UNIVERSITE MOHAMMED V - RABAT

ECOLE MOHAMMADIA D'INGENIEURS

THESE

Présentée pour l'obtention du

DOCTORAT EN SCIENCES

Spécialité : Informatique et Traitement Automatique des Langues

Laboratoire Electronique et Communication

Centre d'Etudes Doctorales Sciences et Techniques pour l'Ingénieur

Par

Mohamed OUTAHAJALA

Apprentissage d'un étiqueteur morphosyntaxique de la langue amazighe

Soutenue publiquement, le samedi 06 juin 2015 à 10h, devant le jury composé de:

Pr. Zouhair GUENNOUN	Ecole Mohammedia d'Ingénieurs	Rabat	Président
Pr. Meftaha AMEUR	Institut Royal de la Culture Amazighe	Rabat	Rapporteur
Pr. Noureddine EL FADDOULI	Ecole Mohammedia d'Ingénieurs	Rabat	Rapporteur
Pr. Najib TOUNSI	Ecole Mohammedia d'Ingénieurs	Rabat	Rapporteur
Pr. Violetta CAVALLI-SFORZA	Université Al Akhawayn	Ifrane	Examinatrice
Pr. Lahbib ZENKOUAR	Ecole Mohammedia d'Ingénieurs	Rabat	Directeur de thèse
Pr. Paolo ROSSO	Université Polytechnique de Valence	Espagne	Co-directeur de thèse

Juin 2015

DEDICACE

à mon père et ma mère,

à ma femme et mes enfants,

à mes frères,

à toute ma famille,

à tous mes amis.

REMERCIEMENTS

Je tiens à remercier vivement tout d'abord mes deux directeurs de thèse :

- Lahbib Zenkouar, Professeur de l'Enseignement Supérieur à l'Ecole Mohammedia des Ingénieurs, qui a bien voulu m'accueillir au sein de son équipe et diriger ce travail ; je lui exprime ma sincère gratitude pour sa rigueur scientifique, son suivi permanent et son esprit critique;
- Paolo Rosso, Professeur à l'Université Polytechnique de Valence, pour le suivi qu'il a assuré à ce travail de recherche et pour les conseils judicieux qu'il m'a prodigués tout au long de l'élaboration de cette thèse.

Mes vifs remerciements aux membres du jury qui ont bien voulu étudier, évaluer cette thèse et pour toutes les remarques prodiguées.

Je tiens également à remercier Yassine Benajiba, chercheur à Symanto Research (New York), d'avoir contribué à plusieurs activités liées à ce travail et pour m'avoir fait part de ses idées sur cette recherche.

Je remercie tous mes collègues chercheurs de l'Institut Royal de la Culture Amazighe, plus particulièrement Kamal Ouaqqa, Mustapha Sghir et Lhossain Lgholb d'avoir contribué volontiers à l'annotation morphosyntaxique de la ressource mentionnée dans cette thèse et pour leurs opinions, idées et explications sur toutes les questions se rapportant à la linguistique amazighe. Je remercie également Antònia Martí et Manuel Bertran de l'université de Barcelone pour leur collaboration dans l'amélioration de l'outil AnCoraPipe afin de supporter les propriétés de la langue amazighe.

Je n'oublie pas de remercier M. Ahmed Boukouss, Recteur de l'IRCAM, pour m'avoir autorisé à m'inscrire à l'école doctorale de l'EMI et pour m'avoir permis de participer à plusieurs conférences au Maroc et à l'étranger.

Je remercie également le Conseil pour le Développement de la Recherche en Sciences Sociales en Afrique (CODESRIA) pour m'avoir accordé la bourse de rédaction des mémoires de thèse (SGRT. 17/T12).

Résumé

Comme la plupart des langues qui n'ont que récemment commencé les investigations en Traitement Automatique des Langues (TAL), la langue amazighe est peu dotée en ressources et outils du TAL. Dans ce sens, l'un des objectifs principaux de cette thèse est de doter cette langue de son premier étiqueteur morphosyntaxique.

L'étiquetage morphosyntaxique est la première couche au-dessus du niveau lexical et le niveau le plus bas de l'analyse syntaxique et de toutes les tâches du TAL traitant des niveaux linguistiques supérieurs. Cette tâche produit des informations supplémentaires au texte en entrée ; chose très bénéfique pour les autres tâches du TAL l'utilisant.

Afin d'atteindre cet objectif, nous avons formé deux modèles de classification de séquences, à savoir: les séparateurs à vaste marge (Support Vector Machines, SVMs), et les champs markoviens conditionnels (Conditional Random Fields, CRFs) en utilisant une phase de segmentation. Dans nos expérimentations, nous avons utilisé la technique de 10 fois validation croisée pour évaluer notre approche. Etant donné que nous avons utilisé un corpus d'environ ~ 20k mots, les résultats obtenus sont très prometteurs.

La création des données étiquetées est une tâche difficile, alors que l'obtention des données brutes même si elles nécessitent du temps pour leur prétraitement pour les langues peu dotées, est moins coûteuse. Nous avons eu recours à l'utilisation de ressources externes afin d'améliorer la performance de l'étiqueteur. Ainsi, nous avons construit un corpus d'environ un quart de million de mots, dont nous avons utilisé le caractère informatif des mots hors vocabulaire et la mesure de confiance à même de réduire le taux d'erreur de l'étiqueteur. Pour améliorer la précision de notre étiqueteur morphosyntaxique, nous avons également exploré une ressource lexicale enrichie avec les étiquettes grammaticales.

Mots-clés: Linguistique computationnelle, apprentissage machine, annotation morphosyntaxique, langue amazighe, SVMs, CRFs, TAL.

Abstract

Not unlike most languages that have recently been investigated under a Natural Language Processing (NLP) approach, Amazigh suffers from the scarcity of resources and NLP tools. With the above as background, the main aim of this thesis is to provide this language with its first full-fledged speech (POS) tagger.

POS tagging annotation may well be viewed as the first layer above the lexical level and the lowest level in syntactic analysis along with all the NLP tasks dealing with higher linguistic levels. This task produces additional information for input texts, which is effective for other NLP tasks that make use of it.

In order to develop a POS tagger for the Amazigh language, we trained two sequence labeling models, namely Support Vector Machines (SVMs) and Conditional Random Fields (CRFs), using a tokenizing preprocessing step. In our experiments, we have used the 10 fold cross validation method to evaluate our approach. The obtained results are very promising, even with a small size of labeled data of about 20k words.

While creating labeled data for under resourced languages is a hard task, obtaining raw data, notwithstanding the time they require for their preprocessing, is less costly. We have explored the use of external resources to improve the performance of the tagger. We have, also, built a corpus of about a quarter million words; the informativeness of the non-vocabulary words as well as confidence measure have been used to reduce the error rate of the tagger. To improve the accuracy of our tagger, we have used a lexical resource which includes grammatical labels.

Keywords: Computational linguistics, machine learning, POS tagging, Amazigh language, SVMs, CRFs, NLP.

ملـخّــص

على غرار معظم اللّغات التي لم يتم بعد البدء في معالجتها آليا، تعد اللغة الأمازيغية من اللغات التي تعاني من ندرة الموارد اللغوية وأدوات المعالجة الآلية للغة. وسعيا لتدارك هذا النقص، يسعى هذا البحث إلى المساهمة في توفير بعض الموارد اللغوية الأساسية وتزويدها بنظام دقيق ومتين للوسم الصرفي التركيبي.

الوسم الصرفي التركيبي هو الطبقة الأولى فوق المستوى المعجمي، والمستوى الأخير من التحليل التركيبي وجميع عمليات المعالجة الآلية للغة التي تتناول مستويات لغوية أعلى. هذه العملية تعطي معلومات إضافية للنّص المعالج، ممّا يسمح بأداء أكثر فعّالية لأنظمة المعالجة الآلية للغة التي تستعمله.

من أجل إعداد نظام للوسم الصرف تركيبي للغة الأمازيغية، قمنا بتكوين نموذجين لتسلسل العلامات، وهي آلات المتجهات (SVMs) و الحقول العشوائية المشروطة (CRFs)، باستخدام مستوى تقسيمي كمعالجة مسبقة. في تجاربنا، استخدمنا تقنيات التقييم للتحقق من صحة النتائج. النتائج المتحصل عليها واعدة جدا، وذلك مع صغر حجم البيانات الموسومة المستعملة والتي تتألف من حوالي 20 ألف كلمة .

ونظرا لصعوبة الحصول على البيانات الموسومة، في حين أن البيانات الخامة، والتي غالبا ما تتطلب معالجة مسبقة لجل اللغات قليلة الموارد، هو أقل تكلفة، لهذا فضلنا استخدام الموارد الخارجية لتحسين أداء نظام الوسم الآلي. وبهذا الصدد، قمنا بتطوير خوارزمية، صالحة لجميع اللغات، انطلاقا من متن يضمّ حوالي ربع مليون كلمة، ترتكز على طابع الإفادة ومقياس الثّقة للكلمات خارج المعجم، لتقليل نسبة خطأ النظام الوسمي. كما جمعنا أيضا موردا للمفردات الغنية بعلامات الصرفية التركيبية لتحسين دقة نظام الوسم الآلي.

الكلمات المفتاحية: اللّسانيات الحاسوبيّة، التّعلم الآلي، الوسم الصرفي التركيبي، اللغة الأمازيغية، آلات المتّجهات، الحقول العشوائية المشروطة، المعالجة الآلية للغة .

ⴰⵙⴳⵣⵍ

ⴷⵔⵓⵙⵏ ⵉⵙⵓⴳⴰⵎ ⴷ ⵉⵎⵉⵙⵏ ⵏ ⵓⵎⵙⴽⵍ ⴰⵟⵓⵎⴰⵟⵉⴽ ⵏ ⵜⵓⵜⵍⴰⵢⵉⵏ ⵖⵓⵔ ⵜⵎⴰⵣⵉⵖⵜ ⴰⵎⵎ ⵜⵓⴳⴳⵜ ⵏ ⵜⵓⵜⵍⴰⵢⵉⵏ ⵏⵏⴰ ⵎⵇⵔⴰ ⵉⴱⴷⴰⵏ ⵉⵙⵏⵓⴱⴱⵓⵛⵏ ⴳ ⵓⵎⵙⴽⵍ ⴰⵟⵓⵎⴰⵟⵉⴽ ⵏ ⵜⵓⵜⵍⴰⵢⵉⵏ. ⴰⵢⵏⵏⴰⵖ ⴰⵙ ⵉⴳⴰ ⵢⴰⵏ ⴳ ⵉⵡⵜⵜⴰⵙⵏ ⵏ ⵓⴷⵓⴽⵜⵓⵔⴰ ⴷⴷⵖ ⴰⴷ ⵏⴼⴽ ⵉ ⵜⵎⴰⵣⵉⵖⵜ ⴰⵎⵙⴽⵏⵍ ⴰⵍⵖⵙⵏⴰⵢ ⵏⵏⵙ ⴰⵎⵣⵡⴰⵔⵓ.

ⵉⴳⴰ ⵓⵙⴽⵏⵍ ⴰⵍⵖⵙⵏⴰⵢ ⵢⴰⵜ ⵜⴰⵙⵃⵃⵉⴼⵜ ⵜⴰⵎⵣⵡⴰⵔⵓⵜ ⵏⵏⵉⴳ ⵏ ⵓⵙⵡⵉⵔ ⵏ ⵓⵎⴰⵡⴰⵍ, ⵉⴳ ⴰⵙⵡⵉⵔ ⴰⵎⴰⵣⴷⴰⵔ ⵏ ⵓⵙⴼⵉ ⴰⵎⵙⵏⵉⵢ ⴷ ⴰⴽⴽⵯ ⵜⵉⵎⵙⴽⵉⵔⵉⵏ ⵏ ⵓⵎⵙⴽⵍ ⴰⵟⵓⵎⴰⵟⵉⴽ ⵏ ⵜⵓⵜⵍⴰⵢⵉⵏ ⵏⵏⴰ ⵜⵜ ⵢⵉⵡⵉⵏ ⴳ ⵓⵎⵙⴽⵍ ⵏ ⵉⵙⵡⵉⵔⵏ ⵉⵙⵏⵉⵍⵙⵉⵜⵏ ⵉⵎⴰⴼⵍⵍⴰⵡⵏ. ⵜⴰⵎⵙⴽⵉⵔⵜ ⴷⴷⵖ ⴷⴰ ⵜⴰⴽⴽⴰ ⵉⵏⵖⵎⵉⵙⵏ ⵢⴰⴹⵏⵉⵏ ⵉ ⵉⴹⵕⵉⵚⵏ ⵙ ⵉⵙⴰⴷⴼⵏ ; ⴰⵢⵏⵏⴰ ⵉⵀⵢⵢⴰⵏ ⵉ ⵜⵉⵎⵙⴽⵉⵔⵉⵏ ⵢⴰⴹⵏⵉⵏ ⵏ ⵓⵎⵙⴽⵍ ⴰⵟⵓⵎⴰⵟⵉⴽ ⵏ ⵜⵓⵜⵍⴰⵢⵉⵏ.

ⴰⴼⴰⴷ ⴰⴷ ⵏⵏ ⵏⴰⵡⴹ ⴰⵡⵜⵜⴰⵙ ⴷⴷⵖ, ⵏⴳⴰ ⵙⵉⵏ ⵡⴰⵏⴰⵡⵏ ⵏ ⵓⵏⵎⴰⵍⴰ ⵏ ⵜⴳⵣⵣⵓⵎⵉⵏ ⵏⵏⴰ ⵜⵏ ⵉⴳⴰⵏ : ⵉⴱⴹⵉⵜⵏ ⵉⴷ ⴱⵓ ⵜⴰⵙⴳⴰ ⵜⴰⴱⴰⵔⴰⵡⵜ (SVMs) ⴷ ⵉⴳⵔⴰⵏ ⵉⵎⴰⵔⴽⵉⴱⵉⵜⵏ ⵉⴼⴰⴷⴰⵏⵏ (CRFs) ⵙ ⵓⵙⵎⵔⵙ ⵏ ⵓⵣⵎⵣ ⵏ ⵓⴳⵣⴰⵎ. ⴳ ⵜⵙⵉⵔⵉⵎⵜ ⵏⵏⵖ, ⵏⵙⵎⵔⵙ ⵜⴰⵜⵉⵃⵏⵉⵢⵜ ⵏ 10 ⵏ ⵜⵉⴽⴽⴰⵍ ⵏ ⵓⵙⵖⵣⵏ ⵉⵎⵢⴰⴽⵛⴰⵎⵏ ⴰⴼⴰⴷ ⴰⴷ ⵏⵙⵜⵍ ⵜⴰⵎⵢⴰⴷⴰⵙⵜ ⵏⵏⵖ. ⵜⵉⵢⴰⴼⵓⵜⵉⵏ ⵏⵏⴰ ⴷ ⵉⴼⴼⵖⵏ ⴳⴰⵏⵜ ⵜⵉⵎⴰⴼⴰⵢⵉⵏ ⵡⴰⴳⴳⴰ ⵓⵔ ⵏⵙⵔⵎⵉⵙ ⵖⴰⵙ ⵢⴰⵏ ⵓⴳⵔⵉⵡ ⵏ 20 ⴰⵍⴼ ⵏ ⵜⴳⵓⵔⵉⵡⵉⵏ.

ⵉⵎⴽⵉⵏⵏⴰ ⵏⵙⵙⵏ ⵉⵙ ⵜⵛⵃⵃⴰ ⵜⵓⴳⵉ ⵏ ⵜⵎⵓⵛⴰ ⵉⵜⵜⵓⵙⴽⵏⵍⵏ, ⴰⵔ ⵏⵏ ⵏⵜⵜⴰⴼⴰ ⵓⵎⵓⵣ ⵏ ⵜⵎⵓⵛⴰ ⵉⵍⵍⴰⵏ ⴳ ⵡⴰⴷⴷⴰⴷ ⴰⵎⵏⵣⵓ, ⵎⵃⵃⴰⵔ ⴷⴰ ⵉⵜⵜⵓⵜⵜⵔ ⵛⵉⴳⴰⵏ ⵏ ⵡⴰⴽⵓⴷ, ⵉⵙ ⵢⵓⵀⵏ ⴳ ⵓⵙⵎⴽⵍ ⵏⵏⵙⵏ ⵖⵓⵔ ⵜⵓⵜⵍⴰⵢⵉⵏ ⵎⵉ ⵖⵓⵔ ⵓⵔ ⵍⵍⵉⵏ ⵉⵎⵉⵙⵏ. ⵅⴼ ⵓⵢⴷⴷⵖ ⴰⵙ ⵏⴰⵏⵓⴳ ⴰⵙⵎⵔⵙ ⵏ ⵉⵙⵓⴳⴰⵎ ⵉⴱⵔⵔⴰⵏⵉⵢⵏ ⴰⴼⴰⴷ ⴰⴷ ⵏⵊⵊⵀⴷ ⵜⵉⵢⴰⴼⵓⵜⵉⵏ ⵏ ⵓⵙⴽⵏⵍ.

ⵏⵚⴽⴰ ⵢⴰⵏ ⵓⴳⵔⵉⵡ ⵏ ⵢⴰⵏ 220 ⴰⵍⴼ ⵏ ⵜⴳⵓⵔⵉ, ⵏⵏⴰ ⴳ ⵏⵙⵎⵔⵙ ⴰⵡⵏⵖⵓ ⴰⵏⵖⵎⴰⵙ ⵏ ⵜⴳⵓⵔⵉⵡⵉⵏ ⵓⵔ ⵉⵍⵍⵉⵏ ⴳ ⵓⵎⴰⵡⴰⵍ ⵉⵥⵍⵉⵏ ⴷ ⵓⵙⵃⵃⵓⵍ ⵏ ⵜⵖⵍⵉⴼⵜ ⴰⴼⴰⴷ ⴰⴷ ⵏⴽⵔⵓ ⴳ ⵓⵙⵖⵍ ⵏ ⵜⵣⴳⵍⵜ ⵏ ⵓⵙⴽⵏⵍ.

ⵏⵙⵙⵎⵔⵙ ⴰⵡⴷ ⴰⵙⴰⴳⵎ ⵏ ⵓⵎⴰⵡⴰⵍ ⵉⴱⴱⵓⵖⵍⴰⵏ ⵙ ⵜⵛⵏⵢⴰⵍⵜ ⵜⴰⵏⵊⵕⵓⵎⵜ ⴰⴼⴰⴷ ⴰⴷ ⵏⵙⵙⵇⵍⵓ ⵜⵉⵙⴷⴷⴰ ⵏ ⵓⵙⴽⵏⵍ ⴰⵍⵖⴰⵙⵏⴰⵢ.

ⵜⵉⴳⵓⵔⵉⵡⵉⵏ-ⵜⵉⵙⵓⵔⴰ: ⵜⴰⵊⵏⵢⵓⵔⵜ ⵏ ⵓⴳⵔⵉⵡ, ⴰⵍⵎⵎⵓⴷ ⵏ ⵓⵎⴰⴽⵉⵏⴰ, ⴰⵙⵏⵏⵃⴹ ⴰⵍⵖⴰⵙⵏⴰⵢ, ⵜⵓⵜⵍⴰⵢⵜ ⵜⴰⵎⴰⵣⵉⴳⵀⵜ, SVMs, CRFs, ⴰⵎⵙⴽⵍ ⴰⵟⵓⵎⴰⵟⵉⴽ ⵏ ⵜⵓⵜⵍⴰⵢⵉⵏ.

Table des matières

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABREVIATIONS

AMTS	AMazighe Tag Set
API	L'Alphabet Phonétique International
CAC	Champ Aléatoire Conditionnel
CEI	Commission Electrotechnique Internationale
CoNLL	Conference on Computational Natural Language Learning
CRFs	Conditional Random Fields
ELDA	Evaluations and Language resources Distribution Agency
HMM	Hidden Markov Model
IA	Intelligence Artificielle
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
IRCAM	Institut Royal de la Culture AMazighe
ISO	International Organization for Standardization
LC	Linguistique Computationnelle
LCTL	Less Commonly Taught Languages
LDC	Linguistic Data Consortium
MEMM	Maximum Entropy Markov Model
MUC	Message Understanding Conferences
NLP	Natural Language Processing
OOV	Out Of Vocabulary
PMB	Plan Multilinge de Base
POS	Part Of Speech
RFC	Request For Comment
SNIMA	Service de la Normalisation Industrielle Marocaine
SVMs	Support Vector Machines
TAL	Traitement Automatique des Langues

INTRODUCTION GENERALE

La langue amazighe est essentiellement parlée en Afrique du nord. Elle est également parlée par d'autres communautés dans certaines régions du Niger et du Mali, ainsi que des milliers d'immigrants amazighes partout dans le monde. La langue amazighe regroupe plusieurs dialectes dont aucun n'est considéré comme étant la norme nationale dans aucun des pays comportant des populations amazighophones.

Avec l'émergence de la revendication identitaire, les locuteurs natifs militent pour la sauvegarde et la promotion de leur langue et culture. Pour atteindre cet objectif, certains Etats du Maghreb ont créé des institutions spécialisées, telles que l'Institut Royal de la Culture Amazighe (IRCAM) au Maroc et le Haut-Commissariat de l'Amazighité (HCA) en Algérie. Depuis quelques années, plusieurs publications dans divers domaines ont connu le jour. Néanmoins, cette langue a encore beaucoup de défis à relever (Boukouss, 2012).

Par ailleurs, et comme la majorité des langues dont les recherches en TAL ont récemment commencé, la langue amazighe est peu dotée en ressources langagières et outils du TAL. L'objectif principal de ce travail est de contribuer à la construction de ressources langagières et outils TAL pour l'amazighe, et au développement du TAL en général. Pour ce faire, nous avons œuvré à l'élaboration d'un étiqueteur morphosyntaxique de cette langue, et à la contribution à l'amélioration des méthodes en relation avec cette tâche.

L'analyse morphosyntaxique consiste en l'étiquetage de chaque mot d'une phrase avec une étiquette récapitulant une information morphosyntaxique selon le contexte. C'est la première couche au-dessus du niveau lexical et le niveau le plus bas de l'analyse syntaxique. Cette tâche produit des informations supplémentaires au texte en entrée ; chose très bénéfique pour les autres tâches du TAL.

Toutes les composantes du TAL traitant des niveaux linguistiques supérieurs utilisent l'information morphosyntaxique : il en est ainsi de l'analyse partielle, la désambiguïsation des sens des mots, l'affectation des fonctions grammaticales (Cutting et *al.*, 1992) et la reconnaissance d'entités nommées (Benajiba et *al.*, 2010a; Benajiba et *al.*, 2010b). Conjointement à l'analyse partielle, l'étiquetage grammatical est utilisé dans des tâches plus

complexes (Manning & Schütze, 1999), comme l'acquisition lexicale, l'extraction des informations, la recherche des termes d'indexation dans la récupération de l'information et les questions réponses, l'analyse syntaxique, etc.

Les approches de la linguistique computationnelle (LC) utilisent de plus en plus les collections de données pour l'analyse du langage. Dans ces approches quantitatives, les connaissances sont générées statistiquement, utilisant un corpus annoté. Partant de ce constat, la question de recherche principale de ce travail peut être formulée comme suit:

Etant donné que la langue amazighe standardisée est un composite de plusieurs variantes de cette langue, peut-on développer un étiqueteur morphosyntaxique avec une performance acceptable pour cette langue? Et quel critère choisir pour définir ce seuil de performance ?

Ensuite, vu l'importance de l'évaluation, comment évaluer ses performances?

Par ailleurs, vu que les textes écrits dans les langues peu dotées en ressources langagières en particulier l'amazighe ; sont encore peu nombreux, et partant du fait que la création des données étiquetées est une tâche difficile alors que l'obtention des données brutes est moins coûteuse ; comment utiliser de façon optimale ces ressources en minimisant l'intervention humaine?

Pour répondre à ces questions et atteindre l'objectif de cette recherche, nous dresserons un plan simple et cohérent à nos yeux, composé de cinq chapitres :

Dans le premier chapitre, nous présenterons d'abord les particularités et les défis du TAL pour l'amazighe, ainsi que l'état de l'art quant à l'introduction de la langue amazighe dans l'univers des Technologies d'Information et de Communication. Nous donnerons ensuite un bref aperçu sur la morphologie amazighe ; cet aperçu répond au souci de rendre meilleure la lecture des chapitres suivants de ce travail.

Le deuxième chapitre consistera en une introduction de l'ingénierie des langues et la création de ressources langagières. Nous insisterons plus particulièrement sur la conception et la méthodologie poursuivie pour établir notre corpus et son annotation.

Dans le troisième chapitre, nous procéderons à un tour d'horizon sur l'état de l'art des techniques utilisées pour la création d'étiqueteurs morphosyntaxiques. Plus précisément, nous présenterons les Support Vector Machines (SVMs) et les Conditional Random Fields (CRFs), techniques efficientes dans l'annotation des séquences.

Le quatrième chapitre servira à décrire le jeu d'étiquettes réduit utilisé pour la création d'un premier annotateur morphosyntaxique de la langue amazighe. Nous présenterons ensuite Amazighe Tag Set (AMTS), un jeu d'étiquettes enrichi. Enfin, nous présenterons les résultats des expérimentations sur l'apprentissage supervisé de l'étiqueteur morphosyntaxique amazighe utilisant les CRFs et les SVMs, plus les propriétés lexicales et contextuelles avec une phase de segmentation comme prétraitement.

Dans le chapitre cinq, nous dresserons l'état de l'art des méthodes d'apprentissage semi-supervisé, en particulier, les algorithmes d'auto apprentissage, puis nous présenterons un corpus amazighe d'environ un quart de millions de jetons de données non étiquetées, employé avec le corpus des données étiquetées susmentionné. Par la suite, nous présenterons les résultats des expérimentations de l'auto-apprentissage de l'étiqueteur morphosyntaxique amazighe, en variant les critères de choix des données sélectionnées. Nous expérimenterons également une version adaptée d'un algorithme semi-supervisée en utilisant le caractère informatif des mots hors vocabulaire et une mesure de confiance précise, afin de réduire le taux d'erreur de notre étiqueteur. Enfin nous expérimenterons l'utilisation des lexiques pour l'amélioration de la performance dudit annotateur.

Dans la conclusion, nous insisterons sur les contributions les plus importantes et ouvrirons quelques perspectives de recherche dans le domaine que nous jugeons primordiales quant à l'évolution future du traitement automatique de la langue amazighe.

CHAPITRE 1:

PARTICULARITES ET DEFIS DES NTIC POUR L'AMAZIGHE

1.1. Introduction

Tout utilisateur des technologies de l'information devrait avoir le droit d'accéder à toute information en respectant les conventions en vigueur dans son pays ou sa région. En d'autres termes, les technologies ne doivent pas représenter de nouvelles contraintes pour l'usager ; elles doivent plutôt offrir de nouvelles possibilités qui permettent à chaque individu de développer ses habiletés et d'exploiter totalement son potentiel. Par conséquent, les outils informatiques doivent continuellement être adaptés pour les rendre conformes aux systèmes d'écriture et aux conventions culturelles en usage dans chaque société.

Depuis plusieurs années, des groupes de spécialistes et des organismes internationaux et nationaux s'emploient à définir des normes et standards pour orienter le développement technologique à l'échelle internationale. L'ISO (Organisation internationale de normalisation) et la CEI (Commission électrotechnique internationale) forment un système spécialisé dans la normalisation internationale. Les comités nationaux membres de l'ISO et de la CEI participent au développement de normes internationales dans le cadre de comités[1] pour cet effet.

Dans ce chapitre, nous donnerons une brève description de la langue amazighe, nous présenterons ensuite le système d'encodage de cette langue et les travaux en relation avec son intégration aux NTIC, puis nous donnerons un aperçu de sa morphologie. Enfin, nous dresserons l'état de l'art de l'informatisation de l'amazighe.

1.2. Brève description de la langue amazighe

La langue amazighe appartient à la famille des langues Afro-Asiatiques ou Chamito-Sémitiques (Chaker, 1991; Cohen, 2007). Elle est parlée au Maroc, en Algérie, Tunisie, Libye et dans l'oasis égyptienne de Siwa. Elle est également parlée par beaucoup d'autres communautés dans certaines régions du Niger du Mali et du Burkina Faso et par les communautés amazighes immigrées partout dans le monde. Elle est utilisée par une trentaine de millions de personnes, principalement pour la communication orale.

Au Maroc, une institution[2], l'Institut Royal de la Culture Amazighe (IRCAM), dédiée à la promotion et à la sauvegarde de la culture amazighe, a été créé le 17 octobre 2001.

[1] Le SNIMA (service de normalisation industrielle marocaine) est un membre de l'ISO. Il représente le Maroc dans les travaux de normalisation.

[2] L'IRCAM a son siège à Rabat. Il a été fondé par le dahir royal 1-01-299 du 17 octobre 20011. Il dispose d'une indépendance administrative et financière.

L'amazighe a été introduit dans les médias et dans le système éducatif. L'Alphabet tifinaghe a été reconnu officiellement par le consortium Unicode le 05/07/2004 (IRCAM, 2003a), une chaîne de télévision amazighe a été lancée le premier mars 2010. L'amazighe est enseigné dans diverses écoles marocaines : un peu plus de 3 000 écoles et plus de 600 000 élèves suivent cet enseignement dans les écoles primaires. Au niveau de l'enseignement supérieur, des filières d'études amazighes et des masters ont été créés : actuellement cinq masters existent à Fès, Agadir, Mohammedia, Oujda et Rabat.

Le 01 juillet 2011, les Marocains ont voté favorablement pour la nouvelle constitution, dans laquelle il est stipulé que la langue amazighe constitue une langue officielle du pays au côté de l'arabe[3]. La constitution marocaine stipule également dans son article 5, la création d'un conseil national des langues et de la culture marocaine dont la mission principale est le développement des langues arabe et amazighe, et les diverses expressions culturelles marocaines. Il regroupe l'ensemble des institutions concernées par ces domaines. Les attributions, la composition et les modalités de fonctionnement de ce conseil devraient être déterminées par une loi organique prévue dans le programme législatif du gouvernement 2013-2015[4].

Sur le plan linguistique, la langue amazighe est caractérisée par la prolifération des dialectes en raison de facteurs historiques, géographiques et sociolinguistiques. Au Maroc, par exemple, on peut distinguer trois principaux dialectes: tarifite dans le Nord, tamazighte dans le Centre et tachlhite dans le Sud du pays. Aucun de ces dialectes n'a été considéré comme la norme nationale.

En raison de sa morphologie complexe (Chafiq, 1991 ; Boukhris et *al.*, 2008) ainsi que de l'utilisation des différents dialectes dans sa normalisation, la langue amazighe présente moult défis pour les chercheurs du TAL. Ses caractéristiques majeures sont les suivantes:

- l'amazighe dispose de sa propre graphie: le Tifinaghe (voir la section suivante), qui s'écrit de gauche à droite. La translitération en alphabet latin utilisée dans tous les exemples de ce mémoire est celle qui est présentée dans le tableau 1.1 ;
- il ne contient pas de majuscules;
- à l'instar d'autres langues naturelles, l'amazighe présente, pour le TAL, des ambigüités au niveau des classes grammaticales, des entités nommées, des sens, etc. Par exemple, au niveau grammatical le mot ⵉⵍⵍⵉ (illi) peut fonctionner comme

[3] Voir l'article 5 de la constitution marocaine, http://www.sgg.gov.ma/constitution_2011_Fr.pdf
[4] http://www.sgg.gov.ma/Portals/0/actualite/Plan-legislatif_2013_Ar.pdf

verbe à l'accompli négatif « ⵢⵓ ⵔ ⵉⵏⵉⵔ », signifiant « il n'existe pas », ou comme nom de parenté « ma fille », etc. Au niveau sémantique, un mot peut avoir plusieurs significations ; par exemple, le mot « ⴰⵖⵓⵔⵓ » (afrux) peut signifier selon le contexte l'enfant ou bien le palmier dattier, etc.

- comme la majorité des langues dont les recherches en TAL sont récentes, l'amazighe est peu doté en ressources langagières et outils du TAL. L'état de l'art des ressources et outils existant jusqu'à aujourd'hui est présenté dans la section 8 de ce chapitre ;
- les signes de ponctuation amazighe sont semblables aux signes de ponctuation adoptés au niveau international et ont les mêmes fonctions.

Tableau 1.1. Système d'écriture choisi pour la translitération en latin

Caractères tifinaghes Unicode utilisés au Maroc		Système d'écriture choisi
Code	Caractère	
U+2D30	ⴰ	a
U+2D31	ⴱ	b
U+2D33	ⴳ	g
U+2D33&U+2D6F	ⴳ ⵯ	gw
U+2D37	ⴷ	d
U+2D39	ⴹ	D
U+2D3B	ⴻ	e
U+2D3C	ⴼ	f
U+2D3D	ⴽ	k
U+2D3D& U+2D6F	ⴽ ⵯ	kw
U+2D40	ⵀ	h
U+2D43	ⵃ	H
U+2D44	ⵄ	E
U+2D45	ⵅ	x
U+2D47	ⵇ	q
U+2D49	ⵉ	i
U+2D4A	ⵊ	j
U+2D4D	ⵍ	l
U+2D4E	ⵎ	m
U+2D4F	ⵏ	n
U+2D53	ⵓ	u
U+2D54	ⵔ	r
U+2D55	ⵕ	R
U+2D56	ⵖ	G
U+2D59	ⵙ	s
U+2D5A	ⵚ	S
U+2D5B	ⵛ	c
U+2D5C	ⵜ	t
U+2D5F	ⵟ	T
U+2D61	ⵡ	W
U+2D62	ⵢ	y
U+2D63	ⵣ	z
U+2D65	ⵥ	Z
U+2D6F	ⵯ	w

1.3. Codage de la langue amazighe

Unicode est une méthode de représentation des écritures du monde en texte brut, avec un répertoire contenant près de 100 000 caractères à l'heure actuelle. Il associe à chaque caractère un numéro unique, quelles que soient les langues utilisant ce caractère, et quels qu'en soit la plateforme et le logiciel.

Le standard Unicode contient 17 plans de 64k cellules. L'espace de codage Unicode s'étend de 0 à $10FFFF_{16}$ ou, en décimale, de 0 à 1 114 111. Chaque cellule de ce standard ne peut contenir qu'un seul numéro au maximum, mais la majorité des cellules sont libres, et Unicode a réservé certaines cellules à des éléments du code qui ne correspondent à aucun caractère visible. Le premier plan, plan 00_{16}, s'appelle le plan multilingue de base (PMB). Le PMB comprend les caractères usuels dans les écritures alphabétiques, syllabiques et idéographiques ainsi que divers chiffres et symboles. Pour plus de détails sur le PMB et les autres plans Unicode voir entre autres (Zenkouar, 2004 ; Andries, 2008).

L'alphabet officiel de l'écriture amazighe est le tifinaghe. Les hypothèses portant sur les origines de cette écriture sont nombreuses, mais les études les plus récentes affirment que l'origine de l'écriture tifinaghe est une maturation autochtone à travers une stylisation progressive de dessins rupestres. Cette hypothèse est confirmée par plusieurs découvertes archéologiques de modèles scripturaux dont certains sont antérieurs à l'alphabet phénicien. La figure 1.1 présente un exemple d'écriture rupestre.

Les études sur l'alphabet tifinaghe sont peu nombreuses et beaucoup de recherches spécialisées restent à faire. La seule certitude est que Imazighes disposaient d'un système d'écriture propre qui remonte à une époque où plusieurs cultures n'en avaient pas encore. Cet alphabet (Figure 1.1) a subi des modifications et des variations inévitables depuis son origine jusqu'à nos jours, passant du libyque jusqu'aux néo tifinaghes, en passant par le tifinaghe saharien et les tifinaghes touaregs.

Les lettres tifinaghes s'écrivent de gauche à droite et ne contiennent pas de majuscule. En 2003, les tifinaghes ont été adoptées par le conseil d'administration de l'IRCAM comme alphabet officiel de la langue amazighe. Elles ont été ajoutées officiellement au jeu universel de caractères Unicode, dans sa version 4.1 (Figure 1.2), le 24 juin 2004 lors de la réunion de Toronto. Les caractères tifinaghes occupent la plage[5] Unicode 2D30-2D7F[6].

[5] Les caractères tifinaghes sont donnés en hexadécimal. Pour la conversion entre numéros de caractères et caractères eux-mêmes, les codages UTF8 et UTF16 …etc. voir http://hapax.qc.ca/conversion.fr.html.

[6] http://www.unicode.org/charts/PDF/U2D30.pdf

Figure 1.1. Exemple de gravure rupestre contenant des lettres tifinaghes[7].

La zone tifinaghe Unicode est divisée en quatre sous-ensembles de caractères tifinaghes (Zenkouar 2004):

- le jeu de base utilisé par l'Institut Royal de la Culture Amazighe ;
- le jeu étendu de l'IRCAM ;
- d'autres lettres tifinaghes en usage ;
- des lettres touarègues modernes dont l'usage est attesté.

[7] Foum Chena/Tinzouline-Zagora, vallée de Draa, Maroc : inscription piquetée sur un support vertical (Skounti et al., 2004).

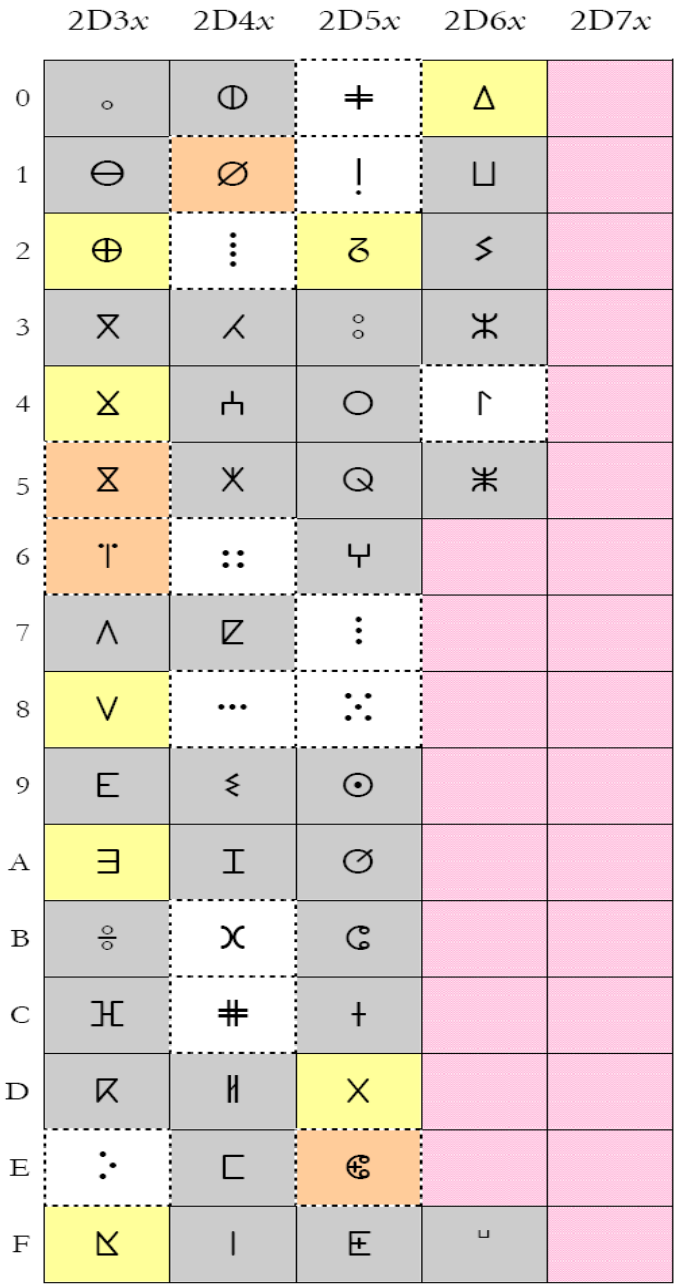

Figure 1.2. Bloc tifinaghe Unicode 4.1

Le nombre actuel des caractères tifinaghes est 59[8], après les deux amendements des versions 6.0 et 6.1 d'Unicode. Les quatre caractères ajoutés sont :

- le caractère de ponctuation *tazarast*, marque tifinaghe de séparation ⵰ dont le code est 2D70 ;

[8] Afin de rechercher, visualiser et lire les propriétés des caractères tifinaghes, on peut utiliser à titre d'exemple l'outil web Uniview : http://rishida.net/scripts/uniview/, ou BabelMap, outil qui marche uniquement sous Windows : http://www.babelstone.co.uk/Software/BabelMap.zip.

