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

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et Télécommunications

Par :

Bani Sabrina
Baroud Nourelhouda

Sur le thème

Une approche basée BERT-NSP pour la prédiction et l'initialisation du réseau social de coopération des Services Web Sémantiques

Soutenu publiquement le 10 / 07 / 2023 à Tiaret devant le jury composé de :

Mr AID Lahcene	MCB	Université IBN-KHALDOUN Tiaret	Président
Mr MEGHAZI Hadj Madani	MAA	Université IBN-KHALDOUN Tiaret	Encadrant
Mr MOSTEFAOUI Kadda	MAA	Université IBN-KHALDOUN Tiaret	Examinateur

2022-2023

Remerciement

Le grand remerciement à notre dieu qui nous a aidés en nous donnant la force et le courage durant ce travaille.

Nous remercions nos parents, nos sœur et nos frères qui nous soutenu durant notre études.

On tient remercier tout particulièrement notre encadreur

Mr.Meghazi Hadj Madani pour son aide, ses encouragements, ses orientations et ses précieux conseils lors de la réalisation de ce travail.

Nous sommes particulièrement heureuse que Mr. Aid Lahcene et Mr.Mostefaoui Kadda, de faire partie du jury de ce travail.

Enfin, on tient à exprimer vivement nos remerciements avec une profonde gratitude à toutes les personnes qui ont contribué de près ou de lois à la réalisation de ce travail.

Dédicace

Je dédie ce modeste travail à :

À mes chers parents, que nulle dédicace ne puisse exprimer mes sincères sentiments, Pour leur patience illimitée, leur encouragement continu, leur aide, en témoignage de mon profond amour et respect pour leurs grands sacrifices.

À mes frères Cheikh ,Mohammed ,Nourelddine et Ibrahim pour leur appui et leur encouragement.

À mes chère sœurs Oumhani et Saliha pour leurs encouragements permanents, et leur soutien moral.

À mon cher binôme Nour el houda.

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

À toutes personnes qui m'ont encouragé ou aidé au long de mes études.

Sabrina.

Dédicace

Je dédie ce modeste travail à :

À mes chers parents, que nulle dédicace ne puisse exprimer mes sincères sentiments, Pour leur patience illimitée, leur encouragement continu, leur aide, en témoignage de mon profond amour et respect pour leurs grands sacrifices.

À mes frères Ahmed et Mohammed pour leur appui et leur encouragement

À mes chère sœurs Kheira ,Khadidja, Malika et Djamila pour leurs encouragements permanents, et leur soutien moral.

À mon cher binôme Sabrina.

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

À toutes personnes qui m'ont encouragé ou aidé au long de mes études.

Nourelhouda .

Résumé

Les services de Web sémantiques visent à automatiser le processus complet des services web : la découverte, la sélection et la composition. Certains suggèrent d'utiliser les liens sociaux entre les services, considérés comme une mémoire du système, pour atteindre cet objectif. Cependant, le développement d'un tel réseau nécessite l'utilisation d'ontologies de domaine, ce qui a entravé l'évolution de ce domaine de recherche. Heureusement, les nouvelles techniques de traitement du langage naturel (NLP) permettent d'obtenir une certaine mesure de sémantique à partir des descriptions textuelles des services web sans recourir à ces ontologies. En exploitant cette opportunité, cette approche propose d'utiliser le modèle NLP BERT pour créer un réseau social de coopération entre les services web, qui contribuera à la mémorisation et à la découverte des séquences de services.

Mots clés : Découverte des Services web, composition des Services Web, L'apprentissage en profondeur, BERT-NSP.

Abstract

Semantic Web services aim to automate the entire process of web services: discovery, selection and composition. Some suggest using the social links between services, considered as a memory of the system, to achieve this goal. However, the development of such a network requires the use of domain ontologies, which has hindered the evolution of this research field. Fortunately, new natural language processing (NLP) techniques make it possible to obtain a certain measure of semantics from the textual descriptions of web services without resorting to these ontologies. By exploiting this opportunity, this approach proposes to use the NLP BERT model to create a social network of cooperation between web services, which will contribute to the memorization and discovery of service sequences.

Keywords : Web Services Discovery, Web Services Composition, Deep Learning, BERT-NSP.

ملـخص

تهدف خدمات الويب الدلالية إلى تمت عملية خدمات الويب بالكامل: الاكتشاف والاختيار والتكوين. يقترح البعض استخدام الروابط الاجتماعية بين الخدمات ، باعتبارها ذاكرة للنظام لتحقيق هذا الهدف. ومع ذلك، فإن تطوير مثل هذه الشبكة يتطلب استخدام علم الوجود للمجال، مما أعاق تطور هذا المجال البحثي. لحسن الحظ ، تتيح تقنيات معالجة اللغة الطبيعية الجديد الحصول على قدر معين من الدلالات من الأوصاف النصية لخدمات الويب دون اللجوء إلى هذه انطولوجيا. من خلال استغلال هذه الفرصة ، يقترح هذا النهج استخدام نموذج NLP BERT لإنشاء شبكة اجتماعية للتعاون بين خدمات الويب ، والتي ستساهم في حفظ واكتشاف تسلسل الخدمة.

الكلمات المفتاحية : اكتشاف خدمات الويب ، تكوين خدمات الويب ، التعلم العميق ،نموذج BERT-NSP

Table des matières

Introduction Générale 1

Chapitre Ⅰ : les services web 3

I.1 Introduction 4

I.2 Les services web 4

I.2.1 Les caractéristiques des services web 5

I.2.2 L'Architecture générale d'un service Web 5

I.2.3 Les types de services web 6

I.2.3.1 SOAP (Simple Object Access Protocol) 6

I.2.3.2 REST (Representational state Transfer) 6

I.3 Le cycle de vie d'un service web 7

I.3.1 La création : 7

I.3.2 La découverte : 7

I.3.3 La sélection 8

I.3.4 La composition 8

I.3.4.1 Orchestration 9

I.3.4.2 Chorégraphie 9

I.3.5 Exécution 9

I.4 Les services web sémantiques 9

I.4.1 L'infrastructure des services Web sémantiques 9

I.4.2 Approches proposées pour la réalisation des SWS 10

I.4.2.1 WSDL-S (Web Service Description Language-Semantic) 10

I.4.2.2 OWL-S (Ontology Web Language for Services): 11

I.4.2.3 Le WSMO (Web Service Modeling Ontology): 12

I.5 Les services web sociaux 12

I.6 La découverte des services web 13

I.6.1 Découverte syntaxique 13

I.6.2 Découverte sémantique 13

I.6.2.1 Limite de découverte des services web sémantique 14

I.6.3 La découverte sociale 14

I.6.3.1 Les approches de la découverte des services web sociaux 14

I.7 Conclusion 16

Chapitre Ⅱ : Les techniques DL NLP pour avoir la sémantique 18

II.1 Introduction 18

II.2 Le traitement du langage naturel (NLP) 18
II.3 Deep Learning 20
II.4 Word Embedding 20
II.4.1 Les Modèles de WE 20
II.4.1.1 Word Embeddings Classique 21
II.4.1.2 Word Embeddings Contextuelle 23
II.5 Les techniques Deep Learning pour le NLP 24
II.5.1 Réseaux de Neurones Récurrents (RNN) 24
II.5.1.1 La limitation des RNN 24
II.5.2 Réseaux Long Short-Term Memory (LSTM) 25
II.5.2.1 Fonctionnement du réseau LSTM 26
II.5.2.2 La limitation des LSTM 26
II.5.3 Le mécanisme d'Attention 27
II.5.4 Transformers 27
II.5.4.1 Les types d'attention 28
II.5.4.2 Architecture du Transformer 29
II.6 BERT 32
II.6.1 Introduction 32
II.6.2 Les types de BERT 33
II.6.3 L'architecture de BERT 34
II.6.3.1 Embedding 35
II.6.3.2 Token Embedding 35
II.6.3.3 Position Embedding : 36
II.6.3.4 Segment embedding : 36
II.6.4 Le mécanisme d'attention dans BERT 36
II.6.5 Autres variantes de BERT 38
II.6.6 Description des taches de MLM &NSP 39
II.6.7 Prédiction de la phrase suivante (NSP) 39
II.6.7.1 Le fonctionnement du BERT-NSP 40
II.7 Conclusion 41
Chapitre Ⅲ : Approche DL pour L'initialisation du RS de collaboration des SW 43
III.1 Introduction 43
III.2 Aperçu sur les réseaux sociaux 43
III.2.1 Représentation Matricielle d'un réseau social 44

III.3 Les services Web sociaux 44
III.3.1 Les services web dans les réseaux sociaux 45
III.3.2 Approche pour élaborer un réseau sociaux des services web 45
III.3.3 Les propriétés sociales de services Web 48
III.3.4 Systèmes de recommandation et plates-formes sociales 48
III.4 Les travaux connexes 49
III.4.1 Discovering web services in social web service repositories using deep variational auto-encoders : 50
III.4.2 Constructing a global social service network for better quality of web service discovery : 50
III.4.3 Mining Social Web Service Repositories for Social Relationships to Aid Service Discovery : 50
III.4.4 Social-Based Web Services Discovery and Composition for Step-by-Step Mashup Completion 51
III.4.5 Collaboration reputation for trustworthy Web service selection in social networks.51
III.5 Notre approche pour l'initialisation du RS de collaboration de SW 51
III.5.1 Extraction des données 53
III.5.2 Traitement des données.....53
III.5.3 Jeu de données d'entraînement Avec label pour la tâche NSP 53
III.5.4 Entraînement du modèle 53
III.5.5 Évaluation du modèle 53
III.5.6 Collecte des prédictions appliquées sur les descriptions des SW 54
III.5.7 Génération des matrices d'adjacences pour le réseau de collaboration.....54
III.5.8 Visualisation de matrice d'adjacence 54
III.6 Discussion des résultats 54
III.7 Tests et analyse du comportement de notre modéle.....60
III.8 Conclusion 62
Conlusion Générale 64
bobliographie 65
Webographie 67

Liste Des Figures

Figure I-1 Les différentes relations entre les spécifications des services web 4
Figure I-2 Structure des services web 5
Figure I-3 Format d'un message SOAP.....6
Figure I-4 Architecture des services web de type REST 7
Figure I-5 Infrastructure des services Web sémantiques 10
Figure I-6 Mapping entre le fichier WSDL et les concepts ontologiques 11
Figure I-7 Les éléments de base d'OWL-S 11
Figure I-8 Composants de WSMO 12
Figure II-1 Relation entre NLP, AI, ML et DL.....19
Figure II-2 Exemples de relations de mots dans l'espace Word2vec 21
Figure II-3 Exemples des architectures CBOW et Skip-gram de Word2vec 22
Figure II-4 Exemple d'architecture de GLOVE 23
Figure II-5 L'architecture des réseaux de neurone récurrent 25
Figure II-6 Unité de base LSTM 26
Figure II-7 Distribution de l'attention entre deux séquences. 27
Figure II-8 Architecture globale des Transformers 28
Figure II-9 Multi-Head Attention 31
Figure II-10 Module du BERT VS GPT, ELMO 33
Figure II-11 L'architecture d'incorporation de mots dans BERT 35
Figure II-12 La description des taches de NSP et MLM 39
Figure II-13 The Next sentence preédiction (NSP) task 41
Figure III-1 Exemple d'un réseau social 44
Figure III-2 Réseau social de compétition de services Web 46
Figure III-3 Réseau social de collaboration de services Web 47
Figure III-4 Réseau social de supervision 47
Figure III-5 Apport réciproque entre le système de recommandation et le réseau social.....49
Figure III-6 Notre approche pour l'initialisation du RS de collaboration de SW.....52
Figure III-7 Le Data set collecté à partir de programmable web. 55
Figure III-8 Le chargement les données d'entraînement et de validation 55
Figure III-9 Illustre l'entrainement de notre modèle. 56
Figure III-10 Le graphe d'accuracy. 58
Figure III-11 Le graphe de Loss. 58
Figure III-12 La nouveau Data set. 59
Figure III-13 La matrice d'adjacence. 59
Figure III-14 Visualisation du réseau de collaboration des SW. 60
Figure III-15 Exemple d'une description d'un service Web. 60
Figure III-16 L'identification des trois meilleurs collaborateurs d'un services web. 61

Liste des Tableaux

Tableau II-1 Nombre des paramètres for BERT-Base 34
Tableau II-2 Nombre des paramètres for BERT-Large 34
Tableau II-3 Autres variantes de BERT 38
Tableau III-1 Comparaisons des performances du Bert- NSP avant et après entrainement des services Web. 57

Introduction Générale

Les services Web sont des applications qui se distinguent par leur capacité à s'auto-décrire, à être modulaires et à présenter un faible couplage. Ils offrent un modèle simple pour la programmation et le déploiement d'applications, en se basant sur des normes et en s'exécutant sur l'infrastructure Web. Grâce à un ensemble de technologies courantes telles que SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery, and Integration) et XML (eXtensible Markup Language), ils permettent de fournir des fonctionnalités accessibles via le Web.

Les opérations essentielles pour tout système de manipulation de services sur Internet sont la découverte, la sélection et la composition des services. Ces opérations exploitent des techniques et des technologies informatiques et linguistiques récemment développées. Un système de découverte et de composition repose sur trois éléments clés : la requête du client, le système intermédiaire et le fournisseur de services. La découverte consiste à recueillir les services correspondant aux sous-requêtes éparpillées sur Internet, telles qu'exprimées par le client.

Avec l'augmentation rapide du nombre et de la diversité des services sur Internet, ainsi que des fonctionnalités et des technologies utilisées, il est devenu crucial de proposer un système permettant aux utilisateurs de découvrir efficacement ces services. Ce système doit leur permettre de sélectionner les meilleurs services répondant à leurs besoins, sans avoir à effectuer manuellement toutes les tâches nécessaires.

Notre travail propose de faire l'investigation pour une approche qui cherche à améliorer le processus de la découverte des services web sociaux en utilisant les techniques du Deep Learning. Nous essayons de se focalise sur un réseau social de collaboration et se pencher plus précisément sur le modèle BERT-NSP côté DL .

Ce mémoire est composé de trois chapitres qui couvrent les sujets suivants :

Le premier chapitre, intitulé "Les services web", aborde en détail les Services Web. Nous commencerons par définir ces services, en décrivant leurs caractéristiques, leurs types et leur cycle de vie. Ensuite, nous nous concentrerons sur les services web sémantiques, en examinant leur infrastructure ainsi que les approches proposées pour leur réalisation. Nous discuterons également du concept des Services Web Sociaux. Enfin, nous aborderons la problématique liée à la découverte des Services Web.

Le deuxième chapitre, intitulé "Les techniques d'apprentissage profond pour le traitement du langage naturel", présente différentes méthodes couramment utilisées pour l'analyse des données en traitement du langage naturel. Nous mettrons particulièrement l'accent sur les techniques d'apprentissage profond (Deep Learning) et leur application dans le domaine du traitement du langage naturel.

Comme nous examinerons les modèles de Word Embeddings (WEs) et nous nous concentrerons spécifiquement sur les techniques de Deep Learning, et visant le modèle BERT et BERT- NSP.

Le troisième chapitre, intitulé "Approche DL pour l'initialisation du RS de collaboration des SW ", il présente l'approche que nous avons adoptée pour la découverte des services web. Nous décrirons aussi les outils que nous avons utilisés pour mettre en œuvre notre propre solution, en expliquant toutes les étapes de l'implémentation et en présentant et discutant les résultats obtenus

Une conclusion générale viendra clore ce mémoire par une synthèse et une présentation des perspectives du présent travail.

Chapitre I

Les Services web

Sommaire

I.1 Introduction....4
I.2 Les services web....4
I.2.1 Les caractéristiques des services web....5
I.2.2 L'Architecture générale d'un service Web....5
I.2.3 Les types de services web....6
I.3 Le cycle de vie d'un service web....7
I.3.1 La création :....7
I.3.2 La découverte des services web :....7
I.3.3 La sélection....8
I.3.4 La composition....8
I.3.5 Exécution....9
I.4 Les services web sémantiques....9
I.4.1 L'infrastructure des services Web sémantiques....9
I.4.2 Approches proposées pour la réalisation des services Web sémantiques....10
I.5 Les services web sociaux....12
I.6 La découverte des services web....13
I.6.1 Découverte syntaxique....13
I.6.2 Découverte sémantique....13
I.6.3 La découverte sociale....14
I.7 Conclusion....16

I.1 Introduction

L'objectif initial des services web était de remédier aux problèmes d'interopérabilité et de communication entre divers systèmes informatiques. Auparavant, les applications étaient couramment développées sur des plates-formes et technologies particulières, rendant l'intégration de systèmes informatiques variés difficile. Les services web ont été conçus pour faciliter la communication et l'échange de données entre ces différents systèmes, en utilisant des protocoles standardisés tels que HTTP, XML et SOAP.

Les services web ont également simplifié la mise à disposition de services en ligne en proposant une interface standardisée permettant d'y accéder. Cela facilite la tâche des développeurs qui peuvent aisément créer des applications utilisant ces services. Les utilisateurs peuvent accéder à ces services sans avoir besoin d'installer un quelconque logiciel supplémentaire en utilisant un navigateur web ou une application dédiée.

I.2 Les services web

Selon la définition du W3C (World Wide Web Consortium) un service Web est un système logiciel conçu pour prendre en charge l'interaction interopérable de machine à machine sur un réseau. Il a une interface décrite dans un format exploitable par machine (spécifiquement WSDL). D'autres systèmes interagissent avec le service Web d'une manière prescrite par sa description à l'aide de messages (SOAP), généralement transmis à l'aide des protocoles Internet (HTTP) avec une sérialisation XML[1] conjointement avec d'autres normes liées au Web [1].

La figure I.1 illustre les éléments clés d'un service web, comprenant une interface XML nommée WSDL pour sa description, la faculté d'échanger des documents XML avec d'autres services via le protocole SOAP, et la possibilité d'être repéré au moyen d'un annuaire comme l'UDDI.

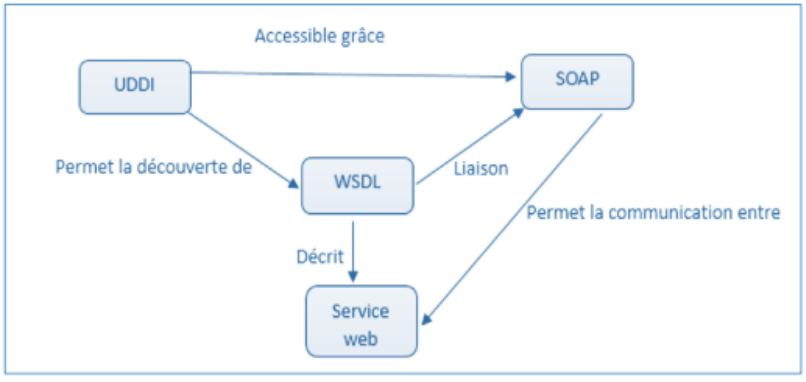

Figure I-1 Les différentes relations entre les spécifications des services web [2].

[1] XML : eXtended Markup Language

I.2.1 Les caractéristiques des services web

Le service web possède les attributs suivants :

- Ses interfaces permettent un accès automatisé aux services pour les applications.
- Il permet l'interaction entre applications.
- Il est accessible à travers le réseau.
- Il partage un contrat de service.
- Il utilise des protocoles internet standardisés et le codage XML [3].

I.2.2 L'Architecture générale d'un service Web

La figure I.2 illustre l'architecture de référence des services web, qui s'appuie sur trois concepts clés :

- Le fournisseur de service (service provider) : il correspond au détenteur du service et, du point de vue technique, est constitué par la plateforme hébergeant le service.
- Le client (service requestor) : il correspond à l'utilisateur du service et, du point de vue technique, est représenté par l'application qui recherche et invoque le service. L'application cliente peut elle-même être un service web.
- L'annuaire de service (Service Registry) : c'est une entité logicielle intermédiaire entre les clients et les fournisseurs de services. Il s'agit d'un registre de descriptions de services offrant des facilités de publication de services pour les fournisseurs, ainsi que des facilités de recherche de services pour les clients [4].

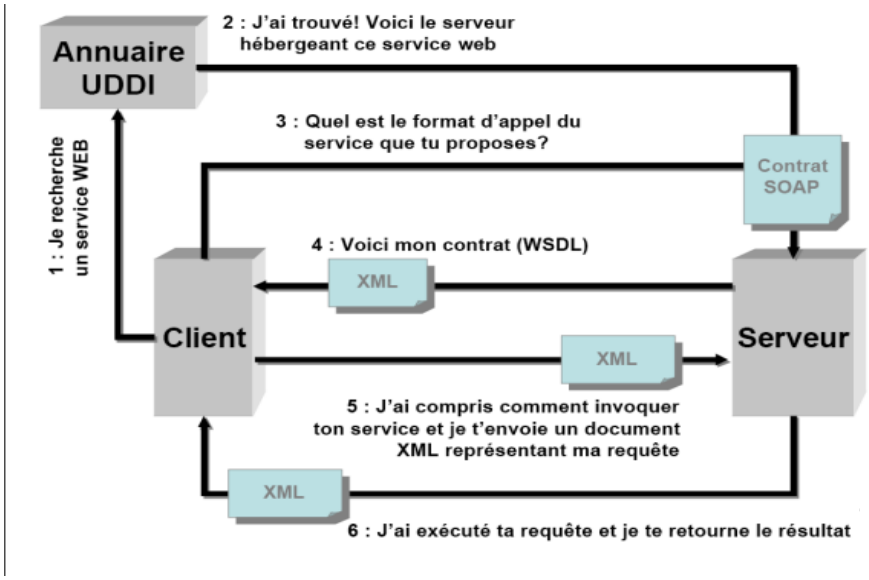

Figure I-2 Structure des services web [5].

I.2.3 Les types de services web

Il existe deux types de services web :

I.2.3.1 SOAP (Simple Object Access Protocol)

SOAP est un protocole basé sur XML qui permet d'appeler des méthodes sur des services distants afin de permettre la communication entre machines (voir figure I.3). Pour cela, il utilise un protocole de transport (HTTP) pour envoyer des messages aux machines distantes et appeler des méthodes RPC [6].

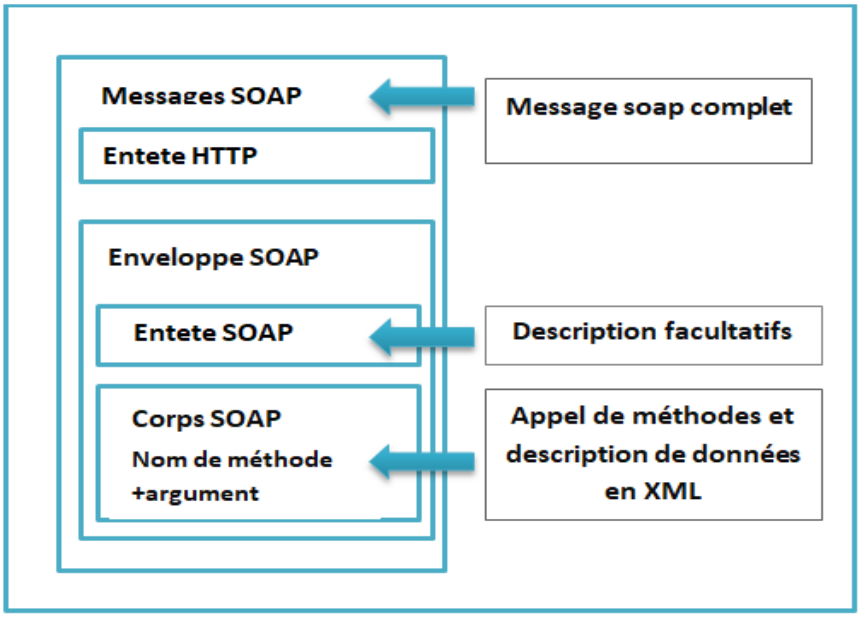

Figure I-3 Format d'un message SOAP.

