UNIVERSITÉ DU QUÉBEC À MONTRÉAL

SYSTÈME DE SEGMENTATION AUTOMATIQUE DES RÉFÉRENCES BIBLIOGRAPHIQUES À BASE DES CHAMPS ALÉATOIRES CONDITIONNELS

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

SEIFEDDINE AISSA

JANVIER 2020

REMERCIEMENTS

Je tiens à exprimer mes vifs remerciements à tous ceux et celles qui m'ont aidé à réaliser ce travail, lesquels sont cités ci-après.

Mon directeur de recherche, Monsieur Aziz Salah, pour avoir accepté d'être mon superviseur et m'avoir offert un suivi constant couplé à une collaboration exemplaire.

La Faculté des sciences et la Fondation de l'Université du Québec à Montréal pour la bourse d'exemption des frais de scolarité majorés et la bourse d'excellence, toutes deux obtenues au fil de mon parcours universitaire.

Mes chers parents, Karem et Raoudha, qui ont tout sacrifié pour me permettre de réaliser ce parcours. Je suis immensément reconnaissant pour leur indéfectible soutien. Chers parents, les mots me manquent pour vous témoigner toute ma gratitude et mon amour, mais je peux quand même souligner ma reconnaissance pour l'appui et la tendresse que vous m'avez offerts depuis ma naissance.

Merci à mes chers beaux-parents, Abdelweheb et Latifa, qui n'ont jamais cessé de m'offrir leur aide, et ce, de plusieurs façons.

Merci à ma chère femme Farah, ma source d'inspiration, pour ton amour et le soutien que tu m'as offert. Je t'exprime ma reconnaissance pour tes encouragements et ton aide constante.

Merci également à mes frères et soeur Amine, Mouhamed Amine et Ghofrane pour leur chaleureux élans et leur affection. Je vous souhaite à tous un avenir brillant.

Pour finir, je voudrais témoigner ma reconnaissance à l'ensemble de ma famille ainsi qu'à tous mes amis.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

RÉSUMÉ

Vu l'évolution des technologies de l'information, il existe de nos jours une multitude de documents qui varient en format, syntaxe et niveau d'abstraction. Cette gigantesque hausse de la quantité d'informations complexifie la tâche d'extraction de données. Cette tâche permet de reconnaître, d'extraire et de structurer un ensemble d'informations spécifiques dans un corpus de documents donné. L'une des tâches les plus importantes dans l'extraction d'information est celle de l'étiquetage de séquences, qui consiste à trouver et étiqueter les segments à partir des données textuelles.

Plusieurs méthodes d'apprentissage automatiques ont vu le jour pour lutter contre le problème d'extraction de données. Parmi elles, nous retrouvons les méthodes d'apprentissage supervisé telles que les champs aléatoires conditionnels (CRF) et les modèles de Markov cachés (HMM). D'autres approches d'apprentissage non-supervisées ont également été utilisées : parmi elles, la méthode On-Demand Unsupervised Learning for Information Extraction (ONDUX). Les techniques qui nous intéressent particulièrement sont celles se basant sur un apprentissage supervisé. Ces méthodes ont été utilisées dans plusieurs domaines, principalement lors de l'étiquetage de séquences.

Dans le cadre de ce travail, nous livrons un système mettant à profit les champs aléatoires conditionnels (CRF) afin d'étiqueter n'importe quelle référence bibliographique. Nous nous basons sur une approche supervisée qui améliore d'une part les résultats de la littérature et d'une autre part généralise leur application pour donner naissance à un classifieur généralisé. Les CRF, que nous utiliserons ici, sont des modèles graphiques probabilistes qui permettent d'extraire et d'identifier différentes données telles que les auteurs, les titres d'articles ou de conférences, la date et le nombre de pages. Cette étude se limitera au problème d'extraction d'informations à partir des références bibliographiques. C'est donc dire que notre champ de recherche pourrait s'étendre à d'autres domaines liés à l'étiquetage de données.

Mots clés : Extraction d'information, Étiquetage de séquences, Apprentissage supervisé, Modèles graphiques probabilistes, Champs aléatoires conditionnels (CRF).

INTRODUCTION

0.1 Mise en contexte

L'évolution des technologies de l'information nous pousse à manipuler une grande quantité de données. En effet, l'augmentation et la variété des documents générés rendent l'exploitation de ces derniers de plus en plus difficile pour les humains. C'est d'ailleurs cette évolution accélérée qui a provoqué la naissance de la tâche que l'on nomme extraction d'information.

L'extraction d'information est une nouvelle technologie qui permet d'analyser un ou plusieurs documents textuels pour en obtenir des informations en vue d'une application précise. L'extraction d'information permet également de faire ressortir les éléments pertinents d'une information structurée. Ce type de tâche est devenu un enjeu de recherche très important, surtout dans le domaine du Traitement Automatique des Langues Naturelles (TALN).

L'une des techniques les plus importantes dans l'extraction d'information correspond à la segmentation d'une séquence du texte, qui permet de détecter le début et la fin d'un segment à partir d'une simple séquence (une phrase, par exemple). Dans le cas d'une séquence représentant une référence bibliographique, la segmentation aide à cerner le début et la fin des éditeurs, des titres et des auteurs. Lors de la reconnaissance d'entités nommées, par contre, la segmentation permet de délimiter le début et la fin des organisations, des emplacements, etc.

Dans l'extraction d'information, l'étape suivant la segmentation est celle de l'étiquetage de séquence. En effet, l'étiquetage permet d'identifier les segments trouvés

dans une séquence, qui seront ensuite identifiés avec l'étiquette correspondante. Nous présentons ici un exemple d'étiquetage des références bibliographiques à partir du dataset Cora (McCallum *et al.*, 2000).

[author **Witten, I. H., Neal R. M., and Cleary J. G.**] [date **(1987).**]
[title **Arithmetic coding for data compression.**] [journal **Communications of the ACM.**] [volume **30,**] [pages **520-540.**]

Il est à remarquer que l'étiquetage de la référence ci-dessus nous a permis d'attribuer les étiquettes *author*, *date*, *title*, *journal*, *volume* et *pages* aux segments correspondants. Présentons maintenant un second exemple d'étiquetage d'une séquence, celui-ci mené lors de la reconnaissance d'entités nommées à partir du jeu de données CoNLL-2003 (Sang et De Meulder, 2003).

[organization **U.N.**] [outside **official**] [person **Ekeus**] [outside **heads**] [outside **for**]
[location **Baghdad**] [outside **.**]

Il est à remarquer que l'étiquetage de la séquence ci-dessus nous a menés vers l'attribution des étiquettes *person*, *location*, *organization* et *outside* aux segments correspondants. Lors de l'extraction d'information, nous constatons plusieurs difficultés concernant la qualité et la diversité des données disponibles. En effet, la grande quantité d'informations couplée à leur variété rendent l'extraction d'information manuelle avec des expressions régulières de plus en plus complexe. Par conséquent, et dans le but de faciliter le traitement des données, les chercheurs se sont orientés vers l'automatisation, où plusieurs techniques d'apprentissage automatique ont été exploitées. En effet, plusieurs méthodes basées sur un apprentissage supervisé, telles que les champs aléatoires conditionnels (CRF), les machines à vecteurs de support (SVM), les modèles de Markov cachés (HMM) et les modèles à maximum d'entropie (MEMM), ont été explorés dans les travaux de (Peng

et McCallum, 2006), (Han *et al.*, 2003), (Chieu et Ng, 2003) et (Seymore *et al.*, 1999). D'autres techniques davantage fondées sur un apprentissage non-supervisé ont aussi été exploitées, notamment par (Cortez *et al.*, 2010) et (Cortez *et al.*, 2011).

Bon nombre de travaux sur l'extraction d'information se sont concentrés sur la reconnaissance d'entités nommées. Parmi ceux-ci, citons ceux de (Chieu et Ng, 2003), (Seymore *et al.*, 1999), (McCallum, 2003) et (Han *et al.*, 2003). D'autres méthodes ont également été utilisées pour segmenter les références bibliographiques, notamment par (Hetzner, 2008), (Councill et Kan, 2008) et (Peng et McCallum, 2006).

Notre étude se concentre principalement sur l'étiquetage des références bibliographiques. Ce dernier a pour objectif la segmentation de la référence bibliographique et l'extraction des différents champs qui la composent, par exemple ses auteurs, son titre, etc.

0.2 Problématique

Pour automatiser la tâche d'extraction d'information par la segmentation du texte, plusieurs méthodes d'apprentissage automatique ont été mises à profit. Elles se basent pour la plupart sur des classifieurs permettant d'étiqueter les données issues d'un dataset. Ces classifieurs utilisent des caractéristiques qui ont été fournies par des experts en étiquetage : ainsi, un bon choix de caractéristiques permet l'obtention de bons résultats de prédiction. À ce propos, la question qui se pose est la suivante : comment choisir des caractéristiques favorisant un apprentissage efficace ?

En général, un classifieur issue d'un entraînement supervisé ne peut être appliqué qu'aux entrées qui ressemblent à ses données d'entraînement. Par ailleurs, les

références peuvent provenir de plusieurs sources aux styles qui diffèrent de celui des données d'entraînement, ce qui risque de rendre la tâche du classifieur inefficace. Comment pouvons-nous donc généraliser l'apprentissage d'un classifieur ? Est-il possible de concevoir un système qui permettrait d'étiqueter n'importe quelle référence bibliographique ?

0.3 Motivations

L'extraction d'information par la segmentation du texte est une problématique d'actualité. La portée de cette tâche ainsi que les travaux qui s'intéressent aux références bibliographiques ont motivé notre intérêt à générer un système utilisant un algorithme d'apprentissage capable d'étiqueter n'importe quelle référence bibliographique. Même si cette problématique a été bien étudiée sur plusieurs types de données, elle demeure ouverte dans le cadre d'étiquetage des références bibliographiques. Plusieurs critères en lien avec l'utilité et l'importance de cette tâche nous ont orientés vers le sujet. Parmi ceux-ci, citons :

- la valoristion de l'information pour la prise de décision ;
- le remplissage d'une base de données avec des références bibliographiques permettant de sélectionner l'information souhaitée à travers des requêtes précises ;
- l'utilisation des résultats de l'étiquetage pour d'autres tâches comme la coréférence et l'étude des communautés scientifiques.

0.4 Contributions

Les CRF sont des modèles graphiques probabilistes discriminants introduits par (Lafferty *et al.*, 2001). Afin de pallier aux limitations mentionnées précédem-

ment, et pour valoriser l'algorithme d'apprentissage basé sur les CRF, nous livrons un système permettant d'étiqueter n'importe quelle référence bibliographique. Il s'agit donc de l'objet de ce mémoire. Notre travail se base sur une approche supervisée améliorant d'une part les résultats de la littérature et combinant d'une autre part plusieurs classifieurs CRF dans le but de composer un système généralisé d'étiquetage des références bibliographiques. Notre recherche est caractérisée par les éléments suivants :

1. Elle reproduit et améliore les résultats de l'état de l'art des classifieurs CRF pour l'étiquetage des références bibliographiques du dataset Cora (McCallum *et al.*, 2000).

2. Elle se base sur le principe selon lequel les références varient en styles. Puisque la méthode que nous avons élaborée permet de les détecter automatiquement, nous pouvons retenir l'étiquetage du classifieur entraîné sur le même style que sa source et ainsi obtenir un système d'étiquetage de références bibliographiques généralisé.

0.5 Structure et organisation du mémoire

Le présent mémoire s'organise comme suit. Le deuxième chapitre passe en revue la littérature sur la tâche d'extraction d'information, l'apprentissage automatique et les modèles graphiques probabilistes. Ce chapitre s'intéresse en outre à l'ingénierie des caractéristiques et à la mesure des performances des classifieurs. Le troisième chapitre décrit une revue de la littérature des travaux liés à l'étiquetage de séquences, principalement dédiée à la reconnaissance d'entités nommées et la segmentation des références bibliographiques. Dans le quatrième chapitre, nous présentons plus directement notre approche d'étiquetage de séquences qui se base sur les champs aléatoires conditionnels (CRF) appliquée au dataset Cora.

L'ultime et cinquième chapitre élabore un système valorisant notre approche d'étiquetage de séquences, basée sur les CRF pour une segmentation des références et une construction d'un classifieur généralisé. Enfin, le mémoire se termine par une conclusion récapitulative.

CHAPITRE I

TECHNIQUES D'ÉTIQUETAGE DE SÉQUENCES

Ce chapitre introduit quelques techniques utilisées lors de l'étiquetage de séquences. L'objectif de ce chapitre est de faciliter au lecteur la compréhension de la tâche d'extraction d'information en général et celle de l'étiquetage de séquences en particulier.

Dans cette partie, nous commençons par présenter l'extraction d'information et son importance dans plusieurs domaines. Ensuite, nous définissons l'apprentissage automatique à travers l'apprentissage supervisé et l'apprentissage non-supervisé. Nous relatons par la suite certains modèles d'étiquetage de séquences principalement, les modèles graphiques probabilistes (CRF, HMM). Nous examinons aussi l'ingénierie des caractéristiques et nous soulignons son impact sur les résultats d'étiquetage. Finalement, nous concluons ce chapitre par l'étude des mesures de performances.

1.1 La tâche d'extraction d'information

L'augmentation du nombre de documents et des capacités de traitement électronique de l'information ont accéléré la naissance de l'extraction d'information. En effet, plusieurs difficultés telles que la flexibilité, l'ambiguïté et la variation des données (Tartier, 2001) rendent cette tâche délicate (Gaizauskas, 2002). Par

conséquent, ces variations ont imposé l'automatisation de la tâche d'extraction d'information et ont poussé les chercheurs à étudier davantage ce domaine.

L'extraction d'information permet d'identifier des occurrences particulières, d'extraire les arguments associés et d'organiser les données. À cet égard, l'extraction est réalisée grâce au remplissage de formulaires qui présente la connaissance à rechercher selon une structure bien déterminée. Ces formulaires contiennent un ensemble d'entités, les relations entre celles-ci et les événements qui les impliquent (Yangarber *et al.*, 2000). Lors de l'extraction d'information, les utilisateurs peuvent utiliser les informations extraites pour consulter et alimenter un entrepôt de données (Pazienza, 2006).

La collecte d'information par les techniques d'extraction a montré son efficacité par rapport à la collection réalisée manuellement par les humains. En outre, les études élaborées par (Will, 1993b) et (Will, 1993a) ont prouvé que les processus automatiques d'extraction donnent des résultats dont la précision est plus élevée que les travaux effectués manuellement par les êtres humains.

Beaucoup de travaux qui traitent de ce principe ont été exposés dans la littérature. Principalement, la notion d'extraction d'information a été appliquée souvent sur les données textuelles. En effet, l'étiquetage des références bibliographiques, par exemple, permet d'identifier les noms des auteurs, le titre d'un article ou d'une conférence, la date et les numéros de pages. Nous présentons ci-dessous un exemple d'étiquetage des références bibliographiques.

[author **Rivest, R. L.**] [date **(1987).**] [title **Learning decision lists.**] [booktitle **Machine Learning,**] [volume **2 (3),**] [pages **229-246.**]

[author **B. K. P. Horn.**] [title **Robot Vision.**] [publisher **MIT Press,**] [location **Cambridge, MA,**] [date **1986.**]

La première étape que nous appliquons lors de l'extraction d'information est la segmentation (tokenisation). Dans le cadre d'étiquetage de séquence à partir d'une référence bibliographique par exemple, chaque référence sera segmentée et chaque segment identifié sera par la suite étiqueté. La deuxième étape consiste à segmenter chaque segment obtenu en un ensemble de jetons (mots) qui seront également étiquetés (Sutton *et al.*, 2012). Dans ce cadre, plusieurs notations ont été proposées dans la littérature afin d'étiqueter les jetons d'un même segment. Celle qui nous intéresse le plus est l'utilisation de la notation BIO qui a été introduite par (Ramshaw et Marcus, 1999).

L'étiquetage de séquences avec BIO permet de diviser l'étiquette introduite pour un segment donnée en une séquence d'étiquettes. Autrement dit, le *(B-)* est utilisé pour identifier le premier jeton d'un segment étiqueté, le *(I-)* pour les jetons internes et de fin du même segment. Pour les segments qui sont hors études, la notation BIO étiquette leurs jetons avec *(O-)*. Dans notre cas, tous les segments de la séquence sont d'intérêt. Par conséquent, nous n'allons pas utiliser l'étiquette *(O-)* dans notre processus.

L'étiquetage en BIO permet donc de doubler le nombre d'étiquettes. Chaque étiquette *Étiquette* va plus précisément être divisée en deux étiquettes *B-Étiquette* et *I-Étiquette*. Ces deux étiquettes seront traitées différemment par le modèle utilisé. Nous présentons ci-dessous un exemple d'étiquetage de séquences à partir d'une référence bibliographique avec la notation BIO.

Rivest, R. L. (1987). Learning decision lists. Machine Learning, 2 (3), 229-246.

B-author I-author I-author B-date B-title I-title I-title B-booktitle I-booktitle B-volume I-volume B-pages

Dans cet exemple, nous pouvons remarquer que l'étiquetage par la notation BIO du segment *author*, par exemple, nous a permis d'attribuer l'étiquette *B-author* au premier jeton (*Rivest,*) et l'étiquette *I-author* au reste des jetons (*R.* et *L.*). Le même concept est effectué pour les autres segments.

Dans le reste de ce chapitre, nous allons relater de manière détaillée l'apprentissage automatique en soulignant son importance lors de l'étiquetage de séquences.

1.2 Apprentissage automatique

L'apprentissage automatique est un sous-domaine de l'intelligence artificielle qui utilise des approches statistiques. En effet, le but de l'apprentissage automatique est de bien comprendre la structure des données et de les intégrer par la suite dans des modèles qui seront exploités par des experts.

L'apprentissage automatique diffère des approches informatiques traditionnelles. Ainsi, les algorithmes d'apprentissage automatique permettent aux ordinateurs d'apprendre à partir de données et plus précisément d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune.

La majorité des travaux d'étiquetage de séquences se basent sur des techniques d'apprentis- sage automatique. À ce propos, les types d'apprentissages les plus utilisés sont l'apprentissage supervisé et l'apprentissage non-supervisé.

L'apprentissage supervisé utilise un système dans lequel les classes (attributs) sont prédéterminées et les exemples sont connus. Lors de l'étiquetage de séquence, le processus se base sur deux étapes. La première étape consiste à déterminer un modèle à partir des données étiquetées, alors que la deuxième étape consiste à prédire l'étiquette d'une nouvelle donnée en connaissant le modèle préalablement appris. Plusieurs méthodes d'étiquetage de séquences supervisées ont été publiées dans

la littérature. Nous pouvons citer par exemple les champs aléatoires conditionnels (CRF) et les modèles de Markov cachés (HMM).

Contrairement à l'apprentissage supervisé, l'apprentissage non-supervisé utilise un système avec seulement des données non étiquetées. Plusieurs méthodes d'étiquetage de séquences non-supervisées ont été publiées dans la littérature. Nous pouvons citer celles effectuées par (Cortez *et al.*, 2010) et par (Cortez *et al.*, 2011).

Dans la prochaine section, nous allons étudier certaines méthodes qui ont été utilisées pour l'étiquetage de séquences. Notre intérêt est dirigé vers les méthodes basées sur un apprentissage supervisé.

1.3 Modèles graphiques probabilistes pour l'étiquetage de séquences

Les modèles graphiques probabilistes permettent l'étiquetage de séquences. Plusieurs modèles graphiques génératifs et discriminants existent dans la littérature. Nous avons choisi d'étudier les modèles de Markov cachés (HMM) et les champs aléatoires conditionnels (CRF) étant donné qu'ils ont donné de bons résultats lors de l'étiquetage de séquences.

1.3.1 Les modèles de Markov cachés (HMM)

Les modèles de Markov cachés sont des modèles graphiques probabilistes génératifs introduits par le professeur *Baum* et ses collaborateurs dans les années 1960-1970 (Baum *et al.*, 1970). En effet, ces modèles sont une extension de chaîne de Markov où les états ne sont plus directement observables ; ils sont cachés par un processus d'observations.

Les HMM sont très utilisés en intelligence artificielle, dans la reconnaissance de formes et en traitement automatique du langage naturel. En effet, un modèle de Markov caché est un cas particulier du réseau bayésien et peut être représenté par un graphe comme celui de la Figure 1.1 (Suire, 2011).

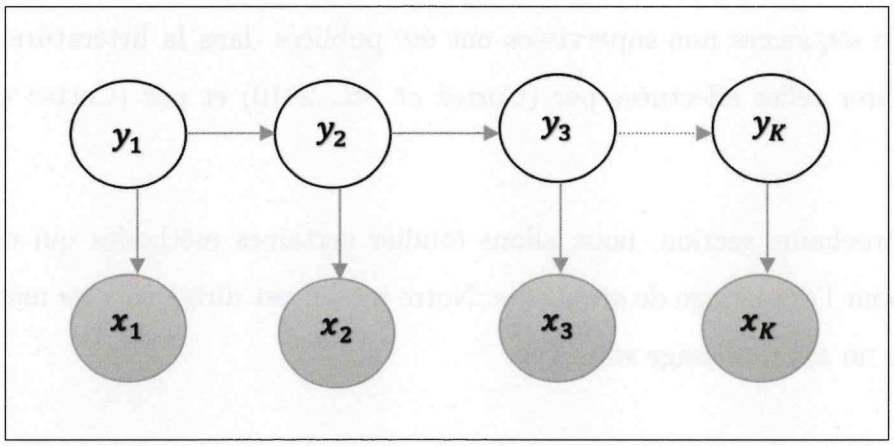


Figure 1.1: Exemple d'un modèle HMM avec x_1,x_2,x_3,...,x_K observations et y_1,y_2,y_3,...,y_K états.

Nous pouvons remarquer que le graphe de la Figure 1.1 est composé de deux couches : la première couche contient les observations x_1,x_2,x_3,...,x_K et la deuxième couche contient les états y_1,y_2,y_3,...,y_K. Dans le domaine du traitement automatique de la langue naturelle (TALN) et plus précisément lors de l'étiquetage de séquences, les états d'un HMM sont les labels (étiquettes) et les observations sont les jetons (mots). La flèche du graphe représente la dépendance entre les états et les observations du modèle.

Pour mieux comprendre ce modèle, nous présentons ci-dessous un exemple d'étiquetage de séquence avec un HMM pour la reconnaissance d'entités nommées.

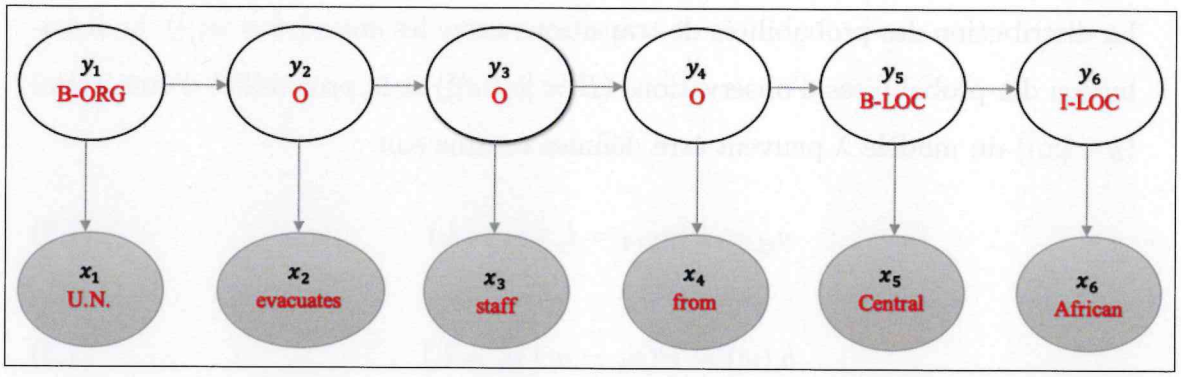


Figure 1.2: Exemple d'étiquetage de séquences avec un modèle HMM pour la reconnaissance d'entités nommées.

D'après l'exemple de la Figure 1.2, le modèle HMM est constitué d'une séquence d'étiquettes y_i (I-PER, O) et une séquence de jetons x_i (C., Cairns, Saglain, etc.).

Un modèle de Markov caché est composé des éléments suivants :

- $y_{1:K} = y_1,y_2,y_3,...,y_K$: Les étiquettes ou les états du modèle ;
- $x_{1:K} = x_1,x_2,x_3,...,x_K$: Les jetons ou les observations du modèle ;
- $A = [a_{ij}]$: La distribution des probabilités de transitions entre les états (c'est-à-dire les probabilités de passer d'un état à l'autre) ;
- $B = [b_i(w)]$: La distribution des probabilités d'observations (c'est-à-dire les probabilités pour chaque état d'émettre chacune des observations possibles) ;
- $\pi = [\pi_i]$: La probabilité d'état initial.

