UNIVERSITE DE TUNIS
Institut Supérieur de Gestion

جامعة تونس
المعهدالعالي للتصرف

www.isg.rnu.tn

Mémoire de Mastère

Présenté en vue de l'obtention du

Diplôme de Mastère en Informatique

par

Marwa Louati

Le traitement automatique des descriptions textuelles des services web en se basant sur des algorithmes d'apprentissage automatique

Soutenu le .. .. devant le jury composé de :

Monsieur	.....	Président
Monsieur	.....	Rapporteur
Madame	Nadia YAÂCOUBI AYADI	Encadrante
Madame	Hadhèmi ACHOUR	Co-Encadrante

À ma chère famille,

Pour la patience et le dévouement dont elle a fait preuve.

–Marwa

Remerciements

Je remercie Dieu tout Puissant de m'avoir permis de mener à terme ce mémoire de mastère qui est pour moi le point de départ d'un merveilleux chemin, celui de la recherche, source de remise en cause permanente et de perfectionnement perpétuel.

Ce travail de mastère présenté dans ce mémoire a été effectué à l'*Institut Supérieur de Gestion de Tunis* ISGT. Je voudrais exprimer ma gratitude et ma sincère reconnaissance à tous ceux qui m'ont encouragé.

Je tiens à remercier le président du jury Monsieur Lamjed Ben Said qui a bien voulu accepter l'évaluation de ce travail.

Je remercie Madame Wahiba ben Abdessalem qui a bien voulu accepter d'être le rapporteur de ce travail.

Je souhaite témoigner ma plus sincère gratitude à mon encadrante Madame Nadia Yaâkoubi Ayadi, maître assistante à l'*Institut Supérieur de Gestion de Tunis* pour son écoute et ses conseils tout au long de cette année. Je la remercie tout particulièrement pour les réflexions que nous avons pu mener ensemble et au travers desquelles elle a partagé une partie de son expérience avec moi.

Un grand merci également à Madame Hadhèmi Achour, maître assistante à l'*Institut Supérieur de Gestion de Tunis* pour son engagement dans le co-encadrement de mon mémoire. Sa participitation aux discussions de fond, ses suggestions éclairées et critiques honnêtes, son enthousiasme et ses encouragements auront été pour moi essentiels.

Enfin, je souhaite remercier ma famille, mes collègues et mes amis pour leur encouragement et leur soutien moral.

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

CHAPITRE 1

Introduction générale

Sommaire

1.1 Introduction

Actuellement, l'accès aux systèmes d'information repose d'une manière progressive sur les technologies *Internet*, ce qui a accentué les propositions de standardisation qui ont accru l'engouement des gens et des organisations de tout type pour le suivi de cette technologie, et qui ont donné lieu à la propagation des services web. Ces derniers sont devenus un besoin inévitable pour l'interaction entre applications.

En effet, la technologie des services web est apparue comme un support fiable offrant la possibilité de dialogue entre deux applications provenant de différentes plateformes et ce, en échangeant des données via *Internet*. Ceci va donner lieu à une utilisation massive de l'internet en répondant aux différents besoins de l'utilisateur d'une manière

dynamique et active. Les services web offrent divers standards qui facilitent le transport, l'invocation, la description et la recherche, à savoir :

- Le protocole SOAP (*Simple Object Access Protocol*) qui s'occupe du transport des messages XML entre les services Web ainsi que leur invocation.
- Le langage WSDL (*Web Services Description Language*) qui s'intéresse à décrire l'interface d'un service Web.
- Le standard UDDI (*Universal Description Discovery and Integration*) supporte la recherche et la découverte automatique des services en décrivant l'annuaire répertoriant ces derniers d'une manière homogène [34].

Pour y accéder, l'internaute doit tout d'abord disposer des différentes descriptions des services qu'il désire. Cependant, le manque de sémantique dans les langages de description entrave le catalogage automatique des descriptions de services lors de la phase de publication ainsi que la phase de découverte. En effet, les problèmes de découverte s'élèvent lorsque l'utilisateur soumet une demande sous forme d'une requête sur des services publiés dans un annuaire et que le degré de similarité entre les descriptions de service requis et celles des services publiés est faible, ce qui l'empêche d'obtenir le résultat exact [11].

Problématique :

Une description textuelle d'un service web est un texte décrivant ses fonctionnalités à savoir, ce qu'il prend en entrée ainsi que ce qu'il génère en sortie. Notre travail se focalise sur les descriptions contenues dans Biocatalogue qui est un registre public, organisé et contrôlé, exposant les services web des sciences de la vie [44]. Notre but est de faciliter la recherche et l'exploitation des services web dans cet annuaire, en garantissant le meilleur niveau d'exactitude ainsi que le temps de réponse optimal. Pour ce faire, nous visons à annoter les Services Web dans le domaine de la bioinformatique, et ce en exploitant les descriptions textuelles qui leur sont associées. Il s'agit en effet, d'extraire à partir de ces descriptions, un ensemble d'annotations sémantiques. Pour cela, nous considérons notre problème comme étant un problème d'étiquetage sémantique consistant à associer

à chaque mot d'une description textuelle, une étiquette parmi les suivantes : le nom du service (SN), les données d'entrée (IN) ainsi que celles de sortie (OU), en se basant notamment sur une ontologie. Notre problème revient donc à un problème d'étiquetage textuel souvent approché à l'aide d'algorithmes d'apprentissage automatique, ou encore à l'aide de classificateurs de séquences dont les plus connus sont : HMM (*Hidden Markov Models*), MEMM (*Maximum Entropy Markov Models*) et CRF (*Conditional Random Fields*).

1.2 Plan du document

Le présent document est organisé de la façon suivante.

Dans le premier chapitre, nous présentons un état de l'art sur la problématique de l'annotation sémantique des Services Web. Nous y présentons dans une première partie, une revue de la littérature des différents travaux d'annotation de SW, puis nous présentons les principaux algorithmes d'apprentissage automatique d'étiquetage de séquences et qui sont généralement utilisés pour l'extraction d'annotations sémantiques à partir de textes.

Dans le deuxième chapitre, nous présentons l'approche que nous proposons pour extraire des annotations sémantiques de SW à partir de textes, et qui consiste à adopter une approche probabiliste d'étiquetage de séquences. Nous présentons en particulier deux méthodes d'étiquetage : une méthode fondée sur les modèles de Markov cachés (HMM), et une méthode fondée sur les champs aléatoires conditionnels (CRF).

Quant au troisième chapitre, il est consacré à l'implémentation des deux méthodes proposées, à l'évaluation et la comparaison des résultats obtenus en termes d'annotation sémantique des SW.

Nous clôturons par une conclusion suivie par les perspectives ouvertes par notre travail.

CHAPITRE 2

État de l'art

Sommaire

2.1 Introduction

Le Web sémantique propose d'accorder les métadonnées, se présentant sous la forme de concepts, aux différentes ressources du Web dans le but de donner une représentation formelle du sens de ces ressources et de favoriser leur traitement automatique. Ce processus, nommé annotation sémantique [4], s'appuie sur des modèles de représentation de connaissances tels que les ontologies. L'annotation sémantique des ressources du web est considérée comme une étape fondamentale pour la construction de l'infrastructure du web sémantique.

Ce chapitre est structuré de la façon suivante. Dans le paragraphe 2.2, nous décrivons le processus d'annotation sémantique. Le paragraphe 2.3 présente les différents modèles statistiques utilisés pour le traitement des séquences.

2.2 Approches pour l'annotation sémantique des services web

Dans cette section, nous présentons un certain nombre de travaux qui se sont intéressés à la génération de descriptions sémantiques de services web. La plupart de ces travaux repose sur la description WSDL [1] d'un service.

2.2.1 MATAWS (A Multimodal Approach for Automatic WS Annotation)

L'approche MATAWS proposée par *Cihan Aksoy* et al.[1], vise à automatiser le processus d'annotation. Elle se distingue par l'exploitation de diverses sources d'informations, entre autres les fichiers WSDL. En effet, son objectif est de mettre l'accent sur la sémantique des données, plus précisément les paramètres d'entrée et de sortie des services web contenus dans une collection de fichiers WSDL, en vue de générer une collection de fichiers OWL-S [10] comme sortie.

1. WSDL (*Web Service Description Language*) : le standard permettant de décrire l'interface d'un service Web

La structure modulaire de l'outil MATAWS comporte 5 éléments de base (Figure 2.1) :

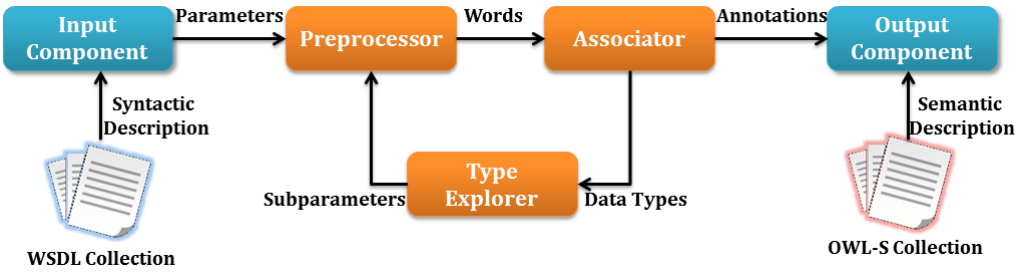

Figure 2.1 – Architecture de MATAWS [1]

Le premier composant de cette architecture, «*Input Component*», s'intéresse à l'extraction de l'ensemble de paramètres définis dans les fichiers WSDL en utilisant un analyseur pour récupérer les noms des paramètres ainsi que leurs types. D'un autre côté, le dernier composant «*Output Component*» se charge de générer une collection de fichiers OWL-S comportant les informations originelles associées aux concepts adéquats.

Le processus d'annotation est assuré par les trois composants restants : le «*Preprocessor*» s'occupe de la décomposition et la normalisation des noms des paramètres déjà récupérés pour que le composant «*Associator*» puisse les traiter. Ce dernier est basé sur le moteur d'inférence *Sigma* [36] qui se charge d'affecter un concept ontologique à un mot. Par exemple, le concept attribué au mot «*school*» est «*Educational Process*». Comme *Sigma* n'est pas toujours capable de trouver le concept approprié, il y a eu recours à l'utilisation de «*Type Explorer*» qui cherche à obtenir des sous-paramètres à partir de certaines propriétés liées aux types de données traitées, qui seront récupérés par la suite par le «*Preprocesseur*».

L'outil MATAWS a été appliqué sur une large collection de descriptions WSDL à partir du projet ASSAM (*Automated Semantic Service Annotation with Machine learning*) [19], contenant 7877 opérations extraites à partir de 816 descriptions. Un paramètre est considéré annoté si l'outil réussira de lui affecter au moins un concept défini dans une ontologie du domaine. Le tableau 2.1 illustre les résultats d'annotation obtenus.

Modification ajoutée	Proportion des paramètres annotés
Pas de prétraitement	39,63%
Décomposition	41,94%
Normalisation	90,01%
Filtrage	69,06%
Explorateur	72,04%

Tableau 2.1 – Le taux obtenu de l'utilisation des fonctionnalités de MATAWS [1]

Les résultats s'améliorent au fur et à mesure de l'ajout d'une phase du processus d'annotation tel que proposé par MATAWS. Dans le cas d'absence du prétraitement, MATAWS peut proposer un concept pour 39,63% des paramètres. Ce taux indique que vers 40% des paramètres sont des mots simples pouvant être récupérés directement dans la base de connaissances *WordNet* et le reste nécessite un prétraitement pour être annoté. Suite à l'introduction de l'étape de décomposition, nous remarquons une brève amélioration de plus de 2%, cela désigne que les mots composés ne contiennent pas des mots directement reconnaissables. Après avoir effectué la phase de normalisation et ce, en remplaçant les abréviations par les mots entiers adéquats, nous remarquons une amélioration importante de l'ordre de +48%. Ensuite, on distingue une chute de -21% lors du passage à la phase du filtrage et ce à cause des «*stop-words*» ou encore de leur concaténation. Enfin, le taux des paramètres annotés atteint un pourcentage de 72,04% et ce peut être justifié par le type des paramètres qui est un type personnalisé, difficile à annoter.

2.2.2 MWSAF : METEOR-S Web Service Annotation Framework

Le souci de la procédure d'annotation est le choix d'ontologies pertinentes ayant une couverture large du vocabulaire. Toutefois, la taille des ontologies d'une part et le nombre exponentiel disponible sur le Web rendent la tâche d'annotation fastidieuse. De ce fait, il est nécessaire d'avoir une solution évolutive pour annoter avec le plus d'automatisation sans perte de qualité. Nous présentons dans ce qui suit METEOR-S Web Service Annotation Framework nommé MWSAF [35] qui se charge d'annoter ces descriptions d'une manière semi-automatique par des métadonnées en se basant sur des

ontologies pertinentes. L'approche fournit un mécanisme d'enrichissement sémantique décrivant à la fois la sémantique fonctionnelle et la qualité de services (propriétés non fonctionnelles) sous la forme d'annotations insérées dans le fichier WSDL. Elle se base sur les techniques de correspondances entre les fichiers WSDL représentant les descriptions des services, et des fichiers OWL se référant aux ontologies. Ces techniques consistent à calculer un degré de similarité entre les éléments d'une description WSDL (tels que le nom d'une opération, le nom d'un paramètre) et les concepts d'une ontologie. Plusieurs algorithmes de *matching* ont été proposés tel que NGram [23].

Dans un premier temps, MWSAF a été testé en se basant sur les domaines météorologiques et géographiques et ce, en appliquant l'algorithme implémenté sur 24 services web dont 15 services appartenant au domaine géographique et 9 au domaine météorologique. Les résultats obtenus sont les suivants :

- Pour une valeur du seuil de catégorisation qui est égale à 0.5, le rappel est égal à 58%.
- Pour une valeur du seuil de catégorisation qui est égale à 0.4, on distingue une catégorisation erronée de deux services.