I.2.3.2 REST (Representational state Transfer)

Le type de service web appelé REST, ou « Representational State Transfer », est un style d'architecture, utilisé pour concevoir des systèmes distribués tels que les applications web et les services web. Contrairement à SOAP ou XML-RPC, qui ont leurs propres protocoles, REST repose sur des conventions et de bonnes pratiques à suivre pour l'échange de données entre des machines.

La conception REST (voir Figure I.4) s'appuie sur les spécifications de base du protocole HTTP, sans ajouter de surcouche. Dans cette architecture, chaque ressource de l'application est accessible par une URL unique. Grâce à REST, la communication entre les clients et les serveurs est efficace et flexible, et l'indépendance des ressources assure une évolutivité aisée [6].

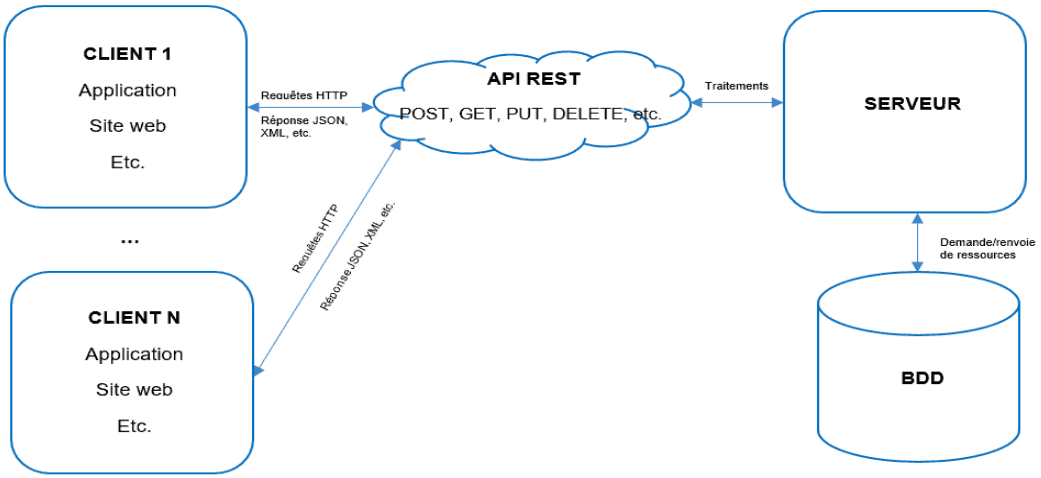

Figure I-4 Architecture des services web de type REST [7].

I.3 Le cycle de vie d'un service web

Voici une description générale des étapes courantes dans le cycle de vie d'un service web :

I.3.1 La création :

La conception et la mise en place d'un service web sont des opérations complexes impliquant plusieurs étapes clés, telles que la conception, le développement et le déploiement d'un site ou d'une application en ligne. Ce processus nécessite généralement la programmation de codes, la conception d'interfaces utilisateur, la gestion de bases de données, la sécurisation des données, ainsi que la configuration d'un serveur pour héberger le service en ligne.

I.3.2 La découverte :

La découverte de services web est le processus qui permet de trouver une description de service web utilisable et inconnue par les machines sur un réseau, tout en répondant à certains critères fonctionnels et/ou non fonctionnels.

Le but d'un service de découverte est de simplifier la recherche de services sur Internet en utilisant des agents tels que des demandeurs, des fournisseurs ou des intermédiaires. Ces agents ont pour rôle respectif de formuler les requêtes, créer les services web et les publier. La découverte d'un service web implique plusieurs étapes :

1. Le service de découverte met en relation le demandeur et le fournisseur.
2. Le demandeur fournit au service de découverte les spécifications fonctionnelles et/ou non fonctionnelles qu'il recherche.

3. Le service de découverte extrait les descriptions de services web fournies par les fournisseurs, tels que UDDI, WSDL, mot-clé, paramètres QoS, OWL-S, DAML-S, RDF, URI, etc.
4. Le service de découverte utilise une méthode de description sémantique pour faire correspondre les spécifications du demandeur avec les descriptions fournies par les fournisseurs.
5. Le service de découverte exécute un mécanisme de matching pour évaluer le degré de correspondance entre la requête du demandeur et les descriptions des fournisseurs.
6. Le service de découverte sélectionne les services web qui correspondent le mieux à la requête du demandeur.
7. Si nécessaire, le service de découverte fournit également un service de composition de services web pour répondre aux différents besoins du demandeur exprimés dans sa requête, étant donné qu'il est de plus en plus difficile de trouver des services web atomiques qui répondent à tous les besoins exprimés [8].

I.3.3 La sélection

Pour choisir le fournisseur optimal d'un service web parmi un groupe de fournisseurs, la sélection des services web vise à identifier ceux qui répondent le mieux aux exigences de l'utilisateur en termes de besoins fonctionnels et non fonctionnels.

Les critères de confidentialité des données sont souvent utilisés pour exprimer les besoins non fonctionnels des services web. Dans le cadre d'une sélection de services web basée sur la confidentialité, deux situations peuvent se présenter :

Si la réponse à la requête d'un client nécessite un seul service web non composite, la sélection est simple : le fournisseur offrant le meilleur critère de confidentialité des données sera choisi pour répondre à la demande. En revanche, si la réponse à la requête du client nécessite la combinaison de plusieurs services existants, la sélection devient plus complexe, car il faut choisir la combinaison de services composants qui répond le mieux aux besoins du client [8].

I.3.4 La composition

La composition est une procédure qui permet de créer des services complexes en combinant des services atomiques de manière ordonnée, pour répondre à des demandes complexes qui ne peuvent pas être satisfaites par des services atomiques isolés.

Les services web composites ont la capacité de négocier et de communiquer intelligemment pour découvrir automatiquement d'autres services qui pourraient améliorer le service composite.

Il existe deux principales approches pour la composition des services web :

I.3.4.1 Orchestration

La composition de services par orchestration consiste à assembler des services selon un ordre et un flux d'exécution préétablis. L'exécution de la composition est supervisée par un coordinateur de services, ce qui implique une logique d'exécution interprétée par ce dernier [9].

I.3.4.2 Chorégraphie

La Chorégraphie est un processus collaboratif où chaque participant décrit son propre rôle et itération. L'interaction des différentes parties, chacune suivant son propre rôle de manière autonome, est utilisée pour déterminer le comportement global du processus [9].

I.3.5 Exécution

Une fois que le service web a été déployé et publié dans un registre de services, il devient opérationnel et peut être appelé par les utilisateurs pour exécuter ses fonctionnalités. Ainsi, l'exécution du service web peut commencer [10].

I.4 Les services web sémantiques

Un service Web sémantique est un service Web dont la description utilise des annotations sémantiques dans un langage précis, telles que des ontologies, pour lui donner une interface interprétable sans ambiguïté. Cette approche facilite l'automatisation de certaines tâches telles que la découverte, la sélection, l'invocation et la composition du service [11].

- La découverte automatique de services Web est actuellement effectuée manuellement par les utilisateurs, qui doivent utiliser des moteurs de recherche ou des annuaires pour trouver les services, lire les pages Web qui les décrivent, puis les exécuter manuellement pour vérifier qu'ils répondent à leurs attentes [11].
- L'invocation automatique de services Web consiste en l'exécution du service par un programme informatique ou un agent logiciel. Cet agent doit être capable de comprendre les descriptions de services afin de fournir les données nécessaires pour l'exécution du service Web [11].
- La composition automatique de services Web est souvent nécessaire pour atteindre l'objectif de l'utilisateur en utilisant plusieurs services Web. L'agent logiciel chargé d'atteindre cet objectif doit disposer des données nécessaires pour sélectionner, composer et inter-opérer automatiquement ces services Web [11].

I.4.1 L'infrastructure des services Web sémantiques

Le développement des services Web sémantiques (SWS) se réalise en trois dimensions : les activités d'utilisation, l'architecture et l'ontologie de service. Chacune de ces dimensions est liée aux besoins des SWS, respectivement au niveau de l'application, de l'infrastructure physique et du conceptuel. Les activités d'utilisation définissent les besoins fonctionnels que doit soutenir un framework implémentant les SWS. L'architecture des SWS décrit les

composants nécessaires pour réaliser ces activités, tandis que l'ontologie de service intègre tous les concepts utilisés pour décrire les services Web sémantiques (voir figure I.5) [12].

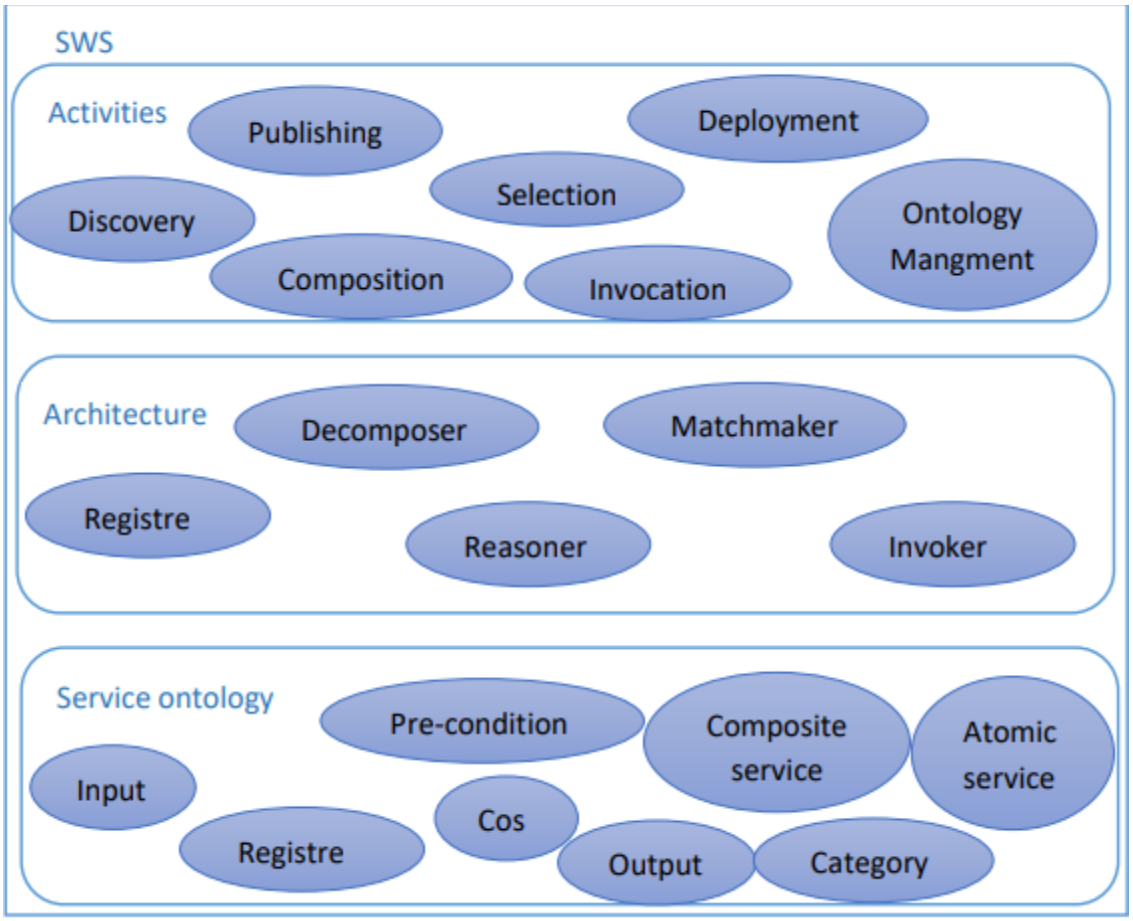

Figure I-5 Infrastructure des services Web sémantiques [13].

I.4.2 Approches proposées pour la réalisation des SWS

Différentes approches ont été présentées dans la littérature pour la mise en œuvre des services Web sémantiques, telles que WSDL-S, OWL-S et WSMO [11].

I.4.2.1 WSDL-S (Web Service Description Language-Semantic)

WSDL-S (Web Service Description Language-Semantic) est un langage de description sémantique des services Web qui permet de lier les descriptions fonctionnelles existantes dans le WSDL d'un service Web à une sémantique, offrant ainsi un mécanisme efficace [12].

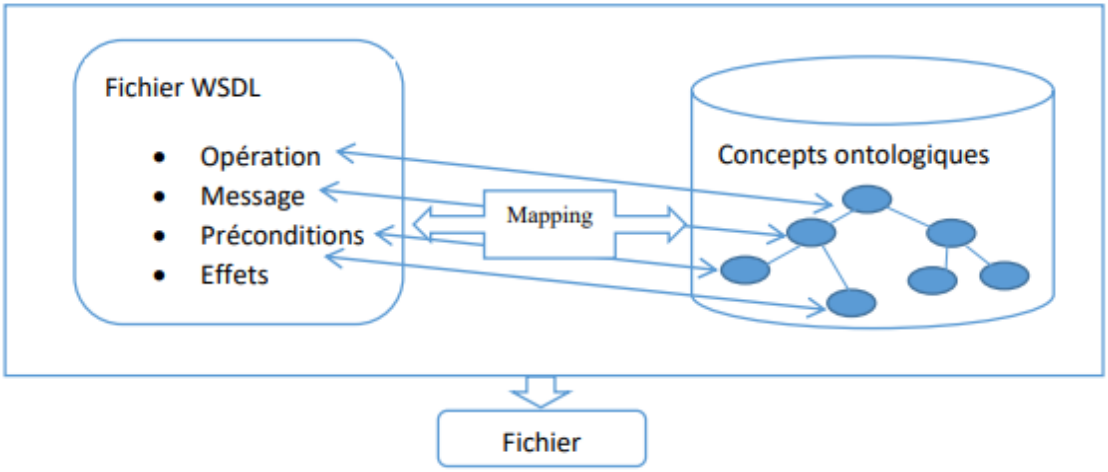

Figure I-6 Mapping entre le fichier WSDL et les concepts ontologiques [14].

I.4.2.2 OWL-S (Ontology Web Language for Services):

L'OWL-S (Ontology Web Language for Services) est un langage d'ontologie pour les services Web sémantiques. Il fournit une série d'ontologies de haut niveau pour décrire les différents aspects d'un service Web sémantique. Ces ontologies sont organisées en trois zones conceptuelles : « Service Profile », « Service Model » et « Service Grounding ».

- La classe Service Profile fournit à un agent les informations nécessaires pour publier ou découvrir un service.
- La classe Service Model définit les conditions requises pour l'exécution d'un service ainsi que ses résultats attendus. Elle inclut des informations sur les entrées, les sorties, les pré-conditions et les effets du service.
- La classe Service Grounding précise la manière dont un agent peut accéder au service en question [12].

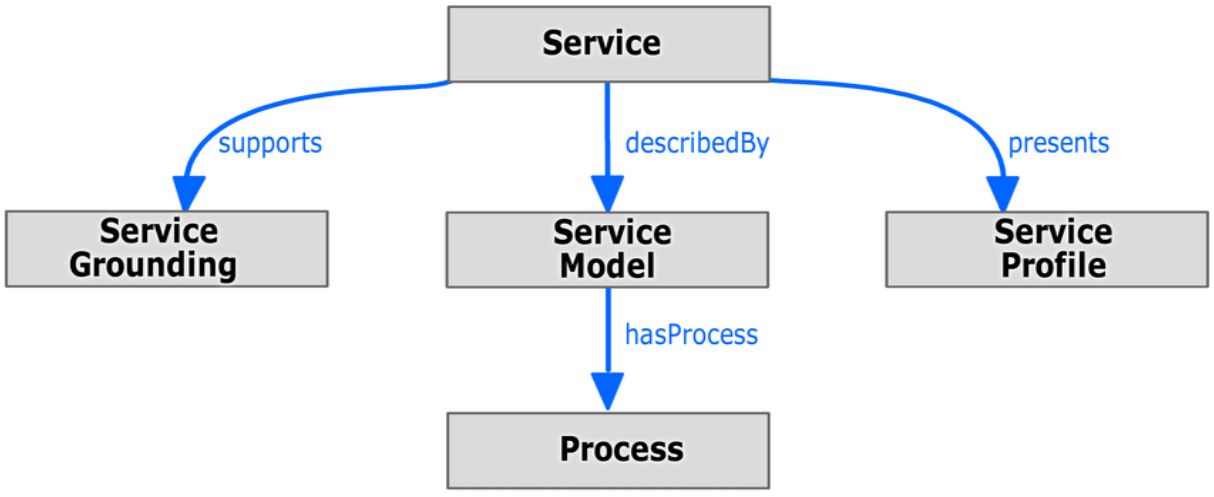

Figure I-7 Les éléments de base d'OWL-S [15].

I.4.2.3 Le WSMO (Web Service Modeling Ontology):

Le WSMO (Web Service Modeling Ontology) vise à décrire de manière exhaustive tous les aspects liés aux services Web sémantiques en proposant un modèle conceptuel fondé sur le WSMF (Web Services Modeling Framework). Le langage de spécification utilisé par le WSMO est le WSML (Web Service Modeling Language) pour ses éléments [12].

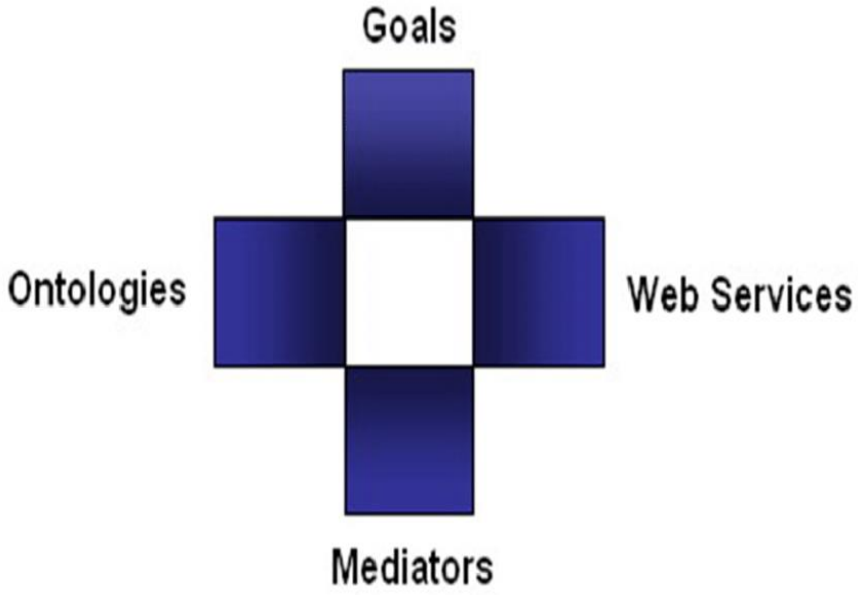

Figure I-8 Composants de WSMO [16].

I.5 Les services web sociaux

Ce type de réseaux est différent des types classiques des réseaux sociaux de personnes. Ces derniers sont basés sur la collaboration absolu et l'entraide entre leurs membres dans le sens où il n y a pas en général de compétition entre les membres. Par contre, dans les réseaux sociaux des Web services (WS), les membres sont principalement compétitifs car chaque WS souhaite faire partie des compositions, remplacer un WS défectueux, ou être nommé comme un WS supplémentaire [17].

Dans les travaux de [18], les auteurs expliquent que construire un réseau social revient à identifier les types de nœuds et de bords qui constituent ce réseau. Dans ce contexte, les services Web sont les seuls constituants et ils désignent des nœuds. En termes de bords, ils proposent trois types d'association pertinente entre les services Web, à savoir : Recommandation(R), Similarité (S) et Collaboration (C).

Dans l'article [19], les auteurs mettent en avant la valeur de l'ajout des réseaux sociaux aux services Web dans le sens où quand les entreprises découvrent et engagent les services

Web pour des besoins de travail, ils sont inclus dans les compositions des services basés à la fois sur les fonctionnalités qu'ils offrent et de la qualité du service qu'ils peuvent garantir. Ce qui implique forcément le besoin de contrats. Cependant, lorsque les consommateurs s'engagent et composent des services, c'est beaucoup plus informel et dynamique, les services Web sont destinés à cohabiter et à être composés, et leurs fonctionnalités et QoS sont interdépendantes avec d'autres services. De plus, ils sont exécutés à distance et avec un certain degré d'autonomie. Leur découverte et leur engagement ultérieur deviennent ainsi des activités sociales, tout comme la collaboration et les interactions soutenues dans les réseaux sociaux [17].

I.6 La découverte des services web

La découverte de services web peut être abordée selon trois perspectives : syntaxique, sémantique et sociale.

I.6.1 Découverte syntaxique

On utilise généralement des mécanismes de découverte syntaxique qui comparent les mots clés de la requête du client à ceux extraits de la description des services (tels que WSDL). Un parseur et un dictionnaire électronique sont utilisés pour extraire les mots clés de la requête et de la description du service. L'algorithme de matching calcule ensuite le degré de similarité entre eux en utilisant une représentation vectorielle et en mesurant le cosinus. Cependant, le principal inconvénient de ces méthodes syntaxiques est qu'elles ne prennent pas en compte l'aspect sémantique de la requête [20].

I.6.2 Découverte sémantique

Malheureusement, la découverte des services Web syntaxique, ne prendre pas en compte l'aspect sémantique essentiel pour répondre aux besoins de l'utilisateur.

La phase de découverte des services Web peut bénéficier de l'utilisation de la technologie Web sémantique pour permettre la découverte sémantique des services Web. Cette méthode repose sur le raisonnement sémantique, ce qui permet d'améliorer la précision des résultats de recherche par rapport aux techniques de découverte de services Web traditionnelles. En utilisant la puissance de calcul pour effectuer des correspondances de manière plus précise, la découverte sémantique des services Web est plus précise et efficace.

La découverte sémantique de services Web consiste à utiliser les concepts d'une ontologie pour évaluer le degré de correspondance entre les mots clés de la requête et ceux de la description du service. Pour ce faire, deux outils principaux sont généralement utilisés pour implémenter les différents concepts et relations sémantiques.

- **Ontologie**

Une ontologie est une structure de connaissances organisée en un ensemble de concepts interconnectés. Elle peut être basée sur des relations sémantiques, de composition et d'héritage, et elle est souvent utilisée dans un domaine particulier de la connaissance. Cependant, contrairement à d'autres domaines, il n'y a pas d'ontologie spécifique pour les services web, ce qui rend leur développement long et laborieux. Par exemple, la création d'une ontologie pour un domaine spécifique tel que la médecine, l'aéronautique ou la finance nécessite un effort considérable [20].

- **WordNet**

La base de données lexicale WordNet, disponible en anglais (mais aussi sous d'autres versions pour des langues comme le français avec WOLF), offre une multitude de relations conceptuelles sémantiques et lexicales entre les différents mots de la langue. Les groupes de synonymes cognitifs que l'on trouve dans WordNet, appelés Synsets, permettent d'enrichir la compréhension et la précision de la recherche d'informations sémantiques [20].

I.6.2.1 Limite de découverte des services web sémantique

Bien que la découverte sémantique des services apporte une précision accrue, il est important de noter que cette méthode peut être coûteuse et complexe à développer. En effet, la création d'une ontologie nécessite des experts du domaine de connaissance ainsi que des ressources pour la modélisation, la validation et la maintenance de l'ontologie. Cela peut engendrer une augmentation de la charge de travail et des coûts associés au développement et à la maintenance de systèmes basés sur des ontologies.