Soit λ un modèle HMM tel que $\lambda = (A, B, \pi)$. La probabilité jointe d'une séquence d'états et observations d'un HMM est définie par (Chikhaoui, 2013) :

$$P_\lambda(y_{1:K}, x_{1:K}) = P(y_{1:K}, x_{1:K}; \lambda) = P(y_1) * \prod_{k=1}^{K-1} P(y_{k+1} \mid y_k) P(x_k \mid y_k) \quad (1.1)$$

La distribution des probabilités de transitions entre les états ($A = [a_{ij}]$), la distribution des probabilités d'observations ($B = [b_i(w)]$) et la probabilité d'état initial ($\pi = [\pi_i]$) du modèle λ peuvent être définies comme suit :

$$a_{ij} = P(y_{k+1} = l_j \mid y_k = l_i) \tag{1.2}$$

$$b_i(w) = P(x_k = w \mid y_k = l_i) \tag{1.3}$$

$$\pi_i = P(y_1 = l_i) \tag{1.4}$$

Avec :

- l_1, l_2, ..., l_i, ..., l_j, ..., l_n : Les valeurs des états (étiquettes) y_k du modèle ;
- w : Un jeton parmi les jetons x_k du modèle.

Le paramètre $\lambda = (A, B, \pi)$ est calculé à partir des données d'apprentissage en maximisant la vraisemblance. Le calcul est effectué comme suit :

$$a_{ij} = P(y_{k+1} = l_j \mid y_k = l_i) = \frac{\#(l_i \rightarrow l_j)}{\#(l_i \rightarrow ?)} \tag{1.5}$$

$$b_i(w) = P(x_k = w \mid y_k = l_i) = \frac{\#(l_i \rightsquigarrow w)}{\#(l_i \rightsquigarrow ?)} \tag{1.6}$$

$$\pi_i = P(y_1 = l_i) = \frac{\#(y_1 = l_i)}{\#(sequences)} \tag{1.7}$$

Avec :

- a_{ij} : Le nombre de fois où l_i est suivi par l_j divisé par le nombre de fois où la transition commence par l_i.

- $b_i(w)$: Le nombre de fois que le jeton w est étiqueté par l_i divisé par le nombre d'occurrences de l_i.
- π_i : Le nombre de séquences qui commencent par l_i divisé par le nombre total de séquences.

Lors de l'étiquetage de séquences, un modèle de Markov caché permet d'avoir toutes les probabilités d'étiquetage. En effet, la probabilité d'étiquetage d'un HMM est présentée comme suit :

$$P(y_{1:K} \mid x_{1:K}) = \frac{P(y_{1:K}, x_{1:K})}{P(x_{1:K})} \tag{1.8}$$

Sachant que :

- $x_{1:K} = x_1.....x_K$: $jeton_1...jeton_K$: La séquence d'observations (jetons) ;
- $y_{1:K} = y_1.....y_K$: $etiquette_1...etiquette_K$: La séquence d'états (étiquettes).

Le choix de l'étiquetage ayant la probabilité la plus élevée est réalisé avec l'algorithme de Viterbi (Forney, 1973). En effet, la prédiction est établie selon (Do et Artières, 2006) :

$$y^*_{1:K} = \arg\max_{y_{1:K}} P(y_{1:K}, x_{1:K}) \tag{1.9}$$

1.3.2 Les champs aléatoires conditionnels (CRF)

Les champs aléatoires conditionnels (Lafferty *et al.*, 2001) sont des modèles graphiques probabilistes discriminants. Comme le montrent plusieurs travaux de la littérature, ces modèles sont très performants lors de l'étiquetage de séquences. Nous citons par exemple les travaux effectués par (McCallum et Li, 2003), (Sha et Pereira, 2003), (Tsuruoka *et al.*, 2009) et (Tellier *et al.*, 2010).

Dans notre étude, nous nous limitons à un CRF linéaire qui peut être représenté par un graphe comme celui de la Figure 1.3.

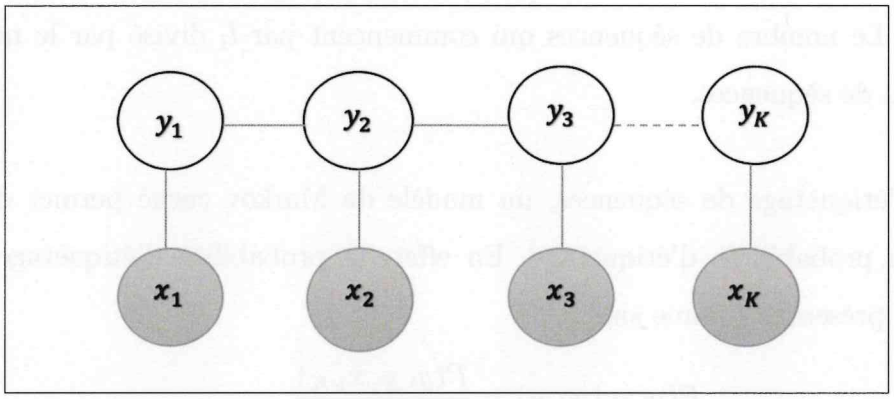


Figure 1.3: Exemple d'un CRF linéaire avec x_1,x_2,x_3,...,x_K observations et y_1,y_2,y_3,...,y_K états.

Nous pouvons remarquer que le graphe de la Figure 1.3 est aussi composé de deux couches : la première couche contient les observations x_1,x_2,x_3,...,x_K et la deuxième couche contient les états y_1,y_2,y_3,...,y_K. Contrairement à un modèle HMM, le graphe d'un CRF n'est pas orienté (pas de dépendance). De ce fait, nous pouvons constater que chaque étiquette influence l'étiquette d'avant et celle d'après, tacitement, de la donnée x complète. Par exemple, l'état (étiquette) y_2 influence l'état y_3 et aussi l'état y_1.

Pour mieux comprendre ce modèle, nous présentons ci-dessous un exemple d'étiquetage de séquences avec un CRF linéaire pour la reconnaissance d'entités nommées.

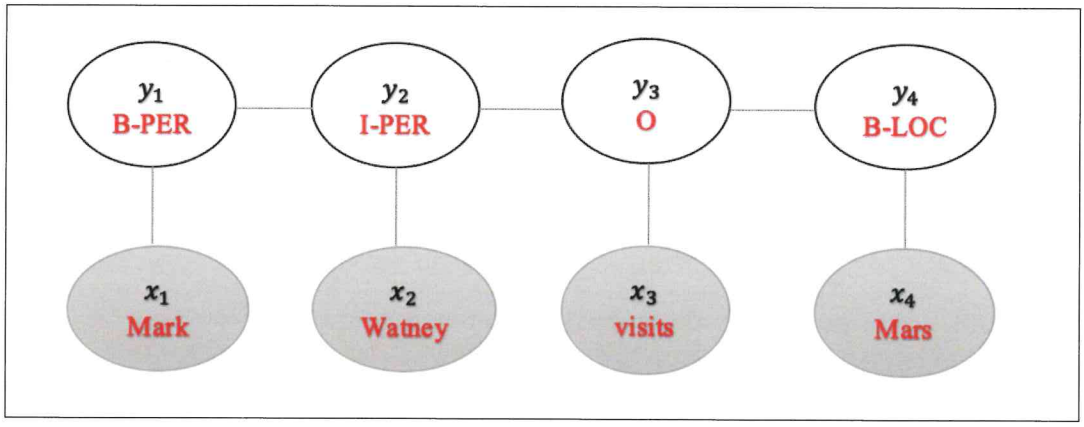


Figure 1.4: Exemple d'étiquetage de séquences avec un CRF linéaire pour la reconnaissance d'entités nommées.

D'après la Figure 1.4, nous pouvons remarquer que lors de l'étiquetage de séquence, le CRF linéaire est constitué d'une séquence d'étiquette y_i (B-PER, I-PER, O, B-LOC) et d'une séquence de jetons x_i (Marl, Watney, etc.).

Un CRF linéaire est composé des éléments suivants (Sutton *et al.*, 2012) :

- $y_{1:K} = y_1,y_2,y_3,...,y_K$: Les étiquettes ou les états du modèle ;
- $x_{1:K} = x_1,x_2,x_3,...,x_K$: Les jetons ou les observations du modèle ;
- $Z(x)$: Le coefficient de normalisation ;
- f_i : La fonction caractéristique i ;
- λ_i : Le poids de la fonction caractéristique i.

La probabilité conditionnelle d'un CRF linéaire peut être représentée par (Constant *et al.*, 2011) :

$$P_\Lambda(y_{1:K} \mid x_{1:K}) = \frac{1}{Z(x_{1:K})} exp(\sum_{k=1}^{K} \sum_{i=1}^{F} \lambda_i f_i(y_{k+1}, y_k, x_k)) \tag{1.10}$$

Sachant que : $\Lambda = (\lambda_i)_{1 \leq i \leq F}$

$$Z(x_{1:K}) = \sum_{y_{1:K}} exp(\sum_{k=1}^{K} \sum_{i=1}^{F} \lambda_i f_i(y_{k+1}, y_k, x_k)) \tag{1.11}$$

L'apprentissage par un CRF linéaire est effectué en maximisant le logarithme de la vraisemblance à partir des données d'entraînement $\{(\mathrm{x}^{(i)}, \mathrm{y}^{(i)}) : i = 1,...M\}$. Par conséquent, le paramètre Λ^* est défini par :

$$\Lambda^* = \arg\max_{\Lambda} \frac{1}{M} \sum_{i=1}^{M} log P_{\Lambda}(\mathrm{y}^{(i)} \mid \mathrm{x}^{(i)}) \tag{1.12}$$

Avec :

- $\mathrm{x}^{(i)}$: Une séquence ;
- $\mathrm{y}^{(i)}$: Les étiquettes.

La prédiction est alors établie selon :

$$\widehat{y} = \arg\max_{y} P_{\Lambda^*}(y \mid x) \tag{1.13}$$

L'un des avantages des champs aléatoires conditionnels (CRF) est qu'ils permettent d'intégrer plusieurs connaissances à travers les fonctions caractéristiques. En effet, l'importance des fonctions caractéristiques lors de l'étiquetage sera déterminée dans la phase d'apprentissage. Nous présentons ci-dessous un exemple de quelques fonctions caractéristiques que nous pouvons utiliser dans un CRF linéaire.

$$f_1(y_{k+1}, y_k, x_k) = \begin{cases} 1 & \text{si } y_{k+1} = \text{B-LOC et capitalized } (x_k) \\ 0 & \text{sinon.} \end{cases} \tag{1.14}$$

$$f_2(y_{k+1}, y_k, x_k) = \begin{cases} 1 & \text{si } y_{k+1} = \text{I-PER et capitalized } (x_k) \\ 0 & \text{sinon.} \end{cases} \tag{1.15}$$

$$f_3(y_{k+1}, y_k, x_k) = \begin{cases} 1 & \text{si } y_{k+1} = \text{B-PER et capitalized } (x_k) \\ 0 & \text{sinon.} \end{cases} \tag{1.16}$$

Les fonctions caractéristiques seront présentées en détail dans la prochaine section.

Nous ne pouvons pas étudier un CRF linéaire sans introduire le surapprentissage qui affecte souvent les résultats de ce modèle. En effet, ce phénomène signifie que le modèle est trop lié aux données d'apprentissage. Autrement dit, le CRF a trop appris les particularités des exemples fournis dans les données d'apprentissage et n'arrive pas à généraliser. Nous allons discuter en détail du surapprentissage dans le reste de ce chapitre.

1.4 Définition d'une fonction caractéristique dans un CRF linéaire

Une caractéristique dans un CRF linéaire est une fonction qui retourne une valeur numérique. En effet, les caractéristiques peuvent être classées selon trois types. Le premier type englobe les caractéristiques prédicats qui retournent 1 si la caractéristique est vérifiée et 0 dans le cas contraire. Le deuxième type contient les caractéristiques numériques qui retournent une valeur numérique (réelle) que nous calculons pour un jeton donné dans une séquence. Le dernier type inclut les caractéristiques catégoriques. Ces dernières sont utilisées dans un domaine catégorique bien déterminé. Ce domaine contient un nombre de catégories qui seront converties en prédicats de telle sorte que le nombre de catégories est égal au nombre de prédicats obtenus. En outre, chaque catégorie représente une caractéristique prédicat.

L'ajout de plusieurs caractéristiques dans un CRF n'améliore pas toujours la prédiction. Par conséquent, le bon choix des caractéristiques est nécessaire pour avoir une bonne prédiction. D'ailleurs, les caractéristiques d'un CRF linéaire sont basées sur des attributs. Ces attributs seront appliqués sur des jetons et retournent comme résultats une valeur numérique.

1.5 Ingénierie des caractéristiques

L'ingénierie de caractéristiques est un processus qui permet de sélectionner et d'extraire des attributs observables ou dérivés à partir des données. En effet, ce processus a un impact très important sur la performance de l'apprentissage. Ainsi, les attributs que nous ajoutons au modèle peuvent soit augmenter la précision soit la diminuer. Nous présentons dans ce qui suit quelques attributs que nous pouvons utiliser dans un CRF. Notons que les caractéristiques d'un modèle CRF sont basées sur les attributs des observations.

L'étude d'ingénierie de caractéristiques ci-dessous est basée sur les travaux élaborés dans (Chieu et Ng, 2003) et (Sutton *et al.*, 2012).

1.5.1 Les attributs prédicats

- Information à partir d'un jeton : Ces attributs sont déduits à partir d'un jeton donné. Par exemple, le jeton contient un ou plusieurs chiffres ; le jeton contient un caractère spécial, etc. (Chieu et Ng, 2002).

- Entre guillemets / crochets : Un attribut qui permet de vérifier si le jeton est entre guillemets ou crochets.

- En majuscule : Un attribut qui permet de vérifier si le jeton est en majuscule ou pas.

- Contient majuscule : Un attribut qui permet de vérifier si le jeton contient au moins un caractère en majuscule.
- Les attributs de mise en page : Ces attributs spécifient la position d'un jeton dans une séquence (au début, au milieu ou à la fin).

1.5.2 Les attributs numériques

- Occurrence d'un jeton : Un attribut qui permet de retourner le nombre de fois où le jeton apparaît dans un dataset.
- Jeton en majuscule : Un attribut qui permet de retourner le nombre de jetons où tous les caractères sont en majuscule dans un dataset.
- Occurrence d'un jeton en minuscule : Un attribut qui permet de retourner le nombre de jetons où tous les caractères sont en minuscule dans un dataset.
- Caractères spéciaux dans un jeton : Un attribut qui permet de retourner le nombre de caractères spéciaux trouvés dans un jeton donné.

1.5.3 Les attributs catégoriques

- Suffixes des jetons : Un attribut qui permet de retourner un suffixe de trois lettres à partir d'un jeton donné.
- Identité d'un jeton : Un attribut qui permet de retourner le même jeton en minuscule.
- Bi-grammes d'un jeton : Un attribut qui permet de retourner une sous-séquence de deux caractères construits à partir d'un jeton donné.
- Catégorie d'un jeton : Un attribut qui permet de retourner la catégorie d'un jeton donné selon les listes établies (lexiques). Par exemple, le jeton

représente un auteur, un titre, une date, etc.

1.6 Induction des caractéristiques

Pour réduire le surapprentissage d'un CRF linéaire, plusieurs techniques ont été utilisées dans la littérature telles que l'utilisation de la régularisation $L2$, la régularisation $L1$ et la régularisation élastique. À ce propos, l'étude de la régularisation est réalisée en se basant sur les travaux de (Sutton *et al.*, 2012), (Peng et McCallum, 2006) et (Lavergne *et al.*, 2010).

Prenons le cas d'un modèle CRF linéaire avec les paramètres $\Lambda = (\lambda_i)_{1 \leq i \leq F}$. Ces paramètres peuvent être estimés en minimisant le logarithme de la vraisemblance. Ainsi, le log de la vraisemblance de l'entraînement $\{(x_i, y_i) : i = 1,...M\}$ peut être défini par :

$$L_\Lambda = -\sum_{i=1}^{M} \log P_\Lambda(y_i \mid x_i) \tag{1.17}$$

La régularisation par $L2$ repose sur le poids de la fonction caractéristique λ_i et sur le paramètre de régularisation c_2 (constante positive). Ainsi, avec le régularisateur $L2$, le logarithme de la vraisemblance sera déterminé par :

$$L_\Lambda = -\sum_{i=1}^{M} \log P_\Lambda(y_i \mid x_i) + c_2 \sum_{i=1}^{F} \lambda_i^2 \tag{1.18}$$

Sachant que M représente le nombre d'échantillons dans l'ensemble d'entraînement.

Un autre choix de régularisation consiste à utiliser la norme $L1$. En effet, la régularisation par $L1$ repose sur le poids de la fonction caractéristique λ_i et sur le paramètre de régularisation c_1 qui est équivalent à c_2 dans le régularisateur $L2$.

Le logarithme de la vraisemblance sera donc présenté par :

$$L_\Lambda = -\sum_{i=1}^{M} \log P_\Lambda(y_i \mid x_i) + c_1 \sum_{i=1}^{F} |\lambda_i| \tag{1.19}$$

Sachant que F est le nombre de fonctions caractéristiques.

La régularisation avec la norme $L1$ favorise la rareté des paramètres appris. Autrement dit, la plupart des λ_i sont égales à 0. En pratique, les modèles qui utilisent un régularisateur $L1$ ont plus ou moins la même précision que les modèles qui utilisent un régularisateur $L2$ (Lavergne *et al.*, 2010).

À part la régularisation avec $L1$ ou $L2$, il existe un autre type de régularisation nommé la régularisation élastique (Zou et Hastie, 2005). En effet, cette régularisation correspond à un modèle de régression linéaire formé par les deux régularisateurs $L1$ et $L2$. À cet égard, la régularisation élastique a donné en pratique de meilleurs résultats par rapport à la régularisation faite avec la norme $L2$ (Täckström *et al.*, 2013). D'ailleurs, avec la régularisation élastique ($L1$+$L2$), le logarithme de la vraisemblance sera déterminé comme suit :

$$L_\Lambda = -\sum_{i=1}^{M} \log P_\Lambda(y_i \mid x_i) + c_1 \sum_{i=1}^{F} |\lambda_i| + c_2 \sum_{i=1}^{F} \lambda_i^2 \tag{1.20}$$

Sachant que c_1 et c_2 sont deux paramètres de régularisation déterminés par la validation croisée.

1.7 Mesure des performances

Pour bien évaluer un modèle d'étiquetage de séquences, nous mesurons au début les performances par type de segment. Ensuite, nous allons mesurer la performance globale du système. Ainsi, l'évaluation de l'étiquetage sera élaborée avec des métriques qui ont été utilisées souvent dans la littérature telles que la précision, le rappel et la F-mesure (Peng et McCallum, 2006).

1.7.1 Mesure de la performance par type de segment

Sachant que :

- $\mathrm{EVS_t}$: L'ensemble de vrais segments de type t (un des labels) ;
- $\mathrm{ESS_t}$: L'ensemble des segments sélectionnés pour le type t (un des labels).

La précision est la proportion des vrais segments sélectionnés parmi les segments sélectionnés :

$$\mathrm{Precision_t} = \frac{\mid \mathrm{EVS_t} \cap \mathrm{ESS_t} \mid}{\mid \mathrm{ESS_t} \mid} \tag{1.21}$$

Le rappel est la proportion des segments correctement sélectionnés parmi les vrais segments :

$$\mathrm{Rappel_t} = \frac{\mid \mathrm{EVS_t} \cap \mathrm{ESS_t} \mid}{\mid \mathrm{EVS_t} \mid} \tag{1.22}$$

La F-mesure (F_1) est la moyenne harmonique de la précision et le rappel :

$$\mathrm{F_{1t}} = \frac{2 \times \mathrm{Precision_t} \times \mathrm{Rappel_t}}{\mathrm{Precision_t} + \mathrm{Rappel_t}} \tag{1.23}$$

La précision globale (accuracy) peut être définie par (Han *et al.*, 2003) :

$$\mathrm{Accuracy_t} = \frac{\sum_t \mid \mathrm{EVS_t} \cap \mathrm{ESS_t} \mid}{\sum_t \mid \mathrm{ESS_t} \mid} = \frac{\mid \mathrm{EVS} \cap \mathrm{ESS} \mid}{\mid \mathrm{ESS} \mid} \tag{1.24}$$

1.7.2 Mesure de la performance par type de jeton

Sachant que :

- $\mathrm{EVJ_t}$: L'ensemble de vrais jetons de type t ;

- $\mathrm{EJS_t}$: L'ensemble des jetons sélectionnés pour le type t.

La précision est la proportion des vrais jetons sélectionnés parmi les jetons sélectionnés :

$$\mathrm{Precision_t} = \frac{\mid \mathrm{EVS_t} \cap \mathrm{EJS_t} \mid}{\mid \mathrm{EJS_t} \mid} \tag{1.25}$$

Le rappel est la proportion des jetons correctement sélectionnés parmi les vrais jetons :

$$\mathrm{Rappel_t} = \frac{\mid \mathrm{EVJ_t} \cap \mathrm{EJS_t} \mid}{\mid \mathrm{EVJ_t} \mid} \tag{1.26}$$

La précision globale (accuracy) peut être définie par (Han *et al.*, 2003) :

$$\mathrm{Accuracy_t} = \frac{\sum_t \mid \mathrm{EVJ_t} \cap \mathrm{EJS_t} \mid}{\sum_t \mid \mathrm{EJS_t} \mid} = \frac{\mid \mathrm{EVJ} \cap \mathrm{EJS} \mid}{\mid \mathrm{EJS} \mid} \tag{1.27}$$

Notons que les métriques par jetons ne reflètent pas la réalité puisque l'étiquetage d'un segment devrait être considéré incorrect si au moins un jeton est mal étiqueté. Malgré tout, elles étaient utilisées dans la littérature lors de la mesure de performance. Nous allons l'utiliser pour se comparer avec les résultats de la littérature.

1.7.3 Mesure de la performance globale d'un modèle

Pour mesurer la performance globale d'un modèle, nous allons calculer la moyenne des précisions, la moyenne des rappels et la moyenne des F-mesures de tous les segments comme suit :

Sachant que :

- $\mid S \mid$: La cardinalité de l'ensemble des types de segments S.

La moyenne des précisions de tous les segments :

$$\text{Precision} = \frac{1}{\mid S \mid} \sum_{t} \text{Precision}_t \tag{1.28}$$

La moyenne des rappels de tous les segments :

$$\text{Rappel} = \frac{1}{\mid S \mid} \sum_{t} \text{Rappel}_t \tag{1.29}$$

La moyenne des F-mesures (F_1) de tous les segments :

$$\text{F}_1 = \frac{1}{\mid S \mid} \sum_{t} \text{F}_{1t} \tag{1.30}$$

1.8 Conclusion

Dans ce chapitre, nous avons présenté la tâche d'extraction d'information basée sur l'étiquetage de séquences. Ensuite, nous avons introduit les modèles d'étiquetage de séquences les plus utilisés dans la littérature (HMM, CRF). Nous avons relaté par la suite les ingénieries de caractéristiques en mettant l'accent sur leur importance afin d'améliorer les résultats d'un étiquetage. Finalement, nous avons présenté les métriques utilisées pour mesurer les performances par jeton et par segment.

Dans le prochain chapitre, nous allons couvrir quelques travaux liés à l'étiquetage de séquences. Premièrement, nous allons discuter de ceux liés à la reconnaissance d'entités nommées. Ensuite, nous allons nous pencher sur les travaux les plus importants associés à la segmentation des références bibliographiques. Finalement, nous allons détailler d'autres approches qui traitent l'extraction des champs à partir d'un article.

CHAPITRE II

REVUE DE LITTÉRATURE SUR L'ÉTIQUETAGE DE SÉQUENCES

L'étiquetage de séquences est l'une des techniques les plus importantes dans l'extraction d'information. Ainsi, cette étude permet d'extraire et d'étiqueter les champs à partir des données textuelles. L'étiquetage de séquences se base sur deux étapes. La première étape consiste à identifier les jetons par tokenisation, tandis que la deuxième étape consiste à attribuer à chaque jeton l'étiquette correspondante.

Dans la littérature, l'étiquetage de séquences est utilisé pour la reconnaissance d'entités nommées. D'autres travaux l'utilisent pour extraire des champs à partir d'un article comme les mots-clés, numéro de téléphone, adresse courriel, etc. Plusieurs chercheurs exploitent aussi ce concept afin d'extraire les champs à partir des références bibliographiques telles que les auteurs, le titre, pages, date, etc.

Nous nous penchons, dans ce chapitre, sur quelques travaux de la littérature qui traitent de l'étiquetage de séquences. Nous avons choisi d'étudier ceux liés à la reconnaissance d'entités nommées, l'extraction des champs, à partir d'un article et de la segmentation des références bibliographiques.

2.1 La reconnaissance d'entités nommées

Dans la littérature, plusieurs recherches liées à l'étiquetage de séquences ont été effectuées pour la reconnaissance d'entités nommées. En effet, cette tâche permet de rechercher des segments de texte catégorisables dans des classes telles que les noms de personnes, noms d'organisations, noms de lieux, etc. Ces objets sont appelés des entités nommées. Pour une meilleure explication, l'entité nommée est indépendante du texte et elle représente quelque chose qui existe. Prenons l'exemple du mot «Montréal» qui représente l'entité nommée «ville». Le même mot peut représenter le champ «location», par exemple, dans un autre contexte.