- le signe de jointure des consonnes  , dont le code est 2D7F, ce signe sert à indiquer que le caractère précédent et le caractère suivant font partie d'un groupe bi-consonne;
- la lettre YE ⵯ dont le code est 2D66 ;
- la lettre YO ⵰ dont le code est 2D67.

En ce qui concerne le jeu de base utilisé au Maroc, l'IRCAM a gardé les phonèmes les plus pertinents pour l'écriture de la langue amazighe au Maroc (Ameur et *al.*, 2004). Ainsi, le nombre des caractères tifinaghes enseignés est 33 (27 consonnes, 2 semi-consonnes et 4 voyelles). Au niveau encodage, Unicode encode 32 caractères uniquement, dont un est le caractère *tamatart*, qui représente la lettre modificative de labialisation « ◌ⷯ » et qui permet de former les deux lettres tifinaghes ⵖ◌ⷯ (g^{w}) et ⴽ◌ⷯ (k^{w}).

1.4. Les normes marocaines de saisie des tifinaghes

Afin de saisir des caractères Unicode, plusieurs méthodes sont possibles : selon le programme, le document ou la plateforme utilisée. La plus simple consiste à utiliser les touches ou le clavier[9].

Pour écrire, on peut utiliser un jeu de caractères qui permet de lire du chinois au Maroc ou du tifinaghe en Inde. Cependant, on a besoin d'un moyen qui permette à l'utilisateur de réagir avec le système. L'une des tâches principales a été de prendre en charge les jeux plus élaborés ou multiples de caractères requis par les diverses applications auxquelles sont destinés les claviers de nos jours. On est parvenu à ce résultat en affectant plusieurs caractères graphiques ou fonctions de commande à chacune des touches d'un clavier, principalement dans le module alphanumérique.

La norme ISO/CEI 9995, Technologies de l'information - Disposition des claviers conçus pour la bureautique, comprend les parties suivantes:

- Partie 1 : Principes généraux pour la disposition des claviers ;
- Partie 2 : Module alphanumérique ;
- Partie 3: Dispositions complémentaires de la zone alphanumérique du module alphanumérique ;
- Partie 4: Module numérique ;

[9] Pour définir son propre clavier on peut utiliser l'outil MSKLC pour le système Windows: http://www.microsoft.com/grobaldev/tools/msklc.mspx, et Ukelele et Key Layout Maker pour Macintosh: http://scripts.sil.org/ukelele, http://scripts.sil.org/key/ayantmaker.

- Partie 5: Module d'édition ;
- Partie 6: Module de fonctions ;
- Partie 7: Symboles employés pour la représentation des fonctions ;
- Partie 8: Affectation de lettres aux touches d'un clavier numérique.

L'application de la partie 1 de l'ISO/CEI 9995, dans la conception des claviers, fournit une interface uniforme et prévisible entre l'utilisateur et les machines de bureau, en divisant le clavier en zones et modules et en affectant des fonctions aux touches. Afin d'utiliser l'alphabet tifinaghe, la norme de prescription des claviers conçus pour la saisie des caractères tifinaghes, présentée dans PNM 17.6.000[10] (Zenkouar & Labonté, 2006b), fixe les prescriptions des claviers conçus pour la saisie des caractères tifinaghes. Elle consiste en l'affectation des codes de caractères aux touches du clavier alphanumérique (Outahajala & Zenkouar, 2007), en suivant le système de numérotation spécifié dans la norme ISO/CEI 9995-1.

Sur l'agencement du clavier à 48 touches graphiques de la norme internationale ISO/CEI 9995-2, les rangées sont identifiées par des lettres s'échelonnant de A à E, la rangée A correspondant à la barre d'espacement et la rangée E correspondant à la rangée la plus éloignée de l'utilisateur assis devant son clavier. Les colonnes sont identifiées par des chiffres variant de gauche à droite (Figure 1.3).

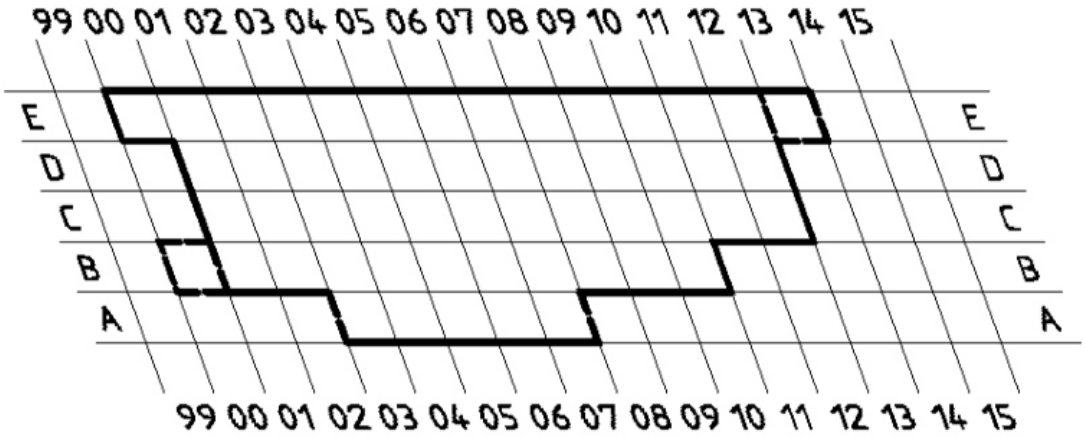

Figure 1.3. Grille du clavier harmonisé à 48 touches graphiques

[10] Ce projet de norme marocaine ainsi que le PNM 17.2.000 ont été visés par l'arrêté du ministre de l'industrie, du commerce et de la mise à niveau de l'économie n° 1422-06 du 17 joumada II 1427 portant homologation de normes marocaines (B.O. n° 5444 du 3 août 2006).

Cette norme nationale définit deux claviers. Le premier clavier, appelé clavier de base, permet la saisie des 33 caractères utilisés au Maroc (Figure 1.4). Ce clavier contient l'ensemble des lettres tifinaghes enseignées dans les écoles marocaines, sans décomposition pour les deux lettres qui contiennent graphiquement une modificative de labiovélarisation, afin de ne recourir à aucun autre moyen de composition pour saisir ces lettres. La modificative est aussi présentée sur le clavier comme caractère à part, pour permettre une éventuelle composition dans le futur de cette lettre avec d'autres caractères tifinaghes.

Figure 1.4. Clavier tifinaghe de base.

Le deuxième clavier, appelé clavier étendu[11], permet la saisie des 55 caractères tifinaghes (Figure 1.5) encodés dans la version 4.1 d'Unicode. Le clavier étendu a été normalisé afin de permettre d'accéder à l'ensemble des caractères tifinaghes. D'autres caractères, qui ont été jugés utiles, ont été intégrés au clavier tels que: le LIANT SANS CHASSE, permettant de composer des ligatures, l'ANTILIANT SANS CHASSE, empêchant la formation des ligatures, le TIRET CADRATIN, l'ESPACE INSÉCABLE, etc.

[11] La norme correspondant à ce clavier étendu ainsi que la norme de classement des chaines de caractères amazighes devraient être mis à jour afin de prendre en compte les 4 nouveaux caractères tifinaghes ajoutés à Unicode, à savoir : le caractère de ponctuation *tazarast*, le signe de jointure des consonnes, la lettre *YE* et la lettre *YO*.

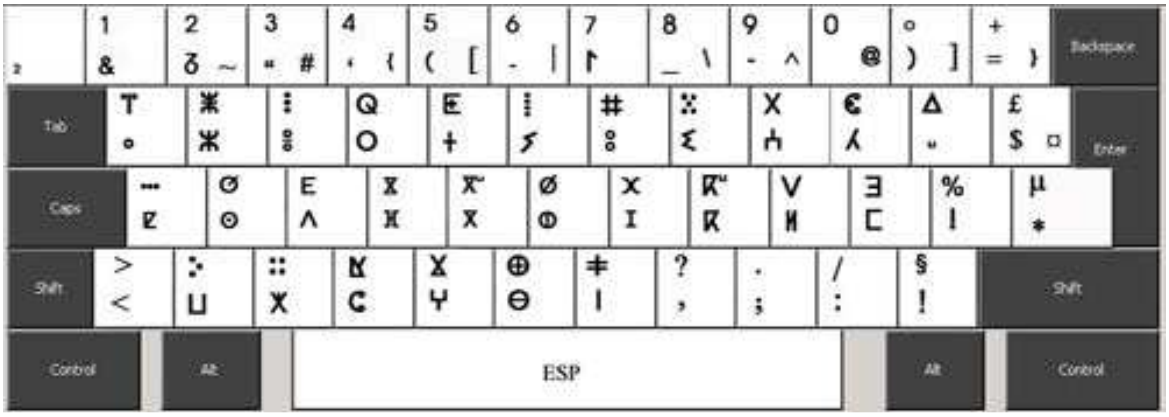

Figure 1.5. Clavier tifinaghe étendu

Afin de permettre aux usagers l'utilisation des claviers et polices de caractères (Ait Ouguengay et *al.*, 2007), le CEISIC[12] a développé plusieurs outils dans ce sens. Ils sont téléchargeables sur le site Web de l'IRCAM[13].

1.5. Les normes marocaines de classement des chaînes de caractères

La norme nationale indique l'ordre des lettres de l'alphabet, de même que la correspondance phonétique de cet alphabet avec les alphabets latin et arabe. Un tel ordre de codage est toutefois insuffisant pour produire un tri informatique correct susceptible d'utiliser par exemple, des caractères spéciaux imbriqués.

La solution pour le classement correct des chaînes de caractères réside dans la décomposition des données initiales, tout en respectant l'ordre lexicographique traditionnel et en assurant une prévisibilité absolue. Cela nécessite quatre niveaux de décomposition :

- le premier niveau de décomposition permet de rendre l'information à trier indépendante de la casse et des signes diacritiques ; en plus, il permet d'éliminer les caractères n'ayant aucun ordre préétabli dans quelque culture que ce soit ;
- le deuxième niveau de décomposition différencie les quasi-homographes, chaînes qui diffèrent seulement par leurs signes diacritiques. En français, les principaux dictionnaires suivent la règle suivante : les accents sont ignorés lors du classement sauf dans les cas d'égalité homographique ;
- le troisième niveau de décomposition différencie les quasi-homographes dont la seule différence réside dans la hauteur de casse ;

[12] Le CEISIC, Centre des Etudes Informatiques, Systèmes d'Information et Communication, est un centre de recherche de l'IRCAM chargé entre autres de la promotion des tifinaghes.

[13] http://www.ircam.ma.

- le quatrième niveau de décomposition brise les dernières égalités entre quasi-homographes qui ne diffèrent que par la présence de caractères spéciaux. Cette différenciation est essentielle pour assurer la prévisibilité absolue des tris et pour trier des chaînes composées uniquement de caractères spéciaux.

Ces quatre niveaux de décomposition peuvent être structurés en utilisant une clé à quatre niveaux. Cette clé sera constituée par la concaténation des sous-clés de la plus haute importance à la plus basse.

La concaténation de ces quatre sous-clés nous fournit la clé utilisée lors de la comparaison des chaînes de caractères. Et c'est exactement le principe sur lequel s'appuie la norme internationale de classement des chaînes de caractères ISO/CEI 14 651 qui utilise une table modèle contenant tous les caractères du jeu universel ISO/CEI 10 646 avec des valeurs attribuées aux quatre sous-clés précitées.

Dans l'ensemble des 55 caractères tifinaghes (de la version 4.1 d'Unicode) («ⵣ » (LETTRE TIFINAGHE YADJ ACADÉMIE BERBÈRE) est plus grande que «ⴰⵏ » et plus petite que «ⴰⵑ »)

⵿ ⴰ ⴹ ⵃ < ⵿ ⵣ ⵿ ⵇ < ⵿ ⴰ ⵑ ⴹ ⵙ < ⵜ ⴹ ⵃ ⴹ ⵏ ⵿ ⵢ

Le classement selon l'ordre des caractères de l'ISO/CEI 10646 est:

⵿ ⵣ ⵿ ⵇ < ⵿ ⴰ ⴹ ⵃ < ⵿ ⴰ ⵑ ⴹ ⵙ < ⵜ ⴹ ⵃ ⴹ ⵏ ⵿ ⵢ .

Un autre exemple qui montre l'intérêt de cette norme est l'utilisation du caractère espace :

⵿ ⵔ ⵔ ⵿ ⵏ [deux espaces] < ⵿ ⵔ ⵔ ⵿ ⵏ [une espace] < ⵿ ⵔⵔ ⵿ ⵏ

Alors que le classement correct est :

⵿ ⵔⵔ ⵿ ⵏ < ⵿ ⵔ ⵔ ⵿ ⵏ [une espace] < ⵿ ⵔ ⵔ ⵿ ⵏ [deux espaces]

Dans ce sens, aucun classement des caractères n'est par ailleurs valable pour toutes les langues, il faut explicitement déclarer la liste de différences et la table modèle commune de la norme ISO/CEI 14651 (**delta)** pour la langue et pour la culture cibles. Pour plus de détails sur la norme marocaine de classement des chaînes de caractères et son delta, voir (Zenkouar & Labonté, 2006a; Outahajala, 2008).

1.6. Identification de la langue et autres renseignements linguistiques

Les indicatifs pour la représentation des noms de langues sont définis dans la norme internationale ISO 639.

Cette norme est composée actuellement de trois parties:

- L'ISO 639-1 utilise des indicatifs sur deux caractères ;
- L'ISO639-2 utilise des indicatifs sur trois caractères ;
- L'ISO 639-3 complétant l'ISO 639-2.

L'ISO 639 connaît des ajouts et des changements dans les indicatifs des langues[14]. C'est le cas de l'amazighe, auquel on a assigné, le 25 octobre 2012, l'indicatif «**zgh**» (IRCAM, 2012). Le tableau 1.2 présente les différentes parties de l'ISO 639[15].

Tableau 1.2. Les trois parties de l'ISO 639.

Partie	Type indicatif	Nombre d'indicatifs	Exemples
ISO 639-1	à deux lettres	136	«fr » pour le français « ar »pour l'arabe
ISO 639-2	à trois lettres	484	« fra »et « fre »pour le français « zgh » pour l'amazighe.
ISO 639-3	A trois lettres	7581	« fra »pour le français « rif » pour le rifain

L'ISO 3166-1 présente un deuxième type de renseignement culturel. L'ISO 3166 comprend trois répertoires différents: alpha-2, alpha-3 et numérique-3. Le tableau 1.3 présente quelques indicatifs de pays ISO 3166-1 à deux lettres.

Tableau 1.3. Exemples d'indicatifs de pays selon la norme ISO 3166.

Pays	Indicatif

[14] Liste sur http://www.loc.gov/standards/iso639-2/php.

[15] La demande d'ajouter un nouveau code de langue se fait en remplissant le lien suivant de l'ISO 639 : http://www.loc.gov/standards/iso639-2/php/iso639-2form.php

Algérie	DZ
Canada	CA
Maroc	MA
Tunisie	TN

L'ISO 3166-2 est une autre norme codée en quatre lettres permettant de désigner des subdivisions en pays (Voir exemples dans le tableau 1.3.).
L'ISO 15924 définit les indicatifs des écritures[16]. Il contient près de 150 systèmes d'écritures, y compris le tifinaghe. Le tableau 1.4 présente quelques exemples d'indicatifs d'écriture[17].

L'IETF (Internet Engineering Task Force), qui élabore les standards de l'internet et établit les demandes de commentaires (RFC), a produit, en 2006, le RFC 4646, qui définit une série d'étiquettes linguistiques. L'étiquette linguistique commence par l'étiquette "langue", indicative de langue ; elle est suivie ensuite par une des cinq étiquettes: l'écriture, la région, les variantes, les extensions et l'usage privé. "zgh-Tfng" est l'exemple de l'étiquette linguistique pour désigner l'écriture tifinaghe pour la langue amazighe[18].

1.7. Morphologie de la langue amazighe

La plupart des mots amazighes ont des racines consonantiques. Les racines des mots peuvent avoir une, deux, trois ou quatre consonnes ; parfois, ils s'étendent à cinq. Les mots sont créés à partir de la combinaison d'une racine et d'un schème. A partir d'une racine donnée, on peut avoir plusieurs dérivés verbaux et nominaux. La nature de la dérivation verbale peut être de plusieurs types: causatif, réciproque, réfléchi, passif, etc. Les dérivés nominaux sont: les noms d'agent, les noms d'action, les noms d'instrument, les noms de lieu, etc.

Par exemple, le mot ⵄⵎⴽⵔⴰⵣ "amkraz" (agriculteur) est un nom d'agent construit à partir de la racine ⴽⵔⵣ "krz" (notion de culture) en suivant un modèle précis ⵄⵎ12ⵄ3 "am12a3" (Figure 1.6), où les numéro 1, 2 et 3 sont remplacés respectivement par la première, la deuxième et la 3ème consonne de la racine.

[16] La liste complète est consultable sur www.unicode.org/iso/15924_fr.html.
[17] La liste complète est consultable sur www.unicode.org/iso/15924/iso15924_fr.
[18] Pour plus de détail sur la syntaxe exacte de ces étiquettes, voir la RFC 4646 : https://www.ietf.org/rfc/rfc4646.txt.

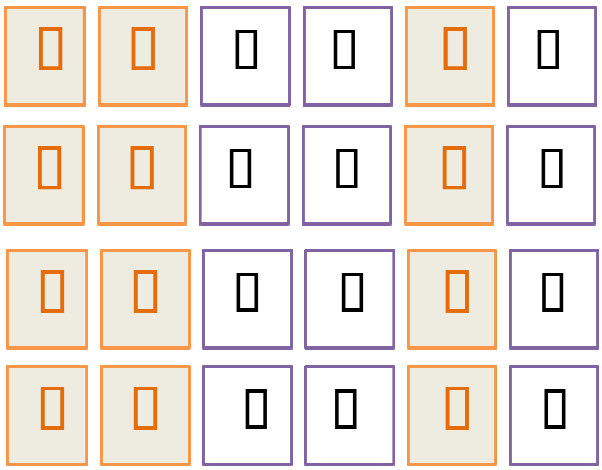

Figure 1.6. Construction des mots suivant un modèle.

Un exemple de dérivation verbale de réciprocité en amazighe est: ⵣⵣ "RZ" (casser)/ ⵜⵜⵣⵣ "mmRZ" (casser réciproquement).

En ce qui concerne l'orthographe, le système adopté au Maroc pour donner lieu aux mots écrits est basé sur un ensemble de règles et de principes (Ameur et *al.*, 2004, 2006, Boukhris et *al.*, 2008). Un graphème, un mot écrit, est une succession de lettres, qui peuvent parfois être une lettre, délimité par des espaces ou des signes de ponctuation.

Afin de faciliter la lecture des chapitres suivants, nous présentons dans ce qui suit certaines règles orthographiques[19] de la langue amazighe, suivies au Maroc, illustrées par des exemples.

- Les noms sont constitués d'un seul mot se produisant entre deux espaces ou bien une espace et un point de ponctuation. Les noms sont toujours accompagnés des affixes morphologiques du genre (masculin / féminin), du nombre (singulier / pluriel) et d'état (libre / construit) comme le montrent les exemples suivants:

- ⴰⵎⵣⴷⴰⴳ (habitant) (masculin singulier), "amzdaG", ⵜⴰⵎⵣⴷⴰⴳⵜ (habitante) (féminin singulier) "tamzdaGt", ⵉⵎⵣⴷⴰⴳⵏ (habitants) (masculin pluriel) "imzdaGn". L'état construit du nom ⴰⵎⵣⴷⴰⴳ (un habitant) est ⵓⵎⵣⴷⴰⴳ "umzdaG";

- Les noms de parenté constituent une classe particulière. Ils forment avec les pronoms personnels un mot entier, par exemple: ⴰⴰⴰⵅ (votre père), "babak" ;

[19] Pour d'amples informations sur les règles orthographiques de la langue amazighe, voir à titre d'exemple (Ameur et al., 2004 ; Ameur et al., 2006b ; Boukhris et al., 2008).

- Les noms de qualité, appelé également adjectifs, constituent un seul mot avec ses indicateurs morphologiques du genre (masculin/ féminin), du nombre (singulier/ pluriel), et d'état (libre/construit) ;

- Les verbes sont des mots graphiques simples avec leur inflexion (personne, nombre, aspect) ou morphèmes dérivatifs ; en voici quelques exemples:

  - ⵜ ⵜ ⴰ ⵣⵣⵍ "ttazzl" signifiant "cours (inacompli) " ;
  - Le verbe est toujours séparé par un blanc des éléments qui le précédent et des éléments qui le suivent, i.e.: ⵙ ⴰ ⵙⵉ (il a pris) ⵜ ⵏ (les)/ "yasi tn";
  - ⴰ ⴰⵏ ⵜ ⵏ ⵙ ⴰ ⵙⵉ "ad tn yasi" (qui signifie "il les prendra");

- Les pronoms sont isolés des mots auxquels ils se réfèrent. Les pronoms en langue amazighe sont soit démonstratifs, exclamatifs, indéfinis, interrogatifs, personnels, possessifs ou relatifs. Un exemple de leur utilisation peut être comme suit:

  - ⵏ ⵏ ⴰ "nna" (signifiant qui) dans l'exemple: ⴰ ⵜⵓⵉⵏ (le chemin) ⵏ ⵏ ⴰ (que) ⵜ ⵔ ⵔ ⵉ ⵜ /ⴰ (tu es passé) "abrid nna tkkit/d";

- Les adverbes sont subdivisés en adverbes de lieu, de temps, de quantité, de manière, et les adverbes interrogatifs. Un exemple d'un adverbe est:

  - ⵣ ⵣ ⴰ ⵔ (tous) "qqaH", comme dans l'exemple: ⵣ ⵣ ⴰ ⵔ (tous) ⵛ ⵉ ⵏ ⵏ ⵏ (les gens) ⵏ ⴰ ⵜ ⵜ ⵉ ⵏ ⵉ ⵏ (parlent)… "qqaH middn da ttinin... ".

- Les focalisateurs, les interjections et les conjonctions sont écrits en un mot entre deux espaces ou bien une espace et un point de ponctuation. Par exemple :

  - ⵛ ⵓ (si) / "mr";

- Les prépositions sont un ensemble de caractères indépendants par rapport au nom qu'elles précèdent ; cependant, si la préposition est suivie d'un pronom personnel, la préposition et le pronom personnel forment une seule chaîne délimitée par des blancs ou bien un blanc et une marque de ponctuation. Par exemple:

  - ⵖ ⵓ (à, vers) "Gr" + le pronom ⵉ (moi) "i" donnent ⵖ ⴰ ⵓⵉ /ⵖ ⵔ ⵓⵉ "à moi, avec moi" (Gari/ Guri) ; qui sont des variantes de la préposition ⵖ ⵓ quand elle est utilisé conjointement avec le pronom personnel ⵉ ;

- Les particules sont toujours isolées. Ils sont de plusieurs types:

  - Les particules aspectuelles telles que ⴰ ⵣ ⵣ ⴰ (aqqa), ⴰ ⵓ (ar), ⴰ ⵏ (ad);
  - La particule de négation ⵔ ⵓ (ur);
  - Les particules d'orientation, comme ⵏ ⵏ (là-bas) ;
  - La particule de prédication ⵏ "d".

Un exemple d'utilisation de la particule ⵉ ⵉ : ⴰ ⵓ ⵙ (prends) ⵜ (le) ⵏ ⵏ (labàs) "awi t nn".

- Les déterminants prennent toujours la forme d'un seul mot délimité par deux espaces. Les déterminants sont divisés en articles, démonstratifs, exclamatifs, articles indéfinis, interrogatifs, chiffres ordinaux, possessifs, présentatifs et quantificateurs. ⴽ ⵓ ⵍ ⵍ ⵓ (chaque) "kullu" est un exemple de quantificateur ;
- Les marques de ponctuation en amazighe marocain sont similaires aux marques de ponctuation adoptées par les langues internationales. Elles ont les mêmes fonctions.

1.8. État de l'art de l'informatisation de l'amazighe

La langue amazighe ne possède pas encore de ressources langagières et d'outils suffisants pour son TAL. Les ressources et outils existants peuvent être subdivisés en deux familles, qui seront traitées dans ce qui suit.

1.8.1. Ressources computationnelles générales

Ces ressources peuvent être subdivisées en trois catégories :

- *Travaux relatifs à la promotion des Tifinaghes*

Ces travaux consistent en :

- la création des claviers et polices de caractères dédiés à l'écriture Tifinaghe (IRCAM, 2003b; IRCAM, 2004), sous les différents systèmes d'exploitation (Linux, Macintoch et Windows). C'est par la suite que ces éditeurs ont commencé la distribution des systèmes d'exploitation en y intégrant les claviers et polices amazighes. A titre d'exemple, la distribution linux Mandriva les a intégrés en 2006. Par ailleurs, Microsoft n'a introduit pour la première fois la graphie tifinaghe et les claviers à son système d'exploitation qu'à la fin 2012 ;
- la création des supports et matériaux éducatifs pour son apprentissage[20]. Dans le même sens, une équipe de l'IRCAM spécialisée en didactique et pédagogie a collaboré, avec le Ministère de l'Education Nationale marocain, pour intégrer l'enseignement de l'amazighe destiné au primaire dans le programme GENIE[21] (GÉNéralisation des Technologies d'Information et de Communication dans l'Enseignement au Maroc) ;

[20] http://www.ircam.ma/ecoleamazighe/index.html et http://www.ircam.ma/tifinaghe/index.html

[21] L'objectif principal en est de permettre d'enrichir les aptitudes et les compétences des enseignants dans leurs pratiques d'enseignement. Pour d'amples informations sur ce programme, consulter: http://www.taalimtice.ma/.

- projet de translittération des textes écrits en alphabet tifinaghe vers l'alphabet arabe ou latin (Ataa Allah et *al.*, 2013). Cet outil permet d'utiliser plusieurs formats de fichiers et contient une fonctionnalité permettant de faire des correspondances entre caractères. Néanmoins, la translittération des textes amazighes écrits en alphabet arabe vers l'alphabet tifinaghe n'est pas opérationnelle et ne peut pas l'être par simple correspondance des caractères des deux alphabets un à un[22] ;
- Le CEISIC a crée plusieurs polices et claviers pour écrire en tifinaghe, et ce pour plusieurs système d'exploitation[23]. Lguensat (2012) a proposé une écriture cursive pour cet alphabet.

- *Dictionnaires et corpus*

Il existe très peu de ressources numériques en relation avec les dictionnaires, les lexiques électroniques et les corpus. Mais on peut citer à cet égard:

- L'application GBDLA (Gestion de la Base de Données Lexicographiques Amazighe) (Iazzi & Outahajala, 2008), conçue afin de fournir, au niveau de l'encodage dont la conception est extensible et modulaire, toutes les informations jugées nécessaires, et ce pour chaque entrée lexicale : entrée normée, prononciation(s), catégorie et sous-catégories grammaticales, nature dérivationnelle, aires géolinguistiques, origine en cas d'emprunt ou de néologisme, domaines, variations flexionnelles ou supplétives, etc. Ces informations permettent également d'interroger, sous différents angles, la base de données, pour relever les ontologies du lexique par domaine : lexique de l'agriculture, lexique de l'artisanat, etc., des familles dérivationnelles, etc. Cette application est opérationnelle en deux versions, une latine et l'autre en tifinaghe. La version latine de cette application contient environ 2 000 entrées lexicales ;
- un corpus d'entités nommées d'environ 23 000 noms de places et 200 noms de personnes (Cieri & Liberman, 2008). Ce corpus n'est pas exploitable dans sa version actuelle, car il n'a pas été révisé et validé par un linguiste ;

[22] Pour implémenter une telle fonctionnalité, il faut prendre en considération le fait que l'Arabe n'utilise pas les voyelles courtes dans sa transcription. Aussi, il faut veiller à ce que les textes translitérés en tifinaghes respectent les règles d'orthographe de l'Amazighe.

[23] Ils sont téléchargeables sur le site dudit institut : http://www.ircam.ma/fr/index.php?soc=telec.

- Un corpus des textes extraits du journal le Monde Amazighe[24] et autres publications et supports de l'IRCAM, collecté par le LDC (Cieri & Liberman, 2008). Tous les textes à part ceux des matériaux de l'IRCAM nécessitent un travail de resegmentation selon les règles d'orthographe de l'amazighe ;
- une base de données terminologique, permettant la compilation et la gestion de la terminologie aménagée par l'Institut (El Azrak & El Hamdaoui, 2011). Une version mobile de cette application, nommée LEXAM, a été implémentée[25] sur Google Play ;
- En collaboration avec des étudiants, le CEISIC a collecté un ensemble de textes des différentes variantes de l'amazighe. Ce corpus contient environ 70 000 mots (Boulaknadel & Ataa Allah, 2011). Il n'est pas mis à la disposition du public ;
- une base de données faite de corpus littéraires, consistant en une application réseau de la collecte des données textuelles et audiovisuelles en relation avec la langue et culture amazighes (Aït Ouguengay et *al.*, 2012).

- *Reconnaissance optique des caractères:*

Plusieurs travaux de reconnaissance optique des caractères (OCR) ont été menés pour la langue amazighe, tels que :

- le premier travail de reconnaissance des caractères tifinaghes a été fait en 2009 (Ait Ouguengay et *al.*, 2009). Il est basé sur les réseaux de neurones. La taille totale du corpus d'apprentissage est d'environ 28 000 images des caractères tifinaghes imprimables, en variant la police de caractère, la taille, etc. Les résultats obtenus sur le corpus de test sont d'environ 98% ;
- Amrouch et ses coéquipiers (Amrouch et *al.*, 2010) ont utilisé les HMMs pour un corpus constitué des caractères manuscrits ;
- El Ayachi et ses coéquipiers (El Ayachi et *al.*, 2010), ont utilisé les moments invariants et la transformée de Walsh dans les phases de prétraitements, et l'extraction des caractéristiques et la programmation dynamique pour la phase de reconnaissance ;
- Dans le travail d'Es Saady et ses co-auteurs (2011), ils se sont basés sur les lignes horizontales et verticales et sur un perceptron multicouche pour la reconnaissance

[24] Ce journal a dernièrement mis en ligne son site web : http://www.amadalpresse.com/

[25] https://play.google.com/store/apps/details?id=ma.ircam.dictionnaire

des caractères tifinaghes imprimables et manuscrits. Le résultat obtenu est de 96.32% pour les caractères manuscrits et 99.32% pour les caractères imprimés.

1.8.2. Ressources TAL pour l'amazighe

Peu de ressources à ce niveau ont été développées pour l'amazighe. Nous présentons ci-après certains de ces travaux:

- Un pack intitulé LCTL berber pack 1.0 regroupe un ensemble d'outils réalisés par le LDC et l'ELDA, en collaboration avec l'IRCAM. Ces outils sont constitués d'un translitérateur, un segmenteur en phrases et en mots et un système de reconnaissance des entités nommées (Simpson et *al.*, 2008, Ait Ouguengay & Bouhjar, 2010). Ce système n'est pas mis à la disposition du public et aucune évaluation le concernant n'a été rapportée ;
- un analyseur morphologique des noms amazighes (Raiss & Cavalli-Sforza, 2012). Le taux de fiabilité des résultats de cet outil atteint plus de 90% dans l'analyse de 1.541 noms extraits du corpus de LDC (cité dans la sous section 1.7.1 ci-dessus) ;
- un correcteur d'orthographe (Es Saady et *al.*, 2009), basé sur l'algorithme de Hanspell. Ce programme intègre des règles du pluriel et du féminin pour proposer les corrections de l'orthographe. Ce programme a été testé sur Open office et Firefox. Néanmoins, il n'est pas disponible sur le web ;
- un concordancier (Boulaknadel, 2009), permettant la recherche d'un mot quelconque dans un ensemble de textes afin d'étudier son emploi. Cet outil n'est pas disponible sur le web ;
- un pseudo-racineur (Ataa Allah & Boulaknadel, 2010), essaie de regrouper les mots de forme apparentée. Ce programme élimine un ensemble de suffixes et de préfixes selon des listes pré établies ;
- un conjugueur des verbes de la langue amazighe, en appliquant les règles de conjugaison définies par le centre d'aménagement linguistique (CAL) de l'IRCAM dans le manuel créé par Laabdelaoui et ses coéquipiers (Laabdelaoui et *al.*, 2012). Ce programme conjugue les verbes amazighes présentés dans le manuel précité. Cependant, pour les autres verbes, la détermination de la classe du verbe par un linguiste est nécessaire pour le verbe en question avant de pouvoir le conjuguer.

1.9. Synthèse

Le domaine des technologies de l'information connaît un développement accéléré. Les concepteurs et les éditeurs se tiennent continuellement au fait des dernières avancées, afin de mettre à jour et d'améliorer les outils et systèmes qu'ils mettent à la disposition du public. Dans ce sens, les normes et les standards informatiques sont établis afin de permettre aux éditeurs des logiciels d'avoir des référentiels de base de développement informatique et aux usagers des technologies d'information de supporter les systèmes d'écriture et de respecter les conventions culturelles en usage dans la société en question.

Dans ce chapitre, nous avons présenté une brève description de la langue amazighe, son encodage, les méthodes de saisie, le classement des chaines de caractères amazighes et autres renseignements linguistiques de cette langue. Ensuite, nous avons présenté des défis qui font face au traitement automatique de cette langue. Ensuite, et pour une meilleure lecture du reste de ce mémoire, en particulier le chapitre 2 suivant, nous avons donné une présentation succincte de la morphologie de la langue amazighe. Enfin nous avons présenté l'état de l'art de son intégration aux nouvelles technologies de l'informatisation.

D'après notre investigation, on constate que la langue amazighe a un long chemin à parcourir en TAL. Aussi, les travaux sont divers mais sans vision claire et les efforts des chercheurs en TAL amazighe demandent à être fructifiés par la synergie de groupe et des échanges mutualisés.

Dans le chapitre suivant, nous présenterons la démarche que nous avons suivie afin de construire le premier corpus annoté morphosyntaxiquement pour la langue amazighe.

CHAPITRE 2:

CONSTRUCTION D'UN CORPUS ANNOTE DE LA LANGUE AMAZIGHE

2.1. Introduction

La linguistique computationnelle utilise de plus en plus les collections de données pour l'analyse du langage. Dans cette approche quantitative, les connaissances sont générées statistiquement.

Après la publication en 1957 de son livre *Structures Syntaxiques*, Chomsky a critiqué l'utilisation des chaînes de Markov, en tant qu'approche statistique utilisée dans la modélisation du langage naturel, en mettant le point sur le fait qu'elles ne sont pas capables de modéliser les structures récursives. Il a soutenu l'idée qu'un corpus ne peut pas servir comme outil utile pour le linguiste vu que certaines phrases ne se produiront pas parce qu'elles sont évidentes, d'autres parce qu'elles sont fausses, d'autres encore parce qu'elles sont impolies.

La linguistique computationnelle est restée influencée par cet avis et les approches symboliques à base de règles prédéfinies étaient les plus utilisées jusqu'aux années 80.

Mais par la suite, plusieurs facteurs ont incité à l'utilisation des techniques statistiques à base de corpus, à savoir :

- Le développement des techniques de traitement de l'information, ainsi que les supports de stockage ;
- les nouvelles avancées en informatique et en ingénierie des langues ;
- les travaux importants des consortiums dans la production des ressources linguistiques et outils pour le TAL ;
- l'existence de larges corpus en ligne pour certaines langues.

Par ailleurs, les attentes des utilisateurs sont devenues plus réalistes. Par exemple, en traduction automatique, les utilisateurs acceptent les traductions même approximatives, sur lesquelles ensuite, ils introduisent, le cas échéant, les changements qu'ils jugent nécessaires.

La langue amazighe, comme la plupart des langues qui n'ont que récemment commencé la recherche en TAL, souffre de la pénurie d'outils et de ressources pour son traitement automatique. Dans cette optique, et vu que les corpus constituent la base de la recherche dans le domaine des technologies langagières de l'homme, nous avons fixé comme objectif stratégique la construction d'un grand corpus annoté pour la langue amazighe.

Dans ce chapitre, nous introduirons l'ingénierie des langues, en particulier l'ingénierie du corpus. Ensuite, nous aborderons la méthodologie suivie pour construire la première ressource

langagière annotée morphosyntaxiquement pour la langue amazighe. Enfin, nous donnerons les utilisations actuelles et éventuelles de cette ressource.

2.2. Ingénierie des langues

Plusieurs méthodes statistiques novatrices en relation avec le traitement du langage naturel ont vu le jour et ont marqué ce domaine de recherche. Citons, à titre d'exemples, les domaines suivants: la recherche d'information textuelle (Salton, 1972), la reconnaissance automatique de la parole (Baker, 1975 ; Jelinek, 1976), l'étiquetage morphosyntaxique (Ratnaparkhi, 1996 ; Brill, 1995), l'analyse syntaxique (Church, 1988), la traduction automatique (Brown et *al.*, 1990) et la compréhension de la parole (Levin & Pieraccini, 1992).

Les recherches en TAL ont toujours eu pour finalité la recherche d'une robustesse optimale sur des données ou des applications réelles. Elles se donnent pour objectif prioritaire la réalisation de systèmes efficaces et opérationnels dont les performances sont estimées au cours de campagnes d'évaluation significatives portant sur des situations proches du réel.

Cette recherche d'efficacité et ce recours systématique à l'évaluation relèvent de la démarche d'ingénierie. Le fait que cette démarche est désormais appliquée à l'ensemble des tâches du traitement automatique des langues constitue la preuve de l'apport majeur des approches probabilistes. De nos jours, les recherches en TAL sont concentrées sur les méthodes statistiques d'ingénierie. De même, les principales revues scientifiques du domaine accordent plus d'importance à ce type de travaux que les travaux à base de règles.

Pour développer les outils de traitement informatique du langage, les chercheurs et les industriels ont besoin de recueillir de grandes quantités de textes écrits et oraux numérisés. Dans cette section, nous présenterons les propriétés des corpus et les types d'annotations existants.