De plus, il existe des problèmes de perte de temps et d'efficacité avec la découverte sémantique. Cette méthode peut prendre du temps pour analyser les requêtes de l'utilisateur et proposer des services pertinents. Si les services déjà consultés ne sont pas enregistrés, cela peut entraîner une perte de temps et d'efficacité, car les suggestions de services proposées peuvent ne pas être adaptées aux besoins de l'utilisateur.

I.6.3 La découverte sociale

Les services Web sociaux ont émergé de la fusion entre les services Web et les réseaux sociaux, ajoutant ainsi des caractéristiques sociales aux services Web grâce à leur interaction [21].

I.6.3.1 Les approches de la découverte des services web sociaux

Il existe différentes approches pour découvrir des services web sociaux. Certaines méthodes utilisent une recherche basée sur le contexte et le domaine de l'industrie pour assurer une découverte appropriée des services web [21].

- **Approches à base d'analyse des réseaux sociaux**

Le problème de la découverte de services Web répondant aux exigences des utilisateurs a été abordé en se basant sur les interactions passées entre les services Web afin de construire un réseau social. Ce réseau social permet à un service Web de recommander un autre service Web avec lequel il souhaite collaborer, tout en permettant de connaître les services Web en compétition pour la sélection.

Les services Web sont représentés par des nœuds dans le réseau social, tandis que les arcs représentent les relations sociales qui les relient. Ces relations ont été classées en fonction de deux axes : similaire (compétition, substitution) ou différent (collaboration).

Cette approche vise à améliorer l'efficacité du registre UDDI en prenant en compte les interactions passées entre les services Web. Elle se fonde sur la similarité entre les services pour construire un réseau social de services Web sémantiques, afin de faciliter la découverte des services Web pertinents.

Le but principal de cette étude est de consolider les services Web en un seul service atomique, afin de favoriser la collaboration et d'éviter la concurrence. Pour atteindre cet objectif, les services Web similaires ont été rassemblés dans des communautés qui décrivent une fonctionnalité spécifique, sans faire référence à un service Web en particulier. Un service Web a été choisi comme « maître » dans chaque communauté, chargé de gérer les membres « esclaves » et de faciliter la gestion de la communauté dans son ensemble.

Cette méthode offre aux utilisateurs la possibilité de sélectionner facilement des services Web, indépendamment de leur regroupement dans des communautés. Des chercheurs ont proposé un réseau social de Cloud appelé « Sky », qui permet de découvrir des Clouds isolés. Cette approche présente plusieurs avantages, notamment une disponibilité accrue et une fiabilité croissante des services Web [21].

- **Approches à base de confiance**

Plusieurs études ont utilisé la notion de confiance pour résoudre le problème de la découverte des services Web. Par exemple, une proposition a été faite pour un modèle flou basé sur la confiance et la réputation des services Web.

Le modèle présenté prend en compte trois paramètres d'entrée, à savoir la confiance d'interaction, la réputation des témoins et la réputation certifiée, pour calculer une valeur globale de confiance en sortie.

La logique floue offre une méthode simple pour traiter des informations imprécises, ambigües, bruyantes ou manquantes en entrée et parvient à une conclusion basée sur celles-ci. Cette méthode simule la façon dont une personne prend des décisions et raisonne. Les chercheurs ont utilisé des triangles et des trapèzes flous pour représenter de manière adéquate les connaissances de confiance données. Ils ont également intégré la dimension sociale dans la

découverte des services Web, en prenant en compte les propriétés sémantiques et structurelles des clients de service [21].

De plus, les auteurs ont intégré la notion de confiance sociale dans leur approche afin d'améliorer la découverte des services Web. Pour cela, ils agrègent trois mesures : la position sociale, la proximité sociale et la similitude sociale entre le client et le fournisseur de services. Cette démarche permet aux services Web d'acquérir une crédibilité plus élevée et d'optimiser leur visibilité auprès des utilisateurs.

Les auteurs abordent la question de l'attraction des services Web dans les réseaux sociaux en établissant des critères permettant d'attirer les services Web et de garantir leur durabilité. Parmi ces critères figurent la confidentialité, la confiance, la fiabilité et la disponibilité. Ces critères permettent d'évaluer l'attractivité du réseau social en termes d'utilité et de confidentialité, ainsi que d'aider le service Web à choisir le réseau social avec lequel il souhaite s'associer. En outre, chaque critère cité ci-dessus permet :

• De protéger les services Web contre les utilisateurs malveillants.
• De garantir la confiance des services Web dans les pairs qu'ils recommandent dans les compositions.
• De surveiller les opérations des services Web pour garantir la qualité de service [21].

I.7 Conclusion

Ce chapitre a permis d'avoir un aperçu général sur les services web, ainsi que sur la découverte de ces services et les différentes approches qui ont été proposées.

Au fil du temps, la découverte de services Web a évolué d'une approche syntaxique à une approche sémantique, grâce au développement du Web sémantique. Cette évolution a conduit à l'utilisation d'ontologies pour décrire et découvrir les services Web. Cependant, la mise en œuvre d'ontologies peut être complexe et les systèmes sémantiques manquent souvent de mémoire pour stocker l'historique des requêtes et des résultats. Les services Web sociaux sont donc apparus comme une alternative, permettant aux utilisateurs de collaborer pour créer, partager et découvrir des services Web de manière plus informelle.

Chapitre II

Les technique DL NLP Pour avoir la sémantique

Sommaire

II.1 Introduction 18
II.2 Le traitement du langage naturel (NLP) 18
II.3 Deep learning 20
II.4 Word Embedding 20
II.4.1 Les Modèles de WE 20
II.5 Les techniques deep learning pour le Traitement Automatique du Langage Naturel 24
II.5.1 Réseaux de Neurones Récurrents (RNN) 24
II.5.2 Réseaux Long Short-Term Memory (LSTM) 25
II.5.3 Le mécanisme d'Attention 27
II.5.4 Transformers 27
II.6 BERT 32
II.6.1 Introduction 32
II.6.2 Les types de BERT 33
II.6.3 L'architecture de BERT 34
II.6.4 Le mécanisme d'attention dans BERT 36
II.6.5 Autres variantes de BERT 38
II.6.6 Description des taches de MLM &NSP 39
II.6.7 Prédiction de la phrase suivante (NSP) 39
II.7 Conclusion 41

II.1 Introduction

Depuis l'avènement de l'informatique, les êtres humains ont cherché à développer des interactions plus naturelles et intuitives avec les machines. Bien que les différents langages de programmation permettent la communication entre humains et machines, il est préférable que cette communication soit plus fluide et naturelle. Ainsi, il est crucial que les machines puissent comprendre le langage naturel des utilisateurs et fournir des réponses facilement compréhensibles pour les humains.

Ce processus est étudié dans la discipline du Traitement Automatique du Langage Naturel (TALN) ou Natural Language Processing (NLP) en anglais. Le TALN se concentre sur la compréhension, la manipulation et la génération du langage naturel par les machines.

Par langage naturel, on entend le langage utilisé par les humains dans leur communication quotidienne, par opposition aux langages artificiels tels que les langages de programmation ou les notations mathématiques [22].

II.2 Le traitement du langage naturel (NLP)

Le NLP, ou Traitement du Langage Naturel, est un domaine de l'informatique qui vise à permettre aux ordinateurs de comprendre le langage de manière similaire aux humains. Il englobe diverses tâches telles que l'analyse des sentiments, la reconnaissance vocale et la génération de réponses aux questions. Le NLP existe depuis les années 1940, au début du développement de l'informatique.

Le NLP comprend plusieurs sous-domaines importants, notamment :

- **Compréhension du Langage Naturel (NLU) :** Cela consiste à comprendre en profondeur les échanges et les données linguistiques afin d'identifier les intentions derrière les paroles ou les écrits humains.
- **Génération du Langage Naturel (NLG) :** Cette étape implique la création automatique de textes dans une langue donnée grâce à l'intelligence artificielle. Les données sont transformées en textes, ce qui permet aux entreprises d'automatiser certains processus manuels [23].

Les ordinateurs n'apprennent pas les langues de la même manière que les humains. Ils utilisent des techniques spécifiques pour analyser et comprendre le langage naturel. De manière générale, on distingue trois approches principales en NLP : les méthodes basées sur des règles, les modèles classiques d'apprentissage automatique et les modèles d'apprentissage en profondeur [24].

- **Méthodes basées sur des règles :**

Ces méthodes reposent principalement sur des règles spécifiques à un domaine et peuvent être utilisées pour résoudre des problèmes simples, comme l'extraction de données structurées à partir de textes non structurés, par exemple des pages web. Cependant, ces

méthodes simples peuvent être limitées par la complexité du langage naturel et s'avérer inefficaces [24].

- **Modèles classiques d'apprentissage automatique :**

Ces approches utilisent des techniques d'apprentissage automatique pour résoudre des problèmes plus complexes. Contrairement aux méthodes basées sur des règles, elles se concentrent réellement sur la compréhension du langage. Elles utilisent des données textuelles prétraitées ainsi que des informations telles que la longueur des phrases et l'occurrence de mots spécifiques. Des modèles statistiques d'apprentissage automatique tels que Naive Bayes ou la régression logistique sont souvent utilisés [24].

- **Modèles d'apprentissage en profondeur :**

Les recherches actuelles se concentrent sur l'utilisation de modèles d'apprentissage en profondeur pour le NLP. Ces modèles surpassent souvent les approches classiques, car ils nécessitent moins de prétraitement des données textuelles. Les couches de neurones peuvent être considérées comme des extracteurs automatiques de caractéristiques, permettant ainsi de construire des modèles de bout en bout avec peu de prétraitement. Les algorithmes de Deep Learning ont généralement des capacités d'apprentissage plus puissantes que ceux de l'apprentissage automatique classique, ce qui conduit à de meilleurs résultats sur des tâches complexes de NLP, telles que la traduction [24].

La figure II.1 illustre la relation de NLP avec l'intelligence Artificiel, Machine Learning et Deep Learning.

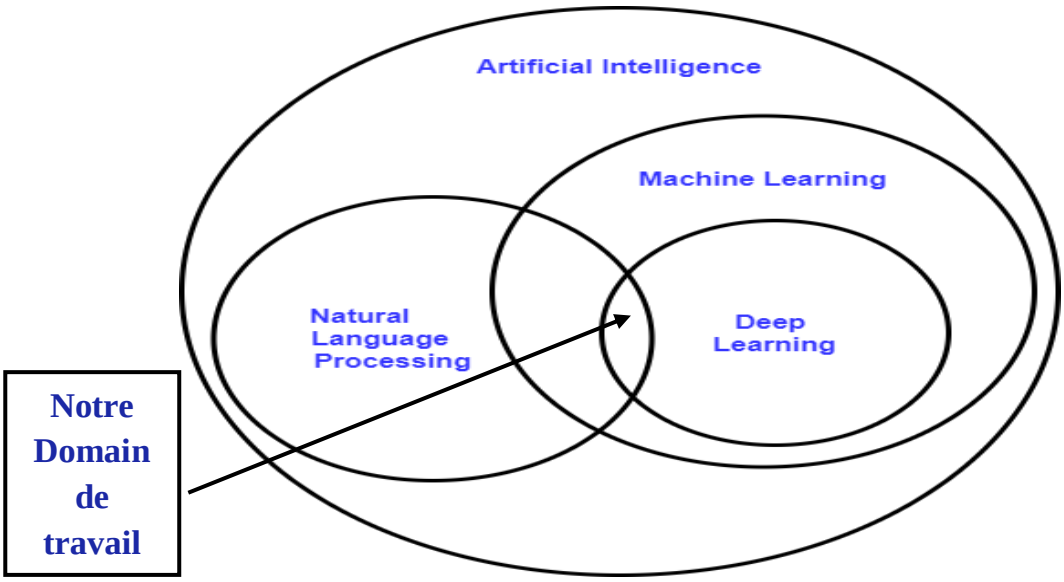

Figure II-1 Relation entre NLP, AI, ML et DL.

Le Deep Learning a eu un impact significatif sur le domaine du NLP, permettant des avancées majeures dans différentes tâches de traitement du langage naturel.

II.3 Deep Learning

Le Deep Learning repose sur l'utilisation de réseaux de neurones artificiels et est spécifiquement conçu pour gérer de grandes quantités de données en ajoutant des couches au réseau.

Un modèle de Deep Learning est capable d'extraire des caractéristiques à partir des données brutes en passant par plusieurs couches de traitement, qui consistent en des transformations linéaires multiples. Il apprend progressivement à partir de ces caractéristiques à travers chaque couche, nécessitant une intervention humaine minimale [25].

Une méthode importante développée grâce au Deep Learning est celle du Word Embedding (incorporation de mots) [22].

II.4 Word Embedding

Le Word Embedding, connu également sous le nom d'incorporation de mots, est une méthode d'encodage qui vise à représenter les mots ou les phrases d'un texte à l'aide de vecteurs numériques dans un modèle vectoriel. L'utilisation de cette technique dans le domaine du traitement du langage naturel (NLP) a permis d'obtenir des représentations quantitatives des mots, lesquelles sont utilisées pour mesurer la similarité sémantique entre les mots, effectuer des tâches de classification de texte, et générer du texte de manière plus précise et cohérente. Cette avancée a entraîné une amélioration significative des performances des modèles de traitement du langage naturel, tout en permettant une meilleure compréhension des structures linguistiques.

L'aspect clé de cette nouvelle représentation est que les mots qui apparaissent dans des contextes similaires ont des vecteurs correspondants relativement proches. Par exemple, si nous représentons les mots « roi » et « reine » par des vecteurs dans un espace vectoriel, nous observons qu'ils sont relativement proches l'un de l'autre. Cette technique s'appuie sur l'hypothèse de Harris, également connue sous le nom d'hypothèse distributionnelle, qui postule que les mots ayant des contextes similaires ont des significations liées [25].

II.4.1 Les Modèles de WE

Il existe différentes techniques de Word Embeddings (WEs) qui se composent de deux parties distinctes. La première partie, principalement basée sur des pointeurs, fait simplement référence aux techniques classiques « statiques » des WEs, où le même mot aura toujours la même représentation, indépendamment du contexte dans lequel il se trouve. Parmi ces techniques, nous pouvons citer Word2Vec et GloVe, par exemple.

La deuxième partie présente de nouvelles techniques de WEs qui prennent en compte le contexte du mot et peuvent être considérées comme des techniques de WEs contextuelles « dynamiques ». La plupart de ces approches utilisent un modèle de langage pour aider à modéliser la représentation des mots. Un exemple typique de cette catégorie est BERT [26].

II.4.1.1 Word Embeddings Classique

- **Word2vec**

Word2vec est une méthode utilisant des réseaux de neurones artificiels pour représenter et capturer les régularités sémantiques et syntaxiques des mots. Elle offre deux architectures : le modèle CBOW (sac-de-mots continus) et le modèle Skip-gram.

Ces modèles se composent de trois couches : une couche d'entrée, une couche cachée et une couche de sortie. Dans le modèle CBOW, la couche d'entrée contient un sac-de-mots, tandis que dans le modèle Skip-gram, elle contient un mot unique. La couche cachée projette les mots d'entrée dans une matrice de poids partagée par tous les mots, et la couche de sortie utilise des neurones softmax[2].

Pour gérer la complexité de la fonction softmax, les auteurs ont introduit deux alternatives : les « échantillons négatifs » et le « softmax hiérarchique ». Ces techniques permettent d'entraîner les modèles sur de grandes quantités de texte et d'obtenir des représentations de meilleure qualité que les modèles plus complexes basés sur la récurrence .

Les modèles Word2vec sont capables de capturer des relations sémantiques et syntaxiques complexes. Par exemple, la distance entre les projections de deux mots peut représenter des notions telles que le singulier-pluriel ou le masculin-féminin, comme illustré dans la Figure (II .2) [27].

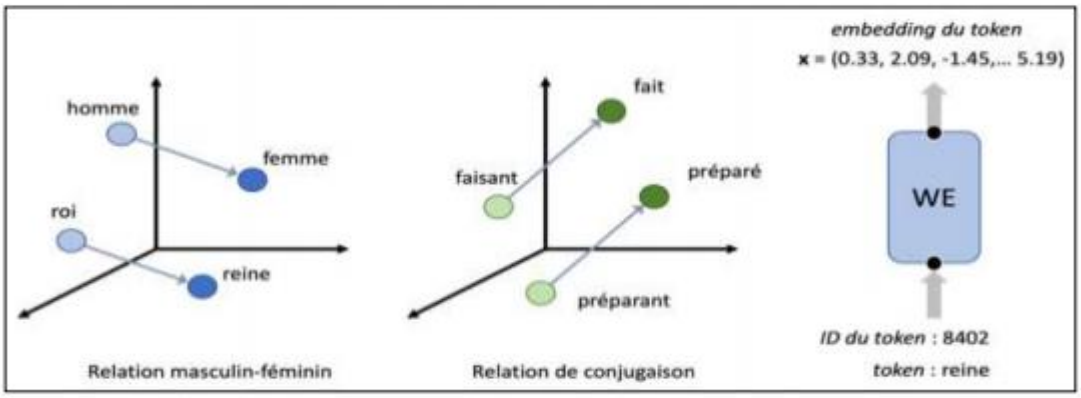

Figure II-2 Exemples de relations de mots dans l'espace Word2vec [28].

- **CBOW**

Dans la Figure (II .3), l'architecture CBOW (Continuous Bag-of-Words) est un réseau de neurones qui prédit le poids d'un mot en fonction de son contexte, c'est-à-dire des mots précédents et des mots suivants. Dans cette architecture, la couche de projection est partagée par tous les mots, ce qui signifie que tous les mots sont projetés dans la même position. Ce

[2] Softmax : Est une fonction transforme un vecteur de valeurs réel en une distribution de probabilités normalisée.

modèle est appelé CBOW pour deux raisons : d'une part, l'appellation « sac-de-mots » indique que l'ordre des mots dans le contexte n'a pas d'influence sur la projection ; d'autre part, l'objectif met l'accent sur l'utilisation d'une représentation continue (WEs) des mots en contexte, ce qui diffère des modèles « sac-de-mots » standards. Dans ce cas, l'apprentissage des WEs consiste à prédire un mot en fonction de son contexte. Cela est réalisé en calculant la somme des WEs du contexte, puis en appliquant un classifieur log-linéaire sur le vecteur résultant pour prédire le mot cible. Enfin, le modèle compare sa prédiction avec la réalité et ajuste la représentation vectorielle du mot en utilisant la rétro-propagation du gradient [27].

- **Skip-Gram (SG)**

L'architecture Skip-Gram tente de prédire le contexte à partir d'un mot donné. La couche d'entrée de ce réseau est donc un vecteur contenant un seul mot. Le mot est projeté dans la couche cachée, puis dans la couche de sortie. Le contexte est ensuite réduit de manière aléatoire à chaque itération. Le vecteur de sortie est comparé à chacun des mots du contexte réduit, et le réseau est corrigé en utilisant la rétro-propagation du gradient.

De cette manière, la représentation du mot d'entrée se rapproche de chacun des mots présents dans le contexte [27].

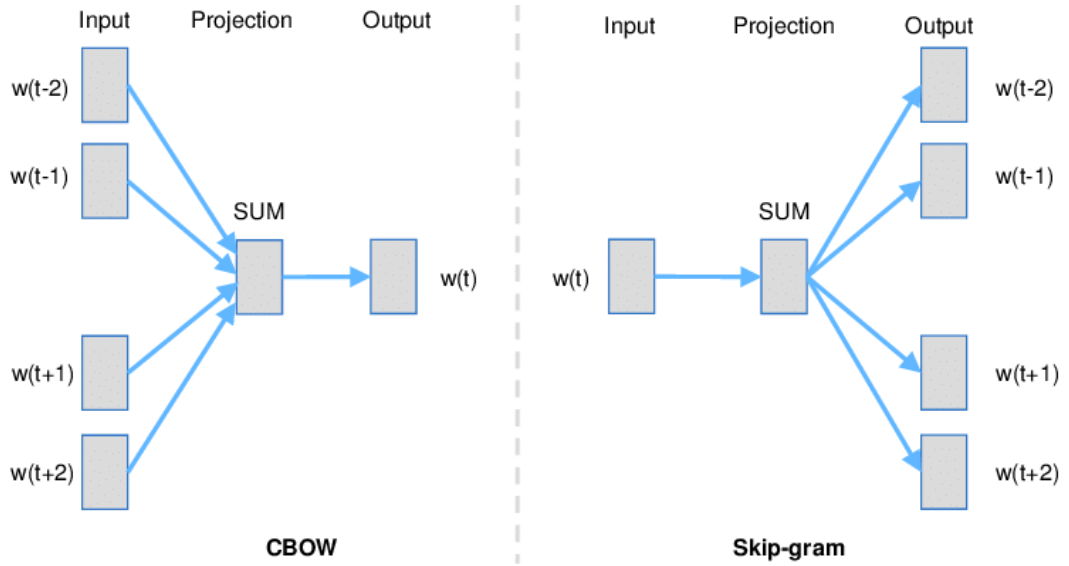

Figure II-3 Exemples des architectures CBOW et Skip-gram de Word2vec [29].

- **Glove**

Glove (Global Vectors for Word Representation) est un algorithme d'apprentissage non supervisé qui permet d'obtenir des représentations vectorielles pour les mots. Il se base sur des statistiques globales de cooccurrence de mots à partir d'un corpus, et les représentations résultantes présentent des structures linéaires intéressantes dans un espace vectoriel [30].

L'approche de Glove repose sur la création d'une matrice de cooccurrence globale des mots (MG) en utilisant une fenêtre contextuelle glissante pour analyser le corpus. Chaque élément de la matrice MGij représente le nombre de fois où le mot mj apparaît dans le contexte du mot mi.

Une fois la matrice MG calculée, un modèle de régression par moindres carrés est utilisé pour entraîner la construction des représentations vectorielles $\overrightarrow{m_i}$ et $\overrightarrow{m_j}$.

Ces représentations vectorielles doivent conserver des informations précieuses sur la cooccurrence des paires de mots mi et mj, telles que :

$$\overrightarrow{m_i} . \overrightarrow{m_j} + b_i + b_j = \log(MG_{ij})$$

MG_{ij} représente le nombre de occurrences du mot j dans le contexte du mot i est représenté par MGij. Les biais scalaires bi et bj sont associés respectivement aux mots i et j [27].

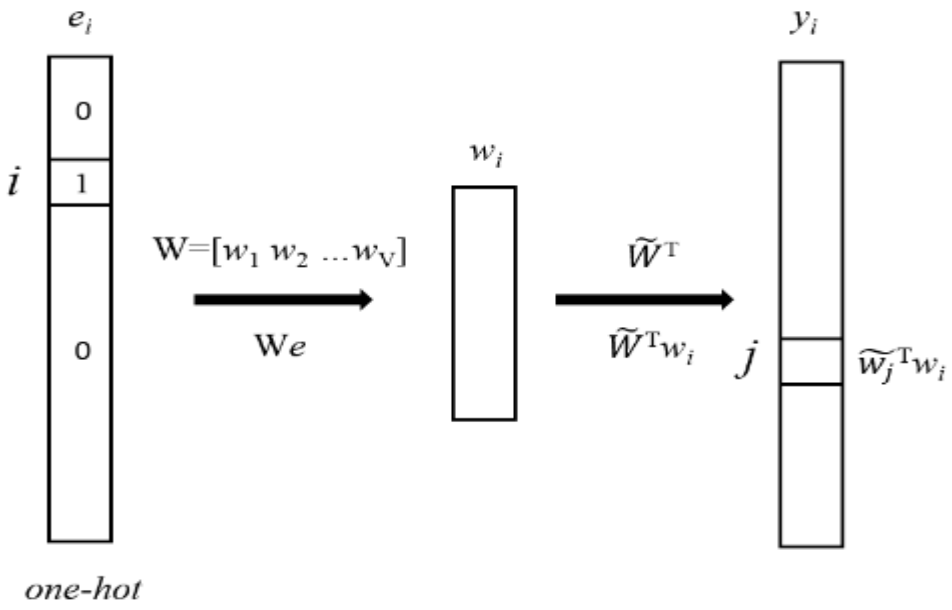

Figure II-4 Exemple d'architecture de GLOVE [29].