Parmi les recherches qui nous intéressent au sujet de la reconnaissance d'entités nommées, nous trouvons l'article qui a été publié par (Sang et De Meulder, 2003). Dans cet article, les auteurs ont introduit le jeu de données CoNLL-2003 qui contient des données en anglais et d'autres en allemand. Dans ce dataset, les phrases sont séparées par des lignes vides et chaque ligne contient quatre champs : le mot (jeton), sa catégorie grammaticale, la vraie étiquette et l'étiquette trouvée.

Plusieurs techniques ont été appliquées sur le dataset CoNLL-2003. À ce propos, (Bender *et al.*, 2003), (Chieu et Ng, 2003) et (Curran et Clark, 2003) ont appliqué le modèle de Markov à maximum d'entropie. D'ailleurs, les auteurs (Chieu et Ng, 2003) ont présenté une approche d'entropie maximale qui n'utilise pas seulement les caractéristiques locales déduites à partir d'un jeton, mais aussi d'autres occurrences de chaque jeton du même dataset pour en extraire des caractéristiques globales qui améliorent les résultats.

Lors de l'étiquetage d'une phrase, le jeton de l'entité nommée au début va avoir l'étiquette *(B-)*, les jetons du milieu vont recevoir l'étiquette *(C-)* et celui de la fin va avoir l'étiquette *(L-)*. Cependant, l'entité nommée composée d'un seul jeton se

verra attribuer l'étiquette *(U-)*. Les autres mots de la phrase vont avoir l'étiquette *(O)*. Une probabilité de transition binaire est définie pour éviter de trouver une séquence d'attributions non admissible (Exemple : B-Loc n'est jamais suivi par L-Per).

Les auteurs (Chieu et Ng, 2003) ont utilisé aussi l'algorithme de Viterbi (Forney, 1973) afin de sélectionner la séquence d'étiquettes ayant la probabilité la plus élevée.

Afin d'évaluer les performances, les auteurs ont testé leur méthode sur le dataset CoNLL-2003 selon deux différents systèmes. Un système ME1 qui utilise des caractéristiques qui proviennent seulement des données de validation et un deuxième système nommé ME2 qui utilise des caractéristiques supplémentaires dérivées à partir des listes de lexiques. En effet, le modèle d'entropie maximale a donné de bons résultats avec les données de test en anglais. Le modèle ME2 a plus précisément donné une F-mesure globale par segment qui est égale à 88,31%. Toutefois, le modèle ME1 a donné seulement 86,84%. D'un autre côté, ce modèle a donné des résultats moins importants avec les données de test en allemand. Ainsi, le modèle ME2 a donné une F-mesure globale par segment qui est égale à 65.67%. En revanche, le modèle ME1 a donné seulement 61.90%.

Les expériences ont montré de même que le modèle ME2 a donné de meilleurs résultats avec les données de validation en anglais (F-mesure globale par segment est égale à 93.01%) qu'avec les données de validation en allemand (F-mesure globale par segment est égale à 63.61%).

Néanmoins, d'autres travaux tels que ceux réalisés par ((Florian *et al.*, 2003) et (Klein *et al.*, 2003)) ont utilisé le modèle d'entropie maximale en combinaison avec d'autres techniques telles que les classifieurs linéaires, les modèles de Markov cachés (HMM) et l'apprentissage par transformation. D'autres méthodes d'extrac-

tion ont été utilisées telles que les machines à vecteurs de support (Mayfield *et al.*, 2003) ainsi que les champs aléatoires conditionnels (McCallum et Li, 2003).

Les meilleurs résultats obtenus avec CoNLL-2003 ont été publiés dans les travaux de (Florian *et al.*, 2003). En effet, les auteurs de cet article ont proposé une approche qui combine le modèle d'entropie maximale, le HMM et d'autres techniques d'apprentissage avec un classifieur linéaire robuste. Dans cette référence, les auteurs ont trouvé une F-mesure globale par segment qui est égale à 88.76% pour les données de test en anglais et 72.41% pour les données de test en allemand. D'autre part, (Chieu et Ng, 2003) ont trouvé aussi de bons résultats en appliquant le modèle de Markov à maximum d'entropie. Ces auteurs ont plus précisément réussi à obtenir une F-mesure globale par segment qui est égale à 88,31% pour les données de test en anglais et 93.01% pour les données d'entraînement en anglais.

Lors de l'étiquetage des entités nommées à partir du dataset CoNLL-2003, le modèle CRF proposé dans la référence (McCallum, 2002) a également donné de bons résultats. Effectivement, la F-mesure globale par segment d'un CRF est égale à 87,5%. Nous présentons dans la Figure 2.1 (Sang et De Meulder, 2003) les différents résultats (précision, rappel et F-mesure globale) par segment que les auteurs ont trouvés avec le dataset CoNLL-2003.

English test	Precision	Recall	$F_{\beta=1}$
Florian	88.99%	88.54%	88.76±0.7
Chieu	88.12%	88.51%	88.31±0.7
Klein	85.93%	86.21%	86.07±0.8
Zhang	86.13%	84.88%	85.50±0.9
Carreras (b)	84.05%	85.96%	85.00±0.8
Curran	84.29%	85.50%	84.89±0.9
Mayfield	84.45%	84.90%	84.67±1.0
Carreras (a)	85.81%	82.84%	84.30±0.9
McCallum	84.52%	83.55%	84.04±0.9
Bender	84.68%	83.18%	83.92±1.0
Munro	80.87%	84.21%	82.50±1.0
Wu	82.02%	81.39%	81.70±0.9
Whitelaw	81.60%	78.05%	79.78±1.0
Hendrickx	76.33%	80.17%	78.20±1.0
De Meulder	75.84%	78.13%	76.97±1.2
Hammerton	69.09%	53.26%	60.15±1.3
Baseline	71.91%	50.90%	59.61±1.2

German test	Precision	Recall	$F_{\beta=1}$
Florian	83.87%	63.71%	72.41±1.3
Klein	80.38%	65.04%	71.90±1.2
Zhang	82.00%	63.03%	71.27±1.5
Mayfield	75.97%	64.82%	69.96±1.4
Carreras (b)	75.47%	63.82%	69.15±1.3
Bender	74.82%	63.82%	68.88±1.3
Curran	75.61%	62.46%	68.41±1.4
McCallum	75.97%	61.72%	68.11±1.4
Munro	69.37%	66.21%	67.75±1.4
Carreras (a)	77.83%	58.02%	66.48±1.5
Wu	75.20%	59.35%	66.34±1.3
Chieu	76.83%	57.34%	65.67±1.4
Hendrickx	71.15%	56.55%	63.02±1.4
De Meulder	63.93%	51.86%	57.27±1.6
Whitelaw	71.05%	44.11%	54.43±1.4
Hammerton	63.49%	38.25%	47.74±1.5
Baseline	31.86%	28.89%	30.30±1.3

Figure 2.1: Résultats des différentes approches avec CoNLL-2003.

2.2 Segmentation des références bibliographiques

Dans la littérature, plusieurs travaux qui traitent de l'étiquetage de séquences ont été publiés. Les travaux qui nous intéressent le plus dans cette partie sont ceux liés à la segmentation des références bibliographiques.

Plusieurs approches ont été appliquées sur le dataset Cora (McCallum *et al.*, 2000). En effet, Cora contient cinq cents références bibliographiques qui ont été étiquetées manuellement avec treize attributs tels que : auteur, titre, pages, volume, date, etc. À cet égard, (Hetzner, 2008) a présenté une approche d'extraction basée sur les modèles de Markov cachés.

Pour son évaluation, l'auteur a testé cette méthode sur le dataset Cora en calculant la précision, le rappel et la F-mesure par segment et par jeton pour chaque étiquette avant et après la suppression de la ponctuation. En outre, avant la suppression de la ponctuation, la macro moyenne de F-mesure par segment est égale à 74,7%. Par ailleurs, celle par jeton est égale à 84,7%. Cependant, après la suppression de la ponctuation, la macro moyenne de F-mesure par segment est égale à 79.8%, tandis que celle par jeton est égale à 86.6%.

Une autre approche réalisée par (Cortez *et al.*, 2010) a été présentée afin d'étiqueter le dataset Cora. Cette fois, les auteurs ont présenté une approche non-supervisée nommée *On-Demand Unsupervised Learning for Information Extraction* (ONDUX).

La méthode ONUDX est dite non-supervisée car elle utilise une base de connaissances qui regroupe l'ensemble de paires des attributs et leurs occurrences. Cette méthode se déroule sur trois étapes : l'étape de blocage, l'étape d'appariement et l'étape de renforcement. Dans l'étape de blocage, la séquence va être divisée en un ensemble de segments. En effet, cette étape se base sur la cooccurrence des

termes d'un même attribut selon la base de connaissances.

L'étape d'appariement permet d'associer à chaque segment trouvé dans l'étape de blocage une étiquette représentée dans la base de connaissance. Pour ce faire, ONDUX utilise des caractéristiques basées sur le contenu. Pour détecter les valeurs textuelles dans un segment, Ondux utilise une fonction appelée *Attribute Frequency* (AF) pour estimer la similarité entre un segment à étiqueter et les occurrences des attributs dans la base de connaissances. Pour détecter les valeurs numériques, Ondux utilise une fonction appelée *Numeric Matching* (NM), afin de mesurer la similarité entre un segment à étiqueter et les occurrences des attributs dans la base de connaissances. Finalement, cette approche sollicite des fonctions binaires à base d'expressions régulières pour détecter les formats spécifiques des URL et des adresses courriel.

La dernière étape est l'étape de renforcement. Celle-ci vérifie les résultats d'étiquetage trouvés dans l'étape d'appariement. À cet égard, les segments qui ne sont pas étiquetés ou les segments qui sont mal étiquetés vont être corrigés. Le pré-étiquetage de l'étape d'appariement peut être utilisé pour induire automatiquement des caractéristiques basées sur la structure qui sont liées avec la transition et la position des valeurs d'attributs dans une séquence. La vérification est effectuée à travers un modèle nommé *Positioning and Sequencing Model* (PSM). En effet, le PSM est un modèle graphique probabiliste formé par deux matrices, soit une matrice de transition (T) qui stocke la probabilité de transition d'un état vers un autre et une matrice de position (P) qui stocke la probabilité d'observation d'un état dans une séquence (position de l'état dans la séquence).

Pour évaluer les performances, les auteurs ont appliqué cette méthode sur le jeu de données Cora en mesurant la précision, le rappel et la F-mesure par segment pour chaque étiquette. Avec ONDUX, la macro moyenne de F-mesure par segment est

égale à 92,1%. Par ailleurs, elle est égale à 90,14% avec un CRF supervisé selon leurs expérimentations.

Les auteurs (Cortez *et al.*, 2010) ont obtenu aussi de bons résultats sur le dataset PersonalBib (Mansuri et Sarawagi, 2006). En effet, ce dataset contient trois cent quatre-vingt-quinze références bibliographiques et sept attributs à extraire (auteur, date, pages, etc.). La méthode ONDUX a donné une macro moyenne de F-mesure par segment qui est égale à 88,8%.

Afin d'étiqueter les références bibliographiques du dataset Cora, les auteurs (Councill et Kan, 2008) ont présenté une autre approche nommée ParsCit. Cette méthode se base sur un modèle de champ aléatoire conditionnel (CRF). L'approche proposée est fournie avec des utilitaires permettant de l'exécuter en tant que service Web ou en tant qu'utilitaire autonome.

La méthode ParsCit se base sur trois étapes : l'étape de prétraitement, l'étape de post-traitement et l'étape d'extraction des contextes de référence. Dans l'étape de prétraitement et pour une extraction correcte, les documents seront convertis tout d'abord en texte brut codé à l'aide de UTF8. Par la suite, cette méthode capte les références bibliographiques dans le texte en utilisant des heuristiques. La méthode débute en cherchant la partie qui contient les références bibliographiques en repérant des mots tels que «Bibliographie», «Références» ou «Note». Une fois que le point de départ des références a été trouvé, ParsCit commence à chercher la fin des références en utilisant une liste d'étiquettes telles que les figures, les tableaux ou la fin d'un document.

Lorsque les références bibliographiques sont détectées, ParsCit utilise plusieurs heuristiques pour capter le début et la fin d'une référence bibliographique. Une fois que cela est fait, nous pouvons appliquer la méthode CRF sur l'ensemble des références bibliographiques. Les données dans les différents attributs sont présentées

selon plusieurs formats et dans différents ordres. Dans l'étape de post-traitement, chaque attribut sera normalisé. Par exemple, l'auteur (M.-Y. Kan and I. G. Councill) va être normalisé et présenté par (M-Y Kan and I G Councill), tandis que le volume (vol. 5) va être normalisé et présenté par (5), etc.

Pour l'évaluation, les auteurs ont appliqué ParsCit sur les datasets Cora et CiteSeer (Lawrence *et al.*, 1999). Avec Cora, cette méthode a donné une micro-moyenne de F-mesure par jeton qui est égale à 95,00% et qui est inférieure à la macro-moyenne de F-mesure par jeton (91,00%) trouvée dans (Peng et McCallum, 2004). La méthode ParsCit a donné aussi de bons résultats avec le dataset CiteSeer (une macro-moyenne de F-mesure par jeton qui est égale à 79,54%).

D'un autre côté, (Peng et McCallum, 2006) ont présenté les résultats trouvés avec les champs aléatoires conditionnels sur le dataset Cora. En effet, le modèle CRF a donné de meilleurs résultats par rapport au modèle HMM. L'expérience réalisée dans cette référence contient 350 références dans les données d'entraînement et 150 références dans les données de test. Les résultats sont calculés par jeton. Avec les HMM, les auteurs ont précisément obtenu une macro-moyenne de F-mesure par jeton qui est égale à 77,6%. En revanche, avec le CRF, ils ont obtenu une macro-moyenne de F-mesure par jeton qui est égale à 91,5%. Lors de l'étiquetage du champ *éditeur* par exemple, nous remarquons une amélioration importante des résultats de F-mesure par jeton avec CRF (87,7%) comparés à ceux trouvés par un modèle HMM (70,8%).

2.3 Extraction des champs à partir d'un article

Dans la littérature, plusieurs recherches liées à l'étiquetage de séquences ont été effectuées pour extraire des champs à partir d'un article. À ce sujet, (Seymore *et al.*, 1999) ont présenté un modèle de Markov caché pour la tâche d'extraction

d'informations à partir des en-têtes de documents de recherche en informatique. Dans cet ordre d'idées, l'en-tête d'un article de recherche contient tous les mots qui précèdent le corps de l'article. Le modèle HMM présenté dans la référence (Seymore *et al.*, 1999) se base sur deux aspects. Le premier aspect consiste à étudier, à partir de données, la structure du modèle d'apprentissage, alors que le deuxième aspect consiste à introduire d'autres données dans la formation des HMM. Cette méthode est proposée dans le but d'améliorer la qualité d'extraction de métadonnées des bibliothèques numériques EbizSearch (Petinot *et al.*, 2003).

Les auteurs (Seymore *et al.*, 1999) ont réalisé plusieurs expériences. Ces dernières ont montré que le HMM a donné de bons résultats lors de l'extraction des champs à partir du dataset EbizSearch. En effet, ce modèle a donné une précision globale (accuracy) par jeton qui est égale à 92,9%.

Dans d'autres travaux, les chercheurs (Han *et al.*, 2003) ont présenté une autre méthode d'apprentissage automatique basée sur les machines à vecteurs de support (SVM), qui sont des modèles géométriques introduits par (Cortes et Vapnik, 1995). Ces modèles sont utilisés pour l'extraction de métadonnées à partir des en-têtes de documents de recherche. L'extraction des caractéristiques est effectuée en utilisant les caractéristiques spécifiques aux mots et aux lignes. D'ailleurs, les auteurs ont réalisé cette méthode pour améliorer la qualité d'extraction de métadonnées des bibliothèques numériques *EbizSearch* (Petinot *et al.*, 2003) et *Citeseer* (Lawrence *et al.*, 1999). En effet, les métadonnées sont des données qui en décrivent d'autres. En outre, ces dernières facilitent la recherche et la manipulation d'instances de données particulières.

Dans cette perspective, l'expérience réalisée par (Han *et al.*, 2003) contient 500 en-têtes dans les données d'entraînement et 435 en-têtes dans les données de test. Cette méthode a donné de meilleurs résultats par rapport à un HMM. La méthode

basée sur un SVM atteint notamment une précision globale (accuracy) par jeton qui est égale à 92,9%, tandis que la précision globale par jeton d'un modèle HMM était égale à 90,1% (Seymore *et al.*, 1999).

2.4 Synthèse des travaux

Nous résumons les travaux que nous avons étudiés tout au long de ce chapitre lors de l'étiquetage de séquences dans le Tableau 2.1.

Tableau 2.1: Synthèse des travaux.

Année	Auteurs	Tâches effectuées	Méthode	Données	Résultats
2008	(Hetzner, 2008)	Nouvelle méthode pour l'extraction de métadonnées de référence.	HMM	Cora	Meilleurs résultats par jeton : macro moyenne de F-mesure=86.6%. Meilleurs résultats par segment : macro moyenne de F-mesure=79.8%.
2003	(Chieu et Ng, 2003)	Reconnaissance d'entités nommées.	Modèle d'entropie maximale	CoNLL-2003	Données de test en anglais : F-mesure globale par segment=88,31%. Données de validation en anglais : F-mesure globale par segment=93.01%.
2010	(Cortez et al., 2010)	Une méthode non supervisée pour l'étiquetage de séquences.	ONDUX	Cora et PersonalBib	Macro moyenne de F-mesure par segment=92,1% pour Cora. Macro moyenne de F-mesure par segment=88,8% pour PersonalBib.
2008	(Councill et al., 2008)	Un package d'analyse de chaîne de référence CRF open source.	ParsCit	Cora et CiteSeer	Micro-moyenne de F-mesure par jeton=95,00% pour Cora. Macro-moyenne de F-mesure par jeton=79,54% pour CiteSeer.
2006	(Peng et McCallum, 2006)	Extraction d'informations à partir de documents de recherche.	CRF	Cora et CiteSeer	Macro-moyenne de F-mesure par jeton=91,5% pour CORA. Macro-moyenne de F-mesure par jeton=93,9% pour CiteSeer.

2002	(McCallum, 2002)	Nouvelle méthode automatique d'induction de caractéristique.	CRF	CoNLL-2003	F-mesure globale par segment=87,5% pour CoNLL-2003.
2003	(Han et al., 2003)	Extraction automatique de métadonnées de document.	SVM	EbizSearch et Citeseer	Précision globale (accuracy) par jeton=92,9%.
1999	(Seymore et al., 1999)	Apprentissage de la structure de modèle de Markov cachée pour l'extraction d'informations.	HMM	EbizSearch	Précision globale (accuracy) par jeton=92,9%.
2003	(Florian et al., 2003)	Reconnaissance d'entités nommées via une combinaison de classificateurs.	Combinaison entre le modèle d'entropie maximale, HMM et d'autres techniques	CoNLL-2003	F-mesure globale par segment=88.76% pour les données de test en anglais.

Après avoir étudié quelques travaux de la littérature qui traitent de l'étiquetage de séquences, nous avons choisi d'utiliser les champs aléatoires conditionnels (CRF) afin de segmenter les références bibliographiques du dataset Cora. Le choix a été effectué en fonction de l'importance et de la performance du CRF. En effet, nous pensons que les résultats trouvés dans la littérature peuvent être améliorés. En outre, l'importance du modèle et le bon choix des caractéristiques rendent le CRF plus performant.

2.5 Conclusion

Dans ce chapitre, nous avons présenté dans un premier temps quelques travaux de la littérature pour la reconnaissance d'entités nommées. Dans un second temps, nous avons décrit d'autres travaux liés à la segmentation des références bibliographiques. Nous avons étudié aussi quelques travaux qui utilisent l'étiquetage de séquences pour extraire des champs à partir d'un article. Finalement, nous avons exposé un tableau qui résume tous les travaux étudiés précédemment.

Dans le prochain chapitre, nous allons utiliser les champs aléatoires conditionnels afin de segmenter les références bibliographiques du dataset Cora. Nous tenons toujours à rappeler que notre objectif dans un premier temps est d'améliorer les résultats trouvés dans la littérature. Une bonne sélection des caractéristiques et l'utilisation de la régularisation sont les deux techniques que nous allons exploiter pour atteindre notre objectif.

CHAPITRE III

ÉTIQUETAGE DU DATASET CORA AVEC LES CHAMPS ALÉATOIRES CONDITIONNELS

Ce chapitre présente une approche d'étiquetage de séquences qui se base sur les champs aléatoires conditionnels (CRF) afin d'étiqueter les références bibliographiques du dataset Cora (McCallum *et al.*, 2000). Dans un premier temps, nous commençons par introduire notre approche à travers les démarches et les techniques utilisées telles que la tokenisation, l'étiquetage BIO et le choix des caractéristiques. Ensuite, nous introduisons l'algorithme d'optimisation adopté, la régularisation et la méthode d'évaluation. Finalement, nous concluons ce chapitre par les expériences élaborées et une comparaison de nos résultats avec ceux de la littérature.

3.1 Approche proposée

Nous présentons dans cette section les démarches et les techniques que nous avons utilisées. Notre approche se base sur les champs aléatoires conditionnels (CRF) afin d'étiqueter les références bibliographiques du dataset Cora. L'objectif est de fournir un modèle performant qui améliore les résultats de l'état de l'art pour le dataset Cora.

3.1.1 Tokenisation des données

La première étape de notre approche est la tokenisation du dataset Cora. En effet, cette étape permet la décomposition d'une séquence en un ensemble de jetons. Pour ce faire, plusieurs techniques existent dans la littérature telles que la tokenisation par caractères spéciaux et la tokenisation par espace. Nous présentons dans ce qui suit un exemple de tokenisation par espace d'une référence bibliographique du dataset Cora.

[author **Witten,**] [author **I.**] [author **H.,**] [author **Neal**] [author **R.**] [author **M.,**] [author **and**] [author **Cleary**] [author **J.**] [author **G.**] [date **(1987).**] [title **Arithmetic**] [title **coding**] [title **for**] [title **data**] [title **compression.**] [journal **Communications**] [journal **of**] [journal **the**] [journal **ACM.**] [volume **30,**] [pages **520-540.**]

Nous présentons aussi un exemple de tokenisation par caractères spéciaux de la même référence bibliographique.

[author **Witten**] [author **,**] [author **I**] [author **.**] [author **H**] [author **.,**] [author **Neal**] [author **R**] [author **.**] [author **M**] [author **.,**] [author **and**] [author **Cleary**] [author **J**] [author **.**] [author **G**] [author **.**] [date **(**] [date **1987**] [date **).**] [title **Arithmetic**] [title **coding**] [title **for**] [title **data**] [title **compression**] [title **.**] [journal **Communications**] [journal **of**] [journal **the**] [journal **ACM**] [journal **.**] [volume **30**] [volume **,**] [pages **520**] [pages **-**] [pages **540**] [pages **.**]

Nous avons testé notre approche avec ces deux derniers et nous avons décidé d'utiliser dans notre démarche la tokenisation par espace, vu la différence de performance. En effet, nous avons réussi à obtenir de meilleurs résultats avec la tokenisation par espace.

3.1.2 Étiquetage BIO

L'étape qui suit la tokenisation du dataset Cora est la préparation des données. En effet, Cora contient cinq cents références bibliographiques. Pour ce faire, nous allons utiliser tout au long de nos expériences trois cent cinquante références pour les données d'entraînement et cent cinquante références pour les données de tests. Le choix des références bibliographiques est effectué aléatoirement.

La prochaine étape est le choix de la notation de l'étiquetage du dataset Cora. Pour cette étape, nous avons choisi d'utiliser la notation BIO que nous avons présentée précédemment afin d'étiqueter chaque jeton, et ce en raison du gain de la performance, puisque, après plusieurs tests, nous avons remarqué que cette notation nous permet d'avoir de meilleures performances. De plus, elle est très utilisée dans la littérature lors de l'étiquetage de séquences.

Le dataset Cora possède 13 attributs. Par conséquent, nous allons avoir 26 attributs avec la notation *BIO*. Autrement dit, chaque attribut *attribut* disposera de deux étiquettes différentes *B-attribut* et *I-attribut*.

3.1.3 Choix des caractéristiques

Le choix de caractéristiques est très important pour l'apprentissage. En effet, un modèle avec trop de caractéristiques engendre un problème de surapprentissage. Cependant, un modèle qui ne contient pas assez de caractéristiques cause un problème de sous-apprentissage. Par conséquent, la sélection et le bon choix des caractéristiques sont très importants pour obtenir de bons résultats de prédiction.

Nous allons utiliser dans notre approche des caractéristiques qui sont inspirées des travaux réalisés par (Peng et McCallum, 2006), (Councill et Kan, 2008) et (Chieu et Ng, 2003). Le choix est fait en gardant seulement les caractéristiques

qui augmentent la performance du CRF. Après plusieurs tests, nous avons réussi à retenir une liste de caractéristiques qui donne de bons résultats d'étiquetage. Cette liste est présentée dans le Tableau 3.1.

Tableau 3.1: Liste des caractéristiques utilisées dans notre approche.

