Le TAL au service desenseignants des langues : mise en oeuvre d'une plate-forme pour l'enseignement du français et de l'arabe, langues étrangères.

Abdelkarim Mars

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Informatique et sciences du langage**

Arrêté ministériel : 7 août 2006

Présentée par

Abdelkarim Mars

préparée au sein du **Laboratoire LIDILEM - EA 609**
dans **l'École Doctorale n°50 - Langues Littérature et Sciences Humaines**

Le TAL au service des enseignants des langues : mise en œuvre d'une plate-forme pour l'enseignement du français et de l'arabe, langues étrangères.

Thèse soutenue publiquement le **21 octobre 2016**,
devant le jury composé de :

M. Georges Antoniadis
Professeur, Université Grenoble Alpes, Directeur de thèse
Mme Lamia Hadrich Belguith
Professeur, Université de Sfax, Tunisie, Rapporteur, Présidente
M. Cédrick Fairon
Professeur, Université catholique de Louvain, Belgique, Rapporteur
M. Thibault Carron
Maitre de conférences HDR, Université de Savoie Mont Blanc, Examinateur
M. Mathieu Loiseau
Maitre de conférences, Université Grenoble Alpes, Examinateur
Mme Amel Yessad
Maitre de conférences, Université Pierre et Marie Curie, Examinateur

Le TAL au service des enseignants des langues : mise en œuvre d'une plate-forme pour l'enseignement du français et de l'arabe, langues étrangères.

Abdelkarim Mars

Remerciements

Je remercie en tout premier lieu Georges Antoniadis, Professeur à l'Université du Grenoble Alpes, de m'avoir accompagné au cours de ces recherches. Sa confiance en moi et son enthousiasme communicatif m'ont permis de mener à bien ces travaux, m'aidant dans les difficultés et me laissant libre d'avancer à ma façon.

Je remercie Lamia Hadrich Belguith, Professeur à l'Université de Sfax Tunisie, de m'avoir fait l'honneur d'accepter d'être rapportrice de cette thèse.

Je remercie Cédrick Fairon, Professeur à l'Université catholique de Louvain Belgique, de m'avoir fait l'honneur d'accepter d'être rapporteur de cette thèse.

Je remercie Mathieu Loiseau, Maître de Conférences à l'Université du Grenoble Alpes, de m'avoir fait l'honneur de faire partie de mon jury de thèse.

Je remercie Amel Yessad, Maître de Conférences à l'Université Pierre et Marie Curie, de m'avoir fait l'honneur de faire partie de mon jury de thèse.

Je remercie Thibault Carron, Professeur à l'Université de Savoie Mont Blanc, de m'avoir fait l'honneur de présider le jury de ma soutenance de thèse.

Je remercie l'ensemble des membres du laboratoire avec qui j'ai partagé d'agréables moments et qui m'ont aidé soit au cours de la thèse, soit lors de la préparation de la soutenance.

Pour finir, je tiens à remercier mes proches. Merci à mes très chères frères, à ma chère soeur. Un grand merci à mes parents pour m'avoir entouré de leur aide et pour m'avoir offert une éducation qui m'a permis d'aboutir à cette thèse, et à ma chère femme.

« Louange à Dieu, le seul et unique »
À mes très chers parents...
À ma femme
Témoignage d'affection et de grande reconnaissance,
À toute ma famille,
À mes très chers soeurs et frères.

Résumé

Aujourd'hui, l'apprentissage des langues assisté par ordinateur est de plus en plus répandu, dans les institutions publiques et privées. Cependant, il est encore loin des attentes des enseignants et des apprenants et ne répond pas encore à leurs besoins. Les systèmes d'apprentissage des langues assisté par ordinateur (ALAO) actuels sont plutôt des environnements de tests des connaissances de l'apprenant et ressemblent plus à un support d'apprentissage traditionnel. De plus, le feedback proposé par ces systèmes reste basique et ne peut pas être adapté pour un apprentissage autonome, car, il devrait être en mesure de diagnostiquer les problèmes d'un apprenant avec l'orthographe, la grammaire, la conjugaison, etc., puis générer intelligemment un feedback adéquat selon la situation de l'apprentissage.

Cette recherche expose les capacités des outils TAL à apporter des solutions aux limitations des systèmes d'ALAO dans le but d'élaborer un système d'ALAO complet et autonome. Nous présentons une architecture complète d'un système multilingue pour l'apprentissage des langues assisté par ordinateur destiné aux apprenants des langues étrangères, français et arabe. Ce système pourrait être utilisé pour l'apprentissage des langues par les apprenants de la langue en tant que langue seconde ou étrangère.

La première partie de nos travaux porte sur l'adaptation des outils et des ressources issues du TAL pour qu'ils soient utilisés dans un environnement d'apprentissage des langues assisté par ordinateur. Parmi ces outils et ressources, il y a les analyseurs morphologiques pour l'arabe et le français, corpus, dictionnaires électroniques, etc. Ensuite, dans la deuxième section, nous présentons la reconnaissance de l'écriture manuscrite en ligne. Dans cette optique, nous exposons une approche statistique basée sur le réseau de neurones, puis, nous présentons la conception de l'architecture du système de reconnaissance ainsi que l'implémentation de l'algorithme de la reconnaissance.

La deuxième partie de notre exposé porte sur l'élaboration, l'intégration et l'exploitation des outils TAL utilisés (analyseurs morphologiques, système de reconnaissance de l'écriture, dictionnaires, etc.) dans notre système d'apprentissage des langues assisté par ordinateur. Nous y présentons aussi les modules ajoutés à la plate-forme pour avoir une l'architecture complète d'un système d'ALAO. Parmi ces modules, figure le générateur de feedback qui permet de corriger les fautes des apprenants et générer un feedback pédagogique

pertinent qui permet à l'apprenant de cerner et ses fautes. Enfin, nous décrivons l'outil de génération automatique des activités pédagogiques variées et automatisées.

Mots clés

ALAO, TAL, Analyse morphologique, feedback pédagogique, langue arabe, réseau de neurones, reconnaissance automatique de l'écriture manuscrite, activités pédagogiques.

Abstract

Today, computer-assisted language learning is becoming more prevalent in public and private institutions. However, it is still far from the expectations of teachers and learners and still does not response to their needs. Existing computer-assisted language learning systems (CALL) are considered a test environments for learner knowledge and more like a traditional learning medium. Moreover, the feedback provided by these systems remains basic and cannot be adapted for autonomous learning because it should be able to diagnose a learner's problems with spelling, grammar, conjugation, then intelligently generate adequate feedback depending on the learning situation.

This research exposes the capabilities of NLP tools to bring solutions to the limitations of CALL systems in order to develop a complete and autonomous CALL system. We present a complete architecture of a multilingual system for computer-assisted language learning for learners of foreign languages, French and Arabic. This system could be used to learn languages by learners of the language as a second or foreign language.

The first part of our work deals with the adaptation of the tools and resources resulting from the NLP to be used in a computer-assisted language learning environment. Among these tools and resources we will use a morphological analyzers for Arabic and French, corpus, electronic dictionaries, etc. Then, in the second section, we present the online handwriting recognition system. In this perspective, we present a statistical approach based on the neural network, then we present the design of the architecture of the recognition system as well as the implementation of the recognition algorithm.

The second part of our paper deals with the development, integration and exploitation of the NLP tools (morphological analyzers, handwriting recognition system, dictionaries, etc.) used in our language learning system assisted by computer. We also present the modules added to the platform to have a complete architecture of a CALL system. Among these modules is the feedback generator which helps to correct the errors of the learners and generate a relevant pedagogical feedback which allows the learner to identify the error. Finally, we describe the tool for automatic generation of varied and automated teaching activities.

Key words

CALL, NLP, Morphological analysis, pedagogical feedback, Arabic language, neural network, handwriting recognition system, educational activities.

Table des matières

Liste des figures

Liste des tableaux

Chapitre 1 : Contexte d'étude

1 Introduction

Dans la dernière décennie, le domaine d'apprentissage des langues assisté par ordinateur (ALAO) a connu un véritable progrès en matière de diversité et de techniques employées. Dans ce domaine, la recherche a servi à orienter l'ALAO vers les directions les plus prometteuses. Dans cette optique, nous ambitionnons, dans cette thèse, l'élaboration d'un système pour l'apprentissage des langues assisté par ordinateur destiné aux apprenants de la langue arabe et de la langue française, en tant que langues étrangères. Cette thèse se situe dans le cadre du projet MIRTO (Multi-apprentissages Interactifs par des Recherches sur des Textes et l'Oral) (Antoniadis 2010) du laboratoire de LIDILEM1 à l'université Grenoble-Alpes. L'objectif de ce projet est la réalisation d'un environnement intelligent multilingue pour l'apprentissage de langues assisté par ordinateur pour l'apprentissage du français et d'arabe.

2 Contexte d'étude

Nos travaux au cours de cette thèse s'inscrivent dans les domaines d'apprentissage des langues assisté par ordinateur (ALAO) et du traitement automatique de langues (TAL), dans le cadre du projet MIRTO (Antoniadis, 2010).

L'objectif principal du projet MIRTO est la conception d'un système multilingue destiné à l'enseignement des langues en utilisant :

[1] http://lidilem.u-grenoble3.fr/

- L'ensemble de logiciels TAL issus des recherches scientifiques et des laboratoires.
- La diversité et la richesse des corpus textuels ou oraux.
- Un ensemble de fonctions TAL (une fonction TAL est obtenue à partir d'un logiciel TAL) (Antoniadis et al, 2005).

Notre but principal dans cette thèse sera la définition et la conception de l'architecture complète d'un système d'ALAO pour le français dans un premier temps, puis son extension pour la langue arabe, selon les mêmes principes qui ont prévalus à la définition et l'élaboration d'un système pour l'enseignement du français comme langue étrangère.

3 Projet MIRTO

3.1 Présentation

D'une manière générale, nous considérons MIRTO (Multi-apprentissages Interactifs par des Recherches sur des Textes et l'Oral) comme une plate-forme de création d'activités pédagogiques basée sur des outils TAL en développement au sein de notre laboratoire. On peut trouver la description du projet dans plusieurs publications, telles que (Antoniadis et al, 2013), (Antoniadis, 2010), (Antoniadis et al, 2002).

L'objectif principal de MIRTO est d'apporter une réponse globale aux problématiques des systèmes d'ALAO en utilisant les technologies TAL et en se fondant sur un travail de collaboration avec des enseignants de la langue, des pédagogues et des didacticiens. De plus, MIRTO devrait permettre l'implantation de fonctions TAL classiques au sein de la plate-forme afin de faciliter la conception, sans compétence informatique préalable, d'activités didactique (Antoniadis et al, 2005).

D'autre part, ce projet devrait apporter de nouvelles possibilités telles que le travail sur de longs textes, la génération automatique d'exercices ou d'aides, la conception de scénarios non linéaires, etc. Aussi, la plate-forme devrait être ouverte pour tout autre modification ou ajout d'autres logiciels, principalement TAL. Par contre, le choix du nombre et de la nature des logiciels intégrés exige un processus d'échange impliquant des enseignants de langue et des experts du TAL.

3.2 État actuel et architecture de MIRTO

Actuellement, les modules existants du projet MIRTO sont destinés à l'apprentissage de la langue française. D'autre part, un site web a été créé permettant la génération de

certaines activités (QCM, questions réponses, etc.). Ce site faisait appel à l'analyseur morphologique de Xerox pour générer les activités. Cet analyseur n'est plus supporté par la société Xerox, donc on peut plus obtenir un nouveau licence, par la suite une des premières tâches de notre travail consiste à remplacer l'analyseur de Xerox par un analyseur libre de droits, afin de créer une plate-forme open source.

Jusqu'ici, le développement de MIRTO reste limité à un prototype de démonstration pour la langue française. Ce prototype peut être décrit comme deux modules principaux : le module TAL et le module didactique. Le premier module permet aux spécialistes du TAL la création des scripts. En utilisant ces scripts, le module didactique offre aux professeurs de langues un système de création afin de créer des activités et des scénarios (figure 1). D'autre part, la structure de MIRTO est composée de quatre niveaux hiérarchiques fonction, scripts, activité et scénario.

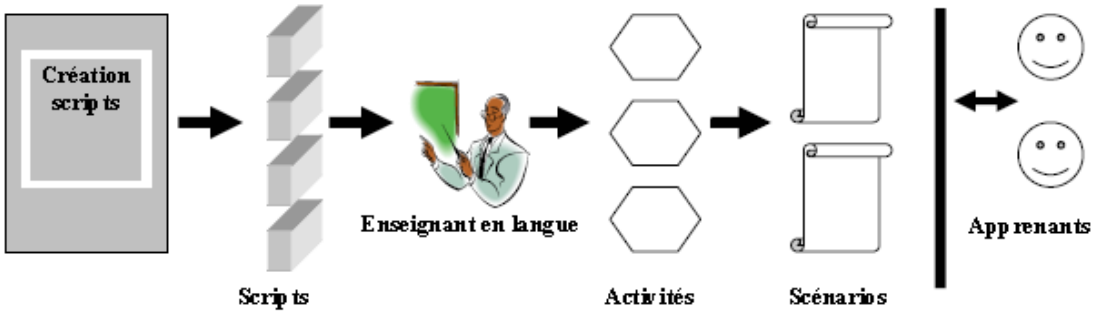

Figure 1 : Schéma fonctionnel de MIRTO (Antoniadis et al, 2004)

3.1.1 Niveau fonction

Les fonctions représentent le niveau inférieur de MIRTO. Ils correspondent à un processus TAL de base telles que tokenisation ou l'identification de la langue. Comme les fonctions sont indépendantes par rapport à une application didactique, ce niveau n'est pas visible par les utilisateurs.

3.1.2 Niveau script

Ce niveau correspond à l'application d'une ou de plusieurs fonctions TAL pour répondre à un objectif didactique. Par exemple, la conception automatisée d'un exercice « texte à trous » est considérée comme un script, car il relie les fonctions d'identification de la langue, la tokenisation, l'analyse morphologique et la création des lacunes en fonction des

paramètres choisis par l'utilisateur. Le travail d'un script est caché à l'enseignant concepteur de scénarios, car il se présente comme une boîte à outils.

3.1.3 Niveau activité

Ce niveau avec le niveau de scénario présente le noyau didactique de MIRTO. Une activité correspond à la contextualisation didactique d'un script (niveau précédent). Son but est d'associer un script avec un texte de la base de données, une instruction, les aides possibles et une évaluation facultative système. Afin de créer un exercice texte à trou, il suffit de choisir d'appliquer le script de l'exemple précédent à un texte tout en précisant les critères de lacunes (par exemple, en cachant les verbes à l'imparfait et en les remplaçants par leur forme infinitif), et en associant une instruction comme " Remplir les trous avec la forme prétérit " (figure 2). La définition des activités est réalisée par l'enseignant grâce au module de création des activités.

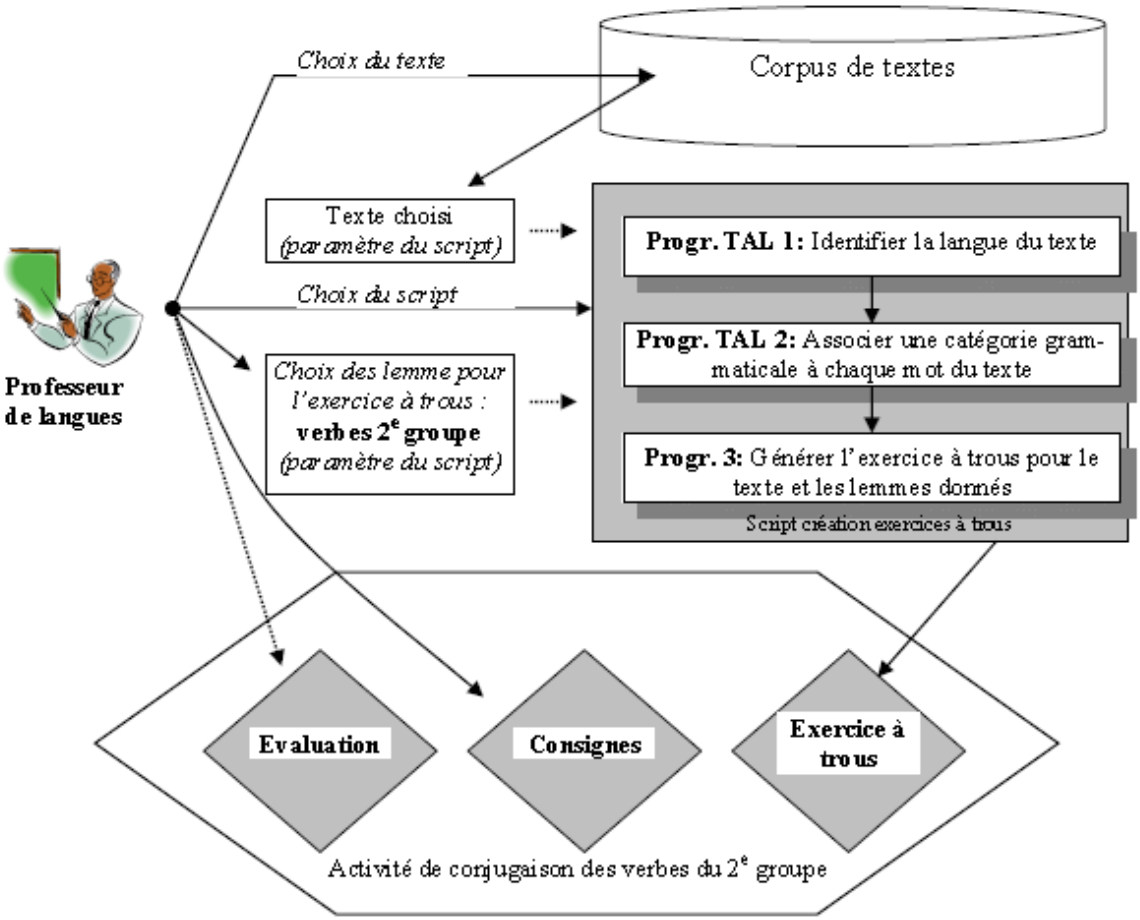

Figure 2 : Conception d'activités (Antoniadis et al, 2004)

3.1.4 Niveau scénario

Ce niveau permet aux enseignants de définir la séquence des activités afin de répondre à leurs objectifs pédagogiques selon la progression de l'apprenant (tableau 1). Cette progression attendue diffère d'un apprenant à un autre. En effet, chaque apprenant aura un processus d'apprentissage personnel lié à différents facteurs. Chaque scénario dépend du processus individuel de chaque apprenant (cours d'apprentissage, d'évaluation ...). Par exemple, selon sa progression dans un scénario donné, un apprenant peut être redirigé vers des activités plus difficiles, ou réessayez une activité sur un autre texte ou simplement avancer dans le scénario.

Niveau	Opération	Utilisateur
Fonctions	Conception	Spécialiste TAL
Scripts	Conception	Spécialiste TAL + Enseignant des langues
Activités	Conception	Enseignant des langues
Scénarios	Conception Utilisation	Enseignant des langues Apprenant

Tableau 1 : Type d'utilisateur et opérations associées (Antoniadis, 2010)

4 L'apprentissage des langues

La première apparition d'une machine destinée à l'apprentissage se situe en 1809, aux USA, où H. Chard a fabriqué la machine « Mode of Teaching Reading » destinée à l'enseignement de la lecture. À partir de cette date, il y a eu plusieurs tentatives pour créer des machines destinées à l'apprentissage. Le but de tous ces travaux est de faire évoluer et progresser le domaine de l'enseignement en général. Dans cette optique, les machines destinées à l'apprentissage devraient apporter des solutions en compléments à l'enseignement traditionnel (Bertin, 2001). Actuellement, le domaine de l'apprentissage des langues est toujours en évolution, et il devient pluridisciplinaire, car il englobe la linguistique, la sociologie, la psychopédagogie, l'informatique, la culture et la communication (Defays et al, 2003).

Selon (Warschauer, 1996), le but essentiel de l'apprentissage des langues assisté par ordinateur est de mettre à la disposition des enseignants et des apprenants des outils informatiques pour l'enseignement. Cela permet de fournir aux apprenants des activités accompagnées d'un feedback et, par la suite, un environnement réaliste et personnalisé. De plus, l'utilisation de l'ALAO facilite l'interaction entre l'apprenant et le système et entre l'apprenant et l'enseignant.

Bien que ce domaine ait bien progressé ces dernières années, il reste encore beaucoup des problèmes non résolus. En étudiant l'état de l'art de l'ALAO, nous avons remarqué que la majorité de systèmes existants sont statiques, limités en nombre et en variétés d'activités, avec un feedback basique, etc. De plus, un système d'ALAO doit être capable de détecter les erreurs en proposant un feedback adéquat avec des explications et des corrections. Cependant, les systèmes actuels n'arrivent pas à gérer ce point indispensable lors d'une situation d'apprentissage. D'autre part, les apprenants commettent plusieurs types d'erreurs, qu'il est parfois difficile d'anticiper, surtout lorsque les productions des apprenants sont libres et ne présentent pas des restrictions liées à la structure.

Dans le cadre de cette thèse, nous nous intéressons aux limites des systèmes d'ALAO et nous proposons comme solution l'utilisation et l'intégration des outils et des ressources issus du TAL.

5 Avantages de l'utilisation du TAL dans l'ALAO

Les systèmes d'ALAO conventionnels sont limités le plus souvent, soit par la répétitivité du matériel d'apprentissage, soit par le manque de liberté offerte aux apprenants, ne permettant la plupart du temps de travailler que sur des activités prédéfinies.

Le Traitement Automatique des Langues (TAL) est considéré comme un domaine de l'Intelligence Artificielle (IA) spécialisé dans le traitement des divers aspects de la langue, tels que le niveau lexical, le niveau morphologique, le niveau sémantique, le niveau syntaxique, ou le niveau pragmatique.

Dans la dernière décennie, l'utilisation de TAL dans des systèmes d'ALAO multilingues devient un axe de recherche important dans la communauté du Traitement Automatique de la Langue (Jung, 2005). De plus, la communauté TAL a créé depuis quelques années plusieurs colloques, ateliers, revues et des sessions spéciales, par exemple TALN[2]

[2] http://www.taln2014.org/site/

(Traitement Automatique du Langage Naturel), ALSIC[3] (Apprentissage des langues et systèmes d'information et de communication), System[4], LREC[5] (Language Resources and Evaluation Conference), etc. Le but principal de ces travaux est de faire évoluer le domaine du TAL et le domaine d'ALAO. La majorité de ces travaux vise à créer des outils et ressources TAL dans le but de les mettre à la disposition des enseignants et des spécialistes d'ALAO. Dans cette optique, nous allons utiliser les solutions et les possibilités des résultats du TAL, les adapter pour l'ALAO, afin d'apporter une solution à certaines limites des systèmes d'ALAO. Bien qu'il existe quelques travaux et ressources disponibles pour le français, on ne trouve que peu des ressources linguistiques disponibles en arabe. De plus, vu la complexité des scripts arabes, les outils TAL arabes sont rares et leur qualité est moyenne. Par conséquent, une partie de notre thèse sera consacrée à la construction des ressources linguistiques arabes et à l'amélioration de certains outils existants.

6 La reconnaissance automatique de l'écriture

La reconnaissance automatique de l'écriture manuscrite est la transcription de données manuscrites en texte numérique, pour une utilisation par l'ordinateur. Ce domaine « débute » dans les années 1950.

Il existe plusieurs types d'écritures manuscrites, tels que les formules mathématiques manuscrites, ou l'écriture manuscrite cursive. L'écriture des caractères imprimés ou cursifs concerne des mots entiers, ou des caractères distincts, ou des combinaisons de caractères cursifs et caractères séparés. Ces variétés d'écriture affectent la qualité de la reconnaissance du système et elles causent beaucoup de problèmes lors de la phase de la segmentation du mot en caractères.

Actuellement, il existe des applications basiques pour l'apprentissage de l'écriture en utilisant des technologies Multimédias (vidéo, flash, etc.). Ces applications proposent des activités, le plus souvent, basiques et rigides, c'est-à-dire les activités sont souvent statiques et définies à l'avance. De plus, elles n'ont pas de feedback qui s'adapte aux différentes erreurs et aux différentes situations d'apprentissage. Cependant, on ne trouve pas une application complète pour l'apprentissage de l'écriture basée sur des outils TAL. L'intégration d'un

[3] http://alsic.revues.org/

[4] http://www.journals.elsevier.com/system/

[5] http://lrec2014.lrec-conf.org/en/

système de reconnaissance de l'écriture manuscrite dans un système d'ALAO peut être considérée comme une innovation dans le domaine de l'apprentissage d'une langue.

7 L'architecture du système d'ALAO

7.1 Architecture actuelle

Actuellement, il existe quelques systèmes d'ALAO avec des architectures différentes. Cependant, la majorité de ces systèmes ne sont pas complets, et ils sont conçus pour un but bien précis (apprendre la grammaire, apprendre l'orthographe, etc.). D'autre part, ces architectures restent traditionnelles, réduites à quelques modules de base et ne proposent qu'un nombre très limité des activités. On trouve aussi des plates-formes qui ont une architecture presque complète, mais elles n'utilisent ni le TAL ni le feedback automatique telles que Claroline[6], Moodle (Laforcade et al, 2012), Ganesha (Abedmouleh et al, 2012), etc. Cependant, les plateformes qui utilisent les outils et ressources TAL restent à l'état actuel basique au niveau de leurs architectures internes, parmi ces plateformes on trouve SALA (Maraoui et al, 2006), le système de (Shaalan, 2003), le système de (Volodina et al, 2012), etc.

7.2 Définition de l'architecture complète

Après la présentation de MIRTO, notre but dans cette thèse est de faire évoluer le système MIRTO afin d'obtenir un système mené d'une architecture complète pour le français dans un premier temps, puis nous ambitionnons l'ajout de la langue arabe en respectant la même architecture.

Dans cette partie, nous décrivons l'architecture complète de notre système d'ALAO qui nous devons implanter lors de cette thèse. Ce système est composé de plusieurs modules (voir figure 3) : l'interface utilisateur (enseignant et apprenant), le générateur des activités, l'analyseur morphologique, le système de la reconnaissance de l'écriture, le module de feedback et l'indexation pédagogique de texte.

[6] http://www.claroline.net/

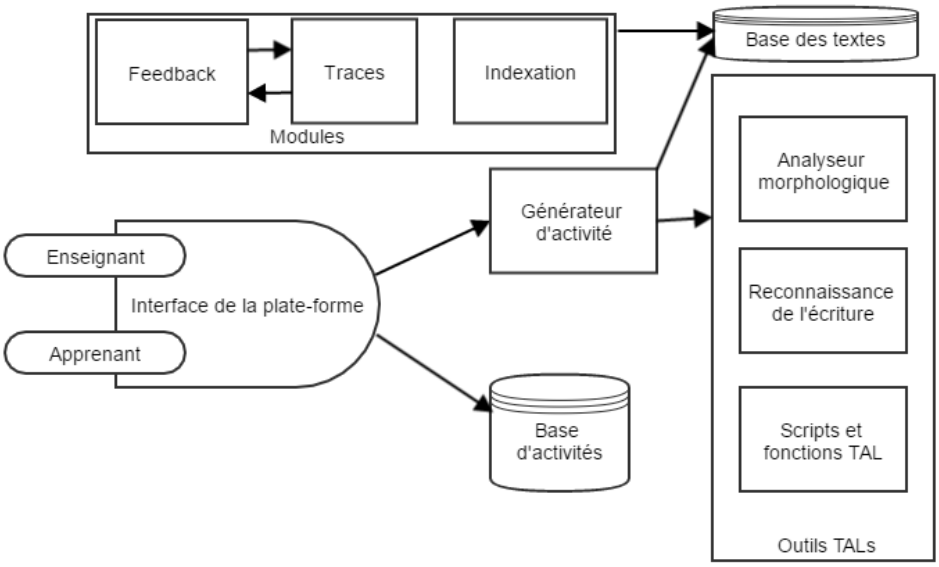

Figure 3 : Architecture d'un système complet d'ALAO

- Module de feedback : un élément essentiel pour un apprentissage efficace de la langue est le feedback. Notre système devrait embarquer un module de feedback qui permet d'évaluer les réponses des apprenons et proposer un feedback pédagogique adéquat.
- Interface utilisateur : le système est destiné aux enseignants des langues et aux apprenants, donc nous trouvons deux interfaces : d'une part, une interface pour l'enseignant, elle lui permet de réaliser toutes les gestions et les paramétrages des activités didactiques et linguistiques ; d'autre part, une interface pour les apprenants de la langue, elle lui permettre de faire les activités et trouver les supports didactiques et linguistiques proposés.
- L'analyseur morphologique : est un outil d'annotation morphosyntaxique des textes. Il permet la génération des textes étiquetés morphologiquement à partir duquel nous générons les activités.
- Système de la reconnaissance de l'écriture : il permet la transformation de l'écriture manuscrite en écriture digitale. Cela nous permet l'interprétation de l'écriture de l'apprenant lors d'une activité d'écriture (Mars et al 2015).
- Le générateur des activités : permet d'offrir aux enseignants des langues la possibilité de concevoir facilement des activités pédagogiques en bénéficiant pleinement d'outils TAL mise à disposition de notre système d'ALAO (l'analyseur morphologique, le système de reconnaissance de l'écriture, etc).

- L'indexation pédagogique de texte : ce module devrait permettre aux enseignants des langues d'accéder à une base des données des textes indexé selon le thème (Mars et al, 2014).

La conception de l'architecture de la plate-forme doit prendre en compte l'aspect dynamique c'est-à-dire l'architecture du système devrait être dynamique pour s'adapter aux évolutions de l'environnement (ajout d'autres modules TAL, ajout d'autres fonctionnalités, mise à jour des outils, etc.).

8 Objectifs de la thèse

L'objectif principal de notre travail est de proposer l'architecture d'un système d'ALAO destinée aux enseignants prenant en compte la langue française et la langue arabe. Ce système est basé sur les technologies TAL. Nous souhaitons intégrer des fonctionnalités qui réponds aux besoins et aux problématiques des enseignants des langues. Pour arriver à cet objectif, nous avons identifié les étapes suivantes :

- Comme nous traitons dans une partie de cette thèse la langue arabe pour laquelle les outils TAL sont rares, nous souhaitons présenter et étudier les caractéristiques de la langue arabe ainsi que ses problématiques.
- Dans le but d'enrichir la plate-forme par plusieurs variétés d'activités, nous souhaitons élaborer un système multilingue pour la reconnaissance de l'écriture manuscrite en ligne. Cela nous permettre de générer plusieurs types d'activité, avec un feedback, destinées à l'apprentissage de l'écriture manuscrite.
- Vu l'état des outils existant pour l'arabe, un travail est fait pour la comparaison des analyseurs morphologiques existants dans un premier temps afin de choisir un analyseur qui donne des résultats acceptable par rapport à l'existant. Une fois le bon analyseur choisi, un autre travail d'amélioration sera fait pour augmenter les performances de l'analyseur sélectionné.
- Le travail inclut, également, l'étude et la comparaison des différents analyseurs morphologiques existants pour le français dans le but d'en choisir un pour notre plate-forme.
- Nous souhaitons intégrer un outil de correction orthographique basée sur un lexique riche qui sera intégré au module de feedback. L'utilisation d'un lexique permet au système de corriger les fautes d'orthographe.

- Le dernier objectif est la conception et la réalisation d'une plate-forme multilingue complète (français et arabe) pour l'ALAO.

9 Organisation de la thèse

Le contenu principal de cette thèse est divisé en neuf chapitres. Le premier chapitre expose la problématique de l'ALAO, les objectifs de la thèse, les solutions proposées, et, enfin, nos contributions.

Le deuxième chapitre est composé de trois parties. Dans la première, nous présentons le domaine d'ALAO, nous décrivons ses enjeux et l'état de l'art de l'apprentissage de la langue en présentant quelques systèmes d'ALAO existants. Dans la seconde partie, nous citons quelques contributions possibles des techniques et outils issus du TAL pour l'ALAO. Enfin, nous montrons les avantages et l'apport du TAL dans l'ALAO, en particulier pour l'arabe.

Dans le chapitre trois, nous présentons la langue arabe et ses caractéristiques, puis nous détaillons les particularités et les spécificités de la morphologie arabe.

Le quatrième chapitre expose l'état de l'art des travaux effectués pour la reconnaissance automatique de l'écriture manuscrite en ligne, ainsi qu'une présentation de la structure générale d'un système de reconnaissance, afin d'exposer les concepts principaux. Ensuite, nous décrivons l'approche choisie pour la réalisation de notre système : approche statistique basée sur le réseau de neurones à décalage temporel (Time Delay Neural Network).

Le chapitre cinq commence par une présentation des différentes démarches à suivre pour aboutir aux objectifs de cette thèse. Puis nous décrivons la composition et l'architecture de notre plate-forme.

Dans le chapitre six, nous décrivons les différentes étapes d'implémentation de notre système de reconnaissance de l'écriture. Par la suite, nous présentons notre approche de construction de nos bases de données d'échantillons manuscrits. Enfin, nous exposons les résultats de reconnaissance obtenus sur les bases de tests collectées.

Au septième chapitre, nous détaillons les démarches à suivre pour aboutir à notre système d'ALAO. Puis, nous présentons l'état de l'art des travaux effectués pour l'analyse morphologique pour l'arabe et le français. Ensuite, après la présentation des analyseurs existants, nous justifions notre choix de deux analyseurs et les étapes à suivre pour améliorer l'analyseur arabe et l'analyseur français. Enfin, nous décrivons les étapes suivies pour la construction des ressources lexicales.

Dans le huitième chapitre, nous présentons les différentes étapes de la construction de notre système d'ALAO. Dans un premier temps, nous décrivons l'architecture globale de notre système. Puis, nous décrivons le module de génération automatique des activités pédagogiques. Ensuite, nous abordons les étapes d'intégration et d'adaptation du système de reconnaissance de l'écriture et nous montrons l'intégration des modules du feedback, de la collecte des traces et de l'indexation pédagogique de textes, pour que l'architecture de la plate-forme soit complète. Enfin, nous décrivons le processus des tests réalisés pour valider l'architecture et le fonctionnement de la plate-forme.

En conclusion de ce document, dans le dernier chapitre, nous récapitulons les travaux et les contributions dans notre thèse, puis nous présentons les perspectives et les futurs travaux de recherche et collaborations.

Chapitre 2 : Le TAL au service de l'ALAO

1 Introduction

Dans le cadre de la formation à distance, l'utilisation des outils TAL dans l'apprentissage des langues assistée par ordinateur, que ce soit en formation initiale ou continue, devient de plus en plus importante. Les pédagogues s'intéressent de plus en plus à employer ces plates-formes dans leurs pédagogies d'enseignement (Robbes, 2009). Aussi, d'autres organismes proposent des outils pour faciliter la mise en place de ce type de plate-forme dédiée à l'enseignement à distance (Lebrun, 2007) (Kaleidoscope, 2005). Par contre, en ce qui concerne la pratique, la plupart des pédagogues ne peuvent se consacrer pleinement à leur rôle par manque du temps ou de connaissance de manipulation des EIAH (Environnement Informatique pour l'Apprentissage Humain) (Garrot, 2008).

Avec les progrès techniques, les systèmes d'ALAO incorporent de nombreux outils TAL (analyseurs morphologiques et syntaxiques, logiciels de traitement de corpus, etc.) et des ressources linguistiques. Il est donc indispensable de connaître l'influence de ces outils et de ces ressources issus du TAL dans le cadre d'activités pédagogiques : peuvent-ils générer des activités pédagogiques informatisées d'un type nouveau ? Si oui, quelles sont les caractéristiques de ces activités, leurs limites ? Et enfin la conception des systèmes d'ALAO permet-elle de mieux exploiter le travail de l'apprenant afin de générer un feedback adéquat ?

2 L'apprentissage de langue assisté par ordinateur

Dans cette section, nous allons définir l'Apprentissage de Langue Assisté par Ordinateur en présentant quelques systèmes d'ALAO ainsi que ses avantages et ses inconvénients.

2.1 Définition

L'ALAO (Apprentissage de Langue Assisté par Ordinateur) a émergé depuis le début des années 1960. De plus, l'ALAO est un domaine de recherche et développement (Levy, 1997). Il intéresse plusieurs branches intervenant dans les domaines des sciences cognitives tels que la linguistique, la psycholinguistique, la linguistique informatique et l'informatique (en particulier l'intelligence artificielle). De plus, ce terme est largement utilisé pour désigner le domaine de la technologie et de l'enseignement des langues secondes (Chapelle, 2001). Aussi, selon Beatty, l'ALAO est un processus dans lequel un apprenant utilise un ordinateur et, par conséquent, améliore son niveau d'apprentissage dans une langue donnée (Beatty, 2003). Afin de développer et faciliter l'apprentissage de la langue, nous avons besoin des technologies, des outils et des supports spécifiques. Parmi ces outils et supports, on peut citer, les dictionnaires, web 2.0, les concordanciers, le wiki, les applications pour générer des activités pédagogiques (quiz, texte lacunaire, traduction, écriture, jeux sérieux…). À cet effet, l'ALAO assigne des rôles principaux à l'ordinateur.

2.2 Défis de l'ALAO

L'ALAO apporte des solutions à plusieurs problématiques liées à l'enseignement des langues secondes. Par contre, les concepteurs des systèmes d'ALAO doivent respecter plusieurs conditions pédagogiques et didactiques liées à l'apprentissage des langues :

- Efficacité de l'apprentissage : le système devrait permettre aux apprenants une acquisition plus vite et avec moins d'efforts des connaissances et des compétences linguistiques.
- Accès : les apprenants doivent obtenir des matériaux ou des supports interactifs qui leur permettent de faciliter le processus d'apprentissage.
- Commodité : les apprenants peuvent étudier et pratiquer avec la même efficacité à travers un large intervalle de temps et de lieux.

- Motivation : les apprenants devraient apprécier le processus d'apprentissage des langues et être, donc, engagés avec beaucoup de motivations.
- efficacité institutionnelle : le temps pour consulter l'aide des enseignants devrait être court.

2.3 Avantages de l'ALAO

Dans un même ordre d'idées, l'ALAO présente plus d'avantages par rapport à l'enseignement classique (c'est-à-dire en classe) pour l'apprentissage des langues secondes.

L'enseignement classique de la langue en salle de classe peut être monotone, ennuyeux, et même frustrant, et les étudiants risquent de perdre l'intérêt et la motivation d'apprentissage (Murphy et al, 2011). La manière dont on utilise l'ALAO peut fournir des moyens aux étudiants d'apprendre la langue à travers des applications sous forme de jeux informatiques, graphiques animés, et des techniques de résolution de problèmes qui peuvent rendre les exercices plus intéressants (Ravichandran, 2000). De plus, l'ALAO peut réduire l'anxiété de l'apprenant en lui offrant un environnement d'apprentissage indépendant (Kongrith et Maddux, 2005). Selon le rapport de (Dat et al, 2005) sur l'intégration de l'ALAO dans l'enseignement des langues étrangères, cette forme d'apprentissage a un effet positif sur la motivation des apprenants, en particulier puisque l'anonymat est préservé, les apprenants se sentent moins sous pression et ils trouvent agréable l'expérience d'apprentissage.

2.3.1 Utilisation optimale du temps d'apprentissage

La flexibilité du temps de l'utilisation de l'ordinateur permet aux élèves de choisir le moment approprié pour l'apprentissage. (Kilickaya, 2007) a souligné l'importance de l'apprentissage souple (apprentissage n'importe où, n'importe quand, de toute façon, etc), ce que l'ALAO permet. Les apprenants peuvent étudier et examiner les matériaux autant de fois qu'ils veulent, sans limites de temps.

2.3.2 Feedback

Les étudiants bénéficient pleinement des avantages du feedback puisqu'il est donné immédiatement. Un feedback positif retardé risque de réduire l'encouragement et le renforcement de l'apprenant et un feedback négatif retardé peut affecter la connaissance cruciale que l'apprenant doit maîtriser. L'ordinateur peut donner un feedback instantané et

aider les élèves à mieux corriger ses idées fausses dès la première étape. Le feedback dans un système d'ALAO permet aux étudiants un apprentissage adapté à leur propre rythme, et de causer moins de frustration aux apprenants (Kilickaya, 2007).

2.3.3 Apprentissage répétitif

En utilisant un système d'ALAO, les apprenants peuvent reprendre le cours qu'ils veulent, cela aide l'apprenant à bien assimiler ce cours. De plus, les matériaux d'ALAO sont adaptés pour la pratique répétitive, cette pratique permet aux apprenants de bien réviser les concepts et les éléments clés dans un domaine précis.

2.4 État actuel de l'ALAO

Bien que l'ALAO offre des avantages à l'apprenant, l'état actuel de ce domaine présente encore des limites et des inconvénients. Parmi ces inconvénients on peut citer :

2.4.1 Accès difficile

Il est nécessaire que les enseignants et les apprenants aient une connaissance de la technologie de base avant d'utiliser l'ALAO dans l'enseignement et l'apprentissage d'une langue seconde. Par conséquent, il faut être habitué à la technologie informatique pour bénéficier des différents avantages de l'ALAO.

2.4.2 Systèmes actuels d'ALAO imparfaits

À l'heure actuelle, les systèmes d'ALAO traitent principalement la lecture, l'écoute, la compréhension et l'écriture, mais leurs fonctions sont encore limitées. En effet, un système d'apprentissage devrait être en mesure de diagnostiquer le problème de l'apprenant avec la syntaxe, la grammaire, l'orthographe.

2.4.3 Incapacité à gérer des situations inattendues

Les situations d'apprentissage auxquelles un apprenant peut faire face sont différentes et toujours en évolution. Les ordinateurs ne peuvent pas gérer le problème des situations d'apprentissage inattendues ou fournir immédiatement une réponse aux questions de l'apprenant comme le font les enseignants. Par conséquent, les systèmes d'ALAO actuels manquent d'interactivité entre l'apprenant et la machine.

2.5 Premiers systèmes d'ALAO

2.5.1 Machine de Pressey

La machine de Pressey est considérée comme la première machine « moderne » destinée à l'enseignement des langues. Cette machine a été élaborée par Sidney Pressey (Pressey, 1927) et a été conçue pour les questions à choix multiples (QCM), en utilisant quatre boutons, sachant que chaque bouton correspond à une réponse possible. Cette machine donne un feedback immédiat de type vrai ou faux.

2.5.2 Système PLATO

D'autres tentatives d'enseigner les langues étrangères spécifiques ont commencé dans les années 1950 et 1960, en utilisant des ordinateurs centraux (Beatty, 2003), mais le premier projet concret a été réalisé avec le système de PLATO (Programmed Logic for Automated Teaching Operation), développé à l'Université de l'Illinois, dans le cadre du projet « National Science Foundation ». La manière dont fonctionne ce système est la suivante : plusieurs enseignants de langue élaborent un cours en trois étapes : proposition des activités des vocabulaires structurés, fournir des explications grammaticales et évaluer la compréhension des apprenants. (Levy, 1997) note que les matériaux du PLATO ont été élaborés pour un certain nombre de langues, en particulier le français. Le système a été conçu pour sauvegarder toutes les traces liées aux activités, non seulement pour les étudiants, mais aussi pour les enseignants.

2.5.3 Système Montevidisco

Avec l'apparition de «micro-ordinateurs», tels que le BBC ordinateur, Apple II et PC IBM, des systèmes d'ALAO ont commencé à se développer dans les années 1980. Cette nouvelle vague a continué d'inclure les projets universitaires impliquant des équipes de designers, des programmeurs et des professeurs de langue, mais cette époque est également marquée par l'émergence d'enseignants programmeurs, utilisant le langage BASIC pour créer des activités pour leurs propres étudiants. Parmi ces projets, on cite le projet Montevidisco, qui a été développé par (Gale, 1989) à l'université de Brigham Young Utah. Il est conçu sous la forme d'un jeu vidéo interactif sur vidéodisque pour apprendre l'espagnol. Le jeu propose une narration arborescente dans lequel l'apprenant doit visité un village, à chaque fois ou il rencontre un personnage une conversation en espagnole commence en les obligeants à choisir

entre quatre réponses possibles. Le choix de la réponse entraîne une conséquence pour la suite de l'histoire.

2.5.4 Projet ATHENA

Le projet ATHENA est parmi les premiers grands projets utilisant l'ALAO, il a été développé en 1983 (Murray et al, 1989). Il introduit la notion de micro-mondes (microworlds) comme une interface pour l'apprentissage des langues (Felshin, 1995). ALLP (ATHENEA Language Learning Project) utilise le traitement automatique du langage naturel, le traitement de la parole et les vidéos interactives.

Dans le cadre d'ALLP, trois types d'exercices ont été mis en œuvre. Le premier est appelé LINGO, il était sous la forme d'une simulation de conversation avec un poltergeist[7]. Dans le deuxième exercice l'apprenant joue le rôle d'un journaliste qui doit interviewer divers acteurs. Le troisième type d'exercice est appelé le classeur intelligent, il s'agit d'une tâche de traduction. L'apprenant doit produire un énoncé espagnol correct en lui donnant des phrases anglaises sous forme d'aide.

2.5.5 Projet LISTEN

Ce projet a commencé en 1993, et il est encore un projet en cours à l'Université Carnegie Mellon (CMU). Le système a été utilisé expérimentalement par des apprenants de la langue anglaise (Poulsen, 2004). Le système utilise la technologie de la reconnaissance automatique de la parole pour « écouter » les apprenants lors d'une lecture d'un texte et déclencher automatiquement des interventions pédagogiques appropriées (Mostow et al, 1994). Le système dispose d'un agent personnalisé, "Emily", qui fournit des informations, et une assistance si nécessaire. Bien que le système ne soit pas conçu pour simuler le dialogue humain-machine, l'agent Emily intervient dans les cas suivants :

- Lorsque l'apprenant a mal interprété un ou plusieurs mots dans la phrase en cours.
- Lorsque l'apprenant se bloque.
- Lorsque l'apprenant clique sur un mot pour obtenir de l'aide.

[7] Phénomène spontané et répétitif, en ajoutant ou déplaçant des séquences sonores dans un enregistrement audio.

2.5.6 Systèmes commercialisés

En plus des initiatives universitaires, il y a eu quelques projets commercialisés de logiciels d'ALAO, dont quelques-uns ont réussi à survivre et à prospérer. Parmi ces projets, on cite le projet Tell Me More développé par la société Auralog (fondée en 1987). Ce système propose des activités (grammaire, conjugaison et orthographe), de plus, il propose des enregistrements sonores à écouter dans le but d'aider l'apprenant à prononcer correctement les mots. Comme les activités sont statiques, Telle Me More propose un feedback à la fin de chaque activité (Berthet, 2001).

Dans cette optique, on trouve aussi le projet Rosetta Stone développé par la société Fairfield Langue (fondée en 1992). L'application propose à l'apprenant une variété d'activité selon son niveau. De même, les activités proposées par cette application sont statiques.

3 Le TAL et ses applications destinées à l'ALAO

Dans les dernières décennies, il y avait beaucoup des recherches qui s'intéressent à l'intégration des outils et ressources TAL dans des systèmes d'ALAO (Antoniadis, 2010), (Mars et al, 2014). En général, les outils TAL tels que l'analyseur morphologique, le système de traduction, le système de la reconnaissance de la parole sont les plus cités. Dans la suite, on en présentera quelques-uns.

3.1 Définition

Le traitement automatique de langue (TAL) a commencé dans les années 1950 comme l'intersection de l'intelligence artificielle et de la linguistique. Le TAL est une méthode de traduction entre les langages humains et l'informatique. Aussi, le TAL peut être définit par la manière dont l'ordinateur peut lire une ligne de texte en l'interprétant d'une manière linguistique et non-binaire. En d'autres termes, le TAL automatise le processus de traduction entre les ordinateurs et les humains. Traditionnellement, les calculs statistiques et les modèles des langages (Chomsky, 1957) ont été employés par le TAL pour l'interprétation de phrases. Les progrès récents dans ce domaine utilisent des logiciels de reconnaissance vocale, la traduction du langage humain, la recherche d'informations et l'intelligence artificielle (Manning et al, 2008). Le TAL est également en cours de développement pour arriver à une compréhension acceptable du langage naturel et pour effectuer une traduction entre un langage humain et un autre.

3.2 Utilisation du TAL dans l'ALAO

L'intégration du TAL dans des logiciels d'ALAO n'est pas une idée nouvelle, nous trouvons quelques systèmes comme ALEXIA (Selva et al, 2000), ELEONORE (Renié, 1995) ou EXILIS (Bruno, 2002) qui utilisent des outils issus du TAL. Généralement, ces outils sont des dictionnaires, des analyseurs morphologiques et syntaxiques, des systèmes du traitement de la parole (reconnaissance et synthèse de la parole), des systèmes de reconnaissance des formes et de patterns (comme exemple, le système de reconnaissance de l'écriture manuscrite) et des outils de traitement de corpus.

3.2.1 L'analyse morphologique et le TAL

La tâche de l'analyse morphologique consiste à déterminer les bonnes étiquettes morphologiques (adjectif, adverbe, verbe, etc.) pour une séquence de mots. Cette tâche est difficile, car les mots sont souvent morphologiquement ambigus. L'analyse morphologique est utile pour un grand nombre d'applications. Elle présente la première étape d'analyse dans de nombreux analyseurs syntaxiques. Elle est nécessaire pour la lemmatisation correcte des mots et elle est utilisée dans l'extraction de l'information, la synthèse vocale, la recherche lexicographique, l'extraction de termes, et aussi bien pour d'autres applications. Généralement, les analyseurs morphologiques utilisent trois modèles : modèles statistiques, modèles à base des règles, modèles basés sur la classification.

3.2.1.1 Approche à base des règles

Cette méthode est basée sur les règles grammaticales ou morphologiques de la langue. Dans ce cas, l'analyseur doit appliquer les règles, soit pour affecter une étiquette à un mot, soit pour désigner les différentes étiquettes possibles lors d'une transition entre deux mots. À titre d'exemple, nous citons l'analyseur morphologique d'Éric Brill (Brill, 1995). Cet analyseur a construit sa base des règles à partir d'un corpus étiqueté manuellement lors de la phase d'apprentissage. Ces règles sont divisées en deux classes :

- Classe des règles lexicales : en se basant sur la propriété lexicale d'un mot, l'analyseur est capable d'affecter une étiquette à chaque mot.
- Classe des règles contextuelles : le rôle de ces règles est d'affiner l'étiquetage après la phase d'étiquetage avec l'application des règles lexicales. Dans cette phase, l'analyseur se base sur le contexte de la phrase, c'est-à-dire, il revient sur les étiquettes

affectées aux mots qui précèdent le mot courant pour lui affecter la nouvelle étiquette (Thibeault, 2004).

3.2.1.2 Analyseur morphologique statistique

Cette approche est la plus utilisée dans le développement des analyseurs morphologiques. Cette approche est basée sur des calculs de probabilités. L'application de cette approche est simple, on attribue l'étiquette la plus utilisée, pour le mot en cours d'analyse, dans le corpus d'apprentissage étiqueté (Brown et al, 1990). Par contre, le point faible de cette approche est qu'il peut affecter des étiquettes qui ne sont pas acceptables grammaticalement. Comme solution a ses faiblesses, les scientifiques proposent l'approche de « n-gramme », pour laquelle on calcule la probabilité d'une séquence donnée d'étiquettes. Autrement dit, l'analyseur détermine l'étiquette la plus probable d'un mot donné en se basant sur la probabilité qu'il apparaisse avec les n étiquettes qui le précèdent. À titre d'exemple, nous citons l'approche basée sur le Modèle de Markov Caché (MMC). Le MMC combine l'approche de « n-gramme » et l'approche probabilistique simple, c'est-à-dire il utilise à la fois la probabilité d'apparition d'une séquence d'étiquette et l'étiquette la plus utilisée dans un corpus d'apprentissage étiqueté. Par conséquent, le choix de l'étiquette pour un mot donné selon le MMC dépend des n étiquettes précédentes. Voici la formule utilisée pour le calcul de la probabilité qu'une étiquette « a » est suivie par l'étiquette « b » : la probabilité d'avoir l'étiquette « a » suivie par l'étiquette « b » est égale à la fréquence d'apparition de « a » avant « b » dans le corpus d'apprentissage, divisé par la fréquence d'apparition de « a » dans le corpus.

$$\mathbf{prob(a|b) = nbr(a \ \& \ b) \ / \ nbr(a).}$$

3.2.1.3 Approches basées sur la classification

Cette approche est basée sur deux types d'apprentissage automatique, le premier est l'apprentissage à base de mémoire et le deuxième est l'apprentissage avec des machines à support de vecteurs.

- Apprentissage à base de mémoire : cette méthode produit, à partir des corpus étiquetés manuellement, des classes et des lexiques. En premier lieu, l'analyseur doit subir une phase d'apprentissage et il doit mémoriser des exemples (d'où vient l'appellation « à base de mémoire »). Finalement, l'analyseur doit effectuer un calcul de similarité entre

les données d'entrée et les données de sortie de la phase d'apprentissage. (Walter et al, 1996)

- Apprentissage avec des machines à support de vecteurs (SVM) : c'est une méthode basée sur une classification binaire, inspirée de la théorie d'apprentissage de Vladimir Vapnik (Vapnik, 1995). Cette méthode est adaptée aux données de grande masse.

3.2.2 Dictionnaires

Les dictionnaires sont très utilisés dans le domaine du TAL (pour le développement des analyseurs morphologiques, les systèmes de reconnaissance de la parole, etc.), ainsi que dans le domaine de l'ALAO.

Pour une grande variété d'applications du TAL, un lexique avec des informations sur la morphologie des mots et ce qu'ils signifient est une composante nécessaire. (Pustejovsky, 1995) montre que les éléments lexicaux peuvent offrir des améliorations majeures dans les opérations de composition associées aux systèmes du TAL.

3.2.3 Reconnaissance automatique de l'écriture manuscrite

La reconnaissance de l'écriture manuscrite est la capacité d'un ordinateur à recevoir et interpréter une entrée manuscrite intelligible à partir des surfaces ou écrans tactiles, ou des stylos numériques et d'autres dispositifs. L'image du texte écrit peut être détectée "hors ligne" à partir d'un morceau de papier par lecture optique (reconnaissance optique de caractères) ou à partir des mouvements de la pointe du stylet qui peuvent être détectés par la surface d'un écran tactile ou un stylet numérique.

La reconnaissance de l'écriture manuscrite est divisée en deux catégories : en ligne et hors ligne.

3.2.3.1 Reconnaissance en ligne

La reconnaissance de l'écriture en ligne implique la conversion automatique de texte de l'état manuscrit vers l'état numérique. Ce type de données est connu comme encre numérique et peut être considérée comme une représentation numérique de l'écriture manuscrite. Le système reçoit les données manuscrites, puis il doit traiter les données, et, enfin, il reconnaît l'écriture en temps réel, c'est-à-dire que le système convertit l'encre en codes de lettres qui sont utilisables dans les applications informatiques et de traitement de texte.

