الجـــــــــمـهـوريـــــة الجـــــــــزائريـة الديمـــــــــــقراطيـة الشـــــــعبيـة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التـــــــــــــعليـم العـــــــــالي و البـــــــــحث العـــــــــلمـي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جـــــــــامعة غـــردايـــــــة
Université de Ghardaia
كليـة العلـــوم و التكنولـوجيـا
Faculté des Sciences et de Technologie
قسم الرياضيات و الإعلام الآلي
Département des Mathématiques et Informatique

MÉMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Informatique

Spécialité : Systèmes Intelligents pour l'Extraction de Connaissances

Par : Khaled MOULAY OMAR & Abdallah TOUATI

Sujet

Étude de similarité des séquences "cas des noms propres"

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

M. Slimane OULED ENAOUI	MCB	Univ. Ghardaia	Président
M. A.ADJILA	MAA	Univ. Ghardaia	Examinateur
M. K.KECHIDA	MAA	Univ. Ghardaia	Examinateur
M. Slimane BELLAOUAR	MCB	Univ. Ghardaia	Directeur de mémoire
M. Djelloul ZIADI	Prof	Univ. Rouen	Co-Directeur de mémoire

Année Universitaire 2018/2019

ملخص

نحن نعيش في عصر ثورة المعلومات والانفجار الرقمي ، الأمر الذي يتطلب عملاً شاقاًا وجهداً كبيراً لإدارة هذا السيل من البيانات.
تختلف هذه البيانات في العديد من الأنواع ، بما في ذلك: الرقمية والنصية والصوتية والفيديو والصور وغيرها...
في هذه المذكرة نتطرق إلى أسماء العلم الجزائرية، و هي من المعطيات التي تستعمل بشكل كبير في الإدارات العمومية و التعامل مع هذا النوع من المعطيات يفرض العديد من المشاكل و التحديات من بينها مشكل تشابه الأسماء.

لمعالجة هذا الإشكال استعملنا تقنية علم الأصوات كمعالجة ابتدائية للأسماء من أجل كتابة قياسية للأسماء العربية باللغة الفرنسية و بعدها طبقنا تقنية تراصف المتتاليات الحرفية، من أجل قياس مدى تشابه هذه الكتابات.
وفي الجانب التطبيقي اخترنا لغة البرمجة C# لإنشاء تطبيق يعالج الإشكالية المطروحة، و جربناه على قاعدة بيانات أنشأناها انطلاقا من معطيات إدارية في ولاية غرداية.
وصلنا إلى دقة جيدة تقدر بحوالي 77% بعتبة قبول تقدر بـ 40%، يمكن تحسينها من خلال دراسة معمقة أكثر.

كلمات مفتاحيه: اسم العلم، التشابه، علم الأصوات، تراصف المتتاليات الحرفية.

Résumé

Nous vivons à l'époque de la révolution de l'information et de l'explosion numérique, qui exige un travail acharné et des efforts considérables pour gérer ce torrent de données. Ces données varient dans de nombreux types, notamment : numériques, textes, audio, vidéo, photo , etc.
Dans ce mémoire, nous faisons référence aux noms propres, qui sont très utilisés posent de nombreux problèmes et défis, y compris le problème de la similarité des noms propres.
Pour traiter la problématique posée, nous avons utilisé la technique de phonétique comme traitement initial des noms afin de standardiser l'écriture des noms propres arabes en langue française, puis avons appliqué la technique d'alignement des séquences afin de mesurer la similarité de ces écritures.
Sur le plan pratique, nous avons choisi le langage de programmation C# pour implémenter la solution et créer une application qui traiter la problématique posée. Et nous avons testé cette application sur une base de données que nous avons créée à partir de données administratives de notre wilaya de Ghardaia.
Nous avons eu des résultats judicieux avec une précision de 77% avec un seuil d'acceptation égale à 40%, peut être améliorer par un étude approfondi .

Mots-clés : *nom propres, similarité des séquences, phonétique, alignement des séquences* .