Word[-3 :]	Retourne les trois derniers caractères d'un jeton.
Word[-2 :]	Retourne les deux derniers caractères d'un jeton.
Word[-1 :]	Retourne le dernier caractère d'un jeton.
Word[0 :1]	Retourne le premier caractère d'un jeton.
Word[0 :2]	Retourne les deux premiers caractères d'un jeton.
Isupper	Retourner vrai si tous les caractères d'un jeton sont en majuscule.
Istitle	Retourner vrai si le jeton est un titre.
Isdigit	Retourne vrai si tous les caractères d'un jeton sont des chiffres.
Initcap	Retourne vrai si le jeton commence par une lettre majuscule.
ContainsDigits	Retourne vrai si le jeton contient au moins un chiffre.
Endwithdot	Retourne vrai si le jeton se termine par un point.
Endwithcomma	Retourne vrai si le jeton se termine par une virgule.
Endwith2POINT	Retourne vrai si le jeton se termine par deux points.
ContainsDot	Retourne vrai si le jeton contient un point.
ContainsDash	Retourne vrai si le jeton contient un tiret.
Acronym	Retourne vrai si le jeton est un acronyme.
LonelyInitial	Retourne vrai si le jeton est un initiale (tel que A.).
SingleChar	Retourne vrai si le jeton est un seul caractère.

CapLetter	Retourne vrai si le jeton contient un seul caractère en majuscule.
Punc	Retourne vrai si le jeton est une ponctuation.
WordItSelf	Retourne le jeton lui-même.
LineStart	Retourne le premier jeton de la séquence.
LineEnd	Retourne le dernier jeton de la séquence.
LineIn	Retourne les jetons au milieu de la séquence.
BIBTEXAUTHOR	Retourne vrai si le jeton appartient au champ lexical des auteurs.
BIBTEXDate	Retourne vrai si le jeton appartient au champ lexical des dates.
BIBTEXNotes	Retourne vrai si le jeton appartient au champ lexical des notes.
BIBTEXInst	Retourne vrai si le jeton appartient au champ lexical des institutions.
Word.lower	L'identité du jeton.

3.1.4 Algorithme d'optimisation du CRF

Nous avons choisi d'utiliser l'algorithme BFGS à mémoire limitée L-BFGS (Liu et Nocedal, 1989). En effet, cet algorithme est une version de l'algorithme BFGS découvert par *Broyden, Fletcher, Goldfarb et Shanno.*

L-BFGS est un algorithme d'optimisation qui fait partie de la famille des algorithmes quasi-Newton et qui est très utilisé en apprentissage automatique. En pratique, cet algorithme améliore le poids des caractéristiques très lentement au début du processus de formation, mais converge rapidement vers les poids des caractéristiques optimaux à la fin.

3.1.5 Régularisation

Nous avons choisi d'utiliser dans notre approche la régularisation afin d'éviter le surapprentissage dans le CRF. Plusieurs types de régularisation existent dans la littérature. Nous avons décidé d'appliquer la régularisation élastique étant donné la quantité des données à traiter.

La régularisation élastique est un modèle de régression linéaire formé par les deux régularisateurs $L1$ et $L2$. Avec ce type de régularisation, le modèle évite d'apprendre par cœur les données d'entraînement.

Pour minimiser le logarithme de la vraisemblance des données d'entraînement avec les deux régularisateurs $L1$ et $L2$, nous avons choisi d'utiliser l'algorithme d'optimisation (L-BFGS) que nous avons introduit précédemment. Avec cet algorithme, nous avons utilisé premièrement une variable $c1$ qui présente le coefficient de régularisation de $L1$ et une variable $c2$ qui présente le coefficient de régularisation de $L2$. Le choix des régularisateurs $c1$ et $c2$ est fait à l'aide de la validation croisée (Arlot, 2018) qui se base sur une technique d'échantillonnage et la méthode de recherche aléatoire (Bergstra et Bengio, 2012). Plusieurs types de validation croisée peuvent être utilisés. Nous avons choisi d'utiliser la validation croisée par k-ensembles puisqu'elle est très utilisée en apprentissage automatique.

Nous avons découpé au début les données d'entraînement en k ensembles à peu près égaux. Ensuite, nous avons fixé deux valeurs pour les deux régularisateurs $c1$ et $c2$ qui sont générées aléatoirement. La validation croisée consiste à sélectionner à chaque fois un des k ensembles qui sera utilisé comme ensemble de validation alors que les autres ensembles d'échantillons $k-1$ seront utilisés pour l'ensemble d'entraînement. Cette opération sera répétée k fois pour chaque couple ($c1$,$c2$) aléatoirement sélectionné en calculant sa performance moyenne. Nous retenons le

meilleur couple ($c1$,$c2$) pour rapporter la performance sur l'ensemble de test.

3.1.6 Méthode d'évaluation

Pour évaluer notre approche, nous nous sommes servi d'un script nommé Conlleval[1] qui est très utilisé dans la littérature et surtout lors de l'étiquetage de séquences.

Conlleval est disponible avec une version Perl ou Python. Il permet d'évaluer les résultats d'un modèle dont les données sont sous format BIO. Les résultats fournis par ce script sont par segment.

Pour mieux comprendre le fonctionnement de Conlleval, nous présentons dans la Figure 3.1 un premier exemple d'un fichier d'entrée pour Conlleval. Nous allons expliquer à travers cet exemple la façon dont les segments prédits sont composés.

	Vrai	Prédit
Aji,	B-author	B-author
McEliece,	I-author	I-author
Evaluation,	B-title	B-author
A.,	B-author	I-author
McCallum,	I-author	I-author

Figure 3.1: Premier exemple d'un fichier d'entrée pour le script Conlleval.

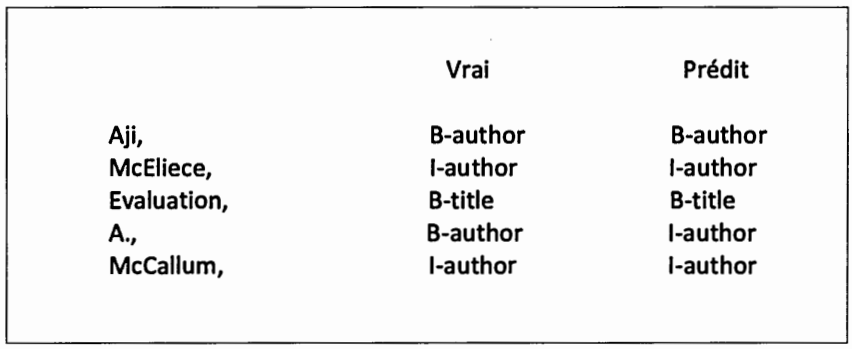
Dans le cas de la Figure 3.1, Conlleval comptabilise deux segments prédits. Le premier segment est (B-author, I-author) et le deuxième segment est (B-author, I-author, I-author) qui est un faux positif (segment qui a été inexactement classé comme positif). En effet, les segments prédits sont composés selon le principe que

1. https://www.clips.uantwerpen.be/conll2002/ner/bin/conlleval.txt

la plus longue chaîne de (I-author) commence par (B-author) ou (I-author).

Dans cet exemple, Conlleval comptabilise seulement 2 segments (2 segments auteur) parmi 3 segments (2 segments auteur, 1 segment titre) et un seul segment étiqueté correctement (segment auteur).

Pour mieux expliquer le concept, nous présentons dans la Figure 3.2 un deuxième exemple d'un fichier d'entrée pour le script Conlleval.



	Vrai	Prédit
Aji,	B-author	B-author
McEliece,	I-author	I-author
Evaluation,	B-title	B-title
A.,	B-author	I-author
McCallum,	I-author	I-author

Figure 3.2: Deuxième exemple d'un fichier d'entrée pour le script Conlleval.

La seule différence entre les deux fichiers des deux figures 3.2 et 3.1 est l'étiquette prédite pour le jeton *Evaluation.*

Dans cet exemple, Conlleval comptabilise 3 segments étiquetés correctement (2 segments auteur et 1 segment titre). Le premier segment est le segment (B-author, I-author). Le deuxième segment est le segment (B-title) et le troisième segment (I-author, I-author) qui est un vrai positif (segment correctement classé).

Une fois que les segments sont comptabilisés par Conlleval, la prochaine étape est de calculer la précision, le rappel et la F-mesure que nous avons introduit dans le chapitre 2.

3.2 Expériences

Nous tenons toujours à rappeler que nous avons effectué la même expérience réalisée dans (Peng et McCallum, 2006). Autrement dit, nous avons utilisé 350 références de Cora dans les données d'entraînement et 150 références dans les données de test. La sélection des références pour l'ensemble de test et d'apprentissage est faite aléatoirement.

Nous avons utilisé comme machine, tout au long des expériences, un MacBook Pro avec un microprocesseur de 2,7 GHz Intel Core i5, une mémoire vive de 8 Go et un disque dur de 500 Go. Le temps d'exécution pour l'entraînement et le test variait entre 18-25 minutes.

Pour la segmentation et l'étiquetage de séquences, nous avons utilisé un paquetage implémenté en python nommé *sklearn-crfsuite*[2]. Ce paquetage utilise une implémentation des champs aléatoires conditionnels (CRF) à chaîne linéaire nommée *CRFsuite*[3].

Nous avons choisi d'utiliser la régularisation et en particulier la régularisation élastique avec les deux régularisateurs $c1$ et $c2$ qui sont initialisés à 0,1. En appliquant la validation croisée par k-ensembles, nous avons varié le nombre d'échantillons et nous avons remarqué que ce changement n'affectait pas les résultats. C'était seulement le temps d'exécution qui augmentait. Par conséquent, nous avons opté pour l'utilisation de trois échantillons (k=3). Le même principe a été appliqué pour le choix du nombre d'itérations dans l'algorithme L-BFGS. En effet, nous avons choisi d'utiliser un L-BFGS avec un nombre maximal d'itérations qui est

2. https://sklearn-crfsuite.readthedocs.io/en/latest/

3. http://www.chokkan.org/software/crfsuite/

égal à 100.

En appliquant la méthode de recherche aléatoire, nous avons fixé le paramètre (n_iter=50) de telle sorte que le *loss* ne diminue pas de plus que 40000. Ce paramètre négocie le temps d'exécution par rapport à la qualité de la solution.

3.3 Résultats et discussions

Nous présentons dans cette section une comparaison entre les performances de l'état de l'art et nos performances. Les résultats de notre approche sont représentés par la moyenne des résultats par segment après 5 exécutions du modèle CRF sur le dataset Cora. En revanche, les résultats de l'état de l'art sont calculés par jeton. Or, les résultats de l'étiquetage par jeton ne reflètent pas la réalité.

Nous listons dans le Tableau 3.2, les valeurs des deux régularisateurs $c1$ et $c2$ que nous avons utilisées lors des 5 exécutions.

Tableau 3.2: Différentes valeurs des deux régularisateurs $c1$ et $c2$ lors des cinq exécutions de CRF sur CORA.

Expériences	C1	C2
Expérience 1	0.237	0.036
Expérience 2	0.089	0.011
Expérience 3	0.341	0.017
Expérience 4	0.124	0.075
Expérience 5	0.154	0.015

Nous présentons aussi dans le Tableau 3.3 une comparaison entre nos résultats et ceux de l'état de l'art (Peng et McCallum, 2006). Notons que nos résultats sont effectués à travers une tokenisation par espace vu le gain de la performance.

Pour plus de détails, nous listons de même dans **l'annexe A** les résultats par jeton et par segment que nous avons trouvés avec la tokenisation par caractères spéciaux.

Tableau 3.3: Moyenne des résultats par segment après cinq exécutions de CRF sur CORA.

Attributs	F-mesure par jeton de l'état de l'art	F-mesure par jeton de notre approche	F-mesure par segment de notre approche
Author	**99.4%**	99.26% ± 0,313	97.46% ± 0,512
Booktitle	93.7%	**98.74% ± 0,195**	93.98% ± 1,054
Date	98.9%	98.90%	98.08% ± 0,404
Editor	87.7%	**92.80% ± 4,094**	98.52% ± 1,812
Institution	94.0%	**96.90% ± 0,948**	84.52% ± 1,809
Journal	**91.3%**	90.26% ± 1,132	91.00% ± 1,155
Location	87.2%	**92.58% ± 0,664**	86.88% ± 1,191
Note	**80.8%**	80.72% ± 6,299	81.42% ± 5,246
Pages	98.6%	**98.76% ± 0,195**	97.81% ± 0,494
Publisher	76.1%	**89.18% ± 1,126**	87.37% ± 1,306
Tech	86.7%	**93.56% ± 2,233**	86.86% ± 1,809
Title	98.3%	**99.22% ± 0,04**	96.86% ± 0,621
Volume	97.8%	**99.40%**	98.15%
Macro-moyenne	91.5%	**94.63% ± 1,015**	92.22% ± 0,523

Nous pouvons remarquer d'après le Tableau 3.3 que les auteurs dans (Peng et McCallum, 2006) ont trouvé par exemple une F-mesure par jeton qui est égale à 93.7% pour l'attribut *Booktitle* et 87.7% pour l'attribut *Editor*. De plus, ils ont trouvé une macro-moyenne de F-mesure par jeton qui est égale à 91.5%.

Avec notre approche, nous avons réussi à avoir, par exemple, une F-mesure par jeton qui est égale à 98.74% pour l'attribut *Booktitle* et 92.80% pour l'attribut *Editor*. Nous avons réussi aussi à récolter une macro-moyenne de F-mesure par jeton qui est égale à 94.63% (après cinq exécutions). En effet, la meilleure macro-moyenne de F-mesure par jeton que nous avons obtenue est égale à 96,44%. De plus, nous avons obtenue une macro-moyenne de F-mesure par segment qui est égale à 92.22%. Ainsi, notre approche a amélioré les performances de la littérature. Nous présentons aussi dans **l'annexe A** les meilleurs résultats par segment de CRF avec la tokenisation par espace du dataset Cora.

Le gain au niveau de la performance est dû d'une part, à l'utilisation de la notation BIO lors de l'étiquetage des jetons et d'autre part, à une bonne sélection des caractéristiques. Nous avons gardé seulement les caractéristiques qui augmentent la performance du CRF. De même, la régularisation a amélioré la performance du modèle à travers le choix des régularisateurs $c1$ et $c2$ qui ont donné la meilleure performance.

3.4 Conclusion

Nous avons présenté tout au long de ce chapitre notre approche à travers les démarches et les techniques que nous avons utilisées. Nous avons réussi d'après les expériences et les résultats obtenus à améliorer les résultats de l'état de l'art pour le dataset Cora. En outre, nous avons fourni un système basé sur les champs aléatoires conditionnels (CRF) qui est plus performant que celui utilisé dans la littérature (Peng et McCallum, 2006). Dans le prochain chapitre, nous allons tester notre approche sur d'autres datasets. L'objectif est de généraliser l'approche pour fournir un système qui permet d'étiqueter n'importe quelles références bibliographiques, quels que soient le style et la source.

CHAPITRE IV

CONCEPTION D'UN CLASSIFIEUR GÉNÉRALISÉ DE RÉFÉRENCES BIBLIOGRAPHIQUES

Nous présentons dans ce chapitre un système basé sur les champs aléatoires conditionnels (CRF) qui permet d'étiqueter n'importe quelle dataset de références bibliographiques. En effet, l'étiquetage d'un dataset avec un classifieur entraîné sur Cora ne donne pas toujours de bons résultats, surtout avec des données de test qui proviennent d'une autre source, ce qui est attendu puisqu'elles n'ont pas la même distribution (n'ont pas le même style). Nous allons aborder cette question de manière détaillée dans le reste du chapitre.

Nous introduisons au début de ce chapitre les datasets et leurs styles. Ensuite, nous présentons la notion du changement de style et les notations utilisées pour définir les différents aspects liés à notre approche. Nous analysons par la suite les résultats d'étiquetage d'un dataset avec le classifieur Cora. Nous exposons aussi les résultats d'étiquetage d'un dataset avec un classifieur ayant le même style. Finalement, nous concluons ce chapitre par la mise en place d'un système qui permet de détecter automatiquement le style des références bibliographiques d'un dataset afin de choisir le meilleur classifieur à utiliser pour l'étiqueter.

4.1 Notations et préliminaire

Nous présentons dans cette section les datasets, leurs styles et la notion du changement de style. Nous listons aussi les différentes notations que nous allons utiliser tout au long de ce chapitre.

4.1.1 Préparation des datasets

Nous avons choisi de construire deux jeux de données à partir de la conférence *International Conference on Machine Learning* (ICML). La première étape consiste à collecter les références bibliographiques de tous les articles de ICML de l'année 2007 et ceux de l'année 2010. Nous avons obtenu alors deux datasets que nous allons nommer respectivement ICML07 et ICML10. La deuxième étape consiste à étiqueter manuellement ces deux datasets. Pour ce faire, nous avons utilisé un outil d'étiquetage manuel nommé *Visual Tagging Tool*[1]. Cet outil nous permet d'étiqueter manuellement chaque référence bibliographique segment par segment. Une fois que nous avons étiqueté nos données, l'outil génère un fichier de sortie qui contient les segments étiquetés. La troisième étape est la tokenisation. Cette dernière nous permet d'effectuer la décomposition d'une référence bibliographique jeton par jeton. Nous avons choisi d'utiliser la tokenisation par espace et la notation BIO pour étiqueter chaque jeton. Ce choix offre un gain de performance en ce qui concerne les résultats.

1. https://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/vtt/current/web/index.html

4.1.2 Styles des datasets

4.1.2.1 Styles du dataset Cora

Les références bibliographiques du dataset Cora possèdent plusieurs styles. En effet, le Tableau 4.1 présente le nombre de références par styles que nous pouvons trouver dans Cora.

Tableau 4.1: Les styles des références bibliographiques dans le dataset Cora.

Styles des références	Nombres de références
Style APA	160
Style IEEE	140
Autres styles	200

D'après le Tableau 4.1, nous pouvons remarquer que les références bibliographiques du dataset Cora sont une mixture entre plusieurs styles.

4.1.2.2 Style du dataset ICML07

Les références bibliographiques du dataset ICML07 ont un style nommé *American Psychological Association* (APA). En effet, le style APA est l'un des styles les plus utilisés. Il fonctionne avec un système (auteur-date). Autrement dit, le segment *auteur* est toujours suivi par le segment *date* dans une référence bibliographique. Nous présentons dans la Figure 4.1 (section 4.1.3) un exemple d'une référence bibliographique ayant le style APA.

4.1.2.3 Style du dataset ICML10

Contrairement à ICML07, les références bibliographiques du dataset ICML10 ont un style nommé *Institute of Electrical and Electronics Engineers* (IEEE). En effet, IEEE est un style de référence qui est souvent utilisé en informatique. Ce dernier utilise un système dans lequel le segment *date* est le dernier segment dans une référence bibliographique. Nous présentons dans la Figure 4.1 (section 4.1.3) un exemple d'une référence bibliographique ayant le style IEEE.

4.1.3 Changement de style des datasets

Pour changer le style des références bibliographiques, nous avons implémenté un script qui transforme les références d'un dataset vers un style bien déterminé. La première étape consiste à représenter le dataset sous un format sans style. Par exemple, nous avons enlevé tous les séparateurs ajoutés par les différents styles utilisés dans Cora. Nous avons alors un dataset Cora brut sans styles.

La deuxième étape consiste à appliquer le script de formatage que nous avons implémenté sur ce dataset. Ce script permet de générer à partir d'un dataset brut un dataset avec un ou plusieurs styles. Nous avons adapté le même concept sur les datasets ICML07 et ICML10. La Figure 4.1 présente un exemple de changement de style d'une référence bibliographique de la norme APA vers la norme IEEE.

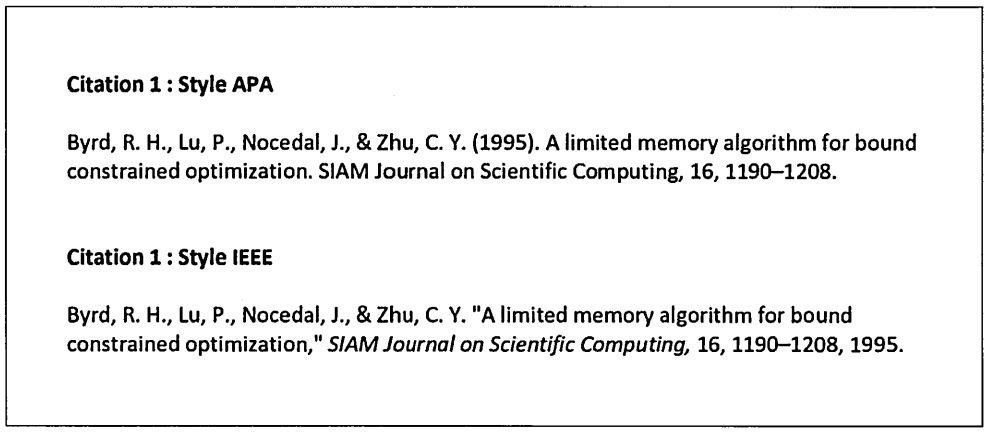


Figure 4.1: Exemple de changement de style d'une référence bibliographique de la norme APA vers la norme IEEE.

4.1.4 Notations

Nous allons utiliser les notations suivantes pour décrire les datasets, les classifieurs et leurs styles tout au long des expériences :

- $D^{style}_{dataset}$: pour faire référence au dataset et son style. Exemples : D^{APA}_{ICML07} et D^{Cora}_{Cora}.
- $C^{style}_{dataset}$: pour faire référence à un classifieur d'un dataset avec son style. Exemples : C^{Cora}_{Cora} et C^{APA}_{Cora}.

4.2 Étiquetage d'un dataset avec le Classifieur C^{Cora}_{Cora}

La première expérience que nous allons présenter contient 350 références du dataset D^{Cora}_{Cora} dans les données d'entraînement pour entraîner le classifieur C^{Cora}_{Cora} et 150 références du dataset D^{APA}_{ICML07} pour les données de test. La Figure 4.2 présente le minimum, le maximum et la moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur C^{Cora}_{Cora} sur le dataset D^{APA}_{ICML07}.

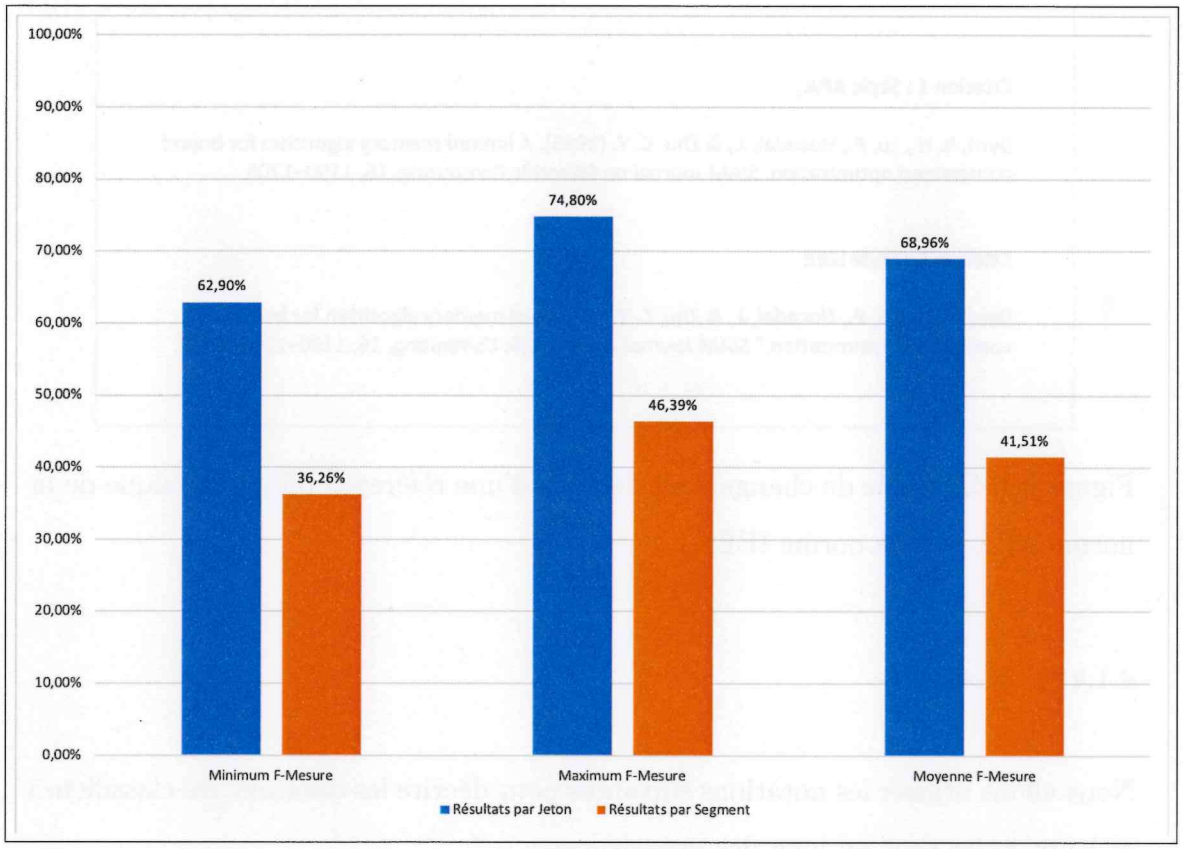


Figure 4.2: Minimum, maximum et moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur C^{Cora}_{Cora} sur le dataset D^{APA}_{ICML07}.