3.2.3.2 Reconnaissance hors ligne

La reconnaissance hors ligne : après que les données soient recueillies sous forme d'une image de l'écriture, le système analyse ces données et les convertit en texte numérique. Ainsi, la vitesse du dispositif de reconnaissance ne dépend pas de la vitesse de l'écriture de l'utilisateur, mais de la spécification du système, en mots ou caractères par seconde.

3.2.3.3 Premiers systèmes de la reconnaissance de l'écriture

- Système Apple-Newton : en 1994, Apple-Newton a été introduit dans le système d'exploitation Newton OS version 2.0. Ce système est destiné à la reconnaissance de l'écriture manuscrite en ligne, il a une précision faible; la société Apple a abandonné le système à cause de ses mauvais résultats (Yaeger et al, 1996).
- ThinkWrite : le système ThinkWrite d'IBM a été introduit dans le système d'exploitation de Microsoft « Windows 3.1 »; il était destiné aux imprimantes (Macneil, 1995) et au système d'exploitation d'Apple, OS2. Selon les documents en ligne, ThinkWrite [5] utilise un système «hybride» en ligne qui exploite l'encre et les informations temporelles pour la reconnaissance de l'écriture. Il utilise un algorithme basé sur la reconnaissance des caractères (Macneil, 1995).
- Graffiti : c'est un produit de la société Palm Computing. Il est disponible pour toutes les plates-formes (Windows, Linux, Mac); il adopte une approche différente de la reconnaissance de l'écriture manuscrite du système Apple-Newton. Graffiti utilise un système de caractères dans lequel toutes les lettres de l'alphabet et tous les symboles sont représentés par un seul segment.

4 L'apport du TAL

Avec les progrès de l'informatique en termes de technologie (intelligence artificielle, apprentissage automatique, recherche d'information, etc.) et en termes de puissance de calcul des machines, nous constatons une amélioration au niveau de performance et de richesse des systèmes d'ALAO et des outils TAL.

4.1 Avantages

Le développement des applications et des méthodes basées sur le traitement automatique du langage naturel pour l'ALAO est devenu un domaine important de la recherche.

L'ALAO s'intéresse, à l'élaboration d'environnements informatiques pour l'apprentissage des langues. Actuellement, quelques systèmes utilisent des outils issus des TAL. Ces outils ont permis la création d'environnements, communément appelés plates-formes. Ces plates-formes offrent différentes fonctionnalités telle que la gestion de dictionnaires et la gestion des activités et ressources pédagogiques (Power, 2002).

Le couple didactique des langues/informatique subit des développements remarquables mais n'en sont pas moins limités par l'approche réductionniste de la langue par l'informatique (Antoniadis et Ponton, 2002). Cette approche réductionniste s'explique par la manière dont l'informatique traite une langue ; l'informatique ne traitant que de la forme, un mot, un syntagme, une phrase... sont vus comme de simples suites de caractères sur lesquels il est possible d'effectuer des opérations comme des comparaisons, des comptages et le calcul des distributions, mais pas d'atteindre les niveaux lexicaux, morphosyntaxiques ou sémantiques des mots ou des phrases. Cette impossibilité à prendre en compte les propriétés intrinsèques de la langue restreint fortement les possibilités des systèmes d'ALAO et, plus globalement, l'utilisation des technologies informatiques en apprentissage des langues.

L'intégration du TAL dans l'ALAO reste encore moins abordée par rapport aux autres domaines du TAL. (Kraif et al, 2004) donnent trois arguments pour expliquer ces phénomènes : (i) ces technologies manquent encore de fiabilité ; (ii) le coût est encore cher et les ressources difficiles à développer ; (iii) les utilisateurs sont peu au courant des possibilités offertes par ces techniques.

4.2 Enjeux

Le TAL apporte des solutions aux limitations des systèmes d'ALAO. Parmi ces avantages, on peut citer :

- Le TAL permet aux apprenants de travailler indépendamment sur la plate-forme, car la génération des activités devient autonome et ne nécessite pas la présence de l'enseignant de la langue.
- L'utilisation des outils et des approches TAL enrichit les systèmes d'ALAO par une grande variété des activités, tels que les exercices d'apprentissage de l'écriture, exercices d'expression orale, dictée, exercices sur les prononciations…
- Les systèmes d'ALAO deviennent capables de détecter, d'expliquer et de corriger automatiquement les erreurs.

- Une véritable personnalisation et adaptation des activités à l'apprenant selon le niveau et selon la progression de niveau de l'apprenant.

Généralement, avant de proposer des approches ou méthode TAL, il faut aborder ces points :

- Seule le TAL permet d'interpréter la langue, non comme une succession de signes et caractères sans signification, mais comme des éléments d'un système à deux niveaux (forme et sens). Dans le cadre de l'apprentissage d'une langue, le fonctionnement de chaque niveau du système doit être considéré, rendu explicite, manipulé et mis en pratique d'une manière efficace et bien ciblée. De plus, le lien entre les deux niveaux est considéré comme une source de difficultés (à cause de la polysémie) dans l'apprentissage de la langue. Par conséquent, il faut le traiter d'une manière appropriée pour créer donc des outils qui permettront aux professeurs de langues de manipuler la langue, il est nécessaire qu'ils travaillent avec la façon dont la langue est réellement structurée, sans les limitations imposées par les techniques informatiques de base.

Il faut toujours connaître l'apprenant pour faciliter la définition de ses besoins d'apprentissage. Ainsi, le concepteur de l'outil TAL peut optimiser l'apprentissage personnalisé. Il est néanmoins incontestable que nous attendons toujours des systèmes capables de fournir ce type d'apprentissage. Le défi de l'ALAO n'est pas l'évaluation des connaissances des apprenants qui constitue la majeure difficulté, mais comment utiliser cette évaluation et, surtout, comment créer automatiquement des activités qui s'adaptent au niveau de l'apprenant dans la langue à apprendre. Le TAL peut apporter des solutions à la fois en termes d'outils d'évaluation et de création d'activité automatique (Desmet, 2006).

Il faut être capable de détecter, d'expliquer et de corriger automatiquement les erreurs des apprenants si l'objectif est de permettre à l'apprenant de travailler de façon autonome et indépendante. Par conséquent, il faut répondre à la question de savoir comment générer automatiquement les explications pour les apprenants, ainsi que la question sur la façon dont on doit fournir une correction automatique aux productions des apprenants. Très peu de travaux ont été faits sur ce point, comme on peut le voir à partir de la revue « CALICO » qui est entièrement consacré à ce sujet (Heift et Schulze, 2003). Actuellement, les types des réponses proposées par les systèmes d'ALAO sont sous la forme de :

- Vrai ou faux.
- Une liste préétablie des réponses correctes.
- Numéro de la réponse correcte.

Par contre, un feedback adéquat doit tenir compte des caractéristiques linguistiques de la réponse de chaque apprenant. Ces caractéristiques permettent alors l'implantation de l'évaluation dans le cadre de l'activité d'enseignement, conformément à l'élève et au profil de l'apprenant. En outre, au lieu de donner une évaluation sous la forme d'une réponse binaire (correcte ou incorrecte), il peut être plus intéressant de donner des éléments de diagnostic, afin d'aider l'apprenant à identifier et à corriger l'erreur lui-même.

Les outils et les systèmes créés ne devraient pas être compliqués et nécessiter des compétences techniques spécifiques. Comme tout domaine de connaissances, le TAL, la linguistique et la didactique, utilisent leurs propres approches et leurs propres concepts qui nécessitent l'utilisation d'outils spécifiques, généralement fondés sur ces concepts. Pour le non spécialiste, l'utilisation de ces outils peut impliquer de longues périodes de formation et une compréhension d'un certain nombre de concepts liés à l'utilisation de chaque outil. Dans le cadre de l'apprentissage des langues, où les enseignants sont a priori des spécialistes en didactique, leurs compétences en informatique et en TAL sont très limitées. Cette situation peut être source de problèmes pour les enseignants, lorsqu'ils veulent se familiariser avec des outils du TAL. Tous les systèmes d'ALAO sont destinés a priori pour les enseignants de langues et pour les apprenants. Par conséquent, ces systèmes devraient répondre à deux impératifs : l'utilisation d'un tel système ne devrait exiger qu'un minimum de compétences non didactiques et il devrait être capable de gérer les concepts didactiques.

5 Utilisation du TAL dans l'ALAO arabe

Les travaux dans le domaine d'ALAO arabe basé sur le TAL pour la langue arabe restent négligeables par rapports aux travaux menés pour les langues latines (tels que le français et l'anglais). De plus, nous ne trouvons que peu d'outil et ressource TAL arabe, cela est dû au fait que l'arabe est une langue complexe et difficile à traiter automatiquement (Khoufi et al, 2015), (aljlayl et al, 2002), (Larkey et al, 2002).

Pour la langue arabe, quelques réalisations ont vu le jour. Indépendamment des techniques utilisées et la valeur ajoutée, ces outils ont tous tenté de faire évoluer l'apprentissage de la langue arabe assisté par ordinateur.

5.1 Outils d'ALAO classiques pour l'arabe utilisant de ressource TAL

Malgré le développement du TAL arabe, ces dernières années, le nombre des outils qui sont mis à la disposition de l'ALAO pour proposer aux enseignants et aux apprenants des environnements interactifs pour l'apprentissage de la langue arabe reste très modeste (Zrigui et al. 2012).

L'un des premiers outils destinés pour l'ALAO de l'arabe est celui réalisé pour la détection de racines des mots (Hegazi et al. 1989). Réalisé en PROLOG22, ce système a été ensuite adapté pour une utilisation dans l'apprentissage pour détecter et corriger les erreurs des apprenants. ArabVISL (Apprentissage visuel interactif de la syntaxe arabe) est un outil interactif en ligne pour l'apprentissage de la grammaire arabe. Le développement de cet outil a été fait à l'université du Danemark par Nielson (Nielsen et al. 2003). ArabVISL permet aux apprenants de l'arabe d'analyser des phrases écrites en arabe. Ces analyses utilisent des scripts TAL.

Shaalan a décrit son système "Arabic CALL" basé sur des techniques TAL (Shaalan 2005), (Shaalan et al. 2006). Ce système permet aux apprenants de produire des phrases dans différents contextes. Selon Shaalan toujours, le système permet aux apprenants de reconnaître les erreurs commises. Les ressources TAL utilisées sont : un outil de vérification de grammaire appelé GramCheck (Shaalan 2005), un analyseur syntaxique et un outil d'analyse des erreurs pour vérifier les réponses des apprenants. En testant les deux premiers outils, nous constatons qu'ils ne sont pas assez performants et ne peuvent pas donner de résultats satisfaisants. Pour l'outil de vérification des erreurs, l'analyse des réponses libres d'apprenants demande des outils plus avancés allant jusqu'à l'analyse sémantique afin de déterminer s'il y a des relations sémantiques voisines (hyperonymie, homonymie, hyponymie, synonymie, polysémie, métonymie, antonymie) entre la réponse donnée et la réponse correcte (Antoniadis et al. 2005a), (Antoniadis et al. 2005c).

Un dernier outil d'apprentissage, basé sur des outils TAL, appelé SALA a été développé (Maraoui 2009). Cet outil permet aux apprenants de l'arabe de faire des exercices de dérivations selon des schèmes ou des activités de conjugaison. Il utilise un conjugueur et un outil de dérivation nominale et verbale. Le feedback reste tout de même classique de style Vrai/Faux.

5.2 Limitation des systèmes d'ALAO arabe actuels

Actuellement, la majorité des systèmes d'apprentissage de la langue arabe n'utilisent pas les technologies et les outils TAL. Le TAL est sûrement le champ manquant pour les systèmes d'ALAO arabe pour une réelle évolution qualitative dans la conception de tels systèmes. On les trouve généralement sur des CD ou des sites Web qui contiennent des exercices d'orthographe ou de grammaire. Cependant, la plupart de ces systèmes ont leurs limites communes au niveau feedback, la qualité et le nombre d'activités, l'interaction apprenant-machine, etc. Ces limites sont :

- Ils ressemblent souvent à des cahiers d'exercices traditionnels desquels ils étaient inspirés.

- D'un point de vue pédagogique, la définition des réponses acceptables à des exercices est fortement contrainte, comme, par exemple, dans les questions d'analyse linguistique.

- Le feedback lors d'une erreur n'aborde pas la source de l'erreur, ne s'adapte pas au niveau de l'apprenant. Par exemple, le système affiche la réponse correcte sans aucune explication de l'erreur à l'apprenant. Cela rend le feedback du système trop générique.

- Rigidité des systèmes, c'est à dire le nombre d'activités est très limité et défini à l'avance, les données utilisées sont prédéfinies, il y a inadaptation de ces activités aux compétences des apprenants …

Afin de permettre aux apprenants de travailler indépendamment, chaque activité doit être capable de détecter, expliquer et corriger automatiquement les erreurs. Les systèmes actuels ne permettent pas une véritable personnalisation et adaptation à l'apprenant parce que les corrections sont prédéfinies.

6 Relations entre le TAL et l'ALAO

Ces dernières années, il y a eu un intérêt croissant pour la communauté du TAL dans le domaine de l'ALAO. En fait, de nombreux nouveaux outils, applications et équipements TAL, sont développés afin de l'intégrés dans les systèmes d'ALAO. Et il est vrai que, même s'il y a encore quelques limitations et des difficultés lors de l'application des outils TAL dans l'ALAO, pour cela le but des nos travaux et de trouver des solutions a ces limitations. Nous

trouvons quelques recherches qui ont commencé a aborder la problématique de l'intégration du TAL dans l'ALAO, dont on cite quelques-uns : (Antoniadis, 2013), (Wahl, 2012), (Nerbonne, 2003), (Heift, 2003).

Selon Nerbonne le rôle de l'ALAO est de fournir des matériaux compréhensibles pour les enseignants et les apprenants. En effet, les outils de TAL peuvent être des ressources d'aide supplémentaires au sein des logiciels d'ALAO afin de permettre aux apprenants de langue et les enseignants d'obtenir facilement des informations sur la langue cible ainsi que pour obtenir certains matériaux. En fait, ces outils peuvent illustrer des structures linguistiques, rendre la langue compréhensible, fournir du matériel et à corriger les erreurs d'entraînements diversifiés, etc.

7 Limite de l'utilisation du TAL dans l'ALAO

Le développement des outils TAL pour l'apprentissage de langue est une tâche difficile et coûteuse en ressources linguistiques. Actuellement, la majorité des approches et des méthodes utilisées par le TAL nécessite la construction des ressources linguistiques étiquetés et annotés (corpus, lexiques, dictionnaires…). Cette tâche est très coûteuse en terme du temps et des ressources, car l'élaboration d'un dictionnaire ou corpus étiqueté fiable nécessite un temps important de construction et un budget conséquent pour payer l'intervention de linguistes dans la phase d'étiquetage (Kraif et al, 2004).

8 Conclusion

Dans ce chapitre, nous avons présenté quelques systèmes d'ALAO existants en étudiant leurs points forts ainsi que leurs limitations. Ces systèmes ont des limitations communes ; actuellement, les types d'activités générés restent limités aux activités traditionnelles, le feedback reste basique du type vrai ou faux, etc. Ensuite, nous avons montré le rôle et l'apport pertinent du TAL dans le domaine d'apprentissage de langue assisté par ordinateur ainsi que les solutions apportées par le TAL. Pour cela, notre but dans cette thèse est de trouver des solutions aux faiblesses des systèmes d'ALAO en utilisant des ressources et des techniques issues du TAL. Par contre, vu la qualité des ressources et outils TAL, nous devons nous focaliser sur l'amélioration ainsi que le développement des certaines ressources et outils TAL avant de commencer la conception et la réalisation de la plate-forme d'ALAO.

Chapitre 3 : Langue arabe et les difficultés de sa traitement

1 Introduction

L'arabe est une langue vivante qui appartient au groupe des langues sémitiques. Le groupe des langues sémitiques comprend d'autres langues vivantes telles que : l'hébreu moderne, l'amharique, l'araméen, le tigrinya et le maltais.

L'arabe standard est la forme écrite commune à tous les pays arabophones ; c'est la langue de la littérature, de la presse et des médias. Elle est une langue fortement flexionnelle qui rend le traitement automatique des textes arabes difficile.

Dans ce chapitre, nous présenterons certains aspects syntaxiques et morphologiques de la langue arabe, ainsi que la complexité et la problématique du traitement automatique de cette langue.

2 Langue Arabe

2.1 Caractéristiques

La langue arabe se compose de vingt-huit lettres de base (Karin, 2005); elle s'écrit et se lit de droite à gauche. Dans l'écriture arabe, il n'y a pas les notions de majuscule et minuscule. Les lettres arabes changent de forme de présentation selon leur position au sein des mots soit isolée, initiale, médiane ou finale (tableau 2). Dans la langue arabe, toutes les lettres se lient entre elles sauf « و », « د », « ذ », « ر », « ز » et « ا » qui ne s'attachent pas à gauche. Donc, les caractères arabes n'ayant pas de forme initiale et médiane s'attachent au caractère qui précède, mais pas à celui qui suit.

à la fin d'une lettre non joignable	à la fin	au milieu	au début
ع	ـع	ـعـ	عـ

Tableau 2 : Variation de la lettre ع « Ayn ».

Un mot arabe se compose de consonnes et de voyelles. Les voyelles se positionnent au-dessus ou au-dessous des lettres (ً,ٌ,ٍ,ّ). Elles sont indispensables à la compréhension et à la lecture correcte d'un texte. Le monde arabe n'utilise les voyelles que pour des textes religieux ou didactiques (en particulier pour les écoles primaires). Généralement, les journaux et les livres ne comportent pas de voyelles.

Dans cette thèse, nous traitons « l'arabe moderne » ou « l'arabe standard ». Il s'agit de la langue écrite dans les journaux et les livres.

IPA	Latin	Name		Final	Medial	Initial	Isolated	IPA	Latin	Name		Final	Medial	Initial	Isolated
[ʈ]	ṭ	ṭāʾ	طاء	ـط	ـطـ	طـ	ط	[ʔ]	ʾ(a)	ʾalif	ألف	ـا	ـ	ـ	ا
[ʐ]	ẓ	ẓāʾ	ظاء	ـظ	ـظـ	ظـ	ظ	[b]	b	bāʾ	باء	ـب	ـبـ	بـ	ب
[ʕ]	ʿ	ʿayn	عين	ـع	ـعـ	عـ	ع	[t]	t	tāʾ	تاء	ـت	ـتـ	تـ	ت
[ɣ]	ġ	ġayn	غين	ـغ	ـغـ	غـ	غ	[θ]	ṯ	ṯāʾ	ثاء	ـث	ـثـ	ثـ	ث
[f]	f	fāʾ	فاء	ـف	ـفـ	فـ	ف	[ʤ]	ǧ	ǧīm	جيم	ـج	ـجـ	جـ	ج
[q]	q	qāf	قاف	ـق	ـقـ	قـ	ق	[ħ]	ḥ	ḥāʾ	حاء	ـح	ـحـ	حـ	ح
[k]	k	kāf	كاف	ـك	ـكـ	كـ	ك	[χ]	ḫ	ḫāʾ	خاء	ـخ	ـخـ	خـ	خ
[l]	l	lām	لام	ـل	ـلـ	لـ	ل	[d]	d	dāl	دال	ـد	ـ	ـ	د
[m]	m	mīm	ميم	ـم	ـمـ	مـ	م	[ð]	ḏ	ḏāl	ذال	ـذ	ـ	ـ	ذ
[n]	n	nūn	نون	ـن	ـنـ	نـ	ن	[r]	r	rāʾ	راء	ـر	ـ	ـ	ر
[h]	h	hāʾ	هاء	ـه	ـهـ	هـ	ه	[z]	z	zāy	زاي	ـز	ـ	ـ	ز
[w]	w	wāw	واو	ـو	ـ	ـ	و	[s]	s	sīn	سين	ـس	ـسـ	سـ	س
[j]	y	yāʾ	ياء	ـي	ـيـ	يـ	ي	[ʃ]	š	šīn	شين	ـش	ـشـ	شـ	ش
		hamza	همزة	ـء	ـ	ـ	ـ	[s]	ṣ	ṣād	صاد	ـص	ـصـ	صـ	ص
								[ɖ]	ḍ	ḍād	ضاد	ـض	ـضـ	ضـ	ض

Figure 4 : Alphabet arabe et les différentes formes d'écriture[8]

[8] http://www.lingvozone.com/Arabic

- **Encodage** : L'encodage principal des caractères arabes est l'ISO-8859-6 (sous Windows il est appelé : Windows-1256). De plus, l'arabe est supporté aussi par l'encodage Unicode.
- **Chiffres** : Les chiffres utilisés sont les chiffres latins au Maghreb et les chiffres « arabo-indiens » au Moyen-Orient. Les chiffres s'écrivent de gauche à droite.

0	1	2	3	4	5	6	7	8	9
٠	١	٢	٣	٤	٥	٦	٧	٨	٩

Tableau 3 : Chiffres arabes avec leur correspondance en chiffres latins.

- Sens d'écriture : La lecture et l'écriture d'un mot arabe se font de droite vers la gauche.
- Marques diacritiques : En arabe, l'écriture des marques diacritiques se fait en même temps avec l'écriture des lettres ou à la fin du mot.
- Schèmes : Le schème, appelé aussi pattern, est un mot composé de trois lettres radicales (f, ف), (E, ع) et (l, ل). Le schème occupe une place importante dans le processus de dérivation des mots arabe (Karin, 2005).
- Ponctuation : Les signes de ponctuation arabe sont identiques à ceux utilisés dans les langues européennes, mais sont renversés, exemple :
  - la virgule renversée " ،"
  - le point-virgule renversé " ؛ "
  - le point d'interrogation inversé " ؟ ".

2.2 Variétés

Les linguistes arabes ont structuré la langue arabe en deux variétés principales, la première variété dite « l'arabe classique » ou « l'arabe littéraire » et la deuxième dite « l'arabe dialectal ». D'autres linguistes ont ajouté une variété intermédiaire écrite et parlée, et désignée par le nom « arabe standard contemporain » (Parkvall, 2010).

- L'arabe classique : C'est la forme linguistique ancienne de l'arabe dont les règles de la grammaire étaient fixées entre le 8e et le 10e siècle. Il est appris dans les établissements d'enseignement à travers la littérature arabe et les cours de théologie.

- L’arabe standard contemporain (moderne) : C’est une variante moins formelle que l’arabe classique. L’arabe standard est utilisé dans la vie officielle, universitaire et administrative. De plus, c’est par le biais de l’arabe standard, que deux locuteurs arabophones d'origines dialectales différentes sont susceptibles de se comprendre.
- L’arabe médian : C’est une forme intermédiaire entre l’arabe moderne et dialectal. On le désigne par le terme « arabe parlé formel » (Tarrier, 1991) et aux pays Maghrébins sous le nom de « arabe médian ». Cette variété est considérée comme une variante simplifiée de l'arabe littéral moderne et une forme élevée de l'arabe dialectal.
- L’arabe dialectal :Iil est souvent utilisé dans l'expression de la vie quotidienne locale. Elle est considérée comme la langue vernaculaire de l'ensemble des arabophones. Par définition, les dialectes arabes sont les langues maternelles des populations des pays arabes, et ces formes linguistiques sont parfois très différentes d'une région à l'autre. On peut distinguer l'arabe dialectal par rapport la langue standard, enseignée à l'école et théoriquement commune à l'ensemble des pays arabophones, par de nombreux points : syntaxe, lexique, phonologie, morphologie, phonétique (Parkvall, 2010).

2.3 Catégories d’un mot

Avant de commencer le traitement automatique de la langue arabe, il faut tenir compte de la classification des unités lexicales de la langue arabe (nom, verbe, pronom…) sur laquelle doit se baser le traitement.

2.3.1 Le verbe

La majorité de mots arabes, dérivent d’un verbe de trois consonnes qui représente lui-même une racine d’un groupe de mots. Par conséquent, le mot en arabe se déduit à partir de la racine en rajoutant des suffixes, des préfixes ou les deux en même temps.

En arabe, la conjugaison des verbes dépend de facteurs suivants :

- Le temps : accompli (passé) ou inaccompli (présent).
- Le nombre du sujet : singulier, duel, pluriel.
- Le genre : masculin ou féminin.
- La personne (première, deuxième, etc).

- La voix : active ou passive.

Voici un exemple qui explique l'effet de ces différents facteurs :

Dans cet exemple, on prend la combinaison de ces trois consonnes (ك + ت + ب) « k + t + b » donne le verbe كتب « écrire ».

Les trois lettres K, T, B, on les trouvera dans tous les mots qui dérivent de cette racine, en rajoutant des préfixes, des suffixes ou les deux à la racine.

La conjugaison des verbes arabes dépend de plusieurs facteurs, parmi ces facteurs le temps. Par contre, la notion du temps est différente de celle du français, il se divise en deux temps : l'accompli et l'inaccompli. L'inaccompli recouvre le présent et le futur et l'accompli représente le passé.

- L'accompli : Il désigne le passé et le verbe conjugué se distingue par des suffixes. Dans notre cas, la conjugaison du verbe كتب « écrire » avec le pluriel féminin donne كتبن KaTaBna, « elles ont écrit » et avec le pluriel masculin donne كتبوا KaTaBuu, « ils ont écrit ».
- L'inaccompli présent : Dans ce cas, l'action est en cours d'accomplissement. Les verbes conjugués se distinguent par les préfixes. Comme exemple, la conjugaison du verbe كتب « écrire » au masculin singulier donne يكتب yaKTuBu, « il écrit » et avec le féminin singulier donne تكتب taKTuBu, « elle écrit ».
- L'inaccompli futur : Dans ce cas, l'action se déroulera au futur et elle est marquée par l'antéposition de س sa ou سوف sawfa au verbe. En ajoutant l'antéposition س sa à notre exemple, on obtient سيكتب sayaKTuBu , « il écrira » et en ajoutant l'antéposition سوف sawfa, on obtient سوف يكتب sawfa yaKTubu « il va écrire ».

2.3.2 Le nom

En arabe, le nom se divise en deux catégories, ceux qui sont dérivés d'une racine verbale et ceux qui ne le sont pas, tel est le cas des noms propres, des noms étrangers...

La déclinaison des noms obéit à plusieurs règles :

- Le féminin singulier : dans la plupart des cas, on ajoute la lettre ة t pour obtenir le nom au féminin singulier (exemple : صغير « petit » devient صغيرة « petite »).
- Le féminin pluriel : généralement, on rajoute les deux lettres ات (exemple : عامل « *ouvrier* » devient عاملات « *ouvrières* »).

- Le masculin pluriel : généralement, on rajoute les deux lettres ون ou les deux lettres ين qui dépendent de la position du mot dans la phrase (sujet ou complément d'objet).

Exemple : معلّم « *enseignant* » devient معلّمين ou معلّمون « *enseignants* ».

- Le pluriel irrégulier : c'est le cas le plus complexe en arabe, pour transformer un mot au pluriel, on doit insérer des lettres au début, au milieu ou à la fin du mot, (exemple : قفل « serrure » devient أقفال « serrures »).

Le problème du pluriel irrégulier dans l'arabe pose un défi à la morphologie, non seulement à cause de sa nature non concaténative, mais aussi parce que son analyse dépend de la structure de chaque mot (Kiraz, 1996) comme pour les verbes irréguliers.

2.3.3 Les particules

En arabe, les particules sont principalement les mots outils de la langue, comme les conjonctions de coordination et de subordination. Elles ont un rôle très important dans l'interprétation de la phrase (Kadri & Benyamina, 1992). Les particules sont classées selon leur fonction dans la phrase, on en distingue plusieurs types (introduction, explication, conséquence). Elles jouent un rôle important dans l'interprétation de la phrase. Elles sont utilisées dans la phrase pour situer des faits ou des objets par rapport au temps ou au lieu ; elles jouent, également, un rôle clé dans l'enchaînement et la cohérence d'un texte.

Par exemple, on trouve des particules temporelles ou spatiales qui désignent :

- Un temps منذ (pendant), قبل (avant), بعد (après).
- Un lieu : حيث (où).

On trouve aussi des particules qui peuvent exprimer des pronoms relatifs (la détermination avec une valeur référentielle), par exemple : اللّذين « ceux », اللّذي « ce », اللّتي « cette ».

De même façon que les noms et verbes arabes, certaines particules peuvent également avoir des préfixes et suffixes ce qui rajoute une complexité quant à leur identification.

2.3.4 Les préfixes

En arabe, les préfixes présentent les morphèmes correspondant aux lettres qui se situent en début de mot. Ils servent à indiquer la personne de la conjugaison des verbes. Le tableau suivant illustre les préfixes verbaux les plus fréquents en arabe.

Lettre	Rôle
أ	Indique la première personne au singulier (je)
نَ	Indique la première personne au pluriel (nous)
ت	Indique la deuxième personne féminine, masculine, singulière et duelle
ي	Indique la troisième personne masculine au singulier, duel, pluriel, masculin et féminin pluriel.

Tableau 4 : Liste des préfixes verbaux les plus fréquents en arabe

2.3.5 Les suffixes

En arabe, les suffixes présentent essentiellement les terminaisons des conjugaisons verbales, ainsi que les marques du pluriel et du féminin pour les noms. On ne trouve jamais une combinaison entre deux suffixes. Le tableau suivant illustre les suffixes les plus fréquents en arabe.

ا	ة	ين	ية	هم	ته	وه	ات
	ه	يه	تك	هن	تم	ان	وا
	ي	ية	نا	ها	كم	تي	ون

Tableau 5: Liste des suffixes les plus fréquents en arabe

2.3.6 Les proclitiques

A la différence des préfixes et suffixes, les proclitiques se combinent entre eux pour donner des informations pertinentes sur le mot arabe (coordination, détermination, traits sémantiques…). Selon (Abbes, 2004), dans le cas des verbes, les proclitiques dépendent exclusivement de l'aspect verbal. Ils comprennent donc tous les pronoms et par conséquent, ils sont compatibles avec tous les préfixes pris par l'aspect. Dans le cas des noms, le proclitique dépend du mode et du cas de déclinaison.

Les proclitiques peuvent exprimer la coordination en utilisant les coordonnants ف « f » et ن « w », l'interrogation en utilisant la lettre أ « hamza », le futur avec la marque س « s ».

Il n'est pas toujours facile de faire la différence entre les proclitiques et les lettres de la racine. Par exemple, la lettre س « s » dans le verbe سافر « il a voyagé » est une lettre de la racine, par contre dans le mot سيذهب « il va partir » c'est un proclitique (il marque le futur).

2.3.7 Les enclitiques

De la même manière que les proclitiques, les enclitiques se combinent entre eux pour donner une post-base composée. On les trouve toujours attachés à la fin du mot pour produire des pronoms qui s'attachent au verbe (complément d'objet direct), au nom et à la préposition (complément d'objet indirect).

Voici quelques exemples d'enclitiques :

- ي « Y » : أعطني « il m'a donné».
- هما « homA » أعطاهما « il les a donnés ».
- هم « hom » أعطاهم « ils l'ont donné ».

2.3.8 Les pré-bases

Les pré-bases en arabe sont obtenues par la concaténation des proclitiques et des préfixes. On peut générer d'une manière automatique les pré-bases en utilisant la liste des proclitiques et des préfixes. Dans la langue arabe, il y a 192 concaténations possibles. Le tableau suivant montre un exemple de pré-bases :

Pré-base	Préfixe	Proclitique
أست (asata)	ت (ta)	أس (asa)
ست (sata)	ت (ta)	س (sa)
أفت (afata)	ت (ta)	أف (afa)
فست (fasata)	ت (ta)	فس (fasa)

Tableau 6 : Exemple de pré-bases

2.3.9 Les post-bases

Les post-bases en arabe sont obtenues par la concaténation des enclitiques et des suffixes. Théoriquement, il y a 233 concaténations possibles qui peuvent se lier après la base.

Selon (Abbes, 2004) les compatibilités dépendent des pronoms décrits par chacune des particules. Le tableau suivant montre un exemple de post-bases :

Pré-base	Suffixe	Enclitique
وهم(wahom)	و(wa)	هم(hom)
وننني(wanany)	ني(nY)	ون(wan)
أنّكم(Annakom)	أنّ(Anna)	كم(kom)
أنّهم(Annahom)	أنّ(Anna)	هم(hom)

Tableau 7: Exemple de post-bases

3 Difficultés du traitement automatique de la langue arabe

3.1 La morphologie de l'arabe

La langue arabe a un lexique très riche. Ce dernier comprend trois catégories principales de mots : noms, verbes et particules, qui se subdivisent elles-mêmes en d'autres sous catégories : préposition, pronom, conjonction, interjection, article, et adverbe. Les noms et les verbes sont le plus souvent dérivés d'une racine trilitère (trois consonnes) (Baloul et al, 2002).

En arabe, le lexique est composé de racines. En général, une racine « جذر » représente une notion, par exemple رقص: danser. On peut dire qu'à partir d'une racine tous les autres concepts liés à une notion sont dérivés selon des schèmes, qui sont finis. Les deux premières lettres de la racine doivent être toujours différentes et dans le même ordre.

En général, à partir d'une racine, on peut dériver un ou plusieurs mots (nom ou verbe) en suivant ces étapes :

1- Choisir la racine.

2- Choisir la notion parmi celles qui sont permises (acteurs par exemple).

3- Choisir le schème correspondant.

4- Utiliser ce schème pour produire le mot souhaitable.

Voici un exemple qui montre ce principe de dérivation :

- Dérivation nominale :

  Racine : « ع م ل » + « م » → معمل « atelier ».

Racine : « ع م ل » + « ا » → عامل « ouvrier ».

- Dérivation verbale :

Racine : « ع م ل » + « س » + « ي » → سيعمل « il va travailler ».

Racine : « ع م ل » + « » → عمل « travaille »

Dans la langue arabe, on peut générer une famille de mots d'un même concept sémantique à partir d'une seule racine à l'aide de différents schèmes. On dit que l'arabe est une langue à racines réelles à partir desquelles on déduit le lexique arabe, en se basant sur des schèmes qui sont des adjonctions et des manipulations de la racine. Les tableaux suivants donnent quelques exemples des schèmes appliqués aux verbes كتب *écrire* et عمل *travailler*.

Schème	Mot	Traduction
فَعَلَ	كَتَبَ (KaTaBa)	a écrit
فَاعِلٌ	كَاتِبٌ (KâTiB)	écrivain
مَفْعلْ	مَكْتَبٌ (maKTaB)	bureau
فُعِلَ	كُتِبَ (KoTiBa)	a été écrit
مَفْعُولْ	مَكْتُوبْ(maKtwB)	Écrit
فُعُلْ	كُتُبْ(KoToB)	Livres

Tableau 8 : Exemple de schèmes pour le verbe كتب « KTB » *écrire.*

Schème	Mot	Traduction
فَعَلَ	عَمَلَ(EaMaLa)	a travaillé
فَاعِلٌ	عَامِلٌ(Eamil)	Ouvrier
مَفْعلْ	مَعْمَلٌ (maEML)	atelier
فُعِلَ	عُمِلَ(EuMiLa)	a été travaillé
مَفْعُولْ	مَعْمُولٌ(maEMwL)	applicable
فَعَلْ	عَمَلْ(EML)	travail

Tableau 9 : Exemple de schèmes pour le verbe عمل « EML » *travailler.*

Les lettres en majuscule (K, T, B, E, M, L) désignent les consonnes de base qui composent la racine (dans cet exemple, la lettre E correspond à la consonne arabe « ع »).

Les voyelles (â, a, i, u, o) désignent les voyelles et les consonnes en minuscule sont des consonnes de dérivation utilisées dans les schèmes.

La majorité des verbes arabes sont dérivés d'une racine composée de trois consonnes. La langue arabe utilise environ 150 schèmes ou patterns dont certains plus complexes, tel l'allongement d'une voyelle de la racine (voyelles longues) ou le redoublement d'une consonne ou l'adjonction d'un ou de plusieurs éléments.

3.2 Structure complexe des mots arabes et agglutination

Un mot graphique en arabe est composé de plusieurs objets complexes. La base de ce mot est appelée « mot minimal ». A partir de ce mot s'ajoutent d'autres constituants supplémentaires tels que les particules sous forme d'extensions. Ce mot graphique devient « mot maximal » lorsqu'il est composé de ces éléments : proclitique (s), préfixe(s), base(s), suffixe (s), enclitique(s) (Dichy 2000).

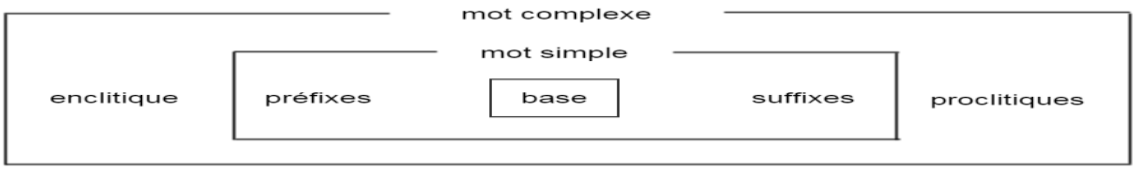

Figure 5 : Structure d'un mot arabe.

Les particules de l'agglutination ont leur nature cursive qui permet aux lettres arabes de se lier les unes aux autres au sein d'un même mot (Kammoun et al, 2010).

Exemple : يسمع « écoute » = ع+م+س+ي

De plus, un mot agglutiné peut former une phrase complète, tel est le cas du mot «أستتذكروننا, > stt*k~rwnnA». Ce mot signifie : « est-ce que vous allez vous souvenir de nous ?».

L'exemple ci-après montre la richesse morphologique de la langue arabe. Afin d'identifier les différentes formes soudées par ces phénomènes d'agglutination, et envisager un traitement automatique, il faut mettre en œuvre une phase spécifique de segmentation (tokenisation).

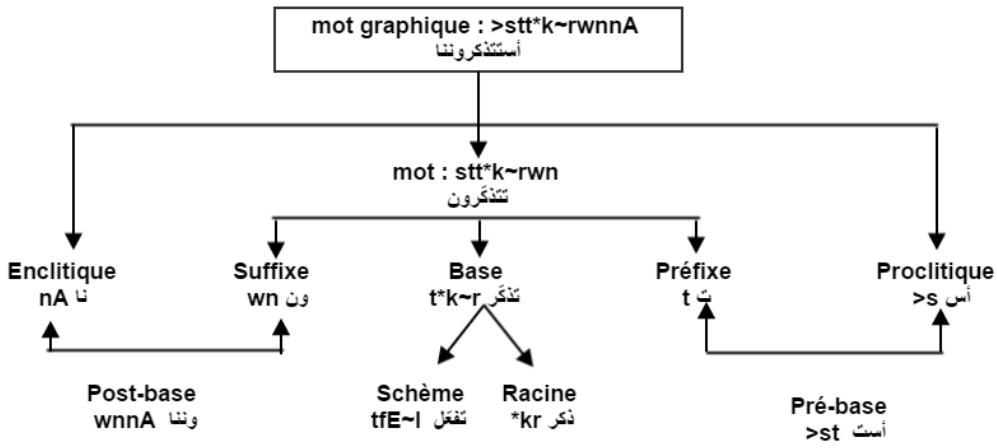

Figure 6 : Décomposition du mot arabe « أستتذكّروننا, >stt*k~rwnnA».

De plus, dans les journaux et les livres, certaines lettres comme alef « ا » remplacent le « آ », « أ » ou « إ » ; pareil pour les lettres « ى » et « ه » qui remplacent respectivement « ي » et « ة » (Xu et al, 2002).

À cause de ces caractéristiques morphologiques et syntaxiques, la langue arabe est considérée comme une langue complexe et difficile à traiter automatiquement (Atwell et al, 2004), (Debili et al, 2002).

4 Problématique de l'analyse morphologique

L'étude de la morphologie constitue l'élément principal du traitement automatique de la langue arabe en raison de ses interactions importantes avec les deux grands axes de la langue : l'orthographe et la syntaxe. La langue arabe possède une morphologie riche et très variée. Cette richesse est la conséquence de la richesse de la terminologie arabe.

L'analyse morphologique de l'arabe nécessite des applications informatiques qui analysent les mots arabes dans un texte donné et l'accordent avec la structure interne. Elle doit comporter une série de processus qui identifie toutes les analyses possibles d'un mot. Ces procédés sont à la fois basés sur la forme et sur la fonction du mot dans la phrase (Thabet 2004), (Hamada 2009), et (Habash 2010). Les processus des analyseurs morphologiques pour les textes arabes incluent la tokenisation, la vérification orthographique, la lemmatisation, la diacritisation, la prédiction des caractéristiques morphologiques des morphèmes d'un mot, l'étiquetage et l'analyse.

De nombreux analyseurs morphologiques pour le texte arabe ont été développés en utilisant plusieurs approches. Parmi ces approches, on cite : syllable-Based Morphology

(SBM), qui est basé sur l'analyse des syllabes du mot ; Root-Pattern Methodology, qui se base sur la racine et le motif du mot à analyser, Lexeme Based- Morphology, basé sur le lexème du mot, c'est-à-dire l'information cruciale qui doit être extraite à partir du mot (Soudi et al, 2001), (Soudi et al 2007).

La langue arabe présente des défis majeurs auxquels sont confrontés les analyseurs morphologiques de cette langue.

4.1 L'orthographe de la langue arabe

L'alphabet arabe se compose de : 25 consonnes et 6 voyelles classées en trois voyelles longues (ا, و, ي) (a, w, y) et trois voyelles courtes écrites comme des signes diacritiques (نَ, نُ ,نِ) (a, u, i). Les lettres arabes changent de forme en fonction de leur position dans le mot. De plus, dans la langue arabe les signes diacritiques sont utilisés en dessus et en dessous des lettres. Ces signes diacritiques sont le Sukun (نْ) pour marquer des lettres muettes (c.-à-d. absence de voyelle courte), la gémination ou l'incorporation[9] Chaddah (نّ) pour indiquer une lettre doublée et le Tanwin (نٌ, نً, نٍ) pour indiquer la marque syntaxique de noms singuliers indéfinis.

4.2 Voyelles arabes

Les voyelles ou diacritiques appartiennent à une classe de symboles dans l'écriture arabe. Nous trouvons quatre classes principales dans la langue arabe pour les voyelles :

- Les voyelles longues : Alif Waw Yay.
- Les voyelles brèves : sont les petits signes qu'on met au-dessus ou au-dessous des lettres.
- Double voyelle : Shaddah « ◌ّ » et Tanwin « ◌ً », « ◌ٌ », « ◌ٍ » (figure 8).
- Hamza

[9] Gemination ou incorporation sont utilisés pour indiquer une lettre doublée qui est habituellement marquée par Chaddah (نّ) dans le texte voyellé. Chaddah ne figure pas dans le texte non-voyellé. Par conséquent, l'absence de Chaddah représente un défi pour les analyseurs morphologiques des textes arabes.

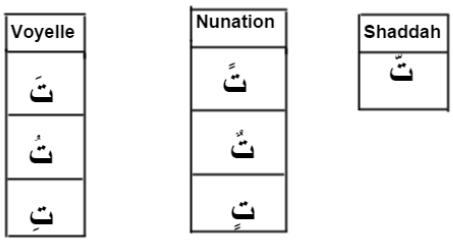

Figure 7 : Différents types de voyelles arabes (voyelle courtes, Nunation et Shaddah)

Dans la langue arabe, les consonnes sont toujours écrites et les voyelles sont facultatives. Par conséquent, l'arabe écrit peut être entièrement voyellé, partiellement voyellé, ou entièrement non voyellé. Généralement, les textes arabes sont non voyellés sauf les textes religieux, les textes utilisés dans l'éducation des enfants et les poèmes. Dans l'arabe moderne, certaines voyelles sont indiquées pour aider les lecteurs à lever l'ambiguïté de certains mots.

4.3 Nature non-linéaire des mots arabes

Les grammairiens arabes considèrent que le but de dérivation est de créer un mot à partir d'un autre en suivant certains schèmes (figure 9). Ces schèmes transmettent les caractéristiques morphologiques et sémantiques aux mots dérivés. Au cours de la procédure de dérivation, des changements peuvent se produire aux lettres de la racine, telles que l'assimilation, l'élision et la gémination (Clark, 2007). Par exemple, le pluriel du mot قَلْب qalb «cœur» est قُلُوب qulub «cœurs»; cela est réalisé en ajoutant la lettre و waw comme un infixe entre la deuxième et la troisième lettre de la racine. Le pluriel du mot مِصْبَاح MissBa'h « Lumière » est مَصَابِيح masabi'h est formé en utilisant le schème spécial de pluriel cassé مَفَاعِيل mafa'il qui réorganise les lettres de la racine et les infixes.

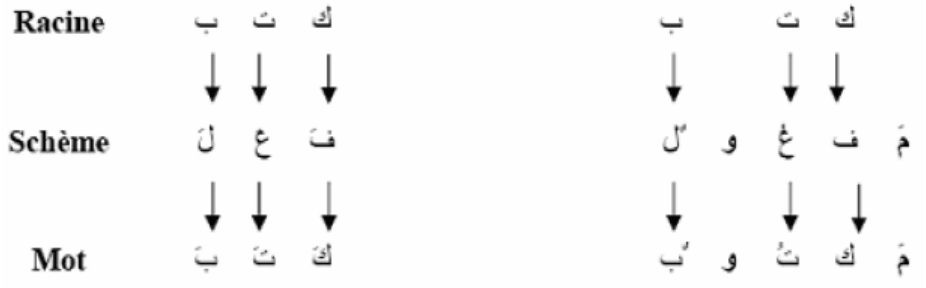

Figure 8 : dérivation de la racine كتب ktb

4.4 Les clitiques arabes

Les clitiques sont les conjonctions, les prépositions et les particules qui sont attachées au début et à la fin des mots. Selon cette classification en clitiques ou affixes, l'article défini est classé comme un proclitique plutôt que d'un préfixe parce que l'article défini ne fait pas partie de la racine, même s'il ne peut pas apparaître comme un mot autonome. Par conséquent, la détermination des formes d'un mot n'est pas possible, car nous ne pouvons pas énumérer toutes les variantes morphologiques de chaque mot. Par exemple, le mot بِوَالِدَيْهِ « Biwalidayhi » « chez ses parents » se compose de quatre morphèmes : بِ bi « avec » est une préposition وَالِدَ, « Walida » «parent» est la tige de nom, ي «deux» est une double lettre et هِ « hi » « son » est un pronom relatif. Le proclitique بِ bi «dans» et l'enclitique هِ « hi » «son» sont appelés clitiques productifs.

La notion des clitiques arabes a participé à la richesse morphologique des formes verbales et des formes nominales de mots arabes.

4.5 Haut degré d'ambiguïté

L'arabe a un haut degré d'ambiguïté pour plusieurs raisons, comme par exemple, l'assimilation ou l'élision des voyelles, l'interaction entre les affixes et les racines radicaux, la segmentation des mots en leurs morphèmes, la ponctuation…

4.6 Assimilation ou élision des voyelles

La présence de voyelles longues dans certaines racines change ou supprime les lettres faibles au cours du processus de dérivation. Par exemple, la lettre faible و « waw » de la racine قول q-w-l est changée en une autre voyelle ou est supprimée de la racine en fonction de l'environnement vocalique. Elle est changée en ا «Alif » dans le verbe passé قَالَ qal « il a dit », et en ي ya dans le passé passif du verbe قِيلَ qila « il est dit » et il est supprimé avec la première personne de verbe au passé قُلْتُ qultu « j'ai dit ».

4.7 Interaction entre les affixes et les racines

Les affixes et les clitiques d'un mot peuvent être homographiques avec les lettres du mot ce qui signifie que l'analyseur morphologique doit faire face à des mots dont les clitiques et les affixes interagissent avec les lettres du mot en produisant toutes les analyses possibles de ces mots. Par exemple, le mot بِطَاقَاتْ bitaaqat peut avoir deux analyses possibles :

Analyse 1 : on considère que la première lettre du mot comme une préposition proclitique بِ «avec», où la racine est ط-و-ق et il signifie « avec les habilités ».

Analyse 2 : la deuxième possibilité est de traiter la première lettre comme une lettre du mot, dans ce cas la racine est ب-ط-ق et il signifie « cartes ».

4.8 Défi de segmentation des mots en leurs morphèmes

Les segments du mot hors contexte peuvent être segmentés en différentes séquences de morphèmes. Par conséquent, les analyseurs morphologiques doivent investiguer correctement toutes les variantes possibles des mots hors contexte (Keskes et al, 2014). Les morphèmes tels que ت ta peut être attachés à des verbes pour indiquer la deuxième personne du singulier au masculin ou au féminin. Par exemple, le morphème ت ta du mot فرمت frmt peut être analysé comme : فَرَمْتَ faramta « tu as haché » (deuxième personne du singulier au masculin) ou فَرَمْتِ faramti « tu as haché » (deuxième personne du singulier au féminin). La même forme peut impliquer un seul morphème فَرْمَتَ farmata « il a formaté » qui représente un mot étranger ; ou trois morphèmes فَرُمْتَ = ت +رم +ف « vous souhaitez » qui a la racine روم r-w-m ; ou فَرَمَتْ= ت +رم +ف faramat « elle a lancé » à partir de la racine رمي r-m-y.

4.9 La ponctuation

L'histoire de la ponctuation arabe moderne a commencé dans la période de la renaissance arabe. Le texte de l'arabe standard moderne est caractérisé par l'incohérence et l'irrégularité dans l'utilisation de signes de ponctuation. En plus de l'introduction tardive de ponctuations aux textes arabes, l'absence d'une formalisation complète de la ponctuation dans les livres de grammaire arabes augmente le problème d'incohérence lors de l'utilisation de la ponctuation dans l'écriture arabe moderne. En outre, l'utilisation de la ponctuation dans le texte arabe est prescriptive plutôt que basée sur une description linguistique de l'utilisation réelle dans les échantillons écrits (Khafaji 2001) et (Attia 2008).

4.10 Segmentation en phrases

La segmentation d'un texte arabe en phrases est une phase délicate, car la ponctuation n'est pas systématique et parfois, ce sont les particules qui délimitent les phrases (Belguith Hadrich et al, 2005).

Selon (Ouersighni, 2001) la segmentation d'un texte arabe peut se faire en deux phases :

- Une segmentation simple basée sur la ponctuation.
- Une segmentation morphologique basée sur les marqueurs morphosyntaxiques ou des mots fonctionnels comme أي « c.a.d », و « et », أو « ou », حتى « quand », لكن « quand »…

Néanmoins, ces marqueurs morphosyntaxiques peuvent jouer un autre rôle dans le texte que celui de séparer les phrases.

4.11 Ambiguïté contextuelle

Le langage naturel est ambigu par nature. Cette ambiguïté, on la constate dans plusieurs niveaux lors du traitement automatique de langue. Par exemple, un mot peut avoir plusieurs sens et plusieurs catégories. Le sens ou la catégorie correcte d'un mot arabe dépend du contexte comme montre l'exemple ci-après.

كتب قصة.	كتب رائعة.
Ktb ksap	ktb raiap
Ktb/VB ksa/NN ./PUNCT[10](1)	Ktb/NN raiap/ADJ ./PUNCT(2)
"Il a écrit une histoire."	"Des merveilleux livres."

Cet exemple montre que le mot « ktb » est ambigu, car il a plus d'une étiquette possible. Dans la première phrase (1) le mot « ktb » est un verbe (écrire), par contre, dans la deuxième phrase (2) le mot « ktb » est un nom (livres).

4.12 Détection des racines

La langue arabe est basée sur des flexions et des dérivations, et les mots ont de nombreuses formes différentes qui résultent de ces phénomènes. Par conséquent, extraire la racine est un problème délicat pour la langue arabe.

Afin de détecter la racine d'un mot arabe, il faut connaître le schème duquel il a été dérivé et supprimer les éléments flexionnels (préfixes, suffixes, antéfixes, post fixes) qui ont été ajoutés au mot. Dans le cas où un mot pourrait être dérivé de plusieurs schèmes différents, la détection de la racine, de laquelle il a été dérivé, est encore plus difficile, en particulier en absence des voyelles (Attia, 2000).

Prenons l'exemple du mot arabe ايمان « Ayman », il y a plusieurs façons possibles de segmentation :

[10] Nous utilisons la translittération de Buckwalter pour cet exemple.

- Les préfixes possibles sont « Ø » ou ا « A » ou اي « Ay ».
- Les suffixes possibles sont « Ø » ou ان « An ».

Ce mot peut aussi représenter un nom propre إيمَان Imène.

5 Conclusion

L'objectif de ce chapitre est de présenter et de montrer certains aspects et caractéristiques de la langue arabe qui nous semblent importants pour la suite de ce travail.

Nous avons commencé par une présentation de la langue arabe ainsi que de ses variétés. Puis, nous avons montré la difficulté du traitement automatique de la langue arabe. En dernière étape, nous avons relevé les caractéristiques morphologiques de la langue arabe, telles que la richesse du mot graphique, l'ambiguïté due à l'absence de voyelles et l'agglutination des mots.

Le domaine du TAL a connu un progrès avec l'évolution de l'informatique. Par contre, ce progrès reste limité pour la langue arabe, car on ne trouve pas des systèmes et outils efficaces pour elle. Cela nous force à améliorer les outils et les systèmes existants pour l'arabe avant de l'utiliser dans nos travaux.

Chapitre 4 : État de l’art de la reconnaissance de l’écriture manuscrite

1 Introduction

La reconnaissance de l'écriture manuscrite fait partie de notre système d'ALAO, il sera employé dans la phase d'apprentissage de l'écriture des caractères et des mots. Dans ce chapitre, nous présentons l'état de l'art de la reconnaissance de l'écriture manuscrite. Tout d'abord, nous allons exposer l'évolution de la reconnaissance de l'écriture. Puis nous décrivons l'architecture générale d'un système de reconnaissance de l'écriture manuscrite. Ensuite, nous citons les différents problèmes liés au domaine de reconnaissance de l'écriture. Enfin, nous décrivons les différentes étapes à suivre pour l’utilisation de l’approche statistique de réseaux de neurones dans un système de reconnaissance de l’écriture manuscrite.

2 Évolution de la reconnaissance automatique de l’écriture manuscrite

L'écriture est un des moyens les plus importants de communication. Il a été utilisé depuis l'antiquité où les symboles ont été utilisés pour exprimer ou donner des informations utiles. Plus tard, l'écriture a été opérée en utilisant un stylo et papier. Aussi, elle a été utilisée pour des besoins personnels comme les rappels ou la prise des notes pour nous-mêmes ou pour la rédaction des lettres, etc.