Dans la figure (II.4) Pour l'entrée, un mot est représenté sous la forme d'un vecteur one-hot. Les matrices WEs agissent en tant que matrices de poids dans ce modèle, ce qui permet à la sortie du modèle d'être un vecteur contenant les produits internes des vecteurs de mots [27].

II.4.1.2 Word Embeddings Contextuelle

- **BERT**

BERT, ou Bidirectional Encoder Representations from Transformers, est un modèle révolutionnaire pour la représentation de textes écrits en langage naturel. Contrairement aux

méthodes d'embedding traditionnelles, la particularité de BERT réside dans sa capacité à créer des représentations contextuelles des mots. Cela signifie qu'un mot n'est pas statiquement représenté, mais sa signification est déterminée en fonction du contexte dans lequel il est utilisé. Par exemple, le mot « baguette » aura des représentations différentes dans les phrases « la baguette du magicien » et « la baguette du boulanger ». De plus, BERT prend en compte le contexte de manière bidirectionnelle en considérant à la fois les mots précédents et les mots suivants dans une phrase.

Le principe d'utilisation de BERT est simple : il est préalablement entraîné sur une vaste quantité de données, puis il peut être adapté à des tâches spécifiques en le ré-entraînant avec nos propres données. Cette adaptation généralement implique l'ajout d'un réseau de neurones à la sortie de BERT, connu sous le nom de fine-tuning [32]. Cette approche permet de tirer parti des connaissances préalables de BERT tout en l'ajustant pour s'adapter aux exigences particulières de la tâche en question.

II.5 Les techniques Deep Learning pour le NLP

Voici quelques-unes des techniques de Deep Learning couramment utilisées dans le NLP :

II.5.1 Réseaux de Neurones Récurrents (RNN)

Les Réseaux de Neurones Récurrents (RNN) sont une variante essentielle des réseaux neuronaux largement utilisés dans le traitement du langage naturel. Ils sont particulièrement adaptés au traitement de données séquentielles telles que des séquences de texte, de son, d'image ou de vidéos. Contrairement aux réseaux de neurones classiques, qui considèrent uniquement des entrées indépendantes les unes des autres, les RNN sont récurrents, ce qui signifie qu'ils effectuent la même tâche pour chaque élément d'une séquence. La sortie dépend ainsi des calculs précédents, ce qui peut être visualisé avec la figure II.5.

Une manière de comprendre les RNN est de les voir comme des modèles dotés d'une mémoire qui capture des informations sur ce qui a été calculé jusqu'à présent. Les RNN permettent d'utiliser les prédictions antérieures comme entrées, grâce à des états cachés [33].

II.5.1.1 La limitation des RNN

En théorie, les réseaux de neurones récurrents (RNN) ont la capacité d'utiliser des informations sur des séquences de longueur arbitraire. Cependant, en pratique, ils sont limités à examiner seulement quelques étapes antérieures. Lorsque la séquence à traiter est trop longue, la rétro-propagation du gradient d'erreur peut provoquer soit une explosion, rendant les valeurs du gradient beaucoup trop grandes, soit une diminution drastique, rendant les valeurs du gradient beaucoup trop petites. Dans ces situations, le réseau n'est plus capable de différencier les informations pertinentes de celles qui ne le sont pas, ce qui l'empêche d'apprendre à long terme [26].

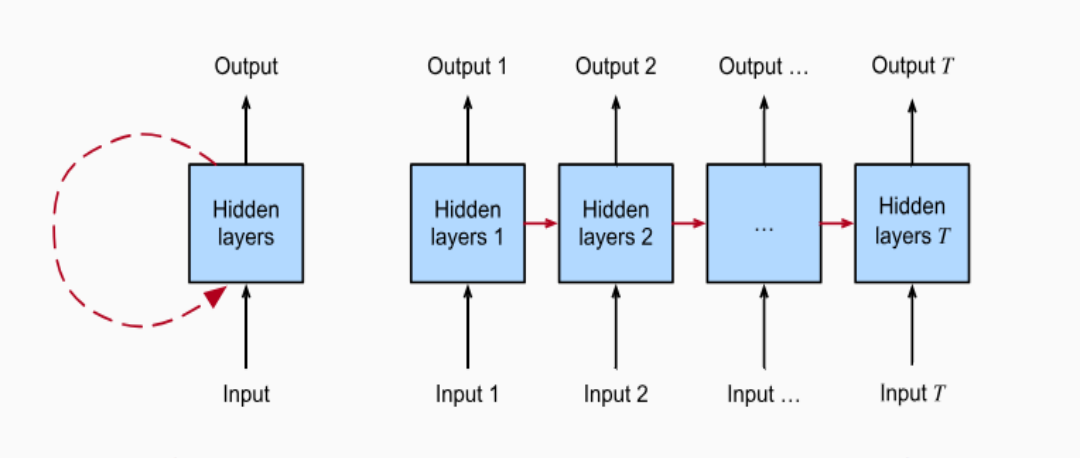

Figure II-5 L'architecture des réseaux de neurone récurrent [34].

Pour résoudre les problèmes rencontrés par les RNN standards, plusieurs variantes ont été développées, parmi lesquelles les LSTM (Long Short-Term Memory) occupent une place prépondérante.

II.5.2 Réseaux de Long Short-Term Memory (LSTM)

Les réseaux de mémoire à long terme à court terme (LSTM) constituent une extension des réseaux neuronaux récurrents, ayant été introduits par Hochreiter et Schmidhuber en 1997. Leur objectif principal est de résoudre le problème de la disparition du gradient qui se produit lorsque des éléments éloignés dans le temps doivent être pris en compte avec leur contexte. Les cellules LSTM sont conçues pour conserver un état de mémoire composé de trois portes qui régulent le flux d'informations et effectuent des actions spécifiques : la porte d'oubli (Forget gate), la porte d'entrée (Input gate) et la porte de sortie (Output gate). Ces portes peuvent être considérées comme des vannes, comme illustré dans la figure II .6, qui jouent les rôles suivants :

- La porte d'entrée décide si l'entrée doit modifier le contenu de la cellule.
- La porte d'oubli décide s'il faut réinitialiser le contenu de la cellule à zéro.
- La porte de sortie décide si le contenu de la cellule doit influencer la sortie du neurone.

Le mécanisme des trois portes est similaire : l'ouverture/la fermeture de chaque vanne est modélisée par une fonction f, généralement une sigmoïde, qui est appliquée à la somme pondérée des entrées. De plus, on retrouve deux types de sorties appelés états (Hidden state et Cell state) [35].

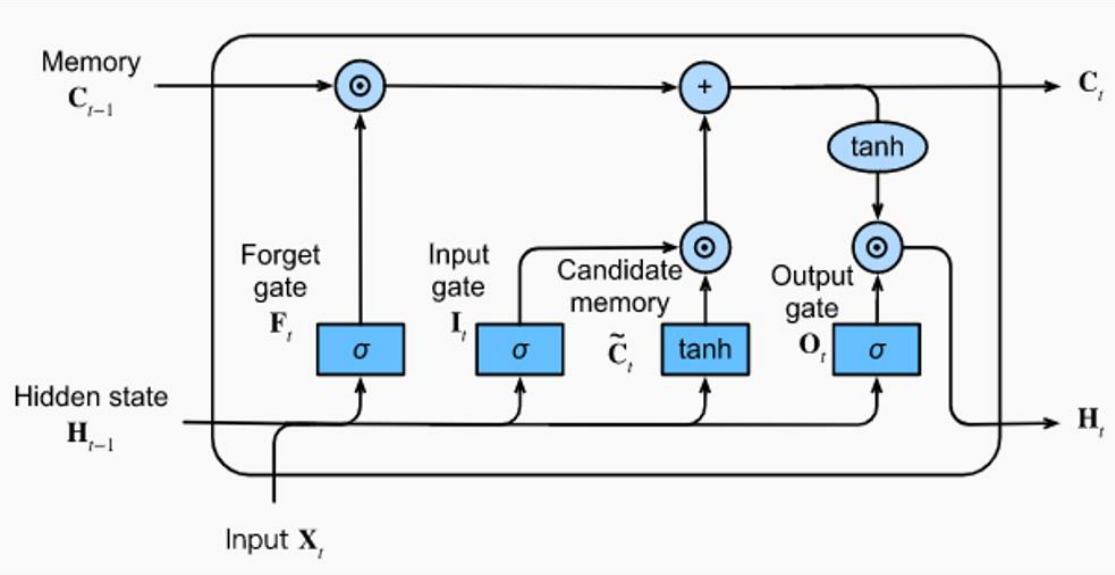

Figure II-6 Unité de base LSTM [35].

II.5.2.1 Fonctionnement du réseau LSTM

Dans ce qui suit, nous allons simplifier et clarifier le fonctionnement d'une cellule LSTM :

Un élément clé des LSTM est l'état de la cellule, représenté par une ligne horizontale dans le diagramme. Les portes permettent de contrôler le flux d'informations en laissant éventuellement passer certaines d'entre elles. Elles sont composées d'une couche de neurones sigmoïdes et d'une opération de multiplication.

La première étape du LSTM consiste à décider quelles informations doivent être oubliées ou ignorées de l'état de la cellule. Cette décision est prise par une couche sigmoïde appelée « couche de porte d'oubli ». Ensuite, nous décidons quelles nouvelles informations doivent être stockées dans l'état de la cellule, ce qui correspond à la mise à jour de l'état. Enfin, l'ancien état (reçu en entrée) est mis à jour dans le nouvel état de la cellule. En résumé, la sortie est obtenue à partir de l'état actuel de la cellule [35].

II.5.2.2 La limitation des LSTM

En dépit d'avoir été développés pour pallier certaines limitations des RNN traditionnels, les LSTM possèdent également leurs propres contraintes. Par exemple, leur complexité supérieure à celle des RNN classiques requiert l'ajustement de nombreux hyper-paramètres tels que le nombre de couches LSTM, la dimension de l'espace caché et le taux d'apprentissage pour obtenir de bonnes performances. De plus, les LSTM peuvent éprouver des difficultés à saisir les dépendances à très long terme au sein d'une séquence.

Les Transformers se présentent comme une alternative puissante aux LSTM dans le domaine du traitement du langage naturel (NLP). Ils utilisent le mécanisme d'attention, qui permet de modéliser les dépendances sans considérer leur distance dans les séquences

d'entrée ou de sortie [36]. Cette approche permet d'obtenir de meilleurs résultats en termes de modélisation des relations entre les éléments d'une séquence.

II.5.3 Le mécanisme d'Attention

Le concept d'attention mesure le degré de relation entre deux éléments de séquences distinctes. En ce qui concerne le traitement du langage naturel (NLP) dans un contexte de séquence à séquence, le mécanisme d'attention permet au modèle de déterminer quels mots de la séquence B nécessitent une attention accrue lorsqu'il traite un mot de la séquence A. Pour illustrer cela, prenons un exemple.

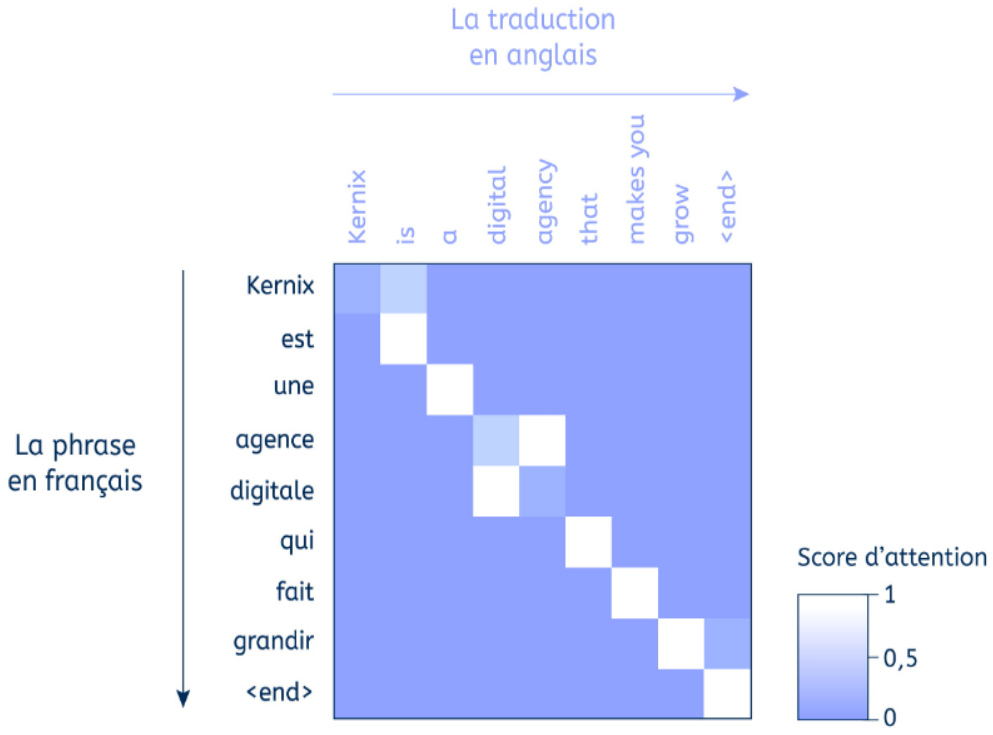

Figure II-7 Distribution de l'attention entre deux séquences [37].

Dans cet exemple particulier (II .7), plus une cellule est claire, plus la connexion entre les deux mots auxquels elle correspond est solide. Il est généralement observé qu'un mot est fortement lié à sa traduction littérale [38].

II.5.4 Transformers

Le Transformer est un modèle d'apprentissage profond largement utilisé dans le domaine du traitement du langage naturel. Il se distingue par l'utilisation de l'auto-attention, un mécanisme qui permet de pondérer différemment l'importance de chaque partie des données d'entrée.

Les Transformers ont rencontré un énorme succès dans le domaine du NLP en raison de leur capacité à capturer les dépendances à longue distance et à modéliser les relations complexes entre les mots dans une séquence. Leur architecture a inspiré la création de plusieurs modèles qui sont aujourd'hui incontournables dans le domaine du NLP.

Un Transformer est composé de deux parties : un encodeur et un décodeur, qui collaborent pour traiter les données. L'encodeur transforme la séquence d'entrée en représentations continues, tandis que le décodeur utilise ces représentations pour générer une séquence de sortie [27]. La figure II.8 illustre l'architecture globale du Transformer et son fonctionnement.

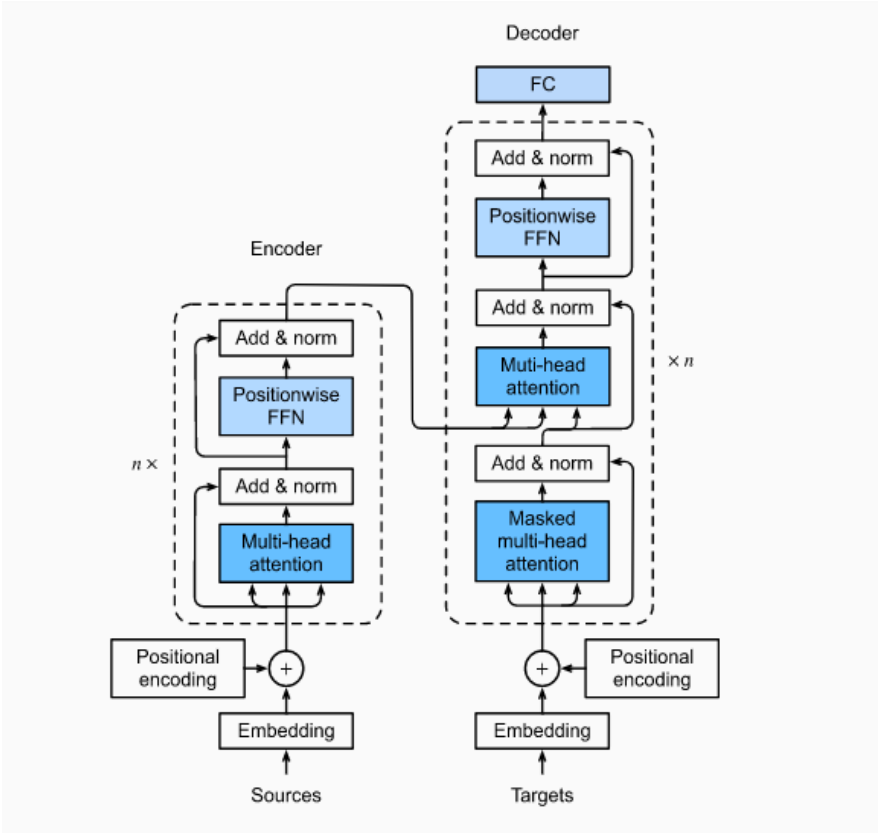

Figure II-8 Architecture globale des Transformers [39].

II.5.4.1 Les types d'attention

Il y a trois types d'attention :

- **L'attention encodeur-décodeur** :

Il s'agit de l'attention entre la séquence d'entrée et la séquence de sortie.

- **L'auto-attention dans la séquence d'entrée :**

C'est un mécanisme qui relie différentes positions au sein d'une même séquence d'entrée afin de calculer une représentation de cette séquence. En termes simples, l'auto-attention permet de créer des connexions similaires à l'intérieur de la même séquence.

- **L'auto-attention masquée dans la séquence de sortie :**

Cette forme d'auto-attention est limitée aux mots qui précèdent un mot donné. Cela évite toute fuite d'informations lors de l'apprentissage du modèle. Pour chaque étape, les mots qui apparaissent après le mot en question sont masqués. Par exemple, pour l'étape 1, seul le premier mot de la séquence de sortie n'est pas masqué. Pour l'étape 2, les deux premiers mots ne sont pas masqués, et ainsi de suite [36].

Le calcul de l'attention peut être effectué à l'aide de trois matrices : Clés, Valeur et Requête (Key, Value et Query en anglais). Ces matrices sont obtenues en multipliant le vecteur d'entrée X par des matrices de poids qui sont apprises pendant l'entraînement. Chaque colonne des matrices correspond à un mot de la séquence [36].

-Requête : $Q = XW_q$, *considéré comme le mot courant.*

-Clé : $K = XW_k$, *considéré comme un mécanisme d'indexation (unique) pour Le vecteur valeur.*

-Valeur : $V = XW_v$, *considéré comme l'information contenue dans le mot d'entrée.*

Les matrices W_q , W_k *et* W_v *sont apprises en étant entraînées conjointement lors de l'apprentissage du modèle. Le Self-Attention est calculé via la formule suivante :*

$$Attention\ (Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V,$$

Où d_k représente la dimension des matrices Q et K. Le calcul de similarité entre les matrices Q et K est effectué en utilisant le produit scalaire entre la requête et toutes les clés, suivi d'une fonction softmax. Ce produit scalaire mesure à quel point Q et K sont similaires, et plus la valeur est élevée, plus la similarité est forte.

Ensuite, cette similarité est utilisée pour pondérer les valeurs de la matrice V. Les valeurs qui sont multipliées par un softmax élevé reçoivent davantage d'attention par rapport aux autres. Cette étape permet de focaliser l'attention sur les informations les plus pertinentes.

II.5.4.2 Architecture du Transformer

Le modèle Transformers s'inspire du schéma classique Encodeur-Décodeur pour son architecture. La phase d'encodage comprend six (06) encodeurs qui sont enchaînés les uns après les autres. La phase de décodage est composée de six (06) décodeurs, également connectés en série, mais chaque décodeur prend en entrée la sortie du sixième encodeur en plus des autres informations.

Les six (06) encodeurs et décodeurs ont tous la même structure, et leur nombre peut varier selon les besoins (peu importe combien il y en a, le principe reste le même).

Chaque encodeur reçoit en entrée la sortie de l'encodeur précédent, tandis que le premier encodeur reçoit un vecteur d'embedding en tant qu'entrée initiale. De même, chaque décodeur prend en entrée la sortie du décodeur précédent ainsi que les mots déjà encodés. Le dernier décodeur est connecté à un bloc de Réseau de Neurones Linéaire suivi d'une fonction Softmax [38].

II.5.4.2.1 Couche d'encodeur

Nous avons maintenant la couche d'encodeur. Il contient deux sous-couches :

- un Multi-Head Attention.
- un réseau Feed Forward.

Il existe également des connexions résiduelles autour de chacune des deux sous-couches suivies d'une normalisation de couche pour accélérer l'apprentissage des réseaux de neurones.

Nous pouvons en plus empiler l'encodeur N fois pour coder davantage les informations, où chaque couche a la possibilité d'apprendre différentes représentations de l'attention, augmentant ainsi potentiellement la puissance prédictive du réseau de transformers [36].

- **Input Embeddings et Positional Encoding (connexion résiduelle)**

La première étape, Input Embeddings, consiste à envoyer l'entrée dans une couche embedding layer chaque mot est représenté par un vecteur avec des valeurs continues.

L'étape suivante, Positional Encoding, consiste à injecter des informations de position dans les embeddings. Les Embeddings représentent un jeton de dimension d, où les jetons ayant une signification similaire seront plus proches les uns des autres. Mais les Embeddings n'encodent pas la position relative des jetons dans une phrase. Ainsi, après avoir ajouté le Positional Encoding, les jetons seront plus proches les uns des autres en fonction de la similitude de leur signification et de leur position dans la phrase [36].

Une façon de faire est d'utiliser les fonctions sinus et cosinus. Pour chaque indice pair (respectivement impair) sur le vecteur d'entrée, nous créons un vecteur à l'aide de la fonction sinus (respectivement cosinus). Nous additionnons ensuite ces vecteurs à leurs Input Embeddings correspondants. La formule de calcul du Positional Encoding est la suivante :

$$PE_{(pos,2i)} = sin\ (pos/10000^{\frac{2i}{d_{model}}}),$$

$$PE_{(pos,2i+1)} = cos\ (pos/10000^{\frac{2i}{d_{model}}}),$$

- **Multi-Head Attention**

Dans l'encodeur, le Multi-headed Attention utilise un mécanisme de Self-Attention. Pour le mettre en œuvre, les matrices Requête, Clé et Valeur sont divisées en h matrices de taille réduite. En appliquant le Self-Attention à ces matrices découpées, nous obtenons des résultats plus précis. Par la suite, ces matrices sont regroupées en une seule matrice avant d'être soumises à une couche linéaire finale. Cette dernière étape implique également une multiplication avec une matrice de poids supplémentaire [36], ce qui contribue à améliorer la représentation des données.

$$MultiHead(Q, K,V) = Concat\ (head_1, \dots, head_h)W^0$$

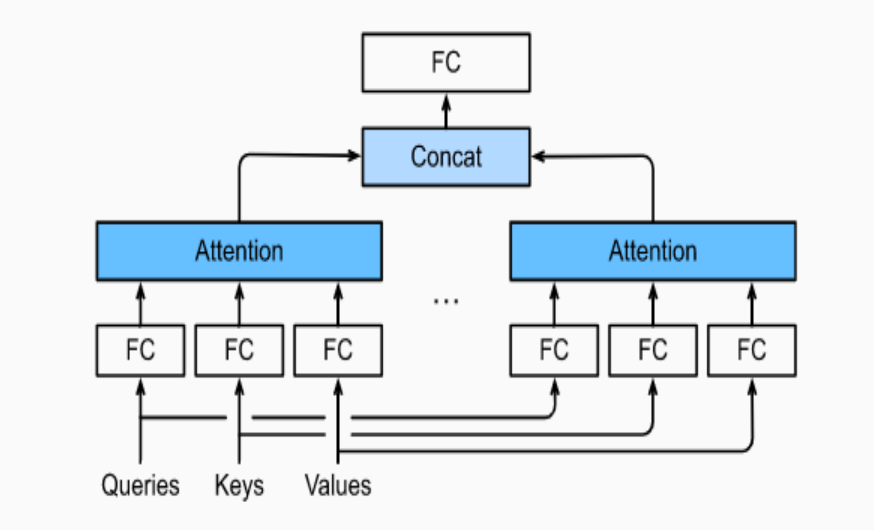

Figure II-9 Multi-Head Attention [39].