D'après la Figure 4.2, le CRF a donné seulement une moyenne de F-mesure par jeton qui est égale à 68.96%. En revanche, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 41.51%. Ces résultats montrent que les deux datasets D^{APA}_{ICML07} et D^{Cora}_{Cora} ont deux distributions différentes.

La deuxième expérience que nous allons présenter utilise 350 références du dataset D^{Cora}_{Cora} dans les données d'entraînement pour entraîner le classifieur C^{Cora}_{Cora} et 150 références du dataset D^{IEEE}_{ICML10} pour les données de test. La Figure 4.3 présente le minimum, le maximum et la moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur C^{Cora}_{Cora} sur le dataset D^{IEEE}_{ICML10}.

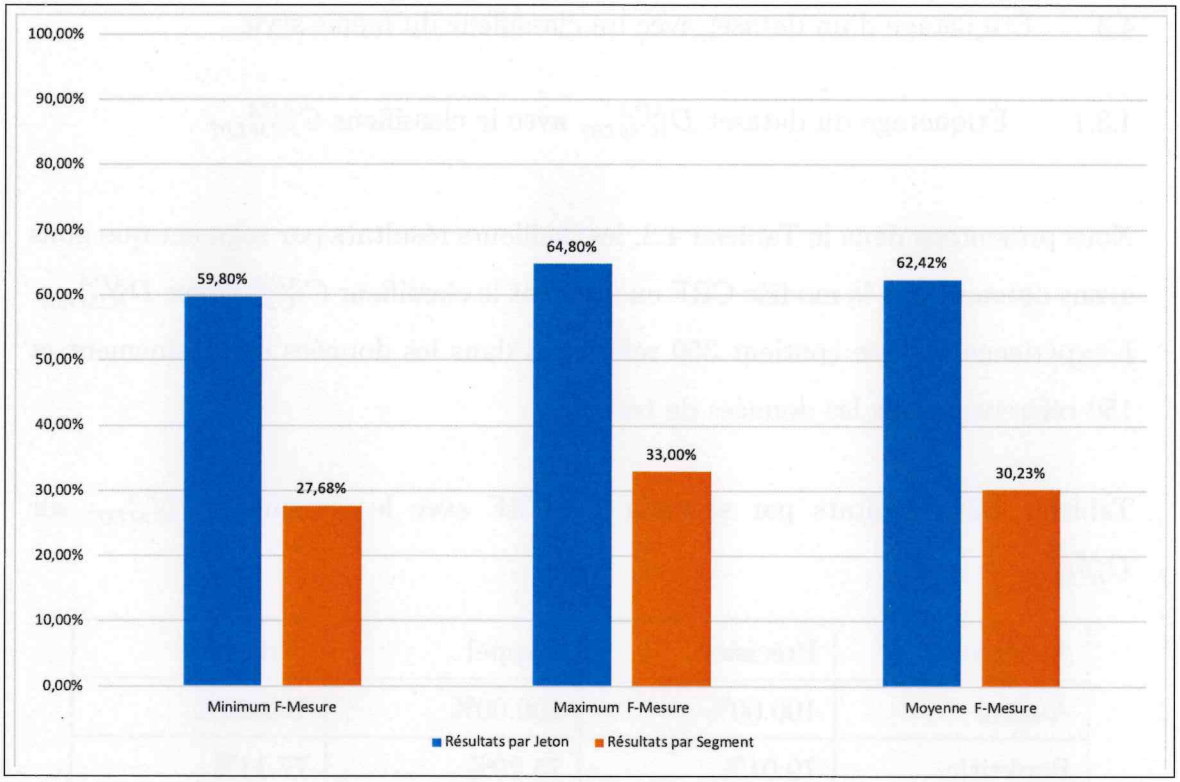


Figure 4.3: Minimum, maximum et moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur C_{Cora}^{Cora} sur le dataset D_{ICML10}^{IEEE}.

D'après la Figure 4.3, le CRF a donné seulement une moyenne de F-mesure par jeton qui est égale à 62.42%. Cependant, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 30.23%. Ces résultats confirment que les deux datasets D_{ICML10}^{IEEE} et D_{Cora}^{Cora} ont deux distributions différentes. Nous présentons aussi dans **l'annexe B** les résultats d'étiquetage des datasets (ICML07,ICML10) avec un classifieur de style différent.

D'après ces expériences, nous pouvons conclure tel que prévu que le classifieur C_{Cora}^{Cora} ne permet pas d'étiqueter des références bibliographiques qui proviennent d'une autre source étant donné qu'elles non pas la même distribution.

4.3 Étiquetage d'un dataset avec un classifieur du même style

4.3.1 Étiquetage du dataset D^{APA}_{ICML07} avec le classifieur C^{APA}_{ICML07}

Nous présentons dans le Tableau 4.2, les meilleurs résultats par segment que nous avons obtenus avec le modèle CRF en utilisant le classifieur C^{APA}_{ICML07} sur D^{APA}_{ICML07}. L'expérience réalisée contient 350 références dans les données d'entraînement et 150 références dans les données de test.

Tableau 4.2: Résultats par segment de CRF avec le classifieur C^{APA}_{ICML07} sur D^{APA}_{ICML07}.

Attributs	Précision	Rappel	F-mesure
Author	100.00%	100.00%	100.00%
Booktitle	79.01%	75.29%	77.11%
Date	100.00%	99.34%	99.67%
Editor	77.78%	100.00%	87.50%
Institution	100.00%	85.71%	92.31%
Journal	80.77%	84.00%	82.35%
Location	97.44%	92.68%	95.00%
Pages	90.80%	97.53%	94.05%
Publisher	100.00%	100.00%	100.00%
Tech	100.00%	100.00%	100.00%
Title	98.00%	98.00%	98.00%
Volume	100.00%	75.47%	86.02%
Macro-moyenne	93.65%	92.34%	92.67%

Notre approche a donné de bons résultats avec ICML07. En effet, nous avons réussi à avoir une macro-moyenne de F-mesure par segment qui est égale à 92,67%. Pour

plus de détails, nous présentons dans **l'annexe B** les résultats par jeton et les résultats de F-mesure (minimum, maximum, moyenne) après cinq exécutions.

4.3.2 Étiquetage du dataset D^{IEEE}_{ICML10} avec le classifieur C^{IEEE}_{ICML10}

Nous présentons dans cette section, les meilleurs résultats par segment que nous avons trouvés avec le modèle CRF en utilisant le classifieur C^{IEEE}_{ICML10} sur D^{IEEE}_{ICML10}. L'expérience réalisée contient 350 références dans les données d'entraînement et 150 références dans les données de test. Le Tableau 4.3 illustre ces résultats.

Tableau 4.3: Résultats par segment de CRF avec le classifieur C^{IEEE}_{ICML10} sur D^{IEEE}_{ICML10}.

Attributs	Précision	Rappel	F-mesure
Author	99.34%	98.69%	99.02%
Booktitle	80.88%	82.09%	81.48%
Date	100.00%	99.35%	99.67%
Editor	100.00%	100.00%	100.00%
Institution	100.00%	60.00%	75.00%
Journal	80.60%	81.82%	81.20%
Location	80.00%	80.00%	80.00%
Note	100.00%	100.00%	100.00%
Pages	100.00%	100.00%	100.00%
Publisher	76.92%	76.92%	76.92%
Tech	100.00%	66.67%	80.00%
Title	96.05%	95.42%	95.74%
Volume	100.00%	100.00%	100.00%
Macro-moyenne	93.37%	87.77%	89.93%

Notre approche a donnée de bons résultats avec ICML10. En effet, nous avons réussi à avoir une macro-moyenne de F-mesure par segment qui est égale à 89.93% et qui est inférieure à celle obtenue avec ICML07. Pour plus de détails, nous présentons dans **l'annexe B** les résultats par jeton et les résultats de F-mesure (minimum, maximum, moyenne) après cinq exécutions.

Nous avons remarqué tout au long de ces expériences que le CRF a donné de bons résultats lorsque le classifieur et les données de test provenaient du même dataset. En outre, le classifieur du même style donne toujours de bons résultats.

4.3.3 Étiquetage d'un dataset avec un classifieur Cora du même style

Nous présentons dans ce qui suit les résultats des expériences avec le changement de styles du classifieur. Dans la première expérience, nous utilisons 350 références du dataset D_{Cora}^{APA} dans les données d'entraînement pour entrainer le classifieur C_{Cora}^{APA} et 150 références du dataset D_{ICML07}^{APA} pour les données de test. La deuxième expérience contient 350 références du dataset D_{Cora}^{IEEE} dans les données d'entraînement pour entraîner le classifieur C_{Cora}^{IEEE} et 150 références du dataset D_{ICML10}^{IEEE} pour les données de test. La Figure 4.4 présente le minimum, maximum et la moyenne de F-mesure par jeton et par segment après cinq exécutions de ces deux expériences.

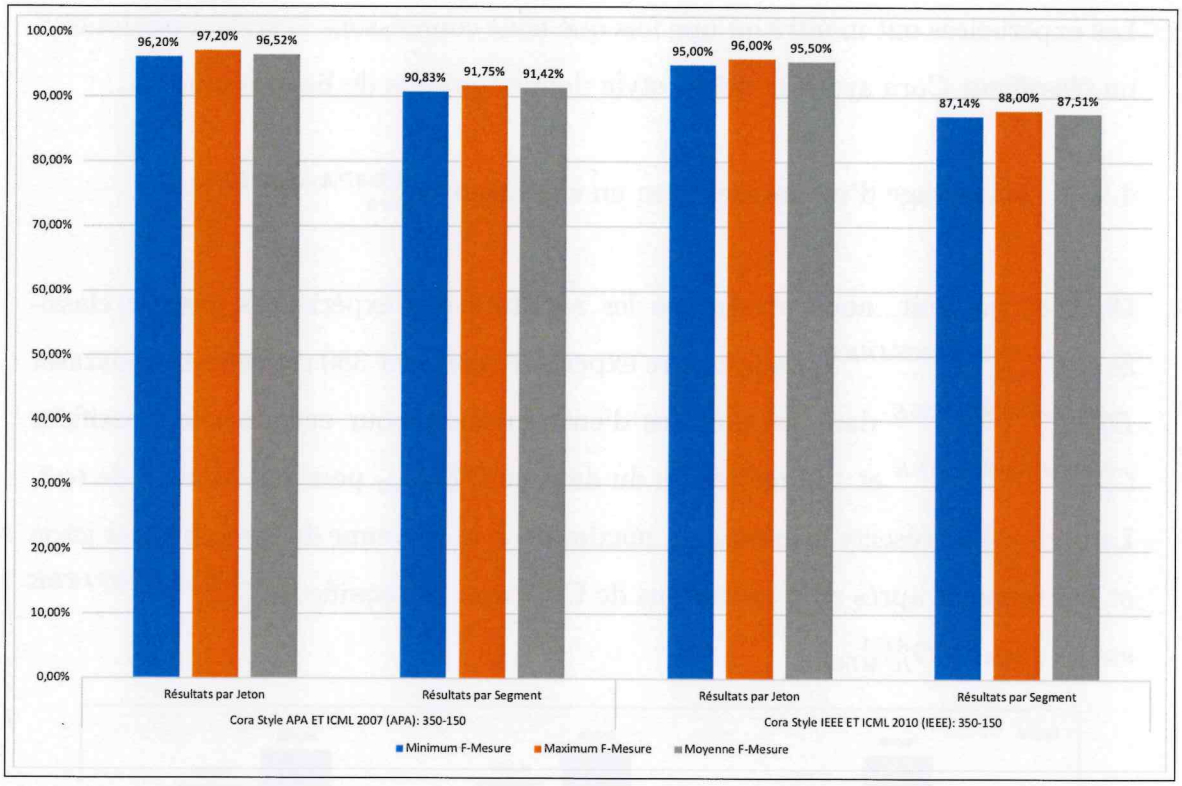


Figure 4.4: Minimum, maximum et moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur C^{APA}_{Cora} sur le dataset D^{APA}_{ICML07} et du classifieur C^{IEEE}_{Cora} sur le dataset D^{IEEE}_{ICML10}.

Après le changement de style du classifieur Cora, le CRF a donné de bons résultats. En effet, lors de l'étiquetage du dataset D^{APA}_{ICML07} avec le classifieur C^{APA}_{Cora}, nous avons réussi à avoir une macro-moyenne de F-mesure par jeton qui est égale à 96.52%. De plus, notre approche a donné une macro-moyenne de F-mesure par segment qui est égale à 91,42%.

Notre approche a donné aussi de bons résultats lors de l'étiquetage du dataset D^{IEEE}_{ICML10} avec le classifieur C^{IEEE}_{Cora}. En effet, nous avons réussi à avoir une moyenne de F-mesure par jeton qui est égale à 95.50%. De plus, le CRF a donné une moyenne de F-mesure par segment qui est égale à 87.51%.

Les expériences ont montré qu'une fois que nous connaissons le style d'un dataset, un classifieur Cora ayant le même style donne toujours de bons résultats.

4.4 Étiquetage d'un dataset avec un classifieur $C_{Cora}^{50\%APA+50\%IEEE}$

Dans ce qui suit, nous présentons les résultats des expériences avec le classifieur $C_{Cora}^{50\%APA+50\%IEEE}$. La première expérience contient 350 références du dataset $D_{Cora}^{50\%APA+50\%IEEE}$ dans les données d'entraînement pour entraîner le classifieur $C_{Cora}^{50\%APA+50\%IEEE}$ et 150 références du dataset D_{ICML07}^{APA} pour les données de test. La Figure 4.5 présente le minimum, maximum et la moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur $C_{Cora}^{50\%APA+50\%IEEE}$ sur le dataset D_{ICML07}^{APA}.

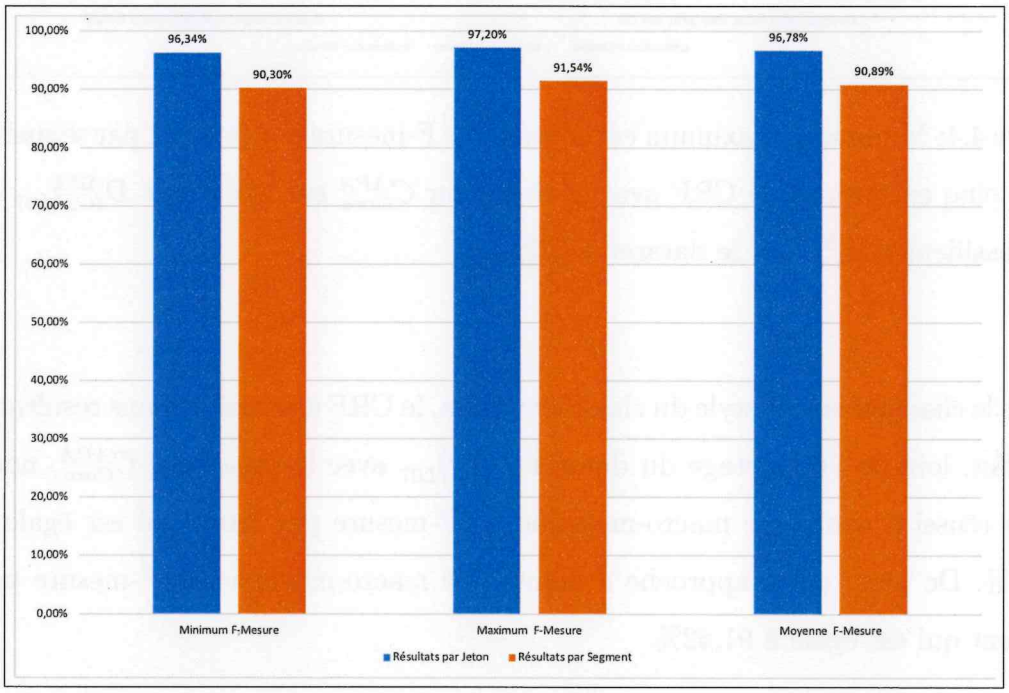


Figure 4.5: Minimum, maximum et moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur $C_{Cora}^{50\%APA+50\%IEEE}$ sur le dataset D_{ICML07}^{APA}.

Après le changement de style du classifieur Cora, le CRF a donné de bons résultats. En effet, nous avons réussi à avoir une moyenne de F-mesure par jeton qui est égale à 96.78%. De plus, le CRF a donné une moyenne de F-mesure par segment qui est égale à 90.89%.

La deuxième expérience contient 350 références du dataset $D_{Cora}^{50\%APA+50\%IEEE}$ dans les données d'entraînement pour entraîner le classifieur $C_{Cora}^{50\%APA+50\%IEEE}$ et 150 références du dataset D_{ICML10}^{IEEE} pour les données de test. Les résultats de F-mesure de cette expérience sont illustrés dans la Figure 4.6.

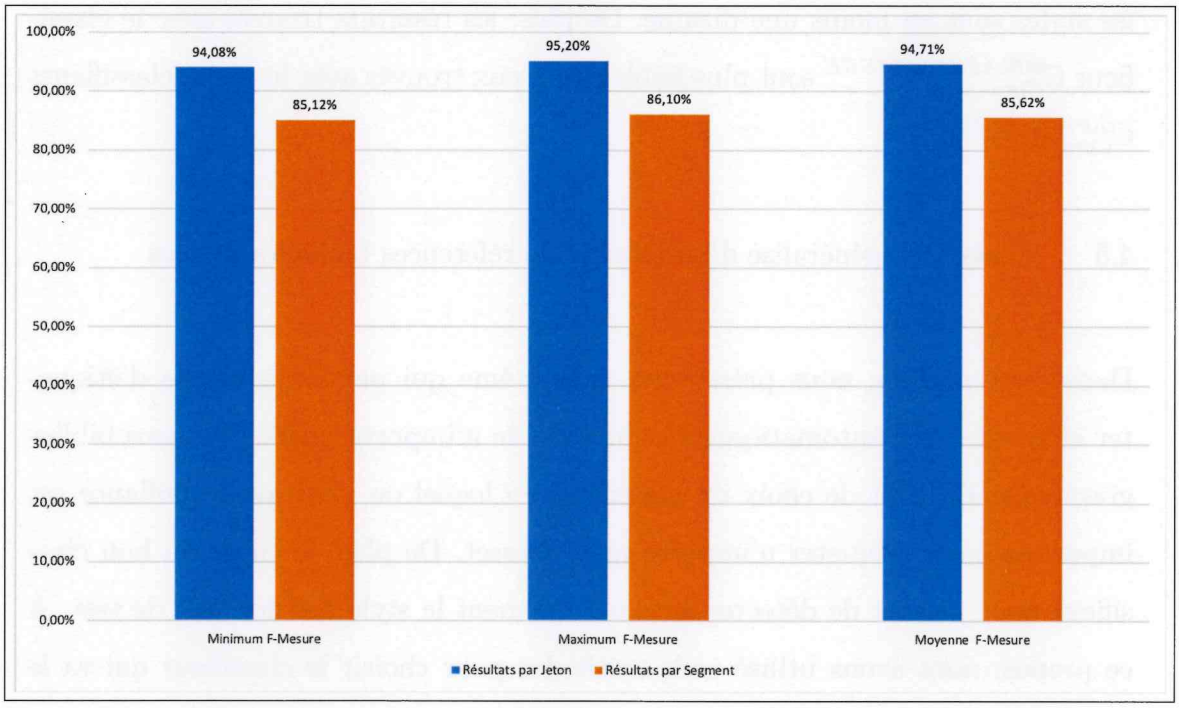


Figure 4.6: Minimum, maximum et moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur $C_{Cora}^{50\%APA+50\%IEEE}$ sur le dataset D_{ICML10}^{IEEE}.

Après le changement de style du classifieur Cora, le CRF a généré de bons résultats. En effet, nous avons réussi à avoir une moyenne de F-mesure par jeton qui est égale à 94.71%. De plus, le CRF a donné une moyenne de F-mesure par

segment qui est égale à 85.62%. Pour plus de détails, nous présentons dans **l'annexe B** une comparaison des résultats avant et après le changement de style des différentes expériences.

Les expériences ont montré qu'un classifieur $C_{Cora}^{50\%APA+50\%IEEE}$ est aussi performant lors de l'étiquetage des datasets ICML07 et ICML10. Ainsi, le classifieur $C_{Cora}^{50\%APA+50\%IEEE}$ peut constituer une solution pour un classifieur généralisé si le dataset comporte seulement deux styles. Cette solution n'est pas généralisable, car nous avons seulement dans le dataset 500 références bibliographiques alors que les styles sont au moins une dizaine. De plus, les résultats trouvés avec le classifieur $C_{Cora}^{50\%APA+50\%IEEE}$ sont plus faibles que ceux trouvés avec les deux classifieurs C_{Cora}^{APA} et C_{Cora}^{IEEE}.

4.5 Classifieur généralisé d'un dataset de références bibliographiques

Dans cette section, nous présentons un système qui permet à la fois d'étiqueter et de détecter automatiquement le style de n'importe quelle référence bibliographique. En effet, le choix du classifieur en lequel on peut avoir confiance est important pour étiqueter n'importe quel dataset. De plus, le choix du bon classifieur nous permet de détecter automatiquement le style des données de test. À ce propos, nous avons utilisé trois méthodes pour choisir le classifieur qui va le mieux.

La première méthode se base sur l'utilisation des heuristiques ou des méthodes de calculs qui offrent rapidement une solution réalisable et aident à la prise de décision. Nous avons utilisé précédemment les deux styles APA et IEEE. La différence entre ces deux styles réside dans la position du segment *date* dans la référence bibliographique. La sélection du classifieur est donc réalisée sur la base d'une bonne classification du champs *date*. Pour appliquer cette méthode, nous avons

implémenté un script qui permet de calculer pour chaque classifieur le nombre de segments où le champs *date* est correctement prédit.

Lors de l'étiquetage du dataset D^{APA}_{ICML07}, le classifieur C^{APA}_{Cora} a annoté correctement 150 segments *date*. En revanche, le classifieur C^{IEEE}_{Cora} a annoté correctement seulement 10 segments *date*. Nous pouvons conclure alors que le classifieur C^{APA}_{Cora} offre une meilleure classification du champs *date*. D'autre part, lors de l'étiquetage du dataset D^{IEEE}_{ICML10}, le classifieur C^{APA}_{Cora} ne trouve aucun segment *date* étiqueté correctement. Toutefois, le classifieur C^{IEEE}_{Cora} étiquette correctement 149 segments *date*. Nous pouvons conclure alors que le classifieur C^{IEEE}_{Cora} offre une meilleure classification du champs *date*.

Les champs aléatoires conditionnels (CRF) permettent de calculer la probabilité à travers une fonction nommée *probability* prédéfinie dans le paquetage *Pycrfsuite*. Ainsi, nous avons utilisé aussi une deuxième méthode qui se base sur la comparaison du logarithme négatif de la vraisemblance. En effet, le classifieur qui donne la plus petite valeur sera choisi. Le logarithme négatif de la vraisemblance d'un dataset 'D' selon un classifieur 'c' est représenté par :

$$-\sum_{x \in D} \log P_c(y_x \mid x) \tag{4.1}$$

Avec :

$$y_x = \arg\max_y P_c(y \mid x)$$

Sachant que : $P_c(y \mid x)$ est la probabilité du CRF selon la formule 1.7.

Nous avons réalisé plusieurs expériences pour chaque classifieur. Nous avons sélectionné aléatoirement dans chaque expérience, des données de test qui sont disjointes et qui proviennent du même dataset. Nous présentons dans le Tableau 4.4 la moyenne et l'écart-type du logarithme négatif de la vraisemblance des datasets selon les différents classifieurs après cinq expériences.

Tableau 4.4: Moyenne et écart-type du logarithme négatif de la vraisemblance des datasets selon les différents classifieurs : Étant donné un dataset, c'est le classifieur de son style qui donne la meilleure performance (le plus petit logarithme négatif de la vraisemblance est le meilleur).

	D^{IEEE}_{ICML10}	D^{APA}_{ICML07}	$D^{Turabian}_{Cora}$	D^{Niso}_{Cora}	D^{ACM}_{Cora}
C^{IEEE}_{Cora}	2699,46 ±52	3532,61 ±54	3749,16 ±145	3252,75 ±75	3589,45 ±141
C^{APA}_{Cora}	3515,24 ±77	2722,23 ±115	3822,72 ±166	3176,16 ±76	3537,73 ±142
$C^{Turabian}_{Cora}$	3518,94 ±68	3621,36 ±42	3429,42 ±144	3405,98 ±69	3638,31 ±67
C^{Niso}_{Cora}	3521,95 ±53	3673,19 ±49	3694,43 ±37	2733,95 ±96	3500,57 ±46
C^{ACM}_{Cora}	3709,50 ±33	3523,38 ±44	3688,86 ±101	3020,95 ±139	2884,18 ±49

D'après le Tableau 4.4, nous pouvons remarquer que le classifieur du même style minimise toujours le logarithme négatif de la vraisemblance du dataset.