2.1 Définition

La reconnaissance automatique de l'écriture manuscrite est la transcription de données manuscrites en format numérique utilisable par l'ordinateur. Ce domaine a été l’objet des recherches scientifiques depuis les années 1950. Depuis cette date, le domaine de la reconnaissance automatique de l'écriture manuscrite commence à progresser. Deux thèmes étaient évoqués à l’époque ; le premier s’intéressait aux approches et méthodes du traitement de ce domaine, le second, s’intéressait aux différentes catégories et domaines d'utilisation de cette technologie.

La reconnaissance de l'écriture peut être classée en deux domaines :

- Reconnaissance en ligne : utilisée avec le stylet numérique ou l’interface tactile.
- Reconnaissance hors ligne : utilisée pour la reconnaissance de l’écriture scannée, la vérification des signatures sur les chèques bancaires, l’écriture sur les photos, etc.

En reconnaissance de l'écriture manuscrite en ligne, des signaux d'écriture sont capturés à partir des traces de stylo sur la surface d'un bloc-notes. Les signaux capturés représentent l’entrée du module de la reconnaissance qui transforme ces signaux en texte digital. En reconnaissance de l'écriture manuscrite hors-ligne, les images statiques de mots écrits sont utilisées comme entrée du système. La figure ci-dessous montre la différence entre les deux domaines. Une différence entre les deux techniques de la reconnaissance est que dans le cas de la reconnaissance en ligne de l'écriture, une phase du traitement rapide et immédiate est nécessaire, alors que dans le cas de la reconnaissance hors ligne le système peut traiter l’entrée sans aucune contrainte du temps. Cependant, avec le progrès technologique, cela pourrait ne pas être toujours le cas, car il est possible de recueillir les formulaires contenants du texte manuscrit en ligne, puis de les traiter dans un système de reconnaissance en ligne sans aucune contrainte du temps.

Figure 9 : Différence entre les systèmes de reconnaissance de l'écriture manuscrite en-ligne et hors-ligne.

Afin de permettre une meilleure compréhension de contenu de ce chapitre, nous introduisons les définitions suivantes :

Point : c'est la représentation d'un point par ces deux coordonnées X et Y à valeurs entière.

Stroke : c'est la représentation, sous forme d'un tableau, d'une suite de points formant un tracé continu (sans main levée). Ce tableau est composé de deux sous-tableaux de même dimension. Le premier donnant les abscisses des points composant le tracé, le second les ordonnées. Pour un segment de n points, nous avons la représentation suivante d'un stroke : [X1, X2….., Xn] [Y1, Y2, …, Yn].

Encre : l'encre est l'élément que l'on traite par le système de reconnaissance de l'écriture pour proposer une interprétation et des alternatives. Une encre est un tableau composé d'un ou de plusieurs strokes. L'ensemble représente le tracé que l'utilisateur a réalisé et dont nous cherchons l'équivalent textuel.

Dans cette optique, une encre peut être définie par des signaux qui représentent les trajectoires de stylo. Ces derniers sont enregistrés comme des coordonnées x et y de chaque point rassemblées avec éventuellement toutes les informations sur la pression de chaque point. Par contre, les signaux hors ligne sont les informations enregistrées dans une image avec un format particulier tel que TIFF ou JPEG. La figure ci-dessous montre les différences entre les deux signaux.

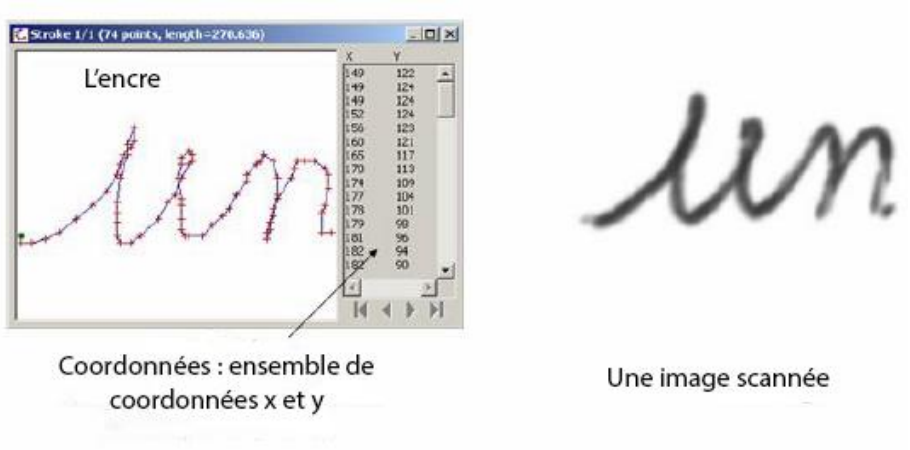

Figure 10 : Signal en-ligne et signal hors-ligne

2.2 Évolution

Les premiers travaux sur la reconnaissance de l'écriture manuscrite ont été réalisés dans les années cinquante et soixante. À cause de la mauvaise performance réalisée par ces systèmes, peu de recherches sur la reconnaissance d'écriture ont eu lieu au cours des années quatre-vingt. Le problème de la reconnaissance de l'écriture manuscrite a été considéré comme étant très difficile à résoudre (Lorette, 2013). Bien que certains systèmes existants aient bien fonctionné pour quelques types d'applications spécifiques, des inconvénients demeurent, à savoir :

- Il est difficile d'analyser la façon dont les systèmes travaillent.
- Il est impossible de localiser précisément l'origine des erreurs faites par ces systèmes et de les corriger afin d'améliorer leurs performances.
- Ils ont besoin d'une longue durée pour donner les résultats.
- Ils ont besoin de très grandes bases de données pour l'entrainement de leurs modèles et pour l'apprentissage.

Le tableau suivant montre l'évolution des systèmes de la reconnaissance de l'écriture manuscrite depuis 1950 :

Période	Approches	Remarques
1950	Le mot était modélisé comme étant composé par plusieurs séquences ; il est défini par les coordonnées de leurs sommets et des bords de chaque mot.	La qualité de la reconnaissance dépend fortement de la capacité du système à segmenter chaque mot.
1960	Une interprétation est faite par le	Cette période a vu une

	système après chaque segmentation.	amélioration rapide en acquisition de l'image avec des progrès de la qualité des équipements.
1970	Développement des approches algorithmiques.	La recherche a consisté principalement à la détection du bord, à trouver le sommet et la segmentation. Nouveaux domaines abordés tels que la reconnaissance des formes.
1980	Une nouvelle direction dans la vision d'ordinateur a émergé sous la forme de vision active. La perception visuelle est traitée comme un processus actif parce que le système de vision s'adapte constamment à un changement d'environnement.	Evolution de la théorie de la décision (arbre de décision)
1990	Des algorithmes plus efficaces : programmation dynamique, modèle de Markov caché, réseaux de neurones, etc.	Le domaine de la reconnaissance a eu plus d'intérêt avec l'apparition de nouveaux besoins (la Poste, applications bancaires, ordinateurs portables), avec l'apparition des nouveaux systèmes d'acquisitions plus appropriés (scanners, papier numérique...)
2000	La combinaison ou la coopération de plusieurs systèmes de reconnaissance indépendants, l'utilisation des lexiques ou dictionnaires et des modèles de langage.	À cette époque, le post-traitement a été suggéré pour améliorer l'efficacité globale du système.

Tableau 10 : évolution des systèmes de la reconnaissance de l'écriture manuscrite depuis 1950

2.3 Limitations du système de reconnaissance de l'écriture

Bien qu'il existe de nombreuses applications de reconnaissance de l'écriture manuscrite en ligne, la technologie n'est pas encore à pleine maturité. Il reste encore quelques améliorations peuvent encore être apportées aux systèmes de reconnaissance de l'écriture manuscrite.

En écriture en ligne, le signal d'entrée est constitué d'une séquence temporelle de strokes. Un stroke représente l'écriture du stylo dès le moment de début de l'écriture (stylo en

bas) jusqu' au moment où il est levé (stylo en haut). Généralement, les caractères manuscrits sont généralement écrits en séquence, et chaque caractère doit être terminé avant le début de la prochaine. De plus, les caractères suivent généralement l'ordre spatial, de gauche à droite, sauf dans certains caractères qui contiennent des points (Exemple : i et j) et des croix ou traits (Exemple : t et x). Dans ces cas, la partie sous-jacente d'un mot est d'abord écrite, puis le mot est complété par l'ajout des croix et des points. La présence de ces coups retardés pose quelques problèmes, si on ne les traite pas, on n'obtiendra pas une bonne reconnaissance du texte manuscrit à l'entrée.

Dans certaines applications et dispositifs, afin de fournir de bonnes performances de reconnaissance, des contraintes doivent être imposées aux utilisateurs, comme la position pour écrire. En général, il n'y a pas de système qui peut être utilisé dans tous les environnements. Chaque système a des contraintes pour travailler dans un environnement en particulier.

Dans de nombreuses recherches en reconnaissance de l'écriture manuscrite, les systèmes dépendent du type d'utilisateur, ainsi que de ses besoins. Par exemple, le système pourrait être destiné seulement à la reconnaissance de caractères ou de chiffres manuscrits ou des mots d'un petit lexique spécifique.

Avec le progrès technologique, certaines contraintes dans l'écriture manuscrite sont réduites, mais, il reste quelques problèmes complexes parce que le système de reconnaissance doit gérer diverses limitations (type d'écriture, segmentation, bruit, etc), ce qui va affecter la précision de la reconnaissance.

2.4 Architecture du système de reconnaissance de l'écriture

Actuellement, il existe de nombreuses techniques pour la reconnaissance automatique de l'écriture manuscrite. Par contre, ces systèmes ont un modèle générique de reconnaissance comme dans la figure ci-dessous. L'entrée du système est le mot (ou la phrase) à reconnaître qui est sous la forme d'une encre représentant la trace du crayon ou stylet (les strokes). La sortie du système est une représentation de texte entré sous format d'une encre au système. Dans le modèle, il y a trois composantes principales ; le module d'entrée, le module de reconnaissance et le module de post-traitement. Chaque module effectue ses fonctions nécessaires comme étant des sous-modules dans les modules principaux.

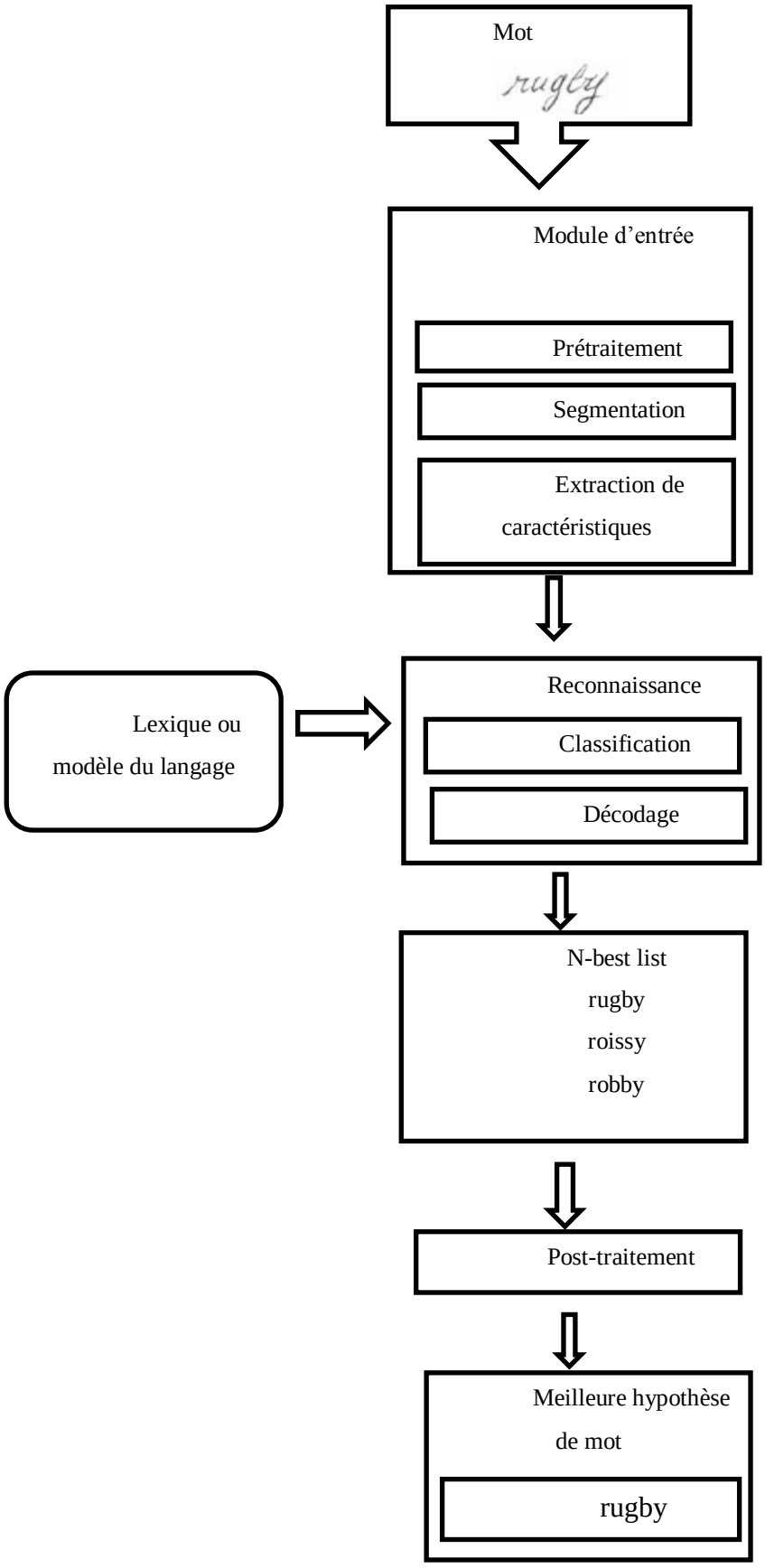

Figure 11 : Schéma générique d'un système de reconnaissance de l'écriture manuscrite.

Premièrement, le mot manuscrit inconnu présenté au système doit être transformé en une forme compréhensible pour le moteur de la reconnaissance. Le module d'entrée doit extraire des informations à partir du mot présenté à l'entrée dans un état propre avant de les passer au module de la reconnaissance. Par conséquent, dans le module d'entrée, l'encre doit être d'abord prétraitée pour enlever les informations indésirables et inutiles qui vont provoquer des difficultés dans le processus de reconnaissance. De plus, ce module effectue des opérations telles que la détection des lignes de référence et la rectification de quelques variations (la rotation du mot, la taille et l'inclinaison du mot) pour améliorer la qualité de l'encre. Deuxièmement, les mots sont segmentés en une séquence des strokes de base tels que des caractères ou parties de caractère. En troisième lieu, l'unité prétraitée doit être segmentée et transformée en une représentation des fonctions compactes. Ce processus consiste à extraire les caractéristiques discriminantes pour construire une liste de vecteurs de caractéristiques afin de l'utiliser dans la phase de la reconnaissance.

Le module de reconnaissance dans le système implique l'utilisation d'un module entraîné qui est capable de reconnaître les unités de base individuelles (les strokes). Ce dernier est appelé modèle du langage. La reconnaissance de l'écriture manuscrite intègre un module de comparaison du motif (ou pattern) avec les modèles de la classe de référence. De plus, ce module mesure un score de similarité (score de la probabilité) entre le motif de test et le motif de similarité. Le score de motif de similarité est utilisé pour décider quel modèle correspond le mieux au motif inconnu (motif à reconnaître).

Le module du post-traitement est utilisé pour vérifier la liste des N meilleurs candidats et peut également procéder au rejet des hypothèses improbables. Avec l'aide de certaines ressources linguistiques telle que le modèle du langage, certaines améliorations dans la reconnaissance peuvent être obtenues. Ces ressources peuvent être un lexique, qui est un dictionnaire des mots ou un modèle de langage qui peut inclure certaines propriétés statistiques ou structurelles d'un langage donné.

2.5 Problèmes de la reconnaissance de l'écriture manuscrite

De nombreux chercheurs ont mené des recherches dans le domaine de la reconnaissance de l'écriture manuscrite dans ces dernières années. Bien que quelques problèmes aient été résolus, il reste encore de nombreux problèmes non résolus. Malgré les progrès scientifiques et le développement de l'informatique la capacité d'un système de

reconnaissance d'écriture reste toujours incomparable à la capacité humaine de reconnaître l'écriture manuscrite.

Parmi les problèmes de l'écriture manuscrite, deux êtres humains n'ont pas exactement la même écriture pour le même mot. Entre les personnes, la variabilité peut inclure l'inclinaison, la taille des caractères, la forme et la façon cursive de l'écriture. Les variations dans l'écriture peuvent également être au niveau des applications, les écritures sont normalement guidées par les champs du texte. Les deux figures ci-dessous montrent un échantillon aléatoire de l'écriture pris à partir de la base de données IRONOFF (Gaudin et al, 1999) qui montre ces différences et un autre échantillon pris à partir de la base de données de (Tappert et al, 1994).

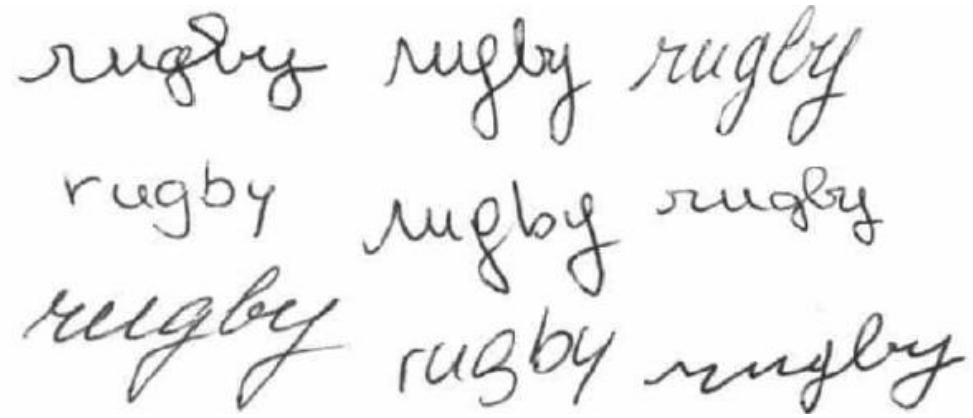

Figure 12 : Variations de style de l'écriture dans l'échantillon de IRONOFF.

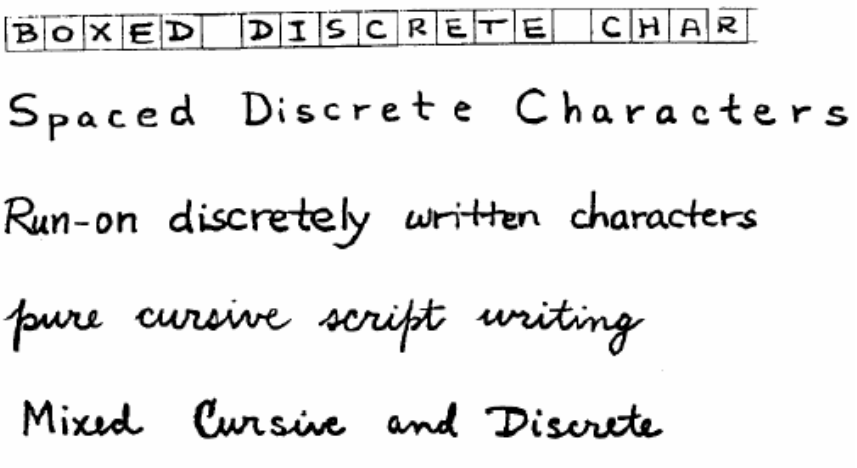

Figure 13 : Différents types d'écritures (Tappert et al, 1994).

D'une manière générale, les personnes ne respectent pas les règles de l'écriture ; par exemple, certains écrivains écrivent de manière très cursive, d'autres préfèrent écrire des caractères disjoints lors de l'écriture d'un mot. Pourtant, certains, mélangent l'écriture cursive et l'écriture disjointe sans aucun ordre ou règle particulière, comme on le voit dans les deux

figures ci-dessus. Ces habitudes d'écriture affectent les performances du module de segmentation d'un système de reconnaissance de mots, car le module trouve des difficultés à déterminer les meilleurs points de segmentation définitifs pour chaque mot.

3 Les réseaux de neurones

Il existe de nombreuses méthodes pour entraîner et modéliser un système de reconnaissance de l'écriture manuscrite. Parmi les méthodes existantes, on cite le Modèle de Markov caché (HMM), les réseaux de neurones, le système expert, le voisin k-plus proche et d'autres techniques ou une combinaison de ces techniques. Certains chercheurs divisent ces méthodes en deux classes principales :

- Syntaxique : la classe qui implique la description des formes des caractères d'une manière abstraite.
- Statistique : où le système apprend à partir des données directement, sans avoir spécifié explicitement la structure des connaissances au système.

Comme nous allons travailler sur l'apprentissage des caractères dans notre plate-forme, nous allons adapter l'approche du réseau de neurones, car elle donne de meilleurs résultats au niveau caractère.

3.1 Définition

Le réseau de neurones est un modèle formé de plusieurs cellules élémentaires simples, fortement interconnectées. Chaque cellule du réseau est appelée « neurone ». De plus, les réseaux de neurones sont capables de stocker des connaissances empiriques et de les rendre disponibles à l'usage. Ces derniers sont des classificateurs discriminants utilisés souvent dans l'entraînement des systèmes (Jain et al, 2000). La littérature sur le RN est énorme parce qu'il a été très largement utilisé dans de nombreux domaines (Informatique, physique, chimie, finance, santé, etc.). Les RN ont été utilisés dans le système de reconnaissance de l'écriture manuscrite avec succès. Cependant, par rapport aux autres approches, ils nécessitent plus de calculs. Le RN est un classificateur statique qui nécessite des fonctions vectorielles de taille fixe. En raison de ces propriétés, le RN est considéré comme une approche très performante dans la reconnaissance de caractères, de chiffres et des graphèmes.

Comme toutes les autres approches, le RN a quelques faiblesses comme classificateur discriminant, parmi lesquelles :

- Généralement, le RN est connu par son besoin d'une quantité importante de données pour l'entraînement. Par conséquent, il n'est pas adapté dans le cas d'entraînement du système avec de petites ressources (Ganapathiraju et al, 2004). Donc, il est assez difficile d'appliquer l'approche du RN avec une quantité limitée de données d'entraînement.
- Le processus d'optimisation de base de gradient lors de l'apprentissage du RN dépend du principe de minimisation du risque empirique (empirical risk minimization) qui utilise l'algorithme de rétro-propagation (Rumelhart, 1986). Bien que cette méthode garantisse une bonne performance sur les données d'apprentissage, par contre, la performance sur les données de test n'est pas garantie.
- L'entraînement du système est considérablement plus lent. En fait, l'entraînement du RN ne garantit pas une optimisation maximale.

Théoriquement, un réseau de neurones est une combinaison de plusieurs fonctions élémentaires appelées neurones (ou neurones formels). Chaque neurone formel réalise, à chaque instant t, une fonction non linéaire, qui, au sein du réseau, représente soit les sorties des neurones, soit des variables exogènes. La figure ci-dessous montre une représentation graphique d'un réseau de neurones.

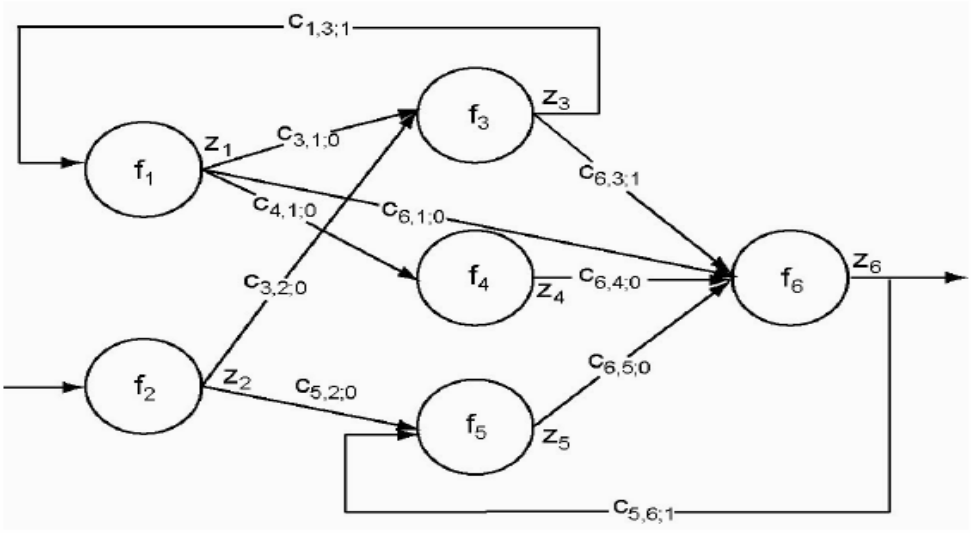

Figure 14 : réseau de neurones représenté par un graphe orienté.

3.2 Les perceptrons multi-couches

Généralement, les perceptrons multi-couches (PMC) (Badran et al, 2001) sont qualifiés souvent par le terme « boîtes noires », car, il est difficile de les interpréter, les

implémenter et les régler, ils sont souvent utilisés dans le domaine de la reconnaissance (écrit, visage, parole, signature, etc.).

3.2.1 Architecture d'un PMC

Un PMC contient trois couches : couche d'entrée, une ou plusieurs couches cachées et une couche sortie. Si le neurone de chaque couche est connecté à tous les autres neurones de la couche inférieure, dans ce cas, le réseau est appelé réseau complètement connecté.

La figure ci-dessous montre un PMC à quatre couches. Dans ce PMC, la couche d'entrée contient trois paramètres, la première couche cachée possède cinq neurones, la seconde couche contient quatre neurones. Enfin, la couche de sortie (phase de décision) ne doit contenir qu'un seul neurone.

Dans un PMC, le but des couches cachées est l'extraction de l'information pertinente à partir de l'entrée afin de résoudre les problèmes de décision. Généralement, les entrées sont présentées dans leurs formes brutes (données originales).

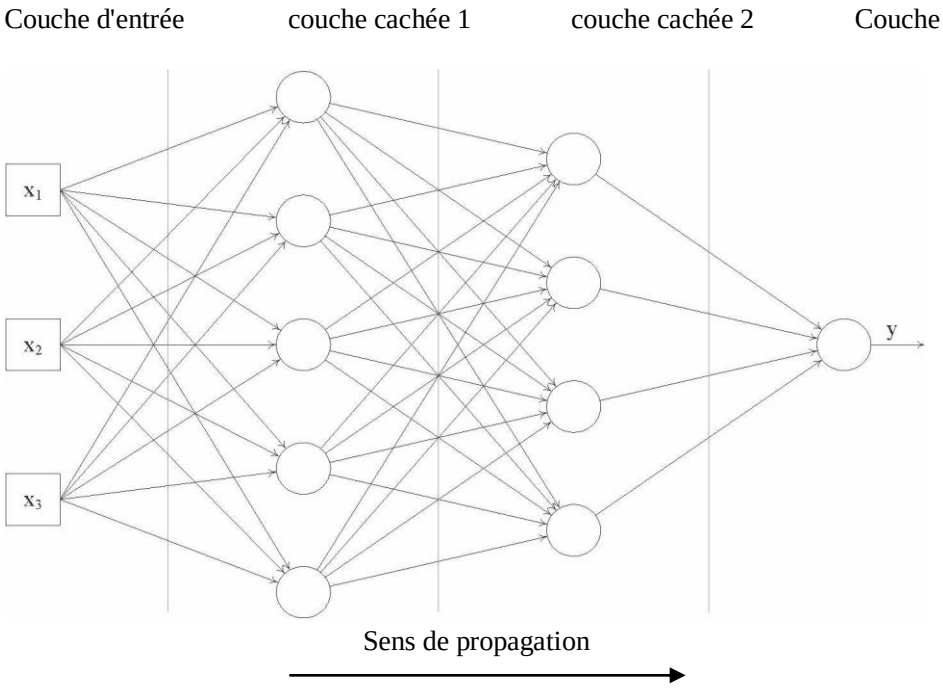

Figure 15 : Exemple d'un perceptron multi-couches

Généralement, le calcul de la sortie dans un PMC se fait de gauche à droite, on commence par le calcul de sortie de chaque neurone à partir de la première couche cachée jusqu'à la dernière couche cachée et, enfin, on calcule la sortie y. Si on applique le PMC sur un système de reconnaissance de caractères, généralement, on trouve dans la sortie du réseau le même

nombre de neurones que celui de classes à reconnaître (par exemple, on trouve 10 neurones dans le cas des chiffres de 0 à 9). Généralement, pour déterminer la décision finale en reconnaissance de l'écriture, on utilise un vecteur qui détermine les probabilités d'une classe par rapport à l'entrée de PMC. Selon (Bridle, 1990), il faut utiliser la fonction Softmax comme fonction de transfert pour obtenir en sortie de probabilités entre 0 et 1.

$$\text{Softmax}(v_i) = P(Classe = Ci | X = x) = \frac{e^{v_i}}{\sum_k e^{v_k}}$$

Avec :

- i est l'indice du neurone qui correspondant à la classe Ci.
- k est l'indice balayant tous les neurones qui appartient à la même couche.
- v désigne le potentiel du neurone.
- x correspond à l'entrée des neurones.

Concernant le calcul dans les couches cachées, il existe deux fonctions de transfert possible ; la fonction sigmoïde et la fonction tangente hyperbolique (Harrington, 1993).

$$a - Sigmoide(v_i) = \frac{1}{1+e^{-v_i}} \text{ entre [0 et 1]}$$

$$b - Tangente\ hyperbolique(v_i) = \frac{\sinh(v_i)}{\cosh(v_i)} = \frac{e^{2v_i}-1}{e^{2v_i}+1}$$

3.2.2 Apprentissage

Avant d'associer un élément à une classe, il faut tout d'abord apprendre à discriminer cette classe et être capable d'extraire les règles ou les problèmes avec quelques échantillons en apprentissage, car il est impossible avec une seule encre de récupérer tous les prototypes possibles.

3.2.3 Les bases d'apprentissage

En général, l'apprentissage se fait d'une manière supervisée, à partir d'une base de données suffisamment importante, découpée en deux, voire trois sous-bases. La première sous-base est utilisée pendant la phase d'apprentissage. Elle permet d'apprendre les poids du réseau de neurones. La deuxième sous-base est appelée base de test. Elle est utile dans le réglage des méta-paramètres de l'algorithme d'apprentissage (nombre de couches cachées, pas de gradient, nombre de neurones, etc.). Elle permet l'évaluation optimiste des différentes erreurs. Enfin, la troisième sous-base, si on l'utilise, elle sert comme base de validation afin

d'évaluer le système sur des nouveaux exemples. Le découpage de ses bases dépend de nombre d'échantillons et de nombre de paramètres du PMC.

3.2.4 Algorithme d'apprentissage

Le but de l'apprentissage est la correction des erreurs liées aux classements des données. Au cours de l'entraînement supervisé du système, ce dernier adapte ses paramètres PMC ajustables pour que la sortie tende vers le résultat désiré.

Principalement, on utilise les méthodes, qui sont très efficaces, utilisant le gradient pour adapter les poids. Les algorithmes de calcul de gradient conjugué sont nombreux, par contre, ils sont coûteux en calculs et en mémoire. À cause de ces contraintes, généralement la méthode de descente de gradient stochastique est la plus utilisée dans les processus d'apprentissage. Cette méthode est itérative et elle suit la règle suivante pour la mise à jour (Bottu, 2012) :

$$W_{l,j,i}^{n+1} = W_{l,j,i}^{n} - \mu^{n} * \Delta E(W_{l,j,i}^{n})$$

Avec :

μ^n : désigne le pas de descente.

l : correspond à l'indice de la couche en cours.

i : correspond à l'indice du neurone.

$$\text{Gradient local} : \nabla E_{loc_e}(w_{l,j,i}^{n}) = \frac{\partial d(Y(X_e, W^n), Yd(X_e))}{\partial w_{l,j,i}^{n}} = \frac{\partial d(Y(X_e, W^n), Yd(X_e))}{\partial Y(X_e, W^n)} \frac{\partial Y(X_e, W^n)}{\partial w_{l,j,i}^{n}}$$

$$\text{Gradient total} : \nabla E(w_{l,j,i}^{n}) = \sum \nabla E_{loc_e}(w_{l,j,i}^{n})$$

Équation 1 : formule de calcul du gradient local et total

L'utilisation de l'approche de la rétro-propagation (LeCun, 1987) est très efficace si on souhaite calculer le gradient de la fonction d'erreur d'un PMC, par rapport à la méthode directe (Rossi, 1995). Dans notre système de reconnaissance de l'écriture manuscrite, nous allons adapter cette méthode de calcul du gradient.

Étant donné :

```
e : désigne les exemples de la base d'apprentissage
Nb-e : représente le nombre d'exemples de la base
d'apprentissage
c : indique le numéro de la couche cachée du PMC
Nb-c : indique le nombre de différentes couches
cachées du PMC
Tant que la condition d'arrêt n'est pas satisfaite
  Pour e = 1 jusqu'à Nb-e
     Choisir un exemple de la base d'apprentissage d'une façon aléatoire
       Propagation des caractéristiques de l'exemple de la couche d'entrée
```

```
        vers l'état sortie
        Calcul de la valeur de l'erreur en sortie du
       réseau
        Pour c = Nb-c jusqu'à 1 (rétro propagation)
           Détermination de la valeur de l'erreur pour chaque neurone de la
           couche cachée c
       Fin Pour c
       Pour c = 1 jusqu'à Nb-c
           Correction et réaffectation des poids du réseau
      Fin Pour c
  Fin Pour e
  Vérification de la condition d'arrêt
Fin Tant que
```

Figure 16 : algorithme d'apprentissage d'un PCM

L'algorithme d'apprentissage se déroule donc en trois étapes. La première étape concerne le parcours du réseau de la gauche vers la droite pour propager les données et calculer l'estimation du réseau en sortie, à la fin de parcours, on obtient le vecteur Y qui correspond à l'entrée Xe. Ensuite, il faut parcourir le réseau de la droite vers la gauche pour estimer les erreurs commises par chaque neurone en partant de la couche de sortie vers la couche d'entrée. Enfin, la troisième étape correspond à la correction des poids du réseau en respectant le sens de la propagation.

3.2.5 Calcul du gradient

Dans cette section, nous allons illustrer l'implémentation de la rétro-propagation en calculant le gradient. Le schéma ci-dessous montre les notations nécessaires à la compréhension des équations.

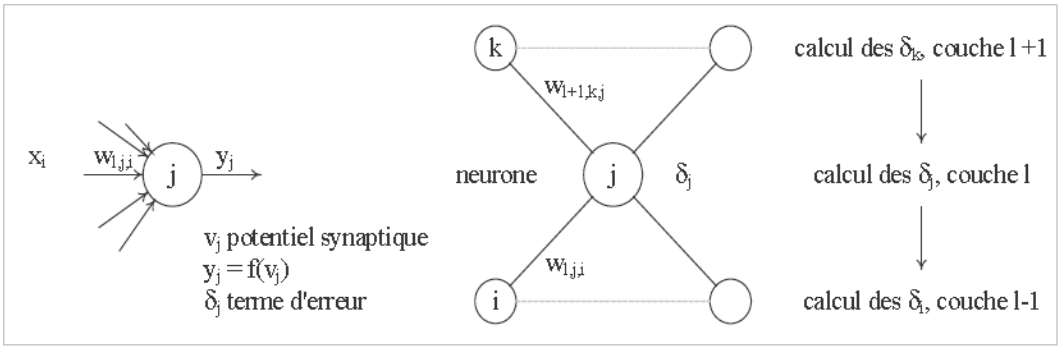

Figure 17 : notations des neurones et calcul de gradient

Dans le domaine de la reconnaissance de l'écriture, la fonction de calcul de coût souvent utilisée est celle de l'erreur quadratique moyenne qui est estimée pour un exemple e, par le carré de la distance euclidienne entre un vecteur yd_k de sortie désirée (1 ou 0) et le vecteur y_k de sortie calculée par le réseau de neurones. Dès qu'on connaît le label du caractère à l'entrée du réseau de neurones, on est capable de déterminer le vecteur des sorties désirées

du réseau. Ce vecteur est mis à 1 pour la sortie correspondant à la classe du caractère et il est à zéro dans les autres classes.

$$E_e(w) = \frac{1}{2} \times \sum_{c=1}^{C} yd_c^e - y_{l,c}^e)^2$$

Avec :

l : représente le numéro de la dernière couche du PMC

Formule de correction des poids : $w_{l,j,i}^{n+1} = w_{l,j,i}^{n} - \mu^n \times \nabla E_e(w_{l,j,i})$

Avec :

l : représente le numéro de la couche du PMC

i, j : désigne l'indice du neurone

Formule de calcul de gradient : $\nabla E_{loc_e}(w_{l,j,i}^n) = \frac{\partial}{\partial w_{l,j,i}^n} (E_e(w))$

$$Gradient\ \nabla E_{loc}(w_{l,j,i}^n) = \frac{\partial E(w)}{\partial v_{i,j}} \times \frac{\partial v_{i,j}}{\partial w_{l,j,i}^n}$$

$$\text{et } \frac{\partial v_{i,j}}{\partial w_{l,j,i}^n} = \frac{\partial}{\partial w_{l,j,i}^n}\left(\sum_k w_{l,j,k}^n \times x_{l-1,k}\right) = x_{l-1,i}$$

$$\text{d'où } \nabla E_{loc}(w_{l,j,i}^n) = \frac{\partial E(w)}{\partial v_{l,j}} \times x_{l-1,i} \text{ on pose } \delta_{l,j} = \frac{\partial E(w)}{\partial v_{l,j}}$$

Afin de calculer le gradient, nous devons mémoriser l'activation de sortie x, le potentiel synaptique v et son indice d'erreur δ. Le calcul de ce dernier diffère en fonction du niveau de la couche. D'abord, on calcule cet indice sur la couche de sortie qui a une équation différente de celles utilisées pour les couches cachées.

Calcul de l'équation pour la couche de sortie :

$$\delta_{l,j} = \frac{\partial E(w)}{\partial v_{l,j}} = \frac{\partial E(w)}{\partial x_{l,j}} \times \frac{\partial x_{l,j}}{\partial v_{l,j}}$$

$$\frac{\partial x_{l,j}}{\partial v_{l,j}} = \frac{\partial}{\partial v_{l,j}} fs(v_{l,j}) = fs'(v_{l,j})$$

Avec :

fs : fonction Softmax et *fs'* sa dérivée.

$$\frac{\partial E(w)}{\partial x_{l,j}} = \frac{\partial}{\partial x_{l,j}}\left(\frac{1}{2} \times \sum_{c=1}^{C}(yd_c - y_{l,c})^2 = -(yd_j - y_{l,j}\right) \text{ car } (x_{l,j} \equiv y_{l,j})$$

D'où $\delta_{l,j} = \left(y_{l,j} - yd_j\right) \times fs'(v_{l,j})$

Maintenant, l'erreur en sortie du réseau est facile à déterminer, elle présente la différence entre le vecteur désiré et la sortie obtenue (propageant l'entrée). Par contre, le calcul d'erreur dans les couches cachées n'est pas évident, car les erreurs sont cachées; pour

cela, il faut estimer l'erreur d'avoir un neurone j qui appartient à la couche 1 a propagé dans les neurones de l'autre couche supérieure l+1 (voir la figure ci-dessus).

Calcul de l'équation pour la couche cachée :

$$\delta_{l,j} = \frac{\partial E(w)}{\partial v_{l,j}} = \sum_k \frac{\partial E(w)}{\partial v_{l+1,k}} \times \frac{\partial v_{l+1,k}}{\partial v_{l,j}}$$

or $\frac{\partial E(w)}{\partial v_{l+1,k}} = \delta$

et $\frac{\partial v_{l+1,k}}{\partial v_{l,j}} = \frac{\partial v_{l+1,k}}{\partial x_{l,j}} \times \frac{\partial x_{l,j}}{\partial v_{l,j}}$

$$\frac{\partial x_{l,j}}{\partial v_{l,j}} = \frac{\partial}{\partial v_{l,j}} fsig(v_{l,j})$$

Avec : *fsig* est la fonction sigmoïde

soit $\delta_{l,j} = \frac{\partial E(w)}{\partial v_{l,j}} = fsig'(v_{l,j}) \times \sum_k \delta_{l+1,k} \times w_{l+1,k,j}$

Par conséquent, l'erreur au niveau des couches cachées est égale à la somme des erreurs pondérées par les poids correspondant aux différentes liaisons avec la pente de son activation. Dès que les indices d'erreur sont calculés à partir de la couche de sortie et en remontant vers la couche d'entrée, on applique la formule de correction des poids sans oublier de choisir un pas adapté. Cette étape est un réglage essentiel sert à la vérification de la bonne convergence de la minimisation de la fonction de coût adapté.

3.2.6 Apprentissage stochastique large

Généralement, l'apprentissage stochastique est plus rapide et moins coûteux comparé à un apprentissage global. Cela est dû à la rapidité de la correction des poids pour chaque exemple présenté, dans le cadre de la reconnaissance de l'écriture manuscrite. En général, le processus d'apprentissage lors de la reconnaissance de mots ou de phrases avec des lexiques de taille importante est très coûteux en temps et en mémoire du stockage. Pour cela, la meilleure solution est de faire une correction après N échantillons en faisant intégrer les erreurs locales dans les N échantillons.

3.2.7 Paramétrage

Lors d'un apprentissage, la rétro-propagation peut être particulièrement lente si on utilise les réseaux multicouches. De plus, dans ce cas, aucune formule ne permet de garantir la convergence de ce réseau vers l'optimum global. Par contre, on peut augmenter la chance du

réseau à converger en ajustant plusieurs facteurs tels que, la normalisation des entrées, l'initialisation des poids (Lecun et al, 1998), la fonction de transfert du type sigmoïde et l'apprentissage global ou stochastique.

3.3 Les réseaux de neurones à convolution

Un réseau de neurones est considéré comme un outil de représentation des connaissances, parmi ces avantages une bonne tolérance aux bruits et la possibilité de lancer un apprentissage automatique des poids avec une forte capacité de généralisation. Nous avons souligné dans la section précédente les principes d'un PMC qui s'appuient sur une vision globale des entrées ainsi que la dimension des entrées. Par contre, l'écriture manuscrite en ligne a des aspects temporels et 2D ainsi que des traits locaux.

3.3.1 Caractéristiques des réseaux à convolution

Généralement, les réseaux de neurones classiques de type perceptron se différencient des réseaux de neurones à convolution par leur architecture et plus précisément par leur caractéristique connexionniste. C'est-à-dire, le neurone d'une couche d'un PMC est connecté à tous les neurones de la couche précédente ; par contre, dans le cas d'un RNC, le neurone n'est connecté qu'à un sous-ensemble de neurones de la couche précédente. De plus, dans un RNC chaque neurone est considéré comme une unité de détection d'une caractéristique locale.

Les RNC présentent quelques contraintes, ils présentent un certain degré d'invariance et de déformation en se basant sur ces trois critères : poids partagés, zones réceptives locales et sous-prélèvement spatial. L'emploi des poids partagés dans un RNC permet la réduction du nombre de paramètres dans le système afin de faciliter la phase de généralisation. Ce type de réseau a été appliqué avec succès dans des systèmes de reconnaissance des chiffres (LeCun et al, 1992).

3.3.2 Réseaux TDNN

Dans le domaine de la reconnaissance de caractères manuscrits, il existe deux types des réseaux à convolution : les TDNN (Time Delay Neural Networks), qui sont les réseaux de neurones à délais temporels, et les SDNN (Space Displacement Neural Networks), qui sont les réseaux de neurones à déplacement spatial. Le TDNN est un réseau à délai employé souvent pour des données séquentielles, il est bien adapté pour la reconnaissance de l'écriture manuscrite en-ligne, par contre les SDNN sont utilisés pour les données de nature spatiales, comme l'écriture hors-ligne par exemple. Pour cette raison, nous avons choisi le réseau du

type TDNN pour le développement de notre système de reconnaissance de l’écriture manuscrite.

L’architecture des RNC se caractérise par deux parties. La première contient les couches basses, implémente toutes les convolutions successives servant à transformer progressivement un ensemble séquentiel de vecteurs caractéristiques en un autre ensemble séquentiel de vecteurs caractéristiques d'ordre supérieur. La deuxième partie correspond à un perceptron multicouche classique, elle prend en entrée toutes les sorties qui viennent de la partie extraction.

4 Conclusion

Dans ce chapitre, nous avons décrit l'état de l'art de la reconnaissance de l'écriture manuscrite en ligne. Nous avons étudié les différentes étapes du système de reconnaissance ; prétraitement, la segmentation, extraction de caractéristiques, la reconnaissance et le post-traitement. Enfin, nous avons montré l'atout de l'utilisation des réseaux de neurones dans le domaine de reconnaissance de l’écriture manuscrite.

Après la présentation de l'état de l'art de la reconnaissance de l'écriture et les difficultés du traitement automatique de la langue arabe, nous allons présenter la méthodologie et les démarches à suivre pour le développement de notre système d'ALAO.

Chapitre 5 : Méthodologie et démarches de recherche

1 Introduction

Nos travaux se trouvent au confluent de deux thèmes de recherche au Laboratoire LIDILEM : le thème d'apprentissage des langues assisté par ordinateur (ALAO) et le traitement automatique des langues (TAL). La thèse que nous présentons est clairement inscrite dans les deux thèmes : la réalisation d'un système d'ALAO et l'exploitation de l'apport du TAL dans l'ALAO.

Après avoir définir l'expression " système complet " et ce qui est nécessaire pour mettre en œuvre efficacement un système d'ALAO, nous présentons la problématique principale de nos travaux de recherche. Ensuite, nous exposons les démarches que nous avons suivies pour arriver à la réalisation de notre plate-forme d'ALAO. Puis, nous décrivons brièvement l'architecture de la plate-forme puisque la description complète de l'architecture sera évoqué dans le chapitre 8. Enfin, nous détaillons le rôle de chaque module avant l'implémentation de la plate-forme.

2 Problématique

Dans la dernière décennie, les systèmes d'apprentissage des langues assistés par ordinateur ont connu un véritable progrès en matière de diversité et de techniques employées. Le domaine de l'ALAO a été forgé par plusieurs théories, plusieurs disciplines et réalisations touchant différentes compétences (linguistique computationnelle, traitement de corpus textuels, etc.). De plus, l'ALAO offre à l'apprenant plusieurs types d'activités : renforcement, test de connaissances, mémorisation, compréhension, activités de découverte, production, etc. Malgré tous ces progrès, le domaine de l'ALAO ne répond pas encore aux besoins des enseignants et des apprenants pour plusieurs raisons. D'abord, dans quelques systèmes d'ALAO, les auteurs conçoivent des activités dédiées à l'enseignement d'une langue étrangère autour d'exercices structuraux de grammaire qui ne prennent pas en compte le contexte. Les interactions apprenant-système restent dans ces systèmes très limités et visent surtout à sanctionner les connaissances de l'apprenant. Par conséquent, ils sont plutôt des environnements de tests des connaissances de l'apprenant et ressemblent plus à un support de cours. Selon (Golonka et al, 2014) la recherche dans le domaine de l'ALAO souffre des mauvais choix des variables à étudier, manque de données pertinentes sur les participants, des études basées sur des utilisateurs non formés à la technologie et un manque d'enquêter sur les principaux facteurs qui peuvent améliorer l'efficacité de l'apprentissage.

D'autres systèmes commencent à utiliser des ressources et outils issus du TAL, mais l'utilisation reste restreinte à cause de l'état imparfait de ces outils surtout dans le cas de la langue arabe. De plus, l'architecture de certains systèmes nécessite des connaissances en informatique pour les utiliser. En ce qui concerne la langue arabe, le nombre des systèmes d'ALAO existants est très réduit, pour la plupart sous la forme d'un prototype, en général jamais utilisé dans des situations réelles d'apprentissage. Enfin, le feedback proposé par les systèmes actuels d'ALAO reste basique et ne répond pas aux besoins de l'enseignant ou de l'apprenant. Cette limitation rend le système incapable de gérer certaines situations d'apprentissage inattendues. Par conséquent, les systèmes d'ALAO existants n'offrent pas une véritable interactivité avec l'utilisateur (enseignant ou apprenant), car, ils ne sont pas en mesure de diagnostiquer les problèmes d'un apprenant avec l'orthographe, la grammaire, la conjugaison, etc., ni de générer intelligemment un feedback adéquat selon la situation de l'apprentissage.

3 Démarche

Un système d'ALAO devrait offrir aux enseignants et aux apprenants des outils pédagogiques supplémentaires et fournir des nouvelles approches pédagogiques.

Parmi les points qu'on vise à résoudre à travers nos travaux de recherche, on cite ces aspects :

- Améliorer l'apprentissage des langues en la rendant plus rapide, plus facile et plus efficace.
- Apporter des moyens innovants pour l'apprentissage qui devraient améliorer les compétences des apprenants plus que les méthodes traditionnelles.
- Intégrer un haut degré d'interactivité entre l'ordinateur et l'apprenant.
- Rendre possible l'accomplissement des tâches complexes (Kongrith et al, 2005).

Pour prendre en considération ces aspects et afin de pallier certaines limitations des systèmes d'ALAO, nous nous proposons d'élaborer dans le cadre de cette thèse, un système d'apprentissage des langues assisté par ordinateur, destiné aux enseignants et aux apprenants de la langue française comme langue seconde ou étrangère et aux apprenants de la langue arabe comme langue seconde ou étrangère. Ce système devrait être capable d'analyser les réponses des apprenants et de générer un feedback adéquat. Notre travail porte essentiellement sur cinq axes.

- La définition d'un système d'ALAO complet : la première phase de notre travail concerne la définition et la conception de l'architecture de notre système d'ALAO.
- Le développement d'un système multilingue pour la reconnaissance de l'écriture manuscrite, dans le but de créer des nouveaux types d'activités pédagogiques pour l'apprentissage de l'écriture.
- L'élaboration de ressources linguistiques et d'outils informatiques. Dans cet axe, nous allons choisir un analyseur morphologique pour l'arabe et le français pour l'analyse de texte. Vu la qualité de l'analyseur morphologique arabe, un travail d'amélioration est nécessaire pour rendre cet outil utilisable dans notre plate-forme. Pour cela nous allons construire un corpus étiqueté pour l'arabe dans le but d'améliorer l'analyse morphologique de textes arabes par l'analyseur.

- L'intégration et le test des outils développés pour le français. Après les phases de développement et d'amélioration de ressources et d'outils TAL, nous allons les intégrer dans notre plate-forme d'ALAO pour les utiliser dans la phase de générations automatique des activités.
- L'intégration et le test des différents modules. Nous complétons le développement afin d'avoir une architecture complète et évolutive, c'est-à-dire, on peut toujours ajouter d'autres modules ou d'autres outils TAL.
- Les tests et évaluations en milieu réel. Dès que le système est complet, nous allons le mettre en ligne sur le réseau local de deux universités étrangères dans le but de tester la performance de notre plate-forme d'ALAO par des apprenants dans un milieu réel.

4 Définition de l'architecture

Avant de commencer la phase de développement de la totalité de la plate-forme, nous avons commencé par l'étude des besoins ainsi que les insuffisances des autres plateformes d'ALAO. La figure ci-dessous montre les différents modules qui constituent notre plate-forme.

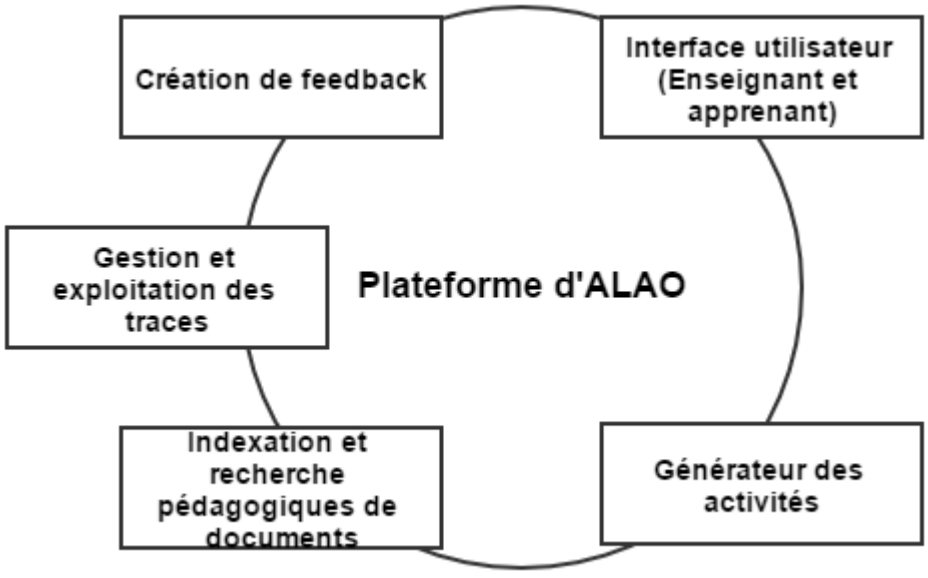

Figure 18 : Architecture générique de la plate-forme

4.1 Interface utilisateur : enseignant et apprenant

Un grand nombre d'études confirment que les apprenants aiment utiliser la technologie dans l'apprentissage des langues étrangères et qu'ils préfèrent utiliser la technologie par rapport aux méthodes et matériaux plus traditionnels (Golonka et al., 2014). Grâce à la technologie, les apprenants ont tendance à être plus engagés dans le processus d'apprentissage, et ont une attitude plus positive envers l'apprentissage. En particulier, les élèves perçoivent l'utilisation des ordinateurs comme une méthode d'apprentissage innovante et attractive. Pour cette raison, nous allons employer des ressources et outils TAL dans notre plate-forme afin de faciliter l'accès aux dernières technologies d'ALAO et les mettre à la disposition des enseignants et des apprenants. Sachant que lors de l'utilisation de la plate-forme, l'accès à l'interface par les utilisateurs devrait être simple et ne nécessite pas des connaissance en informatique.

Comme notre système est destiné à la fois à l'enseignant et à l'apprenant, il faut prévoir dans ce cas deux types d'interface adaptées aux besoins de chaque type d'utilisateur.

- Interface enseignant : Elle devrait permettre aux enseignants de gérer et d'exploiter des ressources textuelles à partir d'une interface unique avec des outils à la fois puissants et faciles à utiliser pour automatiser le travail nécessaire. De plus, le côté technique de la plate-forme devrait être transparent à l'utilisateur (enseignant) qui ne devrait se concentrer que sur l'aspect éducatif (Mars et al, 2014b).
- Interface apprenant : Cette interface devrait permettre aux apprenants de visualiser toutes les activités accessibles créées par les enseignants de la langue, travailler une ou plusieurs activités sur le système avec la possibilité de correction automatique ou l’envoyer directement à l’enseignant.