La sortie de la Multi-Head Attention est ajoutée à l'incorporation positionnelle d'entrée (connexion résiduelle), puis elle traverse une couche de normalisation pour une amélioration de la formulation du texte [36].

- **Réseau Feed Forward**

Le réseau Feed Forward est composé de deux couches linéaires séparées par une fonction d'activation ReLU, ce qui permet d'améliorer sa performance.

$$FFN(x) = max\ (0, xW1 + b1)\ W2 + b2.$$

Après cela, la sortie est additionnée à nouveau à l'entrée du réseau et normalisée. Les connexions résiduelles sont utilisées pour éviter le problème des gradients qui disparaissent dans les réseaux profonds. Les couches de normalisation sont utilisées pour stabiliser le réseau, ce qui réduit considérablement le temps nécessaire à l'apprentissage. Ensuite, le réseau Feed Forward est utilisé pour projeter les sorties d'attention et leur donner potentiellement une représentation plus riche.

Ensuite, la sortie de l'encodeur est transformée en un ensemble de vecteurs d'attention K et V. Chaque décodeur utilise ces vecteurs dans sa couche d'attention Encodeur-Décodeur, ce qui aide le décodeur à se concentrer sur les parties appropriées de la séquence d'entrée (voir le paragraphe suivant) [36].

II.5.4.2.2 Couche du décodeur

Le décodeur se compose de trois sous-couches principales pour améliorer le processus de décodage :

- Multi-Head Attention masqué.
- Multi-Head Encoder-Decoder Attention.
- Réseau Feed Forward.

Chacune de ces sous-couches est suivie d'une connexion résiduelle, qui permet aux informations de contourner directement les sous-couches et d'être préservées, et d'une étape de normalisation, qui stabilise les activations et facilite l'apprentissage.

Le décodeur fonctionne de manière autorégressive, ce qui signifie qu'il génère la sortie mot par mot en se basant sur les sorties précédentes. Il démarre en utilisant un jeton de démarrage et prend en entrée une liste des mots générés précédemment, ainsi que les informations d'Attention fournies par le codeur. Le décodage s'arrête lorsque le décodeur génère le jeton de fin, indiquant ainsi la fin de la génération du texte.

Pour obtenir les probabilités associées à chaque mot généré, le décodeur utilise une couche linéaire qui agit comme un classificateur, suivi d'une fonction softmax, qui normalise les scores et attribue des probabilités à chaque mot possible. Cela permet de sélectionner le mot le plus probable à chaque étape du décodage [36].

- **Classificateur linéaire et softmax final pour les probabilités de sortie**

Le processus de sortie de la couche Feed Forward est amélioré en utilisant une couche linéaire qui agit comme un classificateur. La taille de ce classificateur correspond au nombre de classes que nous avons. Ensuite, une couche softmax est appliquée pour générer des scores de probabilité. Le score de probabilité le plus élevé est sélectionné en tant que mot prédit.

Dans le décodeur, la sortie est ajoutée à la liste des entrées et le processus de décodage se poursuit jusqu'à ce qu'un jeton soit prédit. Dans notre cas, la classe finale correspondant au score de probabilité le plus élevé est attribuée au jeton de fin [36].

Les Transformers offrent une vaste sélection de modèles pré-entraînés capables de réaliser différentes tâches sur des textes, telles que la classification, l'extraction d'informations, la réponse aux questions, le résumé d'informations, la traduction et la génération de texte, et ce, dans plus de 100 langues. Parmi ces modèles, les plus populaires sont BERT (Devlin et al., 2018) et GPT v1-3 (Radford et al., 2018) [36]. Dans la suite, nous allons nous concentrer spécifiquement sur BERT.

II.6 BERT

II.6.1 Introduction

BERT, qui signifie « Bidirectional Encoder Representations from Transformers », est un modèle avancé d'intégration de mots basé sur l'architecture codée du transformateur [40]. Il a été introduit dans le papier « Attention is all you need ». L'un des avantages de cette architecture est sa capacité à traiter les relations entre des mots éloignés de manière plus efficace que les réseaux récurrents (RNN/LSTM) [41]. Nous utilisons BERT comme encodeur de phrase, car il peut fournir avec précision la représentation contextuelle d'une phrase.

BERT est une technologie développée par Google AI en 2018, qui se distingue de ses concurrents tels qu'OpenAI GPT et ELMo de plusieurs manières. Il est plus performant en termes de résultats et de vitesse d'apprentissage. De plus, il peut apprendre de manière non supervisée en utilisant un vaste corpus de données, comme l'ensemble de la version anglophone de Wikipedia, afin d'obtenir une représentation linguistique unique. Cette représentation peut ensuite être adaptée à une tâche spécifique en effectuant un apprentissage supervisé rapide avec peu de données.

En résumé, BERT est un modèle de traitement du langage naturel (NLP) qui a suscité un grand intérêt depuis sa sortie en raison de ses performances supérieures par rapport aux modèles de NLP précédents. Il est utilisé pour des tâches spécifiques telles que la classification de texte, l'extraction de relations et la génération de texte, ce qui en fait un outil polyvalent pour différentes applications de NLP [40].

Une des raisons pour lesquelles BERT est plus performant est sa capacité à prendre en compte le contexte complet d'une phrase ou d'un texte, à la fois en avant et en arrière, c'est-à-dire de manière bidirectionnelle. Contrairement à OpenAI GPT, qui ne regarde que vers l'arrière, ou ELMo, qui combine des vues arrière et avant entraînées de manière indépendante. Cela signifie que BERT a une compréhension plus approfondie du langage et peut donc mieux saisir le sens des phrases et des textes (voir la figure II .10) [40].

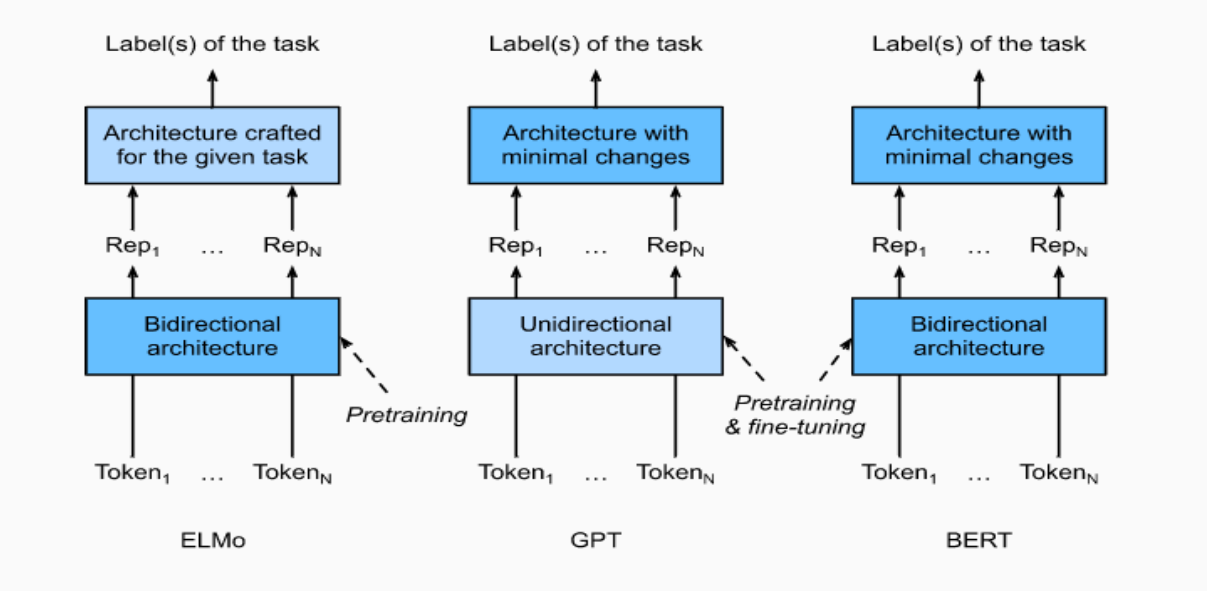

Figure II-10 Module du BERT VS GPT, ELMO [42].

II.6.2 Les variantes de BERT

Il existe deux variantes de BERT :

- **BERT Base**

Il s'agit d'une version plus petite, qui est plus abordable en termes de calcul mais ne convient pas aux opérations complexes d'extraction de texte [43].

Nom du paramètre	nombre du paramètre
Nombre des couches	12
Taille cachée	768
Têtes d'attention	12
Nombre des paramètres	110M

Tableau II-1 Nombre des paramètres for BERT-Base [43].

- **BERT Large**

Il s'agit d'une version plus grande et plus coûteuse en termes de calcul, mais qui permet d'analyser de grandes quantités de données textuelles pour fournir les meilleurs résultats [43].

Nom du paramètre	nombre du paramètre
Nombre des couches	24
Taille cachée	1024
Têtes d'attention	16
Nombre des paramètres	340M

Tableau II-2 Nombre des paramètres for BERT-Large [43].

II.6.3 L'architecture de BERT

Afin de faciliter les explications, nous utiliserons le modèle BERT-Base, Uncased, qui présente les caractéristiques suivantes : l'utilisation exclusive de minuscules, une seule langue, 12 couches, une dimension de couche cachée de 768 et 12 têtes d'attention. Nous allons maintenant examiner la composition de ces couches basées sur une architecture Transformer [41]. Voici l'ordre des sous-couches :

- Le block d'attention
- Le layer de normalisation
- Le layer Feed forward
- Le layer de normalisation

II.6.3.1 Embedding

La première étape du processus consiste à effectuer l'encastrement, qui permet de convertir nos mots en vecteurs. BERT combine trois (03) types d'encastrements pour former son entrée : l'encastrement du jeton (token embedding), l'encastrement de position (positional embedding) et l'encastrement de segment (segment embedding) [41]. Cette combinaison crée une représentation complète des mots dans le modèle BERT.

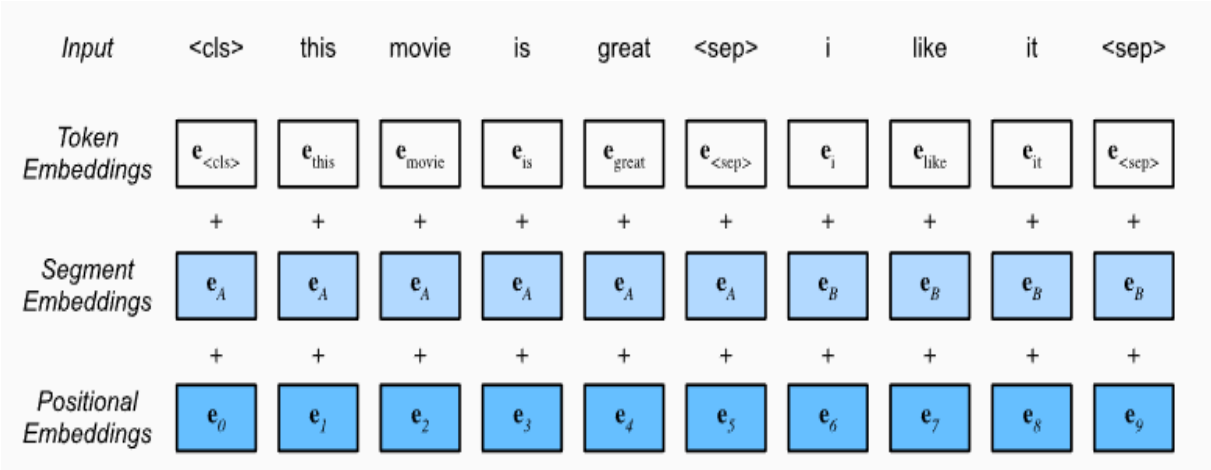

Figure II-11 L'architecture d'incorporation de mots dans BERT [42].

II.6.3.1.1 Token Embedding

Le token embedding est utilisé pour obtenir des informations sur le contenu d'un texte. La première étape consiste à convertir le texte en un vecteur. Afin de gérer un vocabulaire volumineux et d'incorporer de nouveaux mots, un système de token est utilisé. Un token peut être une ou plusieurs lettres, nous allons voir quelques exemples par la suite. L'algorithme BPE (encodage par paires de bytes) peut être utilisé pour identifier les tokens qui décomposent les mots en entrée. Bien que BERT utilise un autre algorithme, le principe est similaire.

Pour commencer, nous recherchons les paires de lettres les plus fréquentes dans notre ensemble de mots. Par exemple, supposons que la paire la plus fréquente soit « er ». Nous remplaçons cette paire par un nouveau token appelé « Z ».

Ensuite, nous substituons toutes les occurrences de « er » par Z. Ce processus est répété jusqu'à atteindre le nombre de tokens souhaité (30 000 tokens pour BERT). À mesure que le nombre de tokens augmente, nous obtenons des tokens qui ressemblent davantage à des mots ou sont des mots. De plus, les lettres individuelles sont toujours ajoutées en tant que tokens, ce qui nous permet de traiter des mots jamais rencontrés.

Une fois que nous avons notre liste de tokens, nous pouvons transformer de manière déterministe notre texte en tokens et vice versa. Pour cela, nous remplaçons les occurrences des différents tokens du plus grand au plus petit. Dans le pire des cas, un mot est représenté par

chacune de ses lettres. Chaque token peut maintenant être encodé dans un vecteur de taille 30 000, avec des zéros partout sauf à un emplacement où nous mettons le chiffre un (encodage one-hot) [41].

• Petite digression :

Par la suite, nous allons souvent projeter des vecteurs d'une taille N vers une taille M. Concrètement, cela signifie que nous prenons notre vecteur V (de taille N), le multiplions par une matrice de poids W (de taille N*M, c'est-à-dire N lignes et M colonnes), ce qui nous donne notre vecteur de sortie U (de taille M).

U = V * W avec * la multiplication matricielle

Les matrices de poids sont les paramètres de notre réseau de neurones qui s'apprennent par descente de gradient [41].

• Suite du token embedding :

Nous pouvons maintenant projeter nos vecteurs (ceux qui représentent les tokens) de taille 30 000 vers un espace plus réduit (pour BERT de dimension 768). Ainsi, chaque token est représenté par un vecteur de taille 768. Ceci est beaucoup plus raisonnable que nos vecteurs de taille 30 000 [41].

II.6.3.1.2 Segment embedding:

Le segment embedding permet de définir l'appartenance à une même phrase. Par exemple, si nous avons la phrase 1 qui utilise 15 tokens, alors notre segment embedding sera composé de 15 fois 0, suivi du reste du vecteur jusqu'à atteindre la longueur maximale de la séquence, qui est de 512. Les valeurs seront alors des 1 pour indiquer qu'il s'agit d'une autre phrase. Cela permet à BERT de prendre en entrée une ou deux phrases [41].

II.6.3.1.3 Positional Embedding:

Le « Positional Embedding » permet aux réseaux de neurones de connaître la position des mots dans la séquence. Sans cette partie, le réseau ne peut pas comprendre l'ordre de la séquence, ce qui entraînerait la même représentation pour deux mots identiques situés à des endroits différents dans la phrase [41].

II.6.4 Le mécanisme d'attention dans BERT

Dans la suite, nous allons examiner en détail le bloc d'attention. Pour commencer, nous étudierons un mécanisme d'attention avec une seule tête, car cela nous permettra de comprendre facilement comment cela se généralise. Le succès de BERT repose principalement sur l'utilisation combinée du mécanisme d'attention et de l'entraînement préalable sur de grandes quantités de données textuelles non étiquetées.

Revenons à nos vecteurs qui représentent chaque jeton avec une taille de 768. Nous allons maintenant examiner comment un jeton unique est « transformé » par le mécanisme d'attention.

Étape 1 : Pour chaque jeton, nous calculons trois vecteurs associés : la clé, la requête et la valeur. Ce calcul peut être considéré comme une projection vers trois espaces, chacun ayant une taille de 768.

Soit V le vecteur de taille 768 représentant un jeton, et Wc, Wr, Wv trois matrices de poids de taille 768*768.

Clé = V*Wc

Requête = V*Wr

Valeur = V*Wv

Étape 2 : Ensuite, nous effectuons la multiplication entre le vecteur requête du jeton actuel et les vecteurs clés transposés des autres jetons. Cette opération génère un vecteur de taille 512, qui représente la mesure de corrélation entre le jeton actuel et tous les autres jetons.

Étape 3 : Par la suite, nous appliquons une transformation à ce vecteur afin qu'il soit normalisé et que la somme de ses éléments soit égale à un. Pour cela, nous utilisons la fonction softmax. Cette transformation nous permet d'interpréter le vecteur résultant comme une mesure de l'importance relative de chaque autre jeton, en termes d'attention à accorder lors du traitement du jeton actuel.

Fonction softmax :

$$\sigma(\mathbf{Z})_{\mathbf{J}} = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ pour tout } j \in \{1, \dots, K\}$$

Étape 4 : Nous procédons à la multiplication de cette proportion par les vecteurs de valeurs de chaque token, ce qui nous conduit finalement à obtenir une nouvelle représentation du token.

Comme mentionné précédemment, BERT comporte 12 têtes d'attention, tandis que nous avons présenté ici seulement une seule tête. Pour parvenir à cela, nous répétons simplement la même opération 12 fois en utilisant des matrices de poids différentes. Ensuite, nous concaténons tous les résultats et les projetons dans la dimension souhaitée, qui est ici de 768[41].

II.6.5 Autres variantes de BERT

Variantes BERT	Formulaire complet	Résume
ALBERT	(A Lite BERT)	Le modèle ALBERT dispose de 12 millions de paramètres, comprenant 768 couches cachées et 128 couches intégrées. Comme prévu, cette version plus légère du modèle a permis de réduire à la fois le temps nécessaire pour l'entraîner et le temps nécessaire pour effectuer des inférences.
RoBERTa	(Robustly Optimized BERT)	RoBERTa est une version améliorée de BERT, basée sur la stratégie de masquage MLM, où le modèle apprend à prédire des sections de texte intentionnellement masquées dans des exemples non annotés. Au niveau technique, RoBERTa modifie certains hyper-paramètres du pré-entraînement de BERT.
CamemBERT	CamemBERT	CamemBERT est un modèle linguistique de pointe pour la langue française, basé sur l'architecture RoBERTa et pré-entraîné sur le corpus multilingue OSCAR. Il améliore les performances dans de nombreuses tâches de traitement du langage naturel en français.
FlauBERT	FlauBERT	FlauBERT est un modèle BERT français développé peu de temps après la sortie de CamemBERT. Il a été entraîné sur un vaste corpus de texte français varié. FlauBERT propose différentes tailles de modèles pré-entraînés, et l'ensemble des données, des codes sources et des modèles sont disponibles au public. Les performances de FlauBERT et de CamemBERT sont très similaires.

Tableau II-3 Autres variantes de BERT [44].

II.6.6 Description des taches de MLM &NSP

Bert est un modèle pré-entraîné sur deux tâches distinctes mais interconnectées dans le domaine du traitement automatique du langage naturel (TALN) : la modélisation du langage masqué (MLM) et la prédiction de la phrase suivante (NSP).

Le MLM est un processus utilisé par Bert pour anticiper les mots masqués au sein d'une phrase. Avant d'être soumis à Bert, 15 % des mots de chaque séquence sont remplacés par le symbole ***[MASK]***. L'objectif de Bert est de prédire les mots d'origine qui ont été masqués en s'appuyant sur le contexte fourni par les autres mots non masqués de la séquence.

La fonction de perte de Bert se concentre exclusivement sur la prédiction des mots masqués, sans prendre en compte la prédiction des mots non masqués.

En ce qui concerne le NSP, il fonctionne en prédisant si deux phrases sont logiquement (ou séquentiellement) liées ou non. Par exemple, prenons les phrases suivantes : « Pierre est malade. » et « Il a la grippe. » Dans la seconde phrase, le pronom « il » fait référence à Pierre. Par conséquent, il existe une connexion logique ou causale entre ces deux phrases [45].

La Figure II.12 illustre les deux tâches NSP et MLM :

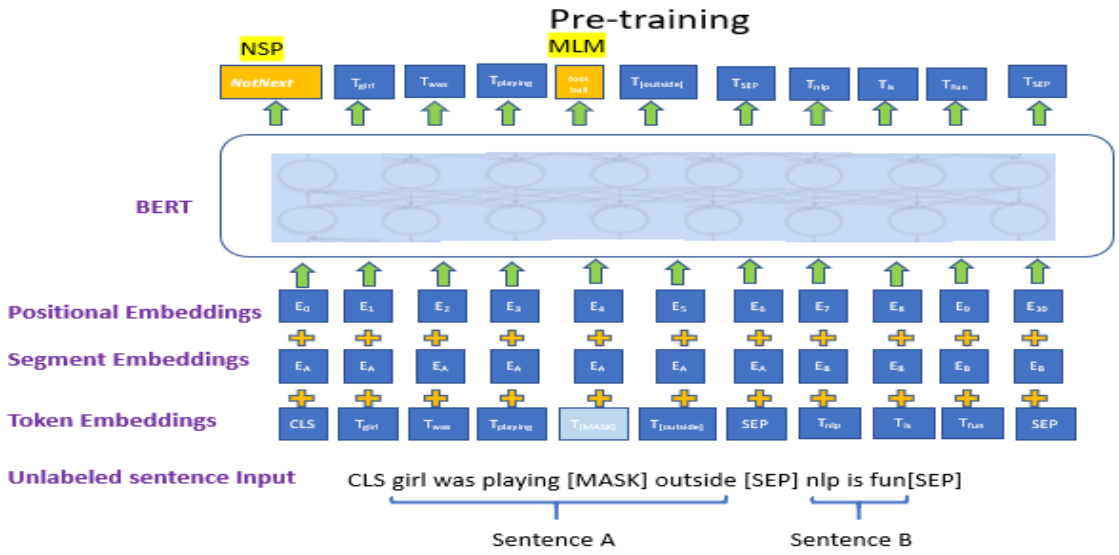

Figure II-12 La description des taches de NSP et MLM [48].

II.6.7 Prédiction de la phrase suivante (NSP)

La tâche NSP (Next Sentence Prediction) est une classification binaire où deux phrases sont envoyées à BERT, et il doit prédire si la deuxième phrase est la continuation logique de la première ou non. En accomplissant cette tâche NSP, notre modèle est capable de saisir la relation entre ces deux phrases.

La compréhension de la relation entre ces deux phrases s'avère précieuse pour des tâches ultérieures telles que la génération de réponses à des questions ou de texte.

II.6.7.1 Le fonctionnement du BERT-NSP

Pour effectuer la classification, nous utilisons la représentation du jeton ***[CLS]*** que nous alimentons dans un réseau feedforward avec la fonction Softmax. Cela nous donne la probabilité que la paire de phrases soit « is Next » (consécutive) ou « not Next » (non consécutive). L'intégration de ***[CLS]*** contient essentiellement une représentation agrégée de tous les jetons [47].