2.2.1. Propriétés des corpus

Un corpus est la collecte de divers matériaux, rassemblés selon un ensemble de critères afin qu'il soit représentatif et balancé. Parmi les paramètres qu'il faut prendre en considération pour qu'un corpus soit balancé, citons, à titre d'exemple, le genre, le domaine, la longueur, le temps, l'époque qu'il représente et le registre du langage. Les propriétés du corpus varient également selon les applications pour lesquelles il sera utilisé. Si le corpus est assemblé pour un objectif précis, ce n'est pas nécessaire d'avoir des exemples de divers genres. Le plus

important dans le domaine de collecte des corpus est d'inclure les spécificités de la langue que le chercheur essaie de maitriser.

Les matériaux composant un corpus peuvent être de genres variés : textes, audio, vidéo, etc. Un corpus peut contenir des textes d'une même langue (monolingue), ou bien de plusieurs langues (multilingue). Il peut être constitué de corpus parallèles, ensemble de couples de textes dont l'un est la traduction de l'autre, ou de corpus comparables, composés de textes dans des langues différentes mais partageant une partie du vocabulaire employé et traitant d'un même sujet, à la même époque et dans un même registre. Les corpus multilingues sont utilisés dans les applications de traduction automatique.

Pour illustrer notre propos, rappelons qu'un des premiers corpus représentatifs de l'anglais américain écrit est le Brown Corpus (Kurčera & Francis, 1967). Il contient environ un million de mots de textes variés. Il a été assemblé en 1961. Un corpus de taille similaire a été réalisé pour l'anglais britannique, nommé le Lancaster-Oslo-Bergen (LOB) Corpus, en 1970 (Johanson et *al.*, 1978). Le Penn Treebank est un corpus commercialisé par le Consortium des Données Linguistiques (LDC[26]), annoté grammaticalement et en partie syntaxiquement. Il contient environ 4,5 millions de mots (Marcus et *al.*, 1993). D'autres corpus plus larges existent : à titre d'exemple le British National Corpus (BNC[27]), qui contient 100 millions de mots. En 2002 est établi, un très large corpus, nommé Bank of English, contenant environ 450 millions de mots collectés à partir de sources écrites et orales. L'Open American National Corpus (OANC[28]), est un corpus entièrement libre et dont l'objectif est d'atteindre 100 millions de mots annotés. Sa deuxième version contient 22 millions de mots annotés.

Pour ce qui est de la langue arabe, le premier corpus annoté est celui qui est réalisé par Khoja et ses coéquipiers (Khoja et *al.*, 2001). Il contient 50 000 mots, annotés avec les étiquettes suivantes: nom défini ou indéfini, verbe, particule, point de ponctuation ou numéro. Dans ce corpus, seuls 1 700 mots ont été annotés avec un jeu d'étiquettes détaillé utilisant le genre, le nombre et d'autres informations morphosyntaxiques.

Parmi les corpus de la langue arabe annotés avec des données morphologiques et syntaxiques, on trouve : le Penn Arabic Treebank (Maamouri et *al.*, 2004) et le Prague Arabic Dependency Treebank (Smrž and Hajič, 2006). Une liste des corpus arabes et dans différents dialectes a été publié dans l'appendix C du livre de Nizar Habash (Habash, 2010).

[26] https://www.ldc.upenn.edu/
[27] http://www.natcorp.ox.ac.uk/
[28] http://www.anc.org/

Le premier corpus connu pour la langue française, nommé le Trésor de la Langue Française, a été réalisé dans les années soixante. Il renferme une grande base de textes bruts. Parmi les corpus français écrits et annotés que nous avons recensés, citons entre autres PAROLE[29], MULTEXT JOC[30] (Véronis & Khouri, 1995), le French Treebank (Abeillé et *al.*, 2003), la base FREEBANK (Salmon-Alt et *al.*, 2004), le corpus Sequoia[31] (Candito & Seddah, 2012).

Vu l'importance de l'utilisation des corpus et leur rôle prépondérant dans la création d'outils informatiques, des institutions spécialisées dans la création de ressources langagières et outils de base pour le TAL ont été créés. Les constructeurs les plus importants sont:

- Le Linguistic Data Consortium (LDC) ;
- European Language Resources Association (ELRA) ;
- International Computer Archive of Modern English (ICAME) ;
- Oxford Text Archive (OTA) ;
- Child Language Data Exchange System (CHELDS).

2.2.2. Types d'annotations

Comme il a été mentionné dans la sous section précédente, un corpus collecté peut être brut ou annoté. Le corpus peut être enrichi par des données multiples : phonétiques, grammaticales, syntaxiques, sémantiques, etc. La figure 2.1 suivante donne un panorama des différents types d'annotations.

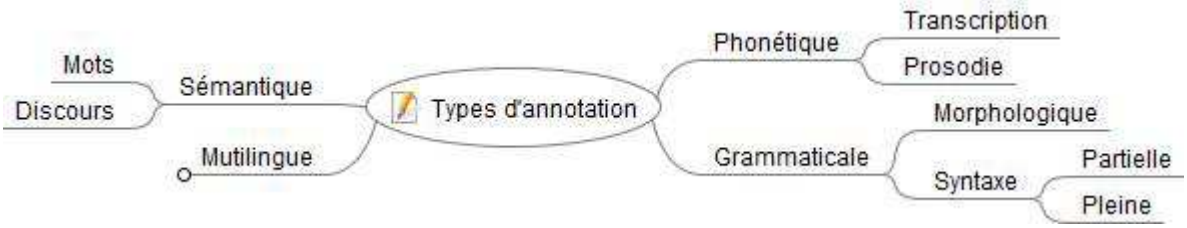

Figure 2.1. Vue synoptique des différents types d'annotation

L'annotation phonétique

Elle consiste en deux types : la transcription de l'oral à l'écrit et l'annotation des morphèmes prosodiques. La transcription consiste en la production d'une représentation orthographique ou d'une représentation phonétique. Dans les deux représentations, des indicateurs tels que les pauses, les hésitations, les débuts des mots inachevés, etc. sont ajoutés aux informations de

[29] http://catalog.elra.info/product_info.php?products_id=565
[30] http://catalog.elra.info/product_info.php?products_id=5341
[31] https://www.rocq.inria.fr/alpage-wiki/tiki-index.php?page=CorpusSequoia

l'étiquetage. L'annotation phonétique est souvent assortie d'un alignement temporel sur le signal de parole. Cet étiquetage est très coûteux en temps. En effet, pour annoter une minute de parole, on a besoin de dix à quinze heures de travail manuel. Les années quatre-vingt dix ont été marquées par l'apparition de logiciels permettant la transcription automatique tels que Via Voice, Dragon Dictate, etc. Néanmoins, les performances ne sont pas satisfaisantes. En ce qui concerne l'alignement de la transcription avec l'enregistrement, plusieurs outils existent, tels que Mbrolign (Malfrère & Dutoit, 1997). Des ajustements manuels sont parfois nécessaires pour un alignement de bonne qualité. Ainsi, l'effort global pour l'étiquetage des signaux de parole est considérablement réduit.

L'annotation grammaticale

Dans ce type d'annotation, on distingue deux grandes familles : l'étiquetage des catégories grammaticales et informations morphosyntaxiques pour la première et l'étiquetage des structures syntaxiques, complètes ou partielles pour la deuxième. Le premier a commencé avec la construction du Brown Corpus dans les années soixante. Il consiste en la distinction de la classe grammaticale et des informations morphosyntaxiques associées aux mots dans le contexte d'énonciation. La nature desdites informations peut être extrêmement variée. Elles représentent toute information permettant de caractériser le comportement d'un mot dans son contexte. En particulier, sont exclues des descriptions morphosyntaxiques les informations de nature coréférentielle, telle que l'indication explicite de l'antécédent dans la description d'un pronom. L'annotation syntaxique consiste en l'annotation des constituants de la phrase ainsi que les relations entre eux formant ainsi un arbre syntaxique. Typiquement, l'annotation partielle est restreinte à l'annotation des parties des textes tels que les groupes nominaux, les groupes verbaux et les groupes prépositionnels.

Les ressources arborées sont très utiles dans le TAL, les recherches linguistiques et l'enseignement des langues étrangères. Cependant, dans l'état actuel de la recherche dans ce domaine, il n'existe pas de système automatique permettant de faire une analyse syntaxique avec un taux de succès acceptable. C'est pourquoi l'analyse partielle est souvent utilisée. Ce dernier type d'analyseur, et bien qu'il doit être suivi d'une phase de correction manuelle, a permis la construction de corpus arborés importants tels que le Penn TreeBank (Marcus et *al.*, 1993).

L'annotation sémantique

Deux grandes familles de ce type d'annotation existent : l'annotation des mots et l'annotation des relations intervenant dans la phrase ou le discours. L'annotation sémantique des mots

consiste en l'annotation des sens de mots qui nécessitent une résolution de la polysémie en contexte. C'est une tâche importante dans de nombreuses applications du TAL, telles que la traduction et la recherche d'information. Parmi les difficultés de cette tâche, on trouve la détermination de la liste des sens pour chaque mot. Des efforts ont été faits dans plusieurs langues pour le développement de ressources annotées sémantiquement, tels que les WordNets (Fellbaum, 1998 ; Black et *al.*, 2006 ; Sagot & Fišer, 2008). Concernant l'annotation des relations qui interviennent dans une phrase ou bien un discours, la plupart des travaux s'orientent vers le marquage des phénomènes référentiels, tels que les anaphores. Le programme MUC (Message Understanding Conferences) a eu une influence déterminante en suscitant des recherches sur les méthodes d'évaluation des systèmes de résolution d'anaphores.

L'annotation multilingue

Elle consiste à faire accompagner les textes de leurs traductions dans une ou plusieurs langues ; cet accompagnement est appelé aussi alignement multilingue ou corpus parallèle. Cet alignement met en relation des unités logiques se correspondant dans deux ou plusieurs textes. Ces unités logiques peuvent être de diverses sortes : paragraphes, segments de document, phrases, syntagmes, mots ou morphèmes. Les premiers articles marquants sont relativement récents : c'est le cas à titre d'exemples de (Simard et *al.*, 1992 ; Church, 1993). Ce type d'annotation est très utile dans la réalisation des outils statistiques de la traduction automatique. Ces outils, comme pour toutes les méthodes statistiques basées sur les corpus, sont d'autant plus efficaces que la qualité des corpus alignés est bonne.

Le corpus de l'Hansard du parlement canadien, corpus anglais-français, est un exemple connu et l'un des premiers corpus parallèles à être numérisés et mis à la disposition des chercheurs en linguistique. Pour l'arabe, plusieurs corpus parallèles ont été réalisés par le LDC, tels que le corpus arabe des diffusions des nouvelles[32] et le blog arabe des textes parallèles[33].

D'autres formes d'annotation existent : l'annotation automatique des images, qui consiste à assigner une légende ou des mots clés à une image numérique ; l'annotation de corpus pour l'analyse des sentiments, des opinions des consommateurs, etc.

Afin de faciliter l'annotation, on utilise des outils d'aide à l'annotation qui sont de deux types. Les premiers sont les outils de manipulation des données tels que les concordanciers, les

[32] http://catalog.ldc.upenn.edu/LDC2007T24
[33] http://catalog.ldc.upenn.edu/LDC2008T02

outils de visualisation du signal de parole et les systèmes de saisie et de vérifications. Ces outils sont indispensables au processus d'annotation et permettent un niveau de qualité appréciable. Plusieurs équipes de recherche se contentent de systèmes de traitement de texte tel que *Emacs* pour linux/Unix, *Microsoft Word* pour Windows, *BBEdit* pour Macintosh, pour la transcription de l'oral ou la saisie d'étiquettes pour l'annotation. Les seconds sont les outils d'annotation automatique, qui permettent un gain de temps au niveau de l'annotation, mais nécessitent plus de temps pour le prétraitement et la correction, vu la complexité des phénomènes linguistiques traités. Quoi qu'il en soit, les outils d'annotation automatique sont très utiles lors des traitements de grandes masses de données.

2.3. Construction d'un corpus amazighe annoté morphosyntaxiquement

Le jeu d'étiquettes morphosyntaxiques représente le formalisme décrivant les comportements des occurrences des mots en contexte. Plus on a d'étiquettes et plus on cherche la précision, plus la probabilité d'erreur de l'étiquetage augmente.

Un effort de normalisation des jeux d'étiquettes a été fait dans le projet EAGLES[34], sur des bases essentiellement linguistiques pour des langues européennes. Cet effort a été testé en grandeur réelle dans le cadre du projet MULTEXT[35] et a abouti à 74 étiquettes. Le nombre d'étiquettes utilisées varie selon les langues. Certains projets présentent un nombre d'étiquettes plus élevé tels que le projet GRACE[36] utilisant 312 étiquettes.

La nature des informations contenues dans la description morphosyntaxique peut être extrêmement variée. Sur la base des caractéristiques de la langue amazighe présentées ci-dessus, le jeu d'étiquettes défini pour réaliser l'étiquetage morphosyntaxique est conçu sur la base de 13 parties du discours, avec deux attributs communs à chacune d'elles, le mot lui-même et son lemme, dont la valeur dépend de l'élément lexical qu'il accompagne.

Chacun des 13 éléments définis dans le Tableau 2.1 est défini par des attributs et, le cas échéant, par des sous attributs.

[34] Eagles: http://www.ilc.cnr.it/EAGLES96/home.html

[35] Multext: http://nl.ijs.si/ME/

[36] Grace: http://www.itl.nist.gov/iad/mig/publications/proceedings/darpa99/html/rel240/rel240.htm

Tableau 2.1. Vue d'ensemble du jeu d'étiquettes avec leurs attributs

Parties du discours	**attributs et sous attributs avec le nombre des valeurs**
Nom	genre(3), nombre(3), état(2), dérivation(2), POS sous classification(4), nombre du possesseur(3), genre du possesseur(3), personne(3)
Adjectif/ nom de qualité	genre(3), nombre(3), état(2), dérivation(2), POS sous classification(3)
Verbe	genre(3), nombre(3), personne(3), aspect(3), négation(2), forme(2), dérivation(2), voix(2)
Pronom	genre(3), nombre(3), personne(3), POS sous classification(7), déictique(3)
Déterminant	genre(3), nombre (3), POS sous classification(11), déictique(3)
Adverbe	POS sous classification(6)
Préposition	genre(3), nombre(3), personne(3), nombre du possesseur(3), genre du possesseur(3)
Conjonction	POS sous classification(2)
Interjection	
Particule	POS sous classification(7)
Focalisateur	
Résiduel	POS sous classification(5), genre(3), nombre(3)
Ponctuation	type de la marque de ponctuation(16)

Par exemple, le nom a les attributs suivants:

- Le genre a trois valeurs : féminin, masculin et commun ;
- Le nombre a trois valeurs : singulier, pluriel et commun ;
- L'état a deux valeurs : libre ou construit ;
- La dérivation a deux valeurs : oui ou non ;
- La POS sous classification a quatre valeurs : commun, de parenté, numéral et propre.

Si le nom est un nom de parenté, les sous attributs sont :

- Le nombre du possesseur, qui a trois valeurs : singulier, pluriel et commun ;
- Le genre du possesseur, qui a trois valeurs : féminin, masculin et commun ;
- La personne du possesseur, qui a trois valeurs : première personne, deuxième personne et troisième personne.

Le nombre théorique des étiquettes possibles est supérieur à 1 900 étiquettes. Les attributs des autres catégories sont présentés dans (Outahajala et *al.*, 2011c).

Par ailleurs, les attributs et sous attributs de ce jeu d'étiquettes pour la langue amazighe ont été implémentés dans AncoraPipe[37]. Le détail de ces informations morphosyntaxiques[38] est présenté dans l'annexe 1 de ce mémoire.

2.4. Processus d'annotation

L'annotation manuelle a été réalisée suivant un processus en 5 étapes comme on peut le voir dans la figure 2.2 suivante:

1- Sélection des textes bruts : afin de constituer le corpus brut, nous avons ciblé tous les textes existants, écrits en amazighe et respectant les règles d'orthographe adoptées officiellement au Maroc. Pour ceci, nous nous sommes contenté des textes révisés par le Centre de l'Aménagement Linguistique(CAL) de l'IRCAM. Ainsi, nous avons pris les textes choisis des manuels scolaires existants, des textes variés extraits du bulletin d'information *inghmisn n usinag*, des extraits du contenu de certains romans, et tous les textes du site Web de l'IRCAM[39] ;

2- Phase de translittération: les textes amazighes produits jusqu'à aujourd'hui sont écrits selon différents systèmes d'écriture. La plupart de ces textes sont écrits avec les polices utilisant des glyphes tifinaghes et des caractères latins, ainsi que tifinaghe Unicode[40]. Dans ce sens, il est à noter que les textes utilisant tifinaghe Unicode sont de plus en plus nombreux (le tableau 2.2 présente les correspondances entre les systèmes d'écritures existants et le système d'écriture choisi) ;

3- Annotation manuelle: ce corpus est annoté morphologiquement en utilisant l'outil d'annotation AncoraPipe, selon le jeu d'étiquettes morphosyntaxique défini pour annoter les corpus amazighes (Outahajala et *al.*, 2010 ; Outahajala et *al.*, 2011c) ;

---

[37] http://clic.ub.edu/ancora/

[38] Les attributs et sous attributs, conçus pour l'annotation morphosyntaxique de la langue amazighe ainsi que d'autres métadonnées jugées utiles, ont été rassemblés dans un fichier XML et sont disponibles sur www.outamed.com.

[39] Version amazighe du site web : www.ircam.ma, consulté en fin 2011.

[40] D'autres systèmes d'écriture pour transcrire le tifinaghe existent. Néanmoins, ils n'ont pas été présentés car les textes qui les utilisent nécessitent des prétraitements divers, dont le plus coûteux est la resegmentation selon les règles orthographiques adoptées et le travail de révision et de validation.

4- Phase de révision: nous avons utilisé XSLT pour générer les fichiers de sortie, permettant une révision plus facile des fichiers annotés, sous XML. La vitesse d'annotation varie entre 80 et 120 mots par heure. Des textes choisis aléatoirement ont été révisés par des linguistes. Les remarques communes de ces derniers ont été généralisées à l'ensemble du corpus annoté lors de la deuxième révision par un annotateur différent. L'accord entre annotateurs est de 94.98% ;

5- Textes annotés : les documents annotés sont sous le format XML. Afin d'analyser, transformer et gérer ce format plusieurs outils existent.

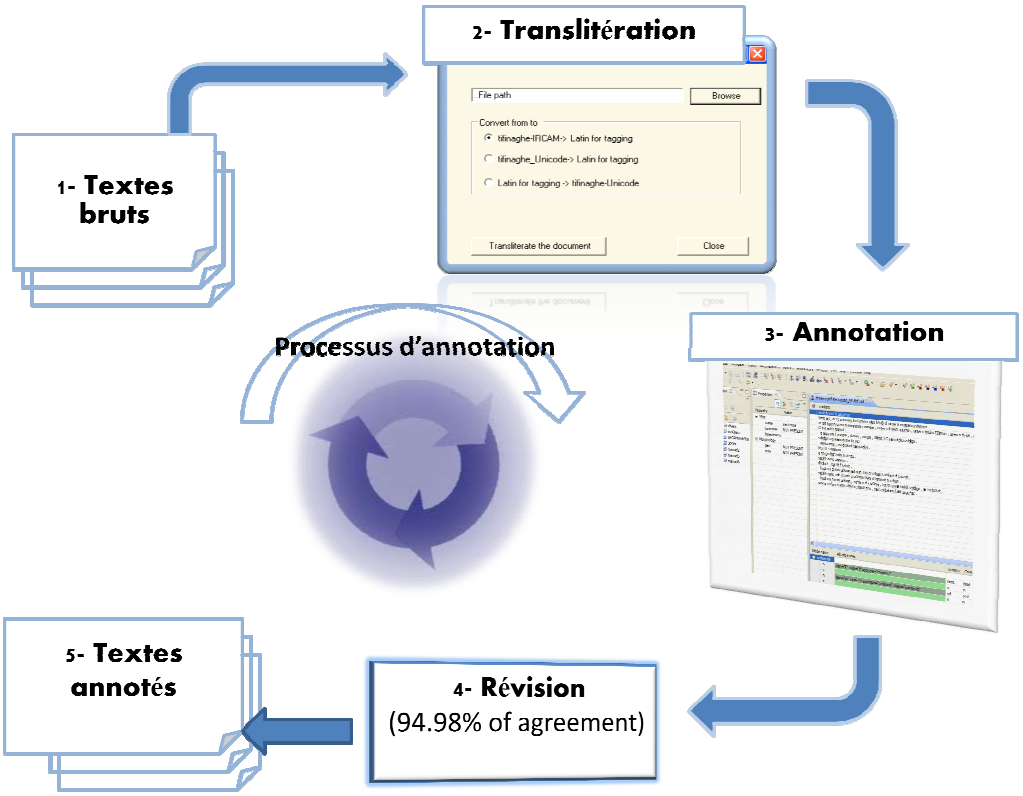

Figure 2.2. Processus d'annotation

2.5. Encodage du corpus

2.5.1. Les systèmes d'écriture

Les textes amazighes produits jusqu'ici ont été transcrits sur la base de systèmes d'écriture différents ; la plupart d'entre eux utilisent les caractères Unicode et les caractères latins.

Il est important de signaler que les textes écrits en Tifinaghe Unicode sont de plus en plus nombreux par rapport aux textes utilisant les autres systèmes d'écriture. Pour des raisons

techniques, le système d'écriture spécifique est choisi sur la base des caractères ASCII uniquement (Outahajala et *al.*, 2010).

Tableau 2.2. Correspondances entre les systèmes d'écritures existants et le système d'écriture choisi

Tifinaghe Unicode		Translitération		Caractères utilisés dans Tifinaghe IRCAM		Système d'écriture choisi
Code	Caractère	Latin	Arabe	Caractère	Codes	
U+2D30	ⴰ	a	ا	A, a	65, 97	a
U+2D31	ⴱ	b	ب	B, b	66, 98	b
U+2D33	ⴳ	g	گ	G, g	71, 103	g
U+2D33&U+2D6F	ⴳ ⵯ	g^w	گ + ◻ [41]	Å, å	197, 229	gw
U+2D37	ⴷ	d	د	D, d	68, 100	d
U+2D39	ⴹ	ḍ	ض	Ä, ä	196, 228	D
U+2D3B	ⴻ	e	يي	E, e	69, 101	e
U+2D3C	ⴼ	f	ف	F, f	70, 102	f
U+2D3D	ⴽ	k	ک	K, k	75, 107	k
U+2D3D&+2D6F	ⴽ ⵯ	k^w	گ + ◻	Æ, æ	198, 230	kw
U+2D40	ⵀ	h	ھ	H, h	72,104	h
U+2D43	ⵃ	ḥ	ح	P, p	80,112	H
U+2D44	ⵄ	E	ع	O, o	79, 111	E
U+2D45	ⵅ	x	خ	X, x	88, 120	x
U+2D47	ⵇ	q	ق	Q, q	81, 113	q
U+2D49	ⵉ	i	ي	I, i	73, 105	i
U+2D4A	ⵊ	j	ج	J, j	74, 106	j
U+2D4D	ⵍ	l	ل	L, l	76, 108	l
U+2D4E	ⵎ	m	م	M, m	77, 109	m
U+2D4F	ⵏ	n	ن	N, n	78, 110	n
U+2D53	ⵓ	u	و	W, w	87, 119	u
U+2D54	ⵔ	r	ر	R, r	82, 114	r
U+2D55	ⵕ	ṛ	◻	Ë, ë	203, 235	R
U+2D56	ⵖ	ɣ	غ	V, v	86, 118	G
U+2D59	ⵙ	s	س	S, s	83, 115	s
U+2D5A	ⵚ	ṣ	ص	Ã, ã	195, 227	S
U+2D5B	ⵛ	c	ش	C, c	67, 99	c
U+2D5C	ⵜ	t	ت	T, t	84, 116	t
U+2D5F	ⵟ	ṭ	ط	Ï, ï	207, 239	T
U+2D61	ⵡ	w	ۋ	W, w	87, 119	W
U+2D62	ⵢ	◻+ ي	ي	Y, y	89, 121	Y
U+2D63	ⵣ	z	ز	Z, z	90, 122	z
U+2D65	ⵥ	ẓ	ژ	Ç, ç	199, 231	Z
U+2D6F	ⵯ	^{w}	◻	Aucun correspondant		w

[41] Il représente le caractère arabe DAMMA reversé, dont le code est 0657.

Un outil de translittération a été conçu (voir la figure 2.3 ci-dessous) afin de générer une translittération vers le système d'écriture cible et de corriger certains éléments tels que le caractère "^" qui existe dans de nombreux textes en raison d'erreurs d'entrée lors de l'introduction des lettres Tifinaghe. Par exemple, la partie de la phrase «⵰ ⵙⵙ ⵏ ⵜⵎⵖⵔⴰ ⵓ» saisie en Tifinaghe Unicode, ou la portion de la phrase : "ass n tm^vra" utilisant les premières polices de tifinaghe, utilisant des glyphes correspondant aux lettres tifinaghes mais employant les caractères latins, sera transcrite: «ass n tmGra» [le jour du mariage]. La présence des caractères "^" dans certains textes est due aux erreurs de saisie des labiovélaires : «ⵅ ̆ » et «ⵄ ̆ »

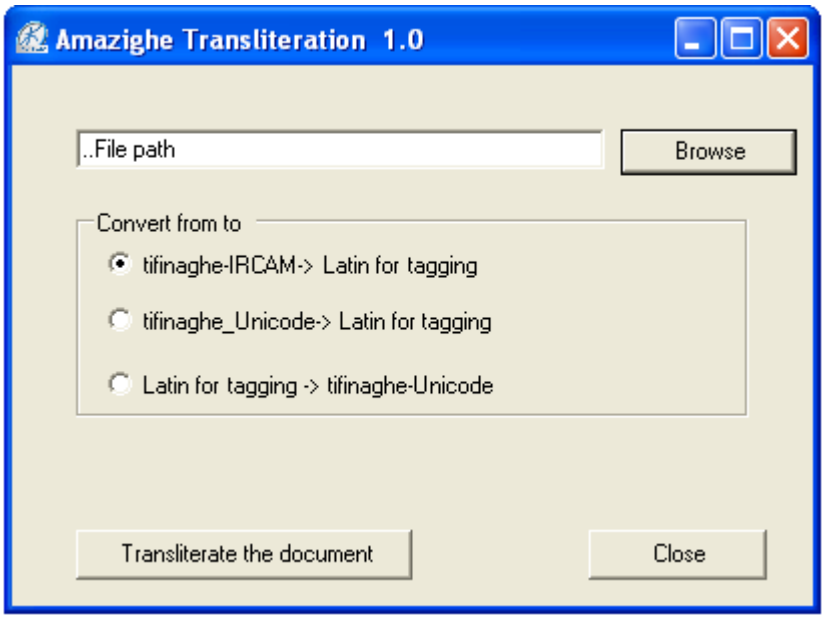

Figure 2.3. Outil de translittération de et vers le système d'écriture choisi.

2.5.2. Description du corpus

Pour constituer notre corpus, nous avons choisi une liste de textes extraits de diverses sources, respectant les règles orthographiques adoptées pour l'amazighe marocain telles que la version amazighe du site Web de l'IRCAM[42], le périodique "Inghmisn n usinag" (newsletter de l'IRCAM) et les trois manuels scolaires du primaire[43] 2, 5 et 6. Le tableau 2.3 suivant présente une description des sources choisies.

Tableau 2.3. Description du corpus.

[42] www.ircam.ma site web de l'Institut Royal de la Culture Amazighe.

[43] Téléchargeable gratuitement à partir de http://www.ircam.ma/amz/index.php?soc=bulle.

Corpus	Nombre de jetons	Nombre de phrases
Manuel 2 (*tifawin a tamazight*)	5079	372
Manuel 5 (*tifawin a tamazight*)	2319	179
Manuel 6 (*tifawin a tamazight*)	3773	253
Site web de l'IRCAM	4258	185
inghmisn n usinag (newsletter de l'IRCAM)	4636	415
Textes Divers	602	34
Total	20667	1438

Après la translittération vers le système d'écriture choisi, ces textes ainsi que leurs spécifications morphologiques sont encodés en utilisant XML. Chaque mot est marqué avec ses attributs et ses sous attributs.

2.6. Outil d'annotation

Nous décrirons plus loin dans cette section l'outil d'annotation et la méthode d'annotation utilisés, comme nous évoquerons quelques fonctionnalités pratiques de l'outil en question.

2.6.1. L'environnement de développement Eclipse

L'environnement de développement Eclipse est un environnement intégré (Integrated Development Environment, IDE désormais), qui permet de créer des projets de développement mettant en œuvre plusieurs langages de programmation, tels que Java, C#, C, C++, Cobol Python, Perl et bien d'autres. Eclipse IDE est principalement écrit en Java à l'aide de la bibliothèque graphique Standard Widget Toolkit (SWT). La spécificité de l'IDE Eclipse vient du fait de son architecture développée autour de la notion de plugin[44]. La base de la plateforme Eclipse est composée de:

- La platforme Runtime, qui démarre la plateforme et gère les plugins ;
- SWT, la bibliothèque graphique de base de l'IDE ;
- JFace, une bibliothèque graphique basée sur SWT ;
- Eclipse Workbench, la dernière couche graphique permettant de manipuler des composants, tels que des vues, des éditeurs et des perspectives.

[44] Nommé également module d'extension ou module externe. C'est un paquet qui complète un logiciel de base, en lui apportant de nouvelles fonctionnalités.

Plusieurs logiciels libres et commerciaux sont basés sur ce logiciel libre. Citons, à titre d'exemple :

Pour les logiciels libres:

- Web Tools Platform project[45] (WTP), qui propose de nombreux outils pour le développement d'applications web en Java ;
- C/C++ Development Tools Project[46] (CDT), environnement complet de développement pour les langages C et C++ pour Eclipse;
- Business Intelligence and Reporting Tools Project (BIRT)[47], environnement de reporting qui interagit avec Java/ Java 2 Enterprise Edition(J2EE) afin de produire des rapports;

Pour les logiciels commerciaux:

- IBM Lotus Notes[48], logiciel de travail collaboratif, utilisé dans des entreprises ou des administrations pour gérer les projets, les courriels et les échanges d'informations autour d'une base commune;
- IBM Symphony[49], suite bureautique comprenant des applications servant à créer, éditer et partager des documents bureautiques, notamment de traitement de texte et des feuilles de calcul;
- WebSphere Studio Application Developer[50], environnement de développement avancé pour créer, tester et déployer l'applications J2EE;

L'outil d'annotation présenté dans la sous section suivante se base sur la version 3.5, appelée Galileo[51] de cet IDE, parue en juin 2009.

2.6.2. Description de l'outil AncoraPipe

L'outil AncoraPipe est un plugin d'Eclipse. Tous les éléments inclus dans Eclipse sont disponibles pour l'annotation de corpus et de développement. Cet outil permet l'annotation de corpus selon différents niveaux linguistiques (Bertran et *al.*, 2008), car il utilise le même

[45] http://www.eclipse.org/webtools/

[46] http://www.eclipse.org/cdt/

[47] http://www.eclipse.org/birt/phoenix/

[48] http://www.ibm.com/developerworks/lotus/library/notes8-new/

[49] http://www-03.ibm.com/software/lotus/symphony/developers.nsf/home

[50] http://www-01.ibm.com/software/integration/wsadie/

[51] http://pro.01net.com/editorial/503837/la-galaxie-eclipse-3-5-galileo-est-accessible-en-telechargement/

format pour les différentes étapes. AncoraPipe a été utilisé dans l'annotation de deux (2) corpus de 500.000 mots chacun ; le premier est en catalan (Ancora-CAT) et le deuxième est en espagnol (Civit & Martí 2004). L'interface de l'outil d'annotation est organisée en différents groupes, dans lesquels les données sont présentées et les boutons et les menus sont disponibles, pour effectuer des opérations sur le corpus, tels que le regroupement et le partage. Pour effectuer l'annotation, de nombreux panneaux sont utilisés: volet d'arborescence corpus répertoire, qui permet à l'utilisateur de sélectionner un fichier ; le panneau des listes phrases montre les phrases d'un fichier ; le panneau arbre permet à l'utilisateur de voir les données du niveau d'annotation avec des lemmes et des mots ; le panneau d'annotation permet d'effectuer les opérations d'annotation sur l'arbre et d'annoter ses nœuds.

L'interface est entièrement personnalisable afin de permettre différents jeux d'étiquettes définis par l'utilisateur. Les exigences pour AnCoraPipe sont les suivantes: Java 1.5 et SWT.

La phase d’annotation commence par la segmentation du texte en entrée selon les espaces typographiques et les points de ponctuation. Par la suite, le fichier est ajouté au corpus segmenté selon le format XML. Pour cet effet, nous utilisons l’interface de la figure 2.4 suivante :

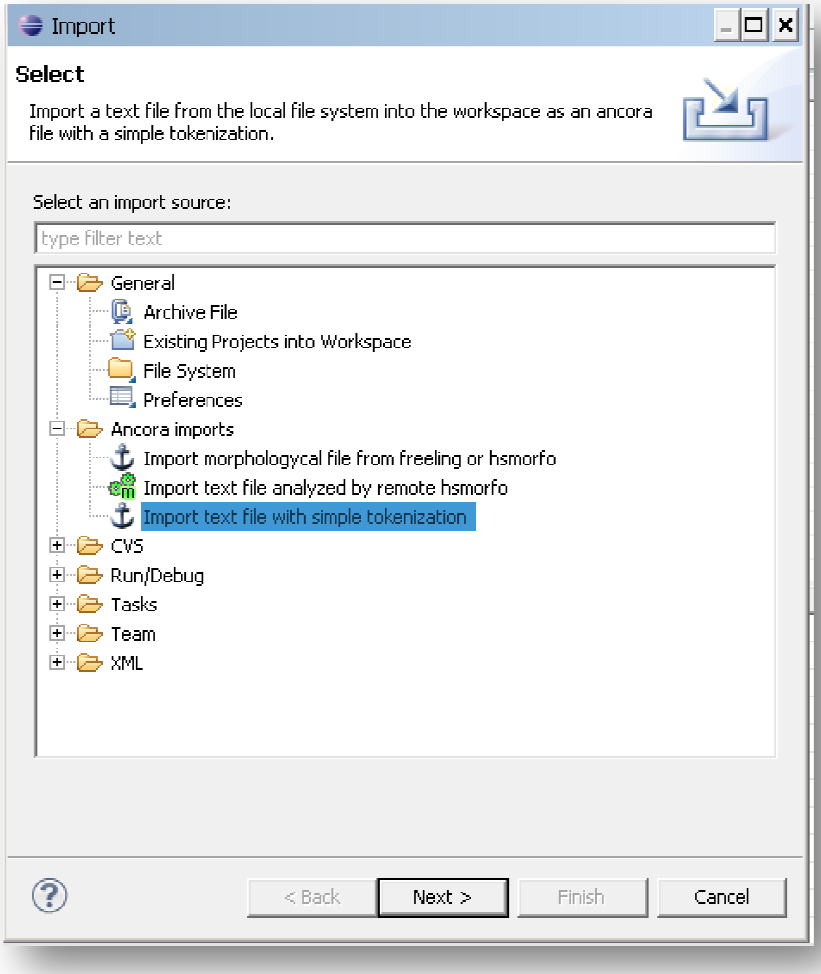

Figure 2.4. Interface d'import et de segmentation du texte

Les informations morphosyntaxiques sont choisies par la suite selon la catégorie grammaticale de base (voir figure 2.5). Le détail des attributs et des sous attributs est présenté dans l'annexe 1 de ce mémoire. L'élément « word » est attribué aux mots dont la catégorie grammaticale est inconnue.

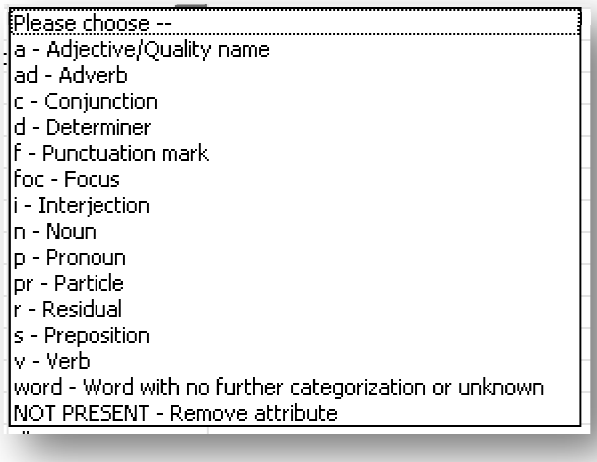

Figure 2.5. Étiquettes de base utilisées pour l'amazighe

En choisissant la catégorie verbe pour le mot « idda » (aller au passé) dans la phrase «llig idda idir ad isrs…», on peut affecter les attributs : aspect, dérivation, forme, genre, nombre, voix, personne et lemme, comme il est montré dans la figure 2.6 suivante :

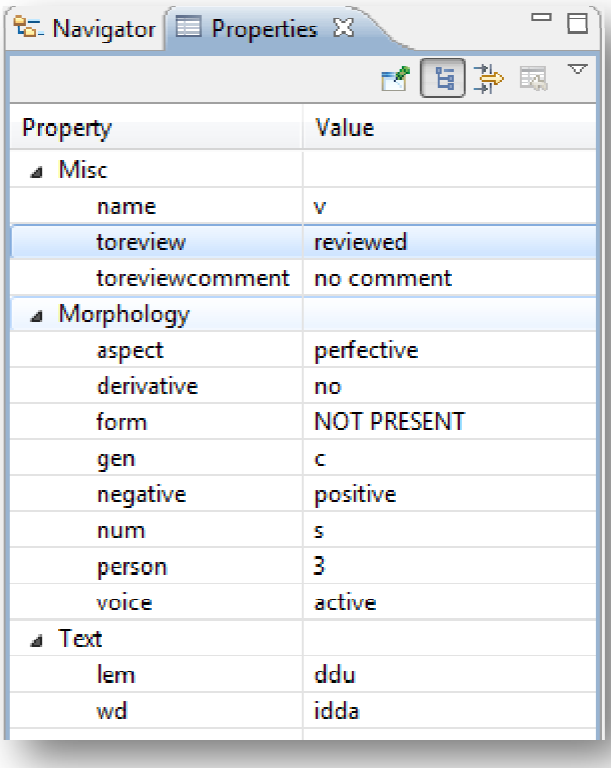

Figure 2.6. Annotation grammaticale du verbe *idda*

A l’aide de cet outil d’annotation, pour chaque mot du corpus, on peut attribuer un commentaire de révision et mentionner que la révision a été faite ou non. Ces deux étiquettes sont très utiles lors du traitement des mots difficiles ou ceux qui font l’objet d’un désaccord et qui nécessitent une discussion entre linguistes.