En vue de remédier aux lacunes trouvées dans cette version de MWSAF, les auteurs ont décidé de l'automatiser en s'appuyant sur l'apprentissage automatique. Le classifieur mis en œuvre est *Naïve Bayes* [47]. Il détermine la probabilité d'appartenance d'un service à une catégorie précise et ce, en ce basant sur le produit des probabilités de chaque mot simple y faisant partie. Ce modèle met en évidence l'exploitation des caractéristiques (nommées en anglais *Features*), qui sont les noms des méthodes et les noms des arguments. Ainsi, au niveau d'une description WSDL, il prend en entrée un vecteur de *Features* extraites dont chacune est représentée par la fréquence d'un mot donné dans la description. Notamment, ce modèle repose sur une base d'apprentissage qui lui est fournie sous forme de matrices de caractéristiques (les lignes correspondent à des vecteurs et les colonnes à des fréquences des mots). Après avoir construit un modèle d'apprentissage à partir d'un corpus d'apprentissage, le rôle du classifieur consiste à prédire le domaine

adéquat aux différents mots d'une description et ce, en produisant des prédictions pour chaque domaine.

Cette approche a été testée sur 37 services dont 21 appartenant au domaine géographique et 16 au domaine météorologique. Les auteurs ont varié la taille de la base d'apprentissage dans le but d'évaluer la qualité des résultats. Ces derniers montrent que lorsqu'on fournit une base d'apprentissage de petite taille, la précision rencontre une chute mais juste après, elle demeure stable de l'ordre de 68%. Ceci semble encourageant car il montre que la taille n'a pas d'impact sur la qualité de catégorisation, ce qui prouve qu'il s'agit d'une approche passable à l'échelle.

2.2.3 **Approche de classification basée sur les techniques d'apprentissage automatique**

Outre l'objectif d'annoter une description WSDL, cette approche s'intéresse à classer automatiquement les services web selon des domaines ou des catégories bien spécifiques et à former un réseau identifiant les différents concepts clés extraits à partir des descriptions textuelles des services Web. Les sources disponibles pour l'extraction de ces descriptions textuelles sont les fichiers WSDL, où il y aura un prétraitement et une normalisation des mots y contenus, les requêtes utilisateurs ainsi que l'exploitation des commentaires accompagnant le code source. La classification est réalisée en se basant sur la méthode SVM (nommée en anglais *Support Vector Machine*) [22], implémentée à l'aide de l'outil LIBSVM [2].

Après avoir effectué les deux phases précédentes, les séquences de mots obtenues seront mappées dans des vecteurs et ce, en utilisant les techniques d'extraction d'informations. Chaque élément correspond à un terme et chaque terme possède un poids affecté moyennant la métrique *tf-idf* [39] (nommée en anglais *term frequency - inverse document frenquency*). L'ensemble de tous les documents est stocké dans une matrice, où les lignes correspondent aux vecteurs de documents et les colonnes aux poids y affectés. L'algorithme SVM est un algorithme supervisé nécessitant un ensemble de documents

2. LIBSVM (*A Library for Support Vector Machine*)

d'apprentissage. En effet, ceci mènera à générer un modèle qui servira par la suite à la phase de prédiction. On présente ci-dessous le processus de classification [8] illustré par la figure 2.2 :

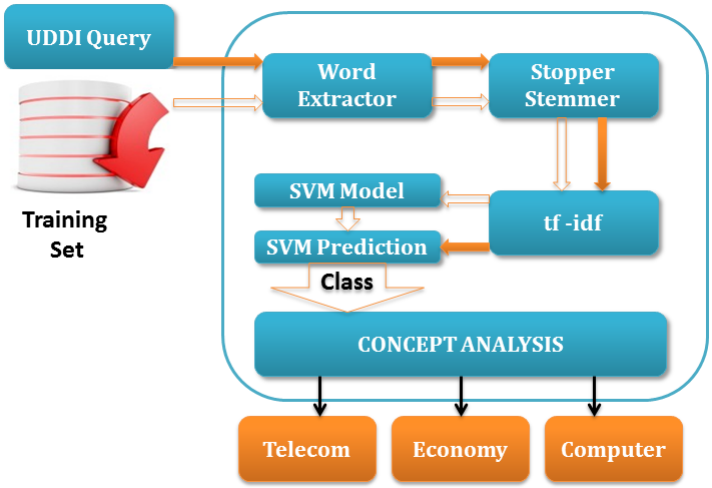

Figure 2.2 – L'approche de classification de services [8]

Pour ce faire, *Bruno et al.* [8] ont utilisé une collection de 205 services web classés sous 11 classes, chaque service est accompagné de sa description définissant sa tâche fondamentale. Suite à la phase d'apprentissage, la méthode SVM est appliquée dans le but de donner une liste ordonnée de classes adéquates à chaque service. La précision obtenue est de l'ordre de 83% de l'affectation réussie de 3 classes. Le taux d'erreur de la classification peut être expliqué par l'ambiguïté existante dans les descriptions textuelles ce qui nécessite leur rectification.

Les trois approches citées auparavant aboutissent à annoter sémantiquement les descriptions des services web, certainement chacune d'elles possède ses propres spécificités qui seront récapitulées dans le tableau 2.2 :

Approche	Rôle	Entrée	Sortie	Limites
MTAWAS	Affectation d'un concept ontologique aux SW	Collection de fichiers WSDL	Collection de fichiers OWL-S	Intervention humaine pour le contrôle des résultats
METEOR-S	Classification des SW selon des domaines	Collection de fichiers WSDL	Collection de fichiers WSDL-S [18]	Présence de mots en commun ainsi que des fréquences de mots communes entres les fichiers WSDL, ce qui engendre un chevauchement ⇒ incapacité d'identifier les *Features* ⇒ domaine erroné
Réseau de concepts	Classification des SW selon des domaines	Descriptions textuelles extraites à partir des fichiers WSDL	Réseau de concept pour les différents services	La mauvaise qualité de la documentation

Tableau 2.2 – Comparaison des approches d'annotation sémantique

2.2.4 Annotation sémantique à l'aide des patrons lexico-syntaxiques

L'approche basée sur les patrons lexico-syntaxiques proposée par Y.Ayadi et al [3] a pour objectif l'étude de la génération semi-automatique d'annotations sémantiques.Un patron lexico-syntaxique est une description d'une expression régulière formée d'un ensemble de mots ou de catégories grammaticales visant à identifier des fragments de texte [21]. Il s'agit d'extraire un ensemble de patrons en exploitant la documentation textuelle des services Web. Cette dernière est bien utile pour construire un corpus d'apprentissage étiqueté grammaticalement et annoté manuellement avec les étiquettes sémantiques souhaitées. La découverte de patrons lexico-syntaxiques s'entame par la précision du type d'information à détecter. Ensuite, il y aura extraction de tous les contextes d'apparition de cette information contenant les données lexicales et syntaxiques de chaque terme. Le patron sera enfin formé après avoir dégagé les caractéristiques syntaxiques et lexicales communes entre les contextes extraits. Une fois les patrons sont définis, ils seront projetés

sur un corpus de test non étiqueté afin de les évaluer.

2.2.5 Synthèse

Les approches d'annotations sémantiques vues dans la section précédente présentent un certain nombre de limitations. En effet, l'approche MATAWS et l'approche MWSAF se basent sur les collections WSDL des SW comme étant des fichiers d'entrée. Cependant, cette source n'est pas efficace pour l'annotation vu sa structure, qui est en fait un document XML décrivant les types de données utilisées par le service ainsi que les messages reçus. D'un autre côté, nous avons cité des approches qui exploitent des descriptions textuelles telles que l'approche de classification qu'ont proposée Bruno et al. et l'approche semi-automatique d'annotation sémantique basée sur les patrons lexico-syntaxique. En effet, parmi les limitations soulevées par cette dernière, nous nous rendons compte que les patrons sont extraits manuellement et peuvent ne pas couvrir tous les cas souhaités lorsque la taille du texte augmente. Ceci peut augmenter le taux d'erreur.

Afin de pallier les limites des approches proposées, nous avons pensé à recourir à un traitement automatique des descriptions textuelles des SW en s'appuyant sur les modèles probabilistes. L'atout de ces modèles est l'utilisation des techniques d'apprentissage automatique ayant pour objectif l'extraction et l'exploitation de l'information contenue dans un ensemble de données et ce, d'une manière automatique. Pour cela, nous proposons un bref état de l'art des différents modèles probabilistes.

2.3 Modèles graphiques probabilistes

Les modèles probabilistes constituent un outil pour les opérations d'étiquetage et de segmentation de séquences textuelles. Ces techniques sont largement adoptées pour les problèmes d'étiquetage grammatical [32], de reconnaissance d'entités nommées [52], et d'analyse de sentiments [45]. On s'intéresse dans cette partie à expliquer le principe de ces modèles en exposant quelques travaux réalisés.

2.3.1 Les modèles de Markov cachés (HMM)

Parmi les approches les plus répandues dans la modélisation des séquences, on cite les méthodes markoviennes qui sont une famille de modèles statistiques pour le traitement et la classification de données structurées, et en particulier les modèles de Markov cachés *(Hidden Markov Models)*. Ces derniers ont connu un succès important dans différents domaines d'application tels que le traitement de texte manuscrits [33, 5, 27], la reconnaissance de la parole ainsi qu'ils ont été intensivement utilisés pour l'analyse de séquences biologiques [48, 49]. Ils ont également été appliqués à d'autres domaines tels que la reconnaissance d'entités nommées [12], la reconnaissance d'images [7] et la modélisation d'un signal musical [9]. Récemment, ils ont été utilisés dans des applications d'extraction d'informations [40] en exploitant des données textuelles.

Ces modèles sont fortement similaires aux automates probabilistes, définis par une structure constituée d'états et de transitions dont chacune est accompagnée par une distribution de probabilité ainsi qu'un symbole. Les HMM se focalisent non pas sur les séquences d'états vu qu'elles représentent la couche cachée du modèle, mais sur les observations émises par ces états. Ils sont exploités pour modéliser ces séquences d'observations, citons à titre d'exemple la température qui aide à prévoir l'état météorologique, les acides nucléiques servant à modéliser une séquence d'ADN, les mots d'un corpus aidant à extraire des informations, etc.

Définition

La modélisation probabiliste d'un HMM requiert la définition des composants qui suivent [37] :

- Un ensemble d'états S
- Un ensemble de symboles d'observations O
- Une matrice de transition T où les cellules indiquent les probabilités de transition d'un état à l'autre.
- Une matrice d'émissions E où les cellules indiquent les probabilités de générations associées aux états : P(o|s) est la probabilité de générer (d'émettre) le symbole o à

partir de l'état s.

Problèmes des Modèles de Markov Cachés

Les fonctionnalités d'un HMM sont l'évaluation de la probabilité d'observation d'une séquence d'observations, l'apprentissage à partir d'un ensemble d'échantillons et la reconnaissance d'une séquence. De ce fait, l'utilisation des HMMs nécessite la résolution des trois problèmes fondamentaux :

- Calcul de la probabilité d'une observation à savoir sa vraisemblance par rapport à un HMM donné. En d'autres termes, c'est le calcul de la probabilité que cette séquence soit produite par le HMM en question. Il existe des algorithmes permettant de la calculer :
  - L'algorithme *Forward-Backward* [50] : fournit une solution exacte faisant intervenir tous les chemins dans le modèle du HMM.
  - L'algorithme *Viterbi* : fournit une solution approximative faisant intervenir uniquement le meilleur chemin (appelé chemin de *Viterbi*) dans le modèle du HMM.
- Reconnaître la séquence d'états la plus probable compte tenu d'une observation. Le calcul de cette probabilité de reconnaissance est souvent obtenu grâce à l'algorithme de *Viterbi*.
- Réajustement des paramètres du modèle du HMM pour maximiser la vraisemblance d'une séquence observée. Ce calcul est fait grâce à des approches basées sur des adaptations de l'algorithme EM (*Expectation-Maximisation*). Nous citons parmi ces approches :
  - Algorithme de *Baum-Welch* : ré-estime les probabilités de transition et d'émission en prenant en considération tous les chemins possibles du HMM.
  - Algorithme de *Viterbi* : réajuste les probabilités en tenant compte du meilleur chemin uniquement.

Le processus de génération d'une séquence d'observations repose d'une part sur le déplacement d'état en état, en partant d'un état initial, suivant les probabilités de tran-

sitions. D'autre part, il se base sur la génération d'un symbole sur chaque état rencontré et ce, en se profitant de la distribution de probabilités d'émission associée à l'état. Lorsque l'observation est émise, le modèle choisit une transition sortante en se basant sur la distribution de probabilités de transition affectée à l'état en question. Cette procédure sera refaite jusqu'à atteindre l'état final.

Les HMM sont exploités dans diverses catégories d'applications, parmi lesquelles nous mentionnons les problèmes de reconnaissance, que ce soit qu'il s'agit de la reconnaissance d'une classe précise telle que la reconnaissance d'une famille de protéines en partant d'une séquence d'acides aminés, ou encore la reconnaissance de la parole où il s'agit de définir un mot parmi un ensemble de mots en s'appuyant sur un signal audio ou une écriture manuscrite. D'un autre côté, les problèmes de segmentation de séquences se considèrent comme une deuxième catégorie d'applications qui s'intéresse au découpage d'une séquence en des sous séquences de types différents. Par exemple, spécifier les régions codantes et celles non codantes dans une chaîne de nucléotides, découper un signal musical en un ensemble de notes, etc. Les HMMs peuvent être utiles pour calculer le chemin ayant la probabilité maximale de générer la séquence de symboles en question.

Comme nous l'avons mentionné, un algorithme efficace permet d'effectuer ce calcul appelé *Forward-Backward*. Il est appelé *Forward* car il calcule sa variable *Forward* α_t(s)= P(o_1 o_2...o_t | s_t=s , H) d'une manière inductive, où l'induction est réalisée en avant, c'est-à-dire en partant du premier symbole jusqu'à atteindre la fin de la séquence. Les paramètres nécessaires sont les suivants :

- o_1 o_2..o_t est la séquence d'observations.
- H est le modèle HMM défini.
- s est l'état d'arrivée à l'instant t.

Bien que l'appellation *Backward* signifie le calcul à l'envers effectué, où il s'agit de déterminer la variable *Backward* β_t (s)= P(o_{t+1} o_{t+2}..o_T | s_t=s , H) et ce, en partant de l'état s en vue d'arriver à l'état final.
Tout ce processus se considère comme une préparation à ce qu'on vise de trouver. En

effet, ce qui nous préoccupe, étant donné un HMM et une séquence d'observations O, c'est de trouver la séquence d'états ayant la probabilité maximale de générer O. Plus précisément, l'objectif est de trouver un chemin appelé chemin de *Viterbi*, permettant de générer la séquence donnée avec cette probabilité. Ceci pourra être réalisé à l'aide de l'algorithme de *Viterbi* qui est un algorithme de programmation dynamique servant à résoudre ce problème d'une manière efficace. La variable exprimant cette idée est λ_t (s)= max P($s_1...s_t$, $o_1..o_t$ | H).