Nous avons utilisé aussi une troisième méthode qui nous permet de choisir le bon classifieur. Cette dernière est basée principalement sur le vote en comparant le logarithme négatif de la vraisemblance obtenu de chaque classifieur pour la même séquence. Plus précisément, si le classifieur X possède un logarithme négatif de la vraisemblance qui est strictement inférieur à celui obtenu avec un classifieur Y, alors le classifieur X gagne un point dans le score et vice versa.

Nous avons réalisé plusieurs expériences pour chaque classifieur où nous utilisons différentes données de test qui proviennent du même dataset. Nous présentons ci-dessous la moyenne et l'écart-type du score des datasets selon les différents classifieurs après cinq expériences. Le Tableau 4.5 illustre les résultats trouvés dans ces expériences.

Tableau 4.5: Moyenne et écart-type des scores des expériences réalisées avec les différents classifieurs sur les datasets : Étant donné un dataset, c'est le classifieur de son style qui donne le meilleur score (meilleur vote).

	D^{IEEE}_{ICML10}	D^{APA}_{ICML07}	$D^{Turabian}_{Cora}$	D^{Niso}_{Cora}	D^{ACM}_{Cora}
C^{IEEE}_{Cora}	113,2 ±4, 4	35,6 ±7	54,4 ±11, 3	50,8 ±4, 5	55 ±4, 2
C^{APA}_{Cora}	36,8 ±4, 4	114,4 ±7	57,4 ±6, 5	46,6 ±7, 4	39,4 ±5, 8
$C^{Turabian}_{Cora}$	41,2 ±4, 3	53,4 ±3, 8	98,4 ±8, 4	59,2 ±5, 4	43,8 ±8, 1
C^{Niso}_{Cora}	45,8 ±6, 3	61,4 ±3, 2	42 ±8, 2	102,8 ±7, 5	49,2 ±6
C^{ACM}_{Cora}	51,4 ±5, 3	38,8 ±4, 6	55,2 ±7, 7	52,8 ±3, 4	110,2 ±6, 4

Nous présentons en détail dans **l'annexe B**, les résultats (logarithme négatif de la vraisemblance, score, moyenne, et l'écart-type) de chaque expérience selon les différents classifieurs et datasets.

D'après ces expériences, nous pouvons conclure que le classifieur du même style minimise plus le logarithme négatif de la vraisemblance. En plus, ce classifieur donne toujours un score plus élevé lors du vote. Nous pouvons donc généraliser l'approche.

4.6 Architecture du système

Nous présentons, dans la Figure 4.7, l'architecture de notre système qui permet d'étiqueter n'importe quelle référence bibliographique.

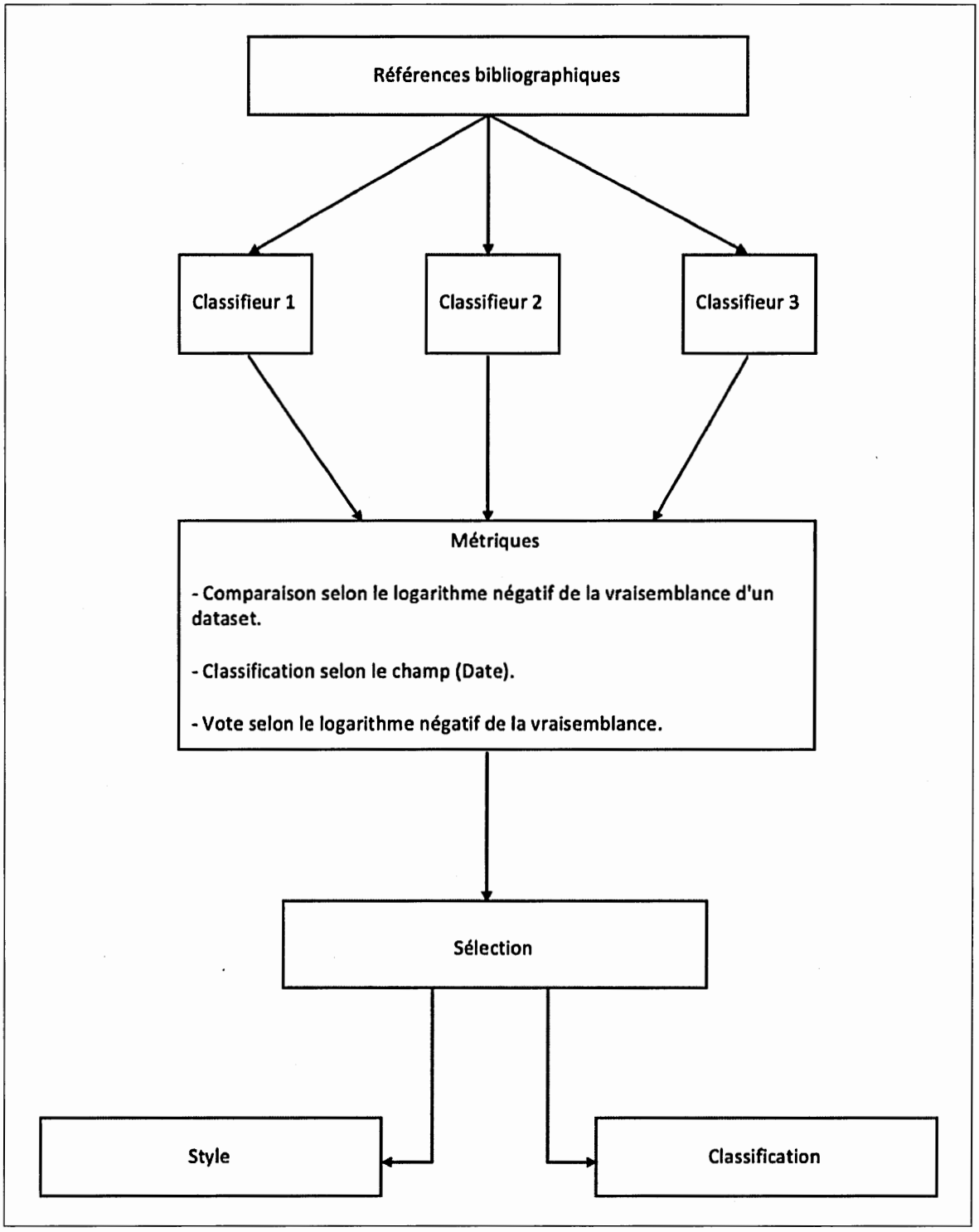


Figure 4.7: Architecture du système d'étiquetage des références bibliographiques.

Tel qu'il est présenté dans la Figure 4.7, le processus de notre système est décrit comme suit. Nous avons au début des références bibliographiques à étiqueter. Ces dernières doivent avoir le même style. Nous supposons que nous disposons pour chaque style connu d'un classifieur entraîné sur D_{Cora}^{style}. La sélection du classifieur est réalisée à travers les métriques que nous avons mentionnées dans le graphe. En effet, l'une des métriques permet de choisir le classifieur de confiance selon la bonne classification du champs *date*. En outre, le classifieur qui étiquette le mieux ce champs est le bon classifieur. Une deuxième métrique qui se base sur la comparaison du logarithme négatif de la vraisemblance est aussi utilisée. Ainsi, le classifieur qui donne la plus petite valeur sera choisi. La dernière métrique se base sur un vote selon le logarithme négatif de la vraisemblance pour la même séquence. En effet, le classifieur qui minimise plus cette valeur gagne un point dans le vote. Le système permet finalement d'étiqueter et de détecter le style des références bibliographiques. Dans les tests que nous avons effectués, ces trois métriques ont toujours concordé avec le bon classifieur.

4.7 Conclusion

Les résultats présentés dans ce chapitre ont montré que l'approche proposée est robuste. En effet, le système que nous avons élaboré permet d'étiqueter n'importe quel dataset de références bibliographiques. L'étiquetage est effectué en choisissant le classifieur avec les meilleures métriques.

CONCLUSION

Dans le cadre de ce mémoire, nous avons exposé la problématique de l'extraction de l'information à partir des références bibliographiques. Nous tenons toujours à rappeler que notre objectif était d'établir et d'améliorer les résultats de l'état de l'art pour le dataset Cora et d'élaborer par la suite une méthode qui étend l'apprentissage aux données provenant de plusieurs sources. Ainsi, nous avons proposé un système basé sur les champs aléatoires conditionnels (CRF), qui permet d'étiqueter une variété de références bibliographiques, quel que soit le style.

Notre approche se déroule sur plusieurs étapes. La première étape est la tokenisation ou la décomposition en jeton de l'ensemble de données. En effet, nous avons choisi d'utiliser la tokenisation par espace étant donné qu'elle a donné de meilleurs résultats par rapport à la tokenisation par caractères spéciaux. L'étape qui suit est l'étiquetage de chaque jeton obtenu à la suite de la tokenisation. Pour ce faire, nous avons choisi d'utiliser la notation BIO que nous avons présentée précédemment. Ce choix est justifié par le gain de performance au niveau des résultats. Nous avons utilisé par la suite la régularisation afin d'éviter le surapprentissage dans le CRF. Nous avons plus précisément opté pour la régularisation élastique. Le choix des régularisateurs $c1$ et $c2$ a été effectué à travers la validation croisée et la méthode de recherche aléatoire. Nous avons utilisé aussi une liste riche de caractéristiques.

Ainsi, les expérimentations que nous avons réalisées ont montré que notre modèle est consistant. En effet, nous avons réussi en premier lieu à établir et améliorer les résultats trouvés dans la littérature. En outre, nous avons réalisé les mêmes

expériences et nous avons obtenu des résultats par jeton et par segment plus performants et plus significatifs. En deuxième lieu, nous avons essayé d'exécuter notre approche sur d'autres ensembles de données que nous avons étiquetés manuellement. Nous avons remarqué que le CRF ne donne pas de bons résultats lorsque les données d'entraînement et les données de test n'ont pas le même style c'est-à-dire qu'ils n'ont pas la même distribution. Par conséquent, nous avons automatisé le choix du classifieur du même style. Ce choix est établi à travers trois méthodes. La première est réalisée sur la base d'une bonne classification du champ *date*. En outre, le classifieur qui étiquette mieux ce champ est le bon classifieur. La deuxième méthode se base sur la comparaison de la somme négative du logarithme de probabilité. Ainsi, le classifieur qui donne la plus petite valeur sera choisi. Nous avons utilisé aussi une troisième méthode basée sur un vote selon le négative du logarithme de probabilité. De ce fait, le classifieur ayant un score plus élevé sera choisi.

Pour conclure, en raison de la qualité des résultats obtenus, nous pensons que notre travail peut être appliqué à d'autres domaines liés à l'extraction d'information et l'étiquetage de séquences. Dans la suite de nos recherches, nous explorerons différentes pistes pour étendre ce travail. Une des possibilités à envisager serait l'application des réseaux de neurones artificiels à notre problématique.

ANNEXE A

RÉSULTATS DÉTAILLÉS DES EXPÉRIENCES LORS DE L'ÉTIQUETAGE DU DATASET CORA AVEC LES CHAMPS ALÉATOIRES CONDITIONNELS

Résultats de CRF avec la tokenisation par caractères spéciaux du dataset Cora

Nous introduisons dans ce qui suit les résultats par jeton (en BIO) et par segment après la tokenisation par caractères spéciaux du dataset Cora.

Résultats par jeton

Nous affichons, dans cette section, les meilleurs résultats par jeton que nous avons trouvés avec le CRF à la suite d'une tokenisation par caractères spéciaux du dataset Cora. Ces résultats sont illustrés dans le Tableau A.1.

Tableau A.1: Résultats par jeton de CRF sur CORA avec la tokenisation par caractères spéciaux.

Attributs	Précision	Rappel	F-mesure
B-author	100%	99.3%	99.6%
I-author	97.7%	99.3%	98.5%
B-Booktitle	97.2%	92.0%	94.5%
I-Booktitle	97.0%	95.5%	96.2%
B-Date	96.1%	97.7%	96.9%
I-Date	98.6%	79.8%	88.2%
B-Editor	86.4%	82.6%	84.4%
I-Editor	86.8%	95.7%	91.0%
B-Institution	88.2%	71.4%	78.9%
I-Institution	87.4%	69.8%	77.6%
B-Journal	88.5%	86.8%	87.6%
I-Journal	74.5%	91.5%	82.1%
B-Location	73.4%	71.6%	72.5%
I-Location	72.2%	72.9%	72.5%
B-Note	100%	66.7%	80.0%
I-Note	85.7%	78.3%	81.8%
B-Pages	89.7%	95.2%	92.4%
I-Pages	92.6%	91.5%	92.0%
B-Publisher	95.6%	82.7%	88.7%
I-Publisher	98.0%	68.1%	80.3%
B-Tech	95.0%	54.3%	69.1%
I-Tech	91.2%	83.9%	87.4%

B-Title	94.3%	93.8%	94.0%
I-Title	97.1%	98.9%	98.0%
B-Volume	95.2%	83.8%	89.1%
I-Volume	62.0%	83.8%	71.3%
Macro-moyenne	94.1%	93.8%	93.8%

Avec la tokenisation par caractères spéciaux, nous avons trouvé comme résultats par jeton une macro-moyenne de F-mesure qui est égale à 93.8%.

Résultats par segment

Nous présentons, dans cette section, les meilleurs résultats par segment que nous avons récoltés après la tokenisation par caractères spéciaux du dataset Cora. Ces résultats sont illustrés dans le Tableau A.2.

Tableau A.2: Résultats par segment de CRF sur CORA avec la tokenisation par caractères spéciaux.

Attributs	Précision	Rappel	F-mesure
Author	97.49%	96.80%	97.14%
Booktitle	91.55%	86.67%	89.04%
Date	95.12%	96.35%	95.73%
Editor	59.09%	56.52%	57.78%
Institution	64.71%	52.38%	57.89%
Journal	86.54%	84.91%	85.71%
Location	65.82%	64.20%	65.00%
Note	83.33%	55.56%	66.67%
Pages	86.28%	92.86%	89.45%
Publisher	88.89%	76.92%	82.47%
Tech	90.48%	54.29%	67.86%
Title	93.75%	93.26%	93.51%
Volume	90.40%	79.58%	84.64%
Macro-moyenne	84.11%	76.18%	79.45%

Avec la tokenisation par caractères spéciaux, nous avons obtenu comme résultats par segment une macro-moyenne de F-mesure qui est égale à 79.45%.

Résultats de F-mesure après cinq exécutions de CRF sur Cora

Dans ce qui suit, nous présentons dans la Figure A.1 les résultats de F-mesure après la tokenisation par caractères spéciaux du dataset Cora. Ces résultats sont réalisés après cinq exécutions de CRF sur Cora.

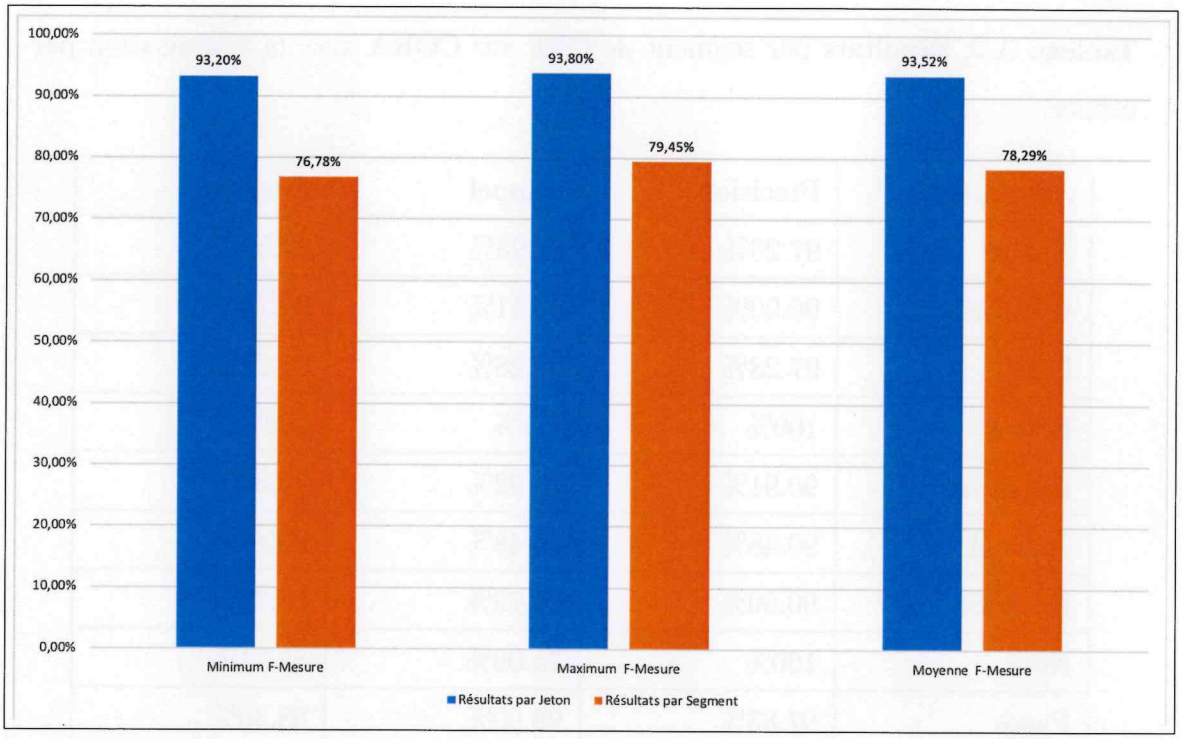


Figure A.1: Minimum, maximum et la moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF sur Cora avec la tokenisation par caractères spéciaux.

D'après la Figure A.1, le CRF a donné une moyenne de F-mesure par jeton qui est égale à 93.52%. En revanche, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 78.29%.

Meilleurs résultats par segment de CRF avec la tokenisation par espace du dataset Cora

Nous présentons ci-dessous les meilleurs résultats par segment que nous avons récoltés avec le modèle CRF sur Cora. Ces résultats sont illustrés dans le Tableau A.3.

Tableau A.3: Résultats par segment de CRF sur CORA avec la tokenisation par espace.

Attributs	Précision	Rappel	F-mesure
Author	97.26%	97.93%	97.59%
Booktitle	96.00%	93.51%	94.74%
Date	97.28%	97.28%	97.28%
Editor	100%	100%	100%
Institution	90.91%	76.92%	83.33%
Journal	90.48%	90.48%	90.48%
Location	90.00%	83.72%	86.75%
Note	100%	75.00%	85.71%
Pages	97.83%	98.90%	98.36%
Publisher	88.89%	86.49%	87.67%
Tech	91.67%	84.62%	88.00%
Title	96.67%	97.32%	96.99%
Volume	98.15%	98.15%	98.15%
Macro-moyenne	95.01%	90.79%	92.70%

Avec la tokenisation par espace, nous avons trouvé comme résultats par segment une macro-moyenne de F-mesure qui est égale à 92.70%.

ANNEXE B

RÉSULTATS DÉTAILLÉS DES EXPÉRIENCES DU CLASSIFIEUR GÉNÉRALISÉ DE RÉFÉRENCES BIBLIOGRAPHIQUES

Résultats par jeton du classifieur C^{APA}_{ICML07} sur le dataset D^{APA}_{ICML07}

Nous présentons dans le Tableau B.1, les résultats par jeton (en BIO) que nous avons obtenus avec le modèle CRF sur le dataset ICML07. L'expérience réalisée contient 350 références du dataset D^{APA}_{ICML07} dans les données d'entraînement pour entraîner le classifieur C^{APA}_{ICML07} et 150 références du dataset D^{APA}_{ICML07} pour les données de test.

Tableau B.1: Résultats par jeton avec BIO de CRF sur ICML07.

Attributs	Précision	Rappel	F-mesure
B-author	100%	100%	100%
I-author	100%	100%	100%
B-Booktitle	87.7%	83.5%	85.5%
I-Booktitle	92.1%	90.2%	91.1%
B-Date	100%	99.3%	99.7%
B-Editor	77.8%	100%	87.5%
I-Editor	79.5%	100%	88.5%
B-Institution	100%	85.7%	92.3%
I-Institution	100%	89.8%	94.6%
B-Journal	82.7%	86.0%	84.3%
I-Journal	79.2%	83.3%	81.2%
B-Location	97.4%	92.7%	95.0%
I-Location	100%	95.5%	97.7%
B-Pages	90.8%	97.5%	94.0%
I-Pages	97.2%	97.2%	97.2%
B-Publisher	100%	100%	100%
I-Publisher	100%	100%	100%
B-Tech	100%	100%	100%
I-Tech	100%	100%	100%
B-Title	100%	100%	100%
I-Title	99.3%	100%	99.7%
B-Volume	100%	75.5%	86.0%
Macro-moyenne	96.3%	96.2%	96.2%

Comme nous pouvons remarquer, nous n'avons pas comptabilisé les étiquettes *I-Volume* et *I-date* étant donné que le jeu de données ICML07 ne contient pas suffisamment d'informations pour ses deux étiquettes. En outre, si nous les comptabilisons, la prédiction sera erronée.

Le Tableau B.1 montre que notre approche a donné de bons résultats avec ICML07. En effet, nous avons réussi à avoir une macro-moyenne de F-mesure par jeton qui est égale à 96.2%.

Résultats de F-mesure après cinq exécutions de CRF avec le classifieur C^{APA}_{ICML07} sur le dataset D^{APA}_{ICML07}

Dans ce qui suit, nous présentons les résultats de F-mesure après cinq exécutions de CRF avec le classifieur C^{APA}_{ICML07} sur D^{APA}_{ICML07}. La Figure B.1 présente le minimum, le maximum et la moyenne de F-mesure par jeton et par segment après cinq exécutions.

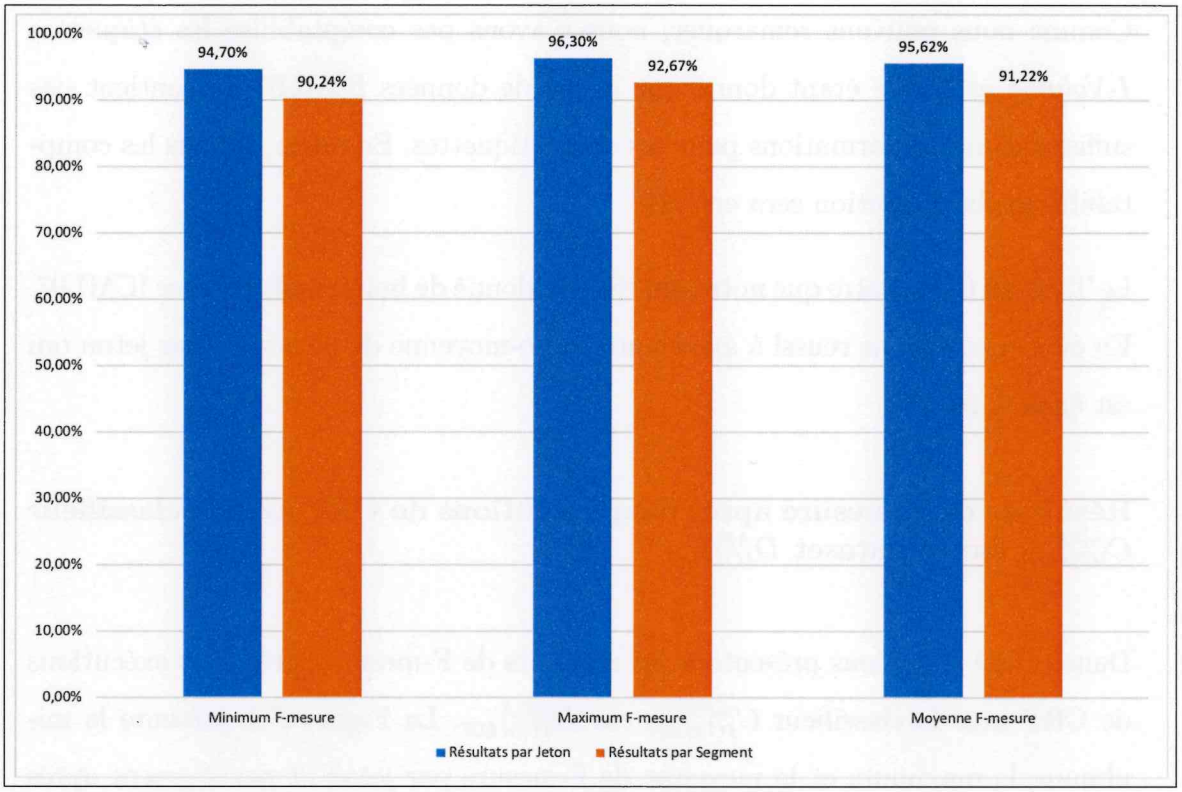


Figure B.1: Minimum, maximum et la moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur C^{APA}_{ICML07} sur D^{APA}_{ICML07}.

D'après la Figure B.1, le CRF a donné une moyenne de F-mesure par jeton qui est égale à 95.62%. Par contre, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 91.22%.

Résultats par jeton du classifieur C^{IEEE}_{ICML10} sur le dataset D^{IEEE}_{ICML10}

Nous présentons dans dans le Tableau B.2 les résultats par jeton (en BIO) que nous avons trouvés avec le modèle CRF sur le dataset ICML10. L'expérience réalisée contient 350 références du dataset D^{IEEE}_{ICML10} dans les données d'entraînement pour entraîner le classifieur C^{IEEE}_{ICML10} et 150 références du dataset D^{IEEE}_{ICML10} pour les données de test.