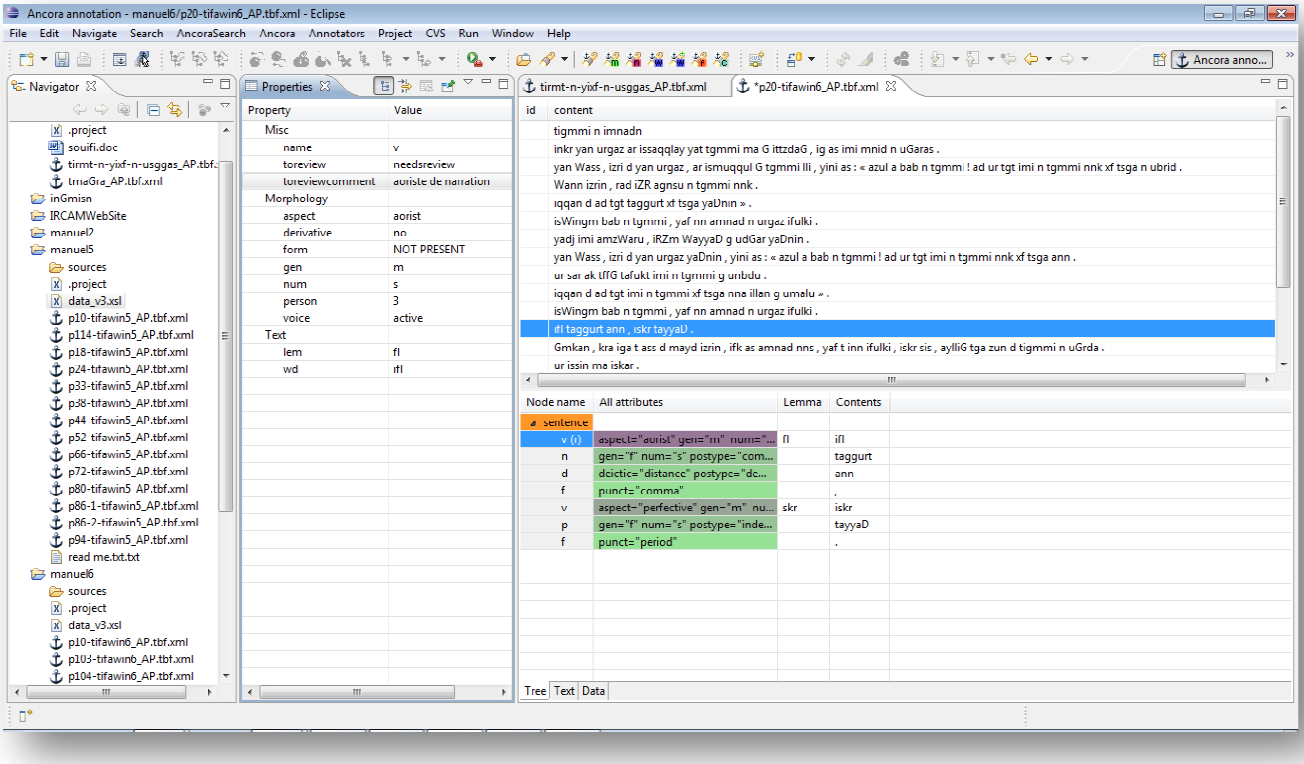

Figure 2.7. Interface principale d’annotation morphosyntaxique

Parmi les fonctionnalités utiles que permet AncoraPipe, il y a la possibilité de segmenter un mot en deux ou plusieurs mots, et à l’inverse, la possibilité de fusionner deux ou plusieurs mots en un seul. Il offre également la possibilité de chercher un mot donné ou une expression régulière dans le corpus entier ou en partie (Voir les figures 2.8 et 2.9). Cette dernière fonctionnalité permet de trouver toutes les occurrences d’un mot ou d’une expression régulière dans le corpus et, par conséquent, la possibilité de consulter l’annotation attribuée à un mot quelconque dans l’ensemble du corpus.

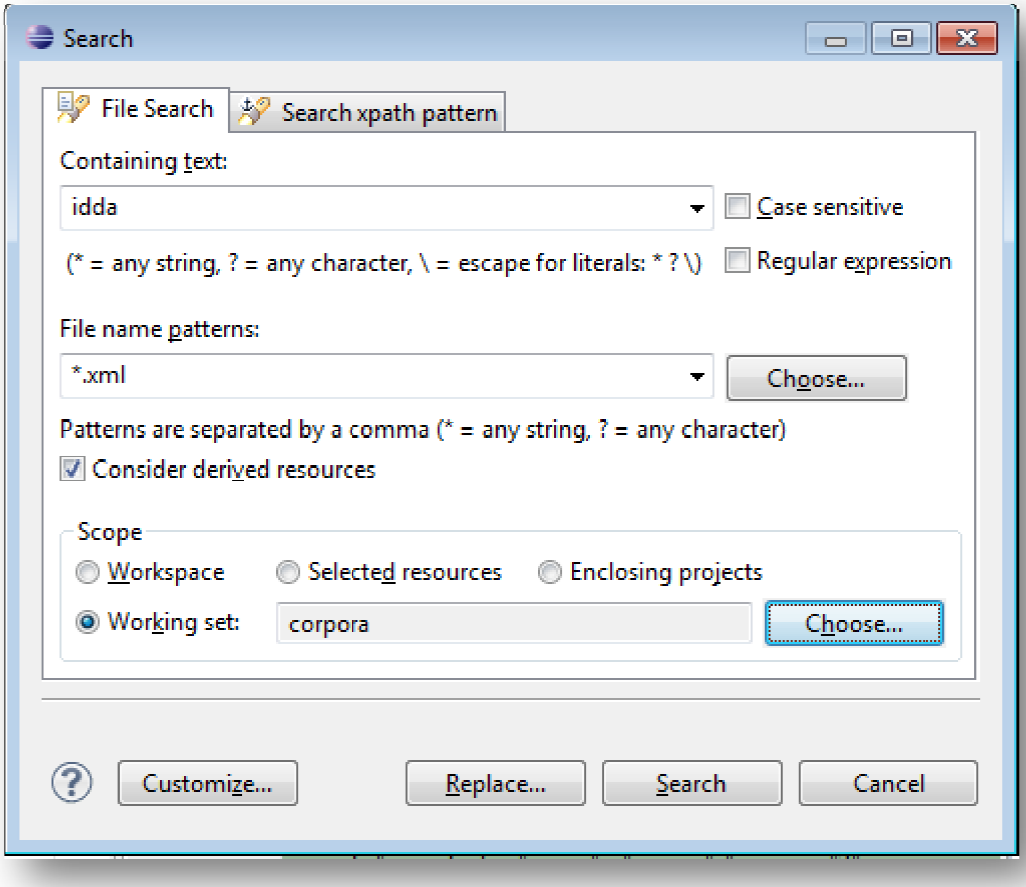

Figure 2.8. Interface de recherche d'AncoraPipe

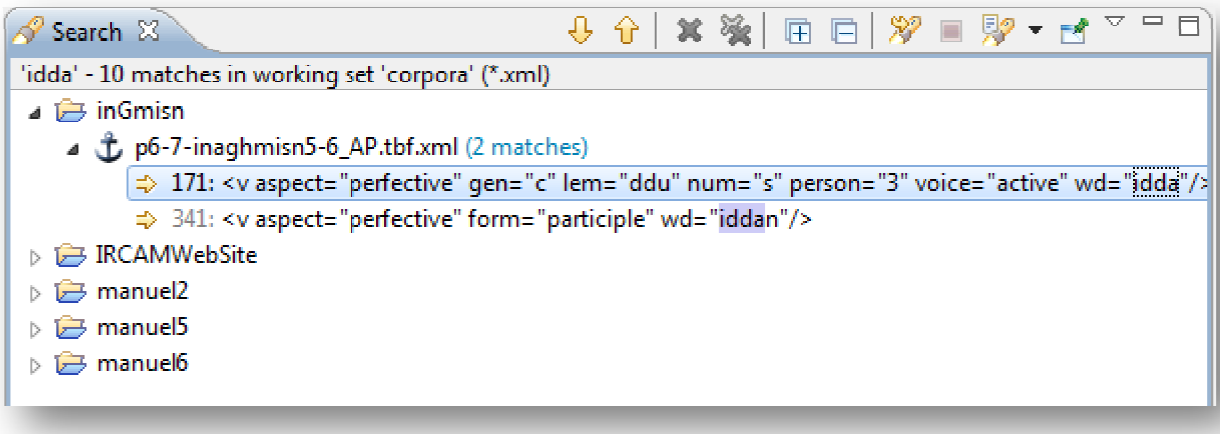

Figure 2.9. Résultat d'une recherche dans AncoraPipe

A chaque texte du corpus, des étiquettes du genre : auteur, langue, source, et un commentaire sont affectés pour l'identifier.

Les documents d'entrée ont un format XML, permettant de représenter les données sous forme de structures arborescentes. La figure 2.10 présente un exemple d'annotation morphosyntaxique d'une phrase extraite d'un texte sur une cérémonie de mariage : "ass n tmGra, illa ma issnwan, illa ma yakkan i inbgiwn ad ssirdn". [Traduction en Français : "le jour du mariage, il y a ceux qui cuisinent, il y a ceux qui donnent aux invités à se laver les mains"]

```
<sentence>
    <n gen="m" lem="ass" num="s" wd="ass"/>
    <s wd="n"/>
    <n gen="f" lem="tamGra" num="s" state="construct" wd="tmGra"/>
    <f punct="comma" wd=","/>
    <v aspect="perfective" gen="m" lem="ili" num="s" person="3" wd="illa"/>
    <p postype= »relative » wd= »ma (ceux) »/>
    <v aspect="imperfective" gen="m" lem="ssnw" num="s" person="3" form="participle"
    wd="issnWan"/>
    <f punct="comma" wd=","/>
    <v aspect=" perfective" gen="m" lem="ili" num="s" person="3" wd="illa"/>
    <p postype= »relative » wd= »ma »/>
    <v aspect="imperfective" form="participle" gen="m" lem="fk" num="s" person="3"
    wd="yakkan"/>
    < s wd="i"/>
    <n gen="m" lem="anbgi" num="p" state="construct" wd="inbgiWn"/>
    <pr postype="aspect" wd="ad"/>
    <v aspect="aorist" gen="m" lem="ssird" num="p" person="3" wd="ssirdn"/>
</sentence>
```

Figure 2.10. Exemple de texte annoté utilisant le jeu d'étiquettes défini pour l'amazighe

Après avoir annoté le corpus collecté selon le processus présenté ci-dessus, nous pouvons synthétiser le nombre des occurrences des éléments du jeu d'étiquettes dans le tableau 2.4 suivant :

Tableau 2.4. Occurrences des parties du discours.

Etiquette de la classe	Désignation	Nombre d'occurrences
V	Verbe	3 190
N	Nom	4 993
A	Nom de qualité	503
AD	Adverbe	516
C	Conjonction	834
D	Déterminant	1076
S	Préposition	2775
FOC	Focalisateur	91
I	Interjection	40
P	Pronom	1 496
PR	Particule	1 593
R	Résiduel (nom étranger, chiffre, date, monnaie, signe mathématique ou autre)	178
F	Ponctuation	3 382
Total		**20 667**

Il est à noter que le corpus présenté dans ce travail a été annoté manuellement par quatre annotateurs. Leur travail consiste en l'affectation des différentes caractéristiques morphosyntaxiques aux textes amazighes susmentionnées dans la section 2.3.

En utilisant cet outil, nous avons étiqueté 20 667 jetons avec un total de 1 438 phrases. Le tableau 2.4 présente les occurrences des parties du discours dudit corpus annoté.

2.7. Difficultés de l'étiquetage grammatical de l'amazighe

Une des difficultés de l'étiquetage est l'ambiguïté: la même forme de surface peut être annotée de différentes manières selon sa position et son utilisation dans la phrase. Voici, à titre d'exemples, quelques cas ambigus tirés du corpus annoté susmentionné:

- ⵉⵏⵉ (ils) peut être un nom commun signifiant « la langue » ou bien le verbe ⵏⵉ (ls) signifiant s'habiller, à l'aoriste ;

- ⴰⵖⴰⵍⵉ (agadir) peut être un nom commun, signifiant " mur ", ou bien un nom d'entité de type lieu ;

- ⵉⵍⵍⵉ (illi) peut avoir au moins les deux catégories grammaticales suivantes: verbe accompli négatif, signifiant "n'existe pas", lorsqu'il est précédé d'une particule de négation, et nom de parenté signifiant " ma fille" ;

- ⵖⵉ ⵙⵖ (insi) peut être un verbe accompli négatif, signifiant “n’a pas passé la nuit”, lorsqu’il est précédé d’une particule de négation, et nom commun signifiant “hérisson” ;

- Quelques mots vides tels que ⵏ (d) qui peut fonctionner comme une préposition, une conjonction de coordination, une particule de prédication ou une particule d’orientation. Par exemple, dans les phrases suivantes, le mot “d” peut être :

- Une conjonction de coordination: ⵜⴰⵎⴰⵣⵉⵖⵜ (Tamazight) ⵏ (et) ⵜⵉⴽⵍⵉⵜⵜⵓⵏⵉⵜⵉⵏ (technologies) ⵜⵉⵎⴰⵢⵉⵜⵜⵉⵏ (nouvelles), “tamaziGt d tiknulujiyin timaynutin”;
- Une préposition: ⵉⵎⴰⵍ (il est allé) ⵏ (avec) ⵜⵓⵔⵉⴷ ⵏ (le chemin),“iman d ubrid”;
- Une particule de prédication: ⵏ (il est) ⴰⵢⴰⵣ ⴰⵔⴳⴰⵣ (un homme),“d argaz”;
- Une particule d’orientation: ⴰⵙⵉ (apporte) ⵏ (ici) ⵜⵉⴽⵍⵉⵜ (bol) ⵜⴰⵎⵊⵊⴰⵀⵉⵜ (moyen), “asi d tikint tamjahdit”.

2.8. Autres utilisations du corpus annoté

Pour étendre ce travail et couvrir les autres niveaux de la langue, à savoir le niveau syntaxique et sémantique, nous avons opté pour une décomposition syntaxique en quatre groupes. Le tableau suivant présente l’ensemble des étiquettes attribuées aux portions des phrases.

Tableau 2.5. Décomposition syntaxique des textes amazighes

Abréviation	**Signification**
Grup.adverb	Groupe adverbial
Grup.nom	Groupe nominal
Grup.prepo	Groupe prépositionnel
Grup.verb	Groupe verbal

Le corpus présenté ci-dessus à été utilisé pour la création d’un dictionnaire de valence (Ouqqua, 2011). Les sous attributs des types de complément, pour le dictionnaire de valence, sont présentés dans le tableau 2.6.

Tableau 2.6. Étiquettes relatives aux types des compléments

Etiquette utilisée	Signification
Adjectival	Complément adjectival
Dative	Complément aditif
Free	Complément libre
Locative	Complément de lieu
Nominal	Complément nominal
Objectival	Complément objet
Prepositional	Complément prépositionnel
Subjectival	Complément subjectival
Temporal	Complément temporel
Verbal	Complément verbal

Dans la figure 2.11 ci-dessous, nous présentons un exemple annoté utilisant les étiquettes présentées ci-dessus pour la phrase : « ha tifaWt ttals xf umuddu nns » [traduction en Français : « Tifawt parle de son voyage »] :

```
<sentence>
    <grup.nom Typecompl="subjectival">
        <d lem="ha" postype="demonstrative" wd="ha"/>
        <n Typecompl="subjectival" gen="f" lem="tifawt" num="s" postype="proper" state="free"
        wd="tifawt"/>
    </grup.nom>
    <grup.verb Typecompl="subjectival">
      <v aspect="imperfective" gen="f" lem="als" num="s" person="3" voice="active" wd="ttals"/>
    </grup.verb>
    <grup.prep Typecompl="dative">
        <s lem="xf" wd="xf"/>
        <n Typecompl="dative" gen="m" lem="amuddu" num="s" postype="common"
        state="construct" wd="umuddu"/>
        <d gen="c" lem="nns" num="s" person="3" postype="possessive" wd="nns"/>
    </grup.prep>
<f punct="colon" wd=":"/>
</sentence>
```

Figure 2.11. Exemple de texte annoté utilisant le jeu d'étiquettes défini pour les besoins du dictionnaire de valence.

2.9. Synthèse

La recherche en linguistique computationnelle est de plus en plus prospectée. Elle permet de dégager des propriétés importantes dans l'étude des langues. Néanmoins, elle ne remplacera pas la recherche qualitative conduite par les linguistes. Mais, grâce aux corpus annotés, elle permet la génération des connaissances et des outils du TAL à base de statistiques.

Ces dernières années ont été marquées par la croissance du nombre de publications en langue et sur la culture amazighes. Cependant, en TAL, la langue amazighe, comme la plupart des langues de diffusion limitée, souffre encore de la pénurie d'outils et des ressources pour son traitement automatique.

La ressource présentée ci-dessus est constituée de plus de 20.000 mots de l'amazighe marocain, choisi à partir de textes variés respectant les règles orthographiques adoptées au Maroc. L'expérience vise également la présentation du processus suivi pour marquer morphosyntaxiquement ce corpus. La vitesse d'annotation est comprise entre 80 et 120 mots par heure et l'accord entre annotateurs est de 94.98%. La même démarche peut être suivie pour l'annotation d'autres langues.

À notre connaissance, le corpus annoté présenté dans le présent chapitre est une première en matière de linguistique computationnelle de la langue amazighe. Il peut également constituer une base pour d'autres types d'annotations syntaxique, sémantique, etc.

Cette ressource, même si elle est de petite taille, est d'une grande utilité pour l'amazighe, comme pour l'apprentissage des étiqueteurs morphosyntaxiques, outils de base pour des travaux plus avancés en TAL.

Dans cette optique, le chapitre suivant présentera l'état de l'art des techniques utilisées pour la réalisation de ce genre d'outils.

CHAPITRE 3:

LES APPROCHES UTILISEES POUR L'ETIQUETAGE MORPHOSYNTAXIQUE

3.1. Introduction

Les domaines traités par l'intelligence artificielle (IA) couvrent les sciences cognitives, la représentation et l'acquisition des connaissances, la robotique, la vision par ordinateur, la reconnaissance des formes, l'apprentissage, le langage naturel, etc. Le TAL est un sous domaine de l'IA, faisant usage de l'ordinateur pour traiter automatiquement le langage humain.

Du traitement de la parole à celui du sens, en passant par celui du texte écrit, le TAL débouche sur les applications les plus diverses : extraction de connaissances, correction orthographique, aide à la traduction et traduction automatique, interrogation de bases de données en langage naturel, web sémantique, etc. Parmi les tâches fondamentales et utiles de plusieurs applications du TAL on retrouve l'étiquetage morphosyntaxique.

L'apprentissage supervisé et semi-supervisé est une sous branche du domaine de l'apprentissage machine et de l'IA plus généralement. En apprentissage supervisé l'objectif est d'apprendre une fonction $h : X \rightarrow Y, x \in X$ étant l'entrée et $y \in Y$ représentant la sortie. Les objets d'entrée sont appelés instances, qui peuvent être de tout type selon la tâche d'apprentissage voulu. L'apprentissage supervisé est utilisé dans plusieurs domaines du TAL, tels que la classification des documents, étiquetage d'une séquence de mots avec une séquence d'étiquettes morphosyntaxiques, comme c'est le cas dans notre travail ; la sortie est une séquence d'étiquettes dont la longueur est égale à la longueur de la chaîne d'entrée.

Dans ce chapitre, nous présenterons l'état de l'art des techniques utilisées pour effectuer l'étiquetage morphosyntaxique, puis nous décrirons les fondements théoriques de l'apprentissage supervisé, en général, et des SVMs et des CRFs en particulier, qui ont donné de bons résultats quant à la classification des séquences.

3.2. Etat de l'art des techniques d'étiquetage morphosyntaxique

De nombreux systèmes d'étiquetage automatique des parties du discours ont été développés pour un large éventail de langues. Parmi ces systèmes, certains s'appuient sur les règles linguistiques et d'autres sur les techniques d'apprentissage automatique (Manning & Schütze, 1999). Les premiers POS tagueurs étaient principalement à base de règles. La construction de tels systèmes nécessitait un travail considérable afin d'écrire manuellement les règles et coder les connaissances linguistiques qui régissent l'ordre de leur application. Un exemple d'étiqueteur à base de règles est TAGGIT, développé par Greene et Robin (Greene, Rubin, 1971). Il contient environ 3300 règles. Ce système atteint une précision de 77%. Par la suite,

l'apprentissage automatique des étiqueteurs s'est avéré à la fois moins pénible et plus efficace que ceux à base de règles. Dans la littérature, de nombreuses méthodes d'apprentissage automatique ont été appliquées pour réaliser des annotateurs morphosyntaxiques, à savoir:

- Les Modèles de Markov Cachés (HMM), dont les états sont les étiquettes ou des tuples d'étiquettes. Les probabilités de transition sont les probabilités d'une étiquette donnant l'étiquette précédente et les probabilités d'émission sont les probabilités d'un mot sachant une étiquette donnée (Rabiner, 1986 ; Charniak et *al.*, 1993) ;

- Le modèle d'entropie maximale MEMM (Ratnaparkhi, 1996) permet la combinaison de diverses formes d'informations contextuelles sans imposer aucune des hypothèses sur les données d'entraînement ; le but en est de maximiser l'entropie de la distribution d'un mot à certaines contraintes contenues dans le corpus de référence ;

- La transformation système basée sur la réduction du taux d'erreur (Brill, 1992 ; Brill, 1995) consiste en l'affectation de l'étiquette la plus fréquente d'un mot donné en utilisant un corpus de référence. Elle procède par la suite à la sélection de la règle qui donne la plus grande erreur. Ce processus est réitéré tant que les résultats d'annotation ne sont pas suffisamment proches de ceux du corpus de référence ;

- Les arbres de décision permettent de construire (Schmid, 1999), sur la base d'un corpus de référence, un outil d'aide à la décision qui utilise les différentes probabilités possibles des étiquettes des mots sous forme d'arbre. La meilleure étiquette attribuée à un mot donné est celle qui donne la plus forte probabilité conditionnelle pour le nœud courant de cet arbre ;

- Les méthodes à base des réseaux de neurones : un réseau de neurones est un modèle de calcul, constitué d'unités de traitement appelées neurones artificiels. Helmut Schmid (Schmid, 1994) présente un modèle de désambiguïsation fondé sur un modèle perceptron multicouche qui calcule une combinaison linéaire des entrées. La fonction de combinaison renvoie le produit scalaire entre le vecteur des entrées et le vecteur des poids synaptiques. L'apprentissage du réseau se fait en adaptant les poids des connections entre les neurones jusqu'à avoir la sortie voulue. Les entrées du modèle sont les mots et les sorties sont les probabilités d'avoir une séquence d'étiquettes sachant la séquence des mots en entrée ;

- Les séparateurs à vaste marge (Kudo & Matsumoto, 2000 ; Giménez & Márquez, 2004) et les champs aléatoires conditionnels (Lafferty et *al.*, 2001; Tellier & Tommasi, 2011 ; Constant et *al.*, 2011), seront respectivement présentés de manière ample, dans les sections 3.3 et 3.4 suivantes.

La qualité des modèles est souvent liée à la quantité de données utilisées dans l'apprentissage. Ainsi, à partir d'exemples appris dans la phase d'apprentissage, les programmes s'appuyant sur cette technique affectent l'étiquette aux mots selon le contexte.

Il existe également des méthodes hybrides qui utilisent à la fois des règles à base de connaissances linguistiques codées manuellement et les méthodes d'apprentissage automatique.

Les résultats des techniques récentes en étiquetage morphosyntaxique sont supérieurs à 95%. Bien que ces méthodes aient une bonne performance, la précision des mots inconnus est beaucoup plus faible que celle des mots connus, ce qui pose problème lorsque le corpus d'apprentissage est de petite taille. La taille du jeu d'étiquettes varie considérablement selon la langue en question et la finalité de la tâche d'étiquetage voulue. Leech (1997) rapporte que le nombre d'étiquettes varie de 32 à 270 dans les principaux corpus anglais. Dans la pratique, la plupart des analyseurs limitent le nombre d'étiquettes et ignorent certaines distinctions difficiles à désambiguïser automatiquement ou sujettes à discussion du point de vue linguistique.

Après cette brève introduction de l'état d'art des techniques utilisées dans la tâche du POS tagging, nous allons présenter dans les sections suivantes le cadre théorique des méthodes d'apprentissage supervisés et plus particulièrement, les SVMs et les CRFs que nous avons utilisés lors des expérimentations pour la création de l'étiqueteur morphosyntaxique amazighe.

3.3. Introduction aux séparateurs à vaste marge

Les séparateurs à vaste marge ou les machines à vecteur support sont une généralisation des classifieurs linéaires et des modèles discriminants permettant, sur la base d'un ensemble de données étiquetées, de maximiser les marges entre classes et de minimiser les erreurs en recherchant des séparateurs optimaux entre les données. Ils reposent sur deux idées clés : la notion de maximisation de la marge, qui consiste en la résolution du problème de maximisation de la distance entre les données des classes et le plan de marge ; et la notion des noyaux utilisée pour traiter les cas où les données ne sont pas linéairement séparables, en transformant l'espace de représentation des données en un espace de plus grande dimension dans lequel les données sont linéairement séparables. Les SVMs ont été introduits par Vladmir Vapnik au début des années 90 (Vapnik, 1995). Ils ont été utilisés à leur début dans les problèmes de classification et de régression. De nos jours, Ils sont utilisés dans plusieurs

domaines de recherche et d'ingénierie tel que le TAL, le diagnostic médical, la biologie, etc. Dans cette section, nous présenterons la forme originale des SVMs : le cas binaire et le cas de multiclasse, ainsi que leur utilisation en TAL.

3.3.1. Les SVMs binaires

Les SVMs binaires consistent en la classification en deux classes : +1 et -1. L'idée est de trouver un hyperplan (droite dans le cas de deux dimensions) optimal séparant le mieux lesdites classes. La marge est la distance entre la frontière de séparation et les échantillons les plus proches, appelés vecteurs supports. Lorsque les données sont linéairement séparables, nous parlons d'une machine à vecteurs supports à marge dure ; et lorsqu'elles ne le sont pas ou bien lorsqu'elles contiennent des données bruitées, nous parlons d'une machine à vecteurs supports à marge souple.

Les SVMs à marge dure

L'hyperplan séparateur est représenté par l'équation (3.1) :

$$H(x) = w^T x + b \tag{3.1}$$

Où $x = (x_1, x_2, \dots, x_n)$ est un vecteur d'entrée et $w = (w_1, w_2, \dots, w_n)$ est un vecteur de poids. En fonction de la valeur de $H(x)$, nous pouvons déterminer la classe comme suit :

$$Classe(x) = \begin{cases} +1 \ \ si \ \ H(x) > 0 \\ -1 \ \ si \ \ H(x) < 0 \end{cases} \tag{3.2}$$

Puisque les deux classes sont linéairement séparables, aucun exemple ne satisfait l'équation $H(x) = 0$. En posant β la distance minimale entre l'hyperplan séparateur et l'exemple le plus proche, l'équation précédente devient :

$$Classe(x) = \begin{cases} +1 \ \ si \ \ H(x) \geq \beta \\ -1 \ \ si \ \ H(x) \leq -\beta \end{cases} \tag{3.3}$$

La région qui se trouve entre les deux hyperplans $H(x) = \beta$ et $H(x) = -\beta$ est appelée région de généralisation de la machine d'apprentissage et les vecteurs appartenant à ces deux plans sont nommés vecteurs supports.

En classification binaire, nous recherchons une classe y_i, ayant l'une des deux valeurs: +1 et -1. Ainsi, en multipliant chacun des deux termes de l'inéquation 3.3 par y_i on obtient la

formule 3.4 suivante:

$$y_i(w^T x_i + b) \geq \beta,\ i \in [\![1, n]\!] \qquad 3.4$$

Dans les SVMs, la frontière de séparation est choisie de manière à maximiser la marge (Figure 3.1). La résolution de ce problème consiste à rechercher la frontière séparatrice optimale, à partir d'un ensemble de données d'apprentissage.

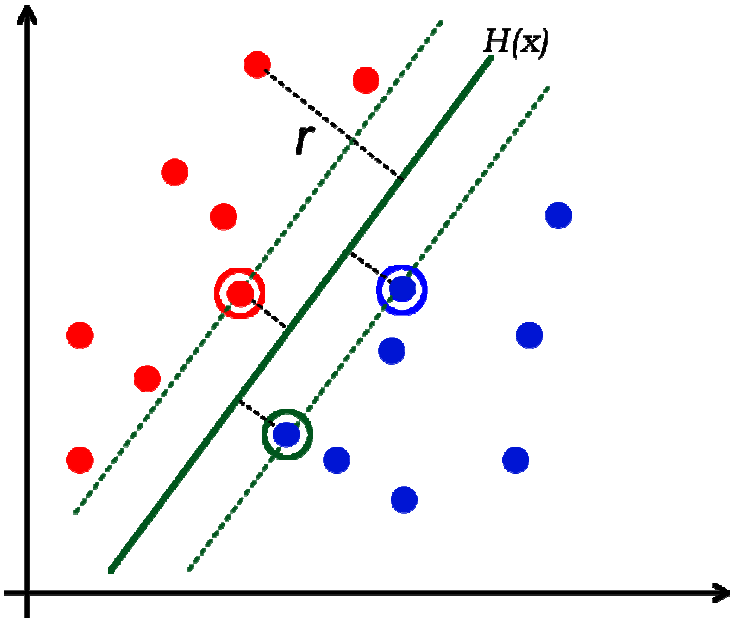

Figure 3.1. Séparation des régions par un hyperplan

La distance séparant l'hyperplan et un exemple donné est:

$$r(x, H) = \frac{|w^T x + b|}{\|w\|} \geq \frac{\beta}{\|w\|} \qquad 3.5$$

L'hyperplan séparateur maximisant cette distance est donné par :

$$\arg\max_{w,b} \left\{ \frac{1}{\|w\|} \min_i [y_i(w^T x_i + b)], avec\ i \in [\![1, n]\!] \right\} \qquad 3.6$$

La maximisation de cette distance revient à maximiser $\|w\|^{-1}$ et par conséquent, à minimiser

$\frac{1}{2}\|w\|^2$. Le carré de la fonction racine est pris pour faciliter les calculs et ne pas inclure la fonction racine carrée de la norme. Le facteur $\frac{1}{2}$, lui, est utilisé afin de faciliter la lisibilité des calculs et du résultat de l'optimisation.

La contrainte $\beta = 1$ est posée afin de simplifier les calculs. En effet, si *w* et *b* sont une solution du problème de maximisation, *aw* et *ab* forment aussi une solution dudit problème, dans lequel *a* est une constante non nulle.

L'hyperplan optimal peut être obtenu en résolvant l'équation :

$$\begin{cases} Minimiser\ \frac{1}{2}\|w\|^2 \\ sous\ les\ contraintes \\ y_i(w^T x_i + b) \geq 1, \forall\ \mathrm{i} \in [\![1, n]\!] \end{cases} \quad 3.7$$

Cette équation se résout par la méthode des multiplicateurs de Lagrange, où le lagrangien s'écrit sous la forme :

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \{y_i(w^T x_i + b) - 1\} \quad 3.8$$

Dans cette équation, les α_i sont les multiplicateurs non négatifs de Lagrange. L'optimum de cette fonction objective *L* est obtenu en la minimisant par rapport à *w* et *b* et l'on obtient successivement les deux équations 3.9. (a) et (b).

$$\begin{cases} w^* = \sum_{i=1}^{n} \alpha_i y_i x_i \quad (a) \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \qquad (b) \end{cases} \quad 3.9$$

En remplaçant la valeur w^* dans l'équation 3.8, nous obtenons :

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j \quad 3.10$$

En remplaçant la valeur w^* dans l'équation 3.1, on obtient l'équation de l'hyperplan suivante :

$$H(x) = \sum_{S} \alpha_i y_i x_i^T x + b \qquad 3.11$$

Dans cette équation, S représente l'ensemble des vecteurs supports. Le terme b peut être calculé à partir de n'importe quel exemple. Néanmoins, et pour des raisons de précision, nous prenons la moyenne de *b* pour tous les vecteurs supports :

$$b = \frac{1}{|S|} \sum_{S} y_i - w^T x_i \qquad 3.12$$

La fonction de décision *H* est calculée pour chaque nouvel exemple *x*.

Les SVMs à marge soft

Dans le cas où les données sont inséparables ou bien contiennent du bruit, des variables de relaxation non négatives ξ_i sont ajoutées aux contraintes sur les marges permettant d'obtenir des marges souples. L'équation obtenue est :

$$y_i(w^T x_i + b) \geq 1 - \xi_i, avec\ i \in [\![1, n]\!] \qquad 3.13$$

Dans le cas où $\xi_i < 1$, l'exemple x_i reste bien classé mais ne respecte pas la marge ; si $\xi_i \geq 1$, alors x_i est mal classé par l'hyperplan. Par conséquent, la recherche de l'hyperplan optimal doit prendre en considération la minimisation des erreurs permises, i.e. minimiser $\sum_{i=1}^{n} \xi_i$.

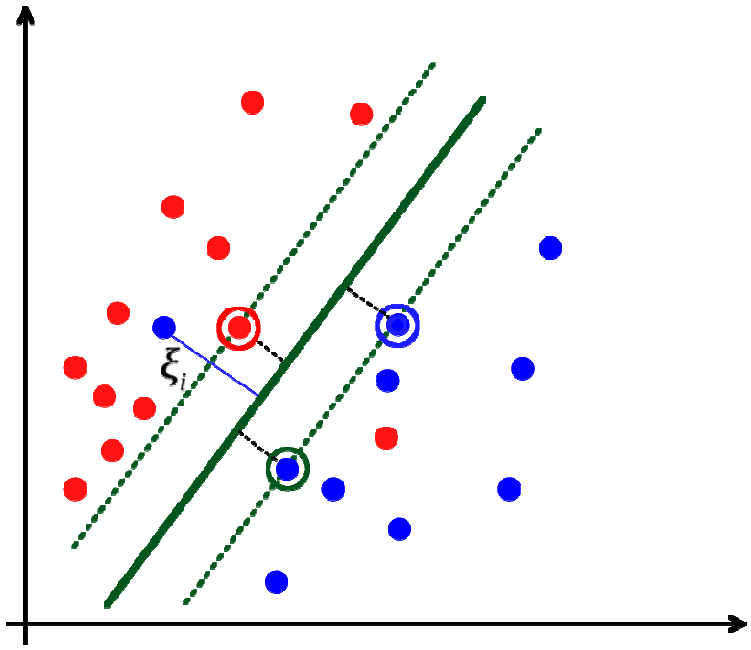

Figure 3.2. SVM binaire à marge souple

L'équation de minimisation 3.7 devient ainsi :

$$\begin{cases} Minimiser\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ sous\ les\ contraintes \\ y_i(w^T x_i + b) \geq 1 - \xi_i, \mathrm{i} \in [\![1,n]\!] \end{cases} \quad 3.14$$

Ici, C est un nombre positif fixe qui représente une balance entre la maximisation de la marge et la minimisation de l'erreur de classification. Il est choisi par l'utilisateur. Après introduction des multiplicateurs de Lagrange et calcul des dérivées partielles, on obtient l'équation de l'hyperplan optimal suivante :

$$H(x) = \sum_{i \in U} \alpha_i y_i x_i^T x + b \quad 3.15$$

U y représente les vecteurs supports non bornés. Le terme *b* est défini par l'équation :

$$b = \frac{1}{|U|}\sum_{i \in U} y_i - w^T x_i \quad 3.16$$

La seule différence avec les SVMs à marge dure est que les α_i sont inférieurs ou égaux à C. Ils ont trois configurations :

- $\alpha_i = 0$; l'exemple x_i est bien classé ;
- $0 < \alpha_i < C$; l'exemple x_i est un vecteur support. Il est appelé dans ce cas vecteur support non borné ;
- $\alpha_i = C$; Dans ce cas $\xi_i \geq 0$ et par conséquent $y_i(w^T x_i + b) = 1 - \xi_i$; x_i est un vecteur support borné. Si $\xi_i < 1$, x_i est bien classé, sinon $\xi_i \geq 1$ et l'exemple x_i est mal classé.

Les conditions de résolution de ce problème d'optimisation sous contraintes sont appelées les conditions Karush-Kuhn-Tucker (Karush, 1939 ; Kuhn and Tucker, 1951)**.**

3.3.2. Les SVMs multi classe

Les méthodes des machines à vecteurs support multi classe réduisent le problème multi classe à une composition de plusieurs hyperplans bi classes, traçant les frontières de décision entre les différentes classes. Ces méthodes décomposent l'ensemble des exemples en sous ensembles, représentant chacun un problème de classification binaire. Plusieurs méthodes ont été utilisées pour étendre la classification binaire à la classification en plusieurs classes. Parmi ces méthodes on trouve les graphes de décision développés par Platt et ses co-équipiers (Platt et *al.*, 2000) et les méthodes basées sur les arbres de décision (Schwenker, 2001 ; Takahashi & Abe, 2002) . Les deux méthodes les plus connues sont une-contre-reste et une-contre-une.

a- Une-contre-reste (1vs R) :

Elle consiste à déterminer, pour chaque classe *k*, un hyperplan la séparant de toutes les autres classes (Vapnik, 1995). Ces dernières sont les classes négatives. Un hyperplan H_k est défini pour chaque classe *k* par la fonction de décision suivante :

$$f_k(x) = \begin{cases} +1 \; si \; H_k(x) > 0; \\ 0 \qquad sinon \end{cases} \qquad 3.17$$

La valeur retournée permet de déterminer si un exemple *x* appartient ou non à une classe *i*. Dans le cas où $f_k(x) = 0$, nous n'avons aucune information sur l'appartenance de x aux autres classes. Ainsi, pour connaître la classe, nous présentons l'exemple *x* aux différentes classes. Si pour une et une seule valeur k_0, $f_{k_0}(x) = 1$ et pour les autres classes $f_{k \neq k_0}(x) = 0$, nous concluons que le vecteur en question appartient à la classe k_0.

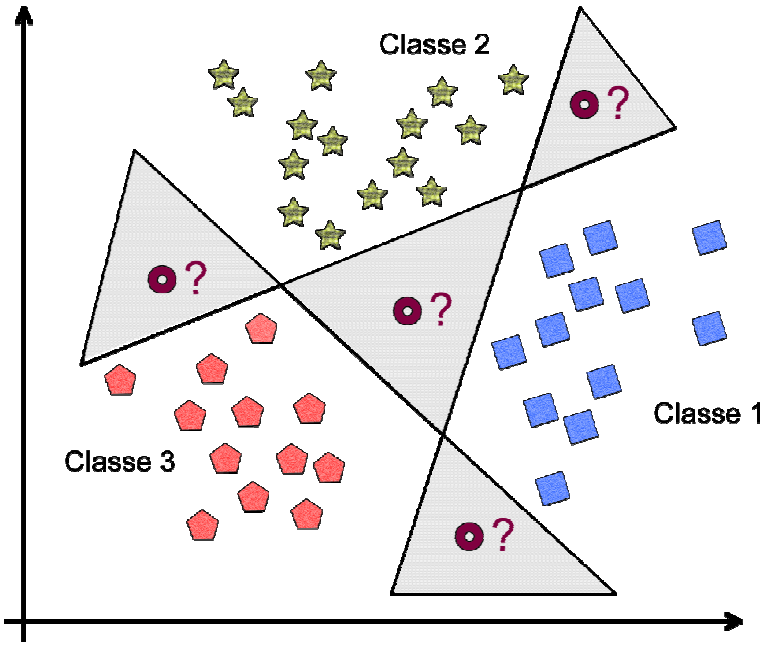

Figure 3.3. Approche une-contre-reste avec des zones d'indécision.