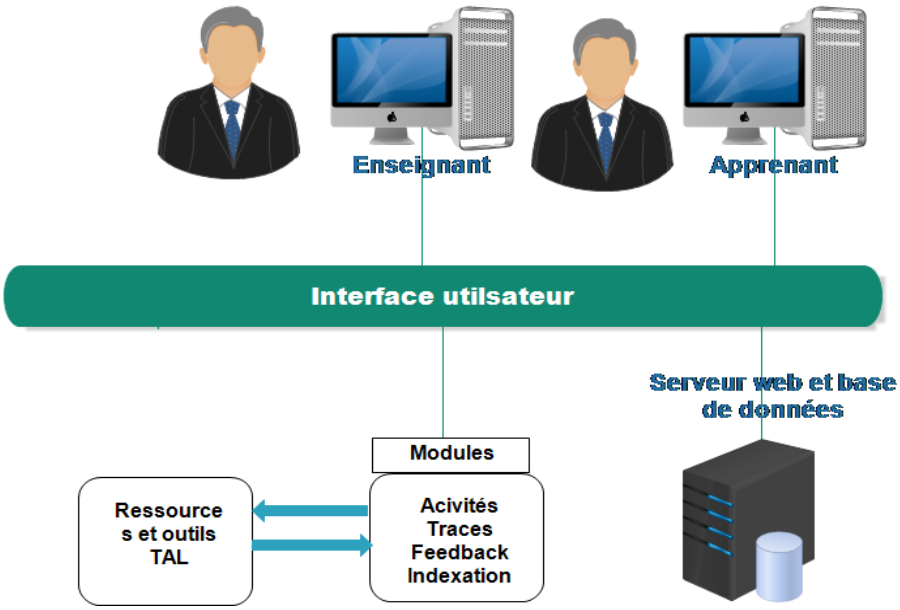

Figure 19 : architecture de l'interface utilisateur

4.2 Module de génération des activités

La génération des activités pédagogiques consiste à fournir à l'enseignant des outils lui permettant de réaliser cette tâche lui-même. Ces outils devraient permettre à l'enseignant la personnalisation des activités pédagogiques proposées à l'apprenant. De plus, le générateur doit assurer une interactivité entre l'enseignant et la plate-forme, ainsi, la manipulation de cet outil devrait être simple et ne nécessite pas des compétences en informatique.

4.3 Module feedback

Lors de l'interaction enseignant–apprenant dans un processus d'apprentissage de langues, le feedback occupe une place particulière. En effet, il fait partie de l'échange conversationnel, il est considéré comme un facteur indispensable à l'apprentissage. Par ailleurs, c'est une composante essentielle de la correction des erreurs des apprenants (Shintani et al, 2015). Le feedback est sans doute considéré comme un module indispensable aux environnements d'apprentissage interactifs, au point de former la partie essentiel d'appréciation de la qualité d'un système d'ALAO.

Dans cette optique et dans le but de rendre notre plate-forme dynamique et répondre aux besoins des apprenants et des enseignants, nous intégrons dans notre plate-forme un

module de feedback permet la génération automatique des corrections adéquates aux réponses des apprenants. Ce module devrait intégrer des techniques du TAL pour analyser la production des apprenants ou modéliser leur connaissance d'une langue étrangère, dans le but de fournir aux apprenants des commentaires et des conseils flexibles dans leur processus d'apprentissage. De plus, le module de feedback devrait fournir un renforcement et des renseignements devraient permettre aux apprenants de prendre en charge l'amélioration de leur production langagière.

4.4 Gestion et exploitation des traces

Lorsqu'on parle de la gestion et l'exploitation des traces, deux questions essentielles se posent. La première porte sur la manière d'acquisition pendant laquelle l'outil devrait enregistré les traces. La deuxième concerne le type des traces qu'on doit collecter et la manière dont on va l'exploiter. Bien qu'aujourd'hui la plupart des travaux de recherche dans le domaine d'ALAO se focalisent sur l'aspect d'acquisition tels que (Lunda et al, 2000), (Guéraud et al, 2004) et (Teutsch et al, 2004). Par contre, l'aspect d'exploitation et d'analyse des traces reste encore peu développé.

Dans notre cas, la gestion et l'exploitation des traces font partie des modules de notre plate-forme. selon (Renié, 2000) et (Davis et al, 2000), il est important lors de la mise en oeuvre d'une activité par les apprenants que le système devrait collecter un ensemble de traces de son travail. De plus, selon Renié, il est possible d'effectuer des analyses statistiques pour corréler les données obtenues dans les bases de données de trace avec des données concernant les caractéristiques individuelles de chaque apprenant. Donc, on peut dire que les traces obtenues après une session de travail qui présenterait non seulement des statistiques mais qui revêlerait également le parcours suivi par l'apprenant qui permettrait à l'enseignant de visualiser une meilleure image du processus d'apprentissage.

Par conséquent, le rôle de ce module dans nos travaux sera l'acquisition et l'exploitation des traces à l'aide d'un outil de traçage intégré à la plate-forme. Cet outil devrait permettre l'obtention des traces formalisées et donc exploitables par nos outils de feedback.

Dans la suite, nous présentons notre vision concernant la gestion de ces deux aspects : l'acquisition et l'exploitation des traces.

4.5 Acquisition des traces

L'acquisition de trace est considéré comme un point fort de l'ordinateur sauf qu'elle est insuffisamment utilisé dans l'apprentissage d'une langue seconde (Garrett, 1998)[11].

Lors de l'acquisition d'une trace, il y a deux défis principal à traiter la complétude et la pertinence :

- Complétude : dans la phase d'acquisition des traces, il faut collecter un maximum des données et fournir suffisamment de détails.
- Pertinence : les traces collectées devraient contenir toutes les informations nécessaires à la phase d'analyse et d'exploitation.
- Les données que nous paraient utiles à collecter sont :
- Les informations personnelles de chaque apprenant.
- Toutes les informations qui concernent les activités travaillées.
- Réponses de l'apprenant.
- Durée de la réponse.
- Toutes les informations qui concernent la génération des activités.

Les traces collectées par l'outil de traçage sont sauvegardées et structuré dans une base de données (figure 21).

[11] The computer's ability to collect data on what students do with technology-based language learning materials [...] gives us for the first time an instrument that will track the learning process rather than assigning a score to the outcome of that process in a test

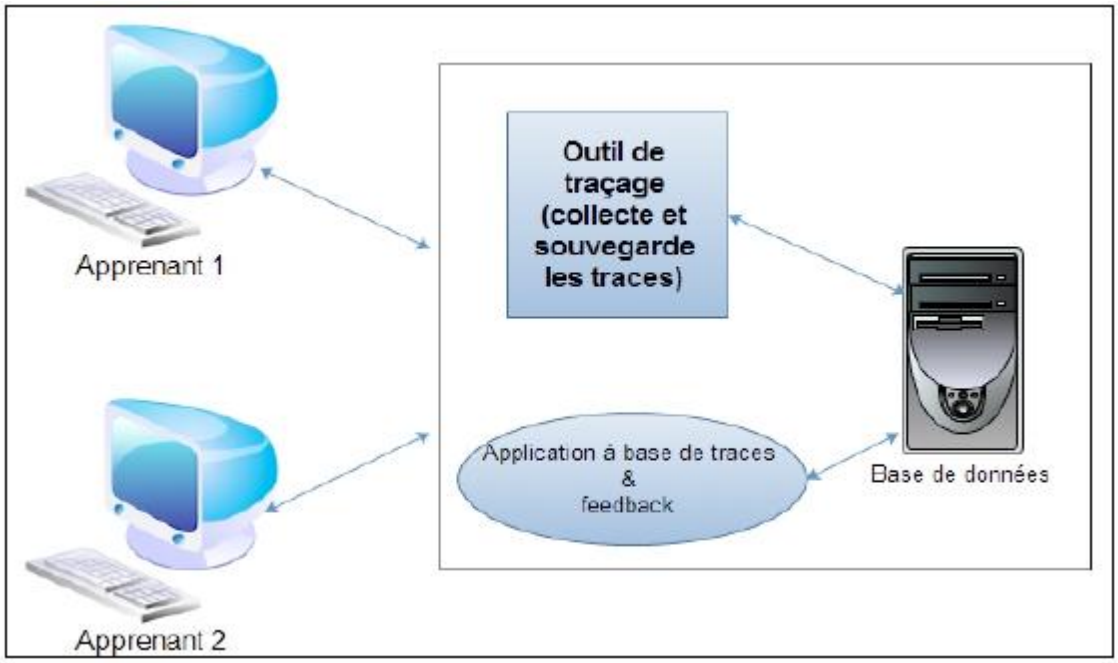

Figure 20 : Collecte et sauvegarde de traces.

4.6 Exploitation des traces

Une fois les traces sont collectées, elles seront accessibles pour la phase d'exploitation. Par contre, dans cette étape il faut qu'on précise notre but et qu'est ce qu'on doit faire exactement avec les traces. bien qu'il existe une méthode d'exploitation simple qui correspond à des accès en lecture et en écriture, il existe aussi d'autres méthodes complexes comme, par exemple, le filtrage, l'agrégation de données ou le calcul de statistiques.

Dans notre cas, l'objectif principal de l'outil d'exploitation est la représentation textuelle des informations et non l'extraction de connaissances d'un niveau sémantique plus élevé. Donc, l'outil devrait permettre l'affichage des informations liées aux réponses des apprenants et l'état d'avancement de chaque apprenant.

4.7 Indexation et recherche pédagogiques de documents

Le but de ce module est de mettre en place une base de textes avec des documents indexés pédagogiquement pour l'enseignement des langues. On procède à une classification automatique des textes en se basant sur le contenu linguistique de chaque texte traité. Cette base de textes indexés permet, grâce à son interface, (Loiseau 2009) (Loiseau 2005) aux enseignants de faire des requêtes basées sur des critères compatibles avec l'enseignement des langues. Cette technique devrait permettre aux enseignants de rechercher des documents

stocker dans les bases de données en utilisant des critères didactiquement pertinents un accès rapide aux ressources textuelles stocker dans les bases de données de la plate-forme.

5 Conclusion

Nous avons présenté dans ce chapitre la problématique de cette thèse et les différentes étapes qui nous permettions d'aboutir à notre solution de développement d'une plate-forme d'ALAO. Ensuite, nous avons abordé le travail de définition et de conception que nous avons entrepris sur l'architecture de la plate-forme qui permettra aux enseignants de la langue de créer des activités dans leur propre zone d'intérêt pédagogique. Enfin, nous avons présenté les modules essentiels au développement de notre plate-forme.

Chapitre 6 : Reconnaissance multilingue de l'écriture manuscrite

1 Introduction

Dans ce chapitre, nous décrivons la réalisation d'un système de reconnaissance de l'écriture manuscrite pour la langue française et la langue arabe.

La reconnaissance de l'écriture manuscrite est une technologie innovante, elle est capable de traiter l'encre numérique pour la convertir en texte digital. Cette encre numérique est obtenue lorsque nous écrivons avec un dispositif de pointage (souris, stylet…) ou directement avec notre index sur une surface tactile.

Nous comptons utiliser la reconnaissance de l'écriture pour la création des activités pédagogiques dans le but d'apprendre l'écriture. De plus, cette technologie devrait être capable de s'adapter à l'écriture de chacun, sans besoin d'apprentissage et quel que soit son style d'écriture (caractères isolés, caractères d'imprimerie ou écriture cursive).

Comme notre plate-forme supporte les deux langues (arabe et français), nous avons décidé de réaliser un système de reconnaissance d'écriture pour celles-ci.

2 Reconnaissance de l'écriture manuscrite du français

Dans un processus de reconnaissance de l'écriture, il y a généralement trois grandes étapes : la reconnaissance des caractères, la reconnaissance des mots et la reconnaissance des phrases. En général, un système de reconnaissance des mots se base sur des modèles construits lors de l'apprentissage d'un système de reconnaissance de caractères. De même, un

système de reconnaissance de phrases se base sur des modèles construits lors de l'apprentissage d'un système de reconnaissance de mots. Par conséquent, les phases les plus importantes dans un processus de reconnaissance sont la phase de reconnaissance de caractères et la phase de reconnaissance de mots.

Par la suite, nous présentons les différentes étapes nécessaires pour développer un système de reconnaissance complet.

2.1 Reconnaissance automatique des caractères

Un système de reconnaissance de caractères manuscrits prend en entrée une encre provenant d'un équipement en ligne (figure 22). Pour que cette encre soit reconnue en caractères digitaux, plusieurs traitements sont effectués par le système. Parmi ces composants, on cite le module qui se charge du prétraitement et de la normalisation. Une fois que l'encre est prétraitée, un autre module extrait les caractéristiques nécessaires. Ces dernières sont utilisées après l'extraction par le module de classification basé sur le réseau de neurones pour donner une hypothèse du caractère.

```
.KEYWORD .TRANS_X
.KEYWORD .TRANS_Y
.TRANS_X 177
.TRANS_Y 309
.COORD X Y
.SEGMENT ? ? ? "Faute de mobilité et d'efficacité dans leur jeu."
.SEGMENT CHARACTER ? ?
.PEN_DOWN
402 29
398 25
400 20
404 14
412 6
412 6
419 21
421 104
424 127
425 127
425 127
422 127
417 125
.PEN_UP
.PEN_DOWN
355 59
355 57
358 51
385 37
408 26
491 1
490 1
488 1
484 0
.PEN_UP
.PEN DOWN
```

Figure 22 : Extrait d'un fichier d'encre montrant le format d'un signal de l'écriture en ligne obtenue à partir d'un stylo numérique.

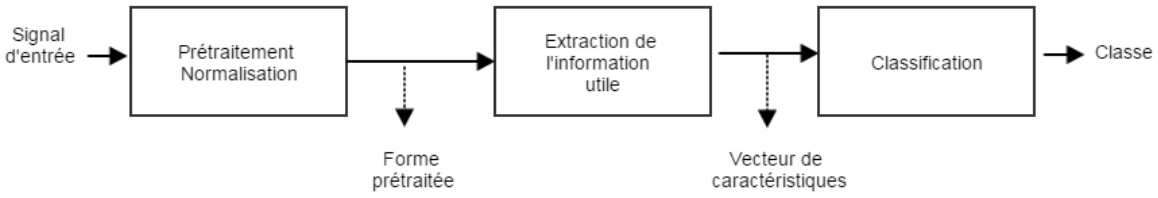

Figure 23 : Processus de la reconnaissance automatique de caractères

Notons que notre système contient ces trois modules (prétraitement, extraction et classification) que nous allons détailler par la suite. Au départ, le système reçoit en entrée un fichier d'encre qui est sous la forme des coordonnées des points [x, y] au format Unipen. Puis, l'encre doit être ré-échantillonnée en points fixes avant de la normaliser. Ensuite, à partir de l'encre normalisée, le système doit extraire pour chaque point un vecteur de sept caractéristiques (nous détaillons ces caractéristiques dans la section 1.1.3). Enfin, le système présentera ces points à un réseau de neurones à délai temporel qui fournira à la fin un vecteur de probabilités associé à chaque classe.

2.1.1 Prétraitement et normalisation

Le prétraitement de l'encre est une étape nécessaire pour améliorer sa qualité qui peut conduire à une meilleure représentation de fonction et une meilleure reconnaissance. Certains bruits, tels que les points répétés ou les imperfections dans le processus de numérisation doivent être éliminés. La procédure de ré-échantillonnage est obligatoire pour améliorer la qualité de l'encre (Kavalieratou et al, 2002) comme montre la figure ci-dessous.

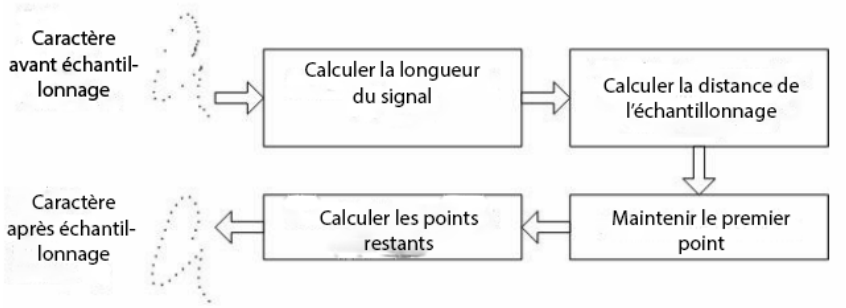

Figure 24 : Ré échantillonnage de l'encre en ligne.

Les autres procédures de prétraitement que nous avons implémentées dans notre système sont :

- La suppression de points doubles : cette étape permet d'enlever les points en double, c'est-à-dire les points qui ont les mêmes coordonnées, dans ce cas un seul point d'entre eux est conservé.

- L'interpolation : une étape d'interpolation linéaire est obligatoire dans le but d'ajouter les points manquants dus à la variation de la vitesse d'écriture (Huang et al, 2007).
- Le lissage : afin d'éliminer les imperfections matérielles et les tremblements des scripteurs, chaque point est substitué avec la mesure du moyen pondérée de ses points voisins (Kavalieratou et al, 2002).

La normalisation de l'encre est une procédure standard dans la plupart des systèmes de reconnaissance. Dans notre cas, la normalisation faite sur l'encre des caractères est principalement destinée à normaliser l'encre initiale de manière à rendre invariante à la translation, à la taille du caractère et au style de l'écriture. La normalisation de l'encre d'un caractère est plus simple que la normalisation d'un mot qui nécessite la détection de toutes les lignes de référence, la ligne basse, la ligne de base, la ligne de haut.

2.1.2 Strokes retardés

Le problème principal des strokes retardés est qu'ils conduisent à la dispersion des différents composants ou segments de caractères qui ne correspondent pas à la séquence attendue du modèle de RN. Afin de corriger ce problème, nous allons créer un module qui ajoute un stroke spécial de connexion afin d'attacher les strokes retardés aux mots auxquels ils appartiennent (Sternby et al, 2009). Cela veut dire que notre système traite dynamiquement les variations liées à l'attachement de diacritiques en utilisant une technique de discrimination entre les différentes hypothèses de mots qui ont les mêmes formes de base, mais un attachement différent des diacritiques.

2.1.3 Extraction de caractéristiques

Pour l'entraînement d'un système de reconnaissance, l'encre résultante après la phase du prétraitement et la normalisation est utilisée pour l'extraction des caractéristiques afin de l'utiliser dans l'entraînement des modèles ou dans la reconnaissance. Dans notre cas, sept valeurs de caractéristiques ont été extraites pour chaque point résultant, les valeurs de caractéristiques pour chaque point x (n), y (n) sont les suivants :

i) Normalisé x (n) entre -1 et 1.

ii) Normalisé y (n) entre -1 et 1.

iii) Cosinus de l'angle composée de la ligne entre le point x (n + 1), y (n + 1) et le point x (n-1), y (n-1) et l'axe x.

iv) Sinus de l'angle composée de la ligne entre le point x (n + 1), y (n + 1) et le point x (n-1), y (n-1) et l'axe x.

v) Cosinus de l'angle de courbure entre le point x (n + 2), y (n + 2) et le point x (n-2), y (n-2) à x (n), y (n).

vi) Sinus de l'angle de courbure entre le point x (n + 2), y (n + 2) et le point x (n-2), y (n-2) à x (n), y (n).

vii) Fixé la valeur binaire à 1 lorsque le stylo est levé, ou -1 lorsque le stylo est posé.

Les caractéristiques (iii) et (iv) donnent l'information de direction et les caractéristiques (v) et (vi) l'information de courbure. La figure ci-dessous montre de façon plus détaillée, les quatre caractéristiques liées au mode de courbures dans (iii), (iv), (v) et (vi). Pour la reconnaissance, les valeurs de caractéristiques pour un seul caractère sont utilisées pour sa reconnaissance. Pour ce faire, nous avons développé un programme en langage c pour l'implémentation de ces règles d'extraction des caractéristiques, ainsi que l'implémentation des formules mathématiques.

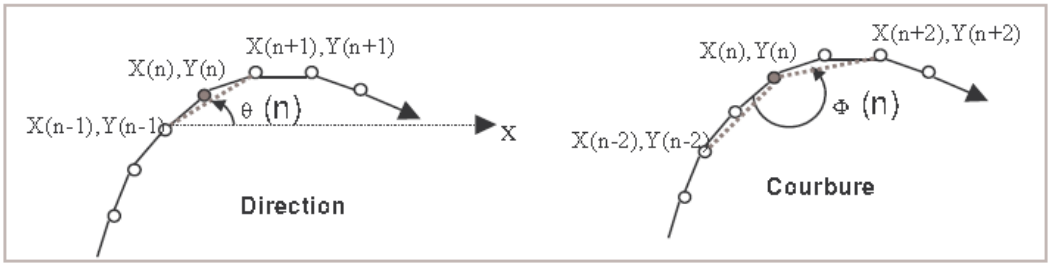

Figure 21 : Caractéristiques de direction et caractéristiques de courbure.

2.1.4 Entraînement et reconnaissance

L'entraînement des réseaux de neurones a été fait en utilisant tous les caractères récupérés à partir de 170 formulaires remplis lors de la phase de la collecte de l'encre. Dans cette partie, nous allons présenter l'implémentation que nous avons réalisée ainsi que les différents algorithmes associés d'apprentissage et de reconnaissance.

2.1.5 Implémentation du système

Dans la phase d'apprentissage, nous avons retenu les deux parties principales : extraction et classification. Le processus d'extraction se fait aux couches basses; il implémente les convolutions successives qui conduisent à la transformation progressive des

caractéristiques en grandeurs significatives par rapport au problème. La réalisation du second processus se fait par un perceptron multicouche classique. Ce dernier reçoit en entrée toutes les sorties qui viennent de la partie extraction des TDNN. De plus, il est possible de paramétrer ces deux processus.

Lors de l'implémentation de notre réseau de neurones, nous avons précisé ses caractéristiques pour les deux parties, TDNN et PMC.

Partie TDNN :

- Le nombre de couches.
- Le nombre de neurones pour chaque couche.
- Le nombre de neurones pour chaque couche selon la direction (propagation ou rétropropagation).
- La taille de la fenêtre temporelle (vue par chaque couche).
- Le délai temporel entre les fenêtres.

De plus, dans notre algorithme, nous avons identifié chaque neurone de la partie TDNN par son emplacement temporel t, et sa couche l. Par la suite, chaque neurone est défini par :

- Une sortie x[l][f][t].
- Une matrice remplie par les poids des entrées, w[l-1][f][f[l-1]][t].
- Le vecteur pour les poids des biais, w_biais[l-1][f].
- La somme des entrées, v[l][f][t].
- L'indice d'erreur pour la rétro-propagation du gradient, y[l][f][t].
- Le gradient, delta[l-l][f][f[l-l]][t].
- Une fonction permettant l'activation du neurone (de type tangent hyperbolique).

Partie PMC :

- Le nombre de couches.
- Le nombre de neurones pour chaque couche.

Chaque neurone de la partie PMC est défini par sa caractéristique f, son emplacement temporel t, et sa couche l. Par la suite, chaque neurone est défini par :

- Une sortie x[l][t].
- Une matrice remplie par les poids des entrées, w[l-1][t][[l-1]].
- Le vecteur pour les poids des biais, w_biais[l-1].
- La somme des entrées, v[l][t].
- L'indice d'erreur pour la rétro-propagation du gradient, y[l][t].

- Le gradient, delta [l-1][t][t[l-1]].
- Une fonction pour les couches cachées du type tangente hyperbolique.
- Une fonction permettant l'activation pour la couche de sortie de type Softmax.

Dans la partie de définition de PMC, la dimension des caractéristiques f a disparu par rapport aux couches de la partie extraction.

Par conséquent, notre réseau fonctionne en trois étapes. D'abord, la première couche du réseau capte les caractéristiques du signal d'entrée. Puis, une ou plusieurs couches cachées appartenant au réseau de neurones (étape de l'extraction) transforment une séquence de vecteurs caractéristiques en une séquence de vecteurs caractéristiques d'ordres supérieurs. Ensuite, un neurone donné capte la caractéristique topologique locale qui décrit la trajectoire du stylet. Chaque neurone a un champ de vision réduit à une fenêtre temporelle limitée. À cause des contraintes des poids partagés, le même neurone doit être dupliqué dans la direction temps afin de détecter si la même caractéristique à différentes places existe ou non tout au long de la trajectoire du signal. Enfin, en utilisant plusieurs neurones dans toutes les positions temporelles, le réseau de neurones doit détecter les caractéristiques différentes. Cependant, chaque neurone doit produire en sortie un nouveau vecteur caractéristique pour la couche supérieure.

2.1.6 Algorithme d'apprentissage et reconnaissance

La première partie dans notre processus de reconnaissance est la propagation de l'échantillon dans le réseau de neurones pendant la phase d'extraction et la phase de classification. L'algorithme ci-dessus décrit notre implémentation pour cette étape.

Phase d'extraction (TDNN) :

```
1. Acquisition des données en utilisant le TDNN
Scan temporel des neurones de la 1ère couche du réseau
Scan sur tous les neurones qui appartiennent à la couche Acquisition
des données
 2. Extraction de caractéristiques (couches cachées et couche de la
    sortie du TDNN) :
    Balayage sur toutes les couches l du réseau
     Balayage temporel des neurones pour chaque couche t
      Balayage basé sur les caractéristiques de la couche f
           Initialisation des sommes pondérées v de tous les neurones
             Balayage effectué sur tous les neurones de la fenêtre associée
                  Calcul des sommes pondérées v associé à chaque neurone
                   Calcul de l'activation x de chaque neurone (tangh)
             Décalage de la fenêtre selon le délai associé
```

Phase classifieur (PMC) :

```
3. Adaptation du réseau au PMC
```

```
4.Classifieur (Couches cachées du PMC)
 Balayage de toutes les couches du PMC
    Balayage temporel de tous les neurones de la couche t
       Initialisation des sommes pondérées v de tous les neurones
        Balayage de chaque neurone de la couche précédente
           Calcul des sommes pondérées v de tous les neurones
           Calcul des sorties x de tous les neurones (tanh)
5. Résultat : dernière couche du PMC
Balayage temporel sur tous les neurones
    Calcul des sommes pondérées v de tous les neurones
    Calcul des activations x de tous les neurones(Softmax)
```

Selon (Hérault et al, 1994) le choix de la méthode d'apprentissage présente un des points importants lors de la mise en œuvre d'une solution basée sur l'approche de réseau de neurones. Idéalement, la méthode choisie doit permettre au réseau de converger rapidement vers le minimum global calculé par la fonction de coût. Selon la complexité des problèmes d'optimisation, plusieurs méthodes peuvent être sélectionnées. Dans notre système, nous avons utilisé une méthode du premier ordre basée sur l'algorithme classique de la rétro-propagation du gradient, dite du gradient stochastique, qui est capable de converger plus rapidement que la méthode du gradient global (Sarle, 1997).

Après le choix de la méthode, nous avons développé cet algorithme pour l'apprentissage de TDNN.

```
Définition et allocation du TDNN
Initialisation aléatoire (contrôlée) des poids
Construction d'une base d'apprentissage normalisée
Pour chaque exemple e appartient à la base
    d'apprentissage
    Ré-échantillonnage de l'exemple e
    Normalisation de l'exemple e
    Extraction et sauvegarde des toutes les caractéristiques de
    l'exemple e
    Sauvegarde du label de l'exemple e
Tant que la condition d'arrêt n'est pas satisfaite
     Pour chaque exemple e appartient à la base
    d'apprentissage
       Propagation de l'exemple e
                Calcul de l'erreur locale
                Si le critère d'erreur locale de l'exemple e non
                satisfait Alors Rétro-propagation de l'erreur
  Vérification de la condition
  d'arrêt
  Fin tant que

     //Calcul du critère
     Pour chaque exemple e appartient à la base
    d'apprentissage
    Propagation de l'exemple e
    Calcul de l'erreur locale
    Calcul de l'erreur cumulée
```

Pour calculer les erreurs, nous utilisons la fonction suivante :

$$E_{CE} = -\sum_{j}[d_j\, log(y_j) + (1 - d_j) \log (1 - y_j)]$$

Avec :

y_j est la sortie de l'exemple j.

d_j est l'entrée de l'apprentissage pour l'exemple j.

L'algorithme d'apprentissage ci-dessus est généralisable pour tous les réseaux de neurones de type perceptron. Par contre, la seule différence réside dans les fonctions de propagation et de rétro-propagation, car il faut prendre en compte la taille de la fenêtre observée par les neurones de la couche intermédiaire (la partie extraction des caractéristiques de réseaux de neurones).

2.2 Reconnaissance automatique de mots et de phrases

Dans le domaine de la reconnaissance, l'adaptation d'un système de reconnaissance du caractère en un système de reconnaissance du mot n'est pas évidente. Dans ce cas, le système devient rapidement très complexe, à la fois dans sa mode d'apprentissage et dans son architecture (LeCun et al, 2001), (Tay, 2002). De plus, le nombre de paramètres peut devenir plus important et nécessite beaucoup de temps dans l'exécution du processus, ou bien encore la phase d'apprentissage devient très contraignante (avec des procédures en plusieurs étapes); éventuellement, il faut un apprentissage du système au niveau lettre ou graphème, puis un deuxième apprentissage au niveau mot (Jaeger, 2000).

Pour la reconnaissance de mots, nous avons gardé l'approche du réseau de neurones, car elle a montré que ce réseau était très adéquat et efficace par rapport à l'objectif visé; de plus, il apporte une bonne performance au niveau de taux de reconnaissance avec une architecture de complexité raisonnable.

2.2.1 Prétraitement et normalisation

Généralement, le système de reconnaissance de l'écriture prend en entrée une encre en ligne. Elle a besoin d'être prétraitée pour éliminer les bruits qui peuvent affecter les performances de la reconnaissance. Ces bruits sont dûs à la variété de l'écriture manuscrite (voir figure ci-dessous) qui dépend de différentes sources telles que l'identité des scripteurs ou auteurs (âge, état physique), la posture et l'environnement lors de l'écriture (à un bureau, en mouvement), mais aussi l'outil ou le dispositif de l'écriture (gros stylo digital, petit stylet, etc.)

et le média utilisé (papier, tableau interactif, écran). Par conséquent, il est préférable d'éliminer ces bruits le plus tôt possible pour faciliter la tâche de reconnaissance à notre système. Cette étape représente le but principal du processus de prétraitement et normalisation du signal d'écriture (l'encre) dans notre système.

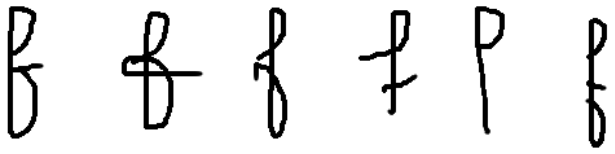

Figure 22 : variété de l'écriture manuscrite de la lettre « f ».

L'étape du prétraitement lors de l'entraînement du système est faite pour toute la base de données de mots, par conséquent les mots seront déjà prétraités à l'entrée du module de l'entraînement. Dans un système de reconnaissance déjà entraîné, le prétraitement d'un mot en entrée se fait au sein du système, avant la phase de la reconnaissance. De même, le prétraitement implique la phase de réduction du bruit et la phase de la normalisation. La réduction du bruit peut être obtenue en limitant la bande passante de la fréquence des données à l'aide de filtration, où les points de rebroussement sont traités comme des points limites pour éviter le lissage des caractéristiques importantes de la forme. L'algorithme de détection de rebroussement capture les rebroussements dominants, en ignorant les petites causées par le bruit. Dans notre système, nous avons effectué une correction de rotation de mots en nous référant à la ligne de base et la normalisation de la taille. À la fin de la phase de normalisation, le système applique l'échantillonnage spatial sur le signal d'écriture.

Cette étape est la même que celle effectuée lors de la normalisation du caractère. De plus, nous avons appliqué le même principe pour les strokes retardés que celui utilisé pour les caractères.

2.2.2 Lignes de références

Dans un processus de reconnaissance des mots, les lignes de référence apportent des informations importantes pour les systèmes. Elles jouent un rôle important dans la normalisation de la taille de l'écriture et l'extraction des caractéristiques géométriques liées à l'orientation et à la position des lettres dans le mot.

Selon (Bengio, 1994), il faut une paramétrisation des lignes proposées pour déterminer les lignes de références. Pour cela nous avons proposé une modélisation simplifiée des lignes en considérants que les lignes de références sont droite ; cette méthode nous permet d'éviter

une instabilité du modèle due à la prise en compte de la courbure de certains mots qui ne sont pas toujours stables.

Pendant la phase de détection des lignes de référence, nous relevons d'abord les minima locaux ainsi que les maxima de l'écriture en effectuant une analyse simple de différentes positions et changements de la direction verticale.

L'algorithme suivant montre notre modélisation du problème de la détermination des lignes de références.

```
Détection des points minima et maxima
Calcul de l'écart type de la distribution en Y des extrema
Initialisation des différents paramètres à estimer : vecteur de translation
et la pente
          Tant que la condition d'arrêt non respectée :

          Analyse du changement global des valeurs yO

   Fin Tant que

 //Affectation des minima
 Pour chaque minima
        Calculer la probabilité d'observation de toutes les lignes de
        référence
        Rechercher la probabilité d'observation maximale
        Comparer les probabilités d'observation maximale avec celle de bruit
        afin d'affecter ou non le minima
        Affectation de minima
 //Affectation des maxima
 Pour chaque maxima
        Calculer la probabilité d'observation des toutes les lignes de
        référence
        Rechercher la probabilité d'observation maximale
        Comparer les probabilités d'observation maximale avec celle des
        bruits afin d'affecter ou non le maxima
        Affectation ou non
 Calculer la pente
 Calcul des ordonnées
 Vérification de la condition d'arrêt
 Fin Tant que
```

2.2.3 Segmentation

Afin de générer des candidats pour les caractères, le mot d'entrée est segmenté en graphèmes. La figure ci-dessous montre la segmentation maximale du mot « un ».

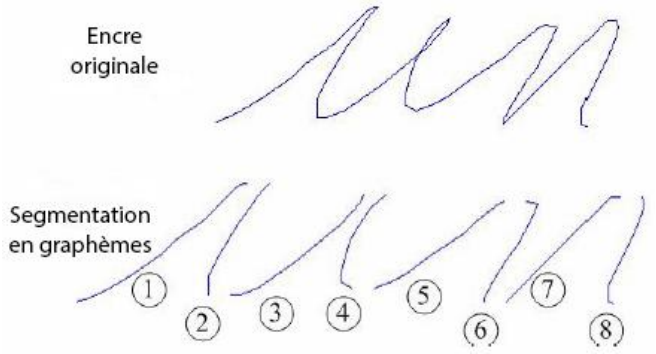

Figure 23 : segmentation maximale du mot "un" selon les coordonnées y (maximale et minimale) du mot.

Les segmentations maximales sont effectuées en se basant sur les coordonnées maximales et minimales selon l'axe (y) du mot. Cela est considéré comme une méthode de base très simple. Dans la figure précédente, le mot "un" est divisé en huit tranches comme indiqué ci-dessus. Pour un mot plus long, il y aura plus de tranches. Les tranches sont combinées pour former une hypothèse de caractère. Le nombre total d'hypothèses affecte la complexité de notre processus d'entraînement ainsi que la précision de la reconnaissance. Ce nombre dépend du nombre total de tranches qui devrait être inclus dans l'hypothèse du caractère. Afin de créer une hypothèse correcte de caractère, le nombre de tranches qui compose le caractère est très important.

Le nombre total d'hypothèses peut être calculé par la formule suivante :

```
if (num < max)
tot=(num-min+1)*(num-min+2)/2;
else
   tot=(num-max)*(max-min+1)+(max-min+1)*(max-min+2)/2;
```

Équation 2 : Détermination du nombre des hypothèses.

Avec :

- tot est le nombre total d'hypothèse généré.
- num est le nombre de tranches.
- min et max sont les tranches minimales et maximales dans une hypothèse respectivement.

La figure ci-dessous montre un exemple de segmentation et génération d'hypothèse.

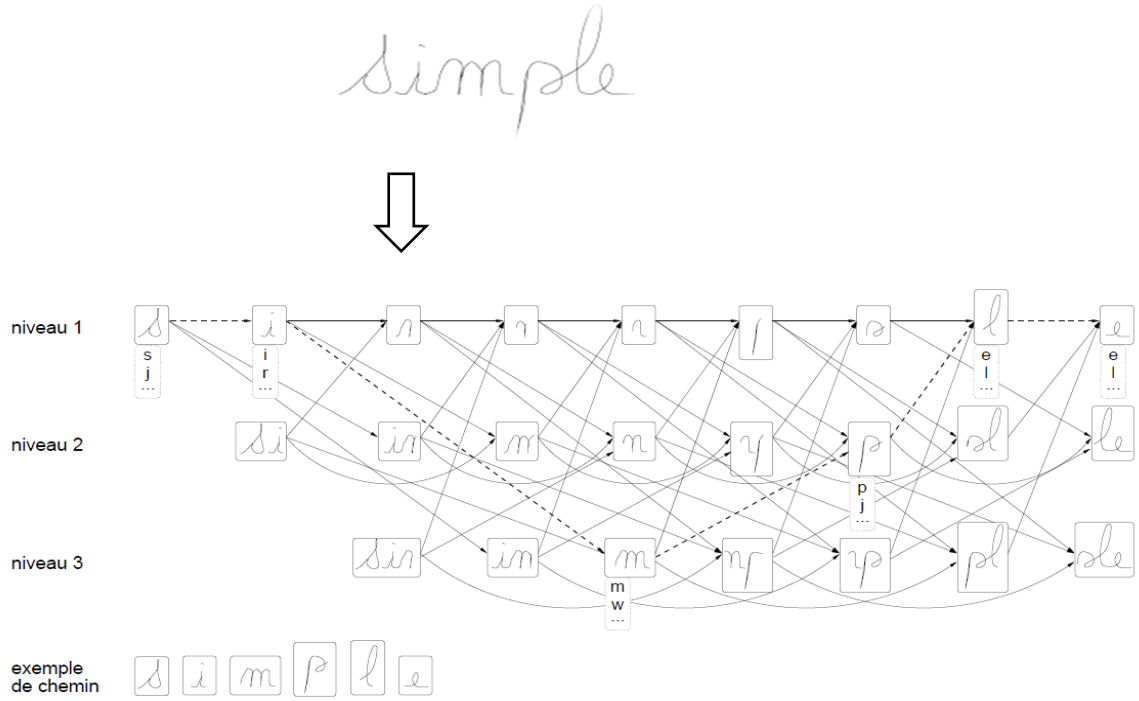

Figure 24 : Génération des hypothèses de caractère : exemple de la segmentation et la génération des hypothèses.

Le nombre maximal de tranches pour un caractère est choisi, on se basant sur les statistiques de l'ensemble de données d'entraînement. Nous avons constaté que la valeur maximale 7 et la valeur minimale 1 sont adaptées, car une lettre minuscule contient en moyenne cinq tranches. De manière à couvrir tous les caractères, nous avons pris en compte l'irrégularité au début et lors de l'écriture, et nous avons choisi la valeur maximale 7. Cela a également été vérifié expérimentalement.

Pour former un mot, seules les hypothèses non-chevauchantes apparentées sont utilisées. La tâche de choisir les hypothèses correctes qui signifient « caractère bien segmenté » réside dans l'algorithme de programmation dynamique dans le réseau des neurones. La figure ci-dessous montre un exemple de la meilleure segmentation qui a conduit à la meilleure reconnaissance en utilisant le réseau des neurones.

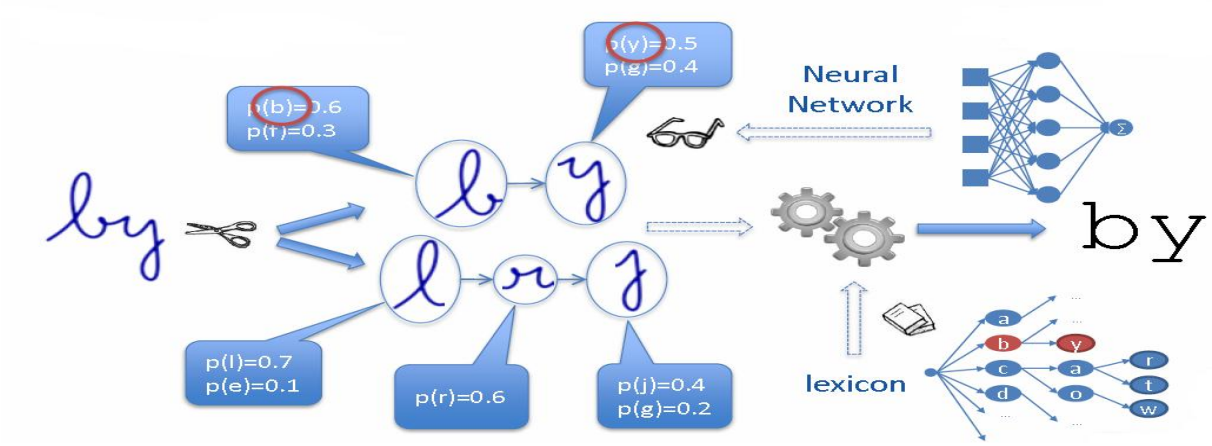

Figure 25 : Processus de reconnaissance d'un mot

Dans cet exemple, la segmentation de l'encre conduit à deux candidats possibles « lrj » ou « by ». L'utilisation de réseau de neurones et le lexique ont aidé à privilégier la solution « by » en s'appuyant sur les ressources de la langue.

2.2.4 Extraction de caractéristiques

Actuellement, notre système est entraîné pour la reconnaissance de caractères; les caractéristiques extraites par notre système sont utilisées pour la détermination des hypothèses de caractères qui ont été formés en joignant la totalité des tranches segmentées. Avant l'extraction des caractéristiques, le ré-échantillonnage de l'encre se fait en se basant sur l'hypothèse du caractère pour normaliser le nombre de points dans l'encre de caractère. Par conséquent, l'objectif de cette étape est la capture des informations les plus appropriées et la discrimination de l'objet à reconnaître.

Dans la phase d'entraînement, le système extrait sept caractéristiques par point, comme dans la phase d'entraînement du système sur les caractères isolés. Cependant, chaque caractère est maintenant découpé dans le mot, par conséquent, la coordonnée x sera différente de coordonnée x des caractères isolés, car les coordonnées d'origine de x subissent une translation par rapport au début du caractère. Cela a pour effet de régler la coordonnée x du caractère à partir d'un point commun qui commence à partir de zéro. La nouvelle coordonnée x pour chaque point dans l'hypothèse du caractère sera comme suit :

$$x_{offset} = x_0$$

$$x_n = x_n - x_{offset} \quad \forall n$$

Les six autres caractéristiques sont essentiellement les mêmes que celles dans l'entraînement des caractères isolés, qui sont les coordonnées y, les deux caractéristiques de direction, les deux caractéristiques de courbure et la / les information(s) de stylo (posé ou bien levé). À la fin de l'extraction des caractéristiques, le système fournit une séquence de vecteurs contenant toutes les caractéristiques requises, c'est-à-dire sept fonctionnalités pour tous les points. Les séquences du vecteur sont ensuite fournies en tant qu'entrées au système de la reconnaissance de l'écriture manuscrite.

2.2.5 Apprentissage et reconnaissance

Pour la reconnaissance de mots, nous utilisons principalement dans notre système le réseau de neurones à convolution, essentiellement le TDNN.

Après la phase de la segmentation et de l'extraction des caractéristiques, on obtient un treillis dans la sortie du TDNN. Puis, à partir de cette sortie, nous calculons la vraisemblance de chaque mot du dictionnaire. Pour cela, nous avons appliqué le même algorithme d'apprentissage utilisé précédemment pour l'apprentissage des caractères.

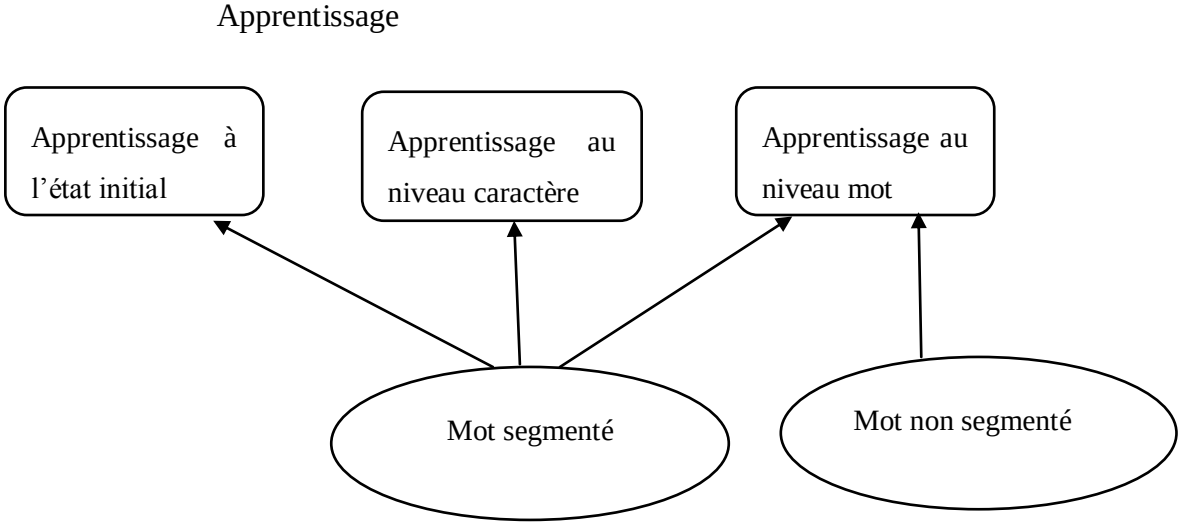

Figure 26 : apprentissage du réseau de neurones sur les mots

Dans un TDNN, les mots sont représentés comme une séquence de caractères ou chaque caractère est modélisé par un ou plusieurs états. Par conséquent, le TDNN peut être considéré comme un dispositif de reconnaissance hybride qui combine les caractéristiques des réseaux de neurones. Chaque neurone dans la couche d'entrée représente un état d'un caractère (au début, au milieu, à la fin ou isolé). Généralement, le score d'un caractère est calculé en trouvant un chemin d'alignement optimal grâce à ses états et en additionnant les activations dans cette voie. De même, le score d'un mot est calculé en trouvant un chemin

d'alignement optimal à travers les états de caractères composant le mot. Par conséquent, le score final est à nouveau obtenu en additionnant toutes les activations dans cette voie.

a. Apprentissage

Un TDNN est formé en trois étapes avec le principe de rétro-propagation. La première et la seconde étape d'apprentissage fonctionnent dans un mode d'alignement forcé, pendant laquelle le TDNN est entraînée avec des données d'entraînement segmentées à la main, c'est-à-dire, les limites de caractères sont connues pour ces mots. Dans la première étape (la formation au niveau du neurone), on suppose que le chemin de reconnaissance correcte dans le réseau reste la même durée dans chaque état du mot. Les états, le long de ce chemin, constituent les données d'apprentissage pour la procédure de rétro-propagation à partir de la couche d'entrée de TDNN. Après quelques itérations, notre supposition (le chemin de la reconnaissance reste pour la même durée dans chaque état) est abandonnée, car maintenant, on doit calculer le chemin de la reconnaissance réelle à travers un modèle de caractère, qui marque le début de l'étape 2 (apprentissage au niveau du caractère). Puis, après quelques itérations, la troisième étape commence par le remplacement de l'alignement forcé dans les étapes 1 et 2 avec l'alignement libre qui lance l'apprentissage au niveau du mot. Cela présente l'avantage que l'apprentissage peut maintenant être effectué sur les données non segmentées. Ainsi, seule une petite partie des données d'apprentissage doit être segmentée manuellement au niveau caractère. Lorsque le réseau a appris avec succès les limites des caractères sur la base d'apprentissage segmentée, l'alignement forcé est remplacé par un alignement libre et l'apprentissage peut être effectué sur de grandes bases de données contenant des données d'apprentissage non segmenté.

b. Utilisation de dictionnaire pour la reconnaissance des mots et des phrases

Notre système de reconnaissance d'écriture en ligne est basé sur un dictionnaire des mots. En général, la taille des dictionnaires influence la performance et le temps de réponse du système.

Pour la recherche des mots dans le dictionnaire, nous utilisons une approche basée sur les arbres (Manke et al, 1996). Elle combine une représentation arborescente du dictionnaire avec des techniques de recherche efficaces pour réduire le temps de recherche en gardant la même performance de système de la reconnaissance. Pratiquement, le principe est simple, d'abord, nous construisons un arbre de recherche pour chaque caractère, ce dernier représente tous les mots commençant par ce caractère spécifique. Puis, lors de la recherche d'un mot, nous n'activons que les racines des arbres, alors que tous les autres nœuds sont inactifs. Enfin,

nous construisons deux listes dont les éléments pointent vers les nœuds actifs : les points de la première liste pointent sur les nœuds actifs dans la phase actuelle et la deuxième liste contient des points qui pointent vers les nœuds qui devraient être actifs dans la phase suivante. Enfin, en continuant la dernière étape itérativement, nous arrivons assez rapidement au mot recherché.

L'approche de l'arbre est facilement applicable pour la recherche des mots afin de reconnaître des phrases entières. En premier lieu, nous insérons un nœud supplémentaire qui représente l'espace blanc entre deux mots dans une phrase. L'extrémité de chaque nœud de l'arbre est reliée à ce nœud (qui représente l'espace blanc), qui à son tour est connecté à chaque nœud racine. En second lieu, nous avons identifié que le passage par le nœud d'espace doit marquer le début d'un nouveau mot. Enfin, nous appliquons le même principe de liste qui est utilisé pour les mots, nous obtenons le bon chemin de notre phrase.

Le dictionnaire utilisé par notre système de reconnaissance contient environ 700 000 mots français qui sont suffisants pour le développement du système (les détails de la construction de ce dictionnaire sont présentés dans le chapitre suivant).

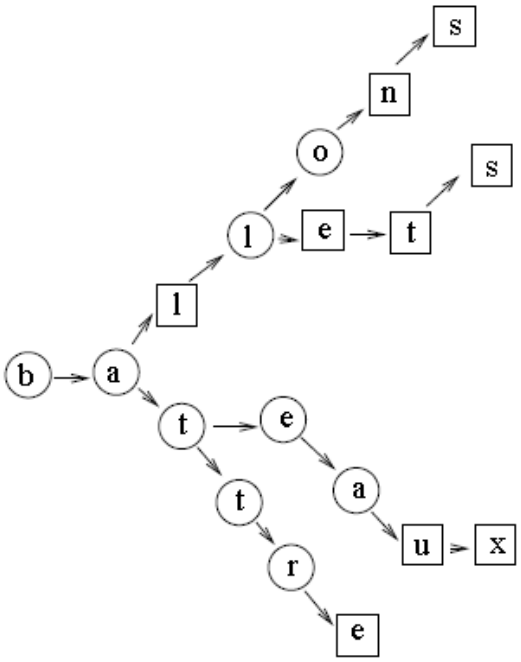

Figure 27 : Architecture d'un dictionnaire en arbre

c. Apprentissage de phrases

Le même mécanisme appliqué pour l'apprentissage de mots fonctionne également pour le niveau suivant, les phrases entières. Nous avons entraîné avec un TDNN des phrases non segmentées, sachant que le TDNN a été déjà entraîné avec des mots; L' algorithme trouve

la meilleure segmentation de toutes les phrases non segmentées en fonction des informations du réseau neuronal déjà appris au niveau du mot, par conséquent la reconnaissance automatique en ligne des phrases ne nécessite que de petites modifications de l'algorithme de recherche et de l'apprentissage du TDNN. En particulier, il faut ajouter un nouvel état qui représente le nouveau nœud dans l'arbre de recherche à la couche d'entrée de TDNN.

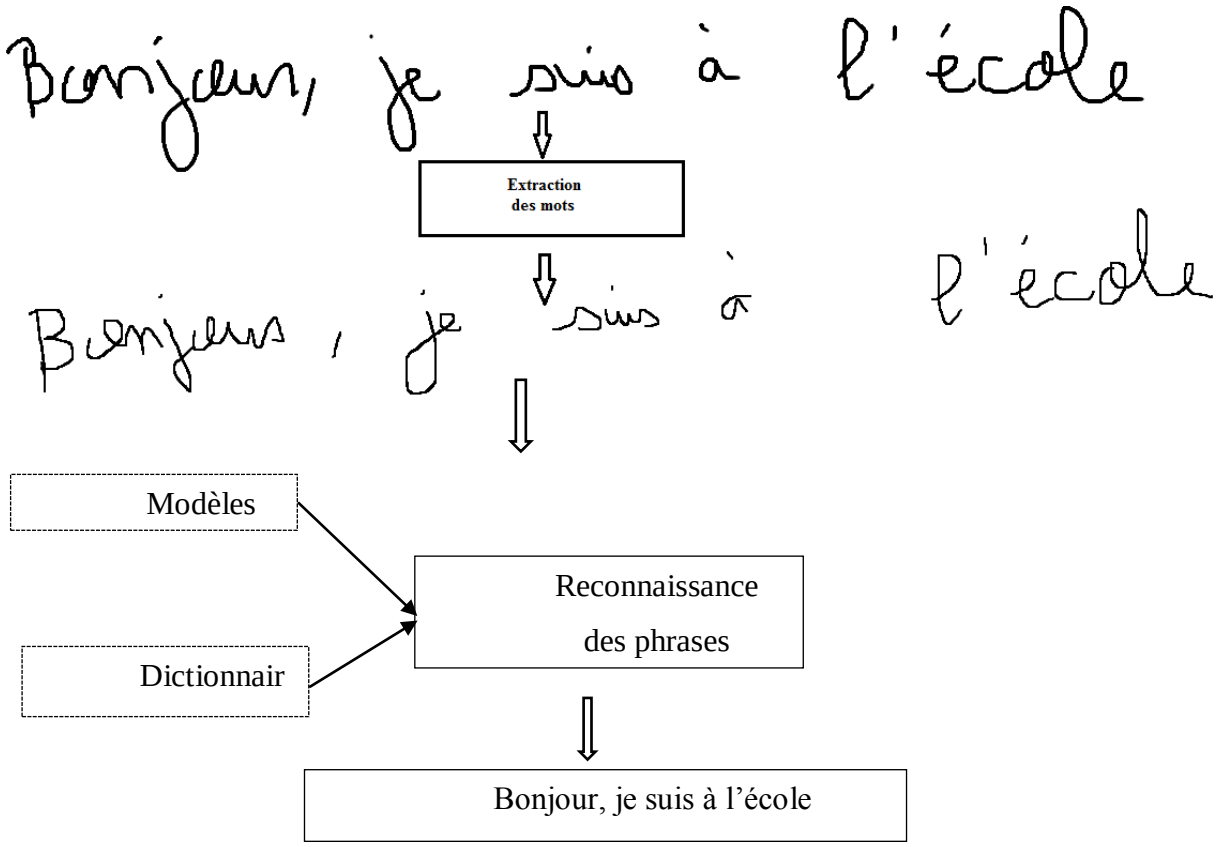

Figure 28 : Processus de la reconnaissance d'une phrase.

d. Calcul de la probabilité

Parmi les étapes, la plus difficile dans la phase d'entraînement ou de reconnaissance concerne le calcul de la probabilité de la vraisemblance du mot. Étant donné l'encre d'un mot à reconnaître ou l'observation de mot O, et un lexique des mots, le mot qui est considéré comme le mot reconnu, W est celui qui a le score le plus élevé parmi tous les mots.

$$W = argmax\, P(W|O)$$

La figure ci-contre montre la complémentarité entre le calcul statistique (calcule de la probabilité) et l'utilisation de dictionnaire.