Avantages et inconvénients des Modèles de Markov Cachés

Comme tout modèle probabiliste, les HMMs possèdent des points forts ainsi que des points faibles, tout est relatif au domaine d'application. On peut citer parmi les avantages des HMMs :

- Le calcul de la probabilité d'appartenance d'une séquence à une classe dès les premières observations, ce qui donne l'opportunité d'éliminer les chemins qui ne sont pas prometteurs dès le début.
- Algorithmes d'apprentissage et de reconnaissance performants (Algorithmes de *Baum-Welch* et de *Viterbi*).
- Robuste pour la reconnaissance de l'écriture manuscrite où chaque caractère est représenté par un HMM et le mot n'est autre qu'une concaténation des HMMs des lettres.

Néanmoins, les HMMs présentent quelques limites qui se résument essentiellement en :

- La construction d'HMM pour l'apprentissage et la reconnaissance d'une seule classe donnée ce qui rend sa capacité de discrimination faible par rapport aux autres méthodes comme champs aléatoires conditionnels.
- La nécessité d'un très grand nombre d'échantillons lors de l'apprentissage pour converger vers des probabilités d'émission et de transitions permettant la bonne prise de décision concernant l'attribution d'un échantillon à une classe ou à une autre lors de la reconnaissance.

2.3.2 Modèles de Markov à Maximum Entropie (MEMM)

Nous avons expliqué dans la section précédente le principe du HMM comme étant une approche générative qui estime la distribution de P(S|O) par l'estimation de la probabilité P(S) et de la probabilité conditionnelle de génération des observations P(O|S). L'approche discriminante est considérée comme une alternative à l'approche générative qui consiste à estimer directement la distribution de la probabilité P(S|O). La différence primordiale c'est qu'elle ne nécessite pas l'estimation de la probabilité de génération des observations. Dans le cas de l'apprentissage automatique, l'approche discriminante consiste à apprendre un ensemble de fonctions discriminantes à partir des données d'apprentissage.

Dans ce cadre, les Modèles de Markov à Maximum Entropie *(Maximum Entropy Markov Models)* [30] sont apparus. Ils sont une variation des HMMs traditionnels comme étant un automate non déterministe à états finis. Certes, les HMMs ont donné de bons résultats dans diverses applications telles que le traitement de signal et ce, en faisant un bon choix des distributions de probabilité d'observations. Cependant, dans ce type d'applications, il est facile d'identifier ces distributions vu l'indépendance des attributs de données, à l'encontre du traitement linguistique qui utilise des attributs interdépendants. Les HMMs n'ont pas pris en compte la modélisation de ces dépendances, vu qu'elle est complexe et coûteuse en termes de ressources.

L'approche adoptée par les modèles MEMM est donc une alternative aux HMMs dans le sens où ils utilisent des probabilités conditionnelles à la place des probabilités jointes, afin d'exprimer ces dépendances en calculant la probabilité de l'état sachant les observations. Nous estimons donc pour chaque état et à un instant t la probabilité conditionnelle $P(s_t|s_{t-1},O_t)$. En effet, l'estimation des probabilités des observations semble inutile étant donné que les MEMM offrent la possibilité d'intégrer un ensemble de caractéristiques descriptives telles que la classe morphologique d'un mot, la casse, etc. Ce dernier est modélisé sous forme de fonctions caractéristiques, représentées par des couples <c,s> où c est la caractéristique observée et s l'état de sortie. Par exemple, on symbolise la relation entre la présence de majuscule et la classe Etudiant par le couple Is_Noun, Student.

À partir de ces descriptions, on construit un ensemble de fonctions caractéristiques F $=\{f_{<c,s>}(O_t,\, s_t)\}$ tel que :

$$f_{<c,s>}(O_t, s_t) = \begin{cases} 1 & \text{Si c}(O_t) \text{ est vrai et s}=s_t \\ 0 & \text{Sinon.} \end{cases}$$

Comme les HMMs, les MEMM ont été intensivement appliqués sur l'étiquetage grammatical [15] ainsi que sur des problèmes de traitement de texte. Dans ce cadre, nous mentionnons que les modèles HMM et MEMM ont été exploités dans diverses applications telles que la segmentation de texte [46]. Les résultats obtenus ont montré que les MEMM dominent les HMMs en termes de performance, ceci revient à l'utilisation de la notion de caractéristiques. En revanche, les résultats obtenus lors de l'application des HMMs dans les problèmes d'étiquetage de séquences ont été meilleurs que ceux du MEMM et ceci est le résultat direct du l'effet de biais. La démarche suivie par les MEMM favorise les séquences les plus fréquentes, ce qui fait que si le modèle possède un état à un seul successeur, l'observation n'aura aucun effet sur l'étiquetage. Cette lacune est réglée par les CRF.

2.3.3 Les champs aléatoires conditionnels (CRF)

Les champs aléatoires conditionnels, fondés par Lafferty et al. en 2001, ont ouvert de nouvelles portes pour l'analyse de séquences dans différents domaines. Ils se distinguent par leur structure de graphe non orienté qui permet de s'affranchir du problème soulevé par les MEMM, notamment l'effet du biais. L'orientation des arcs traduit les dépendances mutuelles entre les états, à l'encontre du HMM et du MEMM où l'orientation des arcs illustre la dépendance causale entre les états. De plus, les CRF permettent de modéliser des processus de nature discriminante, tels que l'annotation et ce, d'une manière plus claire.

Un CRF se distingue par X=x_1, x_2,.., x_T qui est une séquence de T observations et Y= y_1, y_2,.., y_T qui est une séquence de T étiquettes attribuées à la séquence X.

Étant donné ces paramètres, ce modèle définit une probabilité conditionnelle qui est la suivante :

$$P(Y|X) = 1/Z(X)\ \exp(\sum_{t=1}^{T}\sum_{i=1}^{F}\lambda_i f_i(y_{t-1}, y_t, x, t))$$

Tels que :

- Z : le facteur de normalisation utilisé pour rendre P(Y|X) une probabilité valide et il est égale à :

  $$exp(\sum_{t=1}^{T}\sum_{i=1}^{F}\lambda_i f_i(y_{t-1}, y_t, x, t))$$

  .

- λ_i : poids de la fonction $f_i()$, ce paramètre sera appris au cours de la phase d'apprentissage. Il reflète l'importance et la fiabilité de l'information apportée par la fonction binaire f_i. f_i est la notation générale des fonctions caractéristiques qui rendent compte chacune de l'occurrence d'une combinaison d'observations et d'étiquettes particulière. Étant donné E l'ensemble d'étiquettes possibles que peut prendre y_t, et O l'ensemble d'observations possibles que peut prendre x_t.

  $$f_i(y = e_i) = \begin{cases} 1 & \text{Si } y_t = e_i \text{ et } x_t = o_j \\ 0 & \text{Sinon.} \end{cases}$$

Ce modèle est discriminant, en d'autres termes, il relaxe l'hypothèse qu'une observation conditionnée par son étiquette soit indépendante des observations voisines. Les CRF ont été employés dans diverses tâches dans le domaine de traitement de langage naturel, notamment dans l'étiquetage morpho-syntaxique des phrases [25] et l'analyse de surface [41]. Ils sont également exploités pour l'extraction d'informations, notamment pour la recherche de séquences génomiques dans le domaine de la bioinformatique [31], ou encore pour l'identification des entités nommées relatives à des gènes [14]. Les CRF ont montré leur efficacité par rapport aux modèles génératifs employés auparavant dans ce type d'applications. Dans le cadre de la segmentation, les CRF ont été proposés pour segmenter des images couleurs de scènes réelles [17], comme ils sont exploités pour l'analyse et la reconnaissance de diagrammes manuscrits [43]. Zhu et al, ont suggéré un modèle CRF

dans le but de faire l'analyse et l'extraction d'informations dans les documents électroniques [51]. Récemment, les CRF ont été appliqués à la reconnaissance de l'écriture en ligne [2].

2.4 Conclusion

Les modèles probabilistes ont donné lieu à des approches de classification et d'étiquetage de séquences. Plus spécifiquement, les modèles génératifs offrent de bons résultats lorsqu'il s'agit de variables d'entrée facilement déterminables, à l'encontre des observations interdépendantes, où ils deviennent difficilement modélisables. Dans le but de remédier à ces lacunes, les modèles graphiques discriminants sont apparus, en ajoutant la possibilité de capturer les relations fortes entre les différentes variables. Les champs aléatoires conditionnels offrent une flexibilité considérable en incluant la notion de *features*, ce qui montre la supériorité de ce modèle par rapport aux autres modèles cités.

CHAPITRE 3

Une approche d'annotation sémantique des SW à base des modèles probabilistes

Sommaire

3.1 Introduction

Afin d'assurer une recherche rapide et efficace des services web dans le portail Bio-Catalogue [1], il serait utile de disposer d'annotations sémantiques décrivant la signature fonctionnelle de ces ressources. Pour cela, nous proposons une approche pour l'extraction automatique d'annotations sémantiques à partir des documentations textuelles des services web. Ceci consiste à étiqueter les descriptions textuelles de chaque service dans le but d'identifier ses paramètres principaux qui sont : le nom du service, les éléments

1. Le lien du site est : https ://www.biocatalogue.org/

d'entrée ainsi que de sortie et la tâche du service. Nous adoptons deux techniques probabilistes pour mettre en place une approche pour l'étiquetage sémantique.
Dans ce chapitre, nous présentons le principe général de l'approche proposée pour l'annotation sémantique à partir de textes associés aux Services Web (Section 3.2). Nous présentons ensuite l'adaptation des modèles probabilistes HMM et CRF pour la résolution de notre problème d'étiquetage sémantique.

3.2 Principe général de l'approche proposée

Notre travail se focalise sur les descriptions contenues dans BioCatalogue qui est un registre public, organisé et contrôlé, exposant les services web des sciences de la vie, en particulier de la bioinformatique [44]. Il offre une interface de recherche permettant aux experts de localiser des services Web de leur domaine. Il offre aussi aux fournisseurs la possibilité d'annoter les services par des termes libres qui permettent de décrire le service par un ensemble de méta-données facilitant sa localisation par les utilisateurs finaux. Cependant, cette annotation est réalisée manuellement. Elle est par conséquent souvent incomplète et peut donner lieu à des méta-données redondantes, orthographiquement incorrectes et parfois insignifiantes. Biocatalogue compte actuellement 2356 services, 186 fournisseurs et 776 membres [6].

Notre but est de faciliter la recherche et l'exploitation des services web dans cet annuaire. Pour ce faire, nous visons à extraire des annotations sémantiques à partir des descriptions textuelles des services web dans le domaine de la bioinformatique. Ceci se concrétise par l'assignation d'étiquettes ou annotations sémantiques décrivant le service web en question. Une étiquette est une information descriptive aidant à faciliter l'utilisation d'une ressource en lui attribuant du sens. Dans notre cas, il s'agit d'attribuer à chaque Service Web, une étiquette SN (nom de service), une étiquette IN (données d'entrée), une étiquette OU (données de sortie) ainsi qu'une étiquette ST (tâche du service), à partir de sa description textuelle et en se basant notamment sur une ontologie du domaine.

- L'étiquette *Service Name* (*SN*) désigne le nom du service en question. Cet élément figure en général au début de chaque description textuelle, il peut être uniterme tel que «*MView*», «*PRECISService*», etc. ou bien multi-termes tel que «*EMBOSS Backtranambig*», «*EMBOSS Sigscan*», etc.
- L'étiquette *Input* (*IN*) est attribuée aux concepts biologiques que le service exige comme paramètres d'entrée en vue de réaliser sa fonction. De même, l'étiquette IN peut être attribuée à des paramètres unitermes tel que «*peptides*», «*RNA*», etc. ou multi-termes tels que «*Chromosomol positions*», «*Protein sequences*», etc.
- L'étiquette *Output* (OU) est attribuée aux concepts biologiques que le service génère comme résultat après avoir achevé sa tâche. Les paramètres de sortie OU prennent éventuellement des valeurs uniterme tel que «*DNA*» ou encore multi-termes tels que «*Nucleic acid sequence*», «*Multiple sequence alignment*», etc.
- L'étiquette *Service Task* (*ST*) spécifie l'annotation faisant référence à la tâche primordiale du service web. Nous citons à titre d'exemple, le service web «*InterProScan*» qui adjoint diverses méthodes de reconnaissance des signatures de protéines en une seule, en vue de rechercher les domaines protéiques.

Les tableaux 3.1 et 3.2 présentent toutes les étiquettes sémantiques que nous avons définies. En effet, si l'annotation correspond à un concept multi-termes, nous nous servons d'un sous ensemble d'annotations, avec lequel nous pourrons fixer le début, le milieu et la fin de chaque type d'annotation.

Étiquettes appropriées aux Services Names	
SN	Nom de service uniterme
SN-B	Début d'un nom de service
SN-M	Milieu d'un nom de service
SN-E	Fin d'un nom de sevice
Étiquettes appropriées aux Input	
IN	Input uniterme
IN-B	Début d'un input
IN-M	Milieu d'un input
IN-E	Fin d'un input

Tableau 3.1 – Étiquettes appropriées à chaque type d'annotation (partie 1)

Étiquettes appropriées aux Output	
OU	Output uniterme
OU-B	Début d'un output
OU-M	Milieu d'un output
OU-E	Fin d'un output
Étiquettes appropriées aux Service Task	
ST	Tâche du service uniterme
ST-B	Début d'une tâche de service
ST-M	Milieu d'une tâche de service
ST-E	Fin d'une tâche de service
Étiquettes vides	
-	Absence d'annotation
NTN	Absence d'annotation en cas d'un nom
NTP	Absence d'annotation en cas d'une ponctuation
NTNP	Absence d'annotation en cas d'un nom propre
NTV	Absence d'annotation en cas d'un verbe

Tableau 3.2 – Étiquettes appropriées à chaque type d'annotation (partie 2)

Dans notre approche, l'annotation revient à un problème d'étiquetage de séquences, qui consiste à assigner à chaque mot de la description textuelle un tag sémantique parmi l'ensemble d'étiquettes sémantiques préalablement définies. Nous proposons pour cela d'adopter une technique d'étiquetage probabiliste de séquences. Le schéma de la figure 3.1 illustre la modélisation du processus d'étiquetage où l'étiqueteur probabiliste prend en entrée une suite de mots non annotés et génère leurs annotations adéquates.