Si pour deux ou plusieurs hyperplans $f_k(x) = 1$, x appartient à des régions d'ambigüité et l'exemple *x* est dit non classifiable. La figure 3.3 représente un cas de séparation de 3 classes utilisant la méthode une-contre-reste.

Pour les cas appartenant aux zones d'incertitude, l'approche une-contre-reste adopte la classe k, qui maximise l'équation :

$$Arg\ \max_{1\leq k\leq K}({w_k}^T x + b_k) \qquad 3.18$$

Cette méthode utilise le principe du gagnant prend tout. Géométriquement, la classe affectée est celle qui maximise la distance de l'exemple à l'hyperplan ayant $f_k(x) = 0$.

La méthode un-contre-reste est critiquée pour son asymétrie (Scholkopf & Smola, 2002) : elle utilise beaucoup plus d'exemples négatifs que d'exemples positifs.

b- Une-contre-une (1vs 1) :

Elle consiste à utiliser un classificateur pour chaque paire de classes. Elle revient à Knerr et ses co-équipiers (1990) pour les réseaux de neurones. Cette méthode discrimine chaque classe de chaque autre classe, ainsi $k(k-1)/2$ fonctions de décisions sont apprises. Pour chaque paire de classes *i* et *j* la méthode définit une fonction de décision binaire :

$$f_{ij}(x) = \begin{cases} +1\ si\ H_{ij}(x) > 0; \\ 0 \qquad sinon \end{cases} \qquad 3.19$$

L'affectation d'un nouvel exemple se fait par liste de vote. Nous testons un exemple par le calcul de sa fonction de décision pour chaque hyperplan. Pour chaque exemple, on vote pour la classe à laquelle il appartient. Sur la base des $k(k-1)/2$ fonctions de décisions prédéfinies, k autres fonctions de décision sont définies pour calcul du vote pour une classe i donnée. La classe retenue est la plus votée.

La méthode une-contre-une utilise plus d'hyperplans que la méthode une-contre-reste. Néanmoins, les exemples utilisés dans l'entrainement sont limités, et chaque paire de classes prise a moins d'exemples chevauchés qu'une classe contre le reste. Dans le cas d'égalité, la méthode une-contre-une choisit la classe de façon aléatoire.

3.3.3. Utilisation des noyaux

La fonction d'optimisation et de recherche de la marge ne dépend que du produit scalaire $x_i{}^T.x_j$ des exemples d'entrainement tel que cela est mentionné dans l'équation 3.8. L'idée d'utiliser des noyaux (Mercer, 1909 ; Aizerman et *al.*, 1964) consiste en la reconsidération des données non séparables linéairement dans un espace de dimension plus grande, éventuellement infinie et où les données peuvent être séparées. Cette transformation d'espace est faite à l'aide d'une fonction ϕ de transformation vers le nouvel espace, appelé espace de caractéristiques (voir l'exemple de figure 3.4 ci-dessous).

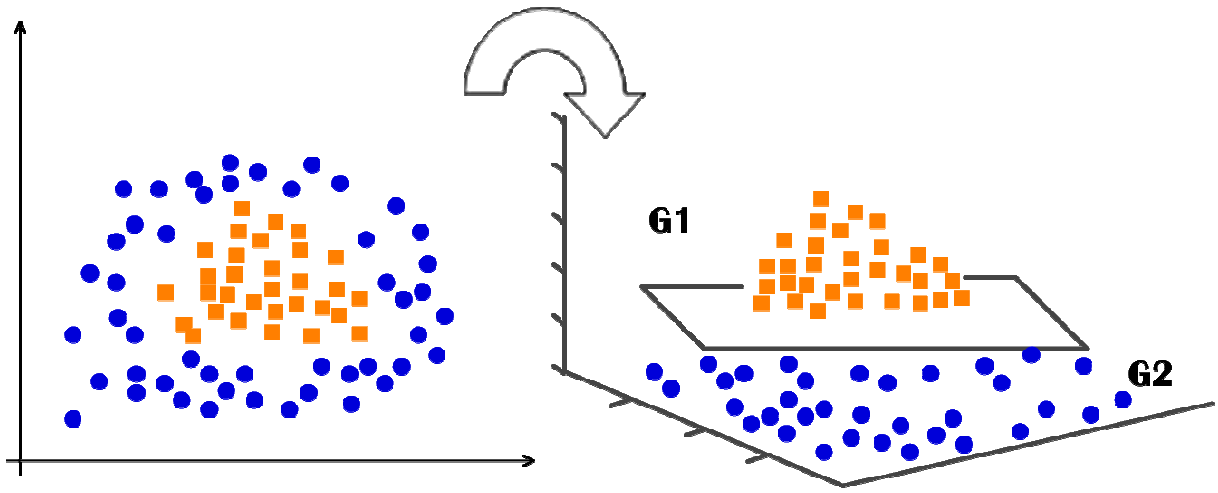

Figure 3.4. Exemple de plongement non linéaire.

La fonction objective à optimiser 3.10 se réécrit en fonction de la fonction de transformation ϕ ainsi:

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(x_i)^T . \phi(x_j) \quad 3.20$$

Le résultat du produit scalaire $\phi(x_i)^T . \phi(x_j)$ est un scalaire. Cette fonction permet d'apprendre des relations non linéaires par des machines linéaires.

Au lieu de rechercher la fonction ϕ, on calcule la fonction $K(x_i, x_j)$. Selon le théorème de Mercer, et considérant la matrice *G* de Gram ci-dessous, cette dernière doit être symétrique et ses valeurs propres positives.

$$G = \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_n) \\ \vdots & \ddots & \vdots \\ K(x_n, x_1) & \cdots & K(x_n, x_n) \end{bmatrix} \quad 3.21$$

L'exemple le plus simple de fonction noyau est le noyau linéaire :

$$K(x_i, x_j) = {x_i}^T . x_j \quad 3.22$$

Les noyaux standards utilisés avec les SVMs sont le noyau polynomial et noyau gaussien.

a- Le noyau polynomial

$$K(x_i, x_j) = ({x_i}^T . x_j)^d \quad 3.23$$

Dans cette équation, *d* est une puissance naturelle. Si $d = 1$, le noyau devient linéaire. Le noyau polynomial non homogène $K(x_i, x_j) = ({x_i}^T . x_j + 1)^d$ est aussi utilisé.

b- Le noyau gaussien

$$K(x_i, x_j) = e^{\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)} \quad 3.24$$

Ici, σ est une constante réelle qui représente la largeur de la bande du noyau.

3.3.4. Applications des SVMs

Les avantages théoriques des SVMs, consistant essentiellement dans la minimisation de l'erreur empirique et structurelle, en ont fait un outil très prisé pour résoudre moult problèmes de classification : en médecine, biologie, détection de spam, TAL, etc. En TAL, les

SVMs sont appliquées dans plusieurs tâches. Par exemple, Kudo et Matsumoto ont utilisé les SVMs dans l'analyse partielle des textes anglais (Kudu, Matsumoto, 2000). Il en est de même dans l'analyse partielle (Diab et *al.*, 2004), la reconnaissance d'entités nommées (Benajiba et *al.*, 2010b) et bien d'autres.

Plusieurs étiqueteurs grammaticaux ont été réalisés sur la base des SVMs. C'est le cas, par exemple pour l'Arabe (Diab 2007) et le Bengali (Ekbal, Bandyopadhyay, 2008). Les SVMs atteignent des performances élevées sans sur-apprentissage même en utilisant plusieurs caractéristiques. Ils réagissent également bien avec les données éparses et bruitées.

Les détails de l'utilisation des SVMs et des caractéristiques utilisées pour la réalisation de notre étiqueteur morphosyntaxique sont présentés dans le chapitre 4 de ce travail.

3.4. Introduction aux champs markoviens conditionnels

Les processus stochastiques ont pour finalité la modélisation des observations et leurs étiquetages. L'étiquetage d'une observation X consiste à trouver la configuration d'étiquettes $\hat{Y}$ qui maximise la probabilité conditionnelle de Y sachant l'observation X:

$$\hat{Y} = \arg\max_{Y \in Y^n} P(Y|X) \qquad 3.25$$

Nous distinguons deux processus permettant de lier ces réalisations : soit génératifs, et dans ce cas, il modélise une probabilité conditionnelle de l'observation sachant l'étiquetage ; soit il est discriminant, et dans ce cas il modélise la probabilité conditionnelle de l'étiquetage sachant l'observation.

3.4.1. Les modèles génératifs

Dans les modèles génératifs définissent une probabilité jointe$P(X, Y)$, tel que les HMMs, on doit énumérer toutes les séquences d'observations possibles. Dans la pratique ceci n'est pas envisageable à cause de l'explosion combinatoire. Ces modèles sont fondés sur la décomposition de la formule de Bayes :

$$P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)} = \frac{P(X,Y)}{P(X)} \qquad 3.26$$

La configuration de $\hat{Y}$ peut être réécrite comme suit :

$$\hat{Y} = \arg\max_{Y \in Y^n} \frac{P(X|Y) \times P(Y)}{P(X)} \quad 3.27$$

Comme la marginale P(X) est une constante, nous pouvons simplifier le calcul de $\hat{Y}$ comme suit :

$$\hat{Y} = \arg\max_{Y \in Y^n} P(X|Y) \times P(Y) \quad 3.28$$

La détermination de $\hat{Y}$ nécessite de modéliser deux éléments $P(X,Y)$ et $P(Y)$. *P(X|Y)* modélise la façon permettant de générer les données observées en supposant que les étiquettes sont connues.

Les limitations de cette approche sont :

- nécessité de disposer d'une grande quantité de données, afin qu'elle soit la plus exacte possible ;
- usage de l'hypothèse d'indépendance des observations. Cette dernière est rarement vérifiée dans les problèmes réels.

Les HMMs utilisent l'hypothèse d'indépendance des observations. Cette hypothèse constitue leur limitation majeure.

3.4.2. Les modèles discriminants

Ces modèles sont fondés sur la modélisation de l'étiquetage sachant les observations. Aussi, ils ne font pas d'hypothèse sur l'indépendance des observations. Dans ces modèles, les probabilités de transitions entre étiquettes peuvent dépendre des observations passées et futures.

La probabilité conditionnelle d'une réalisation Y sachant une observation X peut être décomposée comme un produit des probabilités conditionnelles locales :

$$P(Y|X) = \prod_{i=1}^{n} P(y_i|y_{N_i}, X) \quad 3.29$$

Où N_i correspond aux variables aléatoires voisines de la variable y_i et n le nombre total des variables aléatoires.

Les CRFs sont, avec les Modèles de Markov à Entropie Maximale (MMEMs), les deux principaux modèles discriminants. Bien que les MMEMs aient obtenu de bons résultats sur les tâches d'extraction d'information et de segmentation (MCallum, 2000), ils souffrent du problème du biais du label. En effet, si le graphe est tel qu'un nœud i n'a qu'un successeur i+1, alors la masse de probabilité est entièrement transmise à y_{i+1} indépendamment des observations x, appelé biais du label. Les CRFs permettent de palier ce problème et cela en calculant les poids de transition non normalisée et en calculant un facteur de normalisation sur l'ensemble de la séquence y conditionnellement à x.

3.4.3. Les modèles graphiques

Les modèles graphiques sont un formalisme adéquat pour exploiter et représenter les structures de dépendances entre les entités. Traditionnellement, les modèles graphiques ont été utilisés pour représenter la probabilité distributionnelle *P(X,Y)*, dans laquelle les variables *Y* représentent les étiquettes des entités qu'on cherche à prédire, et les variables *X* représentent les observations. Modéliser une distribution *P(X)* incluant des dépendances vers des caractéristiques variées et complexes peut mener à des modèles complexes. Ignorer ses caractéristiques peut impacter les performances du modèle. Une solution à ce problème consiste à modéliser la distribution condiontionnelle *P(Y|X)*. Et c'est l'approche adoptée par Lafferty et ses co-équipiers(2001).

Définition : Soit *G= (V, E)*, où V est l'ensemble des sommets et E l'ensemble des arcs, un graphe non orienté et soient *X* et *Y* deux champs aléatoires décrivant respectivement l'ensemble des étiquettes, de sorte que, pour chaque nœud i appartenant à V, il existe une variable aléatoire yi dans Y : nous désignons (X, Y) comme étant un champ aléatoire conditionnel si chaque variable aléatoire yi respecte la propriété de Markov suivante :

$$p(yi|X, yj\ i \neq j) = p(yi|X, yj, i \sim j) \qquad 3.30$$

$i \sim j$ signifie que i et j sont voisins dans G.

Par conséquent, cette propriété n'est satisfaite que si chaque variable aléatoire ne dépend que de ses voisins : yi ne dépend que de X et des y_j ses voisins dans le graphe d'indépendance.

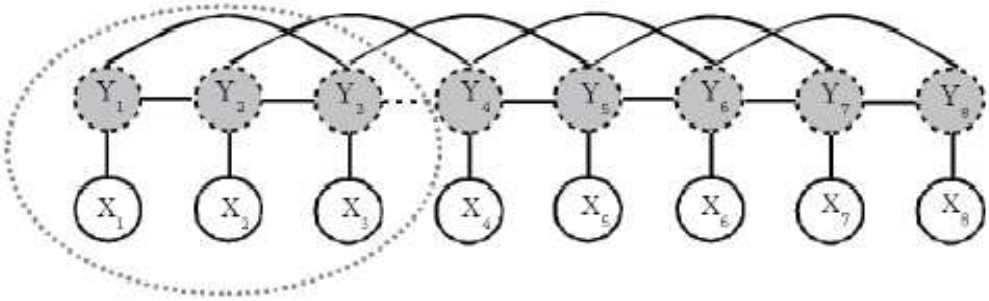

Figure 3.5. Exemple d'un graphe des CRFs.

Dans la figure 3.5, la partie encerclée est une clique de trois nœuds.

3.4.4. Les CRFs

Les champs markoviens conditionnels ou les CRFs sont des processus stochastiques qui modélisent les dépendances entre un ensemble d'observations discrètes réalisées sur une séquence discrète et un ensemble d'étiquettes. Dans le cas de l'analyse morphosyntaxique, la suite des mots est la séquence discrète. En comparaison avec les Modèles de Markov Cachés, un CRF ne repose pas sur l'hypothèse forte d'indépendance des observations entre elles conditionnellement aux états associés.

Un champ aléatoire conditionnel est la probabilité conditionnelle $P(Y|X)$ avec une structure graphique associée, parce que le modèle est conditionnel. Les dépendances entre les observations *x* n'ont pas besoin d'être représentées explicitement ; elles permettent l'emploi des caractéristiques des entrées. Il en est ainsi par exemple en TAL, de l'emploi des mots du voisinage, les préfixes, les suffixes, etc.

Les CRFs sont des modèles graphiques probabilistes, se basant et sur la théorie des graphes et sur la théorie des probabilités. Ces deux théories permettent de modéliser le problème de classification des séquences : la théorie des graphes permet la modélisation des structures de séquence des étiquettes des phrases ; la théorie des probabilités, elle, permet de gérer les ambigüités causées par les séquences des étiquettes.

La modélisation des CRFs est faite sous forme de graphe permettant de contextualiser les relations entre les étiquettes. Sachant les paramètres du modèle graphique et connaissant l'observation, la tâche consiste à trouver la réalisation la plus probable du champ aléatoire correspondant à l'étiquetage.

D'après le théorème de Hammersely-Clifford (Hammersly et *al.*, 1971), la distribution de probabilité p d'un champ de Markov est décomposable comme un produit de fonctions φ_c définies sur cliques, sous graphes complets, maximales c de l'ensemble des cliques C de G. Ainsi, la probabilité d'un étiquetage y étant donnée une observation x s'écrit :

$$p(Y|X) = \frac{1}{Z(X)} \prod_{c \in C} \varphi_c(y_c, X) \qquad 3.31$$

Dans cette équation y_c est la réalisation des variables aléatoires de la clique *c* et *Z(X)* est un coefficient de normalisation défini comme suit :

$$Z(X) = \sum_{y} \prod_{c \in C} \varphi_c(y_c, X) \qquad 3.32$$

Z(X) est un coefficient de normalisation égal au produit des fonctions du potentiel de tous les étiquetages possibles sur la base de la séquence d'observation x.

Lafferty et ses co-équipiers (Lafferty et *al.*, 2001) ont proposé de définir la forme de la fonction φ_c comme étant l'exponentiel des sommes pondérées des fonctions caractéristiques f_k ayant des poids w_k .

$$\varphi_c(y_c, X, W) = \exp\left(\sum_{k=1}^{K} w_k f_k(y_c, X)\right) \qquad 3.33$$

La forme de ces fonctions dépend du domaine d'application. Par exemple, dans le TAL, il s'agit généralement de fonctions binaires qui testent la présence ou l'absence de certaines caractéristiques, telles que les mots précédents, leurs étiquettes, les préfixes, les suffixes…etc. Les poids w_k permettent d'accorder plus ou moins d'importance à chacune des fonctions caractéristiques. Ils sont fixés lors de la phase d'apprentissage en cherchant à maximiser la log-vraisemblance sur un ensemble d'exemples déjà annotés, et qui forment le corpus de référence.

Ainsi, ayant une phrase donnée, nous pouvons transformer le score de cette phrase en une

probabilité $p(Y|X)$ entre 0 et 1, et ce en divisant ce score par le facteur de normalisation Z(X). Par conséquent, la probabilité d'un étiquetage sachant une réalisation d'observations s'exprime ainsi par :

$$\mathrm{p(Y|X)} = \frac{1}{\mathrm{Z(X)}} \exp\Big(\sum_{c \in C} \sum_{k=1}^{K} \mathrm{w_k f_k}\,(\mathrm{y_c}, \mathrm{X})\Big) \qquad 3.34$$

Ayant un ensemble d'entrainement, parmi les méthodes utilisées pour apprendre les poids des fonctions caractéristiques on trouve la méthode du gradient, en l'appliquant aux coefficients $\mathrm{w_k}$:

$$\frac{\partial}{\partial \mathrm{w_k}} log\ \mathrm{p(Y|X)} \qquad 3.35$$

Pour la recherche de l'étiquetage optimal, une façon de faire consiste en le calcul de $\mathrm{p(Y|X)}$ pour chaque étiquetage possible, ensuite choisir l'étiquetage qui maximise cette probabilité. Néanmoins, et vu que nous avons T^L cas d'étiquetage possible pour un jeu d'étiquettes de taille *T* et une phrase de taille L, ce calcul n'est pas possible. Pour résoudre ce problème on utilise l'algorithme de Viterbi (Forney, 1973).

3.4.5. Applications des CRFs

Les CRFs ont attiré l'attention dans plusieurs domaines de recherche, comme c'est le cas, par exemple, dans le traitement de texte (Taskar et *al.*, 2002; Peng & McCallum, 2004), les bioinformatiques (Sato & Sakakibara, 2005; Liu et *al.*, 2005) et la vision par ordinateur (He et *al.*, 2004, Kumar and Hebert, 2003). En relation avec le TAL, les CRFs ont été appliqués à de nombreuses tâches, dont à titre indicatif, l'analyse syntaxique partielle (Sha, Pereira, 2003), l'extraction d'informations à partir des tables (Pinto et *al.*, 2003), la reconnaissance d'entités nommées (Li & McCallum 2003, Benajiba et *al.*, 2010a), etc. Pour l'étiquetage morphosyntaxique, les CRFs ont été utilisés pour de nombreuses langues, telles que l'Amharique (Adafre, 2005), le Tamoul (Lakshmana, Geetha, 2009), etc.

Les détails de l'utilisation des CRFs et des caractéristiques utilisées pour la réalisation de notre étiqueteur morphosyntaxique sont présentés dans les chapitres 4 et 5 de ce travail.

3.5. Synthèse

L'étiquetage morphosyntaxique consiste en l'annotation de chaque mot d'une phrase avec une étiquette récapitulant une information morphosyntaxique selon le contexte. C'est une tâche

importante dans le TAL. Il peut être utilisé dans des travaux d'analyse variés : la recherche de l'information, la synthèse vocale, la traduction automatique, etc.

Plusieurs techniques ont été utilisées pour la réalisation de l'étiquetage morphosyntaxique. Dans ce chapitre nous avons présenté l'état de l'art des méthodes utilisées dans l'étiquetage morphosyntaxique. Nous nous sommes concentré sur l'introduction des vecteurs supports à large marge ainsi que sur les champs aléatoires conditionnels, deux techniques qui ont donné de bons résultats dans les tâches d'étiquetage des séquences.

Dans le chapitre suivant, nous présenterons comment nous avons utilisé les vecteurs supports à large marge et les champs aléatoires conditionnels avec les propriétés lexicales et les propriétés du contexte pour la réalisation d'un segmenteur et d'un étiqueteur morphosyntaxique de la langue amazighe.

CHAPITRE 4:

ETIQUETAGE MORPHOSYNTAXIQUE DE L'AMAZIGHE AVEC USAGE DE LA SEGMENTATION

4.1. Introduction

L’étiquetage morphosyntaxique consiste en l’annotation de chaque mot d’une phrase avec une étiquette récapitulant une information morphosyntaxique selon le contexte. Il permet d’augmenter l’information des mots étiquetés des couches supérieures pour le TAL. Il s'agit de la première couche au-dessus du niveau lexical et du niveau le plus bas de l'analyse syntaxique.

Dans la littérature, plusieurs méthodes d'apprentissage automatique ont été appliquées afin de réaliser des annotateurs morphosyntaxiques, à savoir les HMMs, les MEMMs, les transformations système basées sur la réduction du taux d’erreur, les réseaux de neurones, etc. Parmi ces techniques, nous avons utilisé les SVMs et les CRFs, deux techniques d’annotation des séquences, qui ont donné de bons résultats dans ce genre de tâches.

Dans ce chapitre, nous décrirons dans un premier temps un jeu d’étiquettes réduit, utilisé pour la création du premier annotateur morphosyntaxique de la langue amazighe, basé sur 15 étiquettes morphosyntaxiques. Ensuite, nous présenterons les résultats des expérimentations quant à l’utilité de la segmentation des mots. Puis, nous présenterons AMTS un jeu d’étiquettes enrichi de 28 étiquettes et les expérimentations relatives à la réalisation de l’annotateur morphosyntaxique correspondant. Enfin, nous discuterons les résultats et analyserons les erreurs du système.

4.2. Expérimentation de l’étiquetage morphosyntaxique sur la base d’un jeu d’étiquettes réduit

Définir un jeu de balises adéquat est une tâche essentielle dans la construction d'un POS tagger automatiquement. La conception d’un tel jeu consiste en la définition d’un ensemble de balises traitables qui ne soit ni grand et nuire à la performance de l’apprentissage automatique, ni petit et ne pas offrir ainsi assez d'informations pour être utilisé par les systèmes fédérateurs. Dans (Outahajala et *al.*, 2010) est traité un ensemble contenant 15 balises représentant les principales parties du discours dans la langue amazighe plus les deux étiquettes "S_P" et "N_P" désignant respectivement les prépositions et les noms de parenté lorsqu’ils sont suivis des pronoms personnels. Ce jeu d’étiquettes est résumé dans le tableau 4.1. La distinction entre les types de noms, le genre, la personne, l’aspect et autres informations n'ont pas été inclus dans ce jeu d’étiquettes.

Tableau 4.1. Jeu d'étiquettes de base.

Classe	Désignation
V	Verbe
N	Nom
N_P	Nom de parenté suivi d'un pronom personnel
A	Nom de qualité/Adjectif
AD	Adverbe
C	Conjonction
D	Déterminant
S	Préposition
S_P	Préposition suivie d'un pronom personnel
FOC	Focalisateur
I	Interjection
P	Pronom
PR	Particule
R	Résiduel (nom étranger, chiffre, date, monnaie, signes mathématiques et autres)
F	Ponctuation

Dans ces expérimentations, nous avons utilisé comme ligne de base le modèle de référence basé sur la fréquence des mots (Freq-Base.). Il s'agit d'un algorithme basé sur la fréquence des étiquettes des mots. L'étiquette prévue pour un mot est tout simplement l'étiquette la plus fréquente qui a été associée à celui-ci dans les données de formation. L'implémentation de cet algorithme est librement disponible[52]. Ce modèle ignore totalement le contexte environnant et résout les cas ambigus utilisant uniquement les fréquences des étiquettes. Une référence similaire a été utilisée dans la tâche partagée de la reconnaissance des entités nommées (NER) de la conférence sur l'apprentissage informatique en langage naturel[53] (CoNLL).

Concernant l'implémentation des SVMs et des CRFs pour les tâches de segmentation et d'étiquetage morphosyntaxique de l'amazighe, le processus d'apprentissage a été conduit en utilisant les outils Yamcha[54] et CRF++[55].

[52] http://www.outamed.com/downloads/baseline.txt

[53] http://www.cnts.ua.ac.be/conll2002/

[54] http://chasen.org/~taku/software/yamcha/

[55] http://crfpp.sourceforge.net/

Yamcha est un outil open source basé sur les SVMs. Il emploie la programmation dynamique pour le choix optimal de l'étiquette, et ce en utilisant les propriétés de contexte pour choisir la séquence d'étiquettes maximisant dynamiquement les étiquettes données. Nous avons utilisé YamCha avec la méthode une-contre-une pour la détermination des frontières de décision entre les différentes classes et les noyaux polynomiaux d'ordre deux pour la transformation de l'espace des données. Pour la classification, on a utilisé YamCha avec TinySVM[56], outil public pour la reconnaissance des classes : +1 et -1.

Quant à l'outil CRF++, il s'agit d'une implémentation open source des CRFs pour la segmentation et l'étiquetage des données. Pour ce qui est de la recherche de la séquence optimale des étiquettes, cet outil utilise l'algorithme de Viterbi. CRF++ utilise le même format de données utilisé avec Yamcha.

Dans l'ensemble des expérimentations, présentées dans ce chapitre, nous utilisons ces deux boîtes à outils pour les données d'entrée et nous comparons les performances entre les SVMs et les CRFs.

Voici un exemple du format d'entrée pour la phrase « ar as ttHyyaln i tmGra ann sg usggwas lli izrin ». [En Français : Ils se préparaient à ce mariage là depuis l'année dernière]:

[56] http://chasen.org/~taku/software/TinySVM/

ar	PR
as	S_P
ttHyyaln	V
i	S
tmGra	N
ann	D
sg	S
usggwas	N
lli	P
izrin	V
.	F

Figure 4.1. Un extrait à partir du corpus d'apprentissage

Sur la base de ce jeu d'étiquettes, nous explorons deux groupes d'expérimentations d'étiquetage morphosyntaxique, basées sur les SVMs et les CRFs. Dans le premier sous-ensemble d'expérimentations, nous utilisons le jeu d'étiquettes défini dans le tableau 4.1 et comme propriétés de contexte nous employons les mots qui entourent le mot à étiqueter ainsi que leurs étiquettes dans une fenêtre de +/- 2 mots. Le format des données en entrée est présenté dans la figure 4.1.

Tableau 4.2. Résultats de la 10 fois validation croisée.

Partie#	**BASE**	**SVMs**	**CRFs**
0	70,19	81,01	83,19
1	67,67	76,02	80,7
2	78,15	85,64	87,00
3	72,76	82,56	86,45
4	73,94	83,55	85,80
5	73,32	83,28	86,24
6	65,71	76,59	79,98
7	69,94	79,07	81,79
8	79,64	87,35	88,88
9	75,32	84,64	86,79
Moyenne	**72,66**	**81,97**	**84,68**

Dans le deuxième sous ensemble d'expérimentations, nous utilisons le même jeu d'étiquettes du tableau 4.1, mais nous varions les propriétés lexicales et de contexte. Après plusieurs expérimentations sur les propriétés des préfixes et suffixes en amazighe, nous avons retenu les propriétés suivantes:

1. Les propriétés lexicales n-grammes consistant en les i premiers et dernier n-grammes du jeton, avec i variant de 1 à 4 ;
2. Le contexte lexical, dans lequel il s'agit des jetons avoisinants plus leurs propriétés lexicales n-grammes ;
3. Etiquettes de contexte, qui consistent en les balises prévues pour les deux mots précédents.

Dans le deuxième sous-ensemble d'expérimentations, nous ajoutons aux premières caractéristiques les propriétés lexicales n-grammes du mot à étiqueter et des mots entourant ce mot, avec la même fenêtre de +/- 2. Les propriétés n-grammes se composent des i premiers et i derniers n-grammes caractères, avec i variant de 1 à 4 (voir Figure 4.2 ci-dessous).

ar	a	-	-	-	r	-	-	-	PR
as	a	-	-	-	s	-	-	-	S_P
ttHyyaln	t	tt	ttH	ttHy	n	ln	aln	yaln	V
i	-	-	-	-	-	-	-	-	S
tmGra	t	tm	tmG	tmGr	a	ra	Gra	mGra	N
ann	a	an	-	-	n	nn	-	-	D
sg	s	-	-	-	g	-	-	-	S
usggwas	u	us	usg	usgg	s	as	was	gwas	N
lli	l	ll	-	-	i	li	-	-	P
izrin	i	iz	izr	izri	n	in	rin	zrin	V
.	-	-	-	-	-	-	-	-	F

Figure 4.2. Extrait à partir du corpus d'apprentissage utilisant les propriétés lexicales

Dans nos premières expériences sur l'étiquetage morphosyntaxique (Outahajala et *al.*, 2011a), nous avons montré que la courbe d'apprentissage croît avec l'augmentation de la taille du corpus d'entraînement.

Tableau 4.3. Résultats de la 10 fois validation croisée en utilisant les propriétés lexicales.

Partie#	**BASE**	**SVMs (avec les propriétés lexicales et de contexte)**	**CRFs (avec les propriétés lexicales et de contexte)**
0	70.19	86.86	86.95
1	67.67	83.86	84.98
2	78.15	**91.66**	90.86
3	72.76	88.34	88.58
4	73.94	88.24	88.87
5	73.32	89.99	90.48
6	65.71	85.38	85.38
7	69.94	86.6	87.96
8	79.64	91.38	91.14
9	75.32	90.41	**91.35**
Moyenne	**72.66**	**88.27**	**88.66**

Dans cet ensemble d'expérimentation, nous avons mené et évalué nos expériences en utilisant la validation croisée en 10 parties, i.e. l'entraînement avec 90% du corpus de référence et l'utilisation de 10% pour le test, en répétant l'expérience dix fois et en prenant à chaque fois une tranche différente du corpus.

Afin d'étudier le comportement d'évolution de la courbe d'apprentissage, nous avons commencé par la génération d'un modèle initial M_{init} à partir de 60% des données étiquetées. Les 30% des données étiquetées restantes ont été subdivisées en blocs de 2k jetons. Ceci, a été effectué dans le but d'étudier la performance des modèles générés à partir des données annotées automatiquement. Le choix des données pour la génération de M_{init} n'est pas fait aléatoirement. En effet, on a effectué la validation croisée de 60% du corpus et on a pris le modèle qui a donné la meilleure précision.

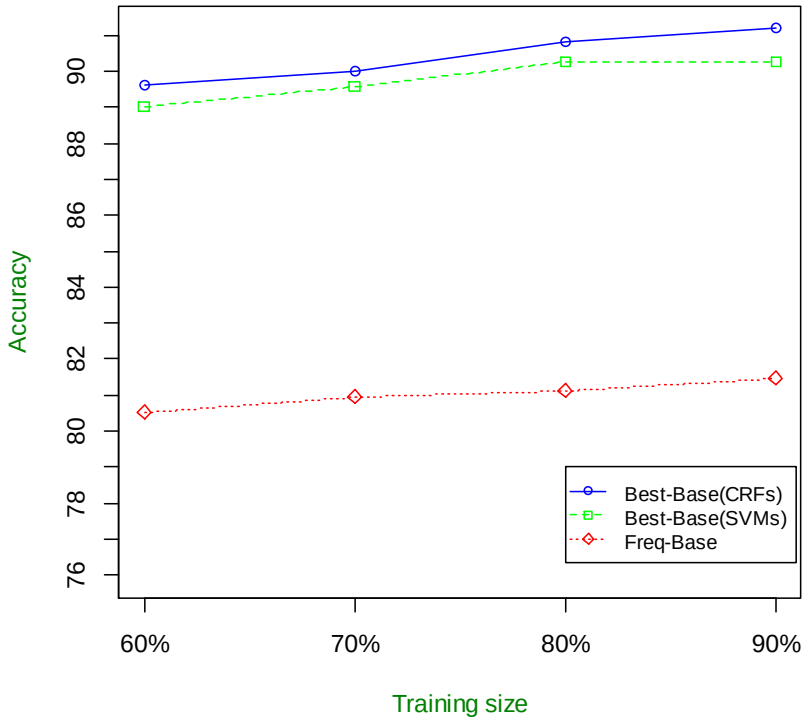

Figure 4.3. Performance de l'étiqueteur en prenant 4 sou partie du corpus

Nous avons pris 4 points du corpus avec un pas de 144 phrases entre deux points successifs. Le premier sous corpus contient 864 phrases, soit l'équivalent de 60% du corpus annoté. Les points suivants contiennent ce corpus de base plus le cumul de 144 phrases choisies aléatoirement jusqu'à atteindre la taille totale du corpus annoté manuellement, soient 1438 phrases. Les résultats des expérimentations sont résumés dans la figure 4.3. Les modèles utilisés sont basés sur les SVMs, les CRFs et Freq-Base, et ce en utilisant les propriétés lexicales et les propriétés du contexte.

La courbe d'apprentissage montre bien que la précision augmente au fur et à mesure que la taille du corpus augmente. Les résultats des SVMs et des CRFs montrent qu'ils dépassent ceux du modèle de base avec environ 8 points.

4.3. Résultats des expérimentations avec une phase de segmentation comme prétraitement

Vu l'impact positif de la segmentation, comme prétraitement dans plusieurs langues telles que pour l'arabe (Diab et *al.*, 2004), nous avons subdivisé les prépositions et les noms de parenté lorsqu'ils sont utilisés conjointement avec les pronoms personnels, et nous avons obtenu de meilleurs résultats, comme il est indiqué dans le tableau 4.4 suivant :

Tableau 4.4. Résultats de la 10 fois validation croisée après une phase de segmentation

Partie#	**BASELINE**	**SVMs**	**SVMs (avec les propriétés lexicales et de contexte)**	**CRFs**	**CRFs (avec les propriétés lexicales et de contexte)**
0	71.09	82.85	87.94	84.46	87.31
1	68.61	78.27	85.06	81.55	85.9
2	78.12	87.59	**92.58**	87.9	91.42
3	73.11	83.95	89.62	87.39	89.22
4	74.5	85.06	89.02	86.93	89.26
5	74.14	86.08	91.38	87.6	91.62
6	67.03	79.27	86.42	82.9	87.18
7	70.70	81.34	86.96	83.69	88.96
8	79.81	**88.54**	92.47	**89.32**	91.79
9	75.53	86.45	91.49	88.65	**92.14**
Moyenne	**73.26**	**83.93**	**89.29**	**86.01**	**89.48**

Pour une meilleure compréhension du comportement du système, nous avons examiné la matrice de confusion pour l'expérience qui a donné la plus grande précision. L'analyse de la matrice de confusion présente toutes les étiquettes erronées, comme le montrent les tableaux 4.5 et 4.6. En Analysant les erreurs les plus fréquentes dans ces deux matrices de confusion, nous avons constaté que les adjectifs sont souvent étiquetés comme substantifs. Cela est dû au fait que les adjectifs peuvent être utilisés comme noms. Toutefois, en ne faisant pas de distinction entre les noms et les adjectifs, nous obtenons une amélioration de 0.73 et un meilleur score, de 90.02%, en utilisant la validation croisée en 10 parties avec les SVMs. Or, en faisant la même expérience avec les CRFs, nous avons obtenu une amélioration de 0.77 et un meilleur score de 90.25% avec la même méthode de validation.

Tableau 4.5. La matrice de confusion en pourcentage en utilisant les SVMs avec les caractéristiques lexicales

	N	**A**	**V**	**P**	**D**	**S**	**C**	**AD**	**PR**	**FOC**	**F**	**I**	**R**
N	93.1	0.3	1.8	0.6	3.9	0	0	0	0	0	0.3	0	0
A	18.2	63.6	18.2	0	0	0	0	0	0	0	0	0	0
V	5.4	0.3	93	0	0	0.7	0	0	0.7	0	0	0	0
P	0.7	0	0.7	91	5.5	0.7	0.7	0	0.7	0	0	0	0
D	3.3	0	1.1	9.9	84.6	0	0	1.1	0	0	0	0	0
S	0.5	0	1	0.5	0	94	2.1	0.5	1.6	0	0	0	0
C	0	0	0	2.1	2.1	2.1	83.3	4.2	4.2	2.1	0	0	0
AD	23.2	0	7.1	1.8	1.8	3.6	1.8	60.7	0	0	0	0	0
PR	0	0	0	0	1.9	0.6	0	0.6	96.8	0	0	0	0
FOC	0	0	0	0	40	0	0	0	0	60	0	0	0
F	0	0	0	0.2	0	0	0	0	0	0	99.8	0	0
I	36.4	0	0	0	0	0	0	0	18.2	0	0	45.4	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0

Le taux d'erreur des pronoms est également élevé en raison du chevauchement important entre eux et les déterminants. Une autre source commune d'erreurs réside dans les verbes. Comme le montre le Tableau 4.6, le POS tagger basé sur les CRFs a étiqueté 4.1% des verbes comme des noms et des adjectifs, et 1.6% comme des prépositions, alors que le POS-taggeur basé sur les SVMs a étiqueté 5.7% des verbes comme des noms et des adjectifs. Pour les autres classes, le POS-tagger basé sur les SVMs a de meilleurs résultats dans l’étiquetage des pronoms, des déterminants, des adverbes, des focalisateurs et des particules.

Tableau 4.6. La matrice de confusion en pourcentage en utilisant les CRFs avec les caractéristiques lexicales.

	N	**A**	**V**	**P**	**D**	**S**	**C**	**AD**	**PR**	**FOC**	**F**	**I**	**R**
N	94.6	2.4	2.1	0.2	0.3	0.1	0.2	0	0	0	0	0	0.1
A	12.6	82.3	4.6	0	0	0	0	0	0	0.6	0	0	0
V	2.6	1.5	93.3	0	0.4	1.5	0	0.4	0.4	0	0	0	0
P	3.7	0	0	75	13.9	0.9	0.9	0.9	3.7	0.9	0	0	0
D	2.4	0	0	4.8	82.5	0	0	2.4	7.9	0	0	0	0
S	0	0	0.2	0.3	0	99	0.5	0	0	0	0	0	0
C	1.7	0	0.6	0	0.6	2.9	91.4	0	2.9	0	0	0	0
AD	23.8	0	0	9.5	0	9.5	14.3	42.9	0	0	0	0	0
PR	0	0	1.1	1.1	2.3	1.1	9.2	1.1	83.9	0	0	0	0
FOC	0	0	0	0	0	0	0	0	50	50	0	0	0
F	0	0	0	0	0	0	0	0	0	0	100	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0
R	3.1	0	1.6	1.6	0	4.7	0	0	4.7	0	6.3	0	78.1

Certaines particules ont un niveau d'ambigüité important : c'est le cas, par exemple, de la particule ∧ (d), qui a plusieurs balises possibles, en fonction du contexte.