Tableau B.2: Résultats par jeton avec BIO de CRF sur ICML10.

Attributs	Précision	Rappel	F-mesure
B-author	100%	98.7%	99.3%
I-author	99.8%	99.8%	99.8%
B-Booktitle	83.8%	85.1%	84.4%
I-Booktitle	79.8%	85.9%	82.7%
B-Date	100%	99.3%	99.7%
I-Date	100%	100%	100%
B-Editor	100%	100%	100%
I-Editor	100%	100%	100%
B-Institution	100%	60.0%	75.0%
I-Institution	100%	60.0%	75.0%
B-Journal	80.6%	81.8%	81.2%
I-Journal	81.1%	87.1%	84.0%
B-Location	80.0%	80.0%	80.0%
I-Location	100%	100%	100%
B-Note	100%	100%	100%
I-Note	100%	100%	100%
B-Pages	100%	100%	100%
I-Pages	100%	100%	100%
B-Publisher	76.9%	76.9%	76.9%
I-Publisher	75.0%	64.3%	69.2%
B-Tech	100%	66.7%	80.0%
I-Tech	100%	66.7%	80.0%

B-Title	100%	99.3%	99.7%
I-Title	99.1%	97.3%	98.2%
B-Volume	100%	100%	100%
I-Volume	100%	100%	100%
Macro-moyenne	95.8%	95.6%	95.6%

Notre approche a donné de bons résultats avec le dataset ICML10. En effet, nous avons réussi à avoir une macro-moyenne de F-mesure par jeton qui est égale à 95,6%.

Résultats de F-mesure après cinq exécutions de CRF avec le classifieur C^{IEEE}_{ICML10} sur le dataset D^{IEEE}_{ICML10}

Dans ce qui suit, nous exposons les résultats de F-mesure que nous avons trouvés en exécutant le CRF avec le classifieur C^{IEEE}_{ICML10} sur le dataset D^{IEEE}_{ICML10}. La Figure B.2 présente le minimum, le maximum et la moyenne de F-mesure par jeton et par segment après cinq exécutions.

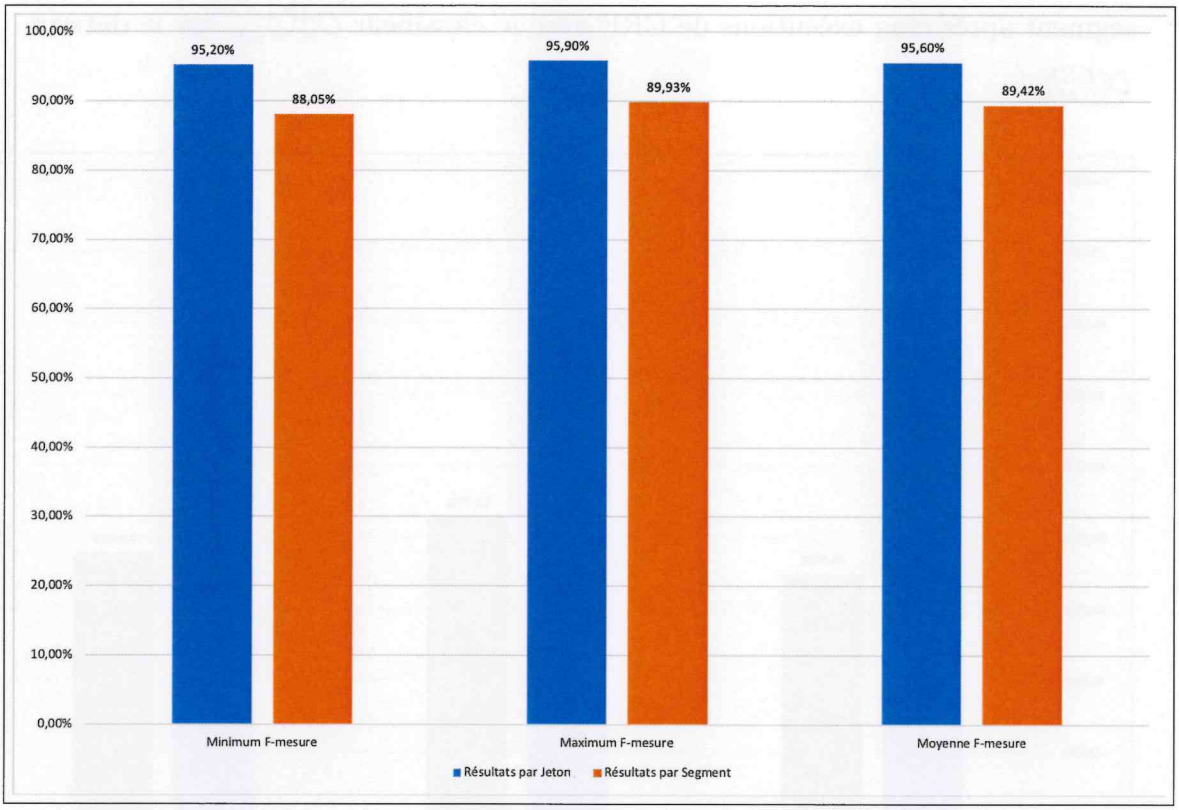


Figure B.2: Minimum, maximum et la moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur C^{IEEE}_{ICML10} sur le dataset D^{IEEE}_{ICML10}.

D'après la Figure B.2, le CRF a donné une moyenne de F-mesure par jeton qui est égale à 95.60%. Cependant, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 89.42%.

Étiquetage d'un dataset avec un classifieur de style différent

La première expérience que nous allons présenter contient 350 références du dataset D^{APA}_{ICML07} dans les données d'entraînement pour entraîner le classifieur C^{APA}_{ICML07} et 150 références du dataset D^{IEEE}_{ICML10} pour les données de test. La Figure B.3 présente le minimum, le maximum et la moyenne de F-mesure par jeton et par

segment après cinq exécutions de CRF avec le classifieur C^{APA}_{ICML07} sur le dataset D^{IEEE}_{ICML10}.

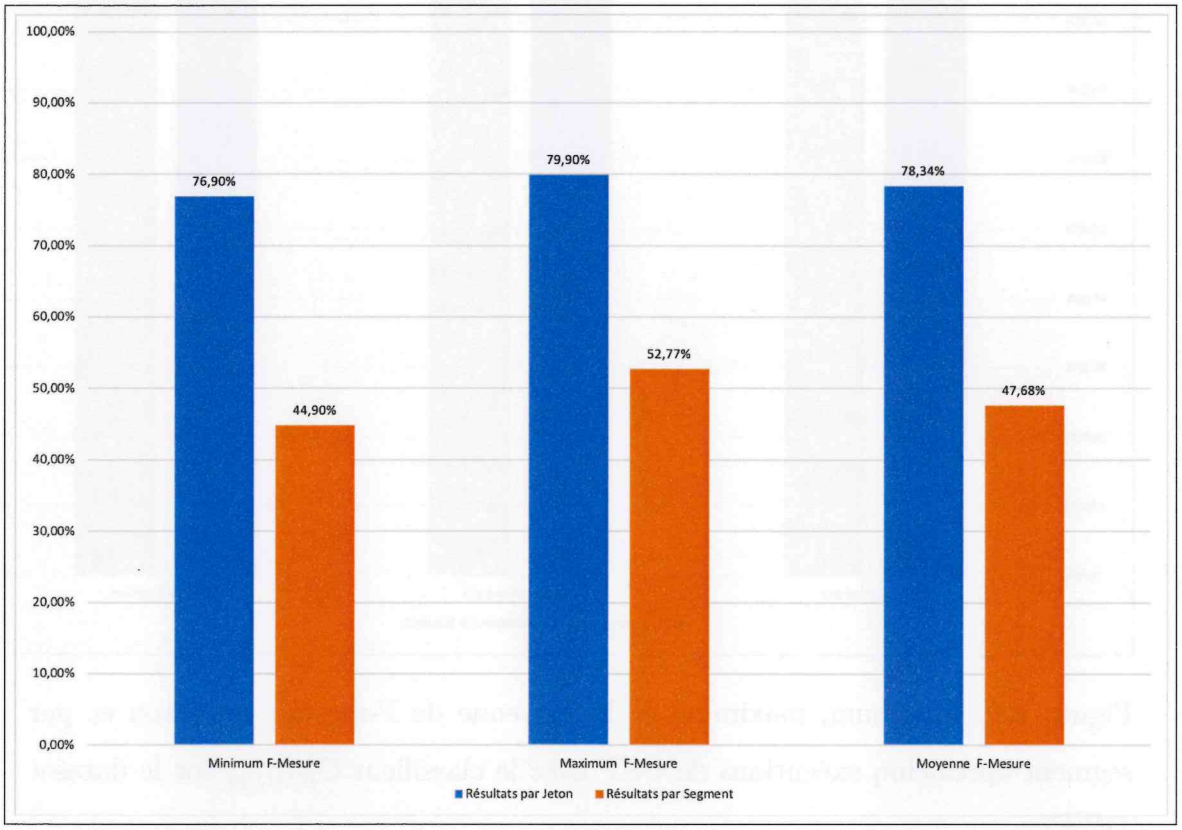


Figure B.3: Minimum, maximum et moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur C^{APA}_{ICML07} sur le dataset D^{IEEE}_{ICML10}.

D'après la Figure B.3, le CRF a donné seulement une moyenne de F-mesure par jeton qui est égale à 78.34%. En revanche, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 47.68%.

La deuxième expérience utilise 350 références du dataset D^{IEEE}_{ICML10} dans les données d'entraînement pour entraîner le classifieur C^{IEEE}_{ICML10} et 150 références du dataset D^{APA}_{ICML07} pour les données de test. La Figure B.4 présente le minimum, le maximum et la moyenne de F-mesure par jeton et par segment après cinq exécutions

de CRF avec le classifieur C^{IEEE}_{ICML10} sur le dataset D^{APA}_{ICML07}.

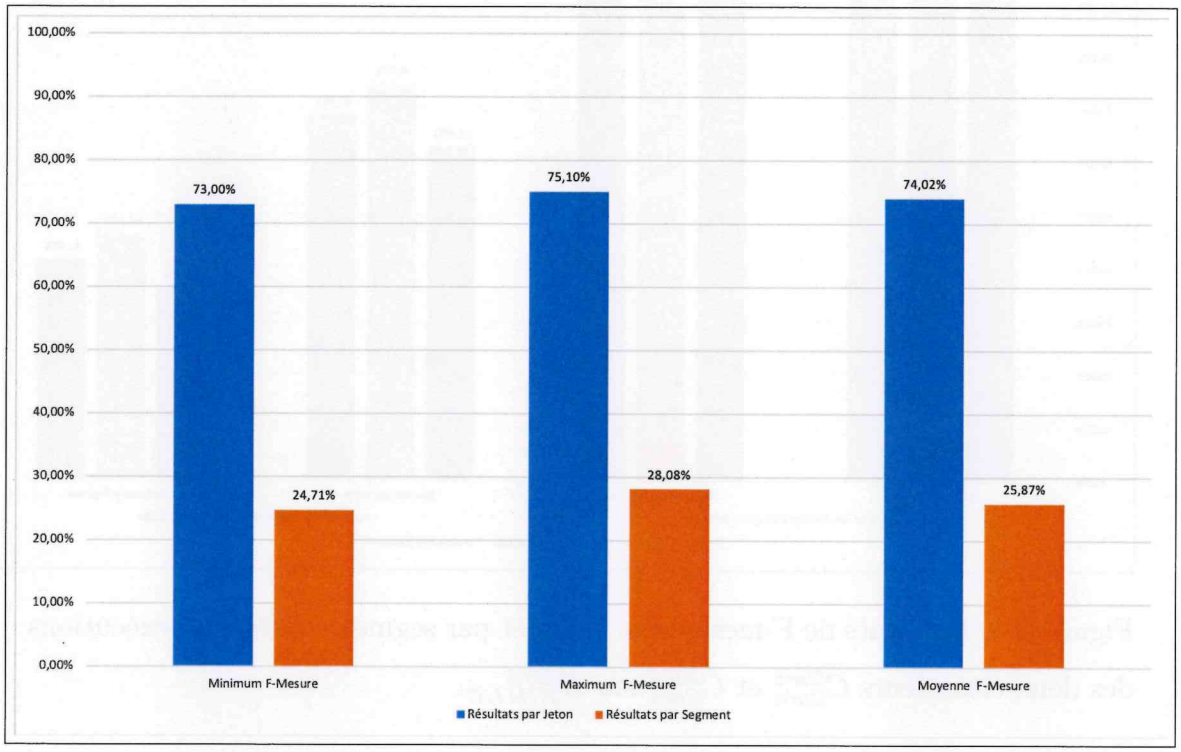


Figure B.4: Minimum, maximum et moyenne de F-mesure par jeton et par segment après cinq exécutions de CRF avec le classifieur C^{IEEE}_{ICML10} sur le dataset D^{APA}_{ICML07}.

D'après la Figure B.4, le CRF a donné seulement une moyenne de F-mesure par jeton qui est égale à 74.02%. Toutefois, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 25.87%.

Comparaison des résultats avant et après le changement de style

Nous allons comparer, dans cette section, les performances obtenues avant et après le changement de style. Premièrement, nous allons comparer les résultats trouvés par les deux classifieurs C^{Cora}_{Cora} et C^{APA}_{Cora} sur D^{APA}_{ICML07}. Les résultats sont mentionnés dans la Figure B.5.

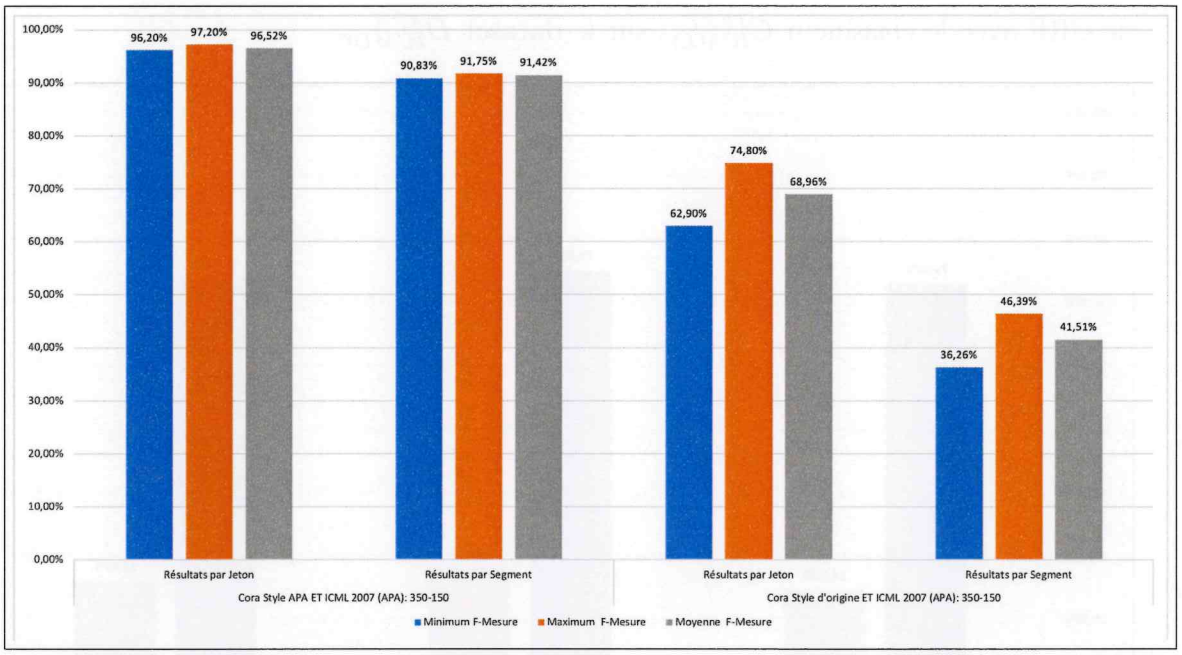


Figure B.5: Résultats de F-mesure par jeton et par segment après cinq exécutions des deux classifieurs C^{Cora}_{Cora} et C^{APA}_{Cora} sur D^{APA}_{ICML07}.

Avant le changement de style, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 41.51%. Par contre, après le changement de style nous avons obtenu une moyenne de F-mesure par segment qui est égale à 91.42%.

Deuxièmement, nous allons comparer les résultats trouvés par les deux classifieurs C^{Cora}_{Cora} et C^{IEEE}_{Cora} sur D^{IEEE}_{ICML10}. Les résultats sont mentionnés dans la Figure B.6.

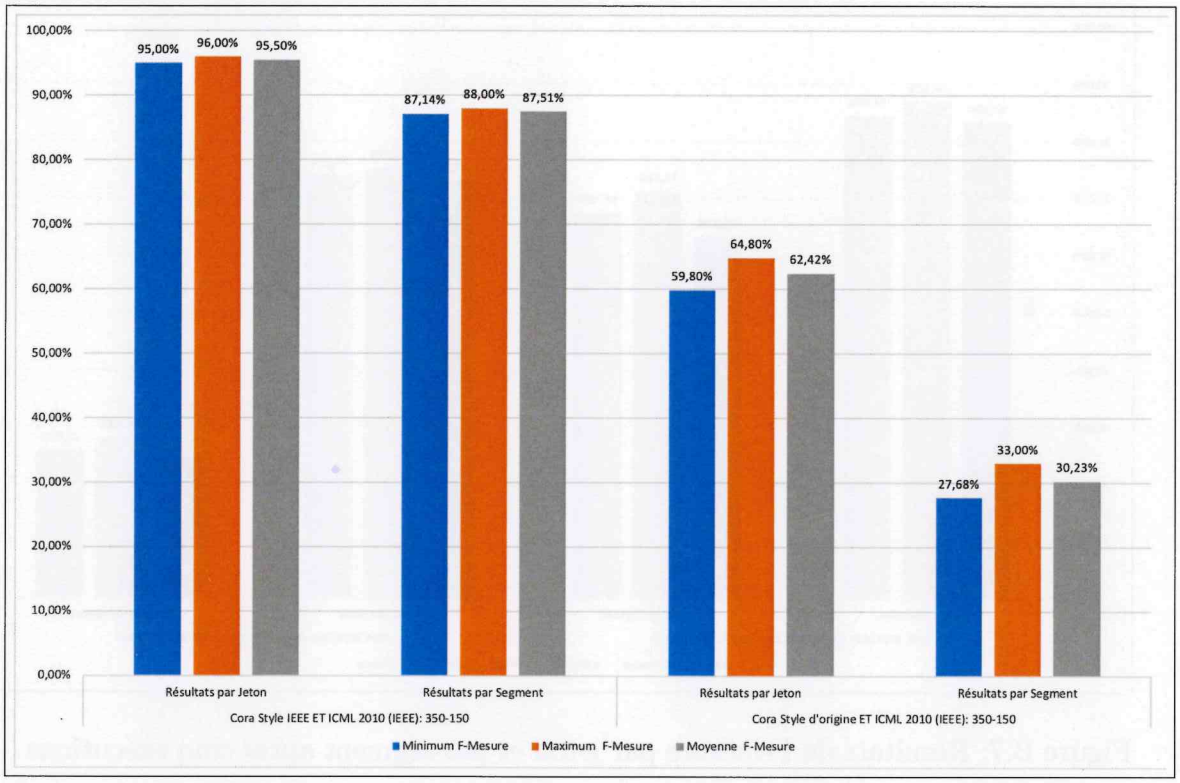


Figure B.6: Résultats de F-mesure par jeton et par segment après cinq exécutions des deux classifieurs C^{Cora}_{Cora} et C^{IEEE}_{Cora} sur D^{IEEE}_{ICML10}.

Avant le changement de style, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 30.23%. Cependant, après le changement de style nous avons obtenu une moyenne de F-mesure par segment qui est égale à 87.51%.

Troisièmement, nous allons comparer les résultats trouvés par les deux classifieurs C^{IEEE}_{ICML10} et C^{APA}_{ICML10} sur D^{APA}_{ICML07}. Les résultats sont mentionnés dans la Figure B.7.

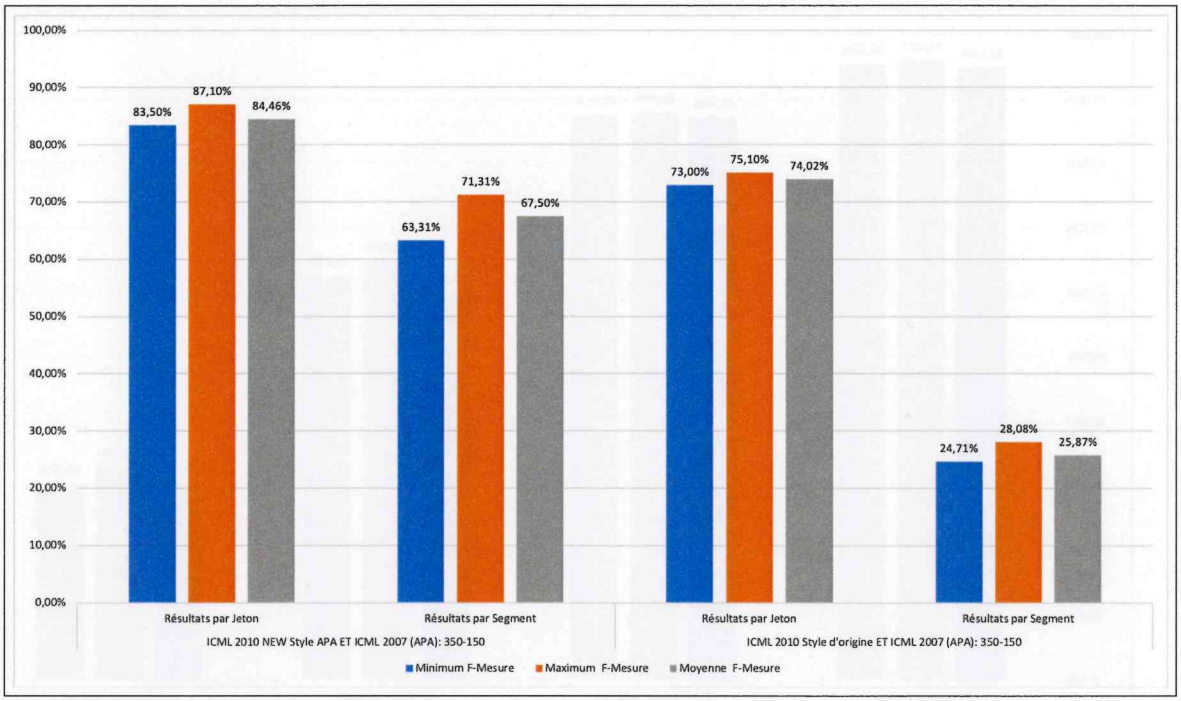


Figure B.7: Résultats de F-mesure par jeton et par segment après cinq exécutions des deux classifieurs C^{IEEE}_{ICML10} et C^{APA}_{ICML10} sur D^{APA}_{ICML07}.

Avant le changement de style, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 25.87%. Toutefois, après le changement de style nous avons obtenu une moyenne de F-mesure par segment qui est égale à 67.50%.

Quatrièmement, nous allons comparer les résultats trouvés par les deux classifieurs C^{APA}_{ICML07} et C^{IEEE}_{ICML07} sur D^{IEEE}_{ICML10}. Les résultats sont mentionnés dans la Figure B.8.

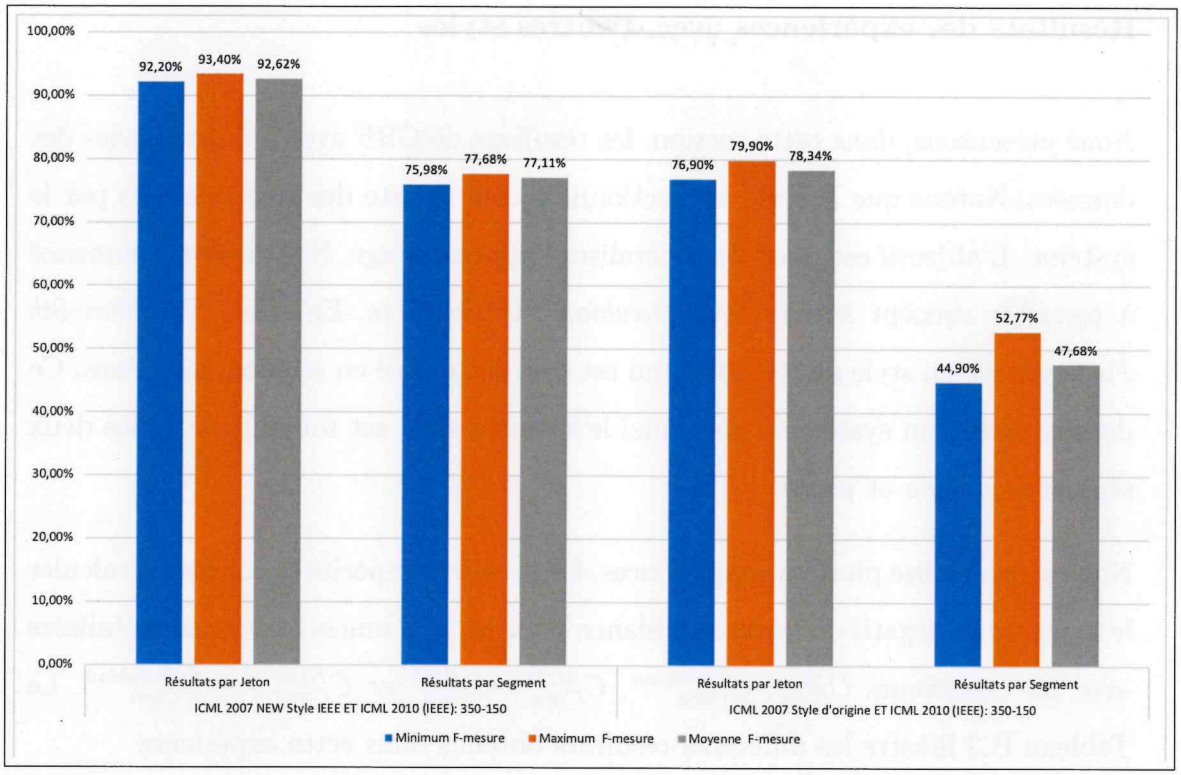


Figure B.8: Résultats de F-mesure par jeton et par segment après cinq exécutions des deux classifieurs C^{APA}_{ICML07} et C^{IEEE}_{ICML07} sur D^{IEEE}_{ICML10}.