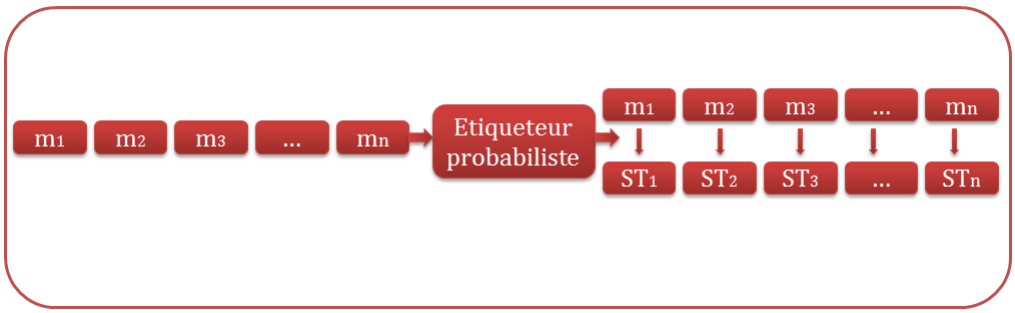

Figure 3.1 – Processus d'étiquetage

D'une manière formelle, cet étiquetage peut être modélisé comme suit : Disposons

d'une séquence d'observations $T = \{m_1, m_2, \ldots, m_n\}$ tel que, $\forall i$, $m_i \in V$ l'espace du langage, il faut trouver la séquence d'états ou encore la séquence d'annotations sémantiques ST = ST_1, ST_2,.., ST_n $\forall j$, $ST_j \in E$ qui est un ensemble fini d'états qui maximise la probabilité conditionnelle :

$$\widehat{ST} = \underset{ST}{argmax}\ \mathrm{P(ST|T)}$$

Pour cela, nous proposons l'application de deux modèles : le modèle de Markov caché qui est un modèle génératif, et les champs aléatoires conditionnels qui est un modèle discriminant. Nous avons choisi HMM comme étant un modèle classique relativement simple et très utilisé dans les problèmes d'étiquetage tels que l'étiquetage grammatical [24]. En effet, notre problème est analogue au problème d'étiquetage grammatical sauf que les étiquettes sont sémantiques, ce qui nous a poussé à tester son efficacité pour la résolution de notre problème.

3.3 Annotation à l'aide du modèle de Markov caché HMM

Le principe du HMM a été déjà élaboré dans le chapitre 2, on s'intéresse dans cette partie à l'application de ce modèle statistique à notre problème d'annotation sémantique. En effet, les HMMs, étant des modèles génératifs, ils reposent sur l'estimation de la probabilité de génération d'une *observation* à partir d'un corpus d'apprentissage et ce, en vue de donner la séquence d'*états* la plus probable. Cette estimation s'effectue en partant d'un état q_i donné sachant l'état qui le précède q_{i-1}. Une observation dans notre cas est représentée par le vecteur V comportant les trois éléments suivants :

- Le mot.
- La catégorie grammaticale du mot qui fait partie d'un ensemble composé de 28 catégories, désignées par leurs abréviations (nom propre NP, adjectif JJ, etc.).
- Le lemme du mot qui est la forme canonique du mot.

L'élément basique dans ce vecteur est la catégorie grammaticale, étant donné que la succession d'un nombre de catégories se considère significative comme observation.

D'une manière formelle, le modèle HMM peut être représenté à l'aide d'un automate à états finis comme suit :

- Q = $\{q_1,q_2,..,q_n\}$: l'ensemble des états qui sont les annotations sémantiques ST
- O = $\{o_1,o_2,..,o_n\}$: l'ensemble des observations qui sont les catégories grammaticales associées aux mots de T appelée CG.
- A : une matrice de transition telle que a_{ij} représente la probabilité de transition d'un état q_i à un état q_j, c'est à dire d'une annotation ST_i, à une annotation ST_j.
- B : une matrice d'émission tel que telle que b_{ij} exprime la probabilité qu'une observation (c'est à dire une catégorie grammaticale) soit générée par un état de ST (dans notre cas une annotation sémantique).

Nous présentons, dans le tableau 3.3, un exemple illustrant un extrait d'une description textuelle d'un service web appelé *EMBOSS Backtranambig*, servant à écrire une séquence d'acide nucléique à partir d'une séquence de protéines. Chaque mot de la description possède en ligne sa catégorie grammaticale, son lemme ainsi que son annotation sémantique.

Mot	**Catégorie grammaticale**	**Lemme**	**Annotation sémantique**
EMBOSS	NP (Nom Propre)	EMBOSS	SN-B
Backtranambig	NP	Backtranambig	SN-E
reads	VVZ (Verbe)	read	-
A	DT (Déterminant)	a	-
protein	NN (Nom)	protein	IN-B
sequence	NN	sequence	IN-E
And	CC (Coordination)	and	-
writes	VVZ	write	-
the	DT	the	-
nucleic	JJ (Adjectif)	nucleic	OU-B
Acid	JJ	acid	OU-M
sequence	NN	sequence	OU-E

Tableau 3.3 – Extrait du corpus d'apprentissage

Dans un modèle de Markov caché, la génération de la séquence d'annotations sémantiques la plus probable nécessite la détermination de paramètres de base qui sont : la matrice de

transition A et la matrice d'émission B et ce, en se basant sur un corpus d'apprentissage annoté manuellement. En vue de chercher cette suite d'annotations ST= (ST_1, ST_2, .., ST_k), on s'appuie essentiellement sur le calcul de la formule suivante :

$$P(ST|CG) = argmax_s tP(CG|ST).P(ST)[42] \tag{3.1}$$

avec ST désigne la séquence d'annotations sémantiques et CG la séquence d'observations qui est dans notre cas représentée par une suite de catégories grammaticales.
Pour ce faire, nous avons besoin de mesurer, d'une part, la probabilité P(CG|ST) qui est représentée par le produit de probabilités suivant :

$$P(CG|ST) = P(CG_1|ST_1).P(CG_2|ST_2)....P(CG_k|ST_k) \tag{3.2}$$

avec k est un entier fixé par l'utilisateur faisant référence aux nombres de mots d'une description ou plus. Chaque élément de ce produit est calculé comme suit :

$$\forall i \in \{1..k\} P(CG_i|ST_i) = Freq(CG_i, ST_i)/Freq(ST_i) \tag{3.3}$$

Ces fréquences d'apparition d'une annotation sémantique donnée (exemple : SN) associée à une catégorie grammaticale précise (exemple NP), sont déterminées à partir de notre corpus d'apprentissage. Un corpus d'apprentissage est un ensemble de descriptions textuelles qui est préalablement annoté de manière manuelle. Dans ce corpus,chaque mot dispose de son étiquette sémantique.
D'autre part, nous nous occupons de mesurer la probabilité P(ST) qui est égale à :

$$P(ST) = P(ST_1).P(ST_2|ST_1).P(ST_3|ST_1ST_2).....P(ST_k|ST_{k-1}ST_{k-2}) \tag{3.4}$$

avec

$$P(ST_i|ST_{i-1}ST_{i-2}) = Freq(ST_{i-2}ST_{i-1}ST_i)/Freq(ST_{i-2}ST_{i-1}) \tag{3.5}$$

Ces pré-calculs servent à construire la matrice de transitions A et la matrice d'émission B. Leurs probabilités sont estimées à partir de notre corpus d'apprentissage annoté

manuellement.

Le modèle HMM génère un nombre fini de séquences qui est de l'ordre de $|Etat|^{|Observations|}$. Ceci peut être schématisé à l'aide d'un réseau représentant toutes les séquences d'annotations possibles.

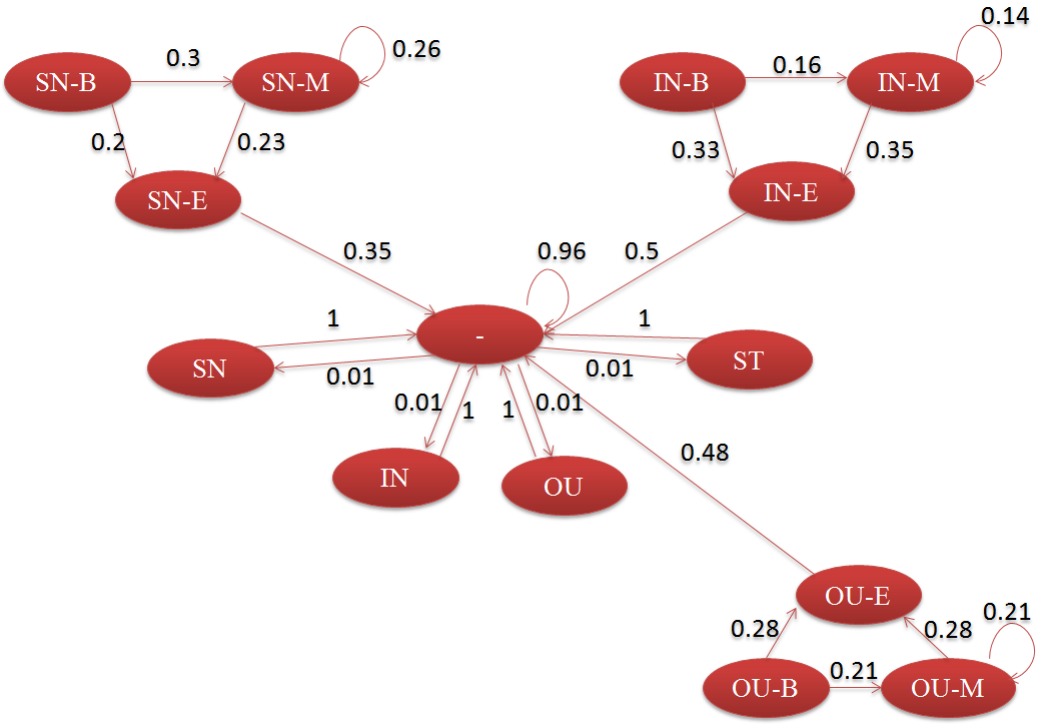

Figure 3.2 – Graphe des transitions dans un HMM

Pour une séquence d'observations donnée, l'algorithme de ***Viterbi*** [29] sert à identifier le meilleur chemin qui maximise sa probabilité de génération. En effet, cet algorithme dynamique se base essentiellement sur les hypothèses d'indépendance du HMM dans le but de retourner la succession d'états la plus probable. Il s'appuie sur un calcul itératif d'une probabilité d'observation $p(e_1..e_i)_q$, pour chaque état caché q du chemin qu'on cherche à trouver et ce, lorsqu'on l'atteint à une étape i du processus. Nous définissons cette probabilité comme suit :

$$P(e_1..e_i)_{q_l} = max(P(e_1..e_{i-1})_{q_k}.P(q_k;q_l).P(q_l;q_i)) \tag{3.6}$$

Si l'état suivant q_m aidera à maximiser la probabilité définie par l'équation 3.6, alors le chemin le plus probable atteignant q_l à l'étape i est composé de celui atteignant q_m à l'étape i-1 et de la transition q_m et q_l.

Nous décrivons le processus de la recherche du chemin le plus probable, illustré par l'algorithme 3.1, étant donnée la matrice de transitions et celle d'émissions déjà estimées.

Algorithme 3.1 Algorithme de *Viterbi*

Données

Entrée : HMM, $e_1..e_n$

Sortie : Le chemin le plus probable vitebi_path, pmax

Début

Initialiser le chemin et les probabilités

Initialiser la séquence d'observations E

Pour toute observation appartenant à E **Faire**

  Pour chaque état ql appartenant à l'ensemble d'états Q **Faire**

    Chercher le maximum de probabilité $P(e_1..e_{i-1})_{q_k}$.P(q_k ;q_l).P(q_l ;e_i)

    Mise à jour du chemin

    Mise à jour de la probabilité $P((e_i)_{q_l}$,$q_m)$

  Fin Pour

  Enregistrer le chemin trouvé

  Enregistrer la probabilité calculée

Fin Pour

Chercher le chemin le plus probable

Retourner le chemin le plus probable trouvé

Fin

Au cours d'une itération de cet algorithme, les divers chemins de même que leurs probabilités adéquates seront stockés et mis à jour et ce, dans le but d'en déduire à la fin le chemin le plus probable d'une séquence d'observations donnée.

En vue d'explorer la piste des modèles graphiques, nous avons choisi d'adapter un modèle discriminant à notre problème d'étiquetage de séquences. Nous présentons dans

la section 3.4 l'annotation sémantique des descriptions textuelles à l'aide des champs aléatoires conditionnels.

3.4 Annotation à l'aide du Conditional Random Fields

Comme nous l'avons déjà mentionné dans le chapitre 2, le modèle des champs aléatoires conditionnels appartient à la famille des modèles graphiques, comme les autres modèles d'étiquetage de séquences. Néanmoins, il existe une différence majeure entre le modèle de Markov caché comme étant un modèle génératif, et les champs aléatoires conditionnels comme étant un modèle discriminant et ce, puisque ces derniers se basent sur le calcul d'une probabilité conditionnelle P (Étiquettes | Observations), contrairement aux HMM définissant une probabilité jointe P (Observations, Étiquettes).

Par ailleurs, la méthode des CRF ne se limitent pas à l'observation courante mais prend en considération son **contexte d'apparition** qui peut inclure, entre autres, les observations précédentes et suivantes. De plus, les CRF permettent d'intégrer différentes structures informatives qui aident à mieux décrire les observations. Un enrichissement des observations offre au système la possibilité de bien identifier les régularités qui existent entre les observations et les états. En effet, des informations syntaxiques ou d'autres sémantiques peuvent être utilisés comme descriptions supplémentaires des mots à cet effet.