Vu l'amélioration des résultats obtenus en utilisant cette phase de segmentation comme prétraitement à l'étiquetage, nous avons décidé de réaliser un segmenteur en morphèmes de la langue amazighe, en nous appuyant sur les méthodes d'apprentissage automatique.

4.4. Segmentation des mots amazighes

La segmentation ou bien la tokenisation est le processus de subdivision des mots en morphèmes[57], i.e. les tokens. Cette tâche est nécessaire pour la langue amazighe ; d'une part, elle permet la segmentation des textes amazighes en unités lexicales de base[58] ; d'autre part, et comme phase de prétraitement à la tâche de l'étiquetage morphosyntaxique, elle permet de donner de meilleures performances, tel qu'il a été démontré expérimentalement dans la section précédente.

[57] Nous désignons ici par morphèmes ou tokens les unités lexicales de base.

[58] Cette opération est d'une grande utilité pour le TAL de cette langue. Par exemple, elle permet d'aider à l'annotation des arbres syntaxiques, en particulier l'annotation des phrases prépositionnelles.

La segmentation des mots amazighes automatiquement n'est pas aussi facile qu'on pourrait le croire. En effet, un mot pourrait ou non être subdivisé en tokens selon le contexte. Par exemple, dans la phrase : « srs aWal xf WayyaD » (parle d'autre chose), le verbe « srs » ne peut pas être subdivisé dans ce contexte. Alors que dans la phrase : « ad srs trart aRTTal lli fllam illan » (lui rendre le crédit qu'il vous doit), le mot srs est la préposition « s » suivie du pronom personnel « s ». Dans la phrase « Grs i Wayyis nnk a Hmad unamir » (égorge ton cheval Hmad Unamir), le mot « Grs » n'est pas divisible, alors que dans la phrase : « abrid Grs » (le chemin vers lui), le mot Grs est constitué de deux tokens : la préposition « Gr » et le pronom personnel « s ». En général, ce problème d'ambigüité de segmentation se pose pour les prépositions et les noms de parenté lorsqu'ils sont suivis des pronoms personnels avec les autres catégories grammaticales, et ce selon le contexte d'utilisation.

Cette tâche de segmentation peut être vue comme un problème d'annotation partielle. Par conséquent, il est possible d'appliquer les méthodes d'apprentissage supervisé. Ramshaw et Marcus (Ramshaw & Marcus, 1995) ont introduit une façon de représenter la tâche d'analyse partielle en tâche d'annotation. Cette représentation est basée sur un jeu d'étiquettes composé de : {I, O, B}, où le I désigne Inside, O Outside et B Bondary. Plusieurs formats utilisant ces 3 étiquettes existent. Tjong et Veenstra proposent les représentations IOB, IOB2, IOE1 et IOE2 (Tjong Kim Sang & Veenstra, 1999). Le plus connu est le format IOB2, que nous allons utiliser pour réaliser notre segmenteur en tokens pour l'amazighe. Par ailleurs, Uchimoto et ses co-équipiers (Uchimoto et *al.*, 2000) ont utilisé les 5 étiquettes :{C; E; U; O; S} pour la représentation des parties dans la tâche d'extraction des entités nommées du Japonais.

Afin de réaliser cette tâche de segmentation, ainsi que la comparaison des SVMs et les CRFs pour cette tâche, nous avons entrainé deux modèles à base de séquences d'étiquettes en utilisant cinq étiquettes : {B-WORD, I-WORD, B-SUFF, I-SUFF, O}.

Le corpus utilisé pour l'entrainement de ce segmenteur a été construit semi-automatiquement à partir du corpus annoté morphosyntaxiquement. Il contient 91 376 tokens. Pour construire notre segmenteur, nous avons entrainé deux modèles de classification des séquences à savoir les SVMs et les CRFs[59]. A notre connaissance, c'est le premier type de segmenteur réalisé pour cette langue. Sur la base des expérimentations effectuées à cet égard, nous avons constaté que les performances des SVMs et des CRFs sont très comparables. Le modèle à base des SVMs a légèrement dépassé celui basé sur les CRFs (99.95% contre 99,89%).

[59] Sur les mêmes outils libres d'apprentissage, à savoir : Yamcha pour les SVMs ; et CRF[54]++, pour les CRFs.

Un travail similaire a été effectué par Mona Diab (Diab et *al.*, 2007) dans la réalisation d'un segmenteur pour l'arabe, en utilisant 10 classes. Un extrait du corpus d'apprentissage du segmenteur amazighe est présenté dans la Figure 4.4.

t	B-WORD
i	I-WORD
r	I-WORD
a	I-WORD
m	I-WORD
BRK	O
n	B-WORD
n	I-WORD
a	I-WORD
BRK	O
g	B-WORD
i	I-WORD
s	B-SUFF

Figure 4.4. Extrait du corpus d'apprentissage du segmenteur

L'espace des caractéristiques est composé des caractères voisins et leurs étiquettes dans une fenêtre de -/+4 caractères. Le choix de la fenêtre du contexte a été choisi sur la base d'expérimentations empiriques.

Tableau 4.7. Résultats de la 10 fois validation croisée de la segmentation des SVMs et des CRFs

Partie#	**SVMs**	**CRFs**
0	99.76	99.62
1	99.60	99.50
2	99.78	99.72
3	99.66	99.62
4	99.85	99.72
5	99.76	99.56
6	99.72	99.59
7	99.75	99.66
8	99.92	99.85
9	**99.95**	**99.89**
Moyenne	**99.77**	**99.67**

Comme il est montré dans le tableau 4.7, les résultats de la 10 fois validation croisée de la segmentation des SVMs et ceux des CRFs sont très comparables. Les résultats des SVMs sont meilleurs que ceux des CRFs dans toutes les parties de la validation croisée ainsi que la moyenne sur les 10 parties de la validation croisée (99.77% vs. 99.67%).

En analysant les résultats obtenus, les matrices de confusion des parties ayant les meilleurs résultats des segmenteurs, nous notons que, sur les 5 classes présentées ci-dessus, les résultats des SVMs sont meilleurs que ceux des CRFs.

Un ensemble de règles[60] a été ajouté pour compléter cette tâche de segmentation, tel que le remplacement de « dig s » et « dag s » signifiant [en lui/elle], par la préposition « dg » suivi du pronom personnel 3ème personne du singulier « s ». Toutefois, ceci pose problème car la fonction inverse n'est pas déterministe. En effet, l'union des deux morphèmes «dg» et «s», par exemple, peut donner soit « digs » ou « dags ». Ainsi, une fois que nous avons subdivisé le mot composé en ses morphèmes constituants, il n'est plus possible de calculer la forme originale après l'étiquetage grammatical en ne nous basant que sur la sortie du POS tagger. Une solution à cette question consiste à prendre la forme la plus utilisée dans le corpus en nous basant sur la fréquence des mots. Pour normaliser le texte de sortie, nous avons

[60] Techniquement, il s'agit d'un ensemble de règles implémentées sous Perl, qui prennent la sortie du segmenteur et génèrent un autre fichier de sortie prenant en considération lesdites règles.

implémenté certaines règles, telles que le remplacement de toute succession des séries des points séparés par les mêmes points mais conçus comme un seul bloc, etc.

4.5. Description du jeu d'étiquettes AMTS

Afin de capturer plus de détails morphosyntaxiques et de servir à plusieurs types d'application, les impératifs suivants ont été pris en considération dans la conception du jeu d'étiquettes AMTS :

- se baser sur la théorie linguistique de la langue amazighe, dont un aperçu est présenté dans le chapitre 1, en plus détaillé que le jeu d'étiquettes présenté dans la section 4.1 ;
- détailler avec un niveau de délicatesse acceptable ;
- prendre en considération les questions relatives à la segmentation ;
- utiliser des étiquettes mnémotechniques.

En effet, la définition d'un jeu d'étiquettes adéquat est une tâche importante dans la construction d'un POS tagueur automatique.

Le tableau 4.8 présente AMTS le jeu d'étiquettes enrichi (Outahajala et *al.*, 2012), composé de 28 étiquettes avec, pour chacun des 13 éléments du jeu d'étiquettes présenté dans (Outahajala, 2011c). Par exemple, nous avons subdivisé la classe N correspondant aux noms en trois sous classes: NN pour les noms communs, NNK pour les noms de parenté et NNP pour les noms propres. PROT représente tous les types de particules autres que les particules vocatives, de négation, d'orientation, de prédication et préverbales. ROT représente ainsi les marques mathématiques et les symboles des monnaies.

Tableau 4.8. Jeu d'étiquettes AMTS

N°	Premier jeu d'étiquettes	AMTS	Désignation
1	N	NN	Nom commun
2		NNK	Nom de parenté
3		NNP	Nom propre
4	V	VB	Verbe, forme de base
5		VBP	Verbe, participe
6	A	ADJ	Nom de qualité
7	AD	ADV	Adverbe
8	C	C	Conjonction
9	D	DT	Déterminant
10	FOC	FOC	Focalisateur
11	I	IN	Interjection
12	PR	NEG	Particule de négation
13		VOC	Vocatif
14		PRED	Particule de prédication
15		PROR	Particule d'orientation
16		PRPR	Particule préverbale
17		PROT	Autres particule
18	P	PDEM	Pronom démonstratif
19		PP	Pronom personnel
20		PPOS	Pronom possessif
21		INT	Interrogative
22		REL	Relative
23	S	S	Préposition
24	R	FW	mot étranger
25		NUM	Numéral
26		DATE	Date
27		ROT	Autres résiduels
28	F	PUNC	Ponctuation

La figure 4.5 reprend le même exemple tiré du corpus annoté, mais cette fois annoté suivant AMTS.

ar	a	-	-	-	r	-	-	-	PRPR
i	-	-	-	-	-	-	-	-	S
s	-	-	-	-	-	-	-	-	PP
ttHyyaln	t	tt	ttH	ttHy	n	ln	aln	yaln	VB
i	-	-	-	-	-	-	-	-	S
tmGra	t	tm	tmG	tmGr	a	ra	Gra	mGra	NN
ann	a	an	-	-	n	nn	-	-	DT
sg	s	-	-	-	g	-	-	-	S
usggwas	u	us	usg	usgg	s	as	was	gwas	NN
lli	l	ll	-	-	i	li	-	-	REL
izrin	i	iz	izr	izri	n	in	rin	zrin	VBP
.	-	-	-	-	-	-	-	-	PUNC

Figure 4.5. Extrait du corpus annoté suivant le jeu d'étiquettes AMTS

4.6. Expérimentations d'étiquetage basées sur AMTS

Nous avons utilisé les deux techniques d'apprentissage supervisé, à savoir les SVMs et les CRFs, pour l'entrainement de l'étiqueteur morphosyntaxique sur la base du corpus annoté manuellement et du jeu d'étiquettes AMTS. Dans ce qui suit, nous présenterons les résultats et discuterons et analyserons les erreurs.

4.6.1. Expérimentations et résultats

Le tableau 4.9 résume les résultats des expérimentations de l'entrainement de l'étiqueteur morphosyntaxique amazighe, en se basant sur AMTS.

Tableau 4.9. Résultats de la 10 fois validation croisée utilisant la segmentation comme phase de prétraitement

Partie#	BASELINE	SVMs	CRFs
0	79.70	85.12	86.02
1	77.36	83.25	84.28
2	84.03	**90.75**	89.48
3	81.00	87.89	88.2
4	80.11	88.36	89.35
5	81.47	90.24	**91.18**
6	77.29	83.18	84.27
7	76.95	83.84	85.32
8	84.22	89.33	90.31
9	86.45	89.20	91.12
AVG	**80.85**	**87.11**	**87.95**

La figure 4.6. montre les résultats d'entrainement des modèles, à base de fréquences, des SVMs et des CRFs, en utilisant les corpus annotés manuellement.

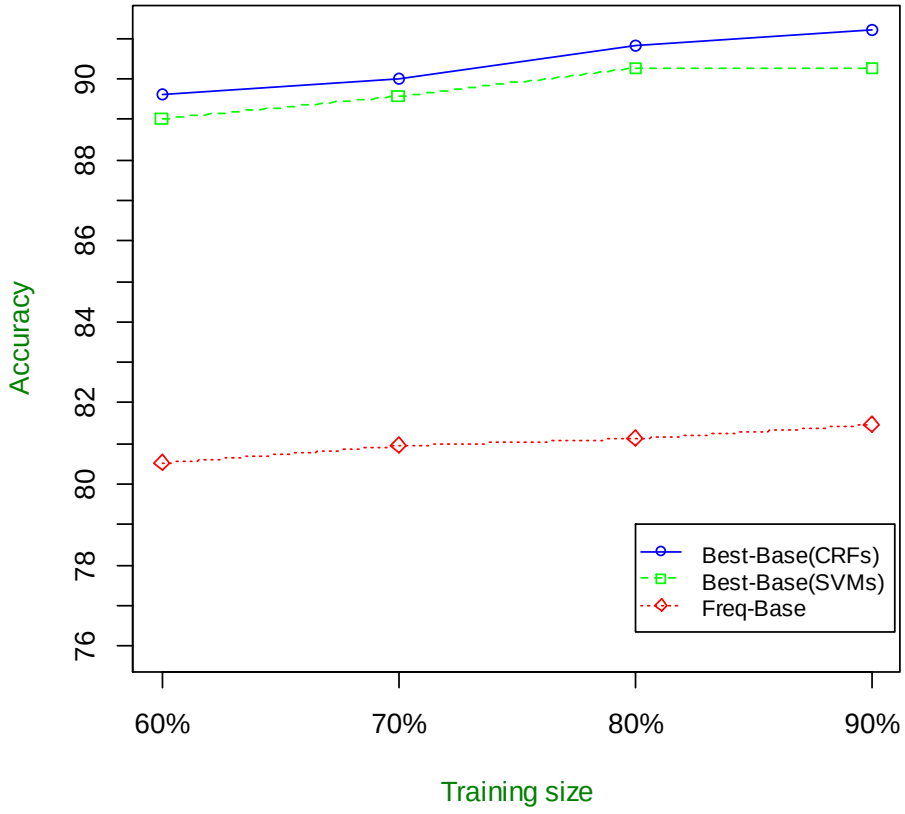

Figure 4.6. Evolution des performances des tagueurs

Comme il est mentionné dans la figure 4.6, la courbe d'apprentissage évolue en fonction de la taille du corpus. Les résultats du modèle à des fréquences sont inférieurs à ceux des SVMs et des CRFs avec au moins 7.5 points.

Nous avons commencé l'apprentissage du modèle avec M_{init}, entrainé sur la base de 60% de la taille du corpus annoté manuellement. A chaque itération, nous ajoutons à ce corpus de base 10% dudit corpus annoté manuellement. Les différences de précision entre les modèles entrainés sur la base de 60% et 90% du corpus sont de l'ordre de 1.55% pour les SVMs et 1.23% et pour les CRFs.

4.6.2. Discussion des résultats et analyse des erreurs

Sur la base du jeu d'étiquettes AMTS, les résultats des CRFs ont dépassé ceux des SVMs. En effet, la précision des CRFs dans les parties de la 10 fois validation croisée a dépassé celle des SVMs (91.18% vs. 90.75%). Aussi, la moyenne de précision des CRFs sur les 10 parties de la validation croisée a dépassé celle des SVMs (87.95% vs. 87.11%).

Ces résultats sont très prometteurs, vu que nous avons utilisé un corpus d'environ ~20k mots. Comparés aux résultats des expérimentations préliminaires basés sur 13 étiquettes, nous avons plus que doublé le jeu d'étiquettes. En contre partie, nous avons perdu 2.18% en moyenne sur les 10 parties de la validation croisée pour les SVMs et 1.53% pour les CRFs.

En comparaison avec l'ancien jeu d'étiquettes et en utilisant la matrice de confusion de la partie dans laquelle les meilleurs résultats ont été obtenus, la précision de la majorité des classes a augmenté. Nous avons obtenu une précision de 96.24% contre 94% pour les prépositions, 60.70% contre 65.38% pour les adverbes, 87.02% contre 84.6% pour les déterminants, 75% contre 60% pour les focalisateurs, et 100% contre 45% pour les interjections. En revanche, la précision des adjectifs et des conjonctions a diminué dans le nouveau jeu d'étiquettes.

En ce qui concerne les classes que nous avons subdivisées en plusieurs sous classes, telles que la classe N représentant les noms (que nous avons subdivisée en NN pour les noms communs, NNK pour les noms de parenté et NNP pour les noms propres), nous avons obtenu une précision de 95.15% contre 94.6% pour les noms en général dans l'ancien jeu d'étiquettes. Néanmoins, la précision des noms propres ne dépasse pas 54.16% ; cela est dû essentiellement à leur rareté dans le corpus d'entrainement. En ce qui concerne la classe V

correspondant aux verbes (que nous avons subdivisée en deux sous classes VB pour la forme de base et VBP pour les participes), la précision de la classe VB atteint 94.22% contre 93.3% dans le jeu d'étiquettes précédent.

L'analyse des ensembles d'entrainement et de test a montré qu'il est difficile de distinguer certaines classes, telles que les adjectifs, les noms communs et les participes. Nous avons noté également que les mots inconnus, ne figurant pas dans le corpus de test, constituent une source importante d'erreurs.

4.7. Synthèse

L'annotation morphosyntaxique permet l'augmentation des informations sur le texte donné. Ces information sont utiles pour la majorité des tâches du TAL telles que la traduction automatique, la reconnaissance des entités nommées, l'extraction de l'information, etc.

Dans ce chapitre, nous avons mené des expérimentations sur l'annotation morphosyntaxique automatique en nous basant sur un premier jeu d'étiquettes contenant 15 étiquettes, qui représente les éléments de base du jeu d'étiquettes présenté dans le chapitre 2, plus les deux étiquettes S_P et N_P (désignant respectivement les prépositions et les noms de parenté lorsqu'ils sont suivis des pronoms personnels). Ensuite, nous avons démontré expérimentalement l'utilité d'une phase de segmentation comme prétraitement. Enfin, nous avons présenté AMTS, jeu d'étiquettes enrichi pour la langue amazighe (il est composé de 28 étiquettes) et avons procédé aux expérimentations relatives à l'annotation morphosyntaxique, en utilisant les CRFs et les SVMs.

Utilisant AMTS, les résultats des CRFs et des SVMs sont très comparables. En effet, les résultats des CRFs dépassent de très peu ceux des SVMs. La précision des CRFs dans les parties de la 10 fois validation croisée a dépassé celle des SVMs (91.18% vs. 90.75%). Aussi, la moyenne de précision des CRFs sur les 10 parties de la validation croisée ont dépassé ceux des SVMs (89.48% vs. 89.29%). Ces résultats sont très prometteurs, vu que nous avons utilisé un corpus de petite taille, annoté manuellement.

Dans le chapitre suivant, et vu que l'obtention des données annotées est très coûteuse, nous expérimenterons l'usage des techniques d'apprentissage semi-supervisé pour l'amélioration des résultats de notre étiqueteur. Aussi, nous examinerons l'usage des lexiques externes.

CHAPITRE 5:

UTILISATION DES RESSOURCES EXTERNES POUR L'AMELIORATION DES RESULTATS DE L'ETIQUETEUR

5.1. Introduction

L’étiquetage des données est coûteux et, dans certains cas, très difficile à réaliser. Il en est ainsi notamment de l’étiquetage morphosyntaxique, de l'étiquetage de la parole, de l'étiquetage des images, etc. Bien qu’elles nécessitent du temps pour leur prétraitement, les données non annotées sont plus faciles à collecter pour les langues de moindre diffusion. C’est le cas de l’amazighe, resté jusqu'à un passé proche essentiellement une langue orale et dans la version standard devrait être l’aménagement de plusieurs dialectes.

L'apprentissage semi-supervisé est une sous branche du domaine de l'apprentissage machine et de l'intelligence artificielle, plus généralement. Il consiste en l’utilisation des données étiquetées ainsi que des données non étiquetées. L’objectif est de construire un classificateur plus précis, en exploitant les données non étiquetées disponibles.

Dans ce chapitre, nous allons dresser l’état de l’art des méthodes d’apprentissage semi-supervisé existantes, en particulier les algorithmes d’auto apprentissage et l’entrainement mutuel. Ensuite, nous présenterons un corpus de textes amazighes bruts d’environ un quart de millions de tokens, que nous utiliserons conjointement avec les données étiquetées mentionnées dans les chapitres précédents. Puis, nous présenterons les expériences d’auto-apprentissage de l’étiqueteur morphosyntaxique amazighe, en variant les critères de choix des données sélectionnées. Enfin, nous expérimenterons l’utilisation des lexiques dans l’amélioration de la performance de l’annotateur.

5.2. Etat de l’art des méthodes semi-supervisées utilisées en TAL

Parmi les algorithmes les plus utilisés dans les problèmes de la classification qui utilise l’apprentissage semi-supervisé, on trouve l’auto-apprentissage et le co-apprentissage.

L’auto-apprentissage (self-training)

Il consiste en l’auto entrainement d’un classificateur initial formé avec quelques données étiquetées et, à chaque itération, on augmente les données étiquetées avec les nouvelles données étiquetées.

Prendre un classificateur initial et ses sorties ayant un bon score et les ajouter au corpus d’entrainement a été proposé par Hindle et Rooth (Hindle & Routh ; 1993) et Hearst (1992). Hindle et Rooth ont utilisé cette technique dans la résolution du problème d’attachement des phases prépositionnelles. Hearst, elle, traite le problème de désambigüisation des sens. Dans les deux problèmes, les auteurs augmentent les statistiques utilisant les données étiquetées où

la confiance des règles de décision est grande pour améliorer les résultats. Yarowsky (1995) est le premier à itérer le processus d'annotation et à refaire l'entrainement du modèle. Le problème majeur que pose cet algorithme est la qualité des données à choisir à chaque itération, vu que les données en sortie du classificateur à une itération donnée sont bruitées.

Cette technique a été appliquée à de nombreux domaines du TAL. Par exemple, Yarowsky (1995) l'a utilisée pour la désambiguïsation des sens des mots ; Zavrel et Daelemans (Zavrel & Daelemans, 2000), Cucerzan et Yarowsky (Cucerzan & Yarowsky, 2002) l'ont également utilisée dans l'annotation morphosyntaxique.

Le co-apprentissage (co-training)

Le co-apprentissage (Blum & Mitchell, 1998 ; Dasgupta, 2002 ; Abney, 2002) est une méthode faiblement supervisée pour l'amorçage de deux ou plusieurs modèles à partir d'un corpus annoté de petite taille. Il commence par l'entrainement d'un classificateur en utilisant la première vue de données étiquetées, et le second classificateur à l'aide d'un second point de vue des données étiquetées. On procède ainsi de suite pour les classificateurs, s'il y en a plus de deux. Les classificateurs sont utilisés pour étiqueter de nouvelles données. Les données étiquetées les plus confiantes sont conservées et ajoutées au corpus d'entrainement. Le processus est réitéré jusqu'à ce qu'un critère d'arrêt soit atteint.

L'entrainement mutuel et ses différentes variantes ont été appliqués à plusieurs problèmes du TAL, notamment la reconnaissance des entités nommées (Collins & Singer, 1995), la reconnaissance des phrases nominales (Pierce & Cardie, 2001), l'analyse syntaxique (Sarkar, 2001 ; Steedman et *al.*, 2003) et l'étiquetage morphosyntaxique (Clark & Curran, 2003 ; Søgaard, 2010).

Plusieurs autres méthodes existent pour effectuer l'apprentissage semi-supervisé, dont les graphes, où les sommets représentent les données étiquetées et non étiquetées. Les arrêtes avec des poids reproduisent la similarité avec les données étiquetées (Blum & Chawla, 2001; Belkin et *al.*, 2204; Chapelle & Zien, 2005). La transduction est une technique utilisée pour entraîner de façon semi-supervisée les machines à vecteurs support. D'autres méthodes existent, comme les méthodes basées sur la régularisation de l'information (Joachims, 2003), les arbres (Kemp et *al.*, 2003), la minimisation de l'entropie (Grandvalet & Bengio, 2004), etc.

Afin d'étudier l'impact de l'utilisation des données étiquetées automatiquement, nous avons implémenté l'algorithme d'auto-apprentissage pour la génération d'un étiqueteur plus précis, en utilisant plusieurs variantes de la mesure de confiance du modèle du langage.

5.3. Présentation du corpus non annoté et des modèles de références

Nous avons utilisé le self-training pour le POS tagging de l'amazighe. Ces expérimentations sont utiles pour les langues peu dotées en ressources annotées. Elles peuvent également être vues comme un moyen d'exploration des aspects de l'auto-apprentissage.

Dans cette section, nous présenterons le corpus brut collecté et ses caractéristiques, les modèles de références et les caractéristiques utilisées dans les expérimentations.

5.3.1. Corpus brut

Le corpus utilisé dans nos expérimentations est tiré de quelques romans amazighes et de données collectées par le Linguistic Data Consortium, en collaboration avec l'IRCAM (Cieri et Liberman, 2008), auxquelles ont été ajoutées certains textes traduits en amazighe par l'IRCAM. Le corpus ainsi collecté a subi multiples prétraitements, à savoir:

- la correction de certaines erreurs fréquentes, telles que le mauvais placement du e muet "ⵉ ". En effet, ce caractère s'écrit dans les deux cas suivants :

  - la succession de plus de trois consonnes radicales identiques à l'intérieur du même mot, comme ⵛⵛⵛ dans ⵣⵛⵛⵉⵛ (zmmem) "écrire" et ⵜⵜⵜ dans ⵜⵉⵜⵜⵙ (tettu) "elle a oublié" ;

  - les radicaux verbaux se terminant par deux consonnes, par exemple ⵛⵍⵉⵍ (mlel) "être blanc",

- pour les textes rédigés en glyphes tifinaghes mais usant des caractères latins, et afin de corriger l'existence erronée du caractère "^", due à une erreur lors de la saisie des lettres emphatiques « ⴽ ̈ » et «ⵅ ̈ » ;

- la translittération de textes écrits dans les différents systèmes d'écriture vers le système d'écriture présenté dans le chapitre 2 ;

- la révision des textes collectés selon les règles orthographiques adoptées par l'IRCAM. Dans ce sens, un programme de segmentation en mots selon les règles de l'IRCAM a été développé[61]. Ce programme utilise une base de données de plus de 41 000 mots amazighes distincts. Cette base de données a été construite en utilisant tous les textes amazighes collectés, dont la segmentation est faite selon les règles orthographiques adoptées au Maroc et qui ont été révisées par l'IRCAM. A ces derniers ont été ajoutés plusieurs autres lexiques tels que (El Gholb, 2011; Agnaou, 2011; Sghir, 2014) ;

[61] www.outamed.com/downloads/word_bondary.pl

- la segmentation, en utilisant le segmenteur amazighe réalisé à cet effet (Outahajala et *al.*, 2013), présenté dans le chapitre 4 ;

Le nombre total des mots du corpus recueilli, révisé et corrigé est de 218 073 mots. Les cinq mots les plus fréquents dans le corpus collecté sont: n, d, ad, g et ur. En segmentant le corpus susmentionné, le nombre total des jetons à partir du corpus recueilli est de 225 901. Ce corpus est disponible gratuitement en téléchargement[62].

Pour calculer la qualité et la complexité de lecture du corpus amazighe brut et segmenté, les trois mesures suivantes, définies dans (Makagonov & Alexandrov, 1999) ont été effectuées :

1- $La\ complexité = C.\log(M)$

Dans cette équation, C est le nombre moyen des caractères dans un mot et M le nombre moyen des mots par phrase ;

2- $La\ variété = n/\log(N)$

Où n est le cardinal du vocabulaire du corpus et N le nombre total des mots.

3- Exactitude de la distribution des fréquences

L'exactitude de la distribution du corpus segmenté, basée sur « le principe du moindre effort » (Zipf, 1949) est présentée dans la figure 5.1. La loi de Zipf stipule que, dans tout corpus, la fréquence d'utilisation de toute forme de mot est inversement proportionnelle à son rang dans le tableau des fréquences.

Le tableau 5.1 donne les résultats des mesures de complexité et de variété des corpus brut et segmenté.

Tableau 5.1. Résultats obtenus pour la complexité et la variété du corpus

Corpus	**Complexité**	**Variété**
Corpus brut	8.60	1950.53
Corpus segmenté	8.38	1884.35

[62] www.outamed.com/downloads/corpusB

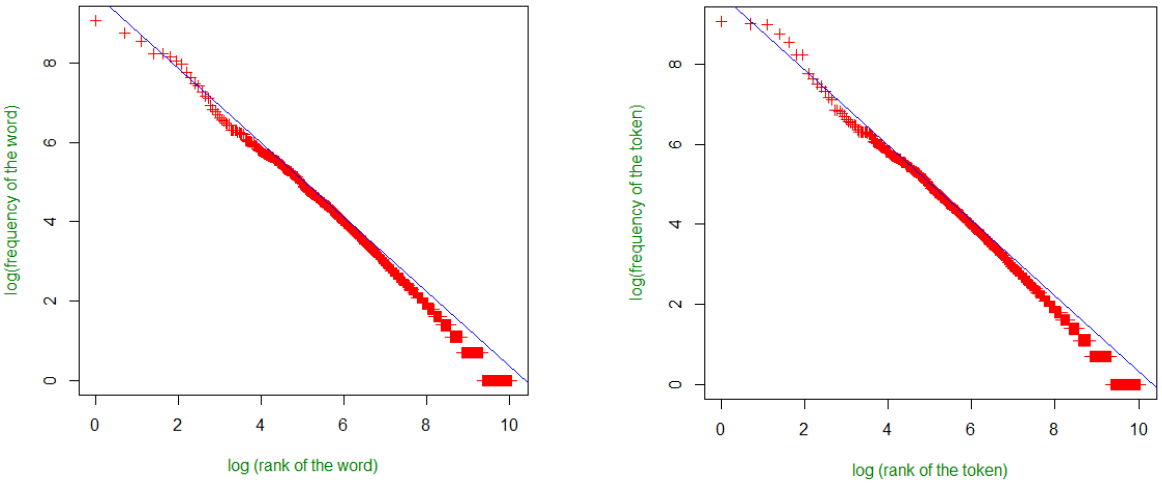

Figure 5.1. Distribution des fréquences des mots et des jetons et la courbe Zipf idéale.

Dans la suite des expérimentations, l'ensemble des données collectées prétraitées et segmentées est noté *U*.

5.3.2. Les modèles de références

Nous avons choisi d'utiliser deux lignes de base comme références dans ces expériences. En outre, nous avons utilisé le dernier jeu d'étiquettes disponible, composé de 28 étiquettes et les CRFs comme modèles de classification des séquences pour la génération des modèles de classification. Les modèles de référence utilisés comme lignes de base sont:

1 – le modèle de référence basé sur la fréquence des mots (Freq-Base.): il s'agit d'un algorithme basé sur la fréquence des étiquettes des mots. L'étiquette prévue pour un mot est l'étiquette la plus fréquente qui a été associée dans les données de l'entrainement. Cette base ignore totalement le contexte environnant et résout les cas ambigüs en utilisant uniquement les fréquences des étiquettes ;

2 – le modèle de référence du meilleur cas (Best-Base.): pour la création de ce modèle, on a commencé par la génération d'un modèle initial M_{init} à partir de 60% des données étiquetées manuellement. Les 30% de données étiquetées restantes ont été subdivisées en blocs de 2k jetons. Cette subdivision est faite afin d'étudier la performance des modèles générés à partir des données annotées automatiquement. Le choix des données pour la génération de M_{init} n'est pas aléatoire. En effet, on a effectué la validation croisée de 60% du corpus annoté manuellement et on a pris le modèle ayant la meilleure précision, parmi les modèles des 10 parties de la validation croisée.

Dans cette partie, nous avons choisi de nous concentrer sur l'applicabilité et les résultats de l'application de l'auto-apprentissage pour la langue amazighe. Les propriétés utilisées sont les mêmes que celles employées dans le chapitre 4.
Dans la suite de ces expérimentations, le corpus du test utilisé est le même que celui utilisé dans le chapitre précédent.

5.4. Expérimentation et résultats

Le but de ces expérimentions est d'évaluer le critère de confiance dans la sélection des phrases pour l'auto apprentissage de notre modèle. Nous partons de l'hypothèse que notre modèle apprend plus quand la mesure de confiance est élevée. Pour évaluer notre approche, nous commençons par un modèle initial M_{init} généré à partir de 60% des données étiquetées.

Pour ce faire, nous étudierons la corrélation entre la mesure de confiance et la probabilité d'obtenir un étiquetage correct. C'est l'estimation des chances d'assigner une étiquette correcte à un mot automatiquement quand la mesure de confiance donnée au mot par le système est élevée. Nous croyons que cette estimation est importante car, lorsque la corrélation observée tend vers 1, la probabilité des données sélectionnées tend à améliorer le système et, lorsque cette probabilité tend vers 0.5, l'amélioration est aléatoire. D'un point de vue de filtrage du bruit, on peut dire que les deux termes en question permettent de déterminer s'il est possible ou non de filtrer le bruit, en se basant sur la probabilité donnée par le système.

Afin d'obtenir l'information requise, nous avons automatiquement annoté 10% du corpus de test utilisant un modèle initial M_{init}. Nous n'avons pas utilisé intentionnellement le corpus de test pour calculer la corrélation entre la mesure de confiance et la probabilité d'obtenir un étiquetage correct. Les étiquettes obtenues ont servi comme données pour le calcul de ladite corrélation.

Dans la Figure 5.2, nous présentons un dessin des points de données montrant qu'il y a une corrélation de 0.78 entre la mesure de confiance du système et la probabilité d'avoir pour cette mesure de confiance une étiquette correcte. Ainsi, dans cette même Figure 5.2, nous pouvons voir une claire régression.

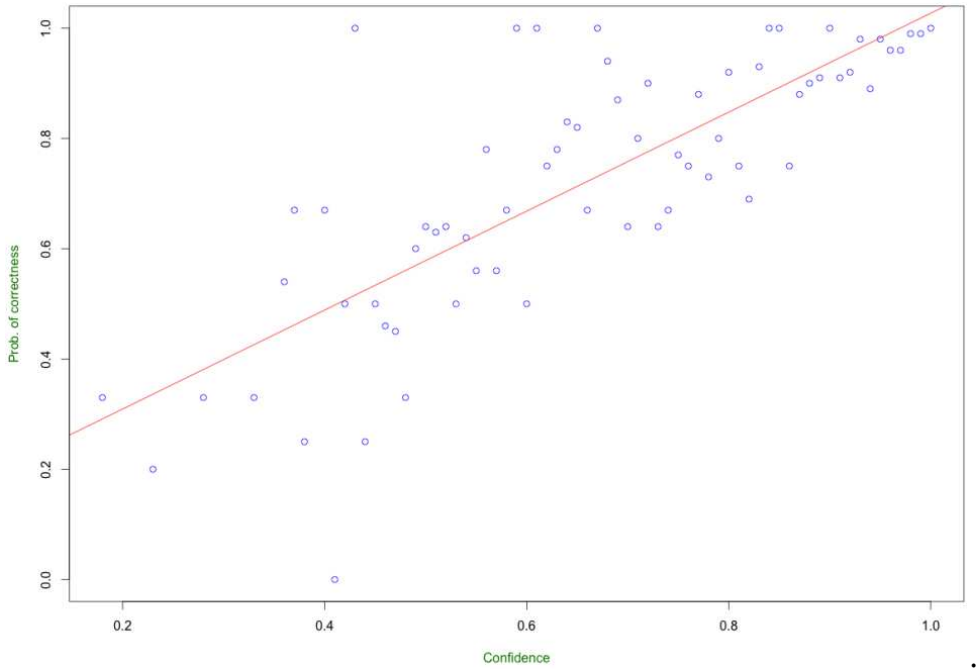

Figure 5.2. Nuage de points du système de confiance et la probabilité d'avoir une étiquette correcte

Calculer une statistique similaire pour la confiance de la phrase a été bloqué par l'asymétrie de la distribution des phrases annotées correctement, contre les phrases annotées de façon fausse : i.e. le nombre des phrases annotées correctement, où tous les mots de la phrase sont correctement annotés, est très petit par rapport au nombre des phrases annotées incorrectement (où au moins un mot de la phrase a une étiquette erronée). Néanmoins, les résultats obtenus sont assez encourageants pour effectuer plus d'expérimentations, en utilisant la mesure de confiance des mots et la mesure de confiance des phrases comme critères de sélection, sur la base de l'algorithme d'auto-apprentissage dont nous présentons, ci-dessous, un ensemble d'expérimentations.

5.4.1. Sélection des données pour l'algorithme d'auto apprentissage

Dans cette sous section, nous étudierons l'utilité de la mesure de confiance du système dans le choix des données à utiliser pour l'auto-apprentissage.

Expérimentation I: utilisation de la mesure de confiance du système pour les mots dans la sélection des données

Pour étudier l'utilité de la mesure de confiance du système pour les mots dans la sélection des données, on a effectué des expérimentations utilisant le modèle initial M_{init} et les données brutes présentées dans la section 5.4.

Les données non étiquetées ont été annotées automatiquement et on a gardé 1 295 phrases du corpus de référence, soit l'équivalent de 90% des données annotées manuellement. Cela nous a permis de comparer les résultats de l'auto apprentissage avec ceux obtenus précédemment.

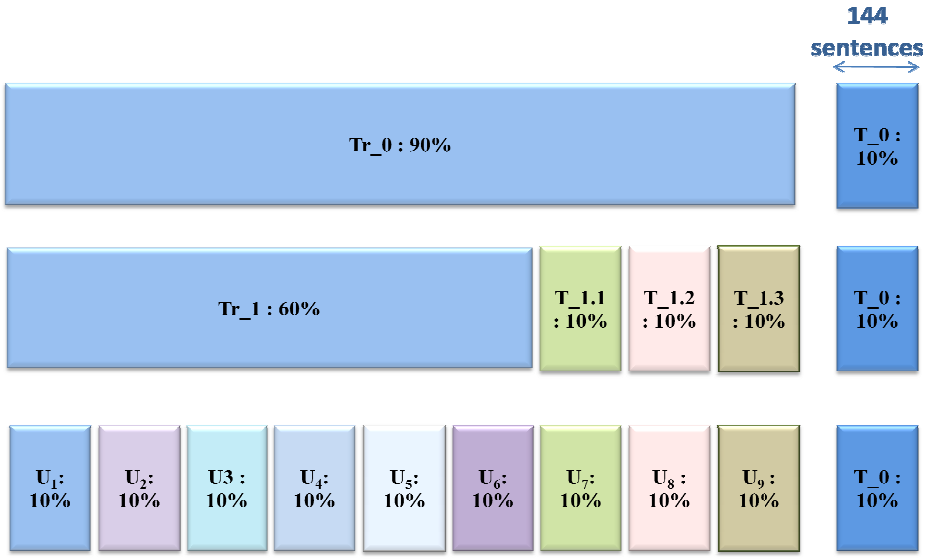

Figure 5.3. Division des données pour les expérimentations préliminaires de l'auto apprentissage

Dans cette expérimentation, le critère de sélection est basé sur la mesure de la confiance donnée par le système et qui n'est d'autre que la probabilité marginale. Ce corpus a été subdivisé en 9 parties : U_1, U_2, U_3, U_4, U_5, U_6, U_7, U_8, et U_9. Chacune des neuf parties U_i, avec i variant de 1 à 9, contient exactement 144 phrases (soit l'équivalent de 10% du nombre total des phrases du corpus annoté manuellement). La subdivision du corpus est présentée dans la Figure 5.3.