Voici comment vous pouvez utiliser BERT pour prédire la consécution des phrases et calculer les logits en détail :

- **Prétraitement des phrases :**

Vous devez formater vos données d'entrée selon le format spécifique utilisé par BERT. Cela implique l'ajout de marqueurs spéciaux pour indiquer le début et la fin des phrases, ainsi que des marqueurs spéciaux pour séparer les phrases. Par exemple, vous pouvez ajouter les marqueurs ***[CLS]*** et ***[SEP]*** pour indiquer le début et la fin de chaque phrase, et ajouter un marqueur ***[SEP]*** pour séparer les deux phrases [48]. Supposons que nous ayons les deux phrases suivantes :

- Phrase A : « Le chat est noir. »

- Phrase B : « Il dort paisiblement. »

Après le prétraitement, les phrases peuvent être représentées comme suit :

« ***[CLS]*** Le chat est noir. ***[SEP]*** Il dort paisiblement. ***[SEP]*** ».

- **Alimentation du modèle :**

Passez les représentations vectorielles des phrases à travers les couches du modèle. BERT utilise une architecture à deux têtes : une tête d'encodage pour représenter les phrases et une tête de classification pour la prédiction de séquence [48].

- **Calcul des logits :**

Pour prédire la consécution des phrases, utilisez la sortie de la tête de classification et appliquez une couche dense avec deux neurones (correspondant aux classes « consécutive » et « non consécutive ») sur la dernière couche de sortie de BERT. Les logits correspondent aux sorties de cette couche dense avant l'application d'une fonction d'activation [48].

- **Prédiction :**

La couche de classification binaire applique une fonction d'activation appropriée (par exemple, softmax ou sigmoid) pour obtenir des probabilités normalisées indiquant si deux phrases sont consécutives. Vous pouvez définir un seuil pour interpréter la sortie comme une

prédiction binaire (par exemple, si la probabilité dépasse 0,5, les phrases sont considérées comme consécutives).

Il est important de noter que l'entraînement de BERT pour la tâche de prédiction de séquence ("NSP" - Next Sentence Prediction) nécessite des données étiquetées où vous indiquez si les paires de phrases sont consécutives ou non. Ces données sont utilisées pour entraîner BERT à prédire correctement la consécution des phrases [48].

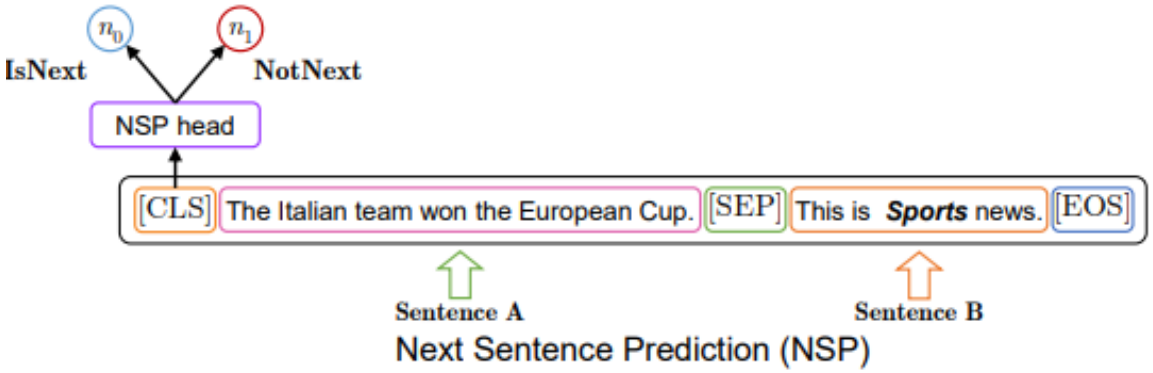

Figure II-13 The Next sentence preédiction (NSP) task [48].

II.7 Conclusion

Dans ce chapitre, nous avons exploré le domaine du traitement automatique du langage naturel (TALN) et examiné les trois principales approches du NLP. Notre attention s'est principalement portée sur les approches du Deep Learning. Ensuite, nous avons détaillé le concept d'incorporation des mots et présenté divers modèles utilisés dans ce domaine. Par la suite, nous avons introduit les techniques de Deep Learning appliquées au TALN, en mettant en évidence leurs avantages et leurs inconvénients. Nous avons également fourni une explication détaillée de l'architecture des Transformers, en accordant une attention particulière à BERT, l'un des modèles les plus couramment utilisés. Nous avons abordé les différents types d'entraînement de BERT, tels que la prédiction de la phrase suivante (NSP) et la modélisation de la langue masquée (MLM).

Chapitre III

Approche DL pour l'initialisation du RS de collaboration des SW

Sommaire

III.1 Introduction 43
III.2 Aperçu sur les réseaux sociaux 43
III.2.1 Représentation Matricielle d'un réseau social 44
III.3 Les services Web sociaux 44
III.3.1 Les services web dans les réseaux sociaux 45
III.3.2 Approche pour élaborer un réseau sociaux des services web 45
III.3.3 Les propriétés sociales de services Web 48
III.3.4 Systèmes de recommandation et plates-formes sociales 48
III.4 Les travaux connexes 49
III.4.1 Discovering web services in social web service repositories using deep variational ... 50
III.4.2 Constructing a global social service network for better quality of web service 50
III.4.3 Mining Social Web Service Repositories for Social Relationships to Aid Service 50
III.4.4 Social-Based Web Services Discovery and Composition for Step-by-Step Mashup ... 51
III.4.5 Collaboration reputation for trustworthy Web service selection in social networks. 51
III.5 Notre approche pour l'initialisation du RS de collaboration de SW 51
III.5.1 Extraction des données 53
III.5.2 Traitement des données 53
III.5.3 Jeu de données d'entraînement Avec label pour la tâche NSP 53
III.5.4 Entraînement du modèle 53
III.5.5 Évaluation du modèle 53
III.5.6 Collecte des prédictions appliquées sur les descriptions des SW 54
III.5.7 Génération des matrices d'adjacences pour le réseau de collaboration 54
III.5.8 Visualisation de matrice d'adjacence 54
III.6 Discussion des résultats 54
III.7 Tests et analyse du comportement de notre modèle 60
III.8 Conclusion 62

III.1 Introduction

De nos jours, la recherche dans le domaine des services Web revêt une importance capitale. Cette importance croissante a donné lieu à de nouvelles méthodes de découverte ainsi qu'à d'autres formes d'interactions, telles que la composition de services et la substitution en cas de pannes. Dans ce travail, notre attention se concentre principalement sur l'une des méthodes d'apprentissage en profondeur connue sous le nom de BERT-NSP, utilisée pour l'initialisation du réseau social de collaboration des services Web. Ce choix est justifié par le fait que ce modèle est le plus approprié dans notre contexte, et pour la tâche que nous nous efforçons d'accomplir, à savoir la découverte des « séquences de services » d'une certaine manière.

L'initialisation du réseau social de collaboration des services Web est un processus essentiel qui consiste à établir des liens entre les différents services Web disponibles. Cela permet aux services de collaborer et d'échanger des informations de manière efficace. L'utilisation de BERT-NSP pour accomplir cette tâche permet de prédire la compatibilité et la relation entre les services Web en analysant leurs descriptions. Cette approche améliore considérablement la manière dont les liens de collaboration des services Web sont établis.

III.2 Aperçu sur les réseaux sociaux

Les réseaux sociaux ont trouvé leur utilisation dans divers domaines tels que les sciences sociales et politiques, l'intelligence artificielle (IA) et l'e-business. Selon [49], l'étude des réseaux sociaux revêt une importance primordiale, car elle permet de mieux comprendre comment et pourquoi nous interagissons, ainsi que la manière dont la technologie peut modifier ces interactions. Au cours des dernières années, le domaine de la théorie des réseaux sociaux a connu une croissance significative, grâce aux avancées de la technologie informatique qui ont ouvert la voie à de nouvelles recherches.

De manière générale, un réseau se compose de nœuds et de liens. Les nœuds représentent différents types d'objets ou d'entités, tels que des individus ou des organisations, tandis que les liens désignent les relations ou les associations entre ces nœuds, comme le degré d'amitié entre deux personnes ou la distance entre deux villes. Les relations peuvent être unidirectionnelles, bidirectionnelles, pondérées ou une combinaison de ces éléments. Les recherches dans différents domaines académiques ont révélé que les réseaux sociaux opèrent à différents niveaux, allant des familles aux nations, et jouent un rôle essentiel dans la résolution de problèmes, la gestion des organisations et le succès individuel dans l'atteinte des objectifs [18].

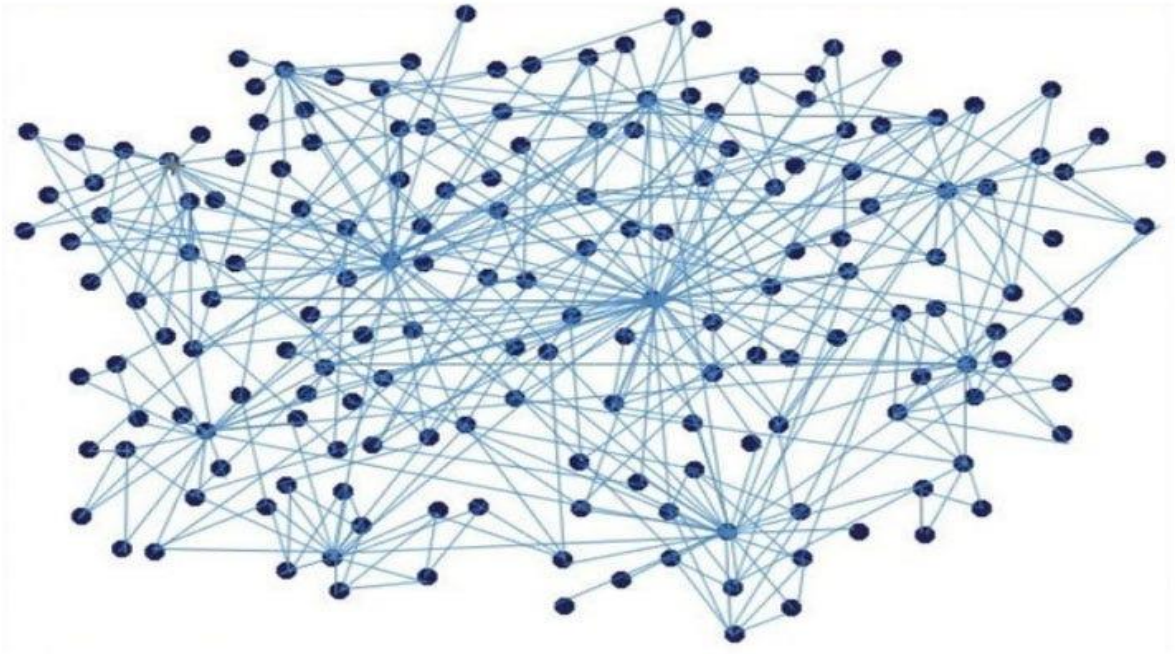

Figure III-1 Exemple d'un réseau social [51].

III.2.1 Représentation Matricielle d'un réseau social

Il existe deux représentations canoniques des réseaux sociaux : les diagrammes nœud-lien et les matrices d'adjacence. La matrice d'adjacence représente chaque nœud du réseau à la fois en tant que ligne et colonne. Lorsque deux nœuds sont connectés, la case correspondant à l'intersection de la ligne et de la colonne est marquée. Traditionnellement, une valeur numérique est utilisée (0 pour indiquer l'absence de connexion, 1 pour indiquer la présence).

La représentation matricielle présente plusieurs avantages par rapport aux diagrammes nœud-lien, le plus important étant l'élimination des superpositions de nœuds et des croisements de liens. Cela est particulièrement utile pour les réseaux de forte densité. Ainsi, la représentation matricielle est importante pour l'analyse des réseaux sociaux et la communication des résultats d'analyse sur ces réseaux [52].

III.3 Les services Web sociaux

Les services Web sociaux sont nés de la fusion des services Web et des réseaux sociaux, en intégrant les interactions des services Web avec des caractéristiques sociales.

Dans le contexte d'un réseau social de services Web (SNS), les services Web interagissent constamment, permettant la formation de nouvelles relations et la modification ou la disparition de relations existantes. L'analyse des réseaux sociaux peut aider les services Web à tirer parti des scénarios de composition précédents, en établissant des relations avec les paires qui ont déjà participé à ces compositions. Le SNS peut être représenté sous la forme d'un graphe, où les nœuds représentent les services Web et les liens représentent les interactions entre les services Web [21].

III.3.1 Les services web dans les réseaux sociaux

Le modèle social se décompose en quatre étapes :

- **Relations liant les services Web**

L'objectif de cette étape consiste à identifier les relations entre les services Web, qui peuvent être de trois types différents :

Les services Web offrant des fonctionnalités similaires peuvent soit être en compétition les uns avec les autres — un seul service étant sélectionné à la fois — soit se substituer les uns aux autres.

Les services Web offrant des fonctionnalités différentes peuvent collaborer pour créer de nouveaux services composites [21].

- **Réseaux sociaux correspondants aux relations**

L'objectif de cette étape est d'identifier les réseaux potentiels qui peuvent mettre en relation les services Web. Chaque relation établie sert de base à la création d'un réseau [21].

- **Construction du réseau social de services Web**

L'objectif de cette étape consiste à définir les composants qui constitueront le réseau social, à savoir les nœuds qui représentent les services Web et les bords qui symbolisent les relations entre ces services [21].

- **Comportement des services Web**

L'objectif de cette étape consiste à identifier le comportement des services Web, qui peut se manifester selon différents types de comportements sociaux observables dans la vie réelle [21].

III.3.2 Approche pour élaborer un réseau sociaux des services web

Trois types de réseaux sociaux ont été identifiés : le réseau social de collaboration, le réseau social de substitution et le réseau social de compétition [53].

Le réseau social de compétition se compose de services Web offrant des fonctionnalités similaires, ce qui les place en concurrence les uns avec les autres. Ils sont connectés entre eux par des liens bidirectionnels, comme indiqué dans la figure III-2. Ces services Web se concurrencent en offrant des fonctionnalités similaires, mais ils se distinguent par leurs propriétés non fonctionnelles, qui doivent répondre aux exigences des utilisateurs. Ainsi, chaque service Web évalue son propre réseau de concurrents afin d'améliorer ses propriétés non fonctionnelles par rapport à ceux des autres [53].

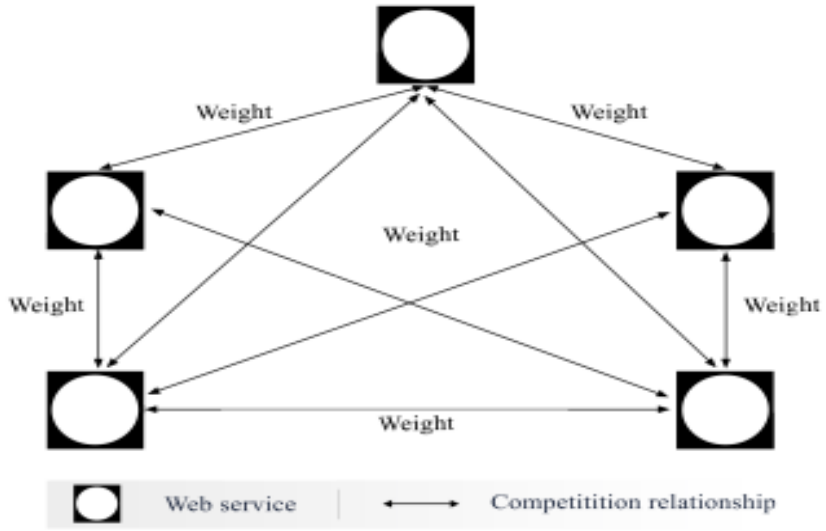

Figure III-2 Réseau social de compétition de services Web [54].

Le réseau social de substitution est similaire au réseau social de compétition, à une différence près : les liens ne sont pas tous bidirectionnels. Dans ce réseau, les services Web offrent les mêmes fonctionnalités, ce qui signifie que n'importe quel service Web peut être choisi comme candidat potentiel pour remplacer un service Web défaillant. Bien que ces services Web soient en concurrence les uns avec les autres, ils peuvent néanmoins s'entraider en cas d'échec, étant donné qu'ils offrent des fonctionnalités similaires. Ainsi, chaque service Web contrôle son propre réseau de substituants afin d'identifier les meilleurs substituants possibles en réponse aux exigences non fonctionnelles des utilisateurs [53].

Le réseau social de collaboration est établi lorsqu'au moins une composition a été réalisée. Pour naviguer dans ce réseau, un nœud d'entrée spécial appelé le service Web « focus » est requis et se distingue des autres nœuds (voir Figure III-3). Tous les liens sortant de ce service Web « focus » sont unidirectionnels et pointent vers d'autres services Web. En combinant leurs fonctionnalités respectives, les services Web sont capables de collaborer efficacement pour répondre aux demandes complexes des utilisateurs. Ainsi, chaque service Web contrôle son propre réseau de collaborateurs, ce qui lui permet de décider s'il souhaite collaborer avec certains pairs en fonction de leurs expériences passées. Il peut également recommander des pairs à d'autres services Web [53].

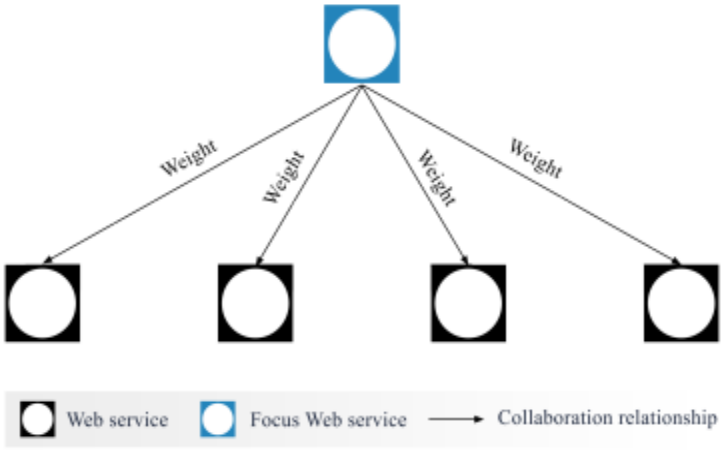

Figure III-3 Réseau social de collaboration de services Web [54].

En outre, deux autres types de réseaux ont été définis dans [54] en se basant sur les relations de supervision et de recommandation.

Le réseau social de supervision est construit à l'aide de deux types de nœuds : le service Web maître et le service Web esclave, reliés par des liens de supervision. Dans ce réseau, il y a un unique service Web maître et plusieurs services Web esclaves. Le point d'entrée du réseau social de supervision est le service Web maître, tandis que les autres services Web sont connectés à celui-ci via des liens unidirectionnels de supervision (voir Figure III. 4) [54].

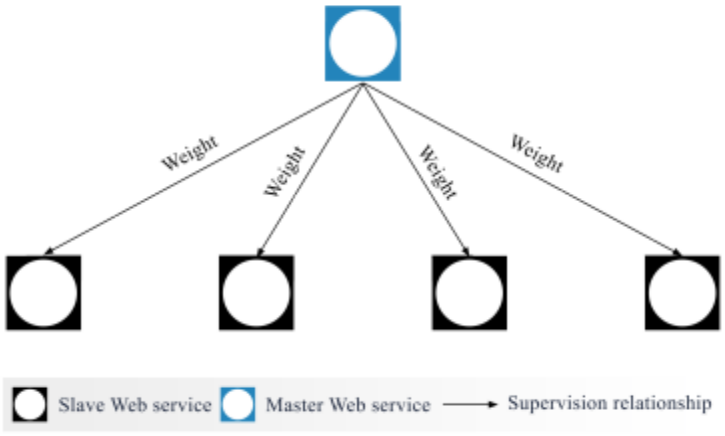

Figure III-4 Réseau social de supervision [54].

Le réseau social de recommandation est établi lorsque plusieurs compositions ont été réalisées dans le réseau social de collaboration. Ce réseau utilise un unique type de nœud

pour représenter les services Web esclaves, ainsi qu'un type de lien pour représenter la relation de recommandation. Il est utilisé pour identifier les paires de services Web avec lesquelles un service Web esclave souhaiterait collaborer, en se basant sur les compositions précédentes rapportées dans le réseau social de collaboration [54].

III.3.3 Les propriétés sociales de services Web

La section 3.3 traite des propriétés sociales des services Web qui ont été identifiées dans la littérature [21]. Les principales propriétés sociales comprennent :

- **La confiance**

Un service est considéré comme digne de confiance s'il peut effectuer ses tâches avec succès.

- **L'égoïsme**

Un service Web est considéré comme égoïste s'il refuse de collaborer avec d'autres services Web, mais accepte leurs demandes en retour.

- **L'imprévisibilité**

Un service est considéré comme imprévisible si ses réponses ne correspondent pas à celles attendues par le service Web.

- **La malveillance**

Un service Web est considéré comme malveillant s'il est impliqué dans un grand nombre de relations trompeuses.

- **La dominance**

Un service Web est considéré comme dominant s'il a la majorité des relations de collaboration en sa faveur.

Ces propriétés sociales sont déterminées à partir des réseaux sociaux des services Web [21].

III.3.4 Systèmes de recommandation et plates-formes sociales

De nos jours, l'accès généralisé à Internet à travers le monde permet à un grand nombre de personnes de se connecter en ligne. Cette expansion du Web 2.0 a entraîné une demande croissante de systèmes de recommandation utilisant des méthodes d'exploration des réseaux et des informations sociales. Ces systèmes exploitent les infrastructures sociales, également connues sous le nom de réseaux sociaux ou de communautés virtuelles [55].

La croissance exponentielle des réseaux sociaux présente de nouvelles opportunités et défis pour la recherche en matière de systèmes de recommandation. La raison principale réside dans le fait que les réseaux sociaux transforment les utilisateurs en contributeurs actifs,

leur permettant de partager leur statut, de commenter et d'évaluer du contenu en ligne. Ainsi, trouver du contenu connexe et intéressant au bon moment et dans le bon contexte nécessite des approches de recommandation innovantes. Parallèlement, la valeur ajoutée principale des plateformes sociales réside dans leur capacité à encourager les interactions entre utilisateurs. Chaque interaction peut être extraite et utilisée pour alimenter le système de recommandation, car cela permet de mieux comprendre les intérêts des utilisateurs et leurs besoins en matière d'information. De plus, l'architecture des réseaux sociaux peut contribuer à générer des recommandations plus fiables (par exemple, en tenant compte de la proximité sociale lors du processus de recommandation, nous avons tendance à accorder plus de confiance aux recommandations provenant de liens étroits). Ainsi, il est clair que les réseaux sociaux offrent une excellente opportunité d'améliorer les systèmes de recommandation [55].

D'un autre côté, les systèmes de recommandation peuvent clairement contribuer à accroître la participation des utilisateurs sur les plateformes sociales, en recommandant de nouveaux amis, des services ou du contenu intéressant. Par conséquent, les utilisateurs seront plus motivés à continuer de participer à la plateforme sociale, car plus ils partagent de contenu, plus le système peut recommander des connexions pertinentes avec un profil précis les concernant [55].

Figure III-5 Apport réciproque entre le système de recommandation et le réseau social.

III.4 Les travaux connexes

Dans cette section, nous allons présenter les travaux connexes qui font partie du même domaine sur lesquels nous allons travailler.

