS).

[39] Guobing Zou, Zhen Qin, Qiang He, Pengwei Wang, Bofeng Zhang, Yanglan Gan « DeepWSC: Clustering Web Services via Integrating Service Composability into Deep Semantic Features », publication in a journal 2020 IEEE Transactions on Services Computing.

[40] Zhao Huang, Wei Zhao « Combination of ELMo Representation and CNN Approaches to Enhance Service Discovery », IEEE, 2020.

[41] Hongfan Ye, Buqing Cao, Zhenlian Peng, Ting Chen, Yiping Wen, Jianxun Liu « Web Services Classification Based on Wide & Bi-LSTM Model », IEEE,2019.

[42] Mohamed S. Alshafaey, Ahmed I.Saleh ,Mohamed F. Alrahamawy « A new cloud-based classification methodology (CBCM) for efficient semantic web service discovery», 2021.

[43] Xin Wang; Jin Liu; Xiao Liu; Xiaohui Cui; Hao Wu « A Novel Dual-Graph Convolutional Network based Web Service Classification Framework », 2020 IEEE International Conference on Web Services (ICWS).

[44] Xin Liu, Hui-Min Cheng, Zhong-Yuan Zhang « Evaluation of Community Detection Methods », 2019.

Webographie

[2] OpenClassrooms Les services Web, Chabane Refes, « https://www.pfl-cepia.inra.fr/uploads/images/GestionDonneesImages/unites/GdpDoc/OpenClassrooms-servicesWeb.pdf » (La dernière consultation 20/06/2022).

[5] WEB SERVICE SOAP SMS, «https://www.allmysms.com/api-sms/web service-soap/ » (La dernière consultation 01/06/2022).

[6] The structure of a SOAP message,« https://www.ibm.com/docs/en/integration-bus/10.0?topic=soap-structure-message » (La dernière consultation 01/06/2022).

[7] Mastering REST Architecture REST Architecture Details, Ahmet Özlü, «https://ahmetozlu.medium.com/mastering-rest-architecture-rest-architecture-details e47ec659f6bc » (La dernière consultation 01/06/2020).

[8] Comprendre l'architecture des web services REST, « https://www.slideserve.com/tassos/comprendre-l-architecture-des-web-services-rest » (La dernière consultation 03/06/2022).

[11] Web Service Semantics – WSDL-S, R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M.T. Schmidt, A. Sheth,K. Verma, « https://www.w3.org/Submission/WSDL-S/ », (La dernière consultation 03/06/2022).

[17] L'Architecture Orienté Service (SOA), « https://www.rapport-gratuit.com/larchitecture-oriente-service-soa// » (La dernière consultation 03/06/2022).

[18] Communication avec SOAP, « https://www.rapport-gratuit.com/communication-avec-soap/ » (La dernière consultation 03/06/2022).

[20] Introduction to Word Embeddings, Chanika Ruchini,

« https://medium.com/analytics-vidhya/introduction-to-word-embeddings-c2ba135dce2f » (La dernière consultation 03/06/2022).

[21] Word Embedding Techniques: Word2Vec and TF-IDF Explained, Adem Akdogan « https://towardsdatascience.com/word-embedding-techniques-word2vec-and-tf-idf-explained-c5d02e34d08 » (La dernière consultation 03/06/2022).

[22] Fundamentals of Bag Of Words and TF-IDF, Prasoon Singh, «https://medium.com/analytics-vidhya/fundamentals-of-bag-of-words-and-tf-idf 9846d301ff22 » (La dernière consultation 03/06/2022).

[23] Understanding Word2Vec, Odunayo Ogundepo, « https://medium.com/analytics-vidhya/understanding-word2vec-39fabe660705 » (La dernière consultation 03/06/2022).

[24] Classification using Long Short Term Memory & GloVe (Global Vectors for Word Representation), Sourav Bhattacharyya, « https://medium.com/analytics-vidhya/classification-using-long-short-term-memory-glove-global-vectors-for-word-representation-254d02d5e158 » (La dernière consultation 03/06/2022).

[25] Word2Vec, GLOVE, FastText and Baseline Word Embeddings step by step, 04/06/2022, Akash Deep ,« https://medium.com/analytics-vidhya/word2vec-glove-fasttext-and-baseline-word-embeddings-step-by-step-d0489c15d10b » (La dernière consultation 04/06/2022).

[26] Text classifiers in machine Learning a practical guide, « https://levity.ai/blog/text-classifiers-in-machine-learning-a-practical-guide » (La dernière consultation 04/06/2022).

[27] Une introduction a l'apprentissage automatique apprendre langue, «http://www.euro-langues.org/une-introduction-a-lapprentissage-automatique-apprendre-langue/ » (La dernière consultation 04/06/2022).

[28] Quest ce que l'apprentissage-supervise, « https://fr.linedata.com/quest-ce-que-lapprentissage-supervise » (La dernière consultation 04/06/2022).

[31] Apprentissage non supervisé, « https://datascientest.com/apprentissage-non-supervise»(La dernière consultation 04/06/2022).

[32] Deep Learning : définition et principes de l'apprentissage profond, « https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501333-deep-learning-definition-et-principes-de-l-apprentissage-profond/ » (La dernière consultation 04/06/2022).

[33] Deep Learning Techniques for Text Classification, «https://medium.datadriveninvestor.com/deep-learning-techniques-for-text-classification-9392ca9492c7 » (La dernière consultation 04/06/2022).

[35] LSTM transformers gpt bert guide des principales techniques en NLP, Houssam Alrachid, « https://blog.ysance.com/lstm-transformers-gpt-bert-guide-des-principales-techniques-en-nlp » (La dernière consultation 04/06/2022).

[36] BERT variants and theirdifferences, « https://360digitmg.com/bert-variants-and-their-differences » (La dernière consultation 04/06/2022).

[37] Exploring BERT variants (Part 1): ALBERT, RoBERTa ELECTRA, Chandan Durgia , «https://towardsdatascience.com/exploring-bert-variants-albert-roberta-electra-642dfe51bc23 » (La dernière consultation 04/06/2022).

[45] Sklearn.metrics.normalized mutual info score, «https://scikitlearn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html » (La dernière consultation 17/06/2022).

[46] Recall precision f1 score, «https://www.insidemachinelearning.com/recall-precision-f1-score/» (La dernière consultation 17/06/2022).

[47] Saturn Cloud Quickstart, « https://saturncloud.io/docs/quickstart/ » (La dernière Consultation 17/06/2022).