3.4.1 Formalisation du problème à l'aide des CRF

Notre approche a pour objectif d'extraire un ensemble d'annotations sémantiques à partir d'un corpus de descriptions textuelles de service Web. En effet, La méthode des champs aléatoires conditionnels (CRF) est une **approche contextuelle** qui se base sur l'étude des régularités entre une observation et son voisinage avec la valeur de son état. Ces régularités qui peuvent être syntaxiques, sémantiques ou grammaticales apparaissent régulièrement dans un même contexte. De ce fait, nous avons proposé d'enrichir les observations par deux types d'informations : morphosyntaxiques et sémantiques. Ainsi, étant donnée une observation O, qui est dans notre cas une description textuelle composée

de :

- Une suite de mots m_i
- Une suite d'étiquettes grammaticales CG_i associée aux mots de la description
- Une suite de valeurs booléennes indiquant l'appartenance d'un mot à un concept ontologique K_i

Le modèle CRF prédit l'état associé à cette observation qui est dans notre cas une annotation sémantique ST_i appartenant à l'ensemble d'annotations défini dans le tableau 3.1.

D'une manière formelle, étant donné un corpus d'apprentissage

D = $\left\{O^{(i)}, ST^{(i)}\right\}_{i=1}^{M}$ avec M est le nombre des descriptions textuelles dans D

Chaque $O^{(i)} = [O_1^{(i)}, O_2^{(i)},.., O_{Ti}{}^{(i)}]$ est la séquence d'observations préparée, avec T est le nombre de mots dans la description i. Une observation $O_i = <m_i, CG_i, K_i>$ tel que K_i correspond à une valeur booléenne indiquant si le mot m_i correspond bien à un concept d'une ontologie du domaine Ot. Nous utiliserons la chaine «onto-concept» pour indiquer que le mot m_i fait référence à un concept ontologique et la chaine «none» si ce n'est pas le cas. Chaque $ST^{(i)} = [ST_1^{(i)}, ST_2^{(i)},.., ST_{Ti}^{(i)}]$ est la séquence d'annotations sémantiques associées.

Nous présentons dans les tableaux 3.4 et 3.5 un extrait du corpus d'apprentissage utilisé :

Mot	**Catégorie grammaticale**	**Concept Ontologique**	**Annotation sémantique**
Get.Trop.gene.Dna. Sample.SPARQL. Service	NP	none	SN
takes	VVZ	none	-
as	IN (Préposition)	none	-
input	NN	none	-
the	DT	none	-
name	NN	none	IN-B
of	IN	none	IN-M

Tableau 3.4 – Extrait du corpus d'apprentissage du CRF (partie 1)

Mot	Catégorie grammaticale	Concept Ontologique	Annotation sémantique
a	DT	none	IN-M
RICE	NP	Onto-concept	IN-M
Genotyping	NN	Onto-concept	IN-M
study	NN	Onto-concept	IN-E
and	CC	none	-
returns	VVZ	none	-
names	NNS (nom au pluriel)	none	OU-B
of	IN	none	OU-M
the	DT	none	OU-M
corresponding	JJ (adjectif)	none	OU-M
DNA	NN	Onto-concept	OU-M
samples	NNS	Onto-concept	OU-E

Tableau 3.5 – Extrait du corpus d'apprentissage du CRF (partie 2)

Contrairement à l'approche basée sur HMM, les CRF permettent d'intégrer différents types d'informations afin de mieux représenter les observations. À cet effet, en plus de la classe morphosyntaxique et sémantique du mot, les CRF considérent aussi les contextes gauches et droites, c'est à dire les observations précédentes et suivantes. Étant donnée la description textuelle de la figure 3.3 d'un service Web nommé «*Geneslico SOAP Service*», étiquetée grammaticalement et annotée par les étiquettes sémantiques appropriées :

Geneslico	SOAP	Service	for	RNA	Secondary	Structure	Prediction
NP	NP	NP	IN	NP	JJ	NN	NN
none	none	none	none	Onto-concept	Onto-concept	Onto-concept	Onto-concept
SN-B	SN-M	SN-E	-	ST-B	ST-M	ST-M	ST-E

Figure 3.3 – Extrait d'une description textuelle annotée

Nous présentons dans la figure 3.4, le graphe biparti associé au modèle CRF. Les régularités du contexte observable sont définies par un ensemble de fonctions caractéristiques f_i qui prennent en paramètre le voisinage C(t) de l'état à l'instant t. En effet, si nous considérons une observation o_t à la position t de la séquence d'observations O, et st_t

l'annotation sémantique à la position t de la séquence d'annotations ST, notre but est de trouver le voisinage C(t) qui maximise la probabilité p(ST|O). La phase d'apprentissage d'un modèle CRF utilise un ensemble d'exemples étiquetés pour trouver la meilleure pondération λ pour chaque fonction caractéristique f_i.

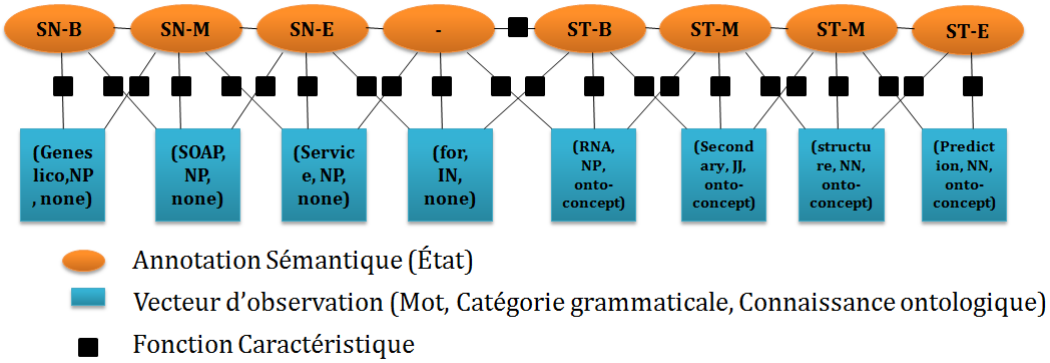

Figure 3.4 – Modélisation d'un exemple en un CRF

Parmi les fonctions caractéristiques générées dans cet exemple, nous décrivons les suivantes. Étant donnée la position 6 dans la séquence d'annotations sémantiques (ST-M), le modèle prendra compte de la valeur des observations i, i-2, i-1, i+1 et i+2 et nous obtenons la fonction suivante :

$$f1(ST_i, O_i) = \begin{cases} 1 & \text{Si } ST_i\text{= ST-M et } ST_i\text{-1= ST-B et } O_i\text{=(Secondary, JJ, onto-concept)} \\ & \text{et } O_{i-2}\text{=(for, IN, none) et } O_{i-1}\text{=(RNA, NP, onto-concept)} \\ & \text{et } O_{i+1}\text{=(structure, NN, onto-concept)} \\ & \text{et } O_{i+2}\text{= (Prediction, NN, onto-concept)} \\ 0 & \text{Sinon.} \end{cases}$$

3.4.2 Processus d'annotation

L'objectif de notre approche est d'évaluer la pertinence des champs aléatoires conditionnels dans le contexte de l'étiquetage de séquences, et plus précisément les séquences de mots contenues dans les descriptions textuelles des services Web. Les expérimenta-

tions que nous menons pour tester l'efficacité de ce modèle dans l'annotation sémantique mettent en œuvre une procédure souvent employée pour évaluer les techniques d'apprentissage. Au premier abord, nous faisons usage d'une séquence d'apprentissage dans le but de former le modèle probabiliste qui sera testé par la suite sur un autre corpus appelé corpus de test en vue de produire les étiquettes sémantiques associées à la nouvelle séquence d'observations. Au cours des deux phases, il y aura une définition des attributs de la séquence d'observations qui seront utilisés par la suite à la généralisation des fonctions caractéristiques (voir figure 3.5).

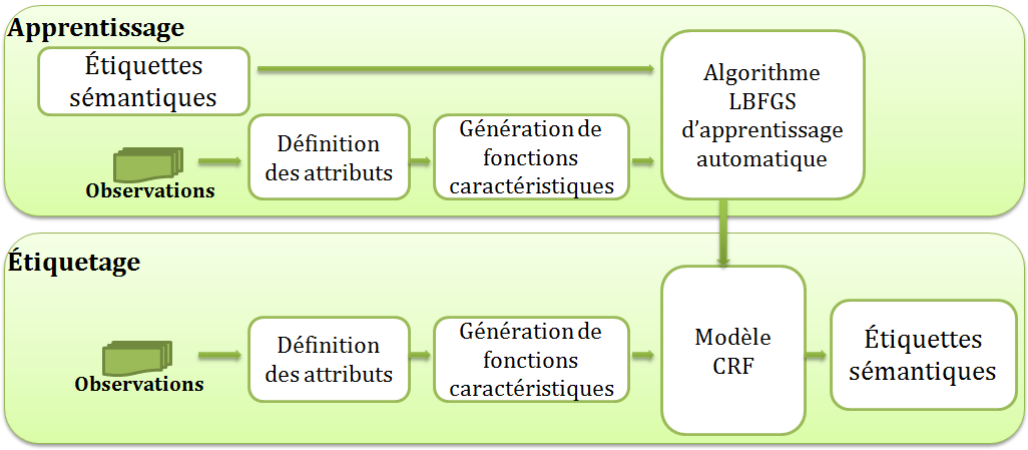

Figure 3.5 – Processus d'annotation sémantique par CRF (inspiré de [52])

3.5 Conclusion

Dans ce chapitre nous avons présenté notre approche d'annotation sémantique des SW qui s'appuie sur l'apprentissage automatique. Nous avons proposé une solution visant à étiqueter des descriptions textuelles et ce, en adaptant les modèles probabilistes HMM et CRF au problème d'annotation sémantique. Les modèles génératifs tel que HMM donnent des résultats acceptables, néanmoins la valeur ajoutée du modèle CRF, qui était la sélection automatique d'un ensemble de caractéristiques pertinentes, s'avère un atout intéressant à citer par rapport à HMM. Notamment, les CRF permettent de capturer des relations fortes entre les variables. Nous avons essayé de mettre en valeur l'ampleur

de ce modèle discriminant qui a donné de bons résultats dans la tâche d'annotation.

CHAPITRE 4

Expérimentations et analyse des résultats

Sommaire

4.1 Introduction

L'objectif de ce chapitre est de valider expérimentalement notre approche automatique d'annotation sémantique de services web. Nous discutons aussi dans ce chapitre les résultats de nos expérimentations en montrant que l'approche proposée corrobore les lacunes déjà citées dans l'état de l'art.

Ce chapitre se structure de la façon suivante. Dans la section 4.2, nous décrivons le corpus de descriptions textuelles utilisé. Dans la section 4.3, nous présentons d'une part l'environnement d'expérimentation, et d'autre part les détails d'implémentation des paramètres du modèle HMM. La section 4.4 présente les expérimentations réalisées pour tester le modèle CRF.

4.2 Corpus de travail

Pour cette expérience, il s'agit d'annoter les descriptions textuelles des SW. Cette tâche est formalisée comme étant une tâche d'étiquetage de séquences. Pour ce faire, nous faisons usage d'un corpus de descriptions offert par Biocatalogue qui est divisé en deux :

- Un corpus d'apprentissage étiqueté grammaticalement, lemmatisé pour l'expérience du HMM et enrichi par la connaissance ontologique pour l'expérience du CRF, ainsi qu'il est étiqueté par les annotations sémantiques déjà citées. Ce corpus comprend 100 descriptions textuelles.
- Un corpus de test composé de 173 descriptions textuelles.

Le corpus d'apprentissage a été annoté manuellement par des experts du domaine. Nous possédons donc une référence associant à chaque mot son annotation sémantique adéquate. Ce corpus est supposé parfait, il servira ainsi à la phase d'apprentissage. Le nombre de colonnes doit être fixe dès le début, chaque mot est représenté sur une ligne ayant ses propres colonnes séparées par un espace. Les descriptions textuelles y contenues sont séparées par une ligne vide.

4.3 Environnement de test du modèle de Markov caché HMM

Suite à la description du principe du HMM qu'on a abordé dans les chapitres précédents, on s'intéresse dans ce qui suit à présenter les résultats obtenus lors de son application sous l'environnement MATLAB.

4.3.1 Outil utilisé

MATLAB [28] est une abréviation de *MATrix LABoratory* développé à l'origine en *Fortran* par *C.Moler*. La version actuelle est écrite en C par *Mathworks Inc.*, elle met à disposition un environnement complet, extensible, facile à utiliser, qui fournit un système interactif intégrant le calcul numérique ainsi que la visualisation. Cet environnement dispose de diverses fonctions mathématiques, scientifiques ainsi que techniques. L'atout de l'approche de MATLAB est qu'elle offre la possibilité d'exploiter les algorithmes des bibliothèques de fonctions spécialisées (*Toolboxes*) et même leur modification ainsi que l'ajout d'autres fonctions. La possibilité d'analyser nos données, le développement d'algorithmes, non moins que la création de modèles, nous ont incité à choisir MATLAB comme outil performant pour l'application de la première partie de notre approche.

4.3.2 Calcul des paramètres

Afin d'obtenir la séquence d'annotations sémantiques la plus probable, nous nous servons des fonctionnalités de MATLAB et ce, en suivant les phases suivantes. Tout d'abord, nous devons manipuler les données contenues dans le corpus d'apprentissage. Pour ce faire, nous importons nos données, en respectant leur format, dans un tableau de cellule nommé *Cell Array*. Ce dernier est un type de données disposant de conteneurs indexés appelés cellules pouvant comporter n'importe quel type de données (données numériques, chaînes de caractères, ou encore la combinaison des deux). Cette importation s'effectue à l'aide de la fonction suivante :

- $Cell\_matrix = textscan(fileID, formatSpec)$ permettant de lire des données à partir d'un fichier texte et les affecter à un *Cell Array*. Le fichier texte est désigné à travers l'identificateur de fichiers *fileID*, et les champs de données à lire sont spécifiés à l'aide de *formatSpec*.

Après avoir importé notre corpus d'apprentissage, nous avons calculé la matrice de transition et celle d'émission et ce de deux manières :

$$[Transition\_Matrix, Emission\_Matrix] = hmmestimate(Sequence, States)$$

La première façon consiste à utiliser la fonction *hmmestimate*() [20] qui calcule le maximum de vraisemblance estimée des probabilités de transition ainsi que d'émission et ce, à partir d'une séquence d'observations *Sequence* et d'une séquence d'états, ou encore d'annotations *States*, déjà connus. Cette fonction génère deux matrices :

- Une matrice de transition *Transition_ Matrix* ayant les états, qui sont les annotations sémantiques (21 annotations), en lignes et en colonnes. Autrement dit, chaque cellule contient la probabilité de transition entre deux états.
- Une matrice d'émission *Emission_ Matrix* ayant les états en lignes et les observations qui sont les catégories grammaticales (27 catégories) en colonnes. Autrement dit, chaque cellule contient la probabilité d'apparition d'une annotation i avec une catégorie j.