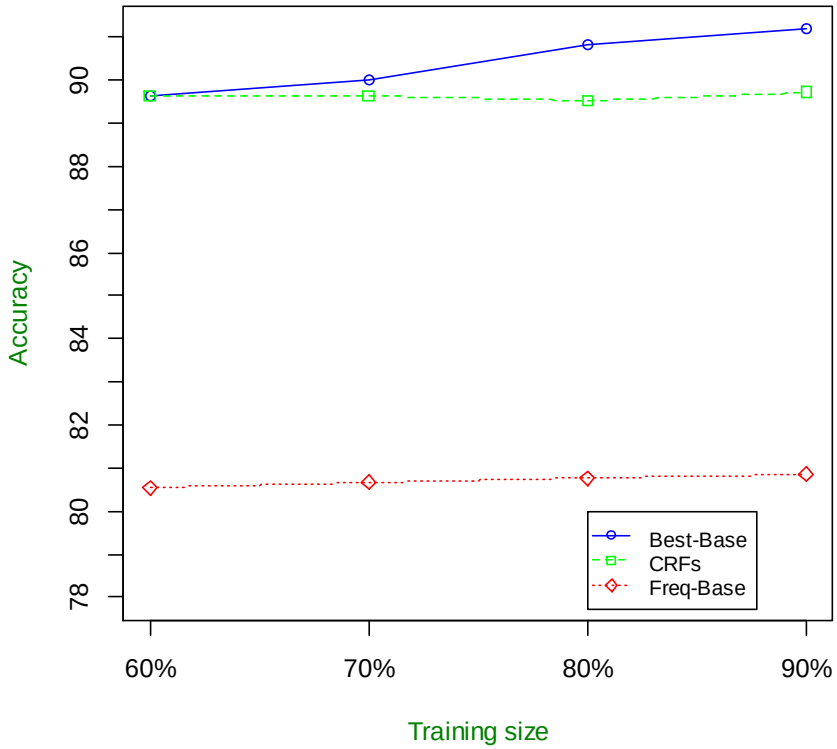

Figure 5.4. Apprentissage automatique utilisant les données filtrées selon la confiance du mot comme moyen de sélection.

Les résultats des expérimentations d'entrainement avec des données bruitées sont présentés dans la figure 5.4. Lorsqu'on utilise la confiance d'un mot donné par le système comme mesure de sélection pour l'apprentissage, les résultats obtenus montrent une légère amélioration de 1% dans la réduction du taux d'erreur.

Expérimentation II: utilisation de la confiance de la phrase pour la sélection des données

Dans cet ensemble d'expérimentations, les données non étiquetées ont été annotées automatiquement à partir de U et nous n'en avons gardé que les meilleures 1 295 phrases, en nous basant sur la mesure de confiance comme critère de sélection des phrases (voir la figure 5.3).

Lors de la création des modèles, nous avons commencé par un modèle initial $M_{init,}$ entrainé sur la base de 60% des données annotées manuellement. Ensuite, à chaque itération de l'expérimentation, nous ajoutons 10% du corpus sélectionné et annoté automatiquement.

La base de sélection des phrases est la probabilité marginale de la phrase. Les résultats de cette expérimentation sont montrés dans la figure 5.5. Les 1 295 meilleures phrases, sélectionnées selon la mesure de confiance donnée aux phrases, ont été subdivisées en les parties U_1, U_2…….U_9. Le taux de réduction d'erreur obtenu selon ce type de sélection des phrases est de l'ordre de 1%.

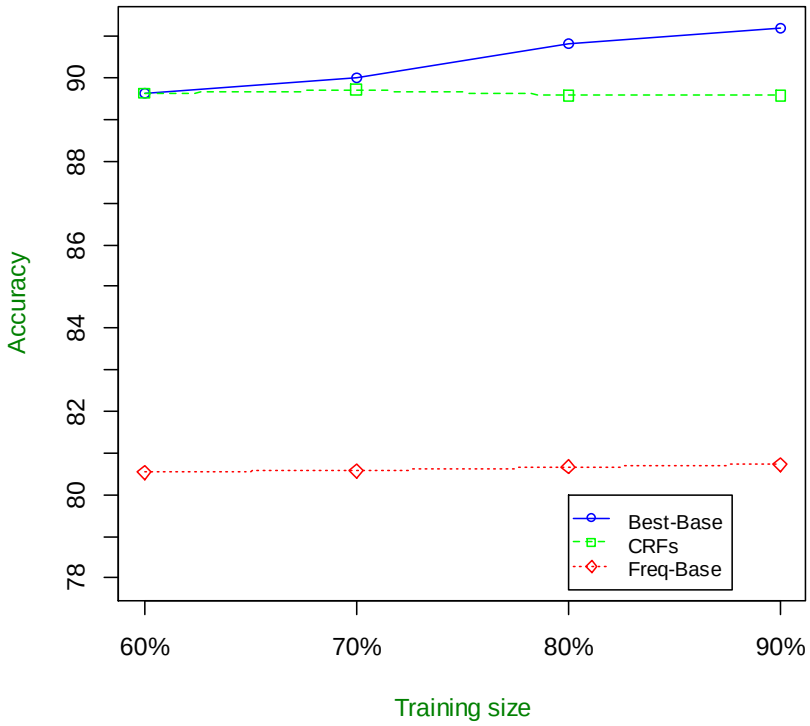

Figure 5.5. Apprentissage automatique utilisant les données filtrées selon la mesure de confiance des phrases

Expérimentation III: choix aléatoire des données pour l'apprentissage

Pour étudier l'effet qu'implique l'ignorance de la confiance et voir si ce critère est important ou non, nous avons conduit une expérimentation où nous commençons par le modèle initial M_{init} et, à chaque itération de l'opération d'apprentissage, nous ajoutons 144 phrases de U annotées automatiquement par M_{init} et choisies aléatoirement parmi les 1 295 phrases présélectionnées de U.

Telles que montrées dans la courbe CRF-R, qui représente le modèle généré sur la base des données choisies aléatoirement (Figure 5.6), les précisions obtenues de l'apprentissage à partir des données choisies aléatoirement sont inférieures à celles qui se basent sur la sélection des données en utilisant la mesure de confiance. Ceci confirme le fait que cette mesure est utile dans la sélection des phrases au niveau de l'auto apprentissage de notre étiqueteur morphosyntaxique.

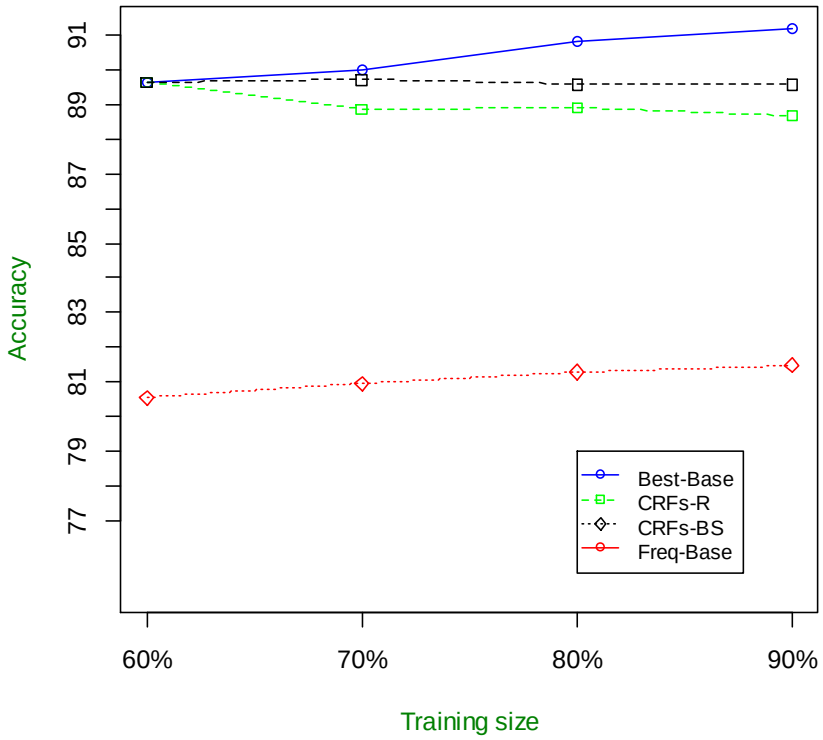

Figure 5.6. Apprentissage à partir de données sélectionnées aléatoirement en comparaison avec les autres moyens de sélection

Afin de vérifier l'hypothèse selon laquelle le bruit de l'auto apprentissage n'empêche pas la réduction du taux d'erreur lors de l'entrainement de notre modèle, nous avons conduit l'expérimentation suivante :

- génération de M_{init} à partir des parties U_1,$U_{2...}$,U_6 constituant 60% de la taille du corpus de référence ;
- ajout, à chaque itération de l'apprentissage, de 144 phrases au corpus d'apprentissage, jusqu'à ce que le corpus d'apprentissage atteigne l'équivalent de 90% du corpus de référence.

Les résultats de l'expérimentation montrent qu'il y a une augmentation de précision de 5.9% entre M_{init} le modèle appris en utilisant U_1, U_2, …U_9. La figure suivante résume l'évolution des résultats :

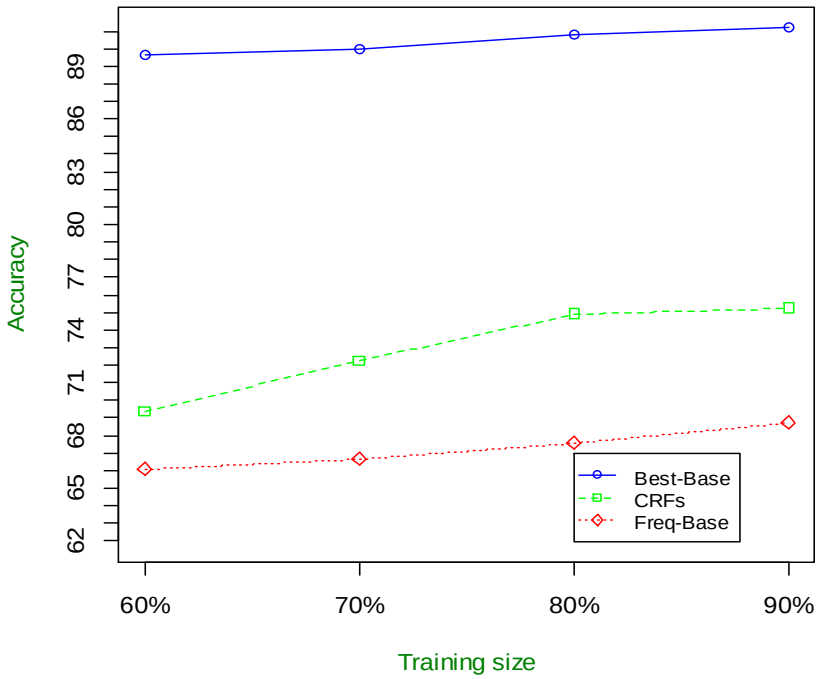

Figure 5.7. Apprentissage sur la base des données annotées automatiquement

5.4.2. Utilisation de la propriété fréquences des OOV

Pour étudier l'impact des mots hors vocabulaire sur notre système, nous avons étudié la performance par rapport au taux des OOV. Le tableau 5.2 résume les résultats des modèles basés sur les fréquences et l'apprentissage ainsi que les statistiques des taux des OOV.

Nous avons remarqué qu'au fur et à mesure que la performance augmente, cette dernière devient difficile à améliorer. Néanmoins, la différence d'amélioration ne diminue pas de façon régulière ; elle fluctue légèrement. Par exemple, le taux d'amélioration (0.81) entre 70% et 80% du corpus est supérieur au taux d'amélioration entre 60% et 70% (0.66) lorsqu'on fait l'entrainement des modèles à l'aide des données annotées manuellement. Parmi les raisons de cette amélioration, on trouve la diminution du taux des OOVs dans le corpus d'entrainement. En effet, le taux des OOVs dans le corpus contenant 80% du corpus de référence est de 11%, contre un taux des OOVs de 15% dans le corpus contenant 60% du corpus de référence.

Tableau 5.2. Mots hors vocabulaire par rapport à la performance

Corpus d'apprentissage	**Précision basée sur le modèle les fréquences**	**Précision basée sur le modèle d'apprentissage**	**Taux des OOV**
60%	80.53	89.63	15%
70%	80.95	90.00	13%
80%	81.14	90.81	11%
90%	81.47	91.18	10%

En analysant les fichiers de sorties de nos étiqueteurs, nous remarquons que les étiquettes renvoyées par le système pour les mots hors vocabulaire sont parfois correctes. Dans ce sens, nous avons conduit des expérimentations sur le caractère informatif, en recherchant les mots hors vocabulaire avec une fréquence supérieure à 1. Ensuite, pour chacune de ces instances nous identifions l'étiquette attribuée par le système la plus fréquente et nous l'affectons dans toutes les phrases contenant au mot hors vocabulaire en question. Puis, nous ajoutons ces dites phrases au corpus d'entrainement et nous refont l'entrainement (Algorithme1). Les résultats obtenus montrent une réduction du taux d'erreur de 1.37%.

Dans l'algorithme 1, identifier l'étiquette la plus fréquente consiste en la recherche de l'étiquette donnée au mot hors vocabulaire parmi les phrases auxquelles ce mot appartient. Ensuite, cette étiquette est assignée au mot hors vocabulaire dans toutes les phrases auxquelles il appartient.

La mesure de confiance et le caractère informatif des OOVs, ainsi que les techniques d'apprentissage semi-supervisé présentent d'intéressantes pistes de recherche. Dans les expérimentations suivantes, nous allons les utiliser conjointement afin d'améliorer les performances de l'étiqueteur morphosyntaxique.

Algorithme 1. *Informativeness(L_0, U)*

```
1   L0 is labeled data, T0 test file, U is Unlabeled data
2   Minit<-- train(L0)
3   For each OOVi in OOV
4    Automatically_tag(U, Model)
5    Tf = Identify most frequent tag to OOVi
6    STi= Assign Tf to select(U, OOVi))
7    U<-- U- STi
8    L <--L0 + STi
9    Modeli<--train (L)
10   Test(T0)
11  End For each
12  Return Model
13  Function Select (U, Word)
14    For each sentence of U
15        If (sentence contains Word) then
16            selected_sentences = selected_sentences + sentence
17        End If
18   End For each
19  Return selected_sentences
```

5.5. Expérimentations de l'utilisation du caractère informatif et de la mesure de confiance comme critères pour l'auto-apprentissage

Vu que la création de données étiquetées est une tâche difficile alors que l'obtention des données brutes est moins coûteuse, même si le prétraitement des langues peu dotées nécessitent souvent plus du temps, nous avons décidé d'utiliser les techniques d'apprentissage semi-supervisé dans le but de trouver un classificateur plus précis qui lie les entrées aux étiquettes, en exploitant le corpus des textes brutes *U*.

Compte tenu de l'impact positif de l'utilisation de la mesure de confiance dans le choix des données lorsqu'on utilise les données étiquetées automatiquement, on a opté pour l'implémentation de l'algorithme d'auto apprentissage pour la génération d'un étiqueteur plus précis, utilisant les variantes de la mesure de confiance de CRF++ et une version adaptée de l'algorithme d'auto apprentissage.

5.5.1. Algorithme d'auto apprentissage

L'algorithme consiste en l'entrainement d'un premier classificateur M_{init} avec un petit corpus de données étiquetées. Ce premier corpus est augmenté par l'ajout des données annotées et le réentrainement de nouveaux classificateurs. Ce processus est réitéré jusqu'à atteindre un point vérifiant un critère d'arrêt. Son algorithme basic est présenté dans l'algorithme 2. La fonction de sélection prend en entrée un ensemble de données avec leur mesure de confiance, et retourne en sortie la phrase ayant la plus grande mesure de confiance. La phrase sélectionnée est ensuite ajoutée au corpus d'auto-apprentissage.

Algorithm 2. *selfTrain(L_0, U)*

L_0 is labeled data, U is Unlabeled data
M_{init} <-- train(L_0)
Loop until *stopping criterion is met*
  L <--L_0 + select(U, Model)
  Model<--train (L)
End loop
Return *Model*
Function *Select(U, Model)*
  selected_sentences = best sentences based on a confidence measure
return *selected_sentences*

Nous avons implémenté l'algorithme d'auto-apprentissage utilisant 60% des données étiquetées manuellement comme corpus de base, noté $L_{0,}$ base de génération du modèle initial M_{init}. La précision de ce modèle est 89.63%. Les données non étiquetées utilisées dans cet algorithme d'auto-apprentissage sont notées *U* et sont présentées dans la section 5.3.

5.5.2. Utilisation de la mesure de confiance du mot dans le choix des données

Dans cette expérimentation, nous avons utilisé 10% des données étiquetées manuellement comme corpus de test. En prenant la moyenne des confiances données par le système aux mots d'une phrase donnée, la meilleure performance obtenue en appliquant l'algorithme d'auto-apprentissage est 89.86%, après 155 itérations. Le taux de réduction d'erreur est de l'ordre de 2.15%. En analysant les phrases sélectionnées, nous avons observé qu'elles sont de taille petite. C'est pourquoi nous avons combiné cette mesure avec un poids de la taille de la phrase selon la formule suivante :

$$Conf\_M = \frac{\frac{Words\_Conf}{sentence\_length} + \alpha(\frac{sentence\_length}{max\_sentence\_length})}{(1+\alpha)Sentence\_length}$$

Dans cette formule, *words_conf* est le total des mesures de confiance d'une phrase donnée, *sentence_length* représente le nombre des tokens de la phrase, *max_sentences_length* est le nombre des tokens de la phrase la plus longue du corpus non étiqueté et α un nombre positif déterminé expérimentalement.

La figure 5.8 suivante résume les résultats de l'expérimentation ; les points du graphe représentent les précisions des modèles auto-appris. A chaque itération de l'expérimentation, nous ajoutons la meilleure phrase, en nous basant sur la mesure de confiance(Conf_M) et sur la taille de la phrase.

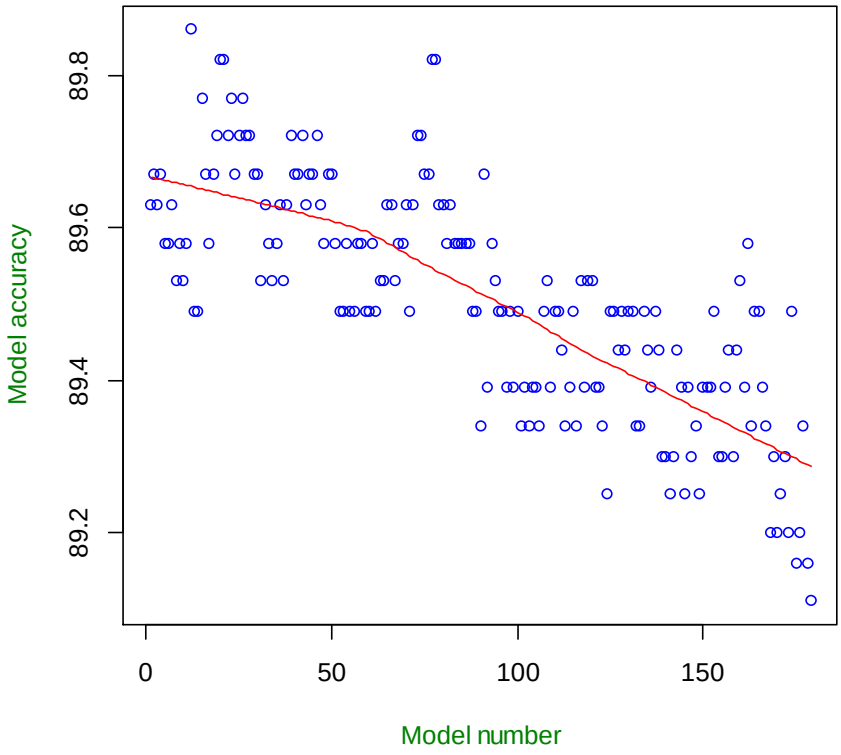

Figure 5.8. Auto apprentissage appliqué à l'étiquetage morphosyntaxique de l'amazighe en utilisant la confiance donnée aux mots par le système dans le choix des données

En variant α, nous obtenons une meilleure précision égale à 89.89% : soit une réduction du taux d'erreur de 2.5%, et ce après 11 itérations du programme, avec une valeur du coefficient α égale à 3. Il est à noter que la valeur de la précision commence à diminuer après quelques itérations (voir la figure 5.8 ci-dessus).

5.5.3. Utilisation de la confiance de la phrase dans le choix des données

En utilisant la mesure de confiance du système donnée aux phrases comme critère de sélection de la meilleure phrase à ajouter aux données annotées, nous obtenons une précision de notre modèle de 89.96%, après 840 itérations. C'est la précision maximale de la courbe d'auto apprentissage lorsque nous choisissons à chaque itération de l'algorithme une phrase. La figure 5.9 résume les résultats de cette expérimentation. Le gain en performance est de 0.33%, soit l'équivalent d'une réduction du taux d'erreur de 3.20%.

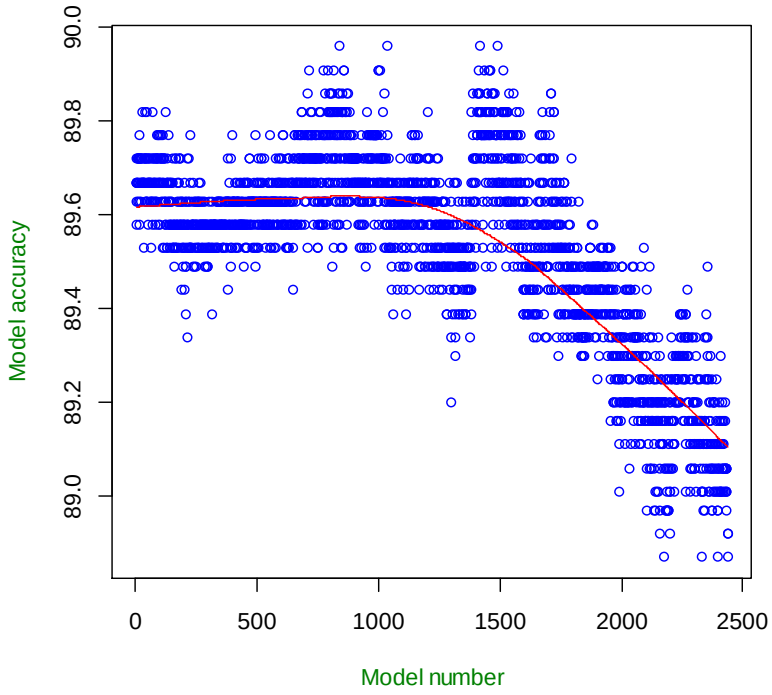

Figure 5.9. Auto apprentissage appliqué à la tâche d'annotation d'étiquetage morphosyntaxique de l'amazighe

5.5.4. Utilisation du caractère informatif et la mesure de confiance lors du choix des données

Vu que les OOVs sont une source importante d'erreurs, nous avons implémenté un nouvel algorithme (Algorithme 3), exploitant à chaque itération les fréquences des étiquettes attribuées par les modèles générés par l'algorithme d'auto apprentissage. Les OOVs choisis sont sélectionnés par ordre décroissant de la mesure de confiance attribuée par le système. A chaque nouvelle itération de l'algorithme, de nouvelles mesures de confiances sont attribuées aux mots et aux phrases du corpus des données non étiquetées *U*.

Algorithme 3. *InformativenessConfidence(L_0, U)*

1 L_0 *is labeled data,* T_0 *test file, U is Unlabeled data*
2 M_{init} *<-- train(L_0)*
3 ***For each*** OOV_i *in OOV sorted by confidence*
4 ST_i*<- select(U,* OOV_i*)*
5 BS_i*<- select_BS(*ST_i*,* $Model_i$*)*
6 *Tf = Identify most frequent tag of* OOV_i
7 BS_i*= Assign Tf, to select(U,* OOV_i*))*
8 *U<-- U-* BS_i
9 *L <--*L_0 *+* BS_i
10 $Model_i$*<--train (L)*
11 *Test(*T_0*)*
12 ***End For each***
13 ***return*** *Model*

Dans cet algorithme, ST_i représente les phrases sélectionnées à partir de *U* pour l'i^{ème} OOV trié selon la mesure de confiance. BS_i représente les meilleures phrases sélectionnées à partir des phrases choisies dans l'ensemble U, ST_i . Le choix des phrases se base sur la mesure de confiance du $Model_i$, i^{ème} modèle généré par cet algorithme.

La figure 5.10 montre l'impact positif de l'utilisation de la propriété de fréquence des *OOVs* et la mesure de confiance du système, et ce, malgré l'utilisation de 41% seulement des OOVs. Nous avons obtenu une précision de 90.24% après 141 itérations : soit une réduction du taux d'erreur de 5.90% (Outahajala et *al.*, 2015).

Le taux de réduction d'erreur obtenu pour ce petit corpus est légèrement meilleur que celui obtenu dans des travaux similaires sur l'apprentissage semi-supervisé. En effet, le taux de réduction d'erreur obtenu est entre 4% et 5% sur la base du corpus de Wall Street Journal (Spoustova, 2009 ; Søgaard, 2010).

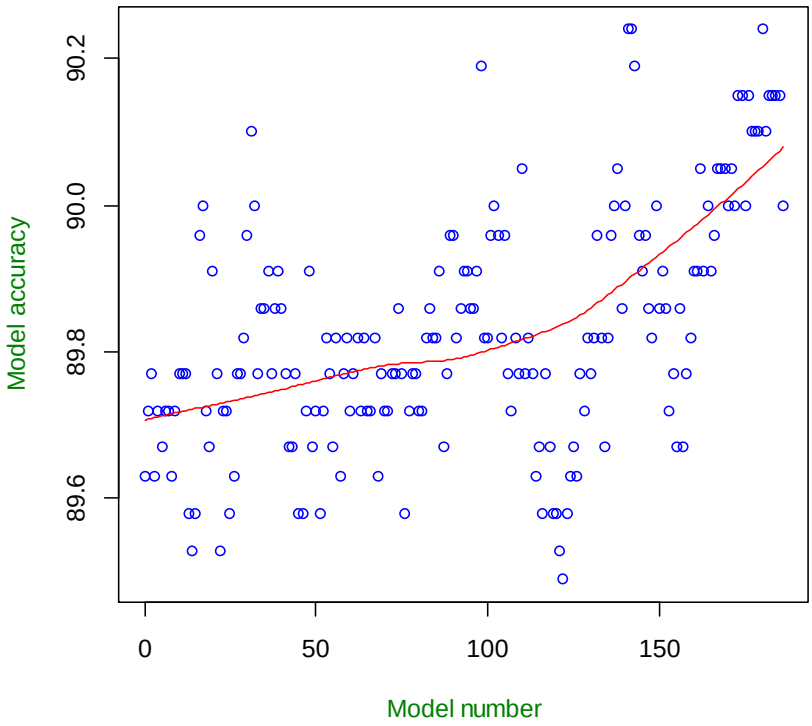

Figure 5.10. Résultats de l'auto-apprentissage utilisant les fréquences des *OOV* dans le corpus *U* et la mesure de confiance du système

5.6. Utilisation des lexiques externes pour l'amélioration des résultats de l'étiqueteur choisi

En analysant les erreurs de la sortie du modèle généré, nous avons constaté plusieurs sources d'erreurs : les mots hors vocabulaire, les entités nommées, les adjectifs et les participes sont souvent étiquetés comme noms communs et vice versa, etc. Pour réduire ces erreurs et améliorer les performances de notre étiqueteur, nous avons décidé de construire un lexique qui aide à l'annotation morphosyntaxique des textes amazighes.

Afin de construire ce lexique, nous avons utilisé plusieurs ressources lexicales existantes, telles que (El Gholb, 2011; Sghir, 2014). Cette ressource lexicale a été utilisée conjointement avec la mesure de confiance du modèle du système, en n'affectant pas l'étiquette morphosyntaxique du lexique, s'il existe, au mot ayant une mesure de confiance inférieure au seuil prédéfini α. Cette affectation se fait selon l'algorithme 4 ci-dessous.

Algorithme 4. *LexiconConfidence(L_0, Lex, α)*

```
1   L0 is labeled data, T0 test file, Lex is a lexicon with POS tags and α
    is a threshold
2   Minit <-- train(L0)
3   T0_Out_With_Conf <-- Minit(T0)
3   For each Wordi in T0_Out_With_Conf
4    If (Confidence_M(Wordi) < α & Lex_contains(Wordi)) then
5        Tag(Wordi) = lexicon_POS(Wordi),
6    End If
7   End For each
8   Function Confidence_M (Word)
9      Return Value of model confidence measure of the Word
10  End Function
```

Dans cet algorithme, L_0 est le corpus d'apprentissage, T_0 est le corpus de test et Lex est le lexique des mots amazighes avec une étiquette morphosyntaxique.

Tableau 5.3. Précision du modèle en fonction du seuil de la mesure de confiance

Valeur du seuil α	**Précision du modèle**
0.1	91.18
0.2	91.23
0.3	91.37
0.4	91.84
0.5	92.36
0.6	92.83
0.7	93.12
0.8	93.45
0.85	93.59
0.90	93.68
0.91	93.78
0.92	**93.82**
0.93	93.78
0.94	93.77

Nous avons ainsi construit un lexique externe, que nous avons employé avec la mesure de confiance du modèle. Le lexique recueilli contient environ 8 000 mots distincts, avec leurs étiquettes morphosyntaxiques respectives. Nous avons obtenu une performance égale à 93.82%, soit un gain de 2.64% en précision avec une valeur de α égale à 0.92 (voir tableau 5.3 ci-dessous).

5.7. Synthèse

L'apprentissage supervisé nécessite beaucoup de temps, vu que le corpus d'apprentissage est construit manuellement par des annotateurs. Aussi, l'étape de révision nécessite une intervention humaine et tout un processus pour la validation du corpus annoté. C'est pourquoi, dans ce chapitre, nous explorons l'utilisation des ressources externes via l'utilisation des méthodes d'apprentissage semi-supervisé et les lexiques externes. L'apprentissage semi-supervisé est une approche qui consiste en l'utilisation des données étiquetées ainsi que des données non étiquetées et dont l'objectif est de construire un classificateur plus précis en exploitant les données non étiquetées.

Dans ce chapitre, nous avons présenté l'état de l'art des méthodes d'apprentissage semi-supervisé, en particulier les algorithmes d'auto apprentissage et le co-apprentissage. Puis, nous avons exposé un corpus amazighe de 218 073 mots, que nous avons prétraités et segmentés afin d'obtenir 225 901 jetons. Nous avons ensuite expérimenté l'utilisation de l'algorithme d'auto-apprentissage à la tâche d'étiquetage morphosyntaxique de l'amazighe. Les résultats obtenus en utilisant le caractère informatif des mots avec la mesure de confiance du modèle sont très prometteurs. En effet, le taux de réduction d'erreur obtenu est de l'ordre de 5.9%. Enfin, nous avons expérimenté l'utilisation des lexiques externes en fonction de la mesure de confiance du modèle et nous avons obtenu une amélioration de performance de 93.82%, soit un gain de précision de l'ordre de 2.64%.

CONCLUSION

La langue amazighe est parlée au Maroc, en Algérie, en Libye, en Tunisie, et en Egypte; elle est également parlée par d'autres communautés dans certaines régions du Niger, du Mali et du Burkina Faso. Elle est également pratiquée par des milliers d'immigrants amazighes partout dans le monde. C'est une langue qui est restée, jusqu'à un passé proche essentiellement orale. Plusieurs dialectes de cette langue existent. En ce qui concerne son orthographe, le système adopté au Maroc est celui conçu par l'IRCAM. Une brève description de ce système a été présentée dans le chapitre 1.

Les approches de la linguistique computationnelle utilisent de plus en plus les collections de données pour l'analyse du langage. Dans ces approches quantitatives, les connaissances sont conçues statistiquement, sur la base de collecte de textes ou d'enregistrements sonores.

La langue amazighe, comme la majorité des langues qui n'ont commencé les recherches en TAL que récemment, souffre encore de la pénurie d'outils et de ressources pour son traitement automatique. Dans cette optique, et vu que les corpus constituent la base de la recherche LC et en TAL, nous avons construit le premier corpus annoté avec les informations morphosyntaxiques relatives à l'amazighe marocain, en suivant un processus bien précis pour assurer sa qualité. Il contient environ 20 000 mots. La vitesse d'annotation est comprise entre 80 et 120 mots par heure. L'accord entre annotateurs est de 94.98%. À notre connaissance, le corpus annoté réalisé dans le cadre de cette thèse est le premier corpus dans son genre pour la langue amazighe. Cette ressource, même si elle est de petite taille, est très utile pour le TAL amazighe. C'est aussi le cas pour l'apprentissage des étiqueteurs morphosyntaxiques, outils de base pour des travaux plus avancés.

Nous avons abordé également la problématique d'attribution automatique des étiquettes morphosyntaxiques. Une des difficultés principales de cette tâche est l'ambiguïté : on peut annoter une même forme de surface de plusieurs façons selon sa position et son utilisation dans la phrase. Afin d'atteindre cet objectif, nous avons formé deux modèles de classification de séquences en nous basant sur les SVMs et les CRFs et en utilisant les propriétés lexicales, les propriétés de contexte, avec une phase de segmentation comme prétraitement. Nous avons choisi la technique de 10 fois validation croisée pour évaluer notre approche. Les résultats

obtenus sont très prometteurs du fait que nous avons utilisé un corpus de ~20k mots environ. C'est le premier segmenteur réalisé pour cette langue. Sur la base des résultats de nos expériences, nous avons constaté que les performances des SVMs et des CRFs sont très comparables. Les résultats des SVMs sont légèrement meilleurs que ceux des CRFs à toutes les parties de la validation croisée ainsi que la moyenne sur les 10 parties de la validation croisée (99.77% vs. 99.67%). La meilleure performance des SVMs a légèrement dépassé celle du modèle basé sur les CRFs (99.95% contre 99.89%). Les deux modèles ont été entrainés sur la base d'un corpus de 91 376 jetons. Les résultats de l'annotation morphosyntaxique ont montré que la précision des CRFs dans les parties de la 10 fois validation croisée a dépassé celle des SVMs (91.18% vs. 90.75%). En nous basant sur un jeu de 28 étiquettes, nous avons constaté que la moyenne de précision des CRFs sur les 10 parties de la validation croisée a dépassé celle des SVMs (89.48% vs. 89.29%).

Vu que l'obtention des données annotées est très coûteuse, nous avons mené des expériences d'auto-apprentissage de notre étiqueteur. Pour cet effet, nous avons collecté un corpus de 218 073 mots. Le nombre total des jetons après la segmentation dudit corpus devient 225 901 jetons. L'implémentation de l'algorithme d'auto-apprentissage en sélectionnant les données non étiquetées sur la base de la mesure de confiance du modèle nous a permis d'obtenir une réduction du taux d'erreur de 3.20%. En adaptant l'algorithme d'auto-apprentissage et en exploitant la propriété de fréquences des OOVs avec la mesure de cofinance du système, nous avons obtenu un meilleur résultat. Le taux de réduction d'erreur obtenu atteint 5.90%.

L'Analyse des erreurs de la sortie du modèle généré fait apparaitre plusieurs sources d'erreurs; les mots hors vocabulaire, entités nommées, les adjectifs et les participes sont souvent étiquetés comme noms communs et vice versa, etc. Afin de réduire ces erreurs et, par conséquent, améliorer les performances de notre étiqueteur, nous avons construit un lexique externe que nous avons employé avec la mesure de confiance du modèle. Nous avons obtenu une performance de 93.82%, soit un gain en précision de 2.64%.

Ce travail a permis de doter la langue amazighe de plusieurs outils et ressources, que nous considérons comme étant des éléments de base pour des travaux plus avancés en TAL de l'amazighe. Nous avons aussi comparé les SVMs et les CRFs à travers des tâches relevant de l'annotation morphosyntaxique et avons trouvé, empiriquement, qu'ils sont très comparables. Nous avons également proposé un nouvel algorithme d'auto-apprentissage, qui se base sur le caractère informatif des OOVs et un autre algorithme utilisant un lexique externe conjointement avec la mesure de confiance du système. Les résultats obtenus sont très

prometteurs et les approches proposées sont indépendantes de la langue amazighe, et par conséquent, elles sont applicables à d'autres langues.

Perspectives de recherche

Au terme de ce travail, plusieurs pistes de recherches, en relation avec l'annotation morphosyntaxique, se dégagent, dont:

- l'utilisation d'un corpus non étiqueté plus grand, avec moins de mots hors vocabulaire, permettra d'améliorer les performances de l'algorithme présenté dans le chapitre 5. Il en est de même pour le lexique recueilli ;
- l'utilisation de l'apprentissage actif (Ringger et *al.*, 2007; Settles, 2009) facilitera la construction d'un plus grand corpus annoté.

Le corpus annoté peut, également, être enrichi en lui attribuant des informations syntaxiques. Ce corpus sera d'une grande utilité dans la réalisation des outils de base performants pour le TAL, tels que les analyseurs syntaxiques, sémantiques et autres.

Plusieurs travaux de recherche seront d'une grande utilité pour la langue amazighe. Ces travaux peuvent concerner deux champs. Le premier concernera la création des ressources linguistiques, avec notamment l'enrichissement du corpus annoté mophosyntaxiquement en le complétant par d'autres types d'annotations syntaxique et sémantique (Outahajala et *al.*, 2014b) ; la création d'un WordNet amazighe ; la création des corpus de textes parallèles, etc.

Le deuxième champ de recherche concernera la création d'outils de base et outils et avancés pour le TAL. A titre d'exemple, un racineur, un analyseur morphologique ; un analyseur syntaxique partiel et syntaxique ; des outils de traduction et d'aide à la traduction ; la reconnaissance de la parole et la synthèse de la parole, etc.

Contributions

Les principales contributions de notre travail touchent deux champs distincts : les ressources linguistiques et les méthodes et outils du TAL.

Le premier champ se décline ainsi :

1. la création d'un corpus amazighe annoté morphosyntaxiquement. Ce corpus contient environ 20 mille mots. L'accord entre annotateurs de ce corpus est de l'ordre de 94.98% ;
2. la collecte et le prétraitement d'un corpus de textes bruts d'environ 220 000 mots ;
3. la collecte d'un lexique d'environ 8 000 mots avec les étiquettes morphosyntaxiques.

Quant aux méthodes et outils pour le TAL :

1. deux étiqueteurs morphosyntaxiques, basés sur les SVMs et les CRFs, ont été réalisés ;
2. un nouvel algorithme basé sur les techniques d'apprentissage semi-supervisé, utilisant le caractère informatif des OOVs et les probabilités marginales des CRFs, a été conçu ;
3. plusieurs scripts pour le traitement de la langue amazighe, tels que celui qui corrige les mauvais placements des espaces typographiques, le modèle à base des fréquences, etc., ont été réalisés.