III.4.1 Discovering web services in social web service repositories using deep variational auto-encoders :

Dans l'article [56], les auteurs présentent une méthode améliorée pour apprendre des caractéristiques à partir de descriptions de services en utilisant des Auto-encodeurs Variationnels, une variante spéciale d'auto-encodeur qui restreint la représentation encodée afin de modéliser des variables latentes. Les auto-encodeurs sont des réseaux neuronaux profonds utilisés pour apprendre de manière non supervisée des codages efficaces. Pour entraîner leur auto-encodeur, l'équipe a utilisé un ensemble de données réel composé de 17113 services extraits du dépôt social d'API de ProgrammableWeb.com. L'efficacité de cette approche a été évaluée en utilisant des métriques de rappel et de précision, démontrant des améliorations significatives par rapport aux techniques classiques telles que les Word Embeddings et la modélisation traditionnelle des fonctionnalités latentes.

III.4.2 Constructing a global social service network for better quality of web service discovery :

Dans l'article [57], les auteurs présentent une proposition visant à relier les îlots de services isolés pour former un réseau social mondial de services, dans le but d'améliorer la sociabilité des services à l'échelle mondiale. Pour atteindre cet objectif, les auteurs suggèrent tout d'abord d'adopter des principes spécifiques aux services sociaux liés, inspirés des principes des données liées, pour la publication des services sur le Web ouvert en tant que services sociaux liés. Ensuite, un nouveau cadre est proposé pour construire le réseau social mondial de services en suivant ces principes spécifiques aux services sociaux liés, en s'appuyant sur les théories des réseaux complexes. Enfin, les auteurs présentent une approche permettant d'exploiter ce réseau social mondial de services en fournissant les services sociaux liés en tant que service.

III.4.3 Mining Social Web Service Repositories for Social Relationships to Aid Service Discovery :

Dans l'article [58], les auteurs proposent une approche novatrice pour la découverte de services web sémantiques (SWS). Leur méthode repose sur l'analyse de graphes contenant des relations entre utilisateurs et services, en utilisant des mesures topologiques légères pour évaluer la similarité des services. Ils identifient ensuite les services « socialement » similaires en exploitant à la fois les relations explicites et implicites présentes dans le graphe. Pour faciliter la découverte, ces services sont regroupés à l'aide d'un algorithme de regroupement basé sur des exemplaires.

Les résultats des expériences menées sur le registre ProgrammableWeb.com, qui est actuellement la plus vaste collection de SWS avec plus de 15 000 services et 140000 relations utilisateur-service, démontrent que le regroupement basé uniquement sur la topologie peut constituer un complément prometteur aux approches basées sur le contenu. Contrairement à

ces dernières, qui nécessitent des opérations de traitement de texte plus longues, l'approche topologique se révèle plus rapide et efficace.

III.4.4 Social-Based Web Services Discovery and Composition for Step-by-Step Mashup Completion

Dans l'article [59], les auteurs se sont penchés sur la recommandation de services Web pour la composition de services dans un environnement de Mashup. Ils ont présenté une approche novatrice visant à aider les utilisateurs finaux en capturant et en analysant les interactions sociales. Cette approche repose sur un graphe social implicite, déduit des intérêts communs des utilisateurs en matière de composition. Le processus de transformation des interactions entre les utilisateurs et les services en un graphe social est décrit, ainsi qu'une méthode potentielle pour utiliser ce graphe afin de générer des recommandations de services. Pour tester cette proposition, les auteurs l'ont mise en œuvre au sein d'une plateforme appelée SoCo, où des expériences préliminaires ont révélé des résultats prometteurs.

III.4.5 Collaboration reputation for trustworthy Web service selection in social networks

Dans l'article [60], les auteurs introduisent un nouveau concept prometteur appelé « réputation de collaboration ». Ce concept repose sur un réseau de services Web collaboratifs, qui se compose de deux mesures distinctes. La première mesure, appelée « réputation d'invocation », permet de calculer la réputation d'un service en se basant sur les recommandations fournies par d'autres services. La seconde mesure, appelée « réputation invoquée », évalue la fiabilité d'un service en fonction de la fréquence de ses interactions avec d'autres services Web.

En utilisant la réputation de collaboration comme critère, une méthode est présentée pour sélectionner des services Web fiables. Cette méthode offre une solution non seulement aux problèmes de sélection de services Web simples, mais aussi aux problèmes de sélection plus complexes. Elle permet ainsi d'améliorer la qualité globale des services sélectionnés en se basant sur des critères objectifs de réputation et de collaboration entre les services Web.

III.5 Notre approche pour l'initialisation du RS de collaboration de SW

Dans notre étude, nous nous sommes penchés sur l'approche de Maamar.Z afin d'améliorer le processus de découverte des services web en utilisant un réseau social de collaboration. Notre approche consiste à identifier les services web qui peuvent être utilisés consécutivement ou indépendamment. Pour ce faire, nous proposons un modèle d'apprentissage profond appelé « BERT-NSP » qui permet de prédire quels sont les services web pouvant fonctionner ensemble. Les étapes de notre approche sont expliquées en détail dans la Figure III.6 ci-dessous.

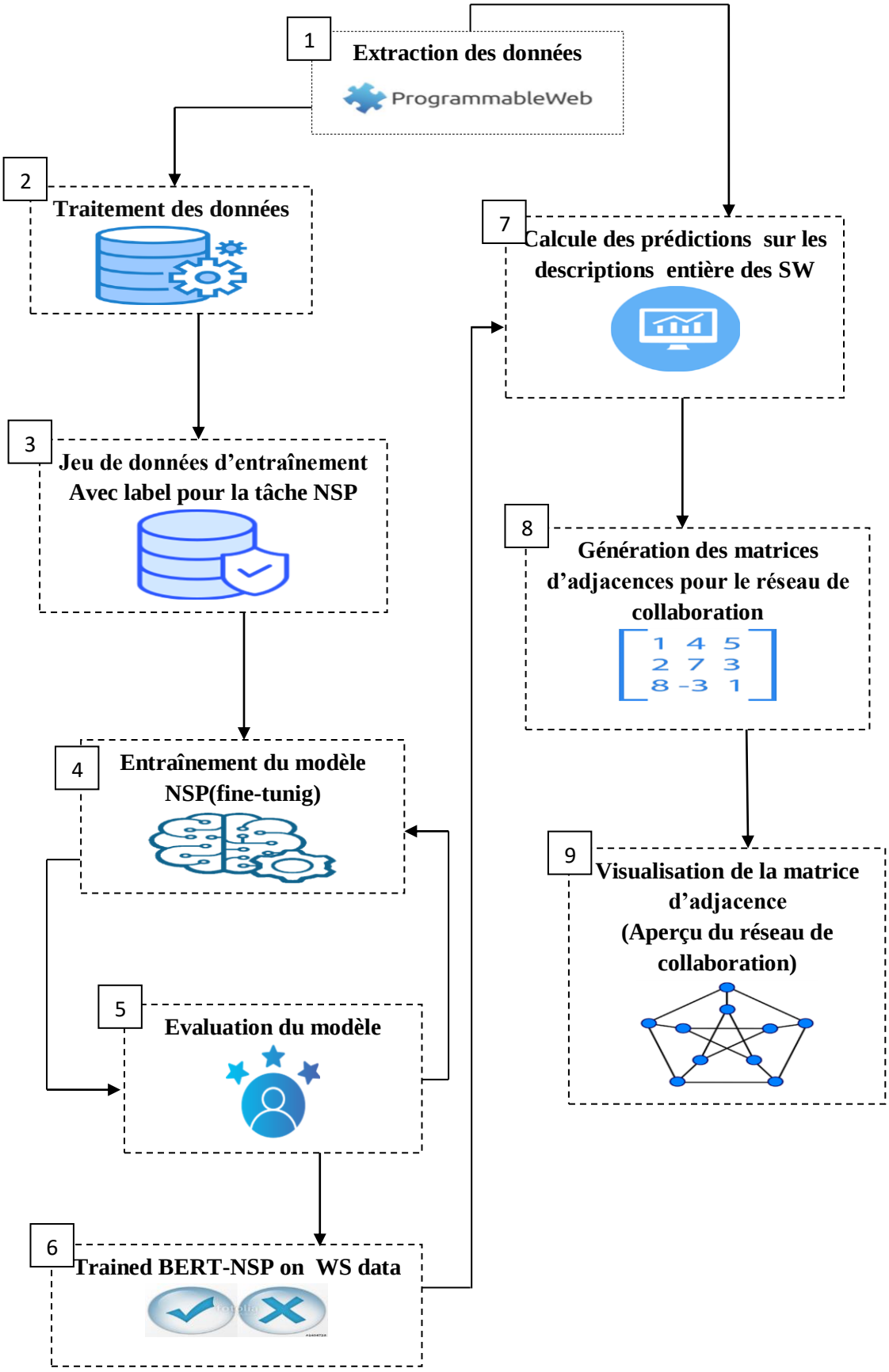

Figure III-6 Notre approche pour l'initialisation du RS de collaboration de SW.

III.5.1 Extraction des données

Dans notre méthodologie de découverte des services Web, nous avons tiré parti d'un ensemble de données authentiques provenant de « programmableweb.com », une référence incontournable en matière d'annuaire de services. Notre ensemble de données comprend des descriptions détaillées des services Web avec lesquels nous avons collaboré, dans le but d'optimiser les résultats de la découverte de ces services.

III.5.2 Traitement des données

Le processus de traitement des données pour l'analyse des services web débute avec les données brutes. Nous commençons par identifier la colonne contenant les descriptions des services web, en éliminant les autres colonnes non pertinentes. Ensuite, nous procédons à un nettoyage préliminaire des données afin de supprimer les caractères spéciaux et le bruit indésirable.

Une fois les données nettoyées, nous les segmentons en phrases individuelles en utilisant des techniques de segmentation de phrases. Ensuite, nous extrayons des paires de phrases en leur attribuant des étiquettes indiquant si elles sont consécutives (1) ou non (0).

Nous appliquons une fonction de « tokenizer » pour encoder les phrases en représentations numériques, facilitant ainsi leur traitement ultérieur.

III.5.3 Jeu de données d'entraînement Avec label pour la tâche NSP

Nous possédons désormais des données annotées avec précision pour la tâche de prédiction de la phrase suivante (NSP). Ces données étiquetées sont de qualité supérieure et peuvent être exploitées pour entraîner notre modèle de manière efficace.

III.5.4 Entraînement du modèle

Le processus d'entraînement du modèle BERT-NSP (Next Sentence Prediction) implique la division de notre jeu de données en ensembles d'entraînement et de validation. Une fois que les données d'entrée sont préparées, le modèle BERT-NSP est entraîné à prédire si deux phrases sont consécutives ou non. Cette étape utilise une fonction de perte, généralement la perte de log-vraisemblance (cross-entropy loss), pour comparer les prédictions du modèle avec les étiquettes réelles (consécutif ou non consécutif). L'entraînement est réalisé sur plusieurs époques (epoch) en utilisant des techniques d'optimisation telles que la descente de gradient stochastique (SGD) ou l'optimisation d'Adam.

III.5.5 Évaluation du modèle

Après avoir terminé l'entraînement, il est important d'évaluer les performances de notre modèle en utilisant l'ensemble de validation. Cette évaluation implique le calcul de diverses métriques spécifiques au problème à résoudre, telles que l'exactitude (accuracy), la précision, le score F1 et la pureté. Ces métriques permettent d'analyser les forces et les faiblesses du

modèle, offrant ainsi des indications sur les ajustements et les améliorations nécessaires pour obtenir un modèle plus fiable et performant.

III.5.6 Collecte des prédictions appliquées sur les descriptions des SW

Une fois que nous avons terminé la phase d'entraînement et d'évaluation du modèle en utilisant des paires de phrases, nous passons maintenant à l'étape où nous considérons l'ensemble des descriptions de services web comme une seule phrase et l'injectons dans notre modèle. L'objectif est de prédire si deux descriptions de services web sont consécutives ou non. En d'autres termes, nous cherchons à déterminer si deux descriptions se suivent dans un ordre séquentiel ou si elles sont disjointes. Nous cherchons à améliorer notre compréhension des relations entre les descriptions de services web en évaluant leur séquentialité.

III.5.7 Génération des matrices d'adjacences pour le réseau de collaboration

Nous avons utilisé le modèle pour extraire des prédictions à partir des descriptions des services web, puis nous avons rempli une matrice d'adjacence carrée avec ces prédictions. Ensuite, nous avons appliqué les formules « comp_normalized » et « complementarity_popularity » sur les éléments de la matrice. Cette étape a permis d'obtenir une matrice d'adjacence représentant un réseau de collaboration des services web.

III.5.8 Visualisation de matrice d'adjacence

La visualisation des liens entre les services web peut être complexe lorsqu'ils sont représentés sous forme matricielle. Afin de faciliter la compréhension et l'observation de ces liens, il est recommandé de convertir la matrice d'adjacence en un graphe en utilisant l'outil « Gephi ». Cette conversion permet une lecture plus claire et une meilleure visualisation des relations entre les services web.

III.6 Discussion des résultats

- **Le Data set utilisé**

Dans le cadre de notre étude, nous avons exploité un échantillon représentatif de données tirées du monde réel, en nous appuyant sur les informations fournies par le célèbre annuaire « programmableweb.com ». Notre Data set ne contient pas moins de 17 923 services Web. Pour notre analyse, nous avons restreint notre attention aux 500 premières descriptions de ces services.

	api_id	api_name	api_pw_url	api_url	api_primary_category	api_secondary_category
0	62687	Google Maps	https://www.programmableweb.com/api/google-maps	https://developers.google.com/maps/	Mapping	[Viewer]
1	63449	Google AJAX Libraries	https://www.programmableweb.com/api/google-aja...	https://developers.google.com/speed/libraries/...	Library	[Application Development]
2	65364	Instagram Graph	https://www.programmableweb.com/api/instagram-...	https://developers.facebook.com/docs/instagram...	Photos	[Mobile, Social]
3	63238	LinkedIn	https://www.programmableweb.com/api/linkedin	https://developer.linkedin.com/docs	Social	[Enterprise]
4	64145	Bing	https://www.programmableweb.com/api/bing	https://azure.microsoft.com/en-us/services/cog...	Search	[Machine Learning]
...	...	...	...	...	...	...
17918	62783	Mint	https://www.programmableweb.com/api/mint	http://www.shauninman.com/archive/2005/10/21/m...	NaN	NaN
17919	62792	Dropcash	https://www.programmableweb.com/api/dropcash	http://www.dropcash.com/doc/api.php	Other	[Charity]
17920	62813	Bunchball	https://www.programmableweb.com/api/bunchball	https://bunchballnet-main.pbworks.com/w/sessio...	Games	NaN

Figure III-7 Le Data set collecté à partir de programmable web.

- **Paramètres d'apprentissage**

Pour réaliser nos résultats, nous avons opté pour les paramètres suivants :

- Max_length = 50
- Batch_size= 40
- Learning_rate= 1e-5
- Nombre epoch=5
- MODEL_BERT = ("bert-base-uncased")

Ensuite, nous avons chargé les données d'entraînement et de validation pour faire l'entrainement.

```
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size=40, shuffle=True)
val_loader = torch.utils.data.DataLoader(val_dataset, batch_size=40, shuffle=False)
```

Figure III-8 Le chargement les données d'entraînement et de validation

Puis faire entrainement, la figure III.9 illustre l'entrainement de notre modèle.

```
num_epochs = 5
optimizer = torch.optim.AdamW(model.parameters(), lr=1e-5)
trained_model, history = fit(train_loader, val_loader, model, optimizer, device, num_epochs)
```

```
T 100%
V 100%
Epoch 1   train loss: 2.3940  train accr: 0.42  val loss: 0.7264  val accr: 0.48
T 100%
V 100%
Epoch 2   train loss: 0.6833  train accr: 0.57  val loss: 0.6700  val accr: 0.59
T 100%
V 100%
Epoch 3   train loss: 0.6633  train accr: 0.60  val loss: 0.6396  val accr: 0.62
T 100%
V 100%
Epoch 4   train loss: 0.6357  train accr: 0.64  val loss: 0.5979  val accr: 0.71
T 100%
V 100%
Epoch 5   train loss: 0.5702  train accr: 0.70  val loss: 0.5260  val accr: 0.74
```

Figure III-9 Illustre l'entrainement de notre modèle.

- **Evaluation du modèle**

Nous évaluons les résultats de notre modèle, selon quatre mesures (précision, Recall, f1- score, la pureté, accuracy, l'information mutuelle normalisée (NMI)).

- **Précision**

Il peut connaître le nombre de prédictions correctes faites.

$$\textbf{Précision} = \frac{\textbf{vrai positif}}{\textbf{vrai positif+faux positif}}$$

Plus il est élevé, plus le modèle d'apprentissage automatique peut réduire le nombre de faux positifs [61].

- **Recall**

Le Recall nous permet de connaître le pourcentage de positifs que le modèle a bien prédit.

$$\textbf{Recall} = \frac{\textbf{vrai positif}}{\textbf{vrai positif+faux négatif}}$$

Plus il est élevé, plus le modèle d'apprentissage automatique peut maximiser le nombre de vrais positifs [61].

- **F1_score**

Le score F1 fournit une bonne évaluation des performances du modèle [61].

$$\textbf{F1\_score} = 2 * \frac{\textbf{recall} * \textbf{precision}}{\textbf{Recall} * \textbf{precision}}$$

- **La pureté**

Calcule la proportion de services correctement regroupés par rapport au nombre total de services [62].

$$Purity\ (B, A)=\frac{1}{N}\sum_{K} MAX\ |B_k \cap A_r|$$

- **Accuracy**

Est utilisée pour évaluer la précision globale d'un modèle de classification. Elle est définie comme le rapport entre le nombre de prédictions correctes et le nombre total d'échantillons dans le jeu de données [62].

$$\textbf{Accuracy} = \frac{\textbf{Nombre de prédictions correctes}}{\textbf{Nombre total d'échantillons}}$$

- **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). Dans cette fonction les informations mutuelles sont normalisées par une moyenne généralisée de H (labels_true) et H (labels_pred), définie par la méthode average_methode [63].

- **Les performances de model BERT-NSP pour les paire des phrase**

Nous allons comparer les résultats du modèle BERT-NSP (Next Sentence Prediction) avant et après l'entraînement sur des paires de phrases. Ces résultats seront présentés dans le tableau ci-dessous.

Méthodes	Précision	F1 score	recall	purity
BERT–NSP « avant »	0.6414	0.7712	0.9669	0.6701
BERT-NSP « après »	**0.7539**	**0.7953**	**0.8415**	**0.7394**

Tableau III-1 Comparaisons des performances du Bert- NSP avant et après entrainement des services Web.

Le tableau met en évidence les performances de notre modèle avant et après son entraînement sur les paires de phrases. Nous avons observé une amélioration moyenne de 8,59 %, ce qui démontre la supériorité des performances de notre modèle après l'entraînement par rapport à avant.

Les figures III.10 et III.11 représentent respectivement le graphe d'exactitude (accuracy) et de perte (loss) du modèle BERT-NSP après l'entraînement.

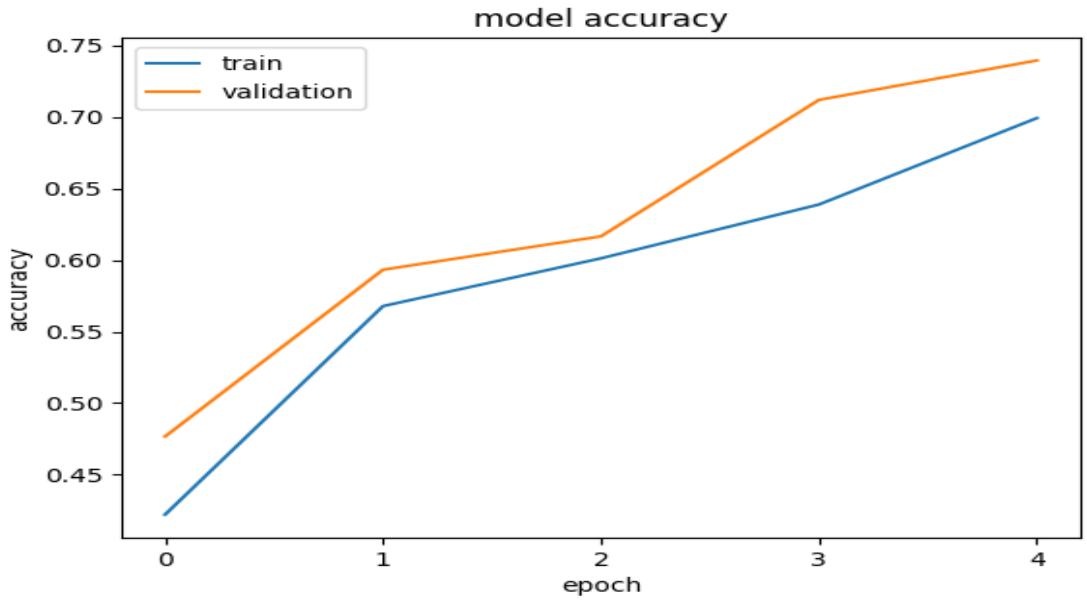

Figure III-10 Le graphe d'accuracy.

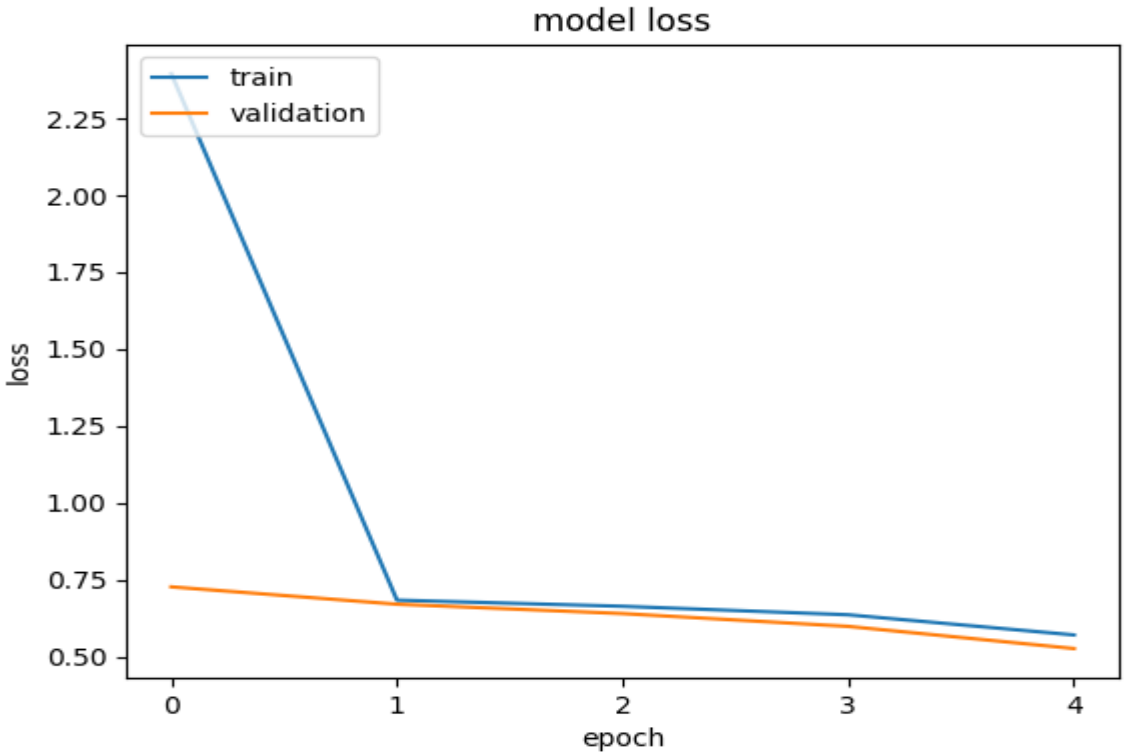

Figure III-11 Le graphe de Loss.