Nous pouvons aussi utiliser la fonction *hmmtrain*() qui, suite à l'introduction d'une séquence d'observations, une estimation initiale de la matrice de transition ainsi qu'une estimation initiale de la matrice d'émissions, estime les probabilités de transitions et d'émissions d'un HMM en s'appuyant sur l'algorithme *Baum-Welch* [16]. Les paramètres de cette fonction se présentent comme suit :

$$[Est\_Trans, Est\_Emis] = hmmtrain\ (Sequence, Tr\_Guess, Emiss\_Guess)$$

La deuxième manière repose sur l'écriture d'un programme C qui s'intéresse à parcourir le corpus d'apprentissage et d'en déduire les matrices demandées, et ce pour avoir des résultats plus précis. Les matrices obtenues lors de l'application de cette méthode seront analysées dans la sous section 4.3.3.

4.3.3 Analyse de la matrice de transitions

On présente la matrice de transition illustrée par les deux figures 4.1 et 4.2 :

	SN	IN	OU	-	NTN	NTV	NTNP	NTP	SN-B	SN-M	SN-E
SN	0	0	0	0,566265	0,108434	0,168675	0,156627	0	0	0	0
IN	0	0	0	0,740741	0,148148	0,037037	0,074074	0	0	0	0
OU	0	0	0	0,857143	0	0,142857	0	0	0	0	0
-	0,008336	0,006946	0,002779	0,409558	0,316755	0,102528	0,118366	0,000556	0,001111	0	0
NTN	0,024903	0,001107	0	0,630327	0,236857	0,074156	0,024903	0	0,004981	0	0
NTV	0	0	0,006757	0,712838	0,163851	0,052365	0,032095	0	0	0	0
NTNP	0,009547	0	0	0,440334	0,126492	0,03222	0,381862	0,005967	0,002387	0	0
NTP	0	0	0	0,333333	0	0,111111	0,555556	0	0	0	0
SN-B	0	0	0	0	0	0	0	0	0	0,3	0,2
SN-M	0	0	0	0	0	0	0	0	0	0,263158	0,236842
SN-E	0	0	0	0,25	0,03125	0,0625	0,125	0	0	0	0
IN-B	0	0	0	0	0	0	0	0	0	0	0
IN-M	0	0	0	0	0	0	0	0	0	0	0
IN-E	0	0	0	0,361538	0,046154	0,046154	0,030769	0,015385	0	0	0
OU-B	0	0	0	0	0	0	0	0	0	0	0
OU-M	0	0	0	0	0	0	0	0	0	0	0
OU-E	0	0	0	0,404762	0,059524	0,035714	0	0	0	0	0
ST	0	0	0	1	0	0	0	0	0	0	0
ST-B	0	0	0	0,0606	0	0,0303	0	0	0	0	0
ST-M	0	0	0	0	0	0	0	0	0	0	0
ST-E	0	0	0	0	0,3333	0	0	0	0	0	0

Figure 4.1 – Matrice de transition (partie 1)

	IN-B	IN-M	IN-E	OU-B	OU-M	OU-E	ST	ST-B	ST-M	ST-E
SN	0	0	0	0	0	0	0	0	0	0
IN	0	0	0	0	0	0	0	0	0	0
OU	0	0	0	0	0	0	0	0	0	0
-	0,015282	0	0	0,009447	0	0	0	0,007224	0	0
NTN	0,001107	0	0	0	0	0	0,000553	0,000553	0	0,000553
NTV	0,013514	0	0	0,013514	0	0	0	0,005068	0	0
NTNP	0	0	0	0	0	0	0	0,001193	0	0
NTP	0	0	0	0	0	0	0	0	0	0
SN-B	0	0	0	0	0	0	0	0	0	0
SN-M	0	0	0	0	0	0	0	0	0	0
SN-E	0	0	0	0	0	0	0	0,03125	0	0
IN-B	0	0,161538	0,330769	0	0	0	0	0	0	0
IN-M	0	0,145161	0,354839	0	0	0	0	0	0	0
IN-E	0	0	0	0	0	0	0	0	0	0
OU-B	0	0	0	0	0,214286	0,285714	0	0	0	0
OU-M	0	0	0	0	0,21875	0,28125	0	0	0	0
OU-E	0	0	0	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0
ST-B	0	0	0	0	0	0	0	0	0,5454	0,3636
ST-M	0	0	0	0	0	0	0	0,0454	0,1818	0,7727
ST-E	0	0	0	0	0	0	0	0	0	0

Figure 4.2 – Matrice de transition (partie 2)

Les probabilités de transition entre les états varient entre 0 et 1. Les valeurs comprises entre 0 et 0.3, qui sont accompagnées d'une puce rouge dans les figures 4.1 et 4.2, désignent des probabilités faibles dont les transitions adéquates s'avèrent peu fréquentes. Autrement dit, nous parlons d'une transition qui apparait rarement dans le corpus d'apprentissage. Nous citons à titre d'exemple, la succession suivante d'annotations :

Les valeurs comprises entre 0.3 et 0.5, qui sont accompagnées d'une puce jaune dans

les figures 4.1 et 4.2, désignent des probabilités moyennes dont les transitions adéquates s'avèrent moyennement fréquentes. Nous citons à titre d'exemple, la succession suivante d'annotations :

- IN-B, IN-E : cette succession apparait d'une manière assez fréquente, sa probabilité est valorisée de 0,330769.

Les valeurs comprises entre 0.5 et 1, qui sont accompagnées d'une puce verte dans les figures 4.1 et 4.2 , désignent des probabilités fortes dont les transitions adéquates s'avèrent fréquentes. Autrement dit, nous parlons d'une transition qui apparait fréquemment dans le corpus d'apprentissage. Nous citons à titre d'exemple, la succession suivante d'annotations :

- ST-M, ST-E : cette succession apparait fréquemment dans le corpus d'apprentissage, sa probabilité est valorisée de 0,7727
- ST,- : cette succession apparait toujours, c'est-à-dire suite à chaque annotation ST, on trouve toujours une annotation no-tag « - ». La probabilité de cette succession est estimée de 1.

4.3.4 Analyse de la matrice d'émissions

On présente la matrice d'émission illustrée par les deux figures 4.3 et 4.4 :

	NP	VVZ	NN	DT	CC	JJ	PP	MD	VH	VVN	IN	VVG	NNS	WDT
SN	0,771084	0	0,228916	0	0	0,012048	0	0	0	0	0	0	0,024096	0
IN	0,148148	0	0,481481	0	0	0,037037	0	0	0	0	0	0	0,333333	0
OU	0	0,071429	0,571429	0	0	0,071429	0	0	0	0	0	0	0,285714	0
-	0,285913	0	0	0,179772	0,057516	0,146152	0,015838	0,016393	0	0	0,183384	0	0	0,008614
NTN	0	0	0,753735	0	0	0	0	0	0	0	0	0	0,246265	0
NTV	0	0,182432	0	0	0	0	0	0	0,005068	0,317568	0	0,182432	0	0
NTNP	0,997613	0	0	0	0	0	0	0	0	0	0	0	0	0
NTP	1	0	0	0	0	0	0	0	0	0	0	0	0	0
SN-B	0,866667	0	0	0,133333	0	0	0	0	0	0	0	0	0	0
SN-M	0,842105	0	0	0	0,052632	0,052632	0	0	0	0	0,052632	0	0	0
SN-E	0,625	0	0,25	0	0	0	0	0	0	0	0	0	0	0
IN-B	0,215385	0	0,615385	0	0	0,138462	0	0	0	0,015385	0	0	0	0
IN-M	0,225806	0	0,354839	0,064516	0,032258	0,129032	0	0	0	0	0,129032	0	0,064516	0
IN-E	0,092308	0	0,584615	0	0	0	0	0	0	0	0	0	0,323077	0
OU-B	0,095238	0	0,404762	0	0	0,333333	0	0	0	0,02381	0	0	0,119048	0
OU-M	0,125	0	0,375	0,03125	0	0,125	0	0	0	0,03125	0,28125	0,03125	0	0
OU-E	0	0	0,571429	0	0	0,02381	0	0	0	0	0	0	0,404762	0
ST	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ST-B	0,212121	0	0,424242	0	0	0,30303	0	0	0	0,030303	0	0	0,030303	0
ST-M	0,045455	0	0,772727	0	0	0,136364	0	0	0	0,045455	0	0	0	0
ST-E	0,033333	0,033333	0,766667	0	0	0	0	0	0	0	0	0,033333	0,133333	0

Figure 4.3 – Matrice d'émission (partie 1)

	VVP	VHZ	TO	CD	RP	RB	WRB	FW	WF	NPS	VBP	VV	Symbôle
SN	0	0	0	0	0	0	0	0	0	0	0	0	0
IN	0	0	0	0	0	0	0	0	0	0	0	0	0
OU	0	0	0	0	0	0	0	0	0	0	0	0	0
-	0,00778	0	0,036121	0,030008	0,000278	0,029453	0,001945	0,000834	0	0	0	0	0,0733
NTN	0	0	0	0	0	0	0	0	0	0	0	0	0
NTV	0,001689	0,001689	0	0	0	0	0	0	0	0	0,070946	0,238176	0
NTNP	0	0	0	0	0	0	0	0	0	0,002387	0	0	0
NTP	0	0	0	0	0	0	0	0	0	0	0	0	0
SN-B	0	0	0	0	0	0	0	0	0	0	0	0	0
SN-M	0	0	0	0	0	0	0	0	0	0	0	0	0
SN-E	0	0	0	0	0	0	0	0	0	0,125	0	0	0
IN-B	0	0	0	0,015385	0	0	0	0	0	0	0	0	0
IN-M	0	0	0	0	0	0	0	0	0	0	0	0	0
IN-E	0	0	0	0	0	0	0	0	0	0	0	0	0
OU-B	0	0	0	0,02381	0	0	0	0	0	0	0	0	0
OU-M	0	0	0	0	0	0	0	0	0	0	0	0	0
OU-E	0	0	0	0	0	0	0	0	0	0	0	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0
ST-B	0	0	0	0	0	0	0	0	0	0	0	0	0
ST-M	0	0	0	0	0	0	0	0	0	0	0	0	0
ST-E	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.4 – Matrice d'émission (partie 2)

Les probabilités d'émission varient entre 0 et 1. Les valeurs comprises entre 0 et 0.3, qui sont accompagnées d'une flèche rouge dans les figures 4.3 et 4.4, désignent des probabilités faibles dont l'apparition d'une annotation (en ligne) conjointement avec une catégorie grammaticale donnée, s'avère assez fréquente. Autrement dit, nous parlons d'un couple (annotation, catégorie grammaticale) qui apparait rarement dans le corpus d'apprentissage. Nous citons à titre d'exemple, le couple suivant :

- (SN, VVZ) : ce couple ne figure pas dans le corpus, nous ne pouvons pas avoir un nom de service ayant une catégorie grammaticale VVZ. La probabilité adéquate est valorisée de 0.

Les valeurs comprises entre 0.3 et 0.5, qui sont accompagnées d'une flèche jaune dans les figures 4.3 et 4.4, désignent des probabilités moyennes dont le couple (annotation, catégorie grammaticale) s'avère moyennement fréquent. Nous citons à titre d'exemple, le couple suivant :

- (SN-M, JJ) : une étiquette SN-M est rarement catégorisée par un adjectif, la probabilité de ce couple est valorisée de 0,052632.
- (OU, NN) : une étiquette OU est souvent catégorisée par un nom, la probabilité de ce couple est valorisée de 0,571429.

Les valeurs comprises entre 0.5 et 1, qui sont accompagnées d'une flèche verte dans les figures 4.3 et 4.4, désignent des probabilités fortes dont le couple (annotation, catégorie grammaticale) s'avère fréquent. Autrement dit, nous parlons d'un couple qui apparait fréquemment dans le corpus d'apprentissage. Nous citons à titre d'exemple, le couple suivant :

- (SN, NP) : ce couple apparait fréquemment dans le corpus d'apprentissage indiquant qu'un nom de service possède fréquemment la catégorie grammaticale du Nom Propre. La probabilité est estimée de 0,771084.
- (ST-E, NN) : ce couple apparait d'une manière fréquente, c'est-à-dire une annotation qui indique la tâche du service est généralement catégorisée par un nom. La probabilité dans ce cas est estimée de 0,76667.

Après avoir déterminé les deux matrices, nous appliquons l'algorithme de *Viterbi* en vue d'obtenir la séquence d'annotations sémantiques la plus probable étant donnée une séuence d'observations. Cet algorithme requiert les paramètres d'entrée suivants :

- La séquence de catégories grammaticales à partir du corpus de test à annoter (séquence d'observations). Ce corpus est organisé sous un format bien déterminé, il possède trois colonnes : le mot, sa catégorie grammaticale ainsi que son lemme. Il comporte 10524 lignes.
- La matrice de transition pré-calculée à partir du corpus d'apprentissage.
- La matrice d'émission pré-calculée à partir du corpus d'apprentissage.

Nous introduisons une séquence de test composée d'une suite de catégories grammaticales. Pour chaque catégorie grammaticale CG_i de la séquence d'observations introduite, le programme de *Viterbi* sous MATLAB calcule la probabilité

$$P(CG_1..CG_{i-1}).P(ST_k, ST_l).P(ST_l, CG_i) \tag{4.1}$$

et choisit celle qui est maximale. Ensuite, il effectue une mise à jour de la probabilité ainsi que du chemin le plus probable formé. Pour ce faire, nous nous servons de la fonction ***hmmviterbi()*** et ce, en introduisant les paramètres demandés :

STATES = ***hmmviterbi*** (*Sequence*, *Transition_ Matrix*, *Emission_ Matrix)*

4.3.5 Évaluation des résultats d'annotation basée sur HMM

L'évaluation de notre approche a pour objectif de mesurer l'écart entre une annotation de référence, autrement dit celle qui existe dans un corpus de test annoté manuellement par un expert, et l'annotation résultante de l'application du modèle probabiliste en question. Pour ce faire, nous nous profitons des mesures suivantes : Précision et Rappel [38].