Abstract

We live in the age of the information revolution and the digital explosion, which requires hard work and considerable effort to manage this torrent of data.
These data vary in many types, including : digital, text, audio, video, photo, etc. In this thesis, we refer to personnel names, which are widely used and this type of data poses many problems and challenges, including the problem of the similarity.
To deal with the problematic, we used the phonetic technique as initial name processing to standardize the writing of Arabic proper nouns in French, then applied the sequence alignment technique to measure the similarity of these scripts.
. In practice, we chose the C# programming language to implement the solution and create an application that addresses the problem. And we tested this application on a dataset that we created from the administrative's data of the wilaya of Ghardaia.
We had good results to a precision of 77% with an acceptance threshold equal to 40%, can be improved by a thorough study.

Keywords :*Personnel Names, Similarity, Phonetic, Sequence Alignment.*

Dédicace

Je dédie ce modeste travail à :

Mes parents Aucun hommage ne pourrait être à la hauteur de l'amour Dont ils ne cessent de me combler, Que dieu leur procure bonne santé et longue vie.

A celui que j'aime beaucoup et qui m'a soutenue tout au long de ce projet **mon épouse** et bien sur A mes jolies filles **HADIL**, **HALA**, **KHADIDJA**, **KHOULOUD** et mes sœurs, mes oncles, mes tentes et mes beaux-parents, sans oublié mes grands-parents.
A toute ma famille, et mes amis, A mon binôme Khaled et toute sa famille. Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

A tous mes collègues.

Abdallah

Dédicace

Je dédie ce modeste travail à :

Mes parents qui représentent pour moi l'exemple des sacrifices du dévouement, l'honnêteté et qui ont fait de moi ce que je suis devenu et qu'ils trouvent en ce mémoire l'expression de mon éternelle affection avec mon amour infini.

A celui que j'aime beaucoup et qui m'a soutenue tout au long de ce projet **mon épouse** et bien sur A mes frères **Touhami**, **Mohammed**, **abdelkader**,**youcef** et mes sœurs, ma tente . A toute ma famille, et mes amis, A mon binôme Abdallah et toute la famille. Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci..

A tous mes collègues.

Khaled

Dédicace

Nous remercions en premier lieu **ALLAH** le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de Mr le Professeur **Djelloul ZIADI** et docteur **Slimane BELAOUAR**, on les remercie pour la qualité de leur encadrement exceptionnel, pour leur patience, leur rigueur et leur disponibilité durant notre préparation de ce mémoire. Nos remerciements s'adressent à tous le staffe de l'institut d'informatique pour leurs aide et leurs soutien moral et ses encouragements.

Nous sommes conscients de l'honneur que nous a fait Mr **Slimane OULAD NAOUI** et en étant président du jury et Mr **A.ADJILA** et Mr **K.KECHIDA**, d'avoir accepté d'examiner ce travail.

Nos profonds remerciements vont également à toutes les personnes qui nous ont aidés et soutenue de prés ou de loin.

Remerciements

TENONS tout d'abord à remercier **Allah** le tout puissant et miséricorde dieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, nous tenons à remercier très chaleureusement **Pr. Djelloul ZIADI** qui nous a permis de bénéficier de son encadrement, sans oblier Monsieur **Slimane BELLAOUR**. Les conseils qu'il nous a prodigué, la patience, la confiance qu'il nous a témoignés ont été déterminants dans la réalisation de notre travail de recherche.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail.

Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

Nous tenons à ne remercier toute personne qui a participé de près ou de loin à l'exécution de ce modeste travail.

Table des matières

Liste des tableaux

Table des figures

List of Algorithms

Introduction générale

DE nos jours nous constatons une explosion du volume de données qui évolue de plus en plus par la création, la communication et le stockage des données par de nombreuses personnes, organisations, entreprises et administrations.
Il existe plusieurs types et catégories de données tel que les textes, les photos, les vidéos, les sons...etc. Dans ce mémoire, nous