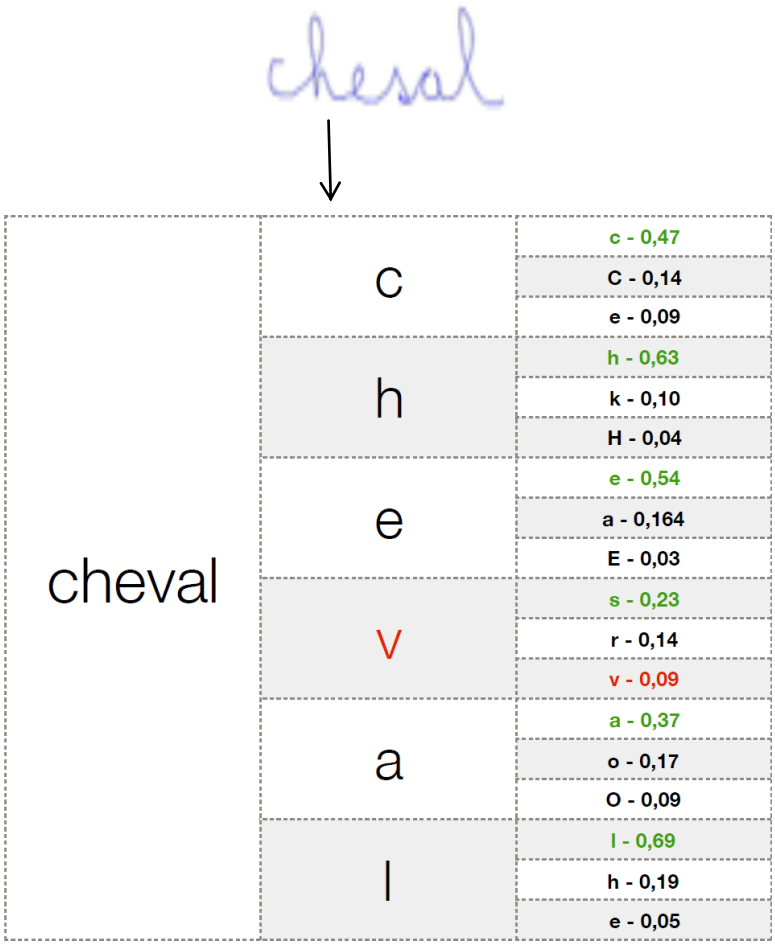

Figure 29 : Rôle de dictionnaire dans le système de reconnaissance de l'écriture.

Nous remarquons que pour la quatrième lettre, le système propose, en première étape, une probabilité de 0,23 pour le « s » et seulement « 0,09 » pour un « v » mais qu'après utilisation du modèle de langage, il préférera la sélection de la lettre « v », on peut donc en déduire que le v était "mal" écrit ou "ambigu".

3 Reconnaissance de l'écriture manuscrite de l'arabe

Après l'implémentation des différents modules de notre système de reconnaissance de l'écriture pour le français, nous allons présenter dans cette section l'adaptation de notre système pour la langue arabe. Donc, on va garder l'approche de réseau de neurones ainsi que tous les algorithmes développés pour le français.

3.1 Difficultés de la reconnaissance de l'écriture arabe

L'écriture arabe est une écriture consonantique et naturellement cursive composée de 28 lettres de base, 12 lettres supplémentaires spéciales (par exemple : ﺇ, ﺃ, ﺆ, ﺄ), et huit signes diacritiques (Bouslama et al, 1998). La langue arabe s'écrit de droite à gauche.

Le traitement automatique de la langue arabe est une tâche très difficile, surtout dans le domaine de la reconnaissance de l'écriture manuscrite pour plusieurs raisons :

- La majorité des lettres changent légèrement leur forme en fonction de leur position dans le mot ; initiale, moyenne, finale, ou tout simplement isolée (voir le tableau ci-dessous).

à la fin d'une lettre non joignable	à la fin	au milieu	au début
ع	ـع	ـعـ	عـ

Tableau 11 : changement de la forme d'une lettre selon sa position, exemple de variation de la lettre ع « Ayn »

- Beaucoup de lettres arabes contiennent des points et des traits en plus collés à la lettre telle que la lettre *Alif Mad* « آ ». Ils sont généralement ajoutés par le scripteur à la fin d'un mot manuscrit reporté.
- Certains mots en arabe contiennent des ligatures, comme le montre la figure ci-dessous.

Correct	ل+ا ← لا
Incorrect	ل+ا ← لا

Figure 30 : Ligature arabe *Lam Alif*[12]

- Dans l'écriture manuscrite arabe, le style de l'écriture est variable d'un scripteur à un autre.
- Un mot arabe doit être écrit de manière cursive et les caractères doivent être connectés au moins à un caractère existant au milieu du mot (voir la figure ci-après).

[12] http://fr.wikipedia.org/wiki/Ligature_%28%C3%A9criture%29

Figure 31 : La connectivité entre les caractères arabes

De plus, les scripteurs arabes ont l'habitude de négliger les signes diacritiques lors de l'écriture (Amin, 1997), (Märgner et al, 2008), car le sens du mot peut être déduit à partir du contexte. Comme nous l'avons déjà montré dans l'état de l'art, dans le traitement automatique de la langue arabe, la segmentation des mots est une tâche difficile.

3.2 Processus de reconnaissance

Dans cette phase, nous allons appliquer notre système de reconnaissance sur la langue arabe, en respectant les démarches présentées dans la figure ci-dessous.

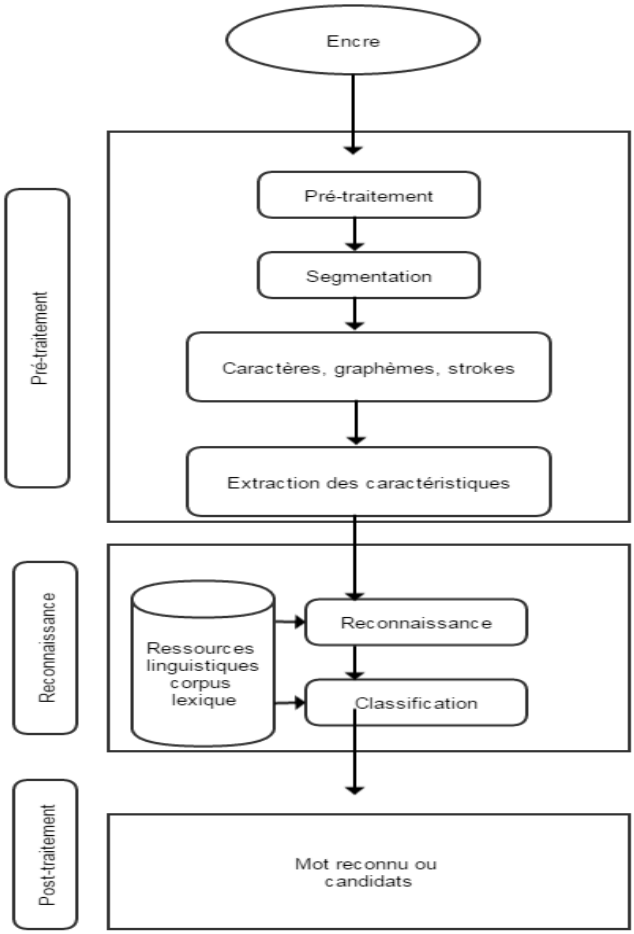

Figure 32 : Processus de la reconnaissance de l'écriture manuscrite pour la langue arabe

3.2.1 Prétraitement et normalisation

Le prétraitement est une phase indispensable dans le processus de la reconnaissance de l'écriture et il est crucial pour arriver à un meilleur taux de reconnaissance. Généralement, avant la reconnaissance, les encres acquises sont généralement prétraitées. Par conséquent, l'objectif principal des étapes de prétraitement est de réduire le bruit, pour éliminer les imperfections matérielles et le tremblement lors de l'écriture, et de normaliser les différents aspects de l'encre (Mezghani et al, 2008).

3.2.2 La détection de référence

Plusieurs approches récentes considèrent un autre niveau important qui doit suivre la phase de lissage, filtrage et ré-échantillonnage. C'est la procédure d'estimation de base du mot selon une référence. Dans cette étape, nous avons appliqué le même algorithme utilisé pour la détection des lignes de référence pour le français.

3.2.3 Manipulation de strokes retardés

Dans les scripts arabes, les strokes retardés sont écrits au-dessus ou au-dessous d'une partie de mot et peuvent apparaître avant, après, ou dans une partie de mot par rapport à l'axe horizontal (Biadsy et al, 2011). Beaucoup des chercheurs en TAL arabe considèrent que les strokes retardés ajoutent plus de complexité au processus de reconnaissance de l'écriture manuscrite en ligne et devraient être totalement éliminées à partir de l'encre dans l'étape de prétraitement (Mezghani et al, 2008), (Daifallah et al, 2009), (Kherallah et al, 2009). Afin de corriger ce problème, nous allons créer un module qui ajoute un stroke spécial de connexion afin d'attacher les strokes retardés au mot auquel ils appartiennent (Sternby et al, 2009). Cela veut dire que notre système traite dynamiquement les variations liées à l'attachement de diacritiques en utilisant une technique de discrimination entre les différentes hypothèses de mots qui ont les mêmes formes de base, mais un attachement différent des diacritiques. Cette technique est appliquée aux mots et aux caractères.

3.2.4 Extraction de caractéristiques

Le but de l'extraction de caractéristiques consiste à réduire le motif d'entrée en extrayant et en calculant les caractéristiques ou les paramètres du signal d'entrée les plus pertinents pour atteindre la meilleure classification des modèles. Pour l'extraction des

caractéristiques, nous avons utilisé les mêmes paramètres utilisés pour le français ainsi que le même algorithme d'extraction.

3.2.5 Segmentation

La segmentation fait référence aux différentes opérations qui doivent être effectuées par le système de reconnaissance pour obtenir des unités de base significatives (stroke, graphème, caractère, etc.) que l'algorithme de reconnaissance peut traiter. Cette phase est composée généralement de deux niveaux. Le premier niveau porte sur l'ensemble du texte et se concentre sur la détection de ligne. Au second niveau, le module met l'accent sur la segmentation de l'entrée en petites unités individuelles telles que les strokes, les graphèmes ou les caractères. Cette opération est parmi les plus difficiles dans le processus de la reconnaissance de l'écriture (Abuzaraida et al, 2010).

3.2.6 Reconnaissance

Le module de reconnaissance dans le système correspond à l'application d'algorithmes de classification. Il utilise les données d'entraînement pour reconnaître les unités individuelles fondamentales. Dans la phase de reconnaissance, le système procède à une comparaison du motif de test avec chaque classe de référence qui représente les mots du lexique et de la mesure d'une note de similarité entre le motif de test et la classe de référence. Le score de similarité est utilisé pour décider quel modèle correspond le mieux à la forme inconnue. La mise en œuvre de ce module de reconnaissance dans les systèmes de reconnaissance nécessite certaines approches telles que l'arbre de décision (Ghods et al, 2010), la programmation dynamique (Elanwar et al, 2007), modèle de Markov caché (Boubaker et al, 2010), réseau des neurones (Jouini et al, 2003), k-plus proche voisin (Daifallah et al, 2009), et d'autres combinaisons de différentes approches (Halavati et al, 2005).

Généralement, le processus de reconnaissance fournit une liste d'hypothèses de mots ou tout simplement le mot ou le caractère reconnu. Le système utilise certaines sources de connaissance sous la forme d'un modèle de langage qui permet au système d'améliorer la qualité de la reconnaissance obtenue. Dans un système de reconnaissance, un modèle de langage peut être le lexique, qui est un dictionnaire ou une liste de mots possibles de reconnaissance ou les mots qui sont autorisés en tant qu'entrée pour le système de reconnaissance, mais peut également inclure des propriétés statistiques ou des structures d'une langue donnée (Ahmad, 2008).

Concernant la phase d'apprentissage et d'entraînement de réseau de neurones, nous avons utilisé le même algorithme développé pour la langue française.

4 Construction des ressources linguistiques et le corpus d'encre

Pour le développement, l'entraînement et le test d'un système de reconnaissance de l'écriture manuscrite, nous avons besoin de trois types de ressources linguistiques ; lexique, corpus de textes et corpus manuscrits. Concernant les lexiques, nous avons gardé les mêmes lexiques utilisés dans notre plate-forme (chapitre 7) pour les deux langues, arabe et française, car, la taille de deux lexiques est importante et permet l'entraînement des réseaux des neurones. Pour le corpus de texte, nous avons gardé le même corpus pour la langue arabe (corpus Wikipédia de 25 000 000 mots). Par contre, nous avons besoin de construire un corpus pour le français. De plus, un système de reconnaissance de l'écriture nécessite un corpus de textes manuscrits pour le développement, l'entraînement et le test du système.

4.1 Construction de corpus de textes pour le français

La collecte d'encre nécessite un corpus de textes varié, riche et fiable (mots corrects, pas de forums ou chats, pas de coquilles). Pour cela, nous avons construit notre propre corpus manuellement. Pour ce faire, nous avons adapté la même approche utilisée pour la construction de corpus arabe en utilisant le web. Actuellement, l'utilisation du web pour la construction rapide des corpus dans différents domaines et différentes langues devient de plus en plus fréquente (Meftouh et al, 2007). Nous avons choisi le site web du journal Le Monde[13] pour la construction de notre corpus, car il est riche au niveau des thèmes (sport, politique, culture, économie, finance, etc.).

La première tâche est d'acquérir des données textuelles à partir du site Le Monde. Pour cela, nous avons aspiré les pages HTML à partir d'une adresse web donnée. Puis, une phase de conversion du HTML en texte brut, afin de rendre le corpus exploitable linguistiquement. La partie cruciale dans cette procédure de construction du corpus est le nettoyage des fichiers des textes obtenus, c'est-à-dire isoler et extraire les informations utiles à inclure dans notre corpus. Nous voulons supprimer les informations non utiles (menu, des liens vers d'autres

[13] http://www.lemonde.fr/

pages, graphiques, images, les styles, les scripts, etc.). Enfin, nous avons obtenu un corpus propre de 15 000 000 mots.

4.2 Construction des corpus d'encre

Un moteur de reconnaissance de l'écriture manuscrite nécessite une base de données qui contient des informations de référence en ce qui concerne la forme de chaque caractère ou de chaque mot. Ces informations de référence aident le système de reconnaissance à interpréter l'encre. De plus, ces bases utilisées permettent la validation du comportement de notre solution à base de réseaux de neurones développée ainsi que les tests de performance.

Pour être générique et être capable d'interpréter autant de variations que possible, les données de référence collectées devraient être fondées sur des écritures autant diversifiées que possible. En effet, les grands corpus avec de grands ensembles de données sont une condition essentielle pour la recherche et le développement dans le domaine de la reconnaissance d'écriture.

Actuellement, il existe des bases de données d'encre qui ont été développées. Par exemple, pour l'écriture latine, de nombreuses bases de données existent comme pour le français UNIPEN (Guyon et al, 1994), IRONOF (Viard-Gaudin et al, 1999), NIST, etc. Ces bases d'encre contiennent des caractères isolés, des chiffres et des mots en français au format UNIPEN. De plus, elles présentent l'avantage de contenir un nombre important de données provenant de plus de 3 000 scripteurs (2200 scripteurs pour UNIPEN et 700 scripteurs pour IRONOF) ; cependant, ces bases ont une qualité d'acquisition des tracés qui n'est pas constante. Vu ces contraintes liées à la qualité de l'encre et au type de contenu (caractère isolé, mot), ces bases ne peuvent pas être exploitées dans nos travaux de recherche et développements d'un système de reconnaissance de l'écriture manuscrite à destination de l'ALAO.

Pour l'écriture arabe, de nombreuses bases de données ont été produites, mais peu sont accessibles au public. En fait, les chercheurs ont développé leurs bases de données privées représentant généralement de petits dictionnaires avec lexique limité ou tout simplement des formes isolées de lettres et / ou des chiffres. Le nombre de scripteurs est aussi limité. En conséquence, jusqu'à présent, il n'y a aucune base de données d'encre complète et robuste consacrée à la reconnaissance de l'écriture manuscrite en de script arabe.

Après l'étude de bases de données d'encre existantes et vue l'incompatibilité (au niveau taille, diversité des scripteurs, contenu, etc.) de ces bases avec nos besoins pour le

développement d'un système d'apprentissage de langue, nous avons décidé de construire nos propres bases. Comme la phase de la collecte de l'encre est très coûteuse, nous avons travaillé cette partie en collaboration avec la société 3AO. Dans cette section, nous décrivons la phase de préparation de la collecte, puis le déroulement de la collecte ainsi les résultats obtenus.

4.2.1 Collecte de l'encre

Dans cette étape, nous devons préparer 200 formulaires pour le français et 200 formulaires pour l'arabe contenant les textes à faire recopier par les intervenants ou les scripteurs. Chaque formulaire est composé de six pages : habituellement, trois pages contiennent l'ensemble des nombres et des mots, deux pages contiennent l'ensemble des phrases courtes et phrases longues, une page contenant des paragraphes. Par conséquent, il faut préparer des données pour générer 200 formulaires (600 pages) par langue. Le but est d'utiliser 170 formulaires pour l'entraînement de notre système et 30 formulaires pour le test. Pour cela nous avons préparé les données nécessaires pour générer 400 formulaires pour le français et l'arabe.

- Les caractères : La première partie du formulaire est composée de l'ensemble des caractères de la langue ; l'alphabet qui contient l'ensemble des lettres d'une langue donnée (minuscules et majuscules), les nombres[14] qui contiennent l'ensemble des nombres utilisés dans la langue et les symboles tels que les signes mathématiques (+, -, *, etc.), les signes de ponctuation, les symboles utilisés souvent dans la rédaction (@, &, #, $, etc.).
- Les mots : Ils représentent l'ensemble des mots extraits à partir de notre lexique.
- Les phrases : Elles représentent l'ensemble des phrases longues et courtes. Ces phrases ont été extraites à partir de corpus (Wikipédia pour l'arabe et Le Monde pour le français). Pour les phrases courtes, la taille d'une phrase doit être entre 30 et 50 caractères (habituellement, 10 phrases courtes par formulaire), pour les phrases longues, la taille d'une phrase doit être entre 90 et 130 caractères (habituellement, trois phrases courtes par formulaire).
- Les textes : Ils contiennent deux ou trois phrases d'une longueur comprise entre 30 et 130 caractères pour un total compris entre 200 et 240 caractères (généralement, un

[14]La langue arabe supporte les nombres arabes (٠, ١, ٢, ٤, etc.) et les nombres latins (1, 2, 3, etc.) car dans certains pays arabes (Tunisie, Maroc, Algérie, etc.) les gens utilisent les numéros latins.

texte par formulaire). Ces paragraphes ont été extraits à partir de corpus (Wikipédia pour l'arabe et Le Monde pour le français).

Les trois ensembles (phrases courtes et longues, et textes) doivent être mélangés alphabétiquement.

Une fois les données prêtes, nous avons vérifié leur qualité (encodage, propreté des lexiques et textes, etc.). Puis, nous avons développé un outil Perl qui permet à partir des données préparées de générer les formulaires finaux. Enfin, une phase de vérification de contenu des formulaires est nécessaire. Cette phase doit être effectuée par une personne native.

4.2.2 Phase de la collecte de l'encre

Une fois les formulaires générés et vérifiés, la prochaine étape consiste à les imprimer dans des formulaires papiers Anoto. Nous avons besoin de 200 intervenants natifs français et 200 intervenants natifs arabes pour le remplissage des formulaires. De plus, nous avons besoin de stylos numériques[15] qui sont nécessaires pour récupérer l'encre à la fin de cette phase. Dans notre collecte, nous avons utilisé des stylos numériques Nokia SU-1B. Ils ont une résolution spatiale de 677 points par pouce (dpi) et une fréquence d'échantillonnage de 100 points par seconde (100Hz).

Figure 33 : Stylo numérique

[15] Un stylo numérique ou stylo intelligent : est un stylo numérique à encre capable d'enregistrer les traits tracés par l'utilisateur, de manière à les transmettre à un ordinateur et les rendre utilisables.

À l'aide de la société 3AO, nous sommes arrivés à obtenir le nombre souhaité des intervenants pour les deux langues. Le tableau suivant montre les résultats obtenus concernant l'encre collectée (les résultats obtenus sont les mêmes pour les deux langues) :

Type de données	Taille de données
Caractères	40 600 caractères
Mots	7 200 mots
Phrases courtes	2 000 phrases
Phrases longues	600 phrases
Textes	200 textes

Tableau 12 : Base des données d'encre obtenue après la phase de collecte.

Notons qu'à la fin de la collecte, nous avons converti la totalité de l'encre, à l'aide d'un programme développé en C, au format Unipen (Guyon et al, 1994).

5 Expériences et tests fonctionnels

Le test d'un système est une étape cruciale pour valider les algorithmes ainsi que les approches utilisées dans la construction de système. Pour cela, nous avons procédé à une phase de test et une phase de comparaison de notre système à un autre qui donne des bons résultats (Yin et al, 2013) pour la langue arabe et la langue française.

Dans la section suivante, nous présentons les résultats de test de notre système pour les deux langues.

5.1 Évaluation et test

Nous avons développé un système de reconnaissance d'écriture en ligne complet qui implémente l'approche de réseaux de neurones. Il y a diverses questions qui doivent être abordées afin de permettre la mise en œuvre du système. Pour tester le système, nous avons mené un certain nombre d'expériences. Nous avons utilisé les bases de données que nous avons collectées pour les expériences afin de tester la validité et l'utilité de notre système à diverses étapes de mise en œuvre.

La base précédemment présentée a été découpée selon un partage de 170 formulaires (1020 pages) pour la base d'apprentissage et 30 formulaires (180 pages) pour la base de test. Nous disposons donc des bases d'encre suivantes avec les mêmes proportions pour l'arabe et le français.

Type de données	Taille de données pour la base d'apprentissage	Taille de données pour la base du test
Caractères	43 510 caractères	6 090
Mots	6 120 mots	1 080
Phrases courtes	1 700 phrases	300
Phrases longues	510 phrases	90
Textes	170 textes	30

Tableau 13 : Taille de la base d'apprentissage et la base de test

Pour la phase d'évaluation, nous avons essentiellement retenu les informations suivantes :

- Le taux de la reconnaissance sur la base de test pour l'ensemble des exemples, en phase d'apprentissage et de généralisation.
- À chaque itération, nous calculons l'évolution de la fonction de coût à minimiser. Pour que l'apprentissage soit correct, la valeur de coût doit décroître, puis se stabiliser. Cette fonction permet aussi le contrôle de la vitesse de convergence du réseau.

Pour calculer la fonction de coût, nous avons choisi tant pour le PMC que pour la structure TDNN la distance de Kullback-Leibler, donnée par l'équation suivante :

$$KL^k = \sum_i d_i^k * \log(d_i^k) + \sum_i d_i^k * \log(s_i^k)$$

Avec :

k : indice de l'exemple.

i : indice de la classe.

d_i : désigne la sortie désirée.

s_i : désigne la sortie réelle d'un neurone appartient à la classe i pour un exemple k.

La distance de Kullback-Leibler sert à évaluer la différence des distributions de probabilités calculées sur la base d'apprentissage entre les classes reconnues et les vrais labels.

5.2 Expériences et résultats

Dans cette section, nous allons présenter les résultats qui correspondent à l'évaluation de notre système de reconnaissance de l'écriture manuscrite. Cette évaluation est effectuée sur les bases d'apprentissage et de tests résultant de la collecte. Nous avons utilisé essentiellement le taux de reconnaissance correcte pour évaluer les performances de notre système. De plus, nous allons évaluer notre système par rapport à celui de MyScript[16].

5.2.1 Système MyScript

La phase d'évaluation d'un système est très importante, car, elle permet de montrer les points forts et les limites du système. Cela nous aide à trouver une éventuelle amélioration de performance de notre système. Afin d'obtenir une évaluation précise de notre système de reconnaissance, nous avons choisi un système performant et qui donne de bons résultats pour la langue arabe et la langue française, MyScript.

Le système MyScript est un logiciel pour la reconnaissance de l'écriture manuscrite en ligne développé par la société Vision Object (Yin et al, 2013). C'est un système payant, mais la société offre aux développeurs la possibilité de tester leur moteur de reconnaissance via le Cloud (serveur à distance en ligne) pour une période de trois mois. Nous avons choisi ce logiciel, car il est fonctionnel pour la langue arabe et pour la langue française et il donne de bons résultats, donc nous avons la possibilité de comparer ces résultats aux nôtres. Par contre, il n'existe pas des informations sur les algorithmes et les approches utilisés par le système MyScript.

Dans le but de tester MyScript sur nos bases, nous avons demandé un code[17] pour accéder à leur système à distance. Puis, nous avons développé un programme, écrit en Perl et en PHP, qui permet de lancer MyScript sur nos bases de tests. Enfin, nous appliquons la formule de calcul de taux de la reconnaissance correcte sur les résultats obtenus par le moteur de MyScript.

[16] http://www.myscript.com/

[17] https://dev.myscript.com/

5.2.2 Reconnaissance des caractères

Généralement, l'étape d'évaluation d'un système de reconnaissance consiste à comparer le taux de reconnaissance correct de ce dernier avec celui du. Nous avons évalué les deux systèmes en utilisant les mêmes bases de tests.

Le tableau suivant montre le résultat donné par le système de la reconnaissance de la langue française sur les sous-bases de caractères :

Type de données	Taille de données pour la base d'apprentissage	Taille de données pour la base du test	Taux Reco (%)	Taux MyScript(%)
Caractères	43 510 caractères	6 090	99,3	95,1

Tableau 14 : Tableau comparatif de résultats obtenus sur les bases de test de caractères français.

Le tableau suivant montre le résultat donné par le système de la reconnaissance de la langue arabe sur les sous-bases de caractères :

Type de données	Taille de données pour la base d'apprentissage	Taille de données pour la base du test	Taux Reco (%)	Taux MyScript(%)
Caractères	43 510 caractères	6 090	98,5	90,3

Tableau 15 : Tableau comparatif de résultats obtenus sur les bases de test de caractères arabes

Les deux tableaux ci-dessus montrent les résultats de l'évaluation de notre système pour l'ensemble de deux sous-bases de tests de caractères arabe et français. Nous pouvons voir que nos résultats sont efficaces, car nous arrivons à connaître presque tous les caractères. En fonction du calcul de taux de reconnaissance, la précision du système est de 99,3 % pour le français et 98.5% pour l'arabe. Lors de la phase de reconnaissance, notre système est presque en temps réel. Il prend moins d'une seconde pour produire les 20 premiers candidats possibles pour un échantillon donné.

Le système de reconnaissance de MyScript donne de bons résultats pour la reconnaissance des caractères français, par contre, les performances de ce moteur est moins bon dans le cas de la reconnaissance de caractères arabes.

La taille importante et la diversité de nos bases des caractères ont permis de réaliser l'apprentissage du Perceptron multicouche et de TDNN dans des bonnes conditions. Cela nous permet d'avoir un taux élevé de reconnaissance sur nos bases de tests.

5.2.3 Reconnaissance des mots et phrases

Afin d'évaluer la performance de deux systèmes sur les bases de tests (mots et phrases), nous avons calculé la précision des mots reconnus par rapport à l'ensemble des mots de la base de test. Cependant, pour le calcul de taux sur la base de phrases, nous avons calculé la précision de chaque mot dans la phrase. Pour calculer la précision de chaque phrase, nous utilisons la formule suivante :

$$W_A = \frac{N - E}{N}$$

Avec : N représente le nombre de mots dans la phrase du test.

Le tableau suivant montre le résultat donné par le système de la reconnaissance de la langue française sur les sous-bases de mots, phrases courtes, phrases longues et textes :

Type de données	Taille de données pour la base d'apprentissage	Taille de données pour la base du test	Taux Reco (%)	Taux MyScript(%)
Mots	6 120 mots	1 080	97,5	90,5
Phrases courtes	1 700 phrases	300	95.2	86,7
Phrases longues	510 phrases	90	94.6	85,2
Textes	170 textes	30	93.1	81.4

Tableau 16 : Tableau comparatif de résultats obtenus sur les bases de test français.

Le tableau suivant montre le résultat donné par le système de la reconnaissance de la langue arabe sur les sous-bases de mots, phrases courtes, phrases longues et textes :

Type de données	Taille de données pour la base d'apprentissage	Taille de données pour la base du test	Taux Reco (%)	Taux MyScript(%)
Mots	6 120 mots	1 080	96.9	88,9
Phrases courtes	1 700 phrases	300	91.7	82,3
Phrases longues	510 phrases	90	89.3	78,9
Textes	170 textes	30	87.8	74,7

Tableau 17 : Tableau comparatif de résultats obtenus sur les bases de test arabe.

Les deux tableaux ci-dessus montrent un comparatif entre les taux de reconnaissance de notre système et celui de MyScript pour les sous-bases de mots, phrases et textes. Cependant, pour bien voir la différence entre les différents résultats obtenus, nous avons affiché les résultats des deux tableaux ci-dessus dans des histogrammes :

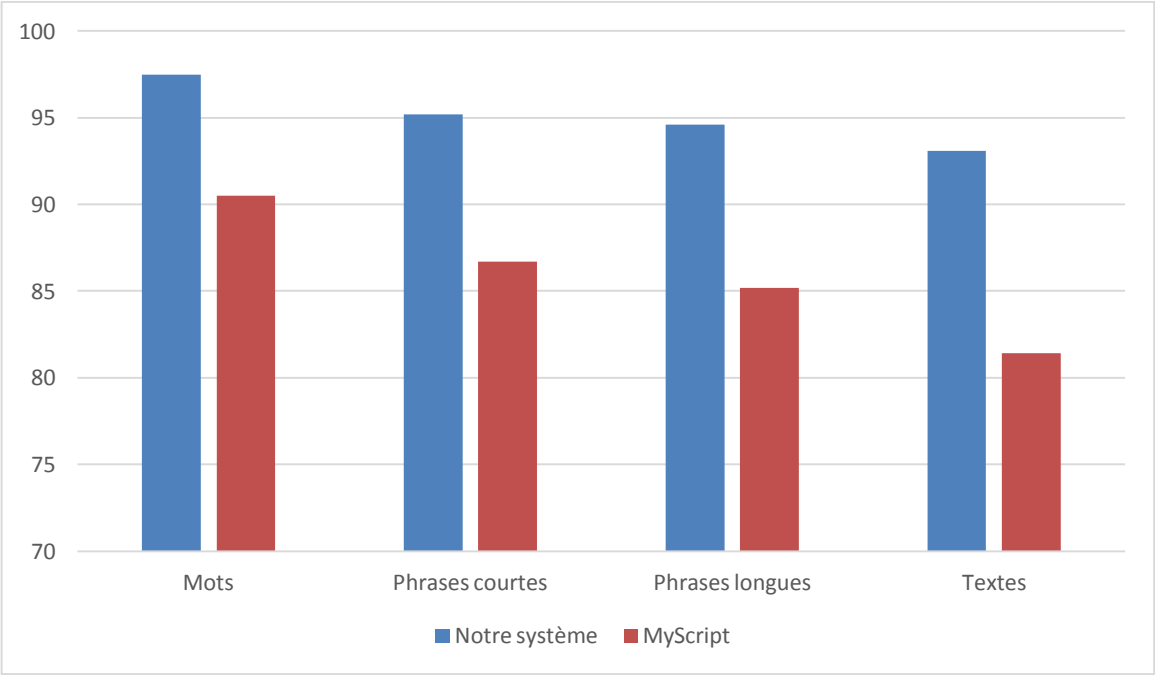

Figure 34 : Histogramme comparatif des résultats obtenus pour le français

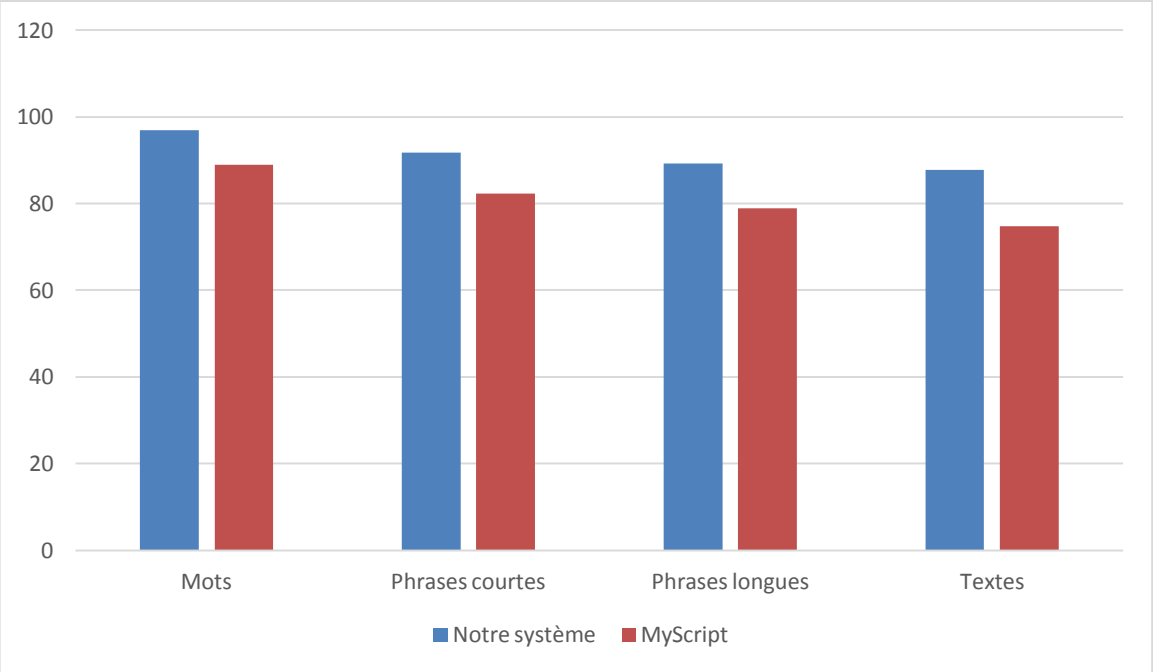

Figure 35 : Histogramme comparatif des résultats obtenus pour l'arabe

Les deux histogrammes montrent bien que notre système donne des résultats meilleurs que ceux obtenus par MyScript. Au niveau du français, le système MyScript donne des résultats acceptables. Par contre, pour la langue arabe les résultats obtenus par MyScript sont faibles surtout pour les phrases longues et les textes.

Le premier histogramme montre les résultats de la reconnaissance pour le français. Le taux d'erreur dans notre système pour les mots sur l'ensemble de la base de test (2,5 %) est faible, cela montre que les TDNN entraînés sur les bases des caractères arrivent à bien segmenter les mots en caractères. De plus, la taille importante de lexique apporte un énorme gain à notre système. Par contre, le test de nos bases de tests des phrases et textes montre que le taux d'erreur augmente avec l'augmentation du nombre des mots.

La comparaison de notre système avec celui de MyScript prouve que nos résultats sont bien classifiés. Cependant, notre système génère quelques fautes de reconnaissance pour les deux langues.

a. Erreurs de la reconnaissance du français

L'analyse des erreurs de reconnaissance de notre système montre trois types d'erreurs :

- Erreurs liées à la segmentation : certains scripteurs ont tendance à écrire les caractères accentués en attachant les accents aux lettres (voir la figure ci-dessous). Cela, rend la phase de segmentation du mot très difficile.

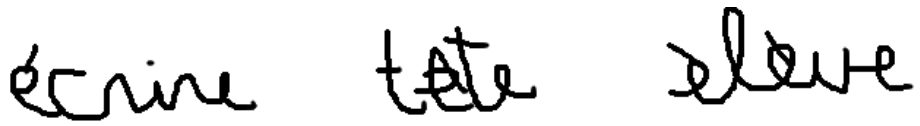

Figure 36 : Échantillons des mots avec accents attachés (écrire, tête, élève).

- Erreurs liées à la mauvaise écriture de la fin du mot : en analysant les mots mal reconnus, nous avons remarqué que certains scripteurs n'écrivent pas la totalité de mots, ils écrivent jute le début après ils mettent un trait pour les restes de caractères.

Figure 37 : Échantillons des mots avec un trait à la fin (reconnaissance, enfant).

- Erreur de mot composé : le système trouve des difficultés à reconnaitre les mots composés qui n'existent pas dans notre dictionnaire, tels que décrochez-moi-ça, m'as-tu-vu, etc.

b. Erreurs de la reconnaissance de l'arabe

De même, l'analyse des erreurs de mots arabes non reconnus montre ces erreurs :

- Erreurs liées aux diacritiques : lors de la phase de segmentation, notre système rencontre des difficultés pour segmenter les mots qui n'existent ni dans les bases d'apprentissage ni dans le dictionnaire. Parmi ces mots, on trouve ceux qui sont écrits avec le diacritique Chadda « ّ », ceux qui sont écrits avec la lettre Alif Madda « آ », etc.

Figure 38 : Échantillon écrit avec le diacritique Chaddah.

- Erreurs liées à l'écriture superposée : dans la langue arabe, beaucoup des personnes écrivent les mots en superposant certaines lettres. Par conséquent, notre système génère des fautes lors de la segmentation des caractères superposés.

Figure 39 : Échantillon écrit avec une superposition de trois premières lettres.

De plus, on trouve certaines erreurs qui ressemblent à celles qui sont commises dans l'écriture manuscrite telles que les erreurs liées aux mots composés, erreurs liées à la mauvaise segmentation des mots inconnus (c'est-à-dire les mots qui n'existent ni dans les bases d'apprentissage ni dans le dictionnaire).

5.3 Développement technique

Pendant le développement de notre système de la reconnaissance de l'écriture manuscrite, nous avons implémenté l'approche de réseau de neurones pour l'apprentissage ainsi que les algorithmes prétraitement, segmentation et extraction de caractéristiques avec le langage c. Pour cela, nous avons utilisé le logiciel Visual studio, version 2010. Toutes les expériences sont réalisées en utilisant un ordinateur Dell avec 6G de RAM et un processeur Intel Xeon de 2,67 GHz dont le système d'exploitation est Windows 7. Notre projet ayant été développé avec le langage C, il peut être exportable pour l'utiliser sous Linux, Mac ou Android.

Nous avons travaillé tout au long de nos développements avec l'encodage Unicode utf-8.

6 Conclusion

Dans ce chapitre, nous avons développé un système multilingue pour la reconnaissance de l'écriture manuscrite. En premier lieu, nous avons décrit, le processus de développement d'un système de reconnaissance de l'écriture française destiné au caractère ainsi qu'un système de reconnaissance de mots. Ensuite, nous avons présenté les différentes tâches de développement d'un système de reconnaissance de l'écriture arabe. Enfin, nous avons montré notre approche de test et les résultats obtenus pour les deux langues, arabe et français.

Nous concluons que la reconnaissance de l'écriture manuscrite est une tâche difficile qui nécessite beaucoup des traitements complexes et coûteux. La disposition d'un dictionnaire très riche et varié est considérée comme une voie prometteuse pour résoudre le problème de reconnaissance lié aux mots avec des diacritiques (par exemple Chadda « ّ ») dans le cas de la langue arabe et le problème de segmentation de mots composés. De plus, l'utilisation d'un modèle du langage pour le français et l'arabe est aussi considérée comme une voie prometteuse pour améliorer le taux de reconnaissance sur les phrases et les textes. L'utilisation de l'approche de réseau des neurones a donné des résultats très efficaces au

niveau caractère, par la suite des bons résultats au niveau mots et phrases. Ces derniers sont très importants puisque nous comptons mettre notre système dans une plate-forme d'apprentissage des langues assisté par ordinateur (ALAO).

Chapitre 7 : Présentation et développement des outils TAL

1 Introduction

Dans ce chapitre, nous abordons brièvement l'apport du TAL dans l'ALAO. Puis, nous présentons quelques travaux dans le domaine de l'analyse morphologique des textes. Par la suite, nous allons parler des deux analyseurs morphologiques nécessaires pour nos travaux. Enfin, nous présentons les étapes d'élaboration de nos ressources lexicales pour l'arabe et le français.

2 L'utilisation du TAL dans l'ALAO

Le traitement automatique de la langue est la technologie qui fournit des méthodes de traitement pour l'analyse des textes humains ; on l'utilise dans des applications telles que la traduction automatique, la reconnaissance de l'écriture ou l'extraction de l'information. Ces méthodes ont subi des améliorations en termes d'efficacité et de robustesse. Certaines recherches concernant l'apprentissage des langues assisté par ordinateur (Antoniadis, 2010), (Mars et al, 2014) ont montré que l'intégration des technologies TAL dans des systèmes d'apprentissage des langues permet de prendre en considération les caractéristiques de la langue de référence et celle de l'apprenant. Cela devrait permettre aux systèmes d'ALAO une meilleure compréhension des productions des apprenants. De plus, ces systèmes devraient être capables de produire un diagnostic détaillé des difficultés linguistiques de chaque apprenant. D'autre part, l'utilisation du TAL dans l'ALAO permet de rendre les environnements d'apprentissages plus autonomes en terme des fiabilités et des robustesses.

3 Outils TAL pour le Français

3.1 Étude des analyseurs morphologiques

L'un des éléments fondamentaux de toute application linguistique est l'analyse morphologique. Dans une application linguistique, le rôle d'un analyseur morphologique est de fournir des informations morphologiques sur le mot en termes de type, nombre, genre, etc.

Actuellement, les analyseurs morphologiques du français sont plus ou moins robustes, en particulier l'étiquetage de mots inconnus (les mots qui ne sont pas dans le corpus d'apprentissage).

Dans la section suivante, nous présentons les analyseurs morphologiques, pour la langue française, les plus utilisés par la communauté du TAL.

3.1.1 L'analyseur Brill Tagger

Cet analyseur est développé par Eric Brill dans le cadre de sa thèse en 1992 (Brill, 92). Brill Tagger utilise une approche à base des règles. Le principe de cette approche est de se baser sur les règles morphologiques ou grammaticales pour l'affectation des étiquettes aux mots. Ces règles sont construites lors de la phase d'apprentissage à partir d'un corpus étiqueté. De plus, Brill tagger classe les règles en deux types :

- Règles contextuelles : l'utilisation de ces règles permet à l'analyseur d'affiner l'assignation de l'étiquette. En utilisant le contexte local de la phrase, l'étiqueteur examine et corrige les étiquettes précédemment affectées (Thibault, 2004).
- Règles lexicales : l'utilisation de ces règles permet à l'analyseur d'affecter l'étiquette du mot selon ses propriétés lexicales.

L'étiquetage se déroule en deux phases : dans la première, l'analyseur assigne l'étiquette la plus probable au mot, puis, dans la deuxième phase, le système applique les règles contextuelles en examinant les étiquettes précédemment affectées en corrigeant les étiquettes qui ne sont pas correctes.

3.1.2 L'analyseur TreeTagger

TreeTagger[18] est un outil d'étiquetage open source développé à l'université de Stuttgart par Helmut Schmid (Schmid, 1994). Cet analyseur permet d'étiqueter automatiquement des textes en plusieurs langues (français, anglais, allemand, italien, néerlandais). Comme approche, TreeTagger utilise les arbres de décision pour estimer la probabilité bi-gramme de décision. C'est-à-dire, qu'on assigne une étiquette au mot à la position i sachant un historique, en utilisant la formule suivante :

$$argmax_{t\in F}\ P\left(w_1 w_{2\ldots w_n}, t_1 t_2 \ldots\right) = P\left(\frac{t_n}{t_{n-2}t_{n-1}}\right).P\left(\frac{w_n}{t_n}\right).P(w_1 w_2 \ldots w_{n-1}, t_1 t_2 \ldots t_{n-1})$$

Équation 3 : Formule de calcul des probabilités n-gramme

L'utilisation de cette formule permet à l'étiqueteur de calculer la probabilité d'une étiquette par rapport aux étiquettes précédentes. Le calcul de la probabilité se fait à partir d'un arbre de décision (Quinlan, 1986).

3.1.3 L'analyseur Melt

C'est un étiqueteur[19] morphosyntaxique multilingue (français, anglais, espagnol, italien, allemand) développé en 2009 dans le cadre du projet ANR EDyLex (Pascal, 2009). Les modèles utilisés par Melt sont entrainés sur le corpus français Treebank (Abeillé et al, 2003). Cet analyseur basé sur les modèles de Markov cachés (HMM) à maximisation d'entropie, utilise un lexique à large couverture : le Leff (Sagot, 2010). Melt utilise 29 étiquettes. Pendant la phase de l'étiquetage, le système assigne l'étiquette la plus probable à chaque mot du texte, en appliquant la formule suivante :

$$\hat{t}_1^n = \arg\max_{t_1^n \in T^n} P(t_1^n|w_1^n) \approx \arg\max_{t_1^n \in T^n} \prod_{i=1}^{n} P(t_i|h_i)$$

Équation 4 : Formule de calcul de la probabilité maximale

Selon (Pascal, 2009) Melt atteint une précision de 97.75 % sur le corpus de test et 86.10% sur les mots inconnus. Ce dernier est extrait à partir du corpus French Treebank et contient 1235 phrases.

[18] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html
[19] http://gforge.inria.fr/projects/lingwb

3.1.4 LIA tagg

LIA_tagg[20] est un outil d'étiquetage open source développé au Laboratoire Informatique d'Avignon (Nasr et al, 2004). Ce système utilise l'approche basée sur les Chaînes de Markov Cachées (HMM). Il utilise 103 étiquettes. Les modèles de LIA tagg sont entraînés sur un corpus journalistique de 600 000 mots (nous n'avons pas d'information sur l'origine du corpus d'apprentissage).

3.1.5 Stanford tagger

C'est un analyseur morphologique[21] développé en 2003 à l'université de Stanford (Toutanova et al, 2003). Il supporte plusieurs langues (français, anglais, arabe, chinois, allemand) et il est open source. Les modèles de Stanford tagger sont entraînés sur le corpus French Treebank (Abeillé et al, 2003) en utilisant un jeu de 15 étiquettes. La particularité de cet analyseur est l'utilisation d'un modèle HMM bidirectionnelle en utilisant des propriétés morphologiques n-gramme sur les suffixes et les préfixes. Stanford tagger atteint une précision de 97.25 % sur le corpus de test.

3.1.6 LGtagger

LGtagger est un analyseur morphologique pour la langue française, développé en 2011 (Constant & al, 2011) à l'université Paris-Est. Il adapte l'approche de champs markoviens conditionnels pour apprendre ces modèles. Comme lexique, LGtagger utilise plusieurs ressources linguistiques françaises open source : DELA (Courtois, 1990), Leff (Sagot, 2010), Prolex (Piton at al, 1999), organisations et nom propre (Martineau et al, 2009). Selon (Constant & al, 2011) l'exactitude de LGtagger est de 97.7 %.

3.2 Choix de l'analyseur morphologique pour notre plate-forme

Dans le cadre de notre recherche, qui consiste à développer une plate-forme d'ALAO pour la langue française, nous avons besoin d'un analyseur morphologique fiable. Ce dernier est indispensable pour générer automatiquement les activités à travers notre plate-forme. Pour le traitement du français, deux analyseurs sont majoritairement utilisés par la communauté du TAL : TreeTagger et Brill (Dejean, 2010). Ces derniers, sont distribués librement dans le

[20] http://lia.univ-avignon.fr/chercheurs/bechet/download_fred.html

[21] http://nlp.stanford.edu/software/tagger.shtml

cadre de la recherche académique. Concernant, le performance de TreeTagger, Allauzen et Bonneau-Maynard (2008) ont montré que TreeTagger est plus performant que Brill. La précision d'étiquetage de treeTagger est de 95,7 % et celle de Brill atteint 94,6 %.

TreeTagger est disponible pour Linux et pour Windows, par contre Brill n'est pas disponible que sous Linux. Comme TreeTagger est Multi-plateformes et plus performant que Brill, nous avons choisi cet analyseur pour la suite de nos travaux.

Dans la section suivante, nous présentons l'analyseur Tree-Tagger, ainsi que son mode de fonctionnement.

3.2.1 Principe de TreeTagger

TreeTagger utilise un modèle probabiliste (statistique) basé sur l'arbre de décision (Schmid, 1994). Les modèles de l'analyseur ont été entraînés sur un corpus de 44 000 mots et un jeu de 33 étiquettes. Le tableau suivant montre les étiquettes utilisées par TreeTagger :

Étiquettes	Correspondant	Étiquettes	Correspondant
ABR	Abréviation	PRP	Préposition
ADJ	Adjectif	PRP:det	Préposition + Article
ADV	Adverbe	PUN	Ponctuation
DET:ART	Article	PUN:cit	Ponctuation de citation
DET:POS	Pronom Possessif	SENT	Balise de phrase
INT	Interjection	SYM	Symbole
KON	Conjonction	VER:cond	Verbe au conditionnel
NAM	Nom Propre	VER:futu	Verbe au futur
NOM	NOM	VER:impe	Verbe à l'impératif
NUM	Numéral	VER:impf	Verbe à l'imparfait
PRO	Pronom	VER:infi	Verbe à infinitif
PRO:DEM	Pronom Démonstratif	VER:pper	Verbe au participe passé

PRO:IND	Pronom Indéfini	VER:ppre	Verbe au participe présent
PRO:PER	Pronom Personnel	VER:pres	Verbe au présent
PRO:POS	Pronom Possessif	VER:simp	Verbe au passé simple
PRO:REL	Pronom Relatif	VER:subi	Verbe à l'imparfait du subjunctif
VER:subp	Verbe au présent du subjonctif		

Tableau 18 : Les étiquettes pour le français.

L'analyseur TreeTagger est basé sur l'approche de l'arbre de décision pour assigner une étiquette à un mot. Cet arbre est binaire, son rôle est d'estimer les probabilités de transition entre les différentes étiquettes.

L'analyseur prend en entrée un fichier texte, puis il segmente chaque phrase en un mot par ligne et enfin, il assigne à chaque mot l'étiquette la plus probable.

Exemple :

Phrase à analyser : Éric est un élève typique du primaire.

Résultat de l'étiquetage :

Éric NAM
est VER:pres
un DET:ART
élève NOM
typique ADJ
du DET:ART
primaire ADJ
. PUN

Avant de lancer la segmentation et l'étiquetage de TreeTagger, il faut vérifier la présence des fichiers suivants :

- French.par : fichier paramètre.
- Exemple.txt : le texte à étiqueter.
- tokenize.pl : programme Perl qui a pour but la segmentation du texte.
- french-utf8-abbreviations : fichier contenant la liste des abréviations utilisées dans la langue française.
- tree-tagger.exe : le programme nécessaire au lancement de l'étiquetage.

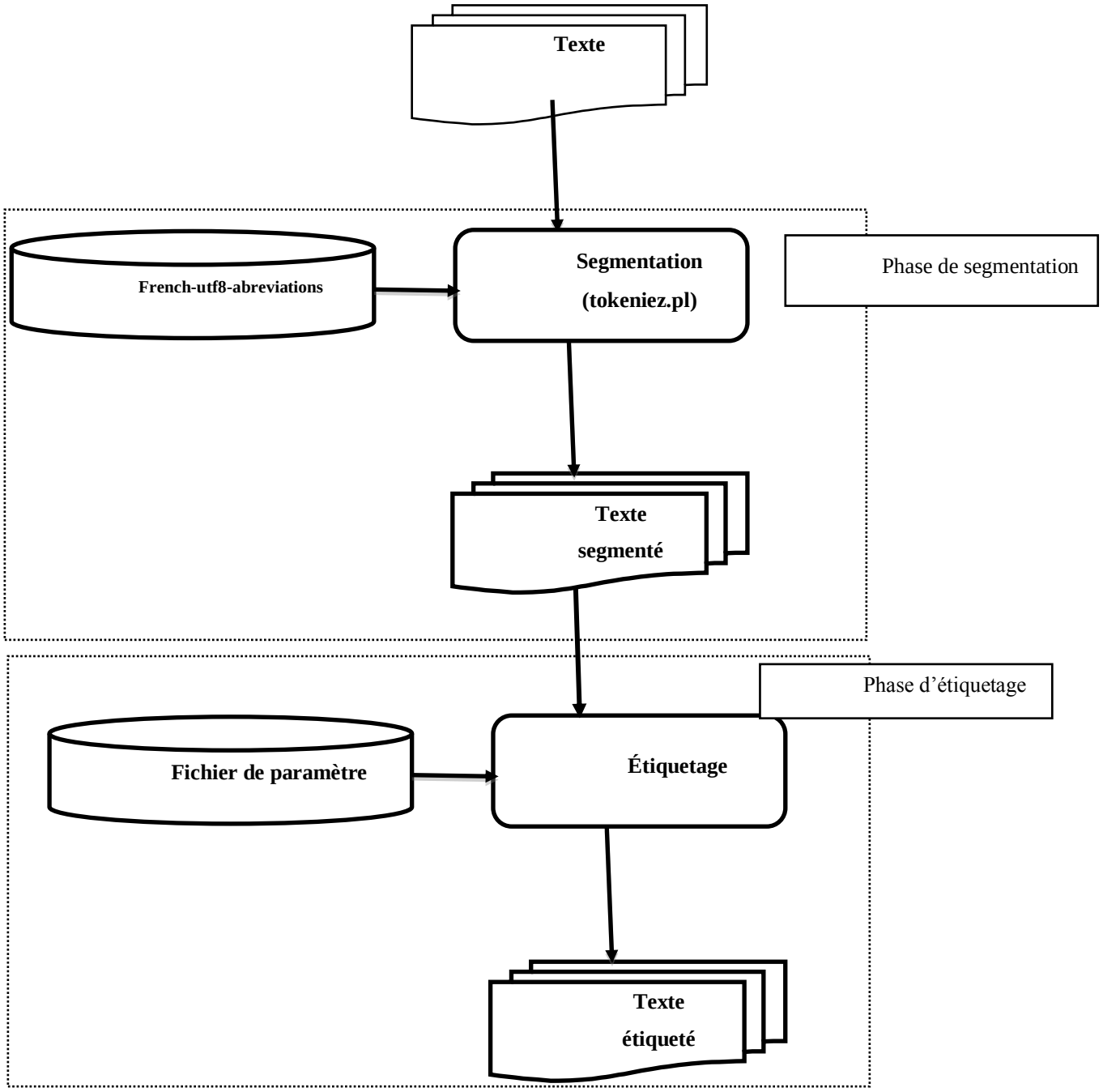

Figure 40 : Phase de l'étiquetage dans TreeTagger.