La *Précision* représente le taux des mots correctement annotés, après avoir appliqué le modèle, pour l'annotation sémantique a_i sur la totalité des annotations trouvées pour le même type. Cette mesure évalue la capacité du modèle utilisé à fournir le plus d'annotations correctes.

$$\textbf{Précision}_i = \frac{\textbf{(Nombre de mots correctement annotés)}_i}{\textbf{(Nombre de mots annotés)}_i}$$

Une autre mesure est considérablement utilisée qui est le *rappel*. Ce dernier représente le taux de mots correctement annotés par le système pour l'annotation i sur la totalité des annotations du même type selon la référence. Cette mesure évalue la capacité du modèle à fournir un nombre d'annotations correctes.

$$\textbf{Rappel}_i = \frac{\textbf{(Nombre de mots correctement annotés)}_i}{\textbf{(Nombre de mots annotés selon la référence)}_i}$$

Nous présentons le compromis entre les deux mesures présentées ci-dessus, en les combinant en une mesure unique assurée par la moyenne harmonique F-mesure.

$$\textbf{F-mesure} = \frac{\textbf{(1 + } \boldsymbol{\beta}^2\textbf{) * (Précision * Rappel)}}{\textbf{(}\boldsymbol{\beta}^2 \textbf{ * (Précision + Rappel)}}$$

Le coefficient $\boldsymbol{\beta} \in R^+$ est un paramètre de pondération qui est dans notre cas fixé à 1, alors la moyenne harmonique est :

$$\textbf{F-mesure} = \frac{\textbf{2 * (Précision * Rappel)}}{\textbf{(Précision + Rappel)}}$$

Nous présentons ci-dessous un tableau récapitulatif (voir tableau 4.1) illustrant les résultats obtenus lors de l'application du modèle de Markov caché.

Annotations	Précision	Rappel	F-mesure
SN	0,01346104	0,43697479	0,02611753
IN	0	0	0
OU	0,07058824	0,42857143	0,12121212
ST	0	0	0
-	0,78392283	0,62910717	0,69803398
NTN	0,64924623	0,39033233	0,48754717
NTP	0	0	0
NTV	0,63731343	0,91044776	0,74978051
NTNP	0,44444444	0,00404449	0,00801603

Tableau 4.1 – Évaluation de l'annotation avec HMM

En se basant sur les précisions des différents types d'annotations sémantiques, nous remarquons qu'en appliquant le modèle génératif HMM, nous avons obtenus des résultats de faible précision. Ceci revient à l'incapacité d'un HMM d'ordre 1 où la probabilité d'observation à un état ne prend en compte que l'état courant et l'état précédent, d'affecter correctement l'annotation sémantique au mot de la séquence de test. En outre, ce type de modèle ne s'appuie que sur une seule catégorie d'observation, ce qui empêche d'exploiter toutes les observations disponibles dans le corpus d'apprentissage. Parmi les autres causes de la faible précision, nous mentionnons que la qualité du corpus d'apprentissage a aussi influencé, vu que l'écart entre les fréquences des différentes annotations est élevé (par exemple la fréquence de SN est égale à 64 et la fréquence de no-tag est égale à 3599). La probabilité de transition entre les états (no-tag, no-tag) est beaucoup plus supérieure aux autres probabilités ce qui favorise par la suite l'assignation massive de cette annotation. La valeur du rappel est beaucoup plus élevée que celle de précision et ce car le nombre d'une annotation ai généré par ce modèle (annotation correcte et non correcte) est proportionnel au nombre de ai dans la référence (le corpus de test annoté manuellement).

Pour mieux visualiser l'écart entre les précisions de chaque type d'annotation ainsi que leurs rappels, nous présentons les différentes valeurs à l'aide d'un histogramme (voir

figure 4.5).

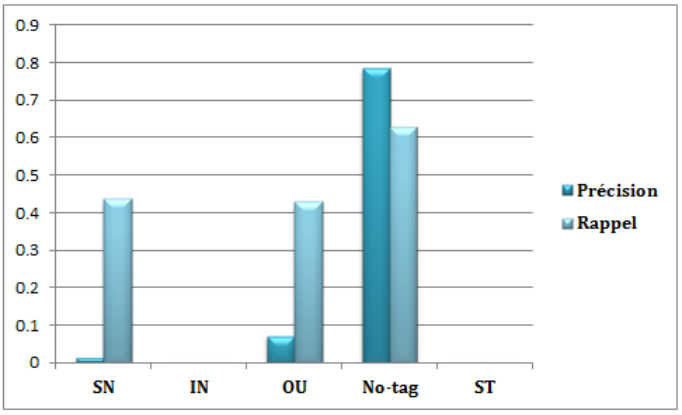

Figure 4.5 – Histogramme illustratif des résultats obtenus en appliquant HMM

4.4 Environnement de test du modèle des champs aléatoires conditionnels CRF

Suite à la description du principe du CRF qu'on a abordé dans les chapitres précédents, on s'intéresse dans ce qui suit à tracer son utilisation ainsi que les résultats obtenus lors de son application sous le logiciel CRF++.

4.4.1 Outil utilisé

CRF++ [13] est une implémentation simple et personnalisée du modèle probabiliste CRF en vue d'étiqueter des données séquentielles. Il est appliqué sur diverses tâches du domaine de traitement de langages naturels telles que la reconnaissance d'entités nommées et l'extraction d'informations. Cette implémentation est écrite en langage C++ comme elle se base sur l'algorithme LBFGS (*Limited-memory Broyden-Fletcher-Goldfarb-Shanno*) [26] qui favorise un apprentissage rapide. CRF++ se caractérise par l'utilisation réduite de la mémoire pour les deux phases d'apprentissage et de test et par le temps pratique de la phase du décodage.

4.4.2 Évaluation des résultats d'annotation basée sur CRF

Le logiciel CRF++, que nous utilisons, est fondé sur le modèle discriminant CRF. À partir des échantillons de descriptions textuelles fournies par le corus d'apprentissage, il définit automatiquement les fonctions caractéristiques, qui sont des fonctions booléennes. Pour ce faire, il spécifie les positions de chaque mot du corpus par la notation %x[ligne, colonne]. Nous présentons dans la figure 4.6 un extrait de notre corpus d'apprentissage.

Input	Data			
EMBOSS	NP	EMBOSS	SN-B	
Sixpack	NP	Sixpack	SN-E	
reads	VVZ	read	-	
a	DT	a	-	<< mot courant
DNA	NN	DNA	IN-B	
sequence	NN	sequence	IN-E	
and	CC	and	-	

Figure 4.6 – Extrait du corpus

La figure 4.7 illustre la définition de l'emplacement de chaque unité lexicale dans l'exemple de la figure 4.6.

Position	Caractéristique
%x[0,0]	a
%x[0,1]	DT
%x[-1,0]	reads
%x[-2,1]	NP
%x[0,0]/%x[0,1]	a/DT

Figure 4.7 – Position des mots dans le corpus

Notamment, la phase d'apprentissage s'est effectuée en 53.33 secondes avec 0.44 secondes pour la lecture des données à partir de notre corpus d'apprentissage qui comporte 7385 lignes. Les informations qui concernent cette phase sont fournies par CRF++ et illustrées dans le tableau 4.2 qui spécifie le nombre de descriptions textuelles et le nombre de caractéristiques déduites :

Nombre de phrases	99
Nombre de caractéristiques	1251088
Nombre de *threads*	2

Tableau 4.2 – Résultats d'apprentissage

L'évaluation de notre approche en appliquant les champs aléatoires conditionnels a pour objectif de mesurer l'écart entre une annotation de référence et l'annotation associée lors de l'application de ce modèle. Pour ce faire, nous nous profitons des mesures suivantes : Précision, Rappel et la moyenne harmonique. Le tableau 4.3 illustre les résultats obtenus.

Annotations	**Précision**	**Rappel**	**F-mesure**
SN-B	0,9	0,28125	0,42857143
SN-E	0,9	0,28125	0,42857143
SN-M	0,9	0,375	0,52941176
SN	0,60869565	0,71186441	0,65625
IN-B	0,84848485	0,41791045	0,56
IN-M	0,84848485	0,90322581	0,875
IN-E	0,84848485	0,41791045	0,56
IN	0,9375	0,41666667	0,57692308
OU-B	0,83333333	0,18181818	0,29850746
OU-M	0,83333333	0,45454545	0,58823529
OU-E	0,83333333	0,18181818	0,29850746
OU	1	0,07142857	0,13333333
ST	0,5	0,3	0,375
ST-B	0,625	0,26315789	0,37037037
ST-M	0,83333333	0,3125	0,45454545
ST-E	0,52941176	0,47368421	0,5
-	0,9595889	0,99021829	0,97466302

Tableau 4.3 – Évaluation des résultats des champs aléatoires conditionnels

En se basant sur les précisions des différents types d'annotations sémantiques, nous remarquons qu'en appliquant le modèle discriminant CRF, nous avons obtenus des résultats de forte précision. Par exemple, la précision obtenue pour l'annotation IN est égale à 0,93. Ceci signifie que le nombre de mots correctement annotés par une annotation IN est très proche du nombre de mots annotés par IN. Ceci revient à l'utilisation des fonctions caractéristiques qui mettent en évidence l'ensemble suivant : le mot, les mots voisins, les préfixes et les suffixes, la casse, les informations sémantiques extraites à partir d'une source tels que les concepts ontologiques. Dans plusieurs cas, nous trouvons que

la valeur du rappel est faible, par exemple, pour l'annotation SN-B le rappel est égale à 0,28. Ceci est à cause du grand écart entre le nombre d'annotations SN-B correctement assignées et le nombre d'annotations SN-B existantes dans la référence qui est beaucoup plus élevé. Cependant, nous remarquons qu'il existe des cas où la valeur du rappel est élevée, par exemple pour l'annotation SN, le rappel est égal à 0,71. Ceci signifie que le modèle a correctement annoté la plupart des noms de services.

Les différentes valeurs de précision et de rappel sont résumées par l'histogramme de la figure 4.8. Nous remarquons que ce modèle a correctement associé les annotations sémantiques à la séquence de test. En effet, les noms de service Web ont été affectés avec une précision de 90%, les entrées des services Web ont été indiqués avec une précision de 88,58%, de même pour les sorties qui ont été indiqués avec une précision de 87,5%.

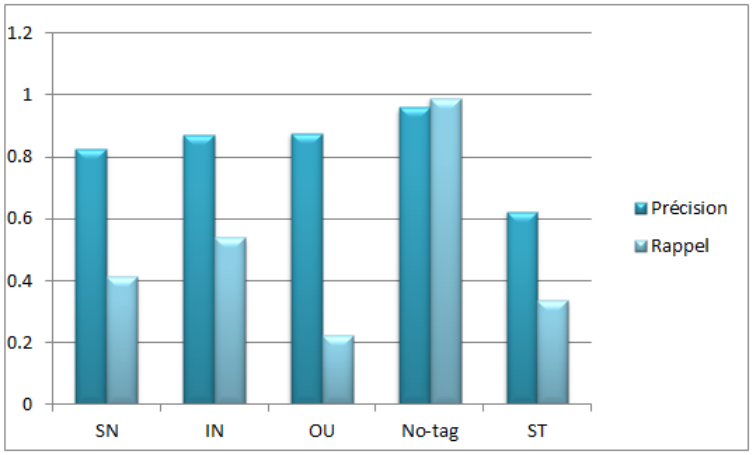

Figure 4.8 – Histogramme illustratif des résultats obtenus en appliquant CRF

4.4.3 Comparaison des résultats obtenus par les deux modèles

Les expérimentations menées par notre approche ont montré l'intérêt de l'utilisation des champs aléatoires conditionnels pour étiqueter les descriptions textuelles des SW. Dans le tableau comparatif 4.4, nous avons mis en évidence les résultats obtenus par les deux modèles. Néanmoins, nous remarquons que les valeurs obtenues lorsque l'étiquetage est réalisé par le modèle de Markov caché sont faibles. En effet, la qualité du

corpus d'apprentissage avait un impact sur la qualité d'annotation. Bien que les champs aléatoires conditionnels obtiennent de bons résultats et ceci revient à l'intégration des caractéristiques des observations.

	CRF			HMM		
Annotations	**Précision**	**Rappel**	**F-mesure**	**Précision**	**Rappel**	**F-mesure**
Service Name	0,827173913	0,4123411	0,510701155	0,013461041	0,43697479	0,02611753
Input	0,870738636	0,53892834	0,642980769	0	0	0
Output	0,875	0,2224026	0,329645888	0,070588235	0,42857143	0,12121212
No-tag	0,959588904	0,99021829	0,974663018	0,78392283	0,62910717	0,69803398
Service Task	0,621936275	0,33733553	0,424978956	0	0	0

Tableau 4.4 – Évaluation des résultats des deux modèles pour chaque type d'annotation

En comparant les résultats des deux modèles graphiques, nous remarquons que les champs aléatoires conditionnels surpassent les modèles de Markov cachés en termes de performance. En effet, les CRF ont assigné les annotations sémantiques avec succès, où les noms de service ont été correctement annotés à 82.71%, les données d'entrée à 87.07%, les données de sortie à 87.5% et les tâches de services à 62.19%. À partir des résultats, nous constatons que la précision de l'étiquetage sémantique à l'aide des CRF surmonte considérablement celle des HMM.

Nous illustrons graphiquement la comparaison entre les précisions obtenues par les deux modèles à l'aide de l'histogramme de la figure 4.9 ainsi que les rappels obtenus à l'aide de l'histogramme de la figure 4.10.