Avant le changement de style, nous avons obtenu une moyenne de F-mesure par segment qui est égale à 47.68%. En revanche, après le changement de style, nous avons récolté une moyenne de F-mesure par segment qui est égale à 77.11%.

D'après les expériences réalisées dans cette section, nous pouvons conclure que le changement de style a vraiment amélioré les performances. De même, nous avons obtenu de bons résultats avec d'autres classifieurs à part Cora. Par conséquent, nous avons réussi à généraliser l'entraînement.

Résultats des expériences avec d'autres styles

Nous présentons, dans cette section, les résultats de CRF avec d'autres styles des données. Notons que le style est sectionné parmi la liste des styles connus par le système. L'objectif est donc de généraliser l'apprentissage. Nous avons commencé à tester le concept sur le style *Turabian 8th Footnote*. En effet, *Turabian 8th Footnote* est un style de référence qui est souvent utilisé en sciences humaines. Ce dernier utilise un système dans lequel le segment *date* est toujours entre les deux segments *volume* et *pages*.

Nous avons réalisé plusieurs expériences. La première expérience consiste à calculer le logarithme négatif de la vraisemblance pour les différentes expériences réalisées avec les classifieurs C_{Cora}^{APA}, $C_{Cora}^{Turabian}$, C_{Cora}^{ACM}, C_{Cora}^{IEEE} et C_{Cora}^{Niso} sur $D_{Cora}^{Turabian}$. Le Tableau B.3 illustre les différents résultats obtenus dans cette expérience.

Tableau B.3: Logarithme négatif de la vraisemblance du dataset $D_{Cora}^{Turabian}$ selon les classifieurs C_{Cora}^{APA}, $C_{Cora}^{Turabian}$, C_{Cora}^{ACM}, C_{Cora}^{IEEE} et C_{Cora}^{Niso}.

Expériences	C_{Cora}^{APA}	$C_{Cora}^{Turabian}$	C_{Cora}^{ACM}	C_{Cora}^{IEEE}	C_{Cora}^{Niso}
Expérience 1	3527,81	3263,35	3506,79	3488,03	3715,8
Expérience 2	3920,22	3601,11	3785,78	3791,11	3694,51
Expérience 3	3940,88	3579,54	3695	3888,76	3631,67
Expérience 4	3749,65	3270,98	3784,27	3869,52	3743,86
Expérience 5	3975,08	3432,16	3672,5	3708,39	3686,31
Moyenne	3822,72	3429,42	3688,86	3749,16	3694,43
Écart-type	166,72	144,70	101,93	145,31	37,16

Nous avons calculé aussi le score des expériences réalisées avec les classifieurs C_{Cora}^{APA}, $C_{Cora}^{Turabian}$, C_{Cora}^{ACM}, C_{Cora}^{IEEE} et C_{Cora}^{Niso} sur $D_{Cora}^{Turabian}$. Le Tableau B.4 présente les différents résultats obtenus à la suite de cette expérience.

Tableau B.4: Scores des expériences réalisées avec les classifieurs C_{Cora}^{APA}, $C_{Cora}^{Turabian}$, C_{Cora}^{ACM}, C_{Cora}^{IEEE} et C_{Cora}^{Niso} sur $D_{Cora}^{Turabian}$.

Score des expériences	C_{Cora}^{APA}	$C_{Cora}^{Turabian}$	C_{Cora}^{ACM}	C_{Cora}^{IEEE}	C_{Cora}^{Niso}
Expérience 1	66 points	84 points	69 points	67 points	38 points
Expérience 2	55 points	95 points	57 points	52 points	30 points
Expérience 3	49 points	101 points	53 points	45 points	45 points
Expérience 4	53 points	103 points	46 points	68 points	42 points
Expérience 5	64 points	109 points	51 points	40 points	55 points
Moyenne	57,4	98,4	55,2	54,4	42
Écart-type	6,52	8,47	7,75	11,35	8,22

Nous avons par la suite testé le concept sur le style *Niso*. En effet, *Niso* est un style de référence qui est souvent utilisé en science de l'information. Ce dernier utilise un système dans lequel le segment *date* est toujours suivi respectivement par les deux segments *volume* et *pages*.

L'expérience consiste à calculer le logarithme négatif de la vraisemblance pour les différentes expériences effectuées avec les classifieurs C_{Cora}^{Niso}, C_{Cora}^{APA}, C_{Cora}^{ACM}, C_{Cora}^{IEEE} et $C_{Cora}^{Turabian}$ sur D_{Cora}^{Niso}. Le Tableau B.5 illustre les différents résultats obtenus dans cette expérience.

Tableau B.5: Logarithme négatif de la vraisemblance du dataset D_{Cora}^{Niso} selon les classifieurs C_{Cora}^{Niso}, C_{Cora}^{APA}, C_{Cora}^{ACM}, C_{Cora}^{IEEE} et $C_{Cora}^{Turabian}$.

Expériences	C_{Cora}^{APA}	C_{Cora}^{Niso}	C_{Cora}^{ACM}	C_{Cora}^{IEEE}	$C_{Cora}^{Turabian}$
Expérience 1	3123,65	2795,76	2956,65	3214,78	3446,91
Expérience 2	3076,54	2634,43	2845,13	3342,47	3516,76
Expérience 3	3298,12	2866,20	3193,36	3129,84	3315,42
Expérience 4	3165,82	2614,23	3178,2	3315,19	3389,64
Expérience 5	3216,7	2759,13	2931,44	3261,5	3361,17
Moyenne	3176,16	2733,95	3020,95	3252,75	3405,98
Écart-type	76,55	96,10	139,65	75,59	69,89

Nous avons calculé aussi le score des expériences réalisées avec les classifieurs C_{Cora}^{Niso}, C_{Cora}^{APA} , C_{Cora}^{ACM}, C_{Cora}^{IEEE} et $C_{Cora}^{Turabian}$ sur D_{Cora}^{Niso}. Le Tableau B.6 présente les différents résultats obtenus dans cette expérience.

Tableau B.6: Scores des expériences réalisées avec les classifieurs C_{Cora}^{Niso}, C_{Cora}^{APA}, C_{Cora}^{ACM}, C_{Cora}^{IEEE} et $C_{Cora}^{Turabian}$ sur D_{Cora}^{Niso}.

Score des expériences	C_{Cora}^{APA}	C_{Cora}^{Niso}	C_{Cora}^{ACM}	C_{Cora}^{IEEE}	$C_{Cora}^{Turabian}$
Expérience 1	47 points	103 points	58 points	44 points	60 points
Expérience 2	36 points	114 points	51 points	53 points	56 points
Expérience 3	58 points	92 points	48 points	56 points	67 points
Expérience 4	42 points	107 points	55 points	47 points	51 points
Expérience 5	50 points	98 points	52 points	54 points	62 points
Moyenne	46,6	102,8	52,8	50,8	59,2
Écart-type	7,41	7,52	3,42	4,53	5,41

Nous avons testé pareillement le concept sur un autre style nommé *Association of Computing Machinery* (ACM). En effet, *ACM* est un style de référence qui est souvent utilisé en informatique. Ce dernier ressemble au style APA. En outre, il fonctionne avec un système (auteur-date). La seule différence entre ces deux styles se traduit par les séparateurs.

L'expérience consiste à calculer le logarithme négatif de la vraisemblance pour les différentes expériences réalisées avec les classifieurs C_{Cora}^{ACM}, C_{Cora}^{APA}, C_{Cora}^{Niso}, C_{Cora}^{IEEE} et $C_{Cora}^{Turabian}$ sur D_{Cora}^{ACM}. Le Tableau B.7 illustre les différents résultats générés par cette expérience.

Tableau B.7: Logarithme négatif de la vraisemblance du dataset D_{Cora}^{ACM} selon les classifieurs C_{Cora}^{ACM}, C_{Cora}^{APA}, C_{Cora}^{Niso}, C_{Cora}^{IEEE} et $C_{Cora}^{Turabian}$.

Expériences	C_{Cora}^{APA}	C_{Cora}^{ACM}	C_{Cora}^{Niso}	C_{Cora}^{IEEE}	$C_{Cora}^{Turabian}$
Expérience 1	3376,69	2854,86	3463,68	3385,15	3615,32
Expérience 2	3519,57	2966,91	3475,27	3549,90	3561,17
Expérience 3	3784,29	2876,38	3585,59	3738,62	3753,42
Expérience 4	3426,31	2820,55	3514,43	3513,87	3671,06
Expérience 5	3581,8	2902,23	3463,91	3759,72	3590,62
Moyenne	3537,73	2884,18	3500,57	3589,45	3638,31
Écart-type	142,41	49,27	46,40	141,60	67,93

Nous avons calculé de plus le score des expériences réalisées avec les classifieurs les classifieurs C_{Cora}^{ACM}, C_{Cora}^{APA}, C_{Cora}^{Niso}, C_{Cora}^{IEEE} et $C_{Cora}^{Turabian}$ sur D_{Cora}^{ACM}. Le Tableau B.8 présente les différents résultats obtenus dans cette expérience.

Tableau B.8: Scores des expériences réalisées avec les classifieurs les classifieurs C_{Cora}^{ACM}, C_{Cora}^{APA}, C_{Cora}^{Niso}, C_{Cora}^{IEEE} et $C_{Cora}^{Turabian}$ sur D_{Cora}^{ACM}.

Score des expériences	C_{Cora}^{APA}	C_{Cora}^{ACM}	C_{Cora}^{Niso}	C_{Cora}^{IEEE}	$C_{Cora}^{Turabian}$
Expérience 1	34 points	116 points	53 points	55 points	41 points
Expérience 2	39 points	111 points	47 points	52 points	35 points
Expérience 3	45 points	105 points	43 points	49 points	37 points
Expérience 4	32 points	101 points	59 points	61 points	49 points
Expérience 5	47 points	118 points	44 points	58 points	57 points
Moyenne	39,4	110,2	49,2	55	43,8
Écart-type	5,88	6,43	6,01	4,24	8,15

Nous avons calculé par la suite le logarithme négatif de la vraisemblance du dataset D_{ICML10}^{IEEE} selon les classifieurs C_{Cora}^{APA}, C_{Cora}^{IEEE}, $C_{Cora}^{Turabian}$, C_{Cora}^{ACM} et C_{Cora}^{Niso}. Le Tableau B.9 illustre les différents résultats trouvés dans cette expérience.

Tableau B.9: Logarithme négatif de la vraisemblance du dataset D_{ICML10}^{IEEE} selon les classifieurs C_{Cora}^{APA}, C_{Cora}^{IEEE}, $C_{Cora}^{Turabian}$, C_{Cora}^{ACM} et C_{Cora}^{Niso}.

Expériences	C_{Cora}^{APA}	C_{Cora}^{IEEE}	$C_{Cora}^{Turabian}$	C_{Cora}^{ACM}	C_{Cora}^{Niso}
Expérience 1	3450,77	2599,90	3481,61	3663,18	3514,52
Expérience 2	3558,50	2732,71	3517,57	3751,66	3485,06
Expérience 3	3602,47	2744,21	3625,96	3711,51	3592,73
Expérience 4	3398,86	2695,18	3549,31	3682,76	3447,4
Expérience 5	3565,61	2725,34	3420,26	3738,43	3570,07
Moyenne	3515,24	2699,46	3518,94	3709,50	3521,95
Écart-type	77,09	52,36	68,56	33,10	53,47

Nous avons calculé de plus le score des expériences réalisées avec les classifieurs C_{Cora}^{APA}, C_{Cora}^{IEEE}, $C_{Cora}^{Turabian}$, C_{Cora}^{ACM} et C_{Cora}^{Niso} sur D_{ICML10}^{IEEE}. Le Tableau B.10 présente les différents résultats trouvés dans cette expérience.

Tableau B.10: Scores des expériences réalisées avec les classifieurs C_{Cora}^{APA} et C_{Cora}^{IEEE} sur D_{ICML10}^{IEEE}.

Expériences	C_{Cora}^{APA}	C_{Cora}^{IEEE}	$C_{Cora}^{Turabian}$	C_{Cora}^{ACM}	C_{Cora}^{Niso}
Expérience 1	32 points	118 points	40 points	50 points	41 points
Expérience 2	41 points	109 points	38 points	53 points	55 points
Expérience 3	35 points	115 points	44 points	42 points	50 points
Expérience 4	33 points	117 points	48 points	54 points	37 points
Expérience 5	43 points	107 points	36 points	58 points	46 points
Moyenne	36,8	113,2	41,2	51,4	45,8
Écart-type	4,4	4,4	4,30	5,35	6,36

D'après le Tableau B.10, nous pouvons remarquer que le classifieur C_{Cora}^{IEEE} a donné un meilleur score lors de l'étiquetage du dataset D_{ICML10}^{IEEE}.

Nous avons calculé de même le logarithme négatif de la vraisemblance du dataset D_{ICML07}^{APA} selon les classifieurs C_{Cora}^{IEEE}, C_{Cora}^{APA}, $C_{Cora}^{Turabian}$, C_{Cora}^{ACM} et C_{Cora}^{Niso}. Le Tableau B.11 illustre les différents résultats trouvés dans cette expérience.

Tableau B.11: Logarithme négatif de la vraisemblance du dataset D^{APA}_{ICML07} selon les classifieurs C^{IEEE}_{Cora}, C^{APA}_{Cora}, $C^{Turabian}_{Cora}$, C^{ACM}_{Cora} et C^{Niso}_{Cora}.

Expériences	C^{APA}_{Cora}	C^{IEEE}_{Cora}	$C^{Turabian}_{Cora}$	C^{ACM}_{Cora}	C^{Niso}_{Cora}
Expérience 1	2817,56	3548,32	3651,76	3474,1	3706,02
Expérience 2	2582,06	3470,17	3679,13	3497,43	3686,51
Expérience 3	2735,00	3589,43	3553,91	3518,3	3617,73
Expérience 4	2875,43	3466,38	3601,41	3521,17	3614,29
Expérience 5	2601,12	3588,75	3620,62	3605,92	3741,43
Moyenne	2722,23	3532,61	3621,36	3523,38	3673,19
Écart-type	115,78	54,61	42,91	44,59	49,91

La dernière expérience consiste à calculer le score des expériences réalisées avec les classifieurs C^{IEEE}_{Cora}, C^{APA}_{Cora}, $C^{Turabian}_{Cora}$, C^{ACM}_{Cora} et C^{Niso}_{Cora} sur D^{APA}_{ICML07}. Le Tableau B.12 présente les différents résultats obtenus dans cette expérience.

Tableau B.12: Scores des expériences réalisées avec les classifieurs C_{Cora}^{IEEE}, C_{Cora}^{APA}, $C_{Cora}^{Turabian}$, C_{Cora}^{ACM} et C_{Cora}^{Niso} sur D_{ICML07}^{APA}.

Expériences	C_{Cora}^{APA}	C_{Cora}^{IEEE}	$C_{Cora}^{Turabian}$	C_{Cora}^{ACM}	C_{Cora}^{Niso}
Expérience 1	104 points	46 points	51 points	37 points	66 points
Expérience 2	120 points	30 points	53 points	32 points	61 points
Expérience 3	116 points	34 points	48 points	38 points	59 points
Expérience 4	123 points	27 points	56 points	46 points	64 points
Expérience 5	109 points	41 points	59 points	41 points	57 points
Moyenne	114,4	35,6	53,4	38,8	61,4
Écart-type	7	7	3,82	4,62	3,26

D'après le Tableau B.12, nous pouvons remarquer que le classifieur C_{Cora}^{APA} a donné un meilleur score lors de l'étiquetage du dataset D_{ICML07}^{APA}.

RÉFÉRENCES

Arlot, S. (2018). Validation croisée. In M. Maumy-Bertrand, G. Saporta, et C. Thomas-Agnan (dir.), *Apprentissage statistique et données massives* 141–174. Paris : Éditions Technip. Preliminary version available at http ://hal.archives-ouvertes.fr/hal-01485508

Baum, L. E., Petrie, T., Soules, G. et Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, *41*(1), 164–171.

Bender, O., Och, F. J. et Ney, H. (2003). Maximum entropy models for named entity recognition. Dans *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 148–151. Association for Computational Linguistics.

Bergstra, J. et Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(Feb), 281–305.

Chieu, H. L. et Ng, H. T. (2002). Named entity recognition : a maximum entropy approach using global information. Dans *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 1–7. Association for Computational Linguistics.

Chieu, H. L. et Ng, H. T. (2003). Named entity recognition with a maximum entropy approach. Dans *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 160–163. Association for Computational Linguistics.

Chikhaoui, B. (2013). *Une Approche Basée Sur L'analyse Des Séquences Pour la Reconnaissance Des Activités Et Comportements Dans Les Environnements Intelligents.* (Thèse de doctorat). Université de Sherbrooke.

Constant, M., Tellier, I., Duchier, D., Dupont, Y., Sigogne, A. et Billot, S. (2011). Intégrer des connaissances linguistiques dans un crf : application à l'apprentissage d'un segmenteur-étiqueteur du français. Dans *TALN*, volume 1, p. 321.

Cortes, C. et Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cortez, E., da Silva, A. S., Gonçalves, M. A. et de Moura, E. S. (2010). Ondux : on-demand unsupervised learning for information extraction. Dans *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 807–818. ACM.

Cortez, E., Oliveira, D., da Silva, A. S., de Moura, E. S. et Laender, A. H. (2011). Joint unsupervised structure discovery and information extraction. Dans *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, 541–552. ACM.

Councill, C. L. G. I. et Kan, M.-Y. (2008). Parscit : an open-source crf reference string parsing package. Dans *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, 661–667., Marrakech, Morocco. European Language Resources Association (ELRA). http ://www.lrec-conf.org/proceedings/lrec2008/.

Curran, J. et Clark, S. (2003). Language independent ner using a maximum entropy tagger. Dans *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003.*

Do, T. M. T. et Artières, T. (2006). Champs de markov conditionnels pour le traitement de séquences. Dans G. Ritschard et C. Djeraba (dir.). *EGC*, volume RNTI-E-6 de *Revue des Nouvelles Technologies de l'Information*, 639–650. Cépaduès-Éditions. Récupéré de `http://dblp.uni-trier.de/db/conf/f-egc/egc2006.html#DoA06`

Florian, R., Ittycheriah, A., Jing, H. et Zhang, T. (2003). Named entity recognition through classifier combination. Dans *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 168–171. Association for Computational Linguistics.

Forney, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, *61*(3), 268–278.

Gaizauskas, R. (2002). An information extraction perspective on text mining : Tasks, technologies and prototype applications. Dans *Euromap Text Mining Seminar, Sheffield.*

Han, H., Giles, C. L., Manavoglu, E., Zha, H., Zhang, Z. et Fox, E. A. (2003). Automatic document metadata extraction using support vector machines. Dans *2003 Joint Conference on Digital Libraries, 2003. Proceedings.*, 37–48. IEEE.

Hetzner, E. (2008). A simple method for citation metadata extraction using hidden markov models. Dans *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, 280–284. ACM.

Klein, D., Smarr, J., Nguyen, H. et Manning, C. D. (2003). Named entity recognition with character-level models. Dans *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 180–183. Association for Computational Linguistics.

Lafferty, J., McCallum, A. et Pereira, F. C. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data.

Lavergne, T., Cappé, O. et Yvon, F. (2010). Practical very large scale crfs. Dans *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 504–513. Association for Computational Linguistics.

Lawrence, S., Giles, C. L. et Bollacker, K. (1999). Digital libraries and autonomous citation indexing. *IEEE COMPUTER*, *32*(6), 67–71.

Liu, D. C. et Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, *45*(1-3), 503–528.

Mansuri, I. R. et Sarawagi, S. (2006). Integrating unstructured data into relational databases. Dans *22nd International Conference on Data Engineering (ICDE'06)*, 29–29. IEEE.

Mayfield, J., McNamee, P. et Piatko, C. (2003). Named entity recognition using hundreds of thousands of features. Dans *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 184–187. Association for Computational Linguistics.

McCallum, A. (2002). Efficiently inducing features of conditional random fields. Dans *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, 403–410. Morgan Kaufmann Publishers Inc.

McCallum, A. (2003). Efficiently inducing features of conditional random fields. Dans *Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03)*.

McCallum, A. et Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. Dans *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 188–191. Association for Computational

Linguistics.

McCallum, A. K., Nigam, K., Rennie, J. et Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*, *3*(2), 127–163.

Pazienza, M. T. (2006). *Information Extraction : A Multidisciplinary Approach to an Emerging Information Technology : A Multidisciplinary Approach to an Emerging Information Technology*, volume 1299. Springer.

Peng, F. et McCallum, A. (2004). Accurate information extraction from research papers using conditional random fields. Dans *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics : HLT-NAACL 2004*, 329–336., Boston, Massachusetts, USA. Association for Computational Linguistics. Récupéré de `https://www.aclweb.org/anthology/N04-1042`

Peng, F. et McCallum, A. (2006). Information extraction from research papers using conditional random fields. *Information processing & management*, *42*(4), 963–979.

Petinot, Y., Teregowda, P. B., Han, H., Giles, C. L., Lawrence, S., Rangaswamy, A. et Pal, N. (2003). ebizsearch : An oai-compliant digital library for ebusiness. Dans *2003 Joint Conference on Digital Libraries, 2003. Proceedings.*, 199–209. IEEE.

Ramshaw, L. A. et Marcus, M. P. (1999). Text chunking using transformation-based learning. In *Natural language processing using very large corpora* 157–176. Springer.

Sang, E. F. et De Meulder, F. (2003). Introduction to the conll-2003 shared task : Language-independent named entity recognition. *arXiv preprint cs/0306050.*

Seymore, K., McCallum, A. et Rosenfeld, R. (1999). Learning hidden markov model structure for information extraction. Dans *AAAI-99 workshop on machine learning for information extraction*, 37–42.

Sha, F. et Pereira, F. (2003). Shallow parsing with conditional random fields. Dans *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 134–141. Association for Computational Linguistics.

Suire, Q. (2011). Reconnaissance de mots manuscrits hors-ligne par champs aléatoires conditionnels. Rapport de stage INSA, Université de Rouen.

Récupéré le 2019-09-03 de `http://clement.chatelain.free.fr/docs/rapportStageQuentin2011.pdf`

Sutton, C., McCallum, A. *et al.* (2012). An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, *4*(4), 267–373.

Täckström, O., Das, D., Petrov, S., McDonald, R. et Nivre, J. (2013). Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, *1*, 1–12.

Tartier, A. (2001). Méthodes d'analyse automatique de l'évolution terminologique au travers des variations repérées dans les corpus diachroniques. Dans *Terminologie et intelligence artificielle. Rencontres*, 191–200.

Tellier, I., Eshkol, I., Taalab, S. et Prost, J.-P. (2010). Pos-tagging for oral texts with crf and category decomposition. *Research in Computing Science*, *46*, 79–90.

Tsuruoka, Y., Tsujii, J. et Ananiadou, S. (2009). Fast full parsing by linear-chain conditional random fields. Dans *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 790–798. Association for Computational Linguistics.

Will, C. A. (1993a). Comparing human and machine performance for natural language information extraction : results for english microelectronics from the muc-5 evaluation. Dans *Proceedings of the 5th conference on Message understanding*, 53–67. Association for Computational Linguistics.

Will, C. A. (1993b). Comparing human and machine performance for natural language information extraction : results from the tipster text evaluation. Dans *Proceedings of a workshop on held at Fredericksburg, Virginia : September 19-23, 1993*, 179–193. Association for Computational Linguistics.

Yangarber, R., Grishman, R., Tapanainen, P. et Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. Dans *Proceedings of the 18th conference on Computational linguistics-Volume 2*, 940–946. Association for Computational Linguistics.

Zou, H. et Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society : series B (statistical methodology)*, *67*(2), 301–320.