Pour lancer l'exécution de TreeTagger, il faut lancer la commande suivante :

perl treetagger\cmd\tokenize.pl -f -a treetagger\lib\french-utf8-abbreviations Exemple.txt | treetagger\bin\tree-tagger.exe french.par -token -lemma > Exemple.tt

Avec :

- token : option pour afficher les tokens dans le fichier résultat.
- lemma : option pour afficher les lemmes dans le fichier résultat.

3.2.2 Arbre de décision binaire

L'arbre binaire est un outil d'aide à la décision et la classification dans le cadre de l'apprentissage statistique. Ainsi, le fait que l'arbre soit binaire rend le parcours de l'arbre plus rapide surtout dans le cas des arbres profonds. La génération de l'arbre de décision se fait d'une manière récursive à partir de plusieurs séquences de mots (n-grammes). La formule utilisée dans la phase de la construction de l'arbre de décision :

$$Iq = -p(C+|C)\sum_{t\in T} p(t|C+)\log_2 p(t|C+) - p(C-|C)\sum_{t\in T} p(t|C-)\log_2 p(t|C-)$$

Avec :

C : contexte de nœud courant.

C+ : le contexte C sachant que la condition q est valide.

C- : le contexte C sachant que la condition q est non valide.

P(C+|C) : la probabilité sachant que la condition q est valide.

P(C-|C) : la probabilité sachant que la condition q est non valide.

P (t |C+) : la probabilité de la troisième étiquette si la condition q est valide.

P (t |C-) : la probabilité de la troisième étiquette si la condition q est non valide.

L'estimation de ces probabilités est faite à partir des fréquences en utilisant la méthode « Maximum Likelihood Estimation »[22] (MLE) selon les formules suivantes :

$$P(C+|C) = \frac{f(C+)}{f(C)}$$

$$P(t|C+) = \frac{f(C-)}{f(C)}$$

$$P(C+|C) = \frac{f(t,C+)}{f(C+)}$$

$$P(t|C-) = \frac{f(t,C-)}{f(C-)}$$

Avec :

f(C) : le nombre des n-grammes dans le corpus d'apprentissage.

f(C+) : le nombre des n-grammes validés.

f(C-) : le nombre des n-grammes non validés.

f(t, C+) : le nombre des n-grammes validés et qui possède un troisième tag est égale à t.

[22] http://en.wikipedia.org/wiki/Maximum_likelihood

f(t, C-) : le nombre des n-grammes non validés et qui possède un troisième tag est égale à t.

Après la phase de construction de l'arbre de décision, une étape d'optimisation qui consiste à certaines simplifications, est nécessaire. Par exemple, si un nœud a deux feuilles, et si le gain d'information au niveau du nœud est inférieur à certains seuils, les deux feuilles sont supprimées et le nœud devient une feuille elle-même. Le calcul de gain d'une information est basé sur cette équation :

$$G = f(C)(I0 - Iq)$$

$$I0 = \sum_{t \in T} p(t|C) \log_2 p(t|C)$$

Avec :

I0 : la quantité d'information nécessaire pour enlever l'ambiguïté du nœud courant.

Iq : la quantité d'information nécessaire après que le résultat de la condition q est connu.

Dans le cas de TreeTagger, la détermination de la meilleure séquence d'étiquettes pour une séquence donnée de mots est faite avec l'algorithme de propagation dynamique. De plus, la construction de l'arbre de décision est faite à partir d'un corpus étiqueté manuellement. L'exemple ci-après montre comment TreeTagger utilise l'arbre de décision lors de la phase de désambiguïsation.

Séquence à étiqueter :

La petite souris.

Det :art ADJ ?

- Étiquetage du mot souris (token isolé) :

VER :pres sourire 0.517

NOM souris 0.307

VER :simp sourire 0.152

VER :impe sourire 0.023

- Étiquetage final :

NOM souris 0.77

VER :pres sourire 0.19

VER :simp sourire 0.03

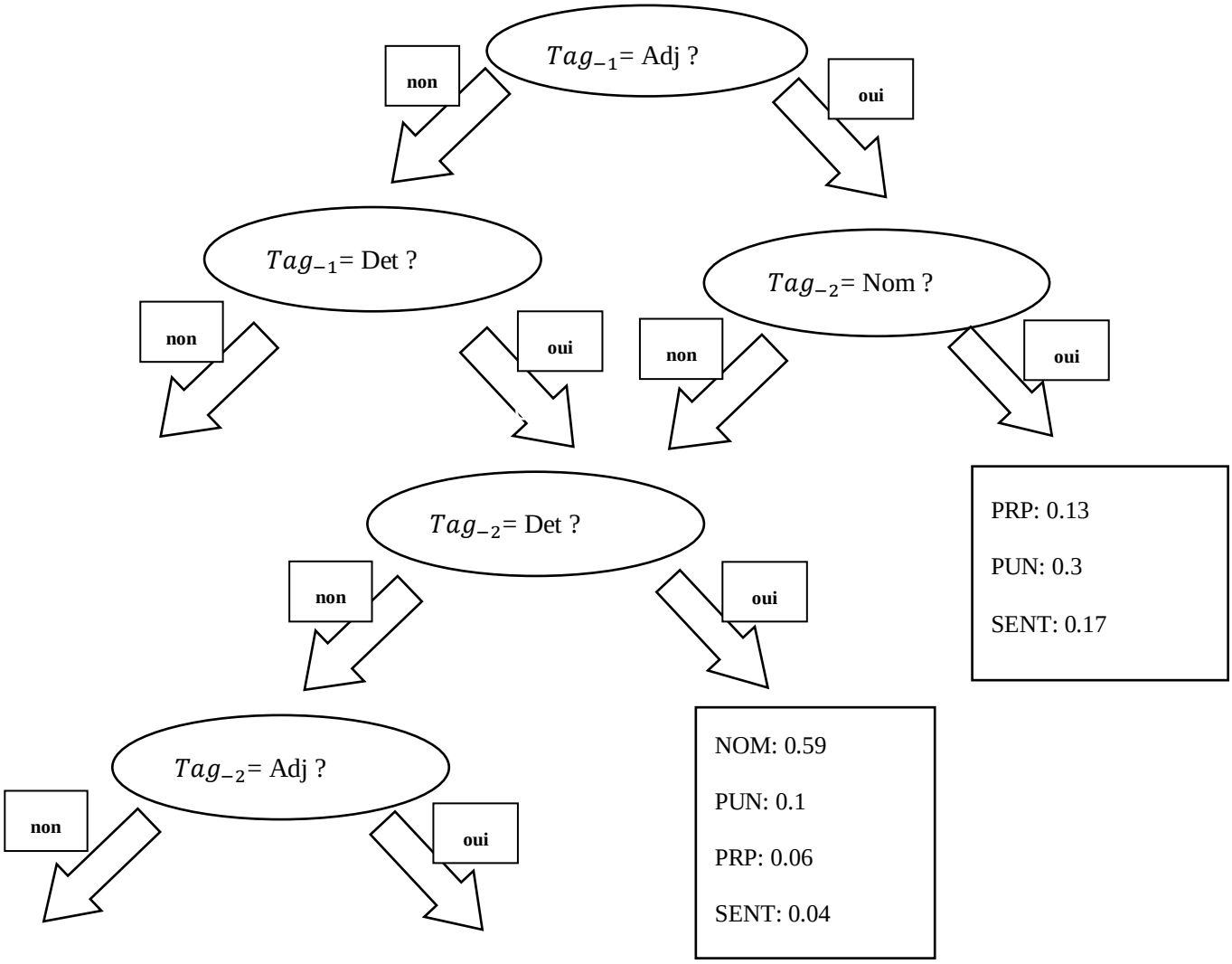

Figure 41 : Arbre de décision établie par TreeTagger pour le mot ambigu souris.

Dans l'exemple ci-dessus, l'analyseur assigne les bonnes étiquettes aux mots : la et petite. Par contre, le mot souris est ambigu car il peut être un verbe (au présent, au passé simple ou à l'impératif) ou bien un nom. Dans ce cas, l'utilisation de l'arbre de décision aide TreeTagger à enlever l'ambiguïté. À partir de l'étiquette n-2 (déterminant) et l'étiquette n-1 (adjectif), le calcul de la probabilité de l'apparition d'une étiquette donnée, sachant les deux étiquettes n-1 et n-2 qui précèdent le mot ambigu sont déterminées, donne ces résultats : Nom : 0.77, VER :pres : 019, VER :simp : 0.03.

Les performances actuelles de TreeTagger (Allauzen et al, 2008), nous permettent de faire nos premiers tests de la plate-forme.

4 Amélioration et développement des outils pour l'Arabe

4.1 Approche

La morphologie de la langue arabe pose des défis particuliers aux systèmes de traitement automatique du langage naturel. Le degré exceptionnel d'ambiguïté dans le lexique arabe, la morphologie riche et le processus de formation des mots à partir de racines rendent la tâche du traitement automatique de langue arabe relativement difficile.

De plus, les ressources pour la langue arabe sont rares, coûteuses (Michael, 2014) et non fiables pour l'utiliser dans l'ALAO. Une solution possible à ce problème est d'améliorer les ressources existantes du TAL pour l'arabe avant de les intégrer dans notre système d'ALAO.

Dans la section suivante, nous présentons les analyseurs morphologiques les plus utilisés dans le TAL.

4.2 Étude des analyseurs morphologiques

Nous trouvons dans quelques recherches (Atwell et al, 2004), (Al-Sughaiyer et al, 2004) un aperçu des outils d'analyse morphologique les plus importants. Par contre, ces recherches ne fournissent pas un rapport détaillé concernant ces outils (performances, efficacités, approches utilisées, etc).

Dans cette section, nous allons effectuer un bref inventaire des analyseurs morphologiques de textes arabes suivants : Aramorph, APT, MorphArab, Sakhr, XEROX, ASVM.

4.2.1 Aramorph

4.2.1.1 Mode de fonctionnement

L'analyseur morphologique de Buckwalter (Buckwalter, 2002) (Buckwalter, 2004), appelé AraMorph, pour l'arabe a été produit par le LDC (Linguistic Data Consortium) en 2002. L'algorithme d'AraMorph pour l'analyse morphologique des textes permet la segmentation d'un mot arabe en trois segments (préfixe, racine, post-fixe). La segmentation se base essentiellement sur trois lexiques et trois tableaux de compatibilité utilisés pour vérifier les combinaisons entre les proclitiques, les racines et les enclitiques :

- Le lexique des préfixes contient 299 entrées.

- Le lexique des suffixes contient 618 entrées.
- Le lexique des racines contient 82 158 racines.
- La table AB pour la vérification de la compatibilité entre les préfixes et les racines (1648 entrées).
- La table BC pour la vérification de la compatibilité entre les racines et les suffixes (1285 entrées).
- La table BC pour la vérification de la compatibilité entre les préfixes et les suffixes (598 entrées).

La longueur d'un préfixe est entre 0 et 4 caractères, la longueur d'un suffixe est entre 0 et 6 caractères.

En plus des encodages standards de la langue arabe, de nombreux chercheurs en traitement des langues naturelles utilisent une translittération orthographique. Par exemple, l'analyseur morphologique AraMorph est basé sur un tableau de translittération[23] fait par Tim Buckwalter (Habash, 2010).

L'algorithme de l'analyse morphologique et l'étiquetage inséré dans le code d'AraMorph permettent la tokenisation, la segmentation des mots, la consultation du dictionnaire, la vérification de la compatibilité et l'affichage du rapport sur le calcul statistique. L'algorithme d'analyse morphologique est contenu dans un script Perl. L'idée de cet algorithme est de donner à chaque token la liste de toutes les annotations considérées comme des solutions possibles, en prenant en compte tous les signes diacritiques. De plus, pour chaque token, l'analyseur permet l'affectation d'une étiquette pouvant correspondre à chaque segment.

4.2.1.2 Limites d'Aramorph

Même si AraMorph est largement utilisé pour l'analyse morphologique de la langue arabe, il a quelques points faibles.

Selon (Atia, 2006) ces faiblesses se résument :

- Aramorph ne gère pas les proclitiques interrogatifs au début du mot (nom ou verbe), par exemple dans le verbe « أأذهب » ou dans le nom « أعلي ».

[23] Translittération de Buckwalter : Translittération utilisée pour représenter les caractères arabes sous forme de caractères latins.

- Aramorph ne traite que certaines formes des verbes à l'impératif (seuls 22 verbes sur 9 198).
- Aramorph ne traite que certaines formes à la voix passive (seuls 1404 verbes sur 9 198).
- Aramorphe utilise un lexique fini des racines et des lexèmes, donc, il ne gère pas la génération des formes fléchies. Par conséquent, le coût de la mise à jour et de la maintenance de lexique pour l'amélioration de résultats sera très élevé.
- Dans l'algorithme d'Aramorphe, il n'y a aucune phase qui traite la désambiguïsation.

4.2.2 L'analyseur APT de Khoja

APT « Arabic Part-of-speech Tagger » est un analyseur morphologique développé par Khoja en 2001 (Khoja, 2001). Il s'agit d'une adaptation de l'analyseur anglais BNC après quelques modifications adaptées aux règles de la grammaire arabe. Cet analyseur emploie une méthode hybride qui combine l'approche statistique et l'approche à base des règles. Ce système utilise 130 étiquettes, essentiellement dérivées du système BNC, et un corpus de 50 000 mots qui sert à l'entraînement du système. Selon Khoja, ce système atteint une précision de 90 %.

Quoique ce système donne une bonne précision sur le corpus de test, il a quelques points de faiblesse dont :

- Les étiquettes affectées manuellement ne contiennent pas toutes les étiquettes possibles qu'un mot peut prendre.
- La taille du corpus d'apprentissage (50 000 mots) n'est pas suffisante pour construire un modèle du langage riche.
- L'outil de la segmentation de la langue arabe ne traite pas le placement de hamza sur le Alif.
- Le système APT n'a pas un module de prétraitement capable de gérer les erreurs qu'on trouve souvent dans les textes arabes. Par exemple, dans les textes journalistiques, les auteurs ont tendance à écrire les mots qui se terminent par yah « ي » en plaçant également deux points sous la lettre par la lettre yah « ى » sans les deux points. À titre d'exemple, le nom Ali « علي » devrait avoir deux points sous la lettre yah, tandis que dans la majorité de textes journalistiques le nom ali s'écrit avec un yah sans les deux points « على ».

4.2.3 L'analyseur MorphArab

MorphArab a été développé en 2002, par l'équipe SILAT à Lyon. Il est programmé avec le langage orienté objet et il a utilisé le corpus DINAR (Abbes, 2004) pour l'entraînement de ses modèles. Le principe de fonctionnement de l'analyseur est simple, il commence par une phase de segmentation de texte en morphème puis il affecte les traits morphosyntaxiques pour chaque mot du texte. Par contre, les performances de cet outil se dégrade lors du traitement des mots inconnus. Le droit d'utilisation de MorphArab n'est pas libre.

4.2.4 L'analyseur Sakhr

Sakhr a été développé en 2004, par Chalabi (Chalabi, 2004). Le plus de cet analyseur, selon sa petite documentation, est sa capacité à couvrir l'arabe moderne[24] et l'arabe classique[25]. L'algorithme de l'analyseur segmente le texte donné en tokens, c'est-à-dire, il segmente le mot en préfixe, racine et suffixe, puis il affecte les traits morphologiques de chaque racine obtenue après la phase de segmentation. La précision de Sakhr est de 90 % sur le corpus du test. Par contre, cet outil n'est pas open source; de même la seule source de documentation pour l'analyseur Sakhr est son site web[26].

4.2.5 Analyseur de XEROX

L'analyseur Xerox « the Xerox Arabic Morphological Analysis and Generation » a été développé par Keen Beesley (Beesley 2001). La première étape de l'algorithme d'analyse morphologique est la segmentation du mot en entrée en unités lexicales à l'aide d'un transducteur à état fini (Farghaly et al, 2003). Puis le système donne à chaque unité lexicale une étiquette qui désigne le trait morphologique et la catégorie de l'unité dans le texte. L'analyseur est basé sur un dictionnaire de 4930 racines et une liste composée de 400 règles qui permettent la génération de 90 000 formes fléchies (lexèmes). Selon Beesly, l'analyseur donne de bons résultats, sauf que l'utilisation de 400 règles engendre beaucoup d'ambiguïtés lexicales, et l'algorithme utilisé par le système ne comporte pas une phase de désambiguïsation.

---

[24] L'arabe littéraire.

[25] L'Arabe du VIIe siècle.

[26] www.sakhr.com

4.2.6 Analyseur ASVM

ASVM a été développé à l'Université de Stanford (Diab et al, 2004). Il contient deux modules : le premier est un segmenteur (il segmente chaque phrase en plusieurs mots), et le deuxième est un étiqueteur morphologique (assigne une étiquette à chaque mot). La technologie d'ASVM est basée sur l'apprentissage supervisé.

ASVM a été utilisé avec succès par plusieurs groupes dans le cadre de la traduction automatique de l'Arabe, en particulier pour l'amélioration de l'alignement et la remise en ordre de la sortie d'un système de traduction (Josep et al, 2008). De plus, ASVM a été utilisé dans la traduction de la parole (Nicolas et al, 2006). On trouve aussi que l'analyseur ASVM a été exploré dans d'autres axes de recherche tels que la recherche d'informations et la reconnaissance d'entités nommées (David et al, 2007).

4.3 Choix de l'analyseur morphologique pour notre plate-forme

Dans le cadre de notre recherche, qui consiste à développer une plate-forme d'ALAO pour la langue arabe, nous avons besoin d'un analyseur morphologique fiable. Ce dernier est indispensable pour générer automatiquement les activités à travers notre plate-forme. Selon (Habash, 2010) l'analyseur le plus utilisé dans la communauté du TAL est ASVM. De plus, il est libre d'utilisation et on peut utiliser le code source de l'outil pour améliorer ses performances.

Dans la section suivante, nous présentons l'analyseur ASVM, ainsi que son mode de fonctionnement.

4.3.1 Architecture d'ASVM

L'analyse morphologique d'ASVM se déroule en deux phases :

a. Segmentation

Elle se concentre principalement sur la segmentation de clitique. Cette segmentation est basée sur la même règle utilisée dans le Penn arabe Treebank (PATB) pour la segmentation des clitiques (Maamouri et al, 2004).

Les clitiques segmentés par ASVM sont les suivants :

- Les conjonctions proclitiques : « و » w et « ف » f.
- Les proclitiques prépositionnels : « ك » k, « ل » l et « ب » b.
- Le proclitique marqueur de futur : « س » s.
- La particule proclitique verbale : « ل » l.

- L'article proclitique défini : « ال » al.
- Les enclitiques pronominaux indiquant les pronoms possessifs / objets : « ه » h, « ي » yah ...

L'entrée de l'analyseur doit être encodée en Buckwalter, qui est une table de correspondance entre les caractères arabes et le code ASCII.

Exemple d'une phrase encodée en Buckwalter :

ولم يحتسب الحكم المجري ساندرو بول ركلة جزاء صحيحة اثر عرقلة داخل المنطقة من قبل اليساندرو

Sortie Buckwalter :

« wlm yHtsb AlHkm Almjry sAndwr bwl rklp jzA' SHyHp Avr Erqlp dAxl AlmnTqp mn qbl AlysAndrw. »

b. Étiquetage morphosyntaxique :

C'est le problème de l'attribution d'une étiquette à chaque mot tel que le verbe, le nom, l'adjectif, etc. Pour ce faire, le module d'étiquetage d'ASVM prend en entrée un texte segmenté. L'étiquetage dans ASVM se fait à travers une approche de classification basée sur les machines à support de vecteurs (SVM). La sortie de l'analyseur peut être présentée sous forme d'un texte segmenté et étiqueté, ou sous la forme d'un texte non segmenté et l'étiquette est attribuée aux mots complets (mot non segmenté en clitique et racine). Les étiquettes utilisées par ASVM sont données dans le tableau suivant.

Étiquette	Catégorie	Étiquette	Catégorie
CC	Conjonction	PRP$	Pronom possessif
PRP	Pronom	NO_FUNC	inconnu
NNP	Nom propre singulier	NNPS	Nom propre pluriel
NN	Nom singulier	WP	Particule interrogative
NNS	Nom pluriel	WRB	Adverbe interrogatif
JJ	Adjectif	VBP	Verbe inaccompli
IN	Proposition	VBN	Verbe passif
FW	Mot étranger	VBD	Verbe à l'imparfait

DT	Déterminant	VB	Verbe à l'impératif
RP	Particule	UH	Interjection
CD	Chiffre	RB	Adverbe

Tableau 19 : les étiquettes utilisées par ASVM

Exemple de sortie d'ASVM, prenant comme exemple la phrase précédente :

ولم يحتسب الحكم المجري ساندرو بول ركلة جزاء صحيحة اثر عرقلة داخل المنطقة من قبل اليساندرو

Dans le texte étiqueté, chaque unité lexicale est suivie d'un slash et de sa catégorie.

w/CC lm/RP yHtsb/VBP Al/DT Hkm/NN Al/DT mjry/JJ sAndwr/NNP bwl/NNP rklp/NN jzA'/NN SHyHp/JJ Avr/IN Erqlp/NN hyskY/NNP dAxl/IN Al/DT mnTqp/NN mn/IN qbl/NN Al/DT lysAndrw/NNP ./PUNC

4.3.2 Ressources utilisées par ASVM

ASVM utilise l'approche SVM de classification supervisée, d'où la nécessité des ressources annotées pour la phase d'apprentissage. ASVM utilise trois corpus (ATB1, ATB2, ATB3) issus d'Arabic Tree Bank (ATB). Ces corpus sont constitués par des articles journalistiques[27] : le journal Al-Hayat distribué par Oumma et le journal An Nahar News, Agence France-Presse et Xinhua News Agency. Les thèmes des articles sont principalement constitués de textes politiques, économiques et sportifs.

Corpus	Nombre des articles	Taille (mots)
ATB1	734	140K
ATB2	501	140K
ATB3	600	340K

Tableau 20 : corpus utilisés par ASVM

Chacun de trois corpus ATB est divisé en 10 % des données de développement, 80 % des données d'entraînement et 10 % des données de test.

[27] https://catalog.ldc.upenn.edu/docs/LDC2005T20/

Corpus	Mots	Phrases
Développement	70188	2304
Entraînement	594683	18970
Test	69665	2337

Tableau 21 : Taille des données utilisées pendant l'expérience

4.3.3 Résultats

L'expérience d'étiquetage doit être lancée sur un texte segmenté et lemmatisé. Selon (Diab et al, 2004) les résultats obtenus avec l'étiqueteur ASVM sont comparés à ceux qui sont obtenus avec une approche à base de règles. Ce dernier assigne l'étiquette la plus fréquente à chaque mot du texte en se basant sur les données d'entraînement. Par contre, si le mot n'existe pas dans les données d'entraînement, le système à base des règles assigne l'étiquette NN comme une étiquette par défaut.

Étiqueteur	Précision
Système à base des règles	92,2
ASVM	96,6

Tableau 22 : Résultat d'étiquetage d'ASVM comparé à un système à base des règles

4.3.4 Points faibles d'ASVM

L'ASVM donne de bons résultats pour les textes arabes. Par contre, dans le développement des outils TAL pour l'ALAO, il faut que l'outil soit fiable et la précision doit être élevée. Même si ASVM est largement utilisé par la communauté scientifique pour l'analyse morphologique de la langue arabe (traduction automatique, recherche d'informations, identification de la langue…), il a quelques points faibles :

- Dans la phase de segmentation, l'article « ال » AL reste toujours attaché au mot. En effet, l'article « ال » AL est considéré comme un clitique dans la langue arabe, il s'écrit attaché au mot. Par conséquent, la présence de l'article « ال » AL attaché au mot après la phase de la segmentation affecte les calculs statistiques et va créer une fausse distinction entre un mot déterminé, c'est-à-dire, il est précédé par l'article

« AL » et le même mot non déterminé, tel est le cas du mot « البيت », qui doit être segmenté en clitique « ال » AL et en racine « بيت ». Or ASVM traite le mot « البيت » comme un mot différent de sa racine « بيت ».

- La lettre « ة » t qui représente le marqueur de féminin est confondu avec la lettre « ت » t lorsqu'un suffixe enclitique est ajouté à un mot. Par exemple, si on ajoute le suffixe « ها » au mot féminin « محبة » mahabat, il devient « محبتها » mahabatoha. Dans ce cas, ASVM considère la lettre t comme une lettre appartenant au mot et non une terminaison féminine.
- L'analyseur ASVM confond la lettre « ى » ay et la lettre « ي » ya. Cette distinction est une distinction lexicale et elle concerne la racine du mot.

Afin de tester avec plus de précision la fiabilité de l'analyseur ASVM, nous avons intégré l'étiqueteur dans notre plate-forme pour analyser morphologiquement les textes arabes. Nous avons remarqué que l'analyseur se trompe souvent sur les noms propres, les adjectifs, les noms, les mots qu'on ne trouve pas dans le corpus d'apprentissage.

Pour résoudre ces faiblesses d'ASVM, nous avons mis à jour les outils de segmentation. De plus, nous avons relancé l'apprentissage de l'étiqueteur sur un corpus plus riche en thèmes et en morphologie.

4.4 Amélioration d'ASVM et construction d'un nouveau corpus arabe

4.4.1 Mise à jour du segmenteur d'ASVM

4.4.1.1 Développement

Avant d'intégrer l'analyseur morphologique ASVM dans notre plate-forme nous avons décidé d'améliorer les performances de l'analyseur ASVM en corrigeant les erreurs faites par son segmenteur.

Pour corriger ces faiblesses, nous avons développé un outil sous la forme d'un script Perl qui normalise le texte à analyser avant la phase de segmentation et corrige la sortie de segmenteur d'ASVM. D'abord, avant de lancer ASVM l'outil normalise les deux lettres « ى » ay et « ي » ya (la seule différence entre ces deux lettres est la présence de deux points sous la lettre ya). Cette étape consiste à vérifier dans un dictionnaire qui contient 392 000 mots arabes (la description de la construction de ce dictionnaire est décrite dans la section 4) à l'aide d'une

expression régulière si les mots dans le texte sont écrits avec la lettre « ى » ou la lettre « ي ». Si le mot est trouvé dans le dictionnaire, mais avec la mauvaise lettre alors l'outil remplace le mot du texte par celui trouvé dans le dictionnaire. Par exemple, le mot « إلى » ILA écrit avec la lettre « ى » sans les deux points au-dessous de la lettre sera remplacé après la vérification dans le dictionnaire par le mot « إلي ». Si l'outil trouve le mot dans le dictionnaire avec les deux variantes (c'est-à-dire ce mot peut être présenté sous les deux formes avec la lettre « ى » et la lettre « ي » comme le mot « على » sous et le mot « علي » Ali), il ne modifie pas le mot et passe aux mots suivants. Dès que l'étape de normalisation est effectuée, l'outil lance le module de la segmentation d'ASVM. Ensuite, notre outil corrige les erreurs liées à la mauvaise segmentation de l'article « ال » AL. L'outil identifie tous les mots qui commencent par l'article « ال ». Par la suite, il cherche dans le dictionnaire si le mot courant existe sans l'article « AL », si oui, alors l'outil segmente le mot au niveau de l'article en deux (par exemple, le mot « البيت » devient « ال » + « بيت »). Sinon, l'outil ne modifie pas le mot et passe aux mots suivants. Enfin, l'outil corrige la confusion faite par le segmenteur d'ASVM sur le marqueur de féminin. L'outil vérifie si le mot segmenté peut avoir une forme masculine, c'est-à-dire le mot existe dans le dictionnaire avec les deux formes (masculin et féminin). Dans ce dernier cas, l'outil segmente le mot en deux (mot + marqueur de féminin « ة »). Par exemple, le mot « مجانية » gratuite est féminin, ce dernier est présent dans le dictionnaire sous les deux formes masculin et féminin, alors l'outil segmente ce mot en deux : « مجاني » + « ة ». Dans la suite de ce chapitre, nous allons appeler l'étiqueteur ASVM initial par ASVM1 et l'étiqueteur ASVM avec la correction de module de segmentation par ASVM2.

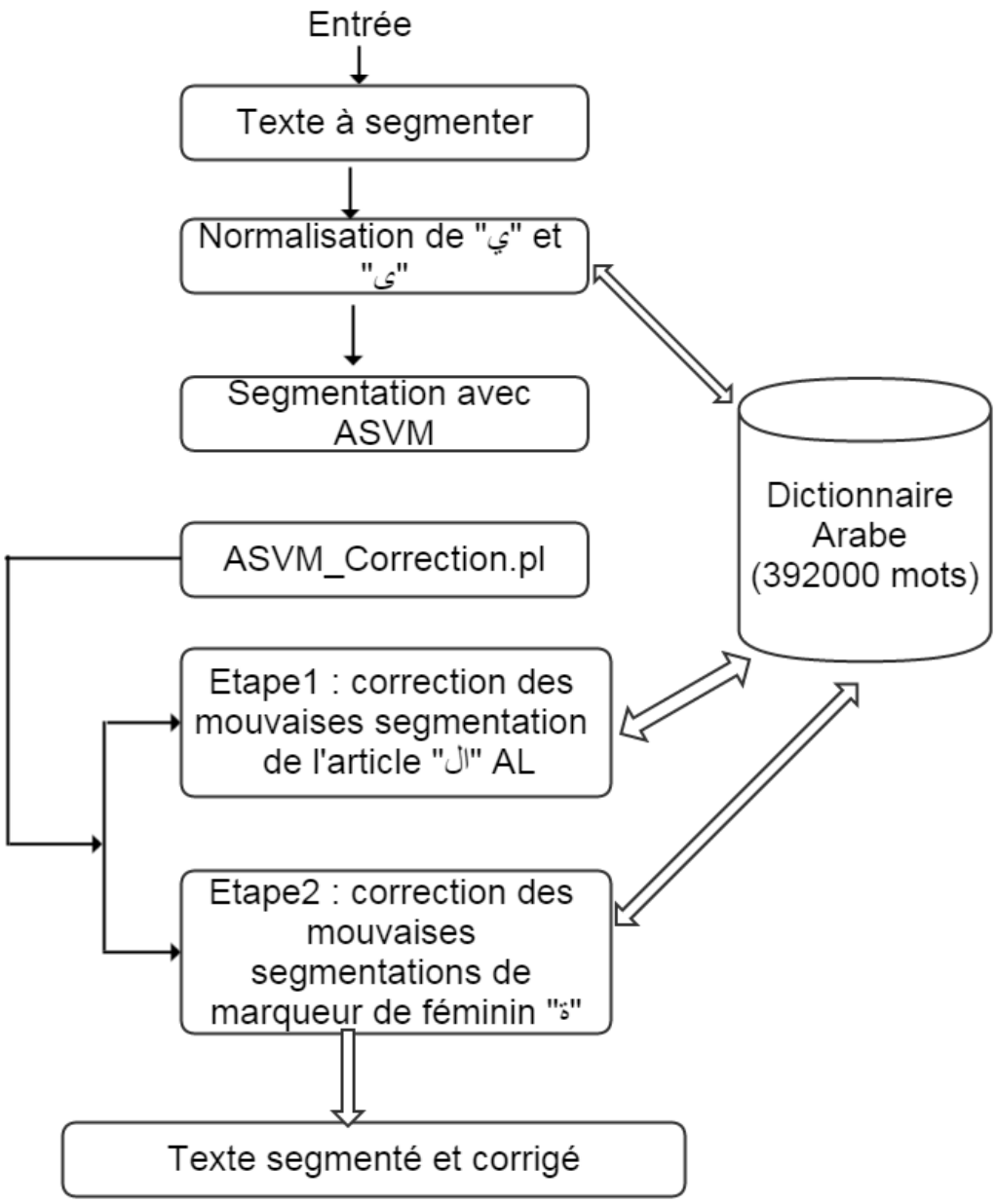

Figure 42 : Nouveau principe du module de segmentation d'ASVM après la correction

4.4.1.2 Construction d'un corpus arabe pour l'évaluation

Afin de vérifier la qualité de notre outil, nous avons besoin d'un corpus arabe segmenté et annoté. Comme les ressources arabes sont rares est parfois inexistants, nous n'avons pas trouvé de corpus fiable et libre pour la langue arabe. Les corpus trouvés sont soit étiquetés automatiquement soit non segmentés. Dans le cadre de notre thèse, nous avons décidé de construire notre propre corpus manuellement. Pour ce faire, nous avons adopté une approche rapide en utilisant le Web, l'analyseur morphosyntaxique ASVM et quatre linguistes arabes. Actuellement, l'utilisation du Web pour la construction rapide des corpus dans

différents domaines et différentes langues devient de plus en plus fréquente (Meftouh et al, 2007). Après quelques recherches et analyses des contenus des sites web journalistiques arabes tels que Aljazeera, Alhayet, Alwatan, nous avons choisi le site d'Aljazeera vu la qualité de ses textes ainsi que la richesse de son contenu (politique, économique, tourisme, sport, etc).

La procédure de construction du corpus se déroule en trois phases : construction du corpus à partir du Web, prétraitement automatique, correction manuelle.

a. Construction du corpus à partir du Web

La première tâche est d'acquérir des données textuelles dans la langue arabe à partir du site Aljazeera. Pour cela, nous avons développé un outil en Perl qui aspire les pages html à partir d'une adresse web donnée. Pour assurer la richesse de notre corpus en vocabulaire, nous avons choisi six thèmes : politique, religion, sport, tourisme, science et technologie et santé. Après la phase d'aspiration des pages web souhaitées, le corpus obtenu ne peut pas être exploitable linguistiquement. Donc nous avons développé un deuxième outil pour la conversion des fichiers HTML en texte brut. La partie cruciale dans cette procédure de construction du corpus est le nettoyage des fichiers des textes obtenus, c'est-à-dire isoler et extraire les informations utiles à inclure dans notre corpus. Nous voulons supprimer les informations non utiles (menu, des liens vers d'autres pages, graphiques, images, les styles, les scripts). Le tableau suivant donne la taille du contenu des corpus par thème :

Thème	Nombre de phrases	Nombre de mots
Politique	2321	6902
Religion	1751	19243
Sport	3742	22538
Tourisme	4611	27223
Science et technologie	4367	26751
Santé	5226	28279
Total	19972	130936

Tableau 23 : Composition des corpus de test

b. Prétraitement automatique

Pour simplifier la tâche de l'étiquetage manuelle aux annotateurs, nous avons effectué un premier étiquetage automatique à l'aide de l'outil ASVM2. La première étape consiste à convertir le corpus obtenu en format Buckwalter (Buckwalter, 2002) car ASVM ne traite que des textes translittérés en Buckwalter (voir en annexe la table de Buckwalter). Pour cela, nous avons développé un outil de translittération en Perl qui prend un texte arabe encodé en utf-8 et rend en sortie un texte translittéré. À la fin, nous avons lancé ASVM2 sur notre corpus pour obtenir un texte segmenté et étiqueté.

Notons que le jeu d'étiquette pour l'annotation automatique de ce corpus est le même que celui utilisé par l'analyseur ASVM.

c. Correction manuelle

Comme ce corpus a été créé pour évaluer l'amélioration de module de la segmentation d'ASVM, alors le corpus doit être fiable et ne pas contenir d'erreurs de segmentation ou d'étiquetage. La meilleure solution pour corriger notre corpus est de passer par une phase de vérification manuelle à l'aide des linguistes natifs[28]. D'abord, nous avons divisé le corpus en quatre parties en conservant la même taille. Ensuite, la phase d'annotation manuelle se déroule en trois étapes :

- Dans la première étape, chaque annotateur vérifie si la pré-segmentation d'ASVM est correcte ou non. Dans le cas où la segmentation est fausse, l'annotateur doit corriger les erreurs.
- De même, dans cette étape, chaque annotateur vérifie si l'étiquetage par ASVM est correct ou non. Dans le cas où l'étiquetage est faux, l'annotateur doit corriger les erreurs.
- Dans cette étape, l'annotateur met un astérisque s'il a le moindre doute sur la faute (au niveau de la segmentation ou bien de l'étiquetage).

Enfin, les annotateurs valident ensemble après une discussion entre eux les fautes où il y a des astérisques, sachant que le prétraitement du corpus a diminué le temps dédié à la segmentation et à l'annotation manuelle du corpus.

[28] Les linguistes sont des étudiants des langues en Master à l'université Stendhal. Ils sont d'origine tunisiens.

4.4.1.3 Évaluation

Après la construction du corpus fiable pour la langue arabe, nous pouvons maintenant évaluer la qualité de l'analyseur ASVM après l'amélioration de son module de segmentation. La meilleure façon d'évaluer la performance de deux modules (segmentation et étiquetage) est de calculer le taux de la segmentation correcte et l'annotation correcte sur le corpus d'évaluation ainsi que la précision et le rappel. Pour cela, nous avons développé un script Perl que prend en entrée le corpus segmenté et étiqueté automatiquement ainsi que le corpus étiqueté manuellement. À la fin, le script affiche en sortie le taux, la précision et le rappel selon les formules suivantes :

- Le rappel mesure la proportion de segmentations correctes faites par ASVM2 par rapport au total de segmentations correctes.
- La précision mesure la proportion de segmentations correctes faites par ASVM2 par rapport au total de segmentations possibles.

Le tableau suivant montre les résultats de la segmentation du corpus d'évaluation.

Système	Taux	Précision	rappel
ASVM1	96.7	83.2	86.6
ASVM2	99.8	97.5	99.3

Tableau 24 : Le nouveau résultat obtenu par ASVM2 par rapport à ASVM1 (en %)

Ce tableau montre une augmentation significative de la performance du module de la segmentation. De plus, nous devons vérifier si notre modification n'a pas affecté le module d'étiquetage d'ASVM. Le tableau suivant montre les résultats de l'étiquetage de notre corpus d'évaluation par ASVM1 et ASVM2.

Système	Taux
ASVM1	90.3
ASVM2	98.8

Tableau 25 : Résultat d'étiquetage obtenu par ASVM1 et ASVM2 (en %)

La performance de l'étiqueteur ASVM2 est nettement supérieur à celle de ASVM1. Par conséquent, les améliorations du module de segmentation d'ASVM ont augmenté la

performance de l'étiquetage et rendent l'analyseur plus utile dans notre plate-forme l'ALAO. Par contre, il reste encore un faible taux d'erreurs dans les résultats d'étiquetage. Ce problème reste un défi majeur pour les différents outils du traitement automatique de la langue.

5 Construction de lexique

5.1 Lexique français

Pour le module de correction orthographique, la plate-forme a besoin d'un lexique propre, fiable et complet (c'est-à-dire d'une large couverture). Ce lexique doit être dans les deux langues : le français et l'arabe. Pour ce faire, nous avons procédé à une méthode simple, tout d'abord, recueillir des lexiques de mots à partir du Web, LDC[29] (Linguistic Data Consortium) ou de ressources utilisées par des logiciels comme ispell, aspell, etc. Certaines listes de mots sont déjà propres : un mot par ligne. Pour l'autre, un processus spécifique doit être appliqué afin d'en extraire une liste de mots propres. Pour cela, nous avons développé des outils en Perl pour le nettoyage, l'extraction de lexiques, etc.

a. Lexique Aspell-fr

Aspell[30] est un dictionnaire multilingue (français, anglais, allemand, etc). Il est fiable et le lexique est bien correct. Il est utilisé comme correcteur graphique par plusieurs logiciels (openoffice, ubuntu, libreoffice etc).

Le lexique est stocké dans un format binaire inaccessible. Pour cela, nous avons développé un script perl qui nous permet d'extraire ce lexique. Ce dernier prend en entrée un fichier binaire et donne en sortie le lexique aspell. À la fin de ce processus, nous avons obtenu un lexique fiable de 400 000 mots.

b. Lexique Ispell

Ispell[31] est un dictionnaire multilingue, dont le français, il est utilisé par plusieurs logiciels comme correcteur orthographique (emacs, fedora, etc). De même, pour extraire le lexique d'Ispell, nous avons développé un script perl. À la fin de l'exécution de ce script, nous avons obtenu un lexique d'environ 300 000 mots.

[29] http://en.wikipedia.org/wiki/Linguistic_Data_Consortium

[30] ftp://ftp.gnu.org/gnu/aspell/dict/0index.html

[31] http://fmg-www.cs.ucla.edu/geoff/ispell-dictionaries.html

c. Lexique Lefff

Le lexique lefff[32] (Lexique des formes fléchies du français) développé par (Sagot, 2010) dans le labo d'INRIA. Il contient environ 400 000 mots français. Ce lexique est open source, et il est disponible sous format texte (mot par ligne avec des informations sur le mot : genre, nombre, etc).

Après la collecte de ces informations, il ne reste qu'à les mettre dans le même dictionnaire. Pour ce faire, nous avons développé un script pour concaténer les différents lexiques dédoublonnés. À la fin, nous avons obtenu environ 700 000 mots français.

5.2 Lexique arabe

Les ressources linguistiques pour la langue arabe sont coûteuses et parfois non fiables, tel est le cas de lexique ou dictionnaire de mots arabes. Après de nombreuses recherches sur des lexiques arabes sur le Web, nous n'avons trouvé que des lexiques payants ou très petits en terme de couverture. Pour cela, nous avons adapté notre propre approche, qui consiste à construire manuellement un lexique à partir des corpus arabes fiables. Nous avons choisi deux ressources pour notre expérience : corpus de Quran et le site arabe de Wikipédia.

a. Construction du lexique à partir de corpus Quran

D'abord, nous avons téléchargé le corpus Quran[33] qui contient 150 000 mots. Sachant que les mots utilisés dans ce corpus sont bien propres et corrects. Ensuite, nous avons développé un script Perl pour le nettoyage du corpus et l'avons segmenté en mot par ligne. Enfin, nous avons obtenu un dictionnaire d'environ 85 000 mots.

b. Construction du lexique à partir de Wikipédia

Dans cette étape, nous avons utilisé le même principe pour la construction du corpus d'amélioration d'ASVM à partir du web (section 1.4.1.2). À partir de site web de Wikipédia[34], nous avons aspiré les pages web en HTML en utilisant le même outil utilisé dans la section ci-dessus. Ensuite, nous avons converti les pages web aspirées en texte brut en suivant le même

[32] http://www.labri.fr/perso/clement/lefff/

[33] www.kaheel7.com/book/quran-arabic-free-download.doc

[34] http://ar.wikipedia.org/

protocole de nettoyage des données textuelles. Enfin, nous avons obtenu un corpus en texte brut d'environ 24 906 692 mots.

Après l'obtention d'un corpus propre et volumineux, nous avons appliqué les mêmes procédures que celles de la section précédente pour obtenir un mot par ligne dans le lexique obtenu. À la fin de l'expérience, nous avons obtenu un lexique de 1 000 000 mots.

Maintenant, nous avons deux lexiques (celui obtenu du corpus Quran et celui du corpus Wikipédia), il suffit de les concaténer ensemble et d'enlever les doublons pour obtenir notre lexique final. En conclusion, le lexique arabe obtenu est d'environ 650 000 mots.

6 Conclusion

Dans ce chapitre, nous avons introduit l'état de l'art des analyseurs morphologiques pour la langue arabe et la langue française, en justifiant nos choix pour les deux langues. Ensuite, nous avons montré les étapes des améliorations effectuées sur l'analyseur ASVM pour le rendre plus efficace avant de l'intégrer dans notre plate-forme. Enfin, nous avons exposé notre méthode de construction de lexiques propres pour le français et l'arabe.

Chapitre 8 : Architecture du système d'ALAO réalisé

1 Introduction

Actuellement, un système d'ALAO doit contribuer à l'évolution et l'innovation des méthodes d'enseignement. D'autre part, un système d'ALAO devrait être intelligent pour expliquer les différentes erreurs commises et donner le feedback adéquat à l'apprenant. Cela augmente les possibilités d'utiliser ces systèmes dans diverses situations d'apprentissage et les rendre plus efficaces et plus autonomes.

Dans ce chapitre, nous présentons l'architecture complète de notre plate-forme pour le français et l'arabe en utilisant des outils de traitement automatique de la langue. Dans un premier temps, nous décrivons l'architecture générale de notre plate-forme. Ensuite, nous présentons l'intégration des différents modules développés (la reconnaissance de l'écriture, l'indexation pédagogique, la génération de feedback, etc.). Enfin, nous concluons ce chapitre par une évaluation de la plate-forme réalisée.

2 L'architecture du système d'ALAO

2.1 Architecture proposée

Nous avons utilisé cette architecture pour l'arabe et le français. Elle est composée de plusieurs modules : l'interface utilisateur (enseignant et apprenant), le générateur des activités, l'analyseur morphologique, le système de la reconnaissance de l'écriture, le module de feedback et l'indexation pédagogique de texte (figure 47). L'interface de la plate-forme présente un moyen d'interaction entre l'apprenant et l'enseignant et entre l'apprenant et le système. L'interface utilisateur fournit aussi les moyens de communication entre l'utilisateur et le système d'ALAO. Elle permet aux enseignants de créer des activités pédagogiques et les rendre accessible à l'apprenant. Notre plate-forme permet de créer plusieurs types d'activités (exercices à trous, question choix multiple, conjugaison, apprendre l'écriture, etc.). Ces activités sont créées à partir d'un texte qui existe déjà dans nos bases de données, ou à partir d'un texte apporté par l'enseignant. Dans ce dernier cas, le texte est analysé morphologiquement.

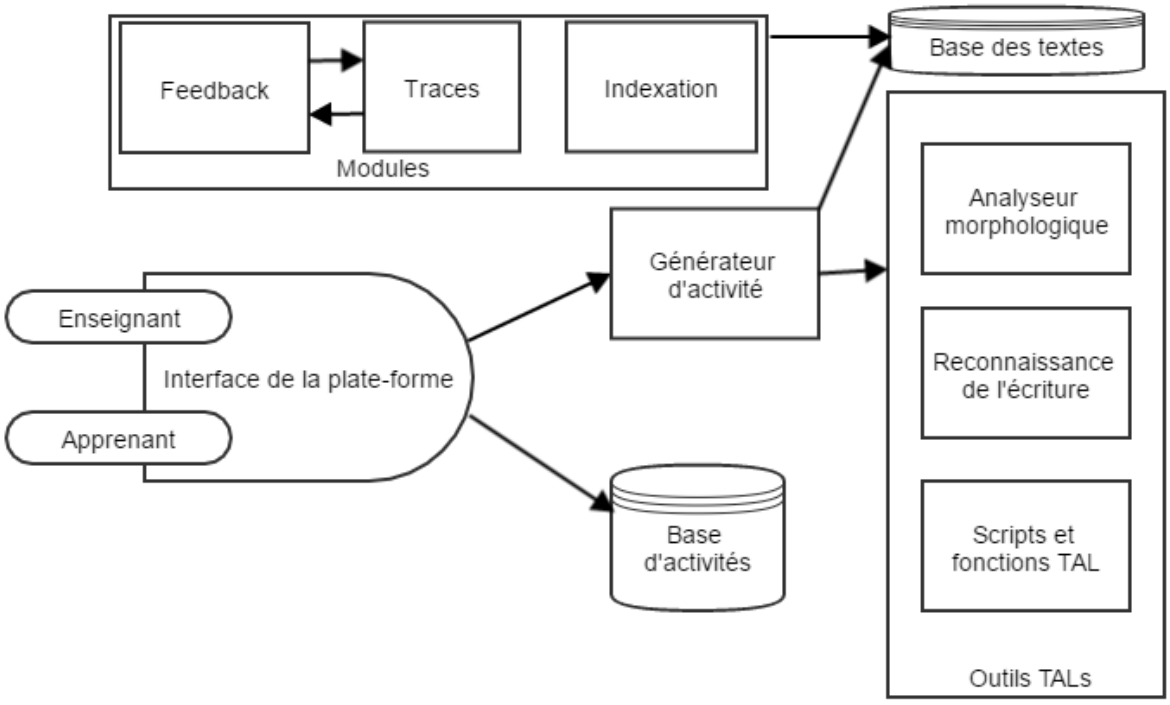

Figure 43 : Architecture globale de la plate-forme

2.2 Modes d'accès

Au niveau de l'architecture externe de notre plate-forme d'ALAO, nous avons créé trois types d'utilisateurs en se basant sur les droits d'accès de chaque utilisateur. Par conséquent, on peut accéder à la plate-forme en tant qu'administrateur, enseignant ou apprenant.

- Mode administrateur : les spécialistes de la langue (les experts didactiques, des linguistes et des enseignants). Le mode administrateur gère les inscriptions des étudiants sur le système et limite l'accès aux ressources pour les apprenants, en ne leur laissant que l'accès vers les exercices proposés par les enseignants. L'administrateur est la seule personne qui peut modifier (ajout, mise à jour ou suppression) les scripts et les outils TAL.
- Mode enseignant : il est destiné principalement aux enseignants, il donne plus des droits aux enseignants afin d'accéder aux scripts et aux générateurs des activités. De plus, ce mode permet à l'enseignant de créer des activités (grammaire, orthographe, conjugaisons, QCM …) à travers une interface simple à maîtriser. L'interface ne nécessite pas des compétences en informatique.
- Mode apprenant : ce mode permet à l'apprenant de retrouver les activités publiques de leurs enseignants, de faire des activités proposées par son enseignant directement sur la plate-forme. Une fois que l'apprenant a travaillé une activité, elle sera sauvegardée dans la base de données et elle sera accessible à l'enseignant afin de l'évaluer si nécessaire et suivre la progression des apprenants.

3 Génération des activités basées sur l'analyse morphologique

Notre système permet la génération des activités pédagogiques grâce à une interface orientée utilisateur, il n'est donc pas nécessaire de recourir à une chaîne de traitement complexe, ni de faire appel à des spécialistes d'informatique ou du TAL. Aussi, les outils TAL exploités par notre système d'ALAO doivent rester invisibles, ce qui donne à notre système une transparence qui permettra aux non-spécialistes de bénéficier des résultats de ces outils.

3.1 Jeu d'étiquettes morphosyntaxiques

Avant de commencer la phase de la conception et la génération des activités, nous devons préciser le jeu d'étiquette que l'on va utiliser dans notre plate-forme lors de la création de l'activité. Ce travail présente une étape cruciale dans la conception de notre plate-forme, car toutes les activités basées sur l'analyseur morphologique dépendent de ce choix. De plus, ce jeu est très important (Sanchez et al, 1995) pour la diversité des activités proposées dans notre système d'ALAO.

3.1.1 Définition de jeu d'étiquette

L'objectif de notre travail est de construire deux ensembles d'étiquettes, catégories grammaticales principales et sous-catégories. Au début, nous avons défini 13 étiquettes principales comme les montre le tableau suivant :

Catégories	Correspondant	Catégories	Correspondant
adj	Adjectif	prep	Préposition
adv	Adverbe	det	Déterminant
pron	Pronom	pc	pronom sujet
noun	nom	pro	pronom indéfini
verb	verbe	neg	Négation
int	Interjection	conj	conjonction
PUN	Ponctuation		

Tableau 26 : Étiquettes de la plate-forme.

En second lieu, nous avons attribué des sous-catégories aux catégories principales suivantes : adjectif, déterminant, verbe, nom et pronom.

- Adjectif : contient deux sous-catégories, le genre (masculin, féminin ou non déterminé) et le nombre (singulier, pluriel).
- Verbe : contient quatre sous-catégories, le temps (indicatif présent, indicatif imparfait, futur simple et passé simple), le mode (indicatif, subjonctif, gérondif, conditionnel et

infinitif), groupe (1er groupe, 2ème groupe et 3ème groupe) et personne (1ère personne singulier, 2ème personne singulier, 3èmme personne singulier, 1ère personne pluriel, 2èmme personne pluriel et 3èmme personne pluriel).

- Déterminant : contient trois sous-catégories, le genre (masculin, féminin ou non déterminé), le nombre (singulier, pluriel) et le type (article défini, article indéfini, adjectif possessif, adjectif démonstratif, adjectif interrogatif et adjectif numéral).
- Nom : contient deux sous-catégories, le genre (masculin, féminin ou non déterminé) et le nombre (singulier, pluriel).
- Pronom : contient quatre sous-catégories, le genre (masculin, féminin ou non déterminé), le nombre (singulier, pluriel), la personne (1ère personne singulier, 2èmme personne singulier, 3èmme personne singulier, 1ère personne pluriel, 2èmme personne pluriel et 3èmme personne pluriel) et le type (pronom possessif, pronom démonstratif, pronom interrogatif, pronom relatif, pronom personnel et pronom indéfini).

Nous avons intégré dans les bases de données de notre plate-forme les étiquettes les plus utilisées dans les activités d'apprentissage des langues. Sachant que nous pouvons toujours ajouter d'autres étiquettes à la base.

3.1.2 Adaptation des étiquettes de TreeTagger

Dans le cas du français, nous avons utilisé l'analyseur morphologique TreeTagger. Ce dernier utilise un jeu de 33 étiquettes (voir le tableau ci-après).

Étiquettes	Correspondant	Étiquettes	Correspondant
ABR	Abréviation	PRP	Préposition
ADJ	Adjectif	PRP:det	Préposition + Article
ADV	Adverbe	PUN	Ponctuation
DET:ART	Article	PUN:cit	Ponctuation de citation
DET:POS	Pronom Possessif	SENT	Balise de phrase
INT	Interjection	SYM	Symbole
KON	Conjonction	VER:cond	Verbe au conditionnel

NAM	Nom Propre	VER:futu	Verbe au futur
NOM	NOM	VER:impe	Verbe à l'impératif
NUM	Numéral	VER:impf	Verbe à l'imparfait
PRO	Pronom	VER:infi	Verbe à infinitif
PRO:DEM	Pronom Démonstratif	VER:pper	Verbe au participe passé
PRO:IND	Pronom Indéfini	VER:ppre	Verbe au participe présent
PRO:PER	Pronom Personnel	VER:pres	Verbe au présent
PRO:POS	Pronom Possessif	VER:simp	Verbe au passé simple
PRO:REL	Pronom Relatif	VER:subi	Verbe à l'imparfait du subjunctif
VER:subp	Verbe au présent du subjonctif		

Tableau 27 : Les étiquettes de TreeTagger.

Comme nous avons des étiquettes plus riches que celle de TreeTagger, nous ne sommes pas obligés de réduire ou de grouper les étiquettes de l'analyseur. Par contre, il manque beaucoup d'informations de base telles que le genre, le nombre, le mode, etc. Pour cela, nous avons trouvé un outil efficace qui permet de prendre la sortie de l'analyseur TreeTagger et il la convertit en une sortie XML plus riche.