L'ensemble des ressources linguistiques et outils du TAL réalisés et présentés dans le cadre de ce mémoire sont librement disponible sur le site : www.outamed.com.

BIBLIOGRAPHIE

Abeillé, A., Clément, L., & Toussenel, F. 2003. Building and Using Parsed Corpora, Chapitre Building a Treebank for French. *Language and Speech series*, Kluwer, Dordrecht.

Abney, S. 1996. Part-of-Speech Tagging and Partial Parsing in Church, K. et *al.*, (ed.), *Corpus-based Msdethods in Language and Speech, Kluwer*, Dordrecht.

Abney, S. 2002. Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 360–367, Philadelphia, PA.

Adafre, S. F. 2005. Part of Speech Tagging for Amharic Using Conditional Random Fields. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pp. 47-54.

Agnaou, F., Bouzandag, A., El Baghdadi, M., El Gholb, E., Khalafi, A., Ouqua, K., & Sghir, M. 2011. *Lexique scolaire*. Publications de l'IRCAM.

Ait Ouguengay, Y., Jaa, J., & Zenkouar, L. 2007. Numérisation du tifinaghe : aspects et problèmes. In *Proceedings of la typographie entre les domaines de l'art et l'informatique, septembre 2004*. Rabat, Morocco. Publications de l'IRCAM.

Ait Ouguengay, Y., & Taalabi, M. 2009. Elaboration d'un réseau de neurones artificiels pour la reconnaissance optique de la graphie amazighe: Phase d'apprentissage. In *Proceedings of Systèmes intelligents-Théories et applications*. Europia productions.

Ait Ouguengay, Y., & Bouhjar, A. 2010. For Standardized Amazigh Linguistic Resources. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*. LREC 2010, Malta, May 17-23, pp. 2699-2701.

Ait Ouguengay, Y., Elhamdaoui, A., Hassnaouy, B., & Feddah, A. 2012. Projet GCAM - Vers une gestion Informatisée du corpus amazighe à l'IRCAM. In *Proceedings of the 5th International Conference on Amazigh and ICT*, NTIC-2012, Rabat, Morocco.

Aizerman, M., Braverman, E., & Rozonoer, L. 1964. Theoretical Foundations Of The Potential Function Method In Pattern Recognition Learning. Automation and Remote Control 25:821--837.

Ameur, M., Bouhjar, A., Boukhris, F., Boukouss, A., Boumalk, A., Elmedlaoui, M., Iazzi, E., & Souifi, H. 2004. *Initiation à la langue amazighe*. Publications de l'IRCAM.

Ameur, M., Bouhjar, A., Elmedlaoui, M., & Iazzi, E. 2006a. *Vocabulaire de la langue amazighe 1*. Publications de l'IRCAM.

Ameur, M., Bouhjar, A., Boukhris, F., Boukouss, A., Boumalk, A., Elmedlaoui, M., & Iazzi, E. 2006b. *Graphie et orthographe de l'amazighe*. Publications de l'IRCAM.

Amrouch M., Es Saady Y., Rachidi A., El Yassa M., & Mammass D. 2009. Printed Amazigh Character Recognition by a Hybrid Approach Based on Hidden Markov Models and the Hough Transform. In *Proceedings of ICMCS'09*, Ouarzazate, Morocco.

Andries, P. 2004. La police open type Hapax berbère. In *Proceedings of the workshop : la typographie entre les domaines de l'art et l'informatique*, pp. Publications de l'IRCAM.183—196.

Andries, P. 2008. *Unicode 5.0 en pratique*. Dunod éditions, Paris.

Ataa Allah, F., & Boulaknadel, S. 2010. Pseudo-racinisation de la langue amazighe. In *Proceedings of TALN 2010*, Montréal, pp.19--23.

Ataa Allah, F., Frain, J., & Ait Ouguengay, Y. 2013. Amazigh Language Desktop Converter. In *Proceedings of 3ème Symposium International sur le Traitement Automatique de la Culture Amazighe*, Beni Mellal, Maroc.

Baker J.K. 1975. Stochastic Modeling for Automatic Speech Understanding. In *Reddy D.R. (Ed.) Speech recognition*, Academic Press, New-York, NJ. pp. 521-542.

Belkin, M., Matveeva, I., & Niyogi, P. 2004. Regularization and Semi-Supervised Leaming on Large Graphs. Lecture Notes in Computer Science, 3120, pp. 624-638.

Benajiba, Y., Diab M., & Rosso P. 2010a. Arabic Named Entity Recognition: A Feature-Driven Study. *IEEE Transactions on Audio, Speech and Language Processing, vol. 15, num. 5.* Special Issue on Processing Morphologically Rich Languages, pp. 926-934.

Benajiba Y., Zitouni I., Diab M., & Rosso P. 2010b. Arabic Named Entity Recognition: Using Features Extracted from Noisy Data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL-2010*, Uppsala, Sweden, July 11-16, pp. 281-285.

Bertran, M., Borrega, O., Recasens, M., & Soriano, B. AnCoraPipe: A tool for multilevel annotation. In *Procesamiento del Lenguaje Natural*, nº 41.

Black, W., Elkateb, S., Rodriguez, H., Alkhalifa, M., Vossen, P., Pease, A., & Fellbaum, C. 2006. Introducing the Arabic wordnet project. In *Proceedings of the 3rd International WordNet Conference* (GWC-06), pp. 295-299.

Blum, A., & Chawla, S. 2001. Learning From Labeled and Unlabeled Data Using Graph Mincuts. In *Proceedings of the 18th International Conference on Machine Learning*, pp. 19--26. Morgan Kaufmann, San Francisco, CA.

Blum, A., & Mitchell, T. 1998. Combining Labeled and Unlabeled Data With Co-Training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLTtions de l'98)*, pp. 92–100.

Boukhris, F. Boumalk, A. El moujahid, E., & Souifi, H. 2008. *La nouvelle grammaire de l'amazighe*. Publications de l'IRCAM.

Boukouss, A. 1995. *Société, langues et cultures au Maroc.* Publications de la Faculté des Lettres de Rabat, Maroc.

Boukouss, A. 2012. *Revitalisation de la langue amazighe : défis, enjeux et stratégies.* Publications de l'IRCAM.

Boulaknadel, S. 2009. Amazigh ConCorde: An Appropriate Concordance for Amazigh. In *Proceedings of 1er Symposium International sur le Traitement Automatique de la Culture AMazighe (SITACAM).* Agadir, Morocco.

Boulaknadel, S., & Ataa Allah, F. 2011. Building a Standard Amazigh Corpus. In *Proceedings of International Conference on Intelligent Human Computer Interaction*. Prague, Tchec.

Boumalk, A., & Naît Zerrad, K. 2009. *Vocabulaire grammatical*. Publications de l'IRCAM.

Brill, E. 1992. A Simple Rule-Based Part Of Speech Tagger. In Proceedings of the Third Conference on Applied Natural Language Processing.

Brill, E. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4), pp 543--565.

Brown P., Cocke J., Pietra S. D., Jelinek F., Lafferty J.D., Mercer R.L., & Rossin P.S. 1990. A Statistical Approach To Machine Translation. *Computational Linguistics*, 16(2), 79-85.

Candito, M. et Seddah, D. 2012. Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. *In 19ème conférence sur le Traitement Automatique des Langues Naturelles*, Grenoble, France.

Chapelle, O., & Zien, A. 2005. Semi-Supervised Classification by Low Density Separation. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*.

Charniak, E., Hendrickson, C., Jacobson, N., & Perkowitz, M.1993. Equations For Part-Of-Speech Tagging. In *AAAI*. pp. 784--789.

Charniak, E. 1997. Statistical Parsing With A Context-free Grammar And Word Statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Menlo Park. AAAI Press/MIT Press.

Chafiq, M. 1991. *أربعة وأربعون درسا في الأمازيغية* [Forty Four Lessons in Amazigh]. éd. Arabo-africaines.

Chaker, S. 1984. *Textes en linguistique berbère - introduction au domaine berbère*, éditions du CNRS, pp 232-242.

Church K. 1988. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the Conference on Applied Natural Language Processing, ACL'1988, Austin, TX*. pp 136-143.

Church, K. 1993. Char_align: a Program for Aligning Parallel Texts at the Character Level. In *Proceedings of ACL'93*, Columbus, Ohio.

Cieri, C., & Liberman, M. 2008. 15 Years of Language Resource Creation and Sharing: A Progress Report on LDC Activities. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, LREC'08, Marrakech.

Clark, S., Curran, J. R., & Osborne, M. 2003. Bootstrapping POS Taggers Using Unlabelled Data. In *Proceedings of CoNLL'03*.

Cohen, D. 2007. Chamito-sémitiques (langues). In *Encyclopædia Universalis*.

Collins, M., & Singer, Y. 1999. Unsupervised Models for Named Entity Classification. In *Proceedings of the Empirical Methods in NLP Conference*, pp 100–110, University of Maryland, MD.

Constant, M., Tellier, M., I., Duchier, D., Dupont, Y., Sigogne, A., & Billot S. 2011. Intégrer des connaissances linguistiques dans un CRF : application à l'apprentissage d'un segmenteur-étiqueteur du français. In *Proceedings of TALN'11*.

Cucerzan, S., & Yarowsky, D. 2002. Bootstrapping a Multilingual Part-Of-Speech Tagger in One Person-Day. In *Proceedings of the 6th Workshop on Computational Language Learning*, Taipei, Taiwan.

Dasgupta, S., Littman M., & McAllester, D. 2002. PAC Generalization Bounds for Co-training. In *Dietterich T. G., Becker S., and Ghahramani Z., eds, Advances in Neural Information Processing Systems* 14, pp. 375--382, Cambridge, MA. MIT Press.

Diab, M., Hacioglu, K., & Jurafsky, D. 2004. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In *Proceedings of Human Language Technology-North American Association for Computational Linguistics (HLT-NAACL).*

Diab, M. 2007. Towards an Optimal POS tag set for Modern Standard Arabic Processing. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria.

El Azrak, N., & El Hamdaoui, A. 2011. Référentiel de la terminologie amazighe : outil d'aide à l'aménagement linguistique. In *Proceedings of the 4th international conference on Amazigh and ICT*, Rabat, Morocco.

Laabdelaoui, R., Boumalk, A., Iazzi, E.M., Souifi, H., & Ansar, K. 2012. Manuel de conjugaison de l'amazighe. Publications de l'IRCAM.

El Gholb, L. 2011. La *conjugaison du verbe en amazighe : élément pour une organisation.* Editions Universitéaires Européennes, Sarrebruck, Allemagne.

El Ayachi, R., Moro, K., Fakir, M., & Bouikhalene, B. 2010. *On the Recognition of Tifinaghe Scripts*. Journal of Theoretical and Applied Information Technology, 20(2), pp. 61--66.

Es Saady, Y., Ait Ouguengay, Y., Rachidi, A., Elyassa, M., & Mammass, D. 2009. Adaptation d'un correcteur orthographique existant à la langue amazighe: cas du correcteur Hunspell. In *Proceedings of 1er Symposium International sur Le Traitement Automatique de la Culture Amazighe (SITACAM),* Agadir, Morocco.

Es Saady, Y., Rachidi, A., El Yassa, M., & Mammass, D. 2011. Amazigh Handwritten Character Recognition Based on Horizontal and Vertical Centerline of Character. *International Journal of Advanced Science and Technology*, vol.33, pp.33--50.

Fellbaum, C. 1998. WordNet. *Blackwell Publishing Ltd*.

Forney Jr, G. D. 1973. The Viterbi Algorithm. In Proceedings of the IEEE, 61(3), pp. 268--278.

Giménez, J., & Màrquez, L. 2004. SVMTool: A General POS Tagger Generator Based on Support Vector Machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal, 26–28 May 2004, pp. 43--46.

Grandvalet, Y., & Bengio, Y. 2004. Semi-Supervised Leaming by Entropy Minimization. In *Lawrence K. Saul, Yair Weiss, and Léon Bottou, eds, Advances in Neural Information Processing Systems 17*, pp. 529--536. MIT Press, Cambridge, MA.

Greene, B.B., & Rubin, G.M. 1971. *Automatic Grammatical Tagging of English*. Providence, R.I.: Department of Linguistics, Brown University.

Habash, N. 2010. *Introduction to Arabic Natural Language Processing*. *Synthesis Lectures on Human Language Technologies* 3(1), 1-187.

He, X., Zemel, R. S., & Carreira-Perpindn, M. A. 2004. Multiscale Conditional Random Fields For Image Labeling. In *Computer vision and pattern recognition,* Proceedings of the 2004 IEEE computer society conference.

Hearst, M. 1992. Noun homonym Disambiguation Using Local Context in Large Text Corpora. In *Proceedings of the 7th Annual Conference of the UW Centre for the new OED and Text Research*, pp. 539--545. Morgan Kaufmann Publishers. San Fransisco, CA.

Hindle, D., & Rooth, M. 1991. Structural Ambiguity and Lexical Relations. *Computational Linguistics*, 19(1), pp, 103–120.

Iazzi, E., & Outahajala, M. 2008. Amazigh Data Base. In *Proceedings of HLT & NLP within the Arabic world: Arabic language and local languages processing status updates and prospects, 6th International Conference on Language Resources and Evaluation*, LREC'08, Morocco, pp, 36--39.

Institut Royal de la Culture Amazighe. 2003a. Proposition de codification des tifinaghes, Rabat, Maroc.

Institut Royal de la Culture Amazighe. 2003b. Conception et mise au point des polices tifinaghes. Centre des Etudes Informatiques, Systèmes d'Information et Communication. http://www.ircam.ma/fr/index.php?soc=telec&rd=1, plan d'action.

Institut Royal de la Culture Amazighe. 2004. Polices et Claviers UNICODE. Centre des Etudes Informatiques, Systèmes d'Information et Communication, http://www.ircam.ma/fr/index.php?soc=telec&rd=3.

Institut Royal de la Culture Amazighe. 2012. Proposition d'ajout d'un nouveau code de langue pour l'amazighe. Rabat, Maroc.

Jelinek F. 1976. Continuous Speech Recognition by Statistical Methods. In *Proceedings of the IEEE, 64*. pp 532-556.

Joachims, T. 2003. Transductive Learning Via Spectral Graph Partitioning. In *Proceeding of The Twentieth International Conference on Machine Learning* (lCML-2003).

Johansson, S. 1980. The LOB Corpus for British English Texts: Presentation and Comments. *ALLC Journal*, 1(1), p. 25-36.

Jurafsky, D., & Martin, J.H. 2009. Speech and Language Processing: An Introduction to Natural Language Processing, *computational linguistics, and speech recognition*, 2nd Ed. New Jersey: Prentice Hall.

Karush, W. 1939. Minima of Functions of Several Variables With Inequalities As Side Constraints. *Master's thesis, Dept. of Mathematics*, Univérsité de Chicago.

Kemp, C., Griffiths, T., Stromsten, S., & Tenenbaum, J. 2003. Semi-Supervised Learning With Trees. *Advances in Neural Information Processing System 16*.

Khoja, S, Garside, R., & Knowles, G. 2001. A Tagset For The Morphosyntactic Tagging Of Arabic. In *Proceedings of Corpus Linguistics.* Lancaster, UK, pp 341–353.

Klein, D., & Manning, M. 2002. A Generative Constituent-Context Model For Improved Grammar Induction. In *Proceedings of the 40th Annual Meeting of the ACL*.

Knerr, S., Personnaz, L., & Dreyfus,J. 1990. Single-Layer Learning Revisited: A Stepwise Procedure For Building and Training a Neural Network. In *Neurocomputing: Algorithms, Architectures and Applications*, Fogelman-Soulie and Herault (eds.). NATO ASI Series, Springer.

Kotsiantis S. 2007. Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 31.pp 249--268.

Krauss, M. 1992. The World's Languages in Crisis. *Language, 68*, 6--10.

Kudo, T., & Yuji Matsumoto, Y. 2000. Use of Support Vector Learning for Chunk Identification. In *Proceedings of CoNLL-2000 and LLL-2000*.

Kumar, S., & Hebert, M. 2003. Discriminative Fields For Modeling Spatial Dependencies In Natural Images. In *Advances in Neural Information Processing Systems 16*.

Kurčera, H., & Francis W. N. 1967. Computational Analysis of Present-Day American English. *Brown University Press*, Providence, RI.

Kuhn H.W., & Tucker, A.W. 1951. Nonlinear Programming. In *University of California Press*, editor, 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics, pp 481--492.

Lafferty, J. McCallum, A., & Pereira, F. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML-01*, pp. 282--289.

Lakshmana Pandian S., & Geetha, T. V. 2009. CRF Models for Tamil Part of Speech Tagging and Chunking. In *Proceeding ICCPOL '09*. Springer-Verlag Berlin, Heidelberg.

Levin E., Pieraccini R. 1995. Concept-Based Spontaneous Speech Understanding. In *Proceeding of the 4th European Conference on Speech Communication and Technology*, Eurospeech’95, Madrid, Espagne. pp 555-558.

Lguensat, M. 2012. *Aménagement graphique de tifinaghe*. Publications de l’IRCAM.

Li W., & McCallum, A. 2003. Rapid Development of Hindi Named Entity Recognition Using Conditional Random Fields and Feature Induction. In *ACM Transactions on Computational Logic*, pp 290--294.

Liu, Y., Carbonell, J., Weigele, P., & Gopalakrishnan, V. 2005. Segmentation Conditional Random Fields (SCRFs): A New Approach For Protein Fold Recognition. In *Research in Computational Molecular Biology*. Springer Berlin Heidelberg, pp. 408--422.

Maamouri, M., Bies, A., & Buckwalter, T. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. *In NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Makagonov, P., & Alexandrov, M. 1999. Some Statistical Characteristics for Formal Evaluation of the Quality of Text books and Manuals. In *Computing Research: Selected papers*, pp 99--103.

Malfrère, F., & Dutoit, T. 1997. High Quality Speech Synthesis for Phonetic Speech Segmentation. In *Proceedings of the European Conference On Speech Communication and Technology*, pp. 2631-2634.

Manning, C., & Schütze, H. 1999. Foundations of Statistical Natural Language Processing. *The MIT Press*.

Marcus, M., Santorini, B., & Marcinkiewicz, M. 1993. Building A Large Annotated Corpus Of English: The Penn Treebank. *Computational Linguistics*, 19:313--330.

Mercer, J. 1909. Functions Of Positive And Negative Type and Their Connection With The Theory of Integral Equations. Philos. Trans. Roy. Soc. London, A 209:415--446.

Ouqua, K. 2011. نحو حوسبة محلاتية للأفعال في الكتاب "tifawin a tamazight"و أبعادها التعليمية المدرسي[Towards verb valence dictionary for Amazigh textbook « tifawin a tamaziGt » and their educational dimensions]. In *Proceedings of the 4th International Conference on Amazigh and ICT*. Rabat, Morocco.

Outahajala, M, Zenkouar, L. 2007. La norme du tri, du clavier et Unicode. In *Proceedings of la typographie entre les domaines de l'art et l'informatique, septembre2004*. Rabat, Morocco. Publications de l'IRCAM, pp. 223--238.

Outahajala, M., Zenkouar, L., Rosso, P., & Martí, A. Tagging Amazigh with AncoraPipe. 2010. In *Proceedings of the Workshop on LR & HLT for Semitic Languages, 7th International Conference on Language Resources and Evaluation*, LREC'10, Malta, May 17-23, pp. 52--56.

Outahajala, M., Benajiba, Y., Rosso, P., & Zenkouar, L. 2011a. POS Tagging In Amazigh Using Support Vector Machines And Conditional Random Fields. In *Natural Language to Information Systems*, LNCS (6716), Springer-Verlag, pp, 238--241.

Outahajala, M. 2011b. Processing Amazighe Language. In *Natural Language to Information Systems*, LNCS (6716), Springer-Verlag, pp,313-317.

Outahajala, M., Zenkouar, L., & Rosso, P. 2011c. Building an annotated corpus for Amazighe. In *Proceedings of 4th International Conference on Amazigh and ICT*. Rabat, Morocco.

Outahajala, M., Benajiba, Y., Rosso, P., & Zenkouar, L. 2012. L'étiquetage grammatical de l'amazighe en utilisant les propriétés n-grammes et un prétraitement de segmentation. *e-TI - la revue électronique des technologies d'information*, Numéro 6.

Outahajala, M., Zenkouar, L., Benajiba, Y., & Rosso, P. 2013. The Development of a Fine Grained Class Set for Amazigh POS Tagging. *Computer Systems and Applications (AICCSA), 2013 ACS International Conference*.

Outahajala M., Zenkouar L., Benajiba Y., & Rosso P. 2014a. Utilisation des CACs et des ressources externes pour l'amélioration des performances de l'étiquetage morphosyntaxique. In: La Revue ASINAG, Special Issue on ICT and Amazighe, vol. 9.

Outahajala, M., Zenkouar, L., Rosso, P. 2014b. Construction d'un grand corpus annoté pour la langue amazighe. *La revue Etudes et Documents Berbères n°33*, pp. 57--74.

Outahajala M., Benajiba Y., Rosso P., Zenkouar L.. 2015. Using Confidence And Informativeness Criteria To Improve POS Tagging In Amazigh. In *Journal of Intelligent and Fuzzy Systems 28, pp. 1319—1330.* Doi : 10.3233/IFS-141417.

Peng, F., & McCallum, A. 2006. Information Extraction From Research Papers Using Conditional Random Fields. In *Information processing & management*, 42(4), pp. 963-979.

Pierce, D & Cardie, C. 2001. Limitations of Co-Training For Natural Language Learning From Large Datasets. In *Proceedings of the Empirical Methods in NLP Conference*, Pittsburgh, PA.

Pinto, D., McCallum, A., Wei, X., & Croft, W. B. 2003. Table Extraction Using Conditional Random Fields. In *Proceedings of the 26th annual international of SIGIR'03*, pp. 235-242, New York, USA.

Platt, J.C., Cristianini,N., & Shawe-Taylor, J. 2000. Advances in Neural Information Processing Systems. *Chapter Large margin DAGs for multiclass classification, volume 12 MIT Press*, pp. 547--553.

Rabiner, L. R., & Juang, B. H. 1986. An Introduction To Hidden Markov Models. *ASSP Magazine, IEEE, 3(1)*, pp. 4--16.

Raiss, H., & Cavalli-Sforza, V. 2012. ANMorph: Amazigh Nouns Morphological Analyzer. In *Proceedings of the 5th International Conference on Amazigh and ICT*, NTIC-2012, Rabat, Morocco.

Ramshaw, L.A., & Marcus, M. P. 1995. Text Chunking Using Transformation-Based Learning. In *Proceedings of the Third Workshop on Very Large Corpora*, Association for Computational Linguistics, pp. 82–94.

Ratnaparkhi, A. 1996. A Maximum Entropy Model for Part-Of-Speech Tagging. In *Proceedings of EMNLP*, Philadelphia, USA.

Ringger, E., McClanahan, P., Haertel, R., Busby, G., Carmen, M., Carroll, J., Seppi, K., & Lonsdale, D. 2007. Active Learning for Part-Of-Speech Tagging: Accelerating Corpus Annotation. In *Proceedings of the Linguistic Annotation Workshop*, pp. 101–108.

Salmon-Alt, S., Bick, E., Romary, L., & Pierrel, J.M. 2004. La FReeBank : vers une base libre de corpus annotés. *In Traitement Automatique des Langues Naturelles TALN'04*, Fès, Maroc.

Salton G. 1972. Experiments in Automatic Thesaurus Construction for Information Retrieval. In *Proceedings of IFIP'1972*, Ljubljana, Slovénie.

Sagot, B., & Fišer, D. 2008. Building A Free French Wordnet From Multilingual Resources. In *OntoLex*, May 2008, Marrakech, Morocco.

Sarkar, A. 2001. Applying Co-Training Methods to Statistical Parsing. In *Proceedings of the 2nd Annual Meeting of the NAACL*, pp 95–102, Pittsburgh, PA.

Sato, K., & Sakakibara, Y. 2005. RNA Secondary Structural Alignment With Conditional Random Fields. *Bioinformatics*, 21(suppl 2), pp. 237--242.

Schmid, H. 1994. part-of-speech tagging with neural networks. In *Proceedings of international conference on Computational Linguistics*, Kyoto, Japan.

Schmid, H. 1999. Improvements in Part-of-Speech Tagging with an Application to German. *In Proceedings of the ACL SIGDAT-Workshop. Academic Publishers*, Dordrecht, 13--26.

Scholkopf, B., & Smola, A.J. 2002. Learning with Kernels Support Vector Machines, *Regularization, Optimization, and Beyond. MIT Press*.

Schwenker, F. 2001. Solving Multi-Class Pattern Recognition Problems With Tree-Structured Support Vector Machines. *Pattern Recognition*, pp. 283–290.

Settles, B. 2009. Active Learning Literature Survey. *Computer Sciences Technical Report 1648*, University of Wisconsin–Madison.

Sha, F., & Pereira F. 2003. Shallow Parsing with Conditional Random Fields. In *Proceedings of Human Language Technology*.

Simard, M., Foster, G., Isabelle, P. 1992. Using Cognates to Align Sentences. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation, Montreal*.

Simpson, H., Cieri, C., Maeda, K., Baker, K., & Onyshkevych, B. 2008. Human language technology resources for less commonly taught languages: Lessons learned toward creation of basic language resources. In *Proceedings of the LREC 2008 Workshop on Collaboration: interoperability between people in the creation of language resources for less-resourced languages*, pp, 7--11.

Sghir, M. 2014. Essai de confection d'un dictionnaire monolingue amazighe: méthodologie et application, Parler de la vallée du Dadès (Sud-Est du Maroc). *Thèse de doctorat.* FLSH Saïs-Fès.

Skounti, A., Lemjidi, A., & Nami, E. M. 2003. Tirra aux origines de l'écriture au Maroc. *Publications de IRCAM*.

Smrž, O., & Hajič, J. 2006. The Other Arabic Treebank: Prague Dependencies and Functions. In *Ali Farghaly eds. Arabic Computational Linguistics*. CSLI Publications.

Søgaard, A. 2010. Simple Semi-Supervised Training Of Part-Of-Speech Taggers. In *Proceedings of the ACL 2010 Conference Short Papers*, pp. 205–208.

Spoustova, D., Hajic, J., Raab, J., & Spousta, M. 2009. Semi-Supervised Training For The Averaged Perceptron POS Tagger. In EACL, Athens, Greece.

Steedman, M., Osborne, M., Sarkar, A., Clark, S., Hwa, R., Hockenmaier, J., Ruhlen, P., Baker, S., & Crim, J. 2003. Bootstrapping Statistical Parsers From Small Datasets. In *Proceedings of EACL 03*.

Takahashi, F., & Abe, S. 2002. Decision-Tree-Based Multiclass Support Vector Machines. 2002. In *Proceedings of the 9th International Conference on Neural Information Processing*, Singapore, pp. 1418–1422.

Taskar, B., Abbeel, P., & Koller, D. 2002. Discriminative Probabilistic Models For Relational Data. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 485-492.

Tellier I., & Tommasi M. 2011. Champs Markoviens Conditionnels pour l'extraction d'information. In *Eric Gaussier & François Yvon*, Eds. Modèles probabilistes pour l'accès à l'information textuelle. Hermès.

Tjong Kim Sang, E. F., & Veenstra, J. 1999. Representing Text Chunks. In *Proceedings of EACL'99*, pp 173–179.

Uchimoto, K., Ma, Q., Murata, M., Ozaku, H., & Isahara, H. 2000. Named Entity Extraction Based on A Maximum Entropy Model and Transformation Rules. In *Proceedings of the ACL 2000*.

Vapnik, Valdimir N. 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, USA.

Véronis, J., & Khouri, L. 1995. Etiquetage grammatical multilingue: le projet multext. In *Traitement Automatique des Langues (TALN)*, 36(1/2), pp. 233–248.

Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189--196.

Zavrel, J., & Daelemans, W. 2000. Bootstrapping a Tagged Corpus Through Combination of Existing Heterogeneous Taggers. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, pp. 17--20, Athens, Greece.

Zenkouar, L. 2008. Normes des technologies de l'information pour l'ancrage de l'écriture Amazighe. *Revue Etudes et Documents Berbères n°27*, pp. 159--172.

Zenkouar, L. 2004. L'écriture Amazighe Tifinaghe et Unicode. *Revue Etudes et Documents Berbères n°22*, pp. 175--192.

Zenkouar, L., Labonté, A. 2006a. PNM 17.2.000 : technologies de l'information - Classement et comparaison de chaînes de caractères tifinaghes.

Zenkouar L., Labonté A. 2006b. PNM 17.6.000 : technologies de l'information - Prescriptions des claviers conçus pour la saisie des caractères tifinaghes.

Zenkouar L., Ait Ouguengay, Y., Outahajala, M. 2006c. Les normes pour la langue amazighe. Bulletin d'information n° 5-6, IRCAM, pp. 45--49.

Zipf, G.K.1949. Human Behaviour And The Principle Of Least Effort. Addison-Wesley.

ANNEXES

Annexe 1: Attributs et sous attributs des étiquettes utilisées dans l'annotation morphosyntaxique de l'amazighe

	Nœud	Description des attributs	Desciption des valeurs des attributs	Description des sous attributs	Description des sous attributs
1	<n> / noun	<gen> / gender	“c” / common “f” / feminine “m” / masculine		
		<num> / number	“c” / common “p” / plural “s” / singular		
		<state> / state	“free” / free “construct” / construct		
		<derivative>/ derivative	"yes" "no"		
		<postype> / PoS subclassification	"common" "proper" "numeral"		
			"parental"	<person>/ person (only for “personal”, “possessive”)	“1” / first person “2” / second “3” / third
				<possessornum> / possessor’s Number	“p” / plural “s” / singular
				<possessorgen> / possessor’s Gender	“f” / feminine “m” / masculine

2	<a>/ adjective	<num> / number	“c” / common “p” / plural “s” / singular		
		<state> / state	“free” / free “construct” / construct		
		<derivative>/ derivative	"yes" "no"		
		<postype> / PoS Subclassification	"qualificative" "ordinal" "numeral"		
3	<v> / verb	<gen> / gender	“c” / common “f” / feminine “m” / masculine		
		<num> / number	“c” / common “p” / plural “s” / singular		
		<form> / form	Imperative Participle		
		<derivative>/ derivative	"yes" "no"		
		<aspect> / Aspect	Aorist		
			Perfective Imperfective	<negative>/ only for perfective and imperfective	"yes" "no"

4	<p> / pronoun	<gen> / gender	“c” / common		
			“f” / feminine “m” / masculine		
		<num> / number	“c” / common “p” / plural “s” / singular		
		<postype> / PoS	“Demonstrative”	<deictic>/deictic only for	
		subclassification	“exclamative”	"demonstrative"	
			“indefinite”		
			“interrogative” “personal”	<autonome>/ only for personal	“autonome” “affix”
			“possessive” “relative”	<person>/ person (only for “personal”, “possessive”)	“1” / first person “2” / second “3” / third
				<possessornum> / possessor’s number (only for “possessive”)	“p” / plural “s” / singular
				<possessorgen> / possessor’s gender (only for “possessive”)	“f” / feminine “m” / masculine

5	<d> / determiner	<gen> / gender	“c” / common “f” / feminine “m” / masculine		
		<num> / number	“c” / common “p” / plural “s” / singular		
		<postype> / PoS subclassification	“article” “demonstrative” “exclamative” “indefinite” “interrogative” “numeral” “ordinal” “possessive”	"yan, wiss" "yan"	for demontrative: “proximity” “distance” “absence”
			“presentative” “quantifier” “other”		
6	<ad>/ adverb	<postype> / PoS subclassification	“place” “time” “quantity” “manner” “interrogative” “other”		

7	<s> / preposition	<gen> / gender	“c” / common		
			“m” / masculine “f” / feminine		
		<num> / number	“c” / common “p” / plural “s” / singular		
		<person>/ possessor's person (only for	“1” / first person “2” / second “3” / third		
		<possessornum>/ possessor’s number	“p” / plural “s” / singular		
		<possessorgen> / possessor’s gender	“f” / feminine “m” / masculine		
8	<c> / conjunction	<postype> / PoS subclassification	“coordinating” “subordinating”		
9	<i> / interjection				
10	<pa> / particle	<postype> / PoS subclassification	“aspect” “orientation” “predicate” “negative” “vocative” "interrogative"		

11	<foc> / focalizer				
12	<f>/punctuation	<punct> / punctuation mark type	“apostrophe” / [‘] “bracket” / [(], [)] “sqbracket” / [[], []] “cubracket” / [{], [}] “colon” / [:] “comma” / [,] “etc” / […] meaning etcetera “exclamation mark” / [¡], [!] “hyphen” / [-] “mathsign” / a sign used in mathematic formule “period” / [.] “question mark” / [¿], [?] “quotation” / [‘], [’], [“], [”] “semicolon” / [;] “slash” / [/] “revslash” / [\]	<punctenclose> / opens or closes the punctuation mark (only for “bracket”, “sqbracket”, “cubracket”, “exclamationmark”, “questionmark”, “quotation”)	“open” “close”
13	<r> / residual	<postype> / PoS subclassification	“foreing” “mathematical” “number” “date” “other”		

Annexe 2: Exemple de texte annoté

Ce paragraphe est un extrait d'un texte sur "tamGra" [mariage], l'un des textes recueillis décrit dans la section 2.6. Il montre le résultat de l'application du jeu d'étiquettes à un échantillon de l'amazighe.

Portion du texte original (en tifinaghe Unicode):

ⵜⵍⵍⴰ ⵜⵎⵖⵔⴰ ⴷⴰⵔ ⵡⴰⴷⵊⴰⵔⵏ ⵏⵏⵖ ⵉⵔⴰ ⵓⵔⴱⴰ ⵏⵏⵙⵏ ⴰⴷ ⵉⵜⴰⵀⵍ. ⴰⵔ ⴰⵙ
ⵜⵜⵃⵢⵢⴰⵍⵏ ⵉ ⵜⵎⵖⵔⴰ ⴰⵏⵏ ⵙⴳ ⵓⵙⴳⴳⵯⴰⵙ ⵍⵍⵉ ⵉⵣⵔⵉⵏ. ⵙⵖⴰⵏ ⴽⵉⴳⴰⵏ ⵏ
ⵉⴼⵛⴽⴰⵏ ⵖⵔⵏ ⵉ ⴽⵉⴳⴰⵏ ⵏ ⵎⴷⴷⵏ, ⵙⵏⴰⵜ ⵜⵎⴰⴹ.

Texte amazighe écrit selon la transcription prédéfinie dans la section 2.6: tlla tmGra dar Wadjarn nnG. ira urba nnsn ad itahl. ar as ttHyyaln i tmGra ann sg usggwas lli izrin. sGan kigan n ifckan Grn i kigan n mddn, snat tmaD.

tlla_ili-verb-perfective-f-s-3 tmGra_tamGra-noun-common-f-s-construct dar_ dar-prep Wadjarn_ adjar-noun-common-m-p-construct nnG_nnG-det-possessive-c-p-1 ._punct-period ira_iri-verb-perfective-m-s-3 urba_arba-noun-commun-m-s-construct nnsn_nnsn-det-possessive-m-p-3 ad_ ad-particle-preverbal itahl_ tahl- verb-aorist-m-s-3 ._.-punct-period ar_ ar-particle- preverbal as_ prep-pronounpGen:c-pronounNum:s-3 ttHyyaln_ Hyyl-verb-imperfective-m-p-3 i_i- prep tmGra_ tamGra-noun-commun-f-s-construct ann_ann-det-demonstrative-distance sg_sg-prep usggwas_asggwas-noun-common-m-s-construct lli_ lli-pron-relative izrin_zri- verb-participle-perfective ._.- punct-period sGan_sG-verb-perfective-m-p-3 kigan_kigan-det-quantity n_n-prep ifckan_afcku- noun-common-m-p -construct ,_ punct-comma Grn_ Gr- verb-aorist-m-p-3 i_ prep kigan_ det-quantity n_n-prep mddn_ middn- noun-common-m-p -construct ,_,-punct-comma uggar_uggar-det-quantity n_ n-prep snat_sin-noun-numeral-f-p -construct tmaD_timiDi- noun-numeral-f-p -construct ._.-punct-period

Annexe 3: Publications

Le travail de recherche présenté dans le cadre de cette thèse à été publié dans plusieurs revues et conférences. Ces publications sont :

1. Outahajala M., Benajiba Y., Rosso P., Zenkouar L. 2015. Using Confidence And Informativeness Criteria To Improve POS Tagging In Amazigh. In *Journal of Intelligence and Fuzzy Systems 28*, pp. 1319—1330. doi: 10.3233/IFS-141417.

2. Outahajala M., Zenkouar L., Benajiba Y., Rosso P. 2014. Utilisation des CACs et des Ressources Externes pour l'Amélioration des Performances de l'Étiquetage Morphosyntaxique. *La revue ASINAG*, Special Issue on ICT and Amazighe, vol. 9.

3. Outahajala, M., Zenkouar, L., Rosso, P. 2014. Construction d'un grand corpus annoté pour la langue amazighe. Revue Etudes et Documents Berbères n°33, pp. 57--74.

4. Outahajala, M., Zenkouar, L., Benajiba, Y. and Rosso, P. 2013. The Development Of A Fine Grained Class Set For Amazigh POS Tagging. Computer Systems and Applications (AICCSA), 2013 ACS International Conference. doi: 10.1109/AICCSA.2013.6616440.

5. Outahajala, M., Benajiba, Y., Rosso, P., & Zenkouar, L. 2012. L'étiquetage grammatical de l'amazighe en utilisant les propriétés n-grammes et un prétraitement de segmentation. *e-TI - la revue électronique des technologies d'information*, Numéro 6. (http://revue-eti.net/).

6. Outahajala, M., Zenkouar, L. 2012. Etiquetage grammatical de l'amazighe marocain en utilisant les techniques d'apprentissage supervisé. In *Proceedings of CNPLET 2012* Boumerdès, Algérie.

7. Outahajala, M., Benajiba, Y., Rosso, P., & Zenkouar, L. 2011. POS Tagging In Amazigh Using Support Vector Machines And Conditional Random Fields. In *Natural Language to Information Systems LNCS (6716)*, Springer-Verlag, pp. 238--241. doi:10.1007/978-3-642-22327-3_28.

8. Outahajala M. 2011. Processing Amazigh language. In *Natural Language to Information Systems LNCS (6716)*, pp. 313-317. doi: 10.1007/978-3-642-22327-3_46.

9. Outahajala, M., Zenkouar, L., & Rosso, P. 2011. Building An Annotated Corpus For Amazighe. In *Proceedings of 4th International Conference on Amazigh and ICT*. Rabat, Morocco.

10. Outahajala, M. and Benajiba, Y. and Rosso, P. and Zenkouar, L. 2011. POS Tagging in Amazigh using Tokenization and n-gram Character Feature Set. In Proceedings of International Symposium Traitement Automatique de la Culture Amazighe, SITACAM-2011, Agadir, Morocco.

11. Outahajala, M., Zenkouar, L., Rosso, P., & Martí, A. 2010. Tagging Amazigh With AncoraPipe. In *Proceedings of the Workshop on LR & HLT for Semitic Languages, 7th International Conference on Language Resources and Evaluation*, LREC'10, Malta, May 17-23, pp. 52--56.