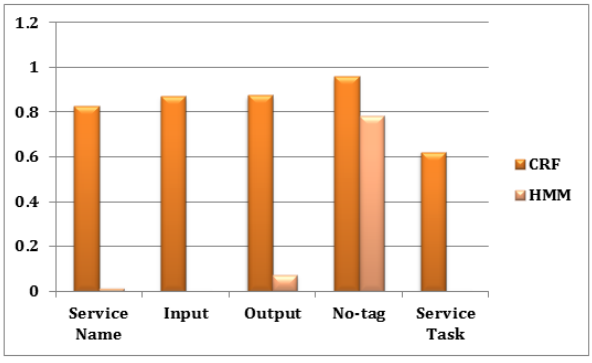

Figure 4.9 – Histogramme illustratif de la comparaison de la précision obtenue par CRF et HMM

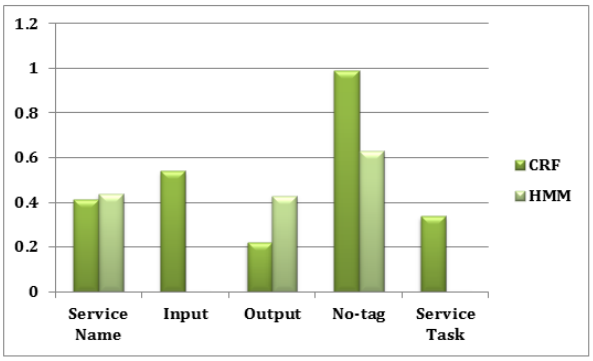

Figure 4.10 – Histogramme illustratif de la comparaison du rappel obtenu par CRF et HMM

4.5 Conclusion

Dans ce dernier chapitre, nous avons présenté le processus que nous avons adopté pour l'implémentation de notre approche sur MATLAB pour le modèle de Markov caché et CRF++ pour les champs aléatoires conditionnels. Ensuite, nous avons présenté les résultats obtenus suite à la phase d'apprentissage du corpus d'apprentissage et l'application de ces deux modèles sur le corpus test. Notamment, les mesures d'évaluation ont montré la performance et la qualité des résultats de l'annotation sémantique.

CHAPITRE 5

Conclusion générale et perspectives

Dans ce mémoire de mastère, nous nous intéressons à la problématique d'annotation sémantique des services Web dans le domaine de la bioinformatique. En effet, celle-ci vise à remédier aux lacunes du standard de description des services Web.

Cependant, la tâche de l'annotation sémantique nécessite beaucoup de temps et d'effort humain. A cet effet, plusieurs approches ont été proposées afin de (semi-) automatiser cette tâche. Dans notre travail, nous proposons une approche automatique d'annotation sémantique qui se base sur des algorithmes d'apprentissage et qui permet d'extraire, à partir de la description textuelle du service Web, quatre types d'information à savoir : le nom du service, ses paramètres d'entrées et sorties ainsi que la tâche du service. L'application des modèles graphiques probabilistes se fait sur un corpus d'apprentissage qui a été préalablement étiqueté grammaticalement puis annoté manuellement pour préciser les informations à extraire.

Le corpus d'apprentissage nous a servi à calculer les paramètres du modèle Markovien tel que les matrices de transitions ainsi que d'émissions. En cas de l'utilisation du modèle CRF, la phase d'apprentissage se base essentiellement sur la détermination des fonctions features.

Notamment, la phase suivante consiste à tester le modèle sur un corpus non annoté

en vue de déterminer les annotations sémantiques. En effet, dans le but de reconnaître convenablement les inputs et outputs unitermes, nous avons intégré une spécification du conceptontologique du domaine permettant de reconnaître les divers concepts de la bioinformatique existants dans le texte, sachant que les inputs et les outputs recherchés ne sont autres que des concepts biologiques.

Au niveau de l'implémentation de notre approche, nous avons utilisé l'outil MATLAB. En effet, nous avons appliqué un ensemble de fonctions sur le corpus d'apprentissage et le corpus test.

En général, les approches à base d'apprentissage offrent de bons résultats mais restent aptes à être améliorés car le résultat dépend de la qualité du corpus d'apprentissage qui doit être bien annoté et de taille suffisamment importante.

Le travail présenté dans ce mémoire ouvre de nombreuses perspectives de recherche. Nous commençons par présenter les perspectives à court terme avant d'étendre sur des perspectives à plus long terme.

La procédure d'annotation sémantique, actuellement, est appliquée sur un corpus de taille réduite. Il est souhaitable de le tester sur un ensemble de descriptions plus large et de diversifier les annotations sémantiques recherchées. En effet, il est envisageable d'ajouter d'autres types d'annotations. Il serait aussi intéressant de mettre en place une approche hybride combinant les modèles génératifs et discriminants et ce, pour exploiter leurs avantages.

Dans notre version actuelle, nous avons appliqué notre approche sur les descriptions textuelles des services web. Nous pouvons réfléchir à travailler sur d'autres ressources du web en vue de faciliter leur découverte.

Bibliographie

[1] Cihan Aksoy, Vincent Labatut, Chantal Cherifi, and Jean-François Santucci. Mataws : A multimodal approach for automatic ws semantic annotation. In *Networked Digital Technologies*, pages 319–333. Springer, 2011. *(Cité pages iii, iv, 5, 6 and 7.)*

[2] Thierry Artières et al. Conditional random fields for online handwriting recognition. In *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006. *(Cité page 20.)*

[3] Nadia Yacoubi Ayadi, R Ben Messaoued, and Hadhèmi Achour. Towards semantic annotation of bioinformatics web services. In *Computer Applications Technology (ICCAT), 2013 International Conference on*, pages 1–7. IEEE, 2013. *(Cité page 11.)*

[4] Sondes Bannour, Laurent Audibert, et al. Vers une approche interactive pour l'annotation sémantique. *23es journées francophones d'ingénierie des connaissances*, 2012. *(Cité page 5.)*

[5] Anne-Laure Bianne-Bernard, Christopher Kermorvant, Laurence Likforman-Sulem, and Chafic Mokbel. Modélisation de hmm en contexte avec des arbres de décision pour la reconnaissance de mots manuscrits. *Document numérique*, 14(2) :29–52, 2011. *(Cité page 13.)*

[6] *Biocatalogue.* URL : `https://www.biocatalogue.org/`, Février 2014. *(Cité page 22.)*

[7] T Brouard, M Slimane, G Venturini, and JP ASSELIN DE BEAUVILLE. Apprentissage du nombre d'états d'une chaîne de markov cachée pour la reconnaissance d'images. In *16Â° Colloque sur le traitement du signal et des images, FRA, 1997*. GRETSI, Groupe dâEtudes du Traitement du Signal et des Images, 1997. *(Cité page 13.)*

[8] Marcello Bruno, Gerardo Canfora, Massimiliano Di Penta, and Rita Scognamiglio. An approach to support web service classification and annotation. In *e-Technology, e-Commerce and e-Service, 2005. EEE'05. Proceedings. The 2005 IEEE International Conference on*, pages 138–143. IEEE, 2005. *(Cité pages iii and 10.)*

[9] Nicholas J Bryan. Hmm analysis and synthesis of acoustic drum signals. 2007. *(Cité page 13.)*

[10] Mark Burstein, Jerry Hobbs, Ora Lassila, Drew Mcdermott, Sheila Mcilraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, et al. Owl-s : Semantic markup for web services. *W3C Member Submission*, 2004. *(Cité page 5.)*

[11] Yassin CHABEB. *Contributions à la Description et la Découverte de Services Web Sémantiques*. "thèse de doctorat", Université d'Évry Val d'Essonne, 2011. *(Cité page 2.)*

[12] Peter Corbett, Colin Batchelor, and Simone Teufel. Annotation of chemical named entities. In *Proceedings of the Workshop on BioNLP 2007 : Biological, Translational, and Clinical Language Processing*, pages 57–64. Association for Computational Linguistics, 2007. *(Cité page 13.)*

[13] *CRF++*. URL : `https://github.com/YipingNUS/DefMiner/blob/master/CRF%2B%2B-0.58/Makefile.am`, Octobre 2013. *(Cité page 46.)*

[14] Aron Culotta, David Kulp, and Andrew McCallum. Gene prediction with conditional random fields. 2005. *(Cité page 19.)*

[15] Pascal Denis and Benoît Sagot. Coupling an annotated corpus and a lexicon for state-of-the-art pos tagging. *Language resources and evaluation*, 46(4) :721–736, 2012. *(Cité page 18.)*

[16] Sean R. Eddy. Profile hidden markov models. *Bioinformatics*, 14(9) :755–763, 1998. *(Cité page 39.)*

[17] Xuming He, Richard S Zemel, and MA Carreira-Perpindn. Multiscale conditional random fields for image labeling. In *Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on*, volume 2, pages II–695. IEEE, 2004. *(Cité page 19.)*

[18] Michael Herrmann, Muhammad Ahtisham Aslam, and Oliver Dalferth. Applying semantics (wsdl, wsdl-s, owl) in service oriented architectures (soa). In *10th Intl. Protégé Conference*, 2007. *(Cité page 11.)*

[19] Andreas Heß, Eddie Johnston, and Nicholas Kushmerick. Assam : A tool for semi-automatically annotating semantic web services. In *The Semantic Web–ISWC 2004*, pages 320–334. Springer, 2004. *(Cité page 6.)*

[20] *Hidden Markov Models (HMM).* URL : `http://www.mathworks.com/help/stats/hidden-markov-models-hmm.html`, Août 2013. *(Cité page 39.)*

[21] Eric Kergosien, Mouna Kamel, Christian Sallaberry, Marie-Noëlle Bessagnet, Nathalie Aussenac-Gilles, and Mauro Gaio. Construction et enrichissement automatique d'ontologie à partir de ressources externes. In *JFO'09*, pages 1–10, Poitiers, France, Décembre 2009. *(Cité page 11.)*

[22] Petr Knoth, Marek Schmidt, and Pavel Smrz. Information extraction - state-of-the-art. Technical report, 2008. *(Cité page 9.)*

[23] Grzegorz Kondrak. N-gram similarity and distance. In *String Processing and Information Retrieval*, pages 115–126. Springer, 2005. *(Cité page 8.)*

[24] Julian Kupiec. Robust part-of-speech tagging using a hidden markov model. *Computer Speech & Language*, 6(3) :225–242, 1992. *(Cité page 25.)*

[25] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. 2001. *(Cité page 19.)*

[26] *LBFGS*. URL : `http://crsouza.blogspot.com/2012/02/limited-memory-broydenfletchergoldfarbs.html`, Octobre 2013. *(Cité page 46.)*

[27] Mélanie Lemaitre. *Approche markovienne bidimensionnelle d'analyse et de reconnaissance de documents manuscrits*. PhD thesis, Université René Descartes-Paris V, 2007. *(Cité page 13.)*

[28] *Matlab*. URL : `http://www.samuelboudet.com/fr/matlab`, Décembre 2013. *(Cité page 38.)*

[29] Henning Christiansen Matthieu Petit. Un calcul de viterbi pour un modèle de markov caché contraint. *5ème Journées Francophone de Programmation par Contraintes, JFPC, Orléans, France. June 3-5*, 2009. *(Cité page 28.)*

[30] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML*, pages 591–598, 2000. *(Cité page 17.)*

[31] Ryan McDonald and Fernando Pereira. Identifying gene and protein mentions in text using conditional random fields. *BMC bioinformatics*, 6(Suppl 1) :S6, 2005. *(Cité page 19.)*

[32] Bernard Merialdo. Modèles probabilistes et étiquetage automatique. *TAL. Traitement automatique des langues*, 36(1-2) :7–22, 1995. *(Cité page 12.)*

[33] Olivier Morillot, Laurence Likforman-Sulem, et al. Reconnaissance de courriers manuscrits par hmm contextuels et modèle de langage. *Document numérique*, 16(2) :69–90, 2013. *(Cité page 13.)*

[34] Nomane Ould Ahmed M'Barek and Samir Tata. Services Web : revue des approches de description sémantique. In *SIIE '2008 : Système d'information et Intelligence Economique, 14-16 Février, Hammamet, Tunisie*, 2008. *(Cité page 2.)*

[35] Abhijit A Patil, Swapna A Oundhakar, Amit P Sheth, and Kunal Verma. Meteor-s web service annotation framework. In *Proceedings of the 13th international conference on World Wide Web*, pages 553–562. ACM, 2004. *(Cité page 7.)*

[36] Adam Pease. The sigma ontology development environment. In *Working Notes of the IJCAI-2003 Workshop on Ontology and Distributed Systems*, volume 71, 2003. *(Cité page 6.)*

[37] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989. *(Cité page 13.)*

[38] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. *(Cité page 44.)*

[39] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5) :513–523, 1988. *(Cité page 9.)*

[40] Kristie Seymore, Andrew McCallum, and Roni Rosenfeld. Learning hidden markov model structure for information extraction. In *AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 37–42, 1999. *(Cité page 13.)*

[41] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics, 2003. *(Cité page 19.)*

[42] Mark Stamp. A revealing introduction to hidden markov models. *Department of Computer Science San Jose State University*, 2012. *(Cité page 27.)*

[43] Martin Szummer and Yuan Qi. Contextual recognition of hand-drawn diagrams with conditional random fields. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, pages 32–37. IEEE, 2004. *(Cité page 19.)*

[44] *taverna*. URL : `http://www.taverna.org.uk/introduction/taverna-features/biocatalogue-integration/`, Mai 2013. *(Cité pages 2 and 22.)*

[45] Marc Vincent and Grégoire Winterstein. Construction et exploitation d'un corpus français pour l'analyse de sentiment. In *Actes de la 20e conférence sur le Traite-*

ment Automatique des Langues Naturelles (TALN'2013), pages 764–771, Les Sables d'Olonne, France, 2013. *(Cité page 12.)*

[46] Hanna Wallach. *Efficient training of conditional random fields.* "thèse de doctorat", Université de Edinburgh, 2002. *(Cité page 18.)*

[47] Ian H Witten and Eibe Frank. *Data Mining : Practical machine learning tools and techniques*. Morgan Kaufmann, 2005. *(Cité page 8.)*

[48] Kyoung-Jae Won, Adam Prügel-Bennett, and Anders Krogh. Training hmm structure with genetic algorithm for biological sequence analysis. *Bioinformatics*, 20(18) :3613–3619, 2004. *(Cité page 13.)*

[49] Byung-Jun Yoon. Hidden markov models and their applications in biological sequence analysis. *Current genomics*, 10(6) :402, 2009. *(Cité page 13.)*

[50] Shun-Zheng Yu and Hisashi Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden markov model. *Signal Processing Letters, IEEE*, 10(1) :11–14, 2003. *(Cité page 14.)*

[51] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. 2d conditional random fields for web information extraction. In *Proceedings of the 22nd international conference on Machine learning*, pages 1044–1051. ACM, 2005. *(Cité page 20.)*

[52] Azeddine Zidoun. *Modèles graphiques discriminants pour l'étiquetage de séquences : application à la reconnaissance d'entités nommées radiophoniques*. "thèse de doctorat", Université de la méditerranée Aix-Marseille II, 2010. *(Cité pages iii, 12 and 34.)*