Cet outil est appelé Flemm[35] (Namer, 2000), il est composé par un ensemble de modules PERL. Il permet l'analyse flexionnelle des corpus étiquetés seulement par TreeTagger ou par Brill Tagger. C'est un outil à base des règles, dans le cas des mots ambigus, il utilise un lexique de mots pour la désambiguïsation. Sinon, Flemm calcule le lemme de chaque mot en se basant sur son étiquette déjà attribuée précédemment par TreeTagger ou Brill Tagger, et attribue les principaux traits morphologiques de chaque mot :

- Pour les adjectifs, il attribue le genre et le nombre.
- Pour les déterminants, il donne le genre, le nombre et le type s'il n'est pas déjà donné par TreeTagger.

[35] http://www.cnrtl.fr/outils/flemm/

- Pour les pronoms, il donne le genre, le nombre et le type s'il n'est pas déjà donné par TreeTagger.
- Pour le nom, il donne le nombre et le genre.
- Pour le verbe, il donne le mode, la personne, et le temps.

Pour simplifier la tâche d'analyse morphologique des textes, nous avons intégré l'outil Flemm dans le module de TreeTagger. Par conséquent, chaque texte analysé par TreeTagger doit être ré-analysé par Flemm pour avoir les principaux traits morphologiques de chaque mot.

Dans notre plate-forme, nous avons utilisé la version 3.1 du Flemm.

3.1.3 Adaptation des étiquettes d'ASVM

Concernant la langue arabe, nous utilisons ASVM comme analyseur morphologique dans notre plate-forme. Ce dernier utilise un jeu de 22 étiquettes.

Pareil que pour le français, nos étiquettes sont plus larges que celle d'ASVM et par la suite, nous ne sommes pas obligés de réduire ou de grouper les étiquettes de l'analyseur.

Pour le cas de l'arabe, on va se limiter aux étiquettes proposées par ASVM vu l'inexistence des outils efficaces avec un jeu d'étiquette plus complet que celui d'ASVM. Malgré ce problème, les étiquettes actuelles d'ASVM nous permettent, dans un premier temps, de créer des activités en utilisant 20 étiquettes morphosyntaxiques.

Actuellement, les enseignants peuvent donc générer leurs propres unités pédagogiques dans un but bien précis qui porte sur les points grammaticaux tels que les verbes, les déterminants, les adjectifs, etc. De plus, ils ont la possibilité de concevoir des activités en adéquation avec les faiblesses des apprenants.

3.2 Paramétrisation et génération automatique des activités

La génération automatique des activités est considérée comme une tâche difficile, car elle nécessite l'intervention de trois domaines : l'informatique, la linguistique et la didactique des langues. L'approche qui sous-tend le système est résolument orientée utilisateur dans la mesure où le système est destiné principalement aux enseignants de langue, qui, a priori, n'ont que peu ou pas de connaissances en TAL ou en informatique. La nature technique du TAL doit être transparente pour les enseignants de langue et seuls les aspects didactiques et pédagogiques doivent être visibles et disponibles pour l'utilisateur. La figure ci-dessous

présente notre diagramme d'activité qui schématise les étapes de processus de génération des activités pédagogiques au sein de notre plate-forme.

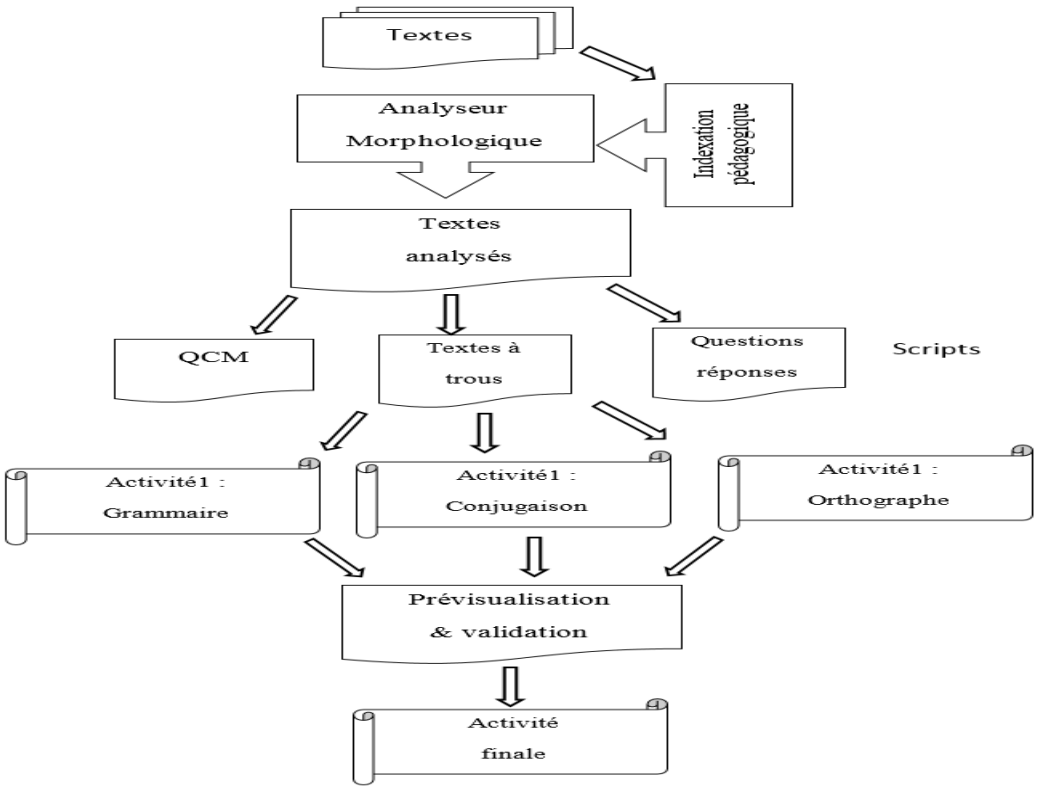

Figure 44 : Diagramme d'activité qui modélise le processus de génération des activités pédagogiques.

La conception d'activités est un niveau purement didactique et elle est opérée par les enseignants de langue. Le processus de génération des activités est composé de plusieurs étapes. Par exemple, les étapes de génération d'une activité texte à trous sont :

- Choix de textes : l'enseignant peut choisir un texte qui existe déjà dans nos bases des données, ou bien, il peut apporter son propre texte.
- Indexation de textes : dans cette étape, nous indexons les textes apportés par l'enseignant selon son thème. Si le texte est choisi à partir de nos bases de textes, on n'a pas besoin de l'indexer une autre fois (il était déjà indexé lors de son ajout aux bases de textes).
- Analyse morphologique : dès que le texte est indexé, il passe à l'entrée de l'analyseur morphologique (TreeTagger ou ASVM) puis, en sortie, on obtient un texte analysé morphologiquement.

- Définition d'un contexte didactique : dans cette étape, l'enseignant ou le didacticien doit définir un contexte didactique pour l'application du script. Cette opération de réglage de script permet à l'enseignant de choisir les éléments d'une base de texte indexé pédagogiquement et de déterminer les éléments (les critères sur la forme, la catégorie et/ou morphosyntaxiques).
- Scripts : ils permettent la paramétrisation et la génération des activités à partir des textes étiquetés. Par exemple, pour créer une activité « texte à trous », en premier lieu, l'enseignant doit choisir le script « texte à trou », en second lieu, il doit choisir les paramètres de l'activité selon son but didactique, enfin, avant la production de l'activité, l'enseignant doit préciser les instructions de l'activité et doit préciser l'aide qui sera donnée à l'apprenant (Antoniadis, 2010).

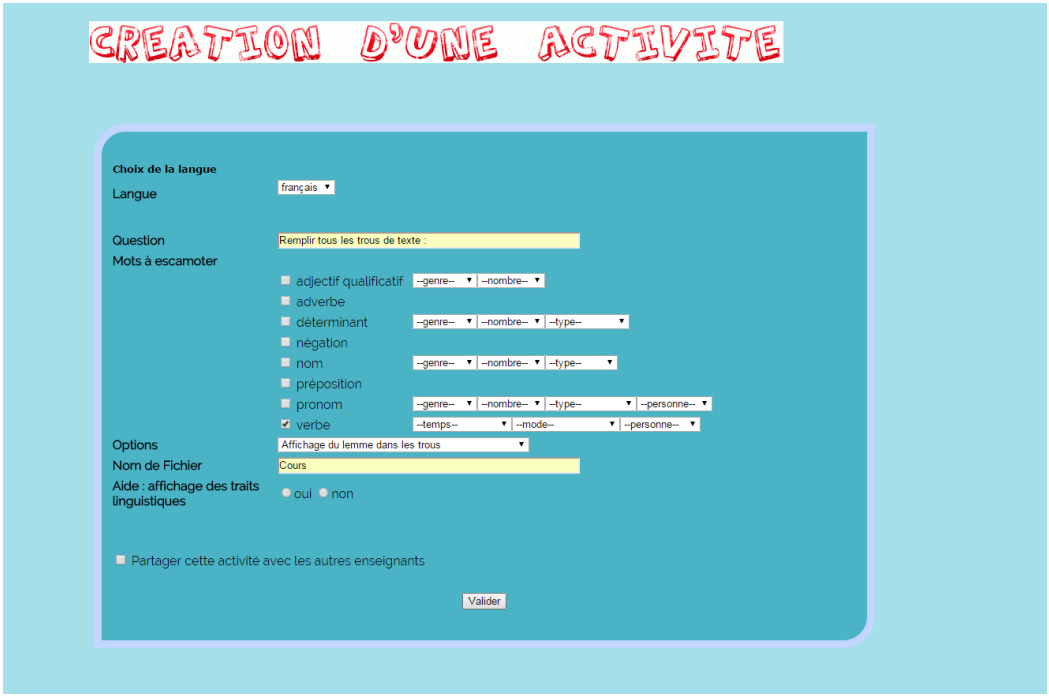

Figure 45 : Interface de paramétrisation d'une activité texte à trous.

- Génération des activités : dès que l'enseignant a paramétré l'activité, il ne lui reste que la validation de ses choix. Puis, la plate-forme génère l'activité selon les paramètres sélectionnés. Enfin, l'enseignant doit vérifier s'il y a des fautes dans l'activité liées aux erreurs générées par l'analyseur et par la suite, l'activité devient valide.

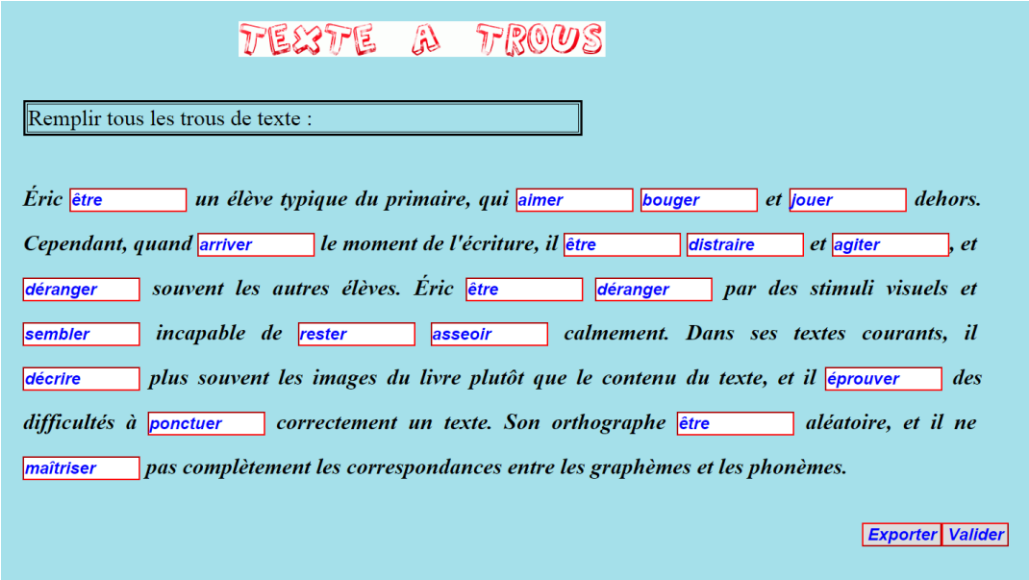

Figure 46 : Exemple d'une activité générée avec affichage des lemmes dans les trous.

Dans notre plate-forme, nous avons intégré la possibilité de regrouper plusieurs étiquettes dans une même activité. Par exemple l'enseignant peut regrouper l'étiquette adverbe et l'étiquette adjectif afin de proposer une activité sur les adjectifs et les adverbes dans le même exercice.

Le système offre la possibilité d'exporter les activités créées sous formats HTML ou XML. De plus, il y a un autre aspect pouvant présent dans une activité : il concerne l'évaluation des résultats obtenus lors de l'utilisation de l'activité par un apprenant, ou, en d'autres termes, l'évaluation des réponses. Cet aspect n'est présent que pour les activités pour lesquelles une évaluation automatique des résultats est possible. Nous détaillons ce module d'évaluation (feedback) dans les sections suivantes.

Un des avantages de notre plate-forme est son aspect ouvert, c'est-à-dire que la plate-forme permet d'ajouter d'autres types d'activités ainsi que d'autres outils et ressources TAL. Pour cela, nous allons essayer d'intégrer un autre outil TAL, le système de la reconnaissance de l'écriture, et de nouveaux scripts qui nous permettent de générer des activités basées sur cet outil.

4 Impact informatique de l'intégration de l'arabe dans un système d'apprentissage.

L'intégration de la langue arabe nécessite certains traitements complexes. De plus, l'affichage correct des caractères arabes est indispensable lors d'une activité d'apprentissage.

Puisque, dans la langue arabe les caractères changent leurs formes selon la position qu'ils occupent dans un mot (au début, au milieu, à la fin ou isolé), il est indispensable de faire une analyse contextuelle sur les mots afin de trouver la forme correcte de chaque caractère. D'autre part, l'arabe s'écrit et se lit de droite à gauche (Moukdade & Large, 2001), pour cela, il faut prendre en compte cette particularité lors de manipulation des données linguistiques arabes et aussi lors de l'affichage des caractères lors d'une activité.

Dans la partie de l'intégration de la langue arabe dans notre plate-forme, nous devons traiter les deux objectifs suivants :

- Facilité l'usage de la langue arabe (lecture ou écriture) par l'enseignant ou l'apprenant.
- Rendre la conception des activités intégralement indépendantes de la langue de dialogue choisie par l'utilisateur (enseignant ou apprenant).

4.1 Codage informatique de l'alphabet arabe

Les ordinateurs traitent les textes en leur assignant un encodage particulier, c'est-à-dire un système qui fait la correspondance entre les données électroniques et les caractères de texte visuels. Dans cette optique, nous devons s'assurer que nous avons choisi le codage correct pour la langue traitée, sinon, l'affichage des caractères ne sera pas correct.

Les codages disponibles actuellement pour l'arabe sont :

- ISO 8859-6 Latin/Arabic
- ASMO 449
- MS Arabic Code Page 1256
- Arabic Mac Code Page
- MS Arabic Dos Code Page 708 (ASMO 708)
- Arabic Windows 3X Code Page
- Arabic Windows 95 Code Page
- Code Page 864 Arabic
- Unicode

4.2 Choix du codage

Nous trouvons des difficultés à manipuler des mots arabes, en particulier dans le processus de sauvegarde et de recherche des mots. Pour cela, dans les travaux de cette thèse, nous avons décidé d'utiliser l'encodage des caractères Unicode, qui est l'UTF-8 pour représenter les mots arabes. Nous avons utilisé ce choix car selon (Habash, 2010) la majorité

des outils TAL tel que les analyseurs morphologique et les systèmes de traduction utilisent l'encodage UTF-8.

Cependant, l'encodage UTF-8 nécessite certains processus supplémentaires pour le manipuler correctement.

L'Unicode représente une collection des algorithmes et des tableaux des propriétés (Delahunt, 2007). Il fournit un numéro unique pour chaque caractère, chaque plate-forme, chaque programme et chaque langue[36]. Un des schémas de codage de caractères Unicode est l'UTF-8. Il est composé d'une séquence de 1 à 6 octets (8 bits).

Dans notre plate-forme, nous avons encodé tous les programmes et scripts utilisés en utf-8 en utilisant la syntaxe suivante :

- HTML : chaque page html doit contenir la ligne suivante : « <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/> ».
- Perl : l'en-tête de chaque programme Perl doit contenir cette déclaration : « binmode(STDOUT, ":utf8") »;
- PHP : tous les affichages textuels devraient être encodés avec la fonction php suivantes : « utf8_decode('Ici mon texte en UTF-8'); ».
- C & C++ : il faut que les fichiers sources (extensions .c, .cpp ou .h) encodés en utf-8.

5 Intégration du système de la reconnaissance de l'écriture

Après l'intégration des analyseurs morphologiques arabes et français dans notre plate-forme, nous allons maintenant intégrer notre système de reconnaissance de l'écriture manuscrite pour les deux langues. Puis, nous allons développer les scripts nécessaires pour la génération des activités. Pour cela, nous avons décidé d'adapter la sortie de notre système de reconnaissance aux besoins des linguistes et des didacticiens.

[36] http://www.unicode.org/standard/WhatIsUnicode.html

5.1 Processus de génération des activités

Généralement, dans un processus d'apprentissage de l'écriture, l'apprenant doit commencer par apprendre l'écriture des caractères, puis l'écriture des mots et enfin l'écriture des phrases. Par conséquent, nous allons intégrer dans notre plate-forme les trois modèles de reconnaissance ; modèle des caractères, modèle des mots et modèle des phrases.

La figure ci-après montre le diagramme d'activité de processus de génération des activités pour l'apprentissage de l'écriture. D'abord, l'enseignant doit définir un contexte didactique pour l'application de script. Puis, il doit choisir le modèle de reconnaissance à utiliser (caractères, mots ou phrases). Enfin, il ne lui reste qu'à définir les instructions et valider la création de l'activité.

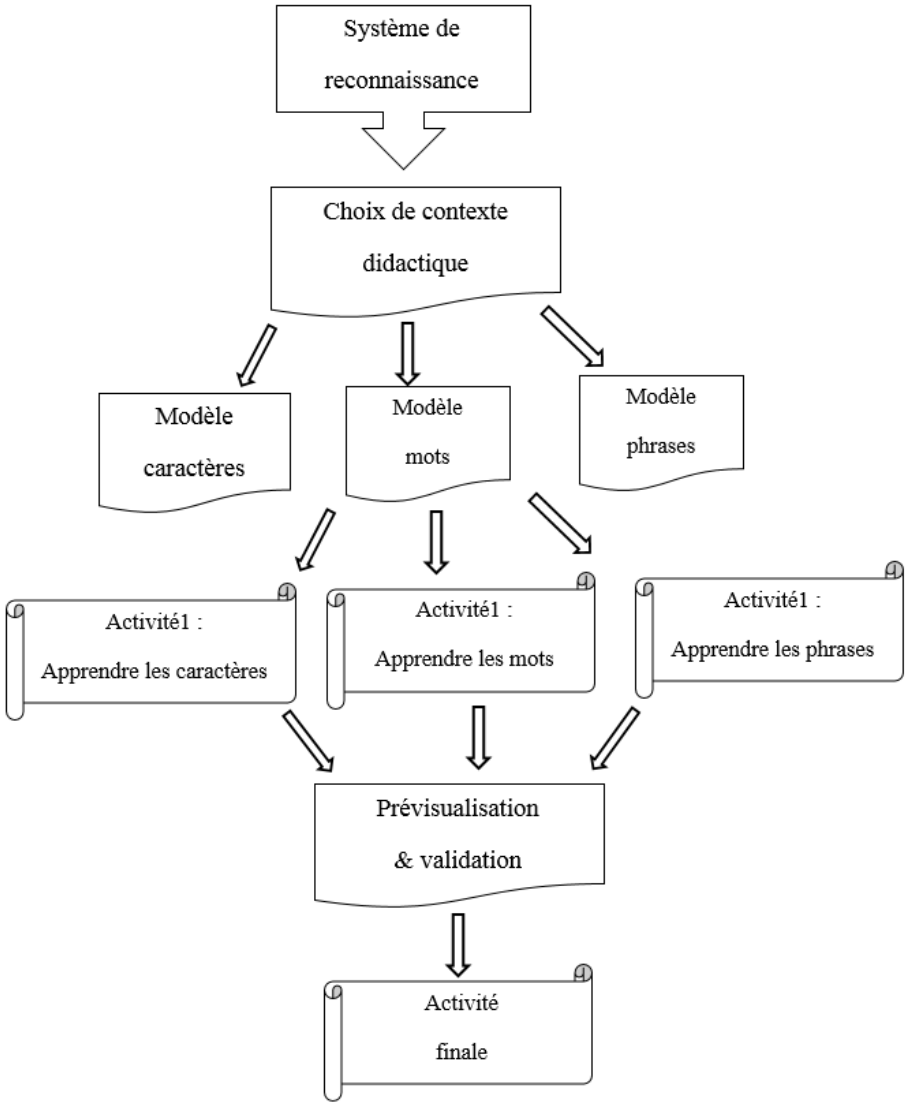

Figure 47 : Diagramme d'activité de processus de génération des activités d'apprentissage de l'écriture.

5.2 Adaptation des modèles

En didactique des langues, une activité doit répondre à un ensemble de critères bien étudiés. De plus, elle doit être bien conçue pédagogiquement pour qu'elle réponde au besoin de l'apprenant (difficulté, niveau, évaluation, etc.). Pour cela, et dans le cadre de l'apprentissage de l'écriture, nous allons montrer un exemple d'utilisation de notre système de reconnaissance dans l'ALAO. Ensuite, pour concevoir d'autres activités, nous devons définir les contextes didactiques ainsi que la conception de l'activité avec des pédagogues et des didacticiens.

Dans le choix de type de l'activité, nous devons répondre à deux questions posées par les didacticiens :

1) Comment peut-on reconnaître à l'aide d'un système de reconnaissance l'écriture exacte des apprenants ?
2) Quelle est la bonne solution pour générer un feedback adéquat dans cette situation d'apprentissage ?

Pour répondre à la première question des didacticiens, nous avons proposé une nouvelle approche de reconnaissance. Elle consiste à utiliser des modèles libres pour la reconnaissance de l'écriture manuscrite.

Pour cela, nous avons relancé la phase d'apprentissage de chaque modèle (caractères, mots et phrases) avec les hypothèses suivantes :

- Attribuer le coefficient zéro au modèle de langage basé sur le dictionnaire.
- Augmenter la probabilité des classes extraite lors de la phase de classification à partir de nos bases d'apprentissage.

En appliquant ces deux hypothèses à notre système, on ne reconnaît que la forme de caractère ou de mot, en se basant sur nos bases d'apprentissages.

Concernant la deuxième question, nous avons gardé les modèles qui utilisent un modèle de langage et qui donnent un taux de reconnaissance élevé pour la génération de feedback. Bien que ces résultats soient élevés, il existe toujours un risque de ne pas reconnaître le bon caractère ou le bon mot. Pour cela, lors de la phase de la création d'une activité d'apprentissage de l'écriture l'enseignant doit vérifier si le système arrive à bien à reconnaître les données proposées (caractères, mots ou phrases) avant de valider la génération de l'activité.

La figure ci-dessous montre une interface de paramétrisation d'une activité destinée à l'apprentissage de l'écriture.

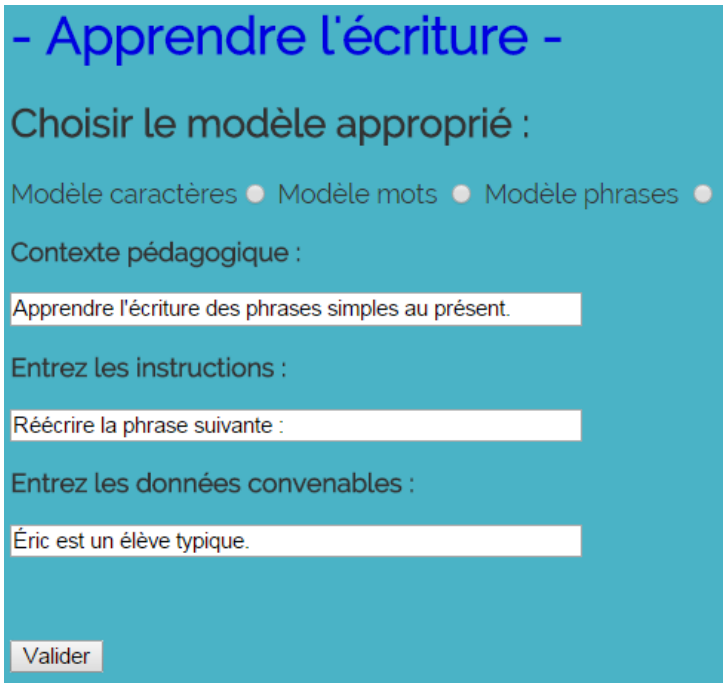

Figure 48 : Paramétrisation d'une activité d'apprentissage de l'écriture.

Dans notre plate-forme, nous avons simplifié la phase de génération des activités destinées à l'apprentissage de l'écriture, elle ne nécessite que deux étapes :

- La paramétrisation : à l'aide d'une interface simple, l'enseignant doit saisir les paramètres de l'activité. D'abord, il doit choisir le bon modèle de reconnaissance (nous utilisons les modèles libres dans cette étape). Ensuite, l'enseignant doit définir le contexte didactique de l'activité ainsi que les instructions destinées aux apprenants. Enfin, il ne reste que la précision des caractères, des mots ou des phrases à réécrire par les apprenants.
- La validation : dès que l'enseignant valide les paramètres, il faut qu'il vérifie si le système arrive à reconnaître les propositions. Une fois la vérification faite, l'activité devient valide et accessible sur la plate-forme.

Figure 49 : Exemple d'une activité générée en utilisant le système de la reconnaissance de l'écriture.

Pour conclure, le développement et l'intégration d'un outil TAL dans une plate-forme d'apprentissage est très coûteux en terme des ressources, de développement, de la charge de la mise en œuvre, du temps et d'effort. De plus, ces outils présentent un problème majeur lié à leurs performances. À l'heure actuelle, aucun outil TAL ne donne des résultats efficaces à 100 %; nous serons dans l'obligation de trouver des solutions pour contourner ce problème de fiabilité et de performance. Par conséquent, les chercheurs (linguistes et informaticiens) restent divisés sur leur valeur à long terme, bien que la plupart d'entre eux affirment que cette méthode d'apprentissage augmente la motivation des apprenants et présente un outil très efficace pour aider les apprenants débutants.

6 Développement d'un outil de sauvegarde des traces

Dans le but de concevoir une architecture complète pour notre plate-forme, nous avons développé un outil de sauvegarde automatisé. Ce dernier est capable de sauvegarder les traces des apprenants et des enseignants dans la base de données (par exemple : réponses des apprenants, type d'activité, temps de réponse, activité générée, etc.).

Lors de la mise en œuvre d'une activité par les enseignants, la plate-forme collecte les traces qui concernent les paramètres choisis, les textes choisis et le contexte pédagogique. Puis, l'outil sauvegarde les traces relatives à la production des apprenants et à l'interaction entre l'apprenant et l'environnement et entre l'apprenant et l'enseignant.

6.1 Avantages de l'utilisation des traces

Notre outil de traçage permet d'obtenir des traces formalisées en XML; elles peuvent être exploitées et traitées par nos outils de feedback. De plus, elles nous permettent d'effectuer certaines statistiques sur le niveau et les progrès des apprenants ainsi que sur l'utilisation de la plate-forme.

Selon (Lunda et al, 2000) et (Lunda et al, 1999) les traces ont un rôle crucial lors de l'évaluation de niveau des apprenants par les enseignants, car elles permettent aux enseignants de ne pas focaliser leur pédagogie que sur les fautes des apprenants, mais aussi sur leur compréhension des concepts et leur méthodologie. Par conséquent, l'utilisation des traces permet à l'enseignant d'assurer un rôle très important dans le suivi de l'avancemet des apprenants, car il a tous les données nécessaires pour évaluer les apprenants (Teutsch et al, 2004).

6.2 Suivi a posteriori

La sauvegarde des traces en une représentation structurée de l'activité et des productions de l'apprenant présente une source fiable pour le suivi d'un apprentissage des langues sur un système d'ALAO. Selon (Guéraud et al, 2004) la mise à la disposition des enseignants d'une synthèse des réponses d'un apprenant permet l'évaluation de l'apprenant et l'observation de l'avancement de l'apprenant durant la formation.

7 Feedback automatique

L'objectif de ce travail est de développer un module pour générer automatiquement un feedback en adéquation avec la réponse de l'apprenant. Ce module est très important pour augmenter la motivation d'apprentissage via la plate-forme chez les apprenants, et permettre, selon (Shintani et al, 2015), l'amélioration du niveau des apprenants.

Actuellement, à l'aide de notre plate-forme l'enseignant peut générer plusieurs activités, elles sont visibles sur le profil de l'enseignant et les apprenants peuvent avoir accès. Une fois l'apprenant ait fini l'activité, le système lui propose deux options, soit il envoie l'activité remplie à l'enseignant directement, soit il utilise une évaluation automatique de ses productions proposées par notre système.

7.1 Difficultés de la génération de feedback

Les productions des apprenants, dans le cadre d'une activité, peuvent avoir des formes très diverses comme des mots, des phrases ou même des textes. L'évaluation des phrases et des textes est un problème difficile : les techniques de TAL ne peuvent pas vraiment donner des informations fiables sur les caractéristiques qui nécessitent une interprétation humaine même pour la simple tâche de détection d'erreurs. Les modèles existants sont silencieux et bruyants à la fois : quelques erreurs ne sont pas détectées et des expressions correctes sont parfois signalées comme des erreurs. À l'opposé, l'évaluation d'un questionnaire à choix multiples est un problème trivial qui n'a pas besoin d'une mise en œuvre coûteuse d'outils TAL. D'autre part, dans le cas où la réponse serait sous la forme d'un mot comme dans le cas d'un texte à trous, alors l'utilisation des outils TAL devient cruciale pour un feedback précis et adéquat (Dzikovska et al, 2014).

7.2 Génération de feedback

Dans le but de développer un module complet pour évaluer automatiquement les réponses des apprenants, nous avons exploité les résultats de recherche de notre laboratoire dans ce domaine (Antoniadis et al, 2005). L'approche proposée par l'équipe consiste à évaluer la réponse de l'apprenant par rapport à la réponse attendue correcte selon un protocole de trois niveaux.

Le principe de ce protocole est basé sur l'utilisation des outils et des ressources TAL pour donner un feedback adéquat. Nous avons implémenté cette approche sous forme d'un algorithme :

```
Acquérir les réponses de l'apprenant
Tant qu'il existe des réponses faire
      Hypothèse 1 : fautes d'orthographe
      Si la réponse donnée est différente alors
      Si la chaîne entrée n'existe pas dans un dictionnaire des formes
fléchies alors
       Écrire (faute d'orthographe)
      Si la chaîne est très proche de la réponse correcte
       Écrire (avertissement sur l'erreur d'orthographe)
      Sinon
       Proposer une liste de mots ressemblant à la fausse réponse donnée
       Choisir le mot correct à partir de la liste
      Hypothèse 2 : fautes au niveau morphosyntaxique
       Intégration du contexte linguistique de l'activité
```

```
        Calculer une analyse morphosyntaxique avec l'étiquetage et la
lemmatisation
      Si le lemme est le même que le lemme de la bonne réponse alors
       Afficher la différence des caractéristiques morphosyntaxiques
      Hypothèse 3 : fautes au niveau sémantique
      Si le lemme est différent du lemme de la réponse correcte
       Rechercher dans le Wordnet sémantique
       Vérifier s'il existe un lien sémantique (synonymie, hyperonymie,
hyponymie, antonymie, etc.)
       Écrire (n'est pas exactement, être plus précis)
      Vérifier la condition d'arrêt
Fin Tant que
```

Cet algorithme est très utile pour évaluer plusieurs types d'activités : combler les lacunes, les questions lexicales, etc. Selon le contexte spécifique et l'objectif d'une activité donnée, le feed-back à l'apprenant peut être très différent. Par exemple, si un exercice lacunaire est conçu pour tester la capacité à conjuguer des verbes en un temps donné, le fait que le lemme de réponse de l'apprenant soit différent n'est pas très important, à condition que la flexion verbale soit correcte. Par conséquent, dans la conception d'un tel scénario d'évaluation, il est important de séparer la comparaison et le feed-back. Nous avons mis en œuvre deux scripts :

- Le script de comparaison qui prend en entrée le contexte linguistique, la réponse attendue, la réponse donnée et renvoie un code de différence tel que : 0 : pas de différence 1.1 : erreur d'orthographe sur la réponse attendue 1.2 : erreur d'orthographe sur une autre réponse (avec une liste de mots proches) 2.1 : lemme incorrect.

- Le script de feed-back qui prend en entrée le code de la différence renvoie un message, tel que : « oui, mais l'orthographe est mauvaise », « être plus précis », etc. Même si l'on peut proposer des messages standards pour chaque code de la différence, l'enseignant doit évidemment être en mesure de modifier un ensemble de messages adaptés en fonction du contexte didactique d'une activité donnée.

La figure ci-contre schématise le principe de notre algorithme lors de la génération automatique de feedback.

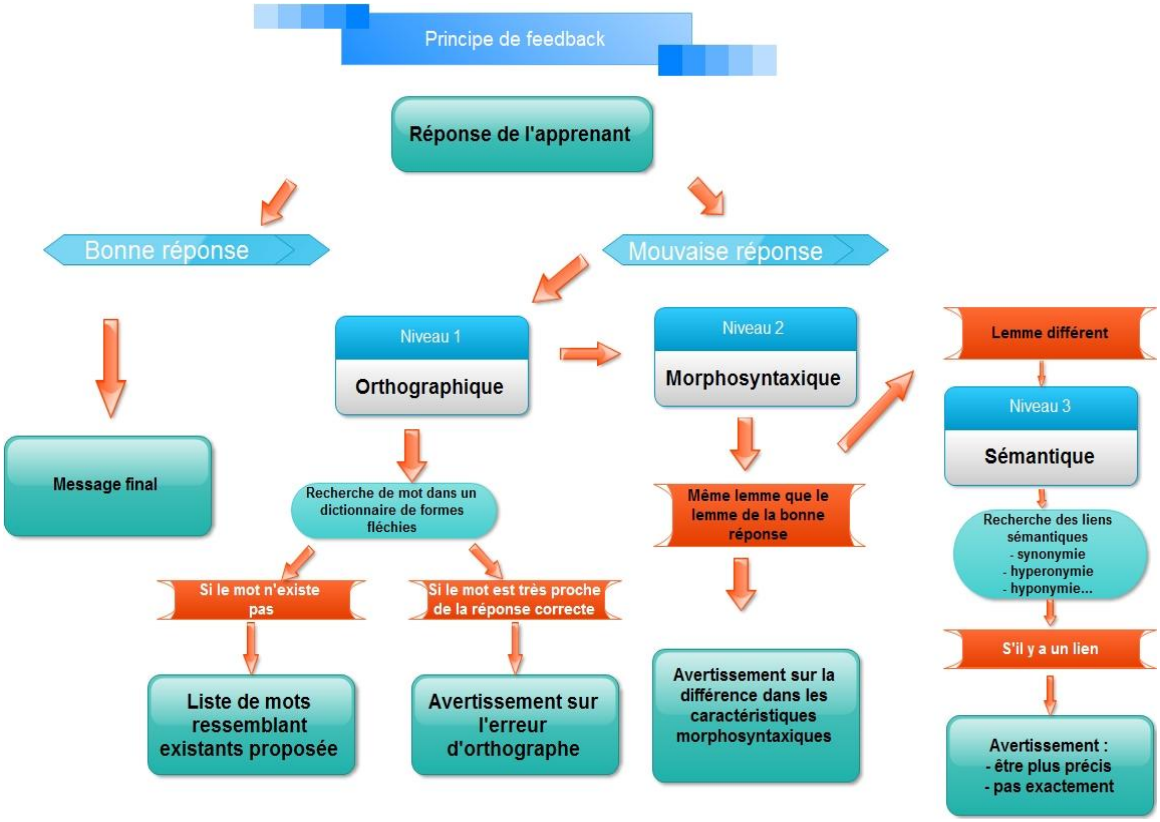

Figure 50 : Schéma explicatif de notre approche pour la génération de feedback intelligent.

7.3 Implémentation

L'objectif de ce travail est de mettre en œuvre les outils et les ressources utilisés pour achever l'implémentation de module feedback pour les deux langues, l'arabe et le français. Pour la partie développement, nous avons utilisé des technologies web (PHP, JavaScript et Ajax) ainsi que des modules PERL.

7.3.1 Cas du français

Lors de l'implémentation de notre algorithme, nous avons défini trois cas : erreur orthographique, erreur morphosyntaxique ou erreur sémantique. Concernant la partie des erreurs orthographiques, nous avons utilisé notre dictionnaire qui contient environ 700 000 mots (voir chapitre 7). De même, pour la deuxième hypothèse concernant les erreurs morphosyntaxiques, nous avons utilisé l'analyseur morphologique TreeTagger. En premier lieu, nous analysons les réponses de l'apprenant avec l'analyseur. En second lieu, nous comparons les lemmes d'origines avec les lemmes de la réponse. Enfin, nous affichons la différence entre les caractéristiques morphosyntaxiques s'ils ont le même lemme.

Comme solution à l'hypothèse trois, nous avons proposé l'utilisation d'un Wordnet sémantique. Théoriquement, un Wordnet est un réseau lexical qui couvre les noms, les verbes, les adjectifs, les adverbes, etc. Cette correction sémantique ne concerne que l'activité texte à trou.

Pour notre plate-forme, nous avons utilisé le Wordnet de Wolf (Sagot et al, 2012). Il est libre contrairement au projet EuroWordnet et il est disponible en téléchargement sur le site[37]. Selon (Sagot et al, 2012), le Wordnet est construit à partir du Princeton Wordnet. De plus, le WOLF est distribué sous le format XML, ce qui nous facilite la tâche d'exploitation et du traitement de contenu de ce Wordnet. Par conséquent, il suffit d'envoyer des requêtes vers le fichier XML pour avoir toutes les informations sémantiques sur le mot en question.

Dans cette thèse, nous avons utilisé la version « 1.0b4 » de WOLF.

7.3.2 Cas de l'arabe

De même pour la langue arabe, une partie de développement des outils et construction des ressources était faite. Nous avons déjà construit un dictionnaire des lexiques arabes qui nous sera utile lors de l'étape de vérification orthographique. Comme il contient environ 650 000 mots, on peut l'utiliser dans le module de feedback. De plus, nous avons amélioré l'analyseur morphologique ASVM afin de l'utiliser dans la phase de génération des activités pour la langue arabe; par la suite, nous avons exploité cet outil pour l'analyse des réponses des apprenants de la langue arabe via notre plate-forme.

Enfin, il reste à implémenter la dernière partie de notre algorithme qui concerne la vérification sémantique. Pour cela, nous avons choisi d'utiliser le Wordnet arabe (AWN : Arabic WordNet). C'est un outil libre d'utilisation, il emploie une base de données lexicale de la langue arabe qui suit le même processus de développement de l'outil Princeton Wordnet anglais et l'outil EuroWordet (Fellbaum et al, 2006). Il utilise l'approche SUMO (Suggested Upper Merged Ontology) pour lier AWN aux deux autres Wordnet (Princeton Wordnet et EuroWordnet). Aussi bien qu'AWN est considéré comme l'un des plus importantes ressources lexicales pour la langue arabe.

Dans cette thèse, nous avons utilisé la version « 2.0.1 » d'AWN. Pour l'exploitation de cet outil, nous avons développé un Script PERL qui nous permet d'envoyer une requête sur un mot donné; il nous rend en sortie les informations sémantiques sur le mot en question. AWN

[37] http://alpage.inria.fr/~sagot/wolf.html

est disponible sous deux formats, un fichier Java de type JAR et un fichier XML[38]. Dans notre implémentation, nous avons utilisé le fichier XML.

8 Indexation pédagogique du texte

8.1 Définition

Avant d'aller plus loin dans la description de la base de texte, nous devons bien définir ce que nous entendons par indexation pédagogique. L'indexation dépend du « langage documentaire » selon lequel il est effectué. Lefèvre définit un langage documentaire comme « langage artificiel qui fournit une représentation formalisée et univoque des documents d'un corpus et des questions qui intéressent un groupe d'usagers, afin de permettre le repérage simple des documents du corpus qui répondent aux questions de ces usagers (Lefèvre, 2004). Comme cette définition l'indique, la définition d'un langage documentaire doit être faite en fonction des utilisateurs potentiels de la base de données à indexer. Par conséquent, l'indexation pédagogique sera concernée par l'indexation des objets, dans le champ de leur utilisation potentielle par les enseignants dans le cadre de leur enseignement (Loiseau, 2004).

8.2 L'utilisation du TAL dans l'indexation pédagogique de texte

Dans le domaine du traitement automatique de langue, les méthodes telles que la TF-IDF (Term Frequency-Inverse Document Frequency) (Deerwester et al, 1990) et l'analyse sémantique latente (Baziz 2005) sont fréquemment utilisés pour extraire des termes à partir de corpus. L'idée de ces méthodes, appelées discriminantes, est qu'un terme est encore plus important lorsqu'il est fréquent dans un corpus texte et peu présent dans les autres. L'aspect discriminatoire vis-à-vis d'autres documents n'est pas valable dans notre contexte. En effet, dans un ensemble de documents pédagogiques sur le même sujet, les mêmes termes sont utilisés de façon répétée et dans ce cas la méthode employé par l'algorithme d'indexation l'utilise comme les mots-clés dans un contexte d'indexation. (Sanderson, 1997) établit une discrimination contre l'indexation sémantique et l'indexation conceptuelle. D'une part, le but de l'indexation sémantique est de lever l'ambiguïté existante dans le couple terme-sens. Cette activité selon (Khan, 2000) pourrait améliorer légèrement la précision d'un système de

[38] http://nlp.lsi.upc.edu/awn/get_bd.php

récupération de l'information dans la mesure que l'homonymie est exacte. D'autre part, le principe de l'indexation repose non seulement sur l'indexation sémantique, mais plus généralement sur l'utilisation du TAL comme charnière de l'outil d'indexation. (Loiseau, 2004) propose une méthode pour l'indexation de documents en tenant compte des co-occurrences de termes ainsi que de la proximité sémantique. Notre outil d'indexation ne traite que l'extraction des thématiques et des mots-clés d'un texte. Nous proposons une méthode pour les outils d'indexation dans laquelle le modèle prend comme entrée un document texte, puis génère, comme sortie la thématique et les mots-clés contextualisés du document.

8.3 Intégration dans la plate-forme

Dans le cadre de notre plate-forme, l'indexation pédagogique pour l'enseignement des langues permettra aux enseignants d'avoir un accès plus facile et plus pertinent à un ensemble de textes répondant à leurs questions, puis à partir de cet ensemble, de sélectionner les plus appropriés à leurs besoins. Il pourrait également permettre la définition d'un ensemble de caractéristiques requises pour les textes dans certaines activités. Cela permettrait aux étudiants de répéter une même activité sans répondre à la même question à plusieurs reprises. L'intégration de la base du texte indexé dans la plate-forme, nous la considérons comme une aide autonome pour les enseignants dans leur préparation de tout type d'activités.

Notre centre d'intérêt est l'apprentissage des langues et notre objectif est essentiellement pédagogique. L'attente de l'utilisateur (enseignant) d'un document indexé pour l'apprentissage serait d'extraire les objets qui peuvent être utilisés dans le cadre de son cours avec une application d'ALAO, cela implique que cette base de textes indexés doit assurer, grâce à son interface, les caractéristiques suivantes (Loiseau et al, 2005) :

- Permettre aux enseignants de faire des demandes sur la base de critères compatibles avec l'enseignement des langues.
- Permettre aux enseignants d'ajouter leurs propres textes à la base.
- Les professeurs de langues ne sont pas nécessairement des informaticiens ; l'interface devrait être aussi conviviale que possible.

8.4 Algorithme d'indexation

Dans un premier lieu, le module d'indexation prend en entrée des textes codés en utilisant l'unicode UTF-8. Le choix de ce codage a été justifié par le fait que, d'une part, la plupart des ressources numériques textuelles sont codées en utilisant cette norme, et d'autre

part parce que l'UTF-8 standard est supporté par les navigateurs les plus populaires. Ensuite, les textes sont traités par le module de gestion qui effectue la classification de tous les textes.

Étant donné :

```
T : ensemble des textes à indexé
Nb-t : représente le nombre de textes en entrée
BMT : base des mots-clés de chaque thème
Tant que la condition d'arrêt n'est pas satisfaite
  Pour T= 1 jusqu'à Nb-t
     Segmenter le texte en mots
       Calculer le score de ressemblance entre les mots
        de textes et le BMT
       Stocker les score de ressemblance dans un
      tableau
       Assigner le thème convenable selon le max de
      score
       Insérer dans la base des données le texte et
      le thème trouvé
  Fin Pour T
  Vérification de la condition
 d'arrêt
Fin Tant que
```

Afin d'implémenter cet algorithme, d'abord, nous avons construit une base de mots-clés appartenant à huit thèmes : technologie, économie, sport, politique, information, éducation, tourisme et culture. Ensuite, le programme prend en entrée un ou plusieurs textes et il segmente chaque texte en mots. Puis, l'algorithme calcule un score de ressemblances entre les mots de texte et les mots-clés de chaque thème. Enfin, il assigne le thème convenable, on se basant sur le score de ressemblance obtenu.

9 Évaluation de la plate-forme

À la fin de la phase d'implémentation de l'architecture complète de la plate-forme, il ne manque que la phase de test et d'évaluation. Cependant, évaluer l'architecture, l'apport du TAL, et l'influence de l'apprentissage des langues assisté par ordinateur (ALAO) sur la qualité de l'apprentissage de la langue est très problématique. Le développement et l'utilisation de l'apprentissage des langues assisté par ordinateur nécessitent souvent la participation de scientifiques, d'ingénieurs informatiques, des linguistes, des experts en intelligence artificielle, des psychologues cogniticiens, des mathématiciens, des développeurs, etc. (Ellis 2004). Depuis l'apparition des systèmes d'ALAO, la phase d'évaluation d'un tel système est considérée comme une tâche complexe, elle doit engager non seulement les chercheurs, mais tous ceux qui utilisent le système d'ALAO telle que les apprenants et les enseignants.

Le tableau ci-dessous représente les différents niveaux d'analyse selon (Chapelle, 2001) pour l'évaluation des systèmes d'ALAO.

Niveau d'analyse	Objectif de l'évaluation	Questions	Méthode d'évaluation
1	Système d'ALAO	Est-ce que le logiciel fournit aux apprenants la possibilité d'interactivité ?	Jugements
2	Activités proposées par l'enseignant	Est-ce que l'activité conçue par l'enseignant peut fournir aux apprenants la possibilité de modifier l'interaction pour la négociation du sens ?	Jugements
3	La performance de l'apprenant au cours d'activités	Est-ce qu'il y a une interactivité entre les apprenants pendant qu'ils travaillent leurs activités ?	Empirique

Tableau 28 : Différents niveaux d'analyse pour l'évaluation des systèmes d'ALAO.

Le premier niveau d'analyse désigne le logiciel d'ALAO. Cela est la cible habituelle pour l'évaluation des systèmes d'ALAO. Il pourrait être considéré comme l'élément le plus facile à évaluer, mais il ne faut pas oublier la pertinence des utilisateurs qui est considérée comme le niveau deux et trois de l'analyse.

Dans le deuxième niveau, le rôle et la participation de l'enseignant sont pris en compte, ainsi que la liberté dans le choix de type et de niveau de l'activité. (Chapelle, 2001). Le troisième niveau se concentre sur le progrès de l'apprenant au cours de son apprentissage à travers la plate-forme.

9.1 Avantages de la plate-forme

Avant de passer à l'étape d'évaluation de la plate-forme, nous commençons par préciser les différents utilisateurs et les avantages offerts par la plate-forme. Actuellement, la plate-forme réalisée est destinée :

- Aux enseignants de langue, lettres et industrie de la langue, linguiste.
- Aux étudiants d'une langue seconde, aux élèves de l'école.
- Aux chercheurs (enseignants et étudiants) en langues, linguistique et industrie de la langue, lettres.

La plate-forme capable d'offrir la possibilité aux enseignants de générer plusieurs types d'activités. De plus, la plate-forme permettre aux enseignants de profiter pleinement des logiciels TAL (tels que l'analyseur morphologique, le générateur d'activités, le module de feedback, le module d'indexation pédagogique des textes, etc.) via une interface simple à utilisée. Aussi, les ressources linguistiques telles que les dictionnaires, les corpus textuels et les corpus d'encre sont mis à la disposition des enseignants pour enrichir le contenu des activités. De plus, la plate-forme dispose d'un module de sauvegarde et d'exploitation des traces et d'un système de reconnaissance de l'écriture manuscrite qui permet la création de plusieurs variétés des activités destinées à l'apprentissage de l'écriture.

Une fois la plate-forme testée et validée, elle sera accessible, via le Web, aux apprenants et aux enseignants.

Dans le but de tester les différentes composantes de la plate-forme et avant de la mettre dans un environnement d'apprentissage réel, nous avons effectué quelques tests expérimentaux par un enseignant et des apprenants de l'université de Tirana dans le cadre du projet de l'AUF (les détails de ce projet est dans la section 8.2). Ces tests ont permis de visualiser les avantages apportés par la plate-forme d'ALAO aux enseignants et aux apprenants, tels que la semi-automatisation, le feedback, l'indexation des textes, etc. D'autre part, ces tests permettent de détecter des éventuelles erreurs de fonctionnement commises lors de la phase de développement.

9.2 Évaluation de la langue française

Une fois que les tests et les expériences sont achevés, nous avons décidé de mettre la plate-forme dans un environnement d'apprentissage réel. Pour cela, nous avons participé avec nos travaux de thèse (partie recherche et partie développement) à un projet international de l'AUF dont l'université Stendhal faisait partie. Il est appelé plate-forme du français technique et des technologies de la langue[39]. Cela nous a permis d'installer notre plate-forme d'ALAO en local dans deux universités européennes, l'université de Tirana et l'université de Moldavie.

[39] Ce projet est international et coordonné par l'Agence Universitaire de la Francophonie (AUF). Le projet a pour objectif de créer une plate-forme destinée aussi bien aux étudiants en Master Technologies de la langue qu'aux enseignants, visant à encourager la réflexion sur les particularités du français technique par la mise à disposition d'un corpus de textes techniques et d'instruments de traitement automatique de la langue (TAL) à des fins pédagogiques.

Avant de donner l'accès à notre plate-forme, nous avons commencé par une formation et démonstration de fonctionnement de la plate-forme aux enseignants de FLE et aux étudiants de Master technologie des langues. En second lieu, nous avons laissé la plate-forme aux enseignants pour l'utiliser quotidiennement lors de préparations et créations d'activités pour leurs apprenants. Au cours de ces tests, les enseignants et les apprenants ont rempli un questionnaire d'évaluation portant sur divers aspects des outils et sur l'utilisation de la technologie TAL pour l'enseignement des langues en général. Les réactions ont été généralement positives; elles incluent quelques suggestions détaillées pour améliorer les outils. De plus, les enseignants et les apprenants ont trouvé l'utilisation de la plate-forme facile et ne nécessitant pas beaucoup de temps pour la prise en mains.

9.3 Évaluation de la langue arabe

Actuellement, il n'y a que la partie d'apprentissage de la langue française qui était testé dans un milieu d'apprentissage réel dans le cadre du projet de l'AUF. Cependant, nous cherchons un éventuel accord avec le LANSAD[40] (Le service de LANgues pour Spécialistes d'Autres Disciplines) de notre université afin de mettre à leur disposition notre plate-forme d'ALAO pour l'arabe dans le but de la mettre dans un environnement de test réel.

10 Conclusion

Nous avons décrit dans ce chapitre l'architecture complète de notre plate-forme multilingue dédiée à l'apprentissage de langues assisté par ordinateur ainsi que l'intégration des outils TAL tels que le générateur automatique des activités, l'outil de traçage, le système de reconnaissance de l'écriture manuscrite, etc. Ils sont à la fois le but de l'environnement et en assurent la cohérence. Malgré l'importance des outils et des techniques employés dans le système d'ALAO, le TAL ne s'adapte pas de façon immédiate à l'apprentissage / enseignement des langues et de la linguistique. La première difficulté est tout d'abord liée aux performances imparfaites des outils de TAL qui doivent de ce fait être utilisés avec circonspection dans les applications pédagogiques, une solution consiste alors à se diriger vers une génération semi-automatique dans un cadre défini.

[40] http://lansad.univ-grenoble-alpes.fr/

Actuellement, le marché est prometteur pour les apprenants qui sont à la recherche d'un moyen intelligent pour apprendre la langue française et surtout la langue arabe, car le domaine d'ALAO arabe est encore restreint à quelques prototypes.

Chapitre 9 : Conclusion et perspectives

1 Contributions

Dans ce travail de thèse, nous nous sommes principalement intéressés au développement d'une plate-forme multilingue pour l'apprentissage de langues assisté par ordinateur (ALAO) en favorisant l'utilisation d'outils du traitement automatique de la langue et des approches issues de l'intelligence artificielle (IA). Cette plate-forme permettra aux enseignants des langues de définir des contextes pédagogiques et de créer plusieurs types d'activités.

Au début de ce travail, nous avons commencé par la présentation générale de la langue arabe et nous avons montré la complexité du traitement automatique de cette langue (segmentation et analyse morphologique, ambiguïté, etc.). Ensuite, nous avons exposé la problématique de cette thèse, puis nous avons présenté notre solution et l'apport de l'utilisation du TAL pour l'ALAO ainsi que les démarches à suivre pour la résolution de ces problèmes. Dans cette optique, nous avons organisé nos démarches sur quatre étapes principales, recherche et comparaison des outils et des ressources TAL disponibles pour l'arabe et le français, construction des ressources linguistiques et amélioration des outils sélectionnés, développement d'un système multilingue pour la reconnaissance de l'écriture manuscrite en ligne et intégration des outils et module TAL dans notre plate-forme d'ALAO.

À la fin de ces démarches, nous avons réussi à construire des nouvelles ressources, un corpus étiqueté pour la langue arabe, deux dictionnaires de taille importante (un pour chaque langue). Aussi bien, nous avons choisi l'analyseur ASVM pour l'arabe et TreeTagger pour le

français. Bien que les expérimentations et les tests de l'outil TreeTagger aient été satisfaisants, par contre, un grand travail d'amélioration de l'outil ASVM a été fait.

Ensuite, nous avons présenté les démarches nécessaires pour la réalisation d'un système multilingue pour la reconnaissance de l'écriture manuscrite en ligne. Au niveau recherche, nous avons adapté l'approche de réseau de neurones vu qu'on traite une langue complexe, l'arabe. Le réseau de neurones a été utilisé dans la reconnaissance de l'écriture manuscrite par d'autres chercheurs, mais surtout au niveau de la reconnaissance de caractères. Aussi bien pour notre thèse, nous avons également mis en place un système de reconnaissance de caractères en utilisant le RN puis nous l'avons généralisé pour reconnaître les mots et les phrases sans contrainte. D'autre part, nous avons construit notre propre démarche d'implémentation d'un système de reconnaissance basé sur six étapes pour les deux langues ; prétraitement et normalisation, détection de ligne des références, segmentation, extraction de caractéristiques, apprentissage et reconnaissance.

Une fois l'implémentation de notre système de reconnaissance achevée, nous avons montré expérimentalement qu'un système basé sur le RN donne des taux élevés concernant la reconnaissance des caractères, en particulier l'utilisation de deux classifieurs, le TDNN et le PMC de RN. De plus, au niveau de la reconnaissance des mots, les systèmes de reconnaissance à base de réseaux de neurones ont un avantage, par rapport aux autres approches, on peut effectuer une optimisation du système au niveau du mot en parallèle avec l'optimisation du système au niveau de caractère. Cette optimisation est possible grâce à l'utilisation de l'approche de la rétro-propagation du gradient lors de la phase d'apprentissage.

L'application de cette approche, nous a permis d'obtenir des bons résultats qui arrivent pour la reconnaissance des caractères jusqu'à 99,3 % pour le Français et 98,5 % pour l'Arabe. D'autre part, nous avons comparé les résultats obtenus avec ceux du système MyScript dans le but de valider les résultats, les approches et les algorithmes utilisés lors de développement de notre système. Nos résultats expérimentaux ont montré que l'approche et l'implémentation proposées donnent des bonnes performances. D'autre part, l'utilisation d'une telle approche, nécessite une grande base de données de bonne qualité pour entraîner les neurones du réseau. La préparation et la collecte de données afin de créer une base de données de qualité fait partie de nos travaux de recherche pour implémenter un système de reconnaissance multilingue. La base de données a été créée par la contribution de 200 scripteurs d'origine française et 200 scripteurs d'origine arabe, pour remplir 400 formulaires de six pages chacun. À la fin, nous avons réussi à obtenir une base des données de qualité.

Une fois les travaux de recherche et développement des outils et des ressources TAL achevés, il fallait maintenant construire l'architecture complète d'un environnement d'ALAO autonome et intelligent. Pour ce faire, nous avons conçu une plate-forme multilingue, intégrant les outils TAL développés précédemment. Nous avons implémenté une architecture complète combinant quatre modules basés sur le TAL :

- Générateur des activités : il permet de générer automatiquement des activités à partir des outils TAL intégrer dans la plate-forme pour la langue arabe. Le générateur est composé de plusieurs scripts qui permettent aux enseignants la paramétrisation et la génération des activités. Chaque script représente un type d'activité (texte à trou, Quiz, apprendre l'écriture).
- Traces : la structuration et la sauvegarde des traces font partie de l'architecture de la plate-forme. Elles permettent la sauvegarde automatique des traces des apprenants et des enseignants liées à toutes les activités faites sur la plate-forme.
- Feedback : ce module sert à évaluer automatiquement et d'une façon autonome les productions des apprenants. Cela se fait à l'aide de l'utilisation des outils TAL qui pourraient analyser les réponses et produire une correction en cas d'erreur. Actuellement, nous avons deux types de feedback, le premier est destiné à l'évaluation des activités traditionnelles (texte à trou, quiz, etc.) en utilisant un algorithme à trois niveaux, orthographiques, morphosyntaxiques et sémantiques. Le second est destiné à la correction de l'écriture manuscrite, il utilise un système de reconnaissance de l'écriture pour l'évaluation et la génération de feedback.

Nous considérons que l'aspect ouvert de notre plate-forme est un grand avantage pour la faire évoluer dans le futur. Dans cette optique, nous pouvons toujours ajouter à la plate-forme une infinité d'activités, intégrer plusieurs outils et ressources TAL, adapter la plate-forme aux besoins des enseignants et des didacticiens.

2 Perspectives et travaux futurs

Bien que le développement des outils TAL et de la plate-forme d'ALAO nous aient permis de créer un système complet destiné aux enseignants des langues et aux apprenants d'une langue étrangère. Cependant, nous avons rencontré un certain nombre de limitations et de difficultés lors de la recherche et développement de ces outils qui nous ont dirigés vers de nouvelles réflexions et nouvelles pistes pour des améliorations possibles de ces ressources et

outils. Par la suite, ces recherches nous ouvrent de nouvelles perspectives et de nouveaux axes de recherche.

L'analyse morphosyntaxique est une application très importante et fondamentale dans le domaine du traitement automatique de la langue, il joue un rôle très important dans le progrès des systèmes d'ALAO, il peut être intégré dans un large éventail d'applications d'ALAO. Dans cette optique, nous considérons que l'amélioration des analyseurs morphologiques intégrés dans la plate-forme est un axe de recherche prioritaire dans un premier temps. En particulier, l'amélioration de l'analyseur ASVM en lui ajoutant des nouveaux traits morphologiques et grammaticaux tels que, le genre, le nombre, la personne, le mode, etc.

D'autre part, nous envisageons la mise à jour de nos dictionnaires en augmentant leur couverture lexicale. Cela permettra d'augmenter les performances des outils de correction orthographique intégrés dans le module de feedback. Bien que notre plate-forme, soit capable de générer un feedback de trois niveaux aux productions des apprenants, nous envisageons en plus comme prochaines améliorations du système la personnalisation de la génération automatique de commentaires de façon pédagogique lors de la réponse de l'apprenant en ajoutant des exemples concrets à partir de nos bases de textes.

Un travail futur peut être suggéré pour améliorer la reconnaissance des phrases et des textes, c'est l'usage d'un système hybride, c'est-à-dire d'un système qui combine deux approches, l'approche du réseau de neurones et l'approche du Modèle de Markov Caché (HMM). Cela nous permettra de corriger les erreurs de classification des mots commises au niveau du modèle de mots lors de la reconnaissance des phrases. Dans ce cas, on réduira le taux d'erreurs lors de la reconnaissance des mots et des phrases dans notre système. De plus, nous avons énuméré un ensemble des points faibles pendant la phase de segmentation, en particulier lorsque le mot ou le caractère est accompagné par des diacritiques, qui bruitent la reconnaissance des unités élémentaires et la reconnaissance des mots. Dans cette optique, nous envisageons l'amélioration du module de prétraitement afin de faciliter la détection des signes diacritiques, l'amélioration de la segmentation en graphèmes et en caractères, amélioration de la détection des lignes des références et d'enrichir notre dictionnaire par des mots composés et mots avec diacritiques.

Il pourrait être envisagé par la suite de coupler notre modèle actuel basé sur le dictionnaire avec un modèle de langage statistique pour améliorer les performances de notre système de reconnaissance au niveau des phrases.

Nous envisageons aussi de mettre à la disposition des enseignants et des apprenants un module qui permet la définition et l'exploitation pédagogique de traces des activités des apprenants, un module pour créer automatiquement des bilans pédagogiques pour chaque apprenant et un module de génération automatique d'aides. Aussi bien pour l'implémentation des activités, il est tout à fait envisageable de concevoir des nouveaux type d'exercices à partir des outils et ressources TAL.

De plus, nous envisageons d'améliorer la première partie de la plate-forme (génération automatique des activités à partir de l'analyseur morphologique) en utilisant les savoir-faire dans le domaine du TAL et l'apprentissage des langues côté LIDILEM.

3 Conclusion

Nous avons présenté une architecture complète d'un système d'apprentissage des langues assisté par ordinateur qui favorise l'utilisation des outils de traitement automatique de langue (TAL). Ces travaux peuvent être considérés comme une contribution au domaine de l'ALAO, au domaine du TAL, et au domaine de l'intelligence artificielle.

Une fois les expérimentations et les tests de notre plate-forme, qui se déroulent actuellement à l'université de Tirana et à l'université Moldavie, achevés, nous envisageons d'implanter l'environnement sur le Web et le mettre à la disposition de la communauté du TAL et ALAO. Cela permettra d'effectuer un travail en collaboration avec les enseignants et les apprenants; de même, la mise en ligne de la plate-forme permettra d'enrichir les ressources et de la faire évoluer.

Bibliographie

1. Abeillé, A. Clément, L., et Toussenel, F. 2003. "Building a treebank for French". Anne Abeillé, editor, Treebanks. Kluwer, Dordrecht.
2. Abbes, R. 2004. " La conception et la réalisation d'un concordancier électronique pour l'arabe ". Thèse de doctorat en sciences de l'information, Lyon, ENSSIB/INSA.
3. Abedmouleh, A., Laforcade, P., Oubahssi L., et Choquet C. 2012. "Identification of LMSs Instuctional Languages: an Analysis Process". IEEE International Conference On Advanced Learning Technologies, Rome, Italie, p. 367-368 - 2012, 4-6 juillet 2012.
4. Abuzaraida, M.A., and Zeki, A.M. 2010. "Segmentation techniques for online Arabic handwriting recognition". Proceedings of 3rd International Conference on ICT4M.
5. Ahmad, A.R. 2008. "Reconnaissance de l'écriture manuscrite en ligne par approche combinant systèmes à vastes marges et modèles de Markov cachés". Université de Nantes, Décembre, 2008.
6. Aljlayl, M., and Frieder, O. 2002. On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach, 11th International Conference on Information and Knowledge Management (CIKM), November 2002, Virginia (USA), pp. 340–347.
7. Allauzen, A. et Bonneau-Maynard, H (2008). " Training and evaluation of POS taggers on the French MULTITAG corpus ". Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-08).
8. Al-Sughaiyer and A. Al-Kharashi. Arabic morphological analysis techniques: A comprehensive survey. Journal of the American Society for Information Science and Technology, 55:189-213, 2004.
9. Amin, A. 1997. "Off line Arabic character recognition". Fourth International Conference Document Analysis and recognition (ICDAR'97), p. 596 (1997).
10. Antoniadis, G., Granger, S., Kraif, O., Ponton, C., et Zampa, V. 2013. NLP and CALL: integration is working. http://arxiv.org/abs/1302.4814
11. Antoniadis, G. 2010. "De l'apport pertinent du TAL pour les systèmes d'ALAO : l'exemple du projet MIRTO". 2ème Congrès Mondial de Linguistique Française: 150.
12. Antoniadis, G., Echinard, S., Kraif, O., Lebarbé, T., et Ponton, C. 2005. "Modélisation de l'intégration de ressources TAL pour l'apprentissage des langues : la plateforme MIRTO", ALSIC, volume 8, novembre 2005, pp65-79.
13. Antoniadis, G., et Ponton, C. 2002. "Le TAL : une nouvelle voie pour l'apprentissage des langues", UNTELE, Compiègne, 28-30 mars 2002.

14. Attia, M. A. 2008. "Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation". Faculty of Humanities, pp. 279. Manchester: University of Manchester.
15. Attia, M, A. 2000. "A large-scale computational processor of the Arabic morphology". A Master's Thesis, Cairo University, (Egypt).
16. Attia, M. 2006. "Accommodating Multiword Expressions in an Arabic LFG Grammar". Advances in Natural Language Processing. FinTAL 2006, Lecture Notes in Computer Science. Vol. 4139, pp. 87 - 98, 2006. Springer-Verlag Berlin Heidelberg 2006.
17. Atwell, E. L., Al-Sulaiti, S., Al Osaimi, B. 2004. "Un Examen d'Outils pour l'Analyse de Corpus Arabes", JEP TALN, Session on Arabic Language Processing, Fès, 2004.
18. Badran, F., et Thirya, S. (2001). " Les Perceptrons Multicouches: de la régression non-linéaire aux problèmes inverses ". Research Report CEDRIC Lab/CNAM.
19. Baloul, S., Alissali, M., Baudry, M., et Boula de Mareüil, P. 2002. "Interface syntaxe-prosodie dans un système de synthèse de la parole à partir du texte en arabe". 24es Journées d'Étude sur la Parole, 24-27 juin 2002 Nancy, pp.329-332.
20. Baziz, M. 2005. "Indexation conceptuelle guidée par ontologie pour la recherche d'information". Thèse de doctorat, Université Paul Sabatier, 2005.
21. Beatty, K. 2003. "Teaching and researching computer-assisted language learning". New York : Longman, p 7 et p 8.
22. Beatty, M. 2009. "looking ahead with PLATO". Senior Learning at University of Wisconsin–Madison, Division of Continuing Studies. Volume XXVI, Number 3 August – September 2009.
23. Beesley, K. 2001. "Arabic Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001", Actes de la conference ACL/EACL 2001, Toulouse, 2001.
24. Belguith Hadrich Lamia, Baccour Leila et Mourad Ghassan, "Segmentation de textes arabes basée sur l'analyse contextuelle des signes de ponctuations et de certaines particules". 12ème conférence sur le Traitement Automatique des Langues Naturelles (TALN'2005), Dourdan, France, 6-10 juin 2005, pp 451–456.
25. Bengio, Y. 1994. "A Connectionist Approach to Speech Recognition". International Journal on Pattern Recognition and Artificial Intelligence, volume 7, issue 4, pages 647-668, 1994.
26. Berthet, N. 2001. Tell me More@Learning : auto-apprentissage des langues et tutorat en ligne. Tapuscrit non publié. Mémoire de DESS de sciences de l'éducation. UPMF-Grenoble.
27. Bertin, J. C. 2011. "Des outils pour des langues, Multimédia et Apprentissage". Paris : Ellipses, 2001. **
28. Biadsy F., Saabni R. and EL-Sana J. "Segmentation-free online Arabic handwriting recognition". Pattern Recognit. Artif. Intell. 25(7), 1009–1033.
29. Boggards, P. 1995. "Aptitudes et affectivité dans l'apprentissage des langues étrangères". Paris, Hatier, 1995.

30. Boubaker, H., Chaabouni, A., Kherallah, M., Alimi, A.M., and El Abed H. 2010. "Fuzzy segmentation and graphemes modeling for online Arabic handwriting recognition". Proceedings of ICFHR 2010, pp. 695–700 (2010).
31. Bouslama, F. and Amin, A. 1998. "Pen-based recognition system of Arabic character utilizing structural and fuzzy techniques". In: Proceedings of Second International Conference on Knowledge-Based Intelligent Electronic Systems, pp. 76–85.
32. Bottou, L. (2012). Stochastic Gradient Tricks, Neural Networks, Tricks of the Trade, Reloaded, 430–445, Edited by Grégoire Montavon, Genevieve B. Orr and Klaus-Robert Müller, Lecture Notes in Computer Science (LNCS 7700), Springer, 2012.
33. Breiman, L., Friedman, J., Olshen, R., and Stone, C. 1984. "Classification and regression trees," The Wadsworth & Brooks, 1984.
34. Bridle, J.S. 1990. "Training Stochastic Model Recognition Algorithms as Networks Can Lead to Maximum Mutual Information Estimation of Parameters", in Advances in Neural Information Processing Systems 2, D.S. Touretzky, ed., pages 211-21 7, 1990.
35. Brill, E. 1992. "A simple rule-based part of speech tagger". Actes de Third Conference of Applied Natural Language Processing.
36. Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F, Lafferty, J., Mercer, R., Roossin, P. 1990. "A statistical approach to machine translation". Computational Linguistics, 16(2):79-85, 1990.
37. Brun, C., Parmentier, T., Sandor, A. et Segond, F. 2002. "Les outils de TAL au service de la eformation en langues". Multilinguisme et traitement de l'information. Paris : Hermès. pp. 223-250.
38. Buckwalter, T. 2004. "Buckwalter Arabic Morphological Analyzer, Version 2.0". LDC Catalog No. LDC2004L02, Linguistic Data Consortium, 2004.
39. Buckwalter, T. 2002. "Arabic Morphological Analyzer Version 1.0". LDC, University of Pennsylvania, 2002. CEDAR. 1993. CEDAR CDROM-1. www.cedar.buffalo.edu/Databases/CDROM1
40. Chalabi, A. 2004. "Sakhrt Arabic Lexicon". Actes de la conférence internationnale NEMLAR, Arabic Language Ressources and Tools le Caire Egypte.
41. Chanier, T. 2000. "Hypermédia, interaction et apprentissage dans des systèmes d'information et de communication : résultats et agenda de recherche". In L. Duquette and M. Laurier, editors, Apprendre une langue dans un environnement multimédia, pages 53--89. Montréal : Editions Logiques, 2000
42. Chanier, T. 1998. "Relations entre le TAL et l'ALAO ou l'ALAO un simple domaine d'application du TAL ? ". Actes de la conférence International conference on natural language processing and industrial application (NLP+IA'98), Moncton, Canada.
43. Chapelle, C. A. 2001. "Computer applications in second language acquisition". New York : Cambridge p 3.
44. Chomsky, N. 1957. "Syntactic Structures". The Hague, Netherlands: Mouton and Co.
45. Clark, A. 2007. "Supervised and Unsupervised Learning of Arabic Morphology". Arabic Computational Morphology, pp. 181-200. Springer.

46. Constant, M. & Sigogne, A. 2011. "MWU-aware part-of-speech tagging with a CRF model and lexical resources". Workshop on Multiword Expressions: from Parsing and Generation to the Real World, ACL 2011.
47. Courtois. 1990. "Un système de dictionnaires électroniques pour les mots simples du français". Langue Française, vol. 87: 1941 – 1947.
48. Daifallah, K., Zarka, N., and Jamous, H. 2009"Recognition-based segmentation algorithm for on-line Arabic handwriting". Proceedings of International Conference on Document Analysis and Recognition, ICDAR 2009, pp. 877–880. Barcelona, Spain, IEEE.
49. Dat, M.-A., Spanghero-Gaillard, N., 2005. L'enseignement des langues et cultures étrangères à l'école primaire : un exemple d'utilisation de document authentique multimédia. CORELA Numéros thématiques | Colloque AFLS.
50. David, F., Jesús, G., Edgar G., Reda, H., Horacio, R., and Mihai, S. 2007. "The UPC System for Arabic-to-English Entity Translation". In Proceedings of ACE 2007, 2007. 90.
51. Davis, James N. et Mary Ann Lyman-Hager (2000) « Développement d'un ALAO convivial en lecture : étude de cas », in Duquette & Laurier, op. cit. : p. 139-158.
52. Debili, F. Achour, H. et Souici, E. 2002. "La langue arabe et l'ordinateur : de l'étiquetage grammatical à la voyellation automatique". Correspondances de l'IRMC, N°71, pp. 10-28, 2002.
53. Deerwester, S. T., Dumais, G., and Harshman, R. 1990. « Indexing by latent semantic analysis », Journal of the American Society for Information Science, vol. 41, n° 6, 1990, p. 391–407
54. Defays, J. M., et Deltour, S. 2003. "Le français langue étrangère et seconde. Enseignement et apprentissage". Liège : Mardaga.
55. Dejean, C., Fortun, M., Massot, C., Pottier, V., Poulard, F. et Vernier, M. (2010). " TALN 2010 , Montréal, 19–23 juillet 2010.
56. DeLaHunt, J. 2007. "Unicode and PHP: A Gentle Introduction", php|architect, Marco Tabini & Associates, 2007, Vol 6 Issue 5, pp. 38-49.
57. Desmet, P. 2006. L'enseignement/apprentissage des langues à l'ère du numérique : tendances récentes et défis. Revue française de linguistique appliquée, 11(1), 119–138.
58. Diab, M. 2009. " Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking". 2nd International Conference on Arabic Language Resources and Tools, 2009.
59. Diab, M., Hacioglu, K., and Jurafsky, D. 2004. "Automatic tagging of Arabic text: From raw text to base phrase chunks". NAACL-HLT, pages 149–152, Boston, USA, 2004.
60. Dichy, J. 2000. "Morphosyntactic specifiers to be associated to Arabic Lexical Entries - Methodological and Theoretical Aspects", Proceedings of the ACIDA' 2000 conference, Monastir (Tunisia), 22-24 March 2000, Corpora and Natural Language Processing vol., pp 55-60, 2000.

61. Ditters, E. 2001. "The description of modern standard Arabic syntax in terms of functions and categories". Langues et Littératures du Monde Arabe, 2:115–151, 2001.
62. Dzikovska, M., Steinhauser, N., Farrow, E., Moore, J., and Campbell, G. 2014. "BEETLE II: Deep Natural Language Understanding and Automatic Feedback Generation for Intelligent Tutoring in Basic Electricity and Electronics". Published in Springer Science, Business Media New York 2014.
63. Eid, M., Mansour, M., and El Saddik, A. 2007. "A haptic multimedia handwriting learning system". Proceedings of the international workshop on Educational multimedia and multimedia education Pages 103-108.
64. Elanwar, R.I., Rashwan, M.A., and Mashali, S.A. 2007. "Simultaneous segmentation and recognition of Arabic characters in an unconstrained on-line cursive handwritten document". Proceedings of World Academy of Science, Engineering and Technology (WASET), International conference on Machine learning and Pattern Recognition MLPR2007, vol. 23, pp. 288–291, Germany (2007).
65. Ellis, R. 2004. "The study of second language acquisition". Oxford applied linguistics. OUP 2004
66. Farghaly, A. et Dichy, J. 2003. "Roots & Patterns VS Stems plus Grammair-Lexis specification: On what basis should a multilingual lexical database centred on arabic be built? ". Acte de la 9ème MT conference, Workshop on Machine translation for semitic language: issues and approaches ; New Orleans, Louisiana, USA .
67. Fellbaum, C., Alkhalifa, M, Black, W., Elkateb, S., Rodríguez, H., and Vossen, P. 2006. "Introducing the Arabic WordNet project". Global Wordnet Conference, Jeju Island, Korea, January, 2006.
68. Felshin, S. 1995. "The Athena Language Learning Project NLP System: A Multilingual System for Conversation-Based Language Learning". Intelligent Language Tutors. Mahwah, NJ : Erlbaum.
69. Gale, L. E. Macario, M., and Dígame, I. 1989. "Developing interactive video for language instruction". Modern Technology in Foreign Language Education, Lincolnwood, Ill.: National Textbook, pp. 235–48 - 1989.
70. Ganapathiraju, A. H., and Picone, J. 2004. "Applications of support vector machines to speech recognition". IEEE Transactions on Signal Processing, 52, 2348 - 2355.
71. Garrett, Nina (1998) “Where do research and practice meet? Developing a discipline”, ReCALL 10-1, p. 7-12.
72. Garrot, E. 2008. "Etude du tutorat in Plate-forme support à l’Interconnexion de Communautés de Pratique (ICP) ". Application au tutorat avec TE-Cap, Chapitre 1. Thèse de Doctorat en informatique, INSA Lyon 2008.
73. Ghods, V., and Kabir, E. 2010. "Feature extraction for online Farsi characters". ICFHR, 2010 12th International Conference on Frontiers in Handwriting Recognition, pp. 477–482.

74. Golonka, E.M., Bowles, A.R., Frank, V.M., Richardson, D.L., Freynik, S., 2014. Technologies for foreign language learning: A review of technology types and their effectiveness. Computer Assisted Language Learning 27, 70–105.
75. Guéraud, V., Adam, J-M., Pernin, J-P., Calvary, G., et David, J-P. 2004. "L'exploitation d'Objets Pédagogiques Interactifs à distance : le projet FORMID". STICEF, vol. 11, 2004.
76. Guyon, I., Schomaker, L., Plamondon, R., Liberman, M., and Janet, S. 1994. "UNIPEN project of on-line data exchange and recognizer benchmarks". International Conference on Pattern Recognition, pages 29–33, Jerusalem, Israel, October.
77. Gaudin, C., Lallican, P. M., Binter, P., et Knerr, S. 1999. "The IRESTE On/Off (IRONOFF) Dual Handwriting Database", p. 455, 1999.
78. Habash, N. 2010. "Introduction to Arabic Natural Language Processing". Morgan & Claypool Publishers.
79. Halavati, R., Jamzad, M. et Soleymani, M. 2005."A novel approach to Persian online hand writing recognition". Trans. Eng. Comput. Technol. 6, 232–236 (2005).
80. Hamada, S. 2009. "Morphological Analyzers for Arabic". Proceedings of the workshop of morphological analyzer experts for Arabic language. Organized by Arab League Educational, Cultural and Scientific Organization (ALECSO), King Abdul-Aziz City of Science and Technology (KACST) and Arabic Language Academy. Damascus, Syria.
81. Harrington, P. (1993). Sigmoid transfer functions in backpropagation neural networks. nal. Chem., 1993, 65 (15), pp 2167–2168. DOI: 10.1021/ac00063a042.
82. Hegazi, A.G., El Sanousi, A., and Kheir El Din A.W. 1989. "Influence of locally isolated avian reovirus /86 on antibody response and delayed hypersensitivity of chicken vaccinated with Newcastle disease virus.J". Egypt. Vet. Med. Ass. 49, (2): 731 -740. 17.
83. Heift, T. 2003. "Multiple Learner Errors and Meaningful Feedback: A Challenge for ICALL Systems". In CALICO 2003: 533-549.
84. Heift, T., and Schulze, M. 2003. "Error Analysis and Error Correction in Computer-Assisted Language Learning". Special issue of CALICO 2003.
85. Hérault, J. et Jutten, C. "1994. Réseaux neuronaux et traitement du signal" Hermes, 1994.
86. Huang, B.Q., Zhang, Y., and Kechadi, M.T. 2007. "Preprocessing techniques for online handwriting recognition". Seventh International Conference on Intelligent Systems Design and Applications, IEEE Computer Society. pp. 793–800.
87. Jaeger, S., Manke. S., Reichert, J., and Waibel, A. 2000. "On-Line Handwriting Recognition: The NPen++ Recognizer". In International Journal on Document Analysis and Recognition (IJDAR'OO), volume 3, pages 169-180, 2000.
88. Jain, A. K., Duin, R.P.W., et Mao, J. 2000. "Statistical Pattern Recognition". A Review. IEEE Trans. on PAMI, 22, 4-37.

89. Jouini, B., Kherallah, M., et Alimi, M.A. 2003. "A new approach for online visual encoding and recognition of handwriting script by using neural network system". 6th International Conference on. Artificial Neural Nets and Genetic Algorithms, pp. 161–166. Springer, Vienna.
90. Josep, M., et Habash, N. 2008. "Using Shallow Syntax Information to Improve Word Alignment and Reordering for SMT". In Proceedings of the Third Workshop on Statistical Machine Translation, pages 53–61, Columbus, Ohio, June 2008. Association for Computational Linguistics. DOI: 10.3115/1626394.1626401 90, 112, 123
91. Jung, U.O.H. 2005. "CALL: past, present and future - a bibliometric approach". ReCALL, vol. 17, 1. pp. 4-17.
92. Kadri, Y., et Benyamina, A. 1992. "Système d'analyse syntaxico-semantique du langage arabe". Mémoire d'ingénieur, université d'Oran Essénia, 1992. ********
93. Kaleidoscpe. 2005. "Network of Excellence Kaleidoscope". http://www.noe-kaleidoscope.org/
94. Kammoun, N., Hadrich Belguith, L. and Ben Hamadou, A. « The MORPH2 new version: A robust morphological analyzer for Arabic texts». JADT'2010, 9-11 June 2010, Rome, Italy.
95. Karin, C, R. 2005. "A Reference Grammar of Modern Standard Arabic". Cambridge University Press. ISBN: 0521777712. 736 pages, 2005.
96. Kavalieratou, E., Fakotakis, N., and Kokkinakis, G. 2002. "An unconstrained handwriting recognition system". International Journal on Document Analysis and Recognition 4, 226–242.
97. Keskes, I., Ben Amara, F,. Zitoune and Hadrich Belguith, L. (2014). "Splitting Arabic texts into elementary discourse units". ACM Transactions on Asian Language Information Processing (TALIP) Volume 13, Issue 2, June 2014, Article No. 9, 23 pages.
98. Khafaji, R. 2001. "Punctuation Marks in original Arabic texts". Zeitschrift fur Arabische Linguistik 40(2001): 7-24.
99. Khan, L. R. 2000. "Ontology-based information selection". PhD Thesis, University of Southern California, 2000.
100. Kherallah, M., Bouri, F., and Alimi, A. 2009. "On-line Arabic handwriting recognition system based on visual encoding and genetic algorithm". Appl. Artif. Intell. 22(1), 153–170.
101. Khoufi, N., Aloulou, C. and Lamia Hadrich Belguith. (2015). "Parsing Arabic using induced probabilistic context free grammar", International Journal of Speech Technology, Springer, DOI: 10.1007/s10772-015-9300-x
102. Kilickaya, F. 2007. "The Effect of Computer Assisted Language Learning on Turkisk Learners". Achievement on The TOEFL Exam.
103. Kiraz, G, A. 1996. "Analysis of the Arabic Broken Plural and Diminutive". In Proceedings of the 5th International Conference and Exhibition on Multi-Lingual Computing (ICEMCO96), Cambridge, UK.
104. Kongrith, K., Maddux, C.D., 2005. Online Learning as a Demonstration of Type II Technology: Second-Language Acquisition. Computers in the Schools 22, 97–110.

105. Kraif, Olivier, Antoniadis Georges Echinard Sandra, Loiseau, Mathieu, Lebarbé, Thomas et Ponton, Claude. 2004. "NLP Tools for CALL: the Simpler the Better". In Proceedings of InSTIL/ ICALL2004: NLP and Speech Technologies in Advanced Language Learning Systems, Venice.
106. Khoja, S. 2001. "APT: Arabic part-of speech tagger". Proceeding of student workshop at the 2nd meeting of the NAACL, (NAACL'01), Carnegie Mellon University, Pennsylvania, 2001.
107. Longcamp, M., Zerbato-Poudou, M.T., and Velay, J.L., 2005. "The influence of writing practice on letter recognition in preschool children: A comparison between handwriting and typing". Acta Psychologica 119 (1), 67–79.
108. Laforcade, P., et Abedmouleh, A. 2012. "Improving the design of courses thanks to graphical and external dedicated languages: a Moodle experimentation". In: Moodle Research Conference 2012, Heraklion, Greece, 14-15 septembre 2012, p. 94-101. **
109. Laporte E. 2000. "Mot et niveau lexical". Jean-marie pierre : Ingenierie des langues, 25-46.
110. Larkey, L. S., Ballesteros, L., and Connell, M. 2002. Improving Stemming for Arabic Information Retrieval: Light Stemming and Cooccurrence Analysis, Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), Tampere, Finland, August 2002, pp. 275–282.
111. Lebrun, M. 2007. "Théories et méthodes pédagogiques pour enseigner et apprendre : Quelle place pour les TIC dans l'éducation ?". 2éme édition revue, De Boeck, Bruxelle.
112. LeCun, Y., Bottou, L., Orr, G., and Muller, K. 2001. "Efficient BackProp" Neural Networks: Tricks of the trade 2001.
113. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. "Gradient-Based Leaming Applied to Document Recognition", Proceedings of the IEEE, vol. 86, no. 11, pages 2278-2324, Nov. 1998.
114. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. 1992. "Handwritten digit recognition with a back¬propagation network". In Lisboa P.G.J., editor, Neural Netwotks, current applications. Chappman and Hall, 1992.
115. LeCun, Y. 1987. "Modeles connexionnistes de l'apprentissage (connectionist learning models) ". 277 pages, June 1987.
116. Lefevre, P. 2004. "La recherche d'informations, du texte intégral au thésaurus". Hermès,Paris.
117. Levy, M. 1997. "CALL: Context and conceptualization". Oxford: Oxford University Press, p 1.
118. Loiseau, M. 2009. "Élaboration d'un modèle pour une base de textes indexée pédagogiquement pour l'enseignement des langues", Ph. D. thesis: Stendhal University, Grenoble 3: 2009
119. Loiseau, M., Antoniadis, G., et Ponton, C. 2005. "Pedagogical text indexation and exploitation for language teaching". Third International Conference on Multimedia and Information & Communication Technologies in Education (MICTE2005). Badajoz.
120. Loiseau, M. 2004. "La description de ressources pédagogiques : état de l'art et application aux ressources textuelles pour l'enseignement des langues", Proceedings of workshop "TAL et apprentissage

des langues", Grenoble, France, http://www.u-grenoble3.fr/lidilem/talal/actes/JourneeTALAL-041022-loiseau.pdf

121. Lorette, G. 2013. "Handwriting Recognition or Reading Situation". At the Dawn Of The 3rd Millenium", in Advances in Handwriting Recognition, World Scientific Publications, pp. 3-13.
122. Lunda, K., et Baker, M.J. 2000. "Interprétations par des enseignants des interactions d'élèves médiatisées par ordinateur", 3ème Colloque international Recherche et formation des enseignants : Didactique des disciplines et formation des enseignants : approche anthropologique, Marseille, France, 2000.
123. Lunda, K., et Baker, M.J. 1999. "Teachers' collaborative interpretations of students' computer-mediated collaborative problem solving interactions", Proceedings of the International Conference on Artificial Intelligence and Education, Le Mans, Juillet 1999. S.P. Lajoie & M. Vivet (Eds.) Artificial Intelligence in Education, pp. 147-154 Amsterdam: IOS Press.
124. Maamouri, M., Bies, A., Buckwalter, T., and Mekki, W. 2004. "The penn Arabic Treebank: Building a large-scale annotated Arabic corpus". NEMLAR conference on Arabic language resources and tools. pp. 102-109, September 2004.
125. MacNeill, D. 1995. "Handwriting Recognizer Specification Matrix". Pen Computing Magazine. August 1995, 32–33.
126. S. Manke, S., Finke, M., and Waibel, A. 1996. "A Fast Search Technique for Large Vocabulary On-Line Handwriting Recognition". Int. Workshop on Frontiers in Handwriting Recognition (IWFHR), Colchester,1996
127. Manning, C., Raghavan, P., and Schuetze, H. 2008. "Introduction to Information Retrieval". Cambridge, UK: Cambridge University Press, 2008.
128. Maraoui, M., Zrigui, M., and Antoniadis, G. 2006. "Use of NLP Tools in CALL System for Arabic". International Journal of Computer Processing of Languages.
129. Mars, A., et Antoniadis, G. "Système d'aide à la création des activités pour l'ALAO". (2014 b).25es Journées francophones d'Ingénierie des Connaissances Clermont Ferrand, du 12 au 16 mai 2014.
130. Mars, A. and Antoniadis, G. 2014. "Arabic language learning system using NLP tools and pedagogically indexed texts". International Conference on Computer Science and Engineering 2014 (ICCSE'2014), 13-15 juin, Hammamet, Tunisia.
131. Mars, A., et Antoniadis, G. 2015. "Handwriting recognition system for Arabic language learning" WCITCA'2014 World Congress on Information Technology and Computer Application, HAMMAMET 2015 – International Journal N&N Global technology.
132. Mars, A. and Antoniadis, G. 2016. "Arabic on-line handwriting recognition system using neural network". International Journal of Artificial Intelligence and Applications (IJAIA), Vol. 7, No. 5, September 2016 p51-59.
133. Martineau, T. Nakamura, L., et Stavroula, V. 2009. "Annotation et normalisation des entités nommées". Arena Romanistica. vol. 4:234–243.

134. Märgner, V., and El Abed, H. 2008. "Arabic and Chinese handwriting recognition. Databases and Competitions: Strategies to Improve Arabic Recognition Systems". vol. 4768. Springer, LNCS (2008)
135. Meftouh, K. Smaïli, K., et Laskri, M.T. 2007. "Constitution d'un corpus de la langue arabe à partir du Web". CITALA '07, Rabat, Morocco, 17-18 juin 2007.
136. Mezghani, N., Mitiche, A., Cheriet, M. 2008. "Bayes classification of online Arabic characters by Gibbs modeling of class conditional densities". IEEE Trans. Pattern Anal. Mach. Intell. 30(7), 1121–1131.
137. Michael, N. 2014. "Improving Arabic Tokenization and POS Tagging Using Morphological Analyzer". Advanced Machine Learning Technologies and Applications (2014) Mitkov. The Oxford Handbook of Computational Linguistics. Oxford University, Oxford.
138. Mostow, J., and Beck, J. E. 2003. "Project LISTEN's Reading Tutor: Interactive Event Description". Supplemental Proceedings of the Tenth International Conference on Artificial Intelligence in Education (AIED2003), Sydney, Australia.
139. Moukdad, H., and Large, A. 2001. "Information Retrieval from Full-Text Arabic Database: Can Search Engine Designed for English do the Job", Libri, 2001, pp. 63- 74.
140. Murphy, L., et Hurd, S. 2011. "Fostering learner autonomy and motivation in blended teaching". In: Nicolson, Margaret; Murphy, Linda and Southgate, Margaret eds. Language Teaching in Blended Contexts. Edinburgh, U.K.: Dunedin Academic Press Ltd, pp. 43–56.
141. Murray, J. H., Morgenstern, D., et Furstenberg, G. "The Athena Language Learning Project: design issues for the next generation of computer based language learning tools". Technology in Foreign Language Education, Lincolnwood, Ill.: National Textbook, pp. 97–118 - 1989.
142. Namer, F. 2000. "FLEMM : Un analyseur flexionnel du français à base de règles". TAL, 41 (2): 523-547.
143. Nasr, A., Béchet, F., and Volanschi A. 2004. Tagging with Hidden Markov Models Using Ambiguous Tags. In Proceedings of COLING 2004, COLING '04, Stroudsburg, PA, USA. QUASTHOFF U.
144. Nerbonne, J. 2003. "Natural Language Processing in Computer-Aided Language Learning". DCU Machine Translation System for IWSLT 2006. In Proc. of the International Workshop on Spoken Language Translation, pages 31–36, Kyoto, Japan, 2006. 90
145. Nicolas, S., and Andy, W. 2006. Machine Translation System for IWSLT2006. In Proc. of the International Workshop on Spoken Language Translation, pages 31–36, Kyoto, Japan, 2006.
146. Nielsen, H., and Carlsen, M. 2003. "Interactive Arabic grammar on the Internet: Problems and solutions". Computer Assisted Language Learning (CALL): An International Journal, 16(1), 95 –112-2003.
147. Ouersighni, R. 2001. "A major offshoot of the DIINAR-MBC project: AraParse, a morphosyntactic analyzer for unvowelled Arabic texts". ACL/EACL 2001 Workshop on Arabic Language Processing, Toulouse July 2001, pp. 9-16.
148. Parkvall, Mikael. 2010. "l'Institutionen för lingvistik", Université de Stockholm 2010 : http://www.tlfq.ulaval.ca/axl/Langues/2vital_inter_arabe.htm

149. Pascal, D., and Benoît, S. 2009. "Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort". In Proceedings of PACLIC 2009, Hong-Kong, China.
150. Piton, D. Maurel, C., and Belleil, A. 1999. "The Prolex Data Base: Toponyms and gentiles for NLP". International Workshop on Applications of Natural Language to Data Bases (NLDB'99), 233–237.
151. Poulsen, R. 2004. "Tutoring Bilingual Students with an Automated Reading Tutor That Listens: Results of a Two-Month Pilot Study". Master's Thesis, DePaul University, Chicago, IL.
152. Pressey, S. L. "A machine for automatic teaching of drill material". School and Society, 25, pp 549-552 - 1927.
153. Power, M. 2002. "Générations d'enseignement à distance, technologies éducatives et médiatisation de l'enseignement supérieur". in Journal of distance éducation (Revue de l'éducation à distance), ACED, Vol. 17, N°2, Ottawa (Canada), 2002
154. Pustejovsky, J. 1995. "The Generative Lexicon". Cambridge (MA), MIT Press.
155. Quinlan, J. R. 1986. "Induction of decision trees". Machine Learning, 81-106.
156. Ravichandran, T. 2000. "Computer Assisted Language Learning" .In the Perspective of Interactive Approach Advantages and Apprehensions.
157. Renié, Delphine (2000) « Apport d'une trace informatique dans l'analyse du processus d'apprentissage d'une langue seconde ou étrangère », in Duquette & Laurier, op. cit. : p. 139-158.
158. Renié, D. 1995. Modélisation informatique de l'acquisition des interrogatives directes en français langue seconde dans leur dimension pragmatique, proposition d'un environnement offrant un apprentissage collaboratif : ELEONORE. Thèse de doctorat, université Clermont II, Clermont-Ferrand.
159. Robbes, B. 2009. "La pédagogie différenciée : historique, problématique, cadre conceptuel et méthodologie de mise en œuvre", janvier 2009. http://www.meirieu.com/ECHANGES/brunorobbes pedagogiedifferenciee.pdf, p. 1–34
160. Rossi, F. 1995. "Second Differentials in Arbitrary F eed-Forward Neural Networks". Technical report, Thomson CSF ISDC, October 1995.
161. Rodriguez, H., Farwell, D., Farreres, J., Bertran, M., Alkhalifa, M., M., and Martí, A. 2008. "Arabic WordNet: Semi-automatic Extensions using Bayesian Inference". Proceedings of the the 6th Conference on Language Resources and Evaluation LREC2008. Marrakech (Morocco), May 2008.
162. Rumelhart, D. E., Hinton, G.E., and Williams, R. J. 1986. "Learning Internal Representation by Error Propagation", Cambridge, MIT Press.
163. Sagot, B., et Fišer, D. 2012. "Automatic extension of WOLF". In Actes de la 12ème Global Wordnet Conference, Matsue, Japon.
164. Sagot, B. 2010. "The Lefff, a Freely Available and Large-coverage Morphological and Syntactic Lexicon for French". LREC'10, Valletta, Malta.
165. Sanchez, F. and Nieto, A. F. "Development of a Spanish Version of the Xerox Tagger". Universidad Autonoma de Madrid, 1995.

166. Sanderson, M. 1997. "Word Sense Disambiguation and Information Retrieval", Technical Report, Dpt of CS at the Univ. of Glasgow, UK.

167. Selva, T., et Chanier, T. 2000. "Génération automatique d'activités lexicales dans le système ALEXIA". Sciences et Techniques Éducatives (STE), vol. 7, 2. pp. 385-412.http://archive-edutice.ccsd.cnrs.fr/edutice-00000282

168. Shaalan, K., Talhami, E.H. "Arabic Error Feedback in an Online Arabic Learning System". Advances in Natural Language Proccesing, Research in Computing Science 18, pp. 203-212, 2006.

169. Shalaan, K. 2005. "An Intelligent Computer Assisted Language Learning System for Arabic Learners". Computer Assisted Language Learning: An International Journal, 18(1 & 2), pp. 81-108, Taylor & Francis Group Ltd 2005.

170. Shaalan K. "Development of Computer Assisted Language Learning System for Arabic Using Natural Language Processing Techniques", Egyptian Informatics Journal, vol. 4, no. 2: Faculty of Comptuers and Information, pp. 131–155, dec, 2003.

171. Shintani, N. and Ellis, R. "2015. Does language analytical ability mediate the effect of written feedback on grammatical accuracy in second language writing?". System, 49, 110-119 - 2015.

172. Schmid, H. 1994. "Probabilistic Part-of-Speech Tagging Using Decision Trees". International Conference on New Methods in Language Processing, Manchester, UK.

173. Sarle, W.S. 1997. Neural Network FAQ, part 2 : Part 2: "Leaming, What are batch, incremental, on-line, off-line, deterministic, stochastic, adaptive, instantaneous, pattern, constructive, and sequential leaming?", periodic posting to the Usenet newsgroup comp.ai.neural-nets.

174. Soudi, A., Bosch, A., and Neumann, G. 2007. "Arabic Computational Morphology. Knowledge-based and Empirical Methods". Dordrecht, The Netherlands: Springer.

175. Soudi, A., Cavalli-Sforza, V., and Jamari, A. 2001. "A Computational Lexeme-Based Treatment of Arabic Morphology". ACL/EACL 2001 Workshop on Arabic NLP. Toulouse, France, Friday 6 July 2001.

176. Sternby, J., Morwing, J., Andersson, J., and Friberg, C. 2009. "On-line Arabic handwriting recognition with templates". Pattern Recognition. J. 42(12), 3278–3286, New Frontiers in Handwriting Recognition.

177. Tappert, C. C., Suen, C. Y. et Wakahara, T. 1994. "The state of the art in on-line handwriting recognition". IEEE Transactions on Pattern Analysis and Artificial Intelligence, 12, 787 - 808.

178. Tarrier. 1991. "A propos de sociolinguistique de l'arabe, présentation de quelques difficultés", Bulletin d'études Orientales XLIII, Damas, Publications de l'Institut Français de Damas.

179. Tay, Y. H. 2002. "Off-line Handwriting Recognition using artificial Neural Network and Hidden Markov Model"- PhD University of Nantes and University Technology Malaysia, 2002.

180. Thabet, N. 2004. "Stemming the Qur'an". COLING 2004, Workshop on computational approaches to Arabic script-based languages. August 28, 2004, pp. 85-88.

181. Teutsch, Ph., Bourdet, JF., and Gueye, O. 2004. "Perception de la situation d'apprentissage par le tuteur en ligne", conférence TICE'2004, Compiègne (France), 2004, p. 59-66.
182. Thibeault, M. " La catégorisation grammaticale automatique : Adaptation du catégoriseur de Brill au français et modification de l'approche", Faculté des lettres université LAVAL Québec. 2004.
183. Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. 2003. "Feature-rich part-of-speech tagging with a cyclic dependency network". In Proceedings of ACL 2003.
184. Vapnik, V. and Vladimir, N. 1995. "The Nature of Statistical Learning Theory". Springer.
185. Viard-Gaudin, C., Lallican, P.M., Knerr, S., and Binter, P. 1999. "The IRESTE ON-OFF (IRONOFF) Handwritten Image Database", ICDAR'99, International Conference on Document Analysis and Recognition, pages 455-458, Bangalore, 1999.
186. Volodina, E., Borin, L., Loftsson, H., Arnbjörnsdóttir, B., and Leifsson, G. 2012. "Waste not want not: Towards a system architecture for ICALL based on NLP component re-use". Workshop on NLP in Computer-Assisted Language Learning.
187. Wahl, H., and Winiwarter, W. 2012. A Prototypical Implementation of the Intelligent Integrated Computer-Assisted Language Learning (iiCALL) Environment ALL) EnvironmentAdvances in Web-Based Learning - ICWL 2012: 11th International Conference, Sinaia, Romania, September 2-4, 2012.
188. Walter, D., and Jakub, Z. 1996. "MBT A Memory Based Part of Speech Tagger Generator" Computational TilburgLinguistics and AI.
189. Warschauer, M. 1996. "Computer-assisted language learning: an introduction". In Fotos S. (ed.) Multimedia language teaching, Tokyo: Logos International, A copy of this article is located at the ICT4LT, 1996.
190. Xu, J., Fraser, A., and Weischedel R. 2002. "Empirical Studies in Strategies for Arabic Retrieval". Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), August 11-15, 2002, pp. 269-274.
191. Yaeger, L., Brandyn, W., and Richard, L. 1996. "Combining Neural Networks and Context-Driven Search for On-Line, Printed Handwriting Recognition". For the Fifth International Workshop on the Frontiers of Handwriting Recognition. University of Essex, England, September 1996.
192. Yin, F., Wang, Q., Zhang, X., and Liu, C. 2013. "Chinese Handwriting Recognition". Int'l Conf. on Document Analysis and Recognition, 2013.
193. Zrigui, M. 2008. Contribution au traitement automatique de l'Arabe. HDR en informatique, Stendhal University, Grenoble 3, Franc.

Annexes

Annexe A : accents arabes

I. En version isolé

- diacritiques plutôt facultatifs utilisés la plupart du temps en version seule:

◌َ	fatha	064E
◌ُ	damma	064F
◌ِ	kasra	0650
◌ً	fathatan	064B
◌ٌ	dammatan	064C
◌ٍ	kasratan	064D

- diacritiques plutôt obligatoires utilisés plutôt rarement dans leur version seule:

◌ّ	shadda	0651
◌ْ	soukoune	0652
◌ٓ	maddah	0653
◌ٔ	hamza au-dessus	0654
◌ٕ	hamza en dessous	0655

II. En version combinée

- diacritiques plutôt obligatoires utilisés la plupart du temps en version combinée (code unique pour les deux entités):

آ	alef avec madda au-dessus	0622
أ	alef avec hamza au-dessus	0623
ؤ	waw avec hamza au-dessus	0624
إ	alef avev hamza en dessous	0625
ئ	yeh avec hamza au-dessus	0626

- diacritiques sur ligatures (reprenant les codes de la partie précédente):

لآ	ligature lam+alef avec madda au-dessus de alef	0644+0622
لأ	ligature lam+alef avec hamza au-dessus de alef	0644+0623
لإ	ligature lam+alef avec hamza en dessous de alef	0644+0625

III. Diacritiques obligatoires attachés aux lettres

ب	lettre beh	0628
ت	lettre teh	062A
ث	lettre theh	062B
ج	lettre jeem	062C
خ	lettre khah	062E
ذ	lettre thal	0630
ز	lettre zain	0632
ش	lettre sheen	0634
ض	lettre dad	0636

ظ	lettre zah	0638
غ	lettre ghain	063A
ف	lettre feh	0641
ق	lettre qaf	0642
ك	lettre kaf	0643
ن	lettre noon	0646
ي	lettre yeh	064A
ة	lettre teh marbuta	0629

2.2 Annexe B

Pays où l'arabe est langue officielle

Pays	Population	Langue(s) officielle(s)
Égypte	60,4 M	arabe
Algérie	29,3 M	arabe
Soudan	29,1 M	arabe
Maroc	29,1 M	arabe
Irak	22,4 M	arabe
Arabie Saoudite	17,1 M	arabe
Yémen	16,1 M	arabe
Syrie	14,9 M	arabe
Tunisie	9,0 M	arabe
Tchad	6,4 M	français/arabe
Israël	5,5 M	hébreu/arabe
Libye	5,4 M	arabe
Jordanie	3,8 M	arabe
Érythrée	3,6 M	arabe/tigrinia

Liban	3,2 M	arabe
Émirats arabes unis	2,1 M	arabe
Oman	2,0 M	arabe
Mauritanie	1,3 M	français/arabe
Koweït	1,3 M	arabe
Barhreïn	601 000	arabe
Comores	590 000	français/arabe
Qatar	516 000	arabe
Djibouti	473 000	français/arabe
Malte	360 000	anglais/maltais (arabe)
Total :	265 millions environ	

2.3 Annexe C

Lettres arabes et ses translitération en Buckwalter

ettre	Description	Buckwalter	ettre	Description	Buckwalter
	Lettre Hamza	‘		Lettre Dal	D
	Lettre Alef avec Madda	ا		Lettre Thal	*
	Lettre Alef avec Hamza au-dessus	>		Lettre Reh	r
	Lettre Waw avec Hamza au-dessus	&		Lettre Zain	z
	Lettre Alef avec Hamza au-dessous	>		Lettre Seen	s
	Lettre Yeh avec Hamza au-dessus	}		Lettre Sheen	$
	Lettre Alef	A		Lettre Sad	S
	Lettre Beh	b		Lettre Dad	D
	Lettre Teh Marbouta	p		Lettre Tah	T
	Lettre Teh	t		Lettre Zah	Z

	Lettre Theh	v		Lettre Ain	E
	Lettre Jeem	j		Lettre Ghain	g
	Lettre Hah	H		Lettre Feh	f
	Lettre Khah	x		Lettre Qaf	q
	Lettre Kef	k		Lettre Heh	h
	Lettre Lam	l		Lettre Waw	w
	Lettre Meem	m		Lettre Alef Maksura	Y
	Lettre Noon	n		Lettre Yeh	y

Extrait de Corpus Quran :

1. وَلَقَدْ خَلَقْنَا الإِنسَانَ وَنَعْلَمُ مَا تُوَسْوِسُ بِهِ نَفْسُهُ وَنَحْنُ أَقْرَبُ إِلَيْهِ مِنْ حَبْلِ الْوَرِيدِ
2. إِذْ يَتَلَقَّى الْمُتَلَقِّيَانِ عَنِ الْيَمِينِ وَعَنِ الشِّمَالِ قَعِيدٌ
3. مَا يَلْفِظُ مِن قَوْلٍ إِلاَّ لَدَيْهِ رَقِيبٌ عَتِيدٌ
4. وَجَاءَتْ سَكْرَةُ الْمَوْتِ بِالْحَقِّ ذَلِكَ مَا كُنتَ مِنْهُ تَحِيدُ
5. وَنُفِخَ فِي الصُّورِ ذَلِكَ يَوْمُ الْوَعِيدِ
6. وَجَاءَتْ كُلُّ نَفْسٍ مَّعَهَا سَائِقٌ وَشَهِيدٌ
7. لَقَدْ كُنتَ فِي غَفْلَةٍ مِّنْ هَذَا فَكَشَفْنَا عَنكَ غِطَاءَكَ فَبَصَرُكَ الْيَوْمَ حَدِيدٌ
8. وَقَالَ قَرِينُهُ هَذَا مَا لَدَيَّ عَتِيدٌ
9. أَلْقِيَا فِي جَهَنَّمَ كُلَّ كَفَّارٍ عَنِيدٍ
10. مَّنَّاعٍ لِّلْخَيْرِ مُعْتَدٍ مُّرِيبٍ
11. الَّذِي جَعَلَ مَعَ اللَّهِ إِلَهًا آخَرَ فَأَلْقِيَاهُ فِي الْعَذَابِ الشَّدِيدِ
12. قَالَ قَرِينُهُ رَبَّنَا مَا أَطْغَيْتُهُ وَلَكِن كَانَ فِي ضَلالٍ بَعِيدٍ

13.قَالَ لا تَخْتَصِمُوا لَدَيَّ وَقَدْ قَدَّمْتُ إِلَيْكُم بِالْوَعِيدِ

14.مَا يُبَدَّلُ الْقَوْلُ لَدَيَّ وَمَا أَنَا بِظَلاَّمٍ لِّلْعَبِيدِ

15.يَوْمَ نَقُولُ لِجَهَنَّمَ هَلِ امْتَلَأْتِ وَتَقُولُ هَلْ مِن مَّزِيدٍ

16.وَأُزْلِفَتِ الْجَنَّةُ لِلْمُتَّقِينَ غَيْرَ بَعِيدٍ

17.هَذَا مَا تُوعَدُونَ لِكُلِّ أَوَّابٍ حَفِيظٍ

18.مَنْ خَشِيَ الرَّحْمَن بِالْغَيْبِ وَجَاءَ بِقَلْبٍ مُّنِيبٍ

19.ادْخُلُوهَا بِسَلامٍ ذَلِكَ يَوْمُ الْخُلُودِ

20.لَهُم مَّا يَشَاؤُونَ فِيهَا وَلَدَيْنَا مَزِيدٌ