- **Appliquant les prédictions sur les descriptions des services web**

Nous avons créé une toute nouvelle panoplie de données qui comprend des descriptions de services Web, ainsi que des informations sur le nombre de followers et les mashups associés.

api_url	api_primary_category	api_secondary_category	api_desc	followers	mashup
https://developers.google.com/maps/	Mapping	['Viewer']	[This API is no longer available. Google Maps'...	4155.0	GymPost.com
https://developers.google.com/speed/libraries/...	Library	['Application Development']	The AJAX Libraries API is a content distributi...	0.0	0
https://developers.facebook.com/docs/instagram...	Photos	['Mobile', 'Social']	Instagram is a photo sharing iPhone app and se...	1341.0	Listagram
https://developer.linkedin.com/docs	Social	['Enterprise']	LinkedIn is the world's largest business socia...	1647.0	Linkedin RSS
https://azure.microsoft.com/en-us/services/cog...	Search	['Machine Learning']	[The Bing API is now the <a href="https://www....	382.0	AskJot
...	...	...	...	...	...
http://www.shauninman.com/archive/2005/10/21/m...	0	0	In limited public-beta as of Jan 2006.	0.0	0
http://www.dropcash.com/doc/api.php	Other	['Charity']	From their site: There are many ways to intera...	0.0	0
https://bunchballnet-main.pbworks.com/w/sessio...	Games	0	From their site: Have a web page somewhere? Ta...	0.0	0

Figure III-12 La nouveau Data set.

Nous avons ensuite utilisé le modèle BERT-NSP entraîné sur notre nouvel ensemble de données pour obtenir des prédictions et générer une matrice d'adjacence en calculant la différence des valeurs de logits.

$$\text{Compl}(i, j) = a_1 - a_2 .$$

```
import numpy as np
# Charger le fichier .npy
matrice = np.load('/content/drive/MyDrive/Nor_Inv_Mat.npy')

# Accéder au contenu du fichier
print(matrice)
```

```
[[13.14909124  1.59607061  1.89491134 ...  1.68485837  1.19037239
   3.09058274]
 [ 1.26703671 18.69867639  1.78369359 ...  1.11800568  1.20518016
   1.27467673]
 [ 1.23597226  1.20861317 23.12866352 ...  1.09561715  1.17754476
   1.20854884]
 ...
 [ 1.21019065  1.18290899  1.26530322 ... 21.55848898  1.22667659
   1.23600585]
 [ 1.15613393  1.21387325  1.28098128 ...  1.13304994 22.20241935
   1.18142711]
 [ 1.49704392  1.34702195  1.22792551 ...  1.14778127  1.12631116
  28.30727704]]
```

Figure III-13 La matrice d'adjacence.

Ensuit nous avons appliqué les formules suivant sur tous les éléments de la matrice

- **Compl_normalized (i , j) = 1 / (1 + exp(- k * (a_1 - a_2))).**
- **Complementarity_popularity (i, j) = (1 – α) * Compl_normalized (i , j) + α * ($popularity_j$ + ε) / ($popularity_i$ + $popularity_j$ + 2 ∗ ε).**

Popularity i et j sont les nombres de followers.

- **Visualisation de notre réseau de collaboration**

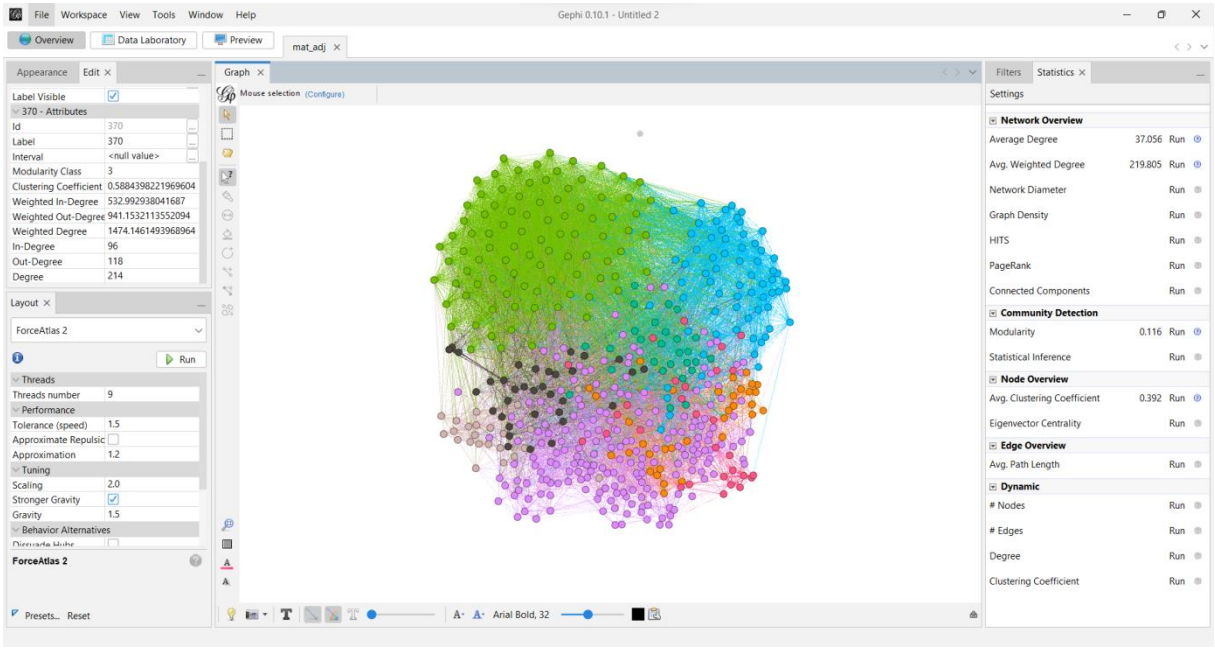

Figure III-14 Visualisation du réseau de collaboration des SW.

III.7 Tests et analyse du comportement de notre modèle

Afin d'évaluer l'efficacité de notre modèle, nous avons effectué des sélections aléatoires de description des services web et extrait les trois premières valeurs correspondantes de la matrice adjacente. Les résultats obtenus sont les suivants :

Prenons l'exemple de la description du service web ayant l'indice 193 dans le jeu de données.

```
data['api_desc'][193]
```

```
'The 8coupons API gives developers access to the full feature set of the 8coupons site. 8coupons brings together all the deals from neighborhood restaurants, bar
s, salons, and stores so that users can find the best deals nearby. The API provides methods such as retrieving dealer types, getting deals by location, getting
deals by store ID and more. The API uses RESTful protocol and responses are formatted in JSON.'
```

Figure III-15 Exemple d'une description d'un service Web.

Ce service permet aux développeurs d'accéder à toutes les fonctionnalités du site 8coupons, incluant la récupération des types de revendeurs, l'obtention d'offres par emplacement, l'obtention d'offres par ID de magasin, etc.

Après avoir analysé la matrice adjacente, nous avons identifié les trois premiers services web avec lesquels le service 193 peut collaborer :

```
import numpy as np
# Charger le fichier .npy
matrice = np.load('/content/drive/MyDrive/Nor_Inv_Mat.npy')

ligne = matrice[193]

# Trouver les indices des trois (3) premières valeurs
indices_min = sorted(range(len(ligne)), key=lambda i: ligne[i])[:3]

print("Indices des 5 valeurs minimales :", indices_min)

Indices des 5 valeurs minimales : [78, 333, 425]

# Afficher la colonne api_desc avec les indices
for index in indices_min:
    value = data['api_desc'].iloc[index]
    print(f"Index: {index}, Valeur: {value}")

Index: 78, Valeur: Helps sellers automate listings, orders, payments, reports, and more. By exchanging data, sellers can integrate Amazon marketplace into their c
Index: 333, Valeur: Integrate your software with Email Marketing from Constant Contact.  Our API allows you to seamlessly integrate our industry-leading Email Mar

Our API allows you to manage email contacts, contact lists and reporting data. Also you can view email marketing campaign results; including which email contacts
Index: 425, Valeur: Provides methods to add, edit and manage advertising campaigns running within Facebook. Your ad management application can work with keywords,
```

Figure III-16 L'identification des trois meilleurs collaborateurs d'un services web.

- Le service 78 aide les vendeurs à automatiser les listes, les commandes, les paiements, les rapports, et plus encore. En échangeant des données, les vendeurs peuvent intégrer le marché d'Amazon dans leurs applications et flux de travail existants.

- Le service 333 s'intègre de manière transparente avec Email Marketing de Constant Contact, fournissant ainsi à nos utilisateurs un accès facile aux fonctionnalités avancées de ce service de marketing par e-mail réputé. Grâce à cette intégration, nous nous engageons à aider nos utilisateurs, qu'ils soient de petites entreprises ou des organisations, à atteindre leurs objectifs de marketing et à réussir dans leurs efforts de communication par e-mail.

- Le service 425 propose des méthodes pour ajouter, modifier et gérer des campagnes publicitaires diffusées sur Facebook. Votre application de gestion des annonces peut fonctionner avec des mots clés, des groupes d'annonces, et même générer des rapports.

Ainsi, les services 78, 333 et 425 peuvent collaborer avec le service 193 pour offrir des fonctionnalités complémentaires dans différents domaines. L'intégration de ces services offre aux vendeurs la possibilité d'automatiser leurs opérations sur Amazon, d'améliorer leurs campagnes de marketing par e-mail et de créer des publicités Facebook plus ciblées, en utilisant les données sur les offres locales disponibles sur le service 193. Cela leur permet

d'optimiser leurs activités commerciales et de proposer de meilleures offres à leurs clients potentiels.

En conclusion, notre modèle a donné des résultats satisfaisants, ce qui démontre sa capacité à établir un réseau de collaboration entre les services web sociaux.

III.8 Conclusion

Dans ce chapitre, nous avons présenté notre approche BERT-NSP pour l'initialisation des RS de collaboration des SW afin d'améliorer la découverte des services web sociaux. Tout d'abord, nous avons étudié les réseaux sociaux et les services web sociaux, en mentionnant quelques travaux pertinents menés dans ce domaine. Ensuite, nous décrivons de manière claire et concise les étapes nécessaires à la mise en place de notre approche , afin de garantir une compréhension complète du processus.

Nous avons commencé par l'extraction d'un ensemble de données à partir de l'annuaire Programmable Web, puis nous avons nettoyé ces données et construis un nouvel ensemble de données pour l'entraînement et l'extraction des prédictions. Ensuite, nous avons réalisé un comparatif des performances de notre modèle avant et après l'entraînement.

Enfin, notre approche offre une alternative puissante aux approches basées sur les ontologies, en permettant une compréhension plus flexible et contextuelle du langage.

Conclusion Générale

Dans ce projet, notre objectif principal était de développer une approche novatrice visant à faciliter la collaboration entre les services web en améliorant la découverte de ces services à l'aide de descriptions textuelles. Nous avons débuté par présenter un état de l'art exhaustif sur les services web, incluant leur définition, leurs technologies fondamentales et les principaux défis associés à leur utilisation. Parmi ces défis, nous avons souligné l'importance cruciale du processus de découverte pour assurer la fourniture de services de qualité. En conséquence, la communauté de recherche a manifesté un vif intérêt pour l'amélioration de la pertinence et de l'efficacité de cette phase de découverte.

L'émergence des nouvelles technologies issues du domaine du Deep Learning et leur succès ont donné lieu à une nouvelle tendance : l'application des techniques de Deep Learning aux données et aux descriptions des services web. Cette approche innovante offre des perspectives prometteuses pour améliorer la découverte des services web en exploitant les capacités des modèles de Deep Learning dans le traitement des informations textuelles.

Notre travail vise à proposer une approche permettant de faciliter la mise en place d'un réseau social des SW sémantiques et ce sans avoir recours à des ontologies et améliorant ainsi la découverte des services web sociaux en employant une technique de Deep Learning appelée "BERT-NSP" pour l'initialisation de leur réseau social de collaboration. Nous avons entraîné et testé notre modèle en utilisant un ensemble de données provenant du monde réel, fourni par "programmableweb.com", un annuaire largement reconnu. En comparant les performances de notre modèle avant et après son entraînement sur cet ensemble de données, nous avons observé des résultats expérimentaux qui ont démontré l'efficacité de notre approche. Ces résultats ont confirmé que notre modèle était capable d'améliorer significativement la découverte des services web en utilisant de simple descriptions textuelles associées à ces services.

Comme perspective à ce travail, nous allons essayer d'exploite les réseaux sociaux que nous avons pu élaborer et combiner des techniques issues du domaines du Sociale Network Analyse (SNA) et des nouveauté du domaines DL appliquées sur les graphes ,afin de proposer une nouvelle approche de découverte et de détection de nouveaux services et de communautés cachées.

Bibliographie

[4]BOUCHAKOUR Ibtissem « reconfiguration dynamique d'un service web d'agents mobiles », thèse de doctorat en Informatique, Université Mohamed Khider, Biskra.

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

[8]AOUICHE Ahmed « Sélection des services Web dans les systèmes distribués :étude comparative », université Larbi Ben M'hidi Oum el Bouaghi.

[9]Okba Kazar, « Une approche pour la découverte sémantique des services Web dans les réseaux mobiles ad-hoc », Université Mohamed Khider, Biskra.

[11]Dr Djamel Benmerzoug, « Composition de Services Web », Université Constantine 2, Abdelhamid Mehri ».

[12]Khedim Asmaa « Accélérer la découverte de services web sémantiques », Université Abdelhamid Ibn Badis, Mostaganem.

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

[17]DJOUDI Youcef « Implémentation d'un Algorithme de Découverte de Services Web dans le Contexte des Réseaux Sociaux », Université Djillali Liabès ,Sidi-Bel-Abbès .

[18]Z. Maamar, L. K. Wives, Y. Badr, and S. Elnaffar. Even web services can socialize : A new service-oriented social networking model. In Intelligent Networking and Collaborative Systems, 2009. INCOS'09. International Conference.

[19]Z. Maamar, H. Hacid, and M. N. Huhns. Why web services need social networks. IEEE Internet Computing, 2011.

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

[25]Slimane Yacine ,Tahar djoudi Salim « Extraction de motifs basée sur world2vec »,mémoire de master informatique ,universite Saad Dahleb Blida -1-.

[27]ATIA HADJER « Accélération du processus d'intégration de termes dans le deep learning »,memoire de master académique en informatique. Université Mohamed Khider,BISKRA.

[30]chikh Farida « Extension d'un modèle d'expansion de requêtes pour la prise en compte de la représentation de type word embedding »,Mémoire de Master Académique. UNIVERSTE MOULOUD MAMMERI DE TIZI-OUZOU.

[33]Mohamed Abd Elmoumen DJABALLAH. « Système de prédiction de la consommation d'énergie basé Deep Learning », Mémoire de Master Informatique. Université de 8 Mai 1945, Guelma.

[35]KLOUL Nawel ,MEDDAH Yasmine « Classification de polarité d'opinions à base d'aspects à l'aide de l'apprentissage profond », Mémoire de master académique en informatique, UNIVERSITÉ MOULOUD MAMMERI DE TIZI OUZOU.

[43]SAADOUNI Oumaima , MEHIA Yassine Salah Eddine,« Analyse des sentiments avec le deep learning » ,Mémoire de master en informatique, Université Larbi Ben M'hidi d'Oum el Bouaghi.

[48]Yi Sun, Yu Zheng , Chao Hao, Hangping Qiu « NSP-BERT: A Prompt-based Few-Shot Learner Through an Original Pre-training Task , Next Sentence Prediction ».

[53]Maamar, Z, Faci, N, Wives, L. K, Yahyaoui, H, and Hacid, H « Towards a method for engineering social web services. In IFIP International Federation for Information Processing».

[54]Maamar, Z, Yahyaoui, H, Lim, E, and Thiran, P « Social engineering of communities of web services. In the 11th IEEE/IPSJ Symposium on Applications and the Internet Munich, Germany».

[55]Johann Stan, Fabrice Muhlenbach, Christine Largeron, «Recommender Systems using Social Network».

[56]Ignacio Lizarralde , Cristian Mateos , Alejandro Zunino , Tim A. Majchrzak , Tor-Morten Grønli , « Discovering web services in social web service repositories using deep variational autoencoders».

[57]Chen, W, Paik, I, Hung, P. C. K. «Constructing a global social service networkfor better quality of web service discovery».IEEE Transactions on Services Computing.

[58]Alejandro Corbellini, Daniela Godoy;,Cristian Mateos,Alejandro Zunino, Ignacio Lizarralde «Mining Social Web Service Repositories for Social Relationships to Aid Service Discovery».

[59]Abderrahmane Maaradji ,Hakim Hacid, Ryan Skraba, Adnan Lateef,Johann Daigremont, Noël Crespi « Social-Based Web Services Discovery and Composition for Step-by-Step Mashup Completion».

[60]Shangguang Wang , Lin Huang , Ching-Hsien Hsu , Fangchun Yang , « Collaboration reputation for trustworthy Web service selection in social networks».

[62]Guobing Zou, Zhen Qin, Qiang He, Pengwei Wang, Bofeng Zhang, Yanglan Gan,
« DeepWSC: Clustering Web Services via Integrating Service Composability into Deep Semantic Features».

Webographies

[1]Web services architecture « Web Services Architecture (w3.org) » (La dernière consultation 02/06/2023).

[2]Découverte d'annuaires de services web dans un environnement distribué «https://www.researchgate.net/publication/233932802_Decouverte_d%27annuaires_de_services_web_dans_un_environnement_distribue » (La dernière consultation 02/06/2023).

[3]Les services web, « https://fr.slideshare.net/dihiaselma/les-web-services »(La dernière consultation 02/06/2023).

[6]Web services SOAP ET Rest « https://fr.slideshare.net/RadhoueneRouached/web-services-soap-et-rest» (La dernière consultation 02/06/2023).

[7]Fonctionnement d'une APIRESTful « https://www.datatransitionnumerique.com/apirest/ » (La dernière consultation 02/06/2023).

[10]Cycle de vie de développement de services Web « http://mgharzouli.e-monsite.com/medias/files/chapitre-1-partie-10.pdf » (La dernière consultation 02/06/2023).

[14]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 13/06/2023).

[15] Main Web service concepts in OWL-S « https://www.researchgate.net/figure/Main-Web-service- concepts -in- OWL-S-fig2_309201011 » (La dernière consultation 13/06/2023).

[16] The four main WSMO components « https://www.researchgate.net/figure/The-four-main- WSMO-components-Roman-et-al-2005-fig2-269705445 » (La dernière consultation 14/06/2023).

[20] SAOULI Hamza « Découverte de services web via le Cloud computing à base ».(La dernière consultation 02/06/2023).

[22] Introduction au NLP (Partie I) « https://www.ekino.fr/publications/introduction-au-nlp partie-i/ » (La dernière consultation 22/06/2023).

[23] NLP définition, avantages et cas d'usage« https://blog.smart-tribune.com/fr/definition-nlp » (La dernière consultation 02/06/2023).

[24]Natural Language Processing (NLP) Définition et principes, « https://datascientest.com/introduction-au-nlp-natural-language-processing » .(La dernière consultation 02/06/2023).

[26] LES RNN, LES LSTM, LES GRU ELMO, « https://lbourdois.github.io/blog/nlp/RNN-LSTM-GRU-ELMO/» (La dernière consultation 02/06/2023).

[28]Pirmin LEMBERGER ,Thomas SCIALOM. DEEP TRANSFER LEARNING – LE TRAITEMENT DU LANGAGE À L'AUBE D'UNE RÉVOLUTION «https://weave.eu/deep-transfer-learning-nlp-revolution » (La dernière consultation 23/06/2023).

[29]Continuous Bag-of-words (CBOW, CB) and Skip-gram (SG) « https://www.researchgate.net/figure/Continuous-Bag-of-words-CBOW-CB-and-Skip-gram-SG-training-model-illustrations_fig1_326588219 » (La dernière consultation 22/06/2023).

[31]The model architecture of GloVe The input is a one hot . representation « https://www.google.com/search?q=L%27architecture+du+mod%C3%A8le+de+GloVe&tbm » (La dernière consultation 24/06/2023).

[32] À la découverte de BERT « https://ledatascientist.com/a-la-decouverte-de-bert/ » (La dernière consultation 25/06/2023).

[34]Recurrent Neural Networks « https://d2l.ai/chapter_recurrent-neural-networks/index.html » (La dernière consultation 16/06/2023).

[36] Attention is all you need comprendre le traitement naturel du langage avec les modèles Transformers « https://france.devoteam.com/paroles-dexperts/attention-is-all-you-need-comprendre-le-traitement-naturel-du-langage-avec-les-modeles-transformers/ » (La dernière consultation 02/06/2023).

[37]Les transformers, la nouvelle technologie qui change notre façon de faire du NLP « https://www.kernix.com/article/les-transformers-la-nouvelle-technologie-qui-change-notre-facon-de-faire-du-nlp/ » (La dernière consultation 14/06/2023).

[38]À la découverte du Ttransformer « https://ledatascientist.com/a-la-decouverte-du transformer/ » (La dernière consultation 18/06/2023).

[39] The Transformer Architecture « https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html » (La dernière consultation 18/06/2023).

[40]BERT : Le "Transformer model" qui s'entraîne et qui représente « https://lesdieuxducode.com/blog/2019/4/bert--le-transformer-model-qui-sentraine-et-qui-represente » .(La dernière consultation 18/06/2023).

[41]L'architecture de BERT« https://medium.com/@butonnico/larchitecture-de-bert-f2528d4ed627 » .(La dernière consultation 19/06/2023).

[42]Bidirectional Encoder Representations from Transformers(BERT) « http://d2l.ai/chapter_natural-language-processing-pretraining/bert.html » (La dernière consultation 02/06/2023).

[44]LSTM, Transformers, GPT, BERT : guide des principales techniques en NLP « https://france.devoteam.com/paroles-dexperts/lstm-transformers-gpt-bert-guide-des-principales-techniques-en-nlp/» .(La dernière consultation 17/06/2023).

[45]BERT : le modèle de langue phare de Google « https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1516189-bert-le-modele-de-langue-phare-de-google/ » (La dernièr consultation 19/06/2023).

[47]Comprendre le modèle BERT « https://ichi.pro/fr/comprendre-le-modele-bert-176257345598377 »(La dernière consultation 02/06/2023).

[48] Intuitive Explanation of BERT- Bidirectional Transformers for NLP « https://www.google.com/imgres?imgurl=https%3A%2F%2Fcdn-images ».(La dernière consultation 20/06/2023).

[49]J. Ethier. Current research in social network theory.In « http ://www.scribd.com/doc/11171859/Current-Research-in-SocialNetwork-Theory » .(La dernière consultation 21/06/2023).

[51]Graph Algorithm for Social Media Network Analysis « https://medium.com/socialpages/graph-algorithm-for-social-media-network-analysis-cbba87e28587 ».(La dernière consultation 24/06/2023).

[52]Représentation Matricielle d'un réseau sociaux «https://groupefmr.hypotheses.org/files/2015/01/henry_fekete_2008_reseaux.pd » (La dernière consultation 25/06/2023).

[61]TOM KELDENICH, Recall, Precision, F1 Score – Explication Simple Métrique en ML, « https://www.inside-machinelearning.com/recall-precision-f1-score »
(La dernière consultation 25/06/2023).

[63]Sklearn.metrics.normlized_mutual_info_score «https://scikitlearn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_scor e.html » (La dernière consultation 26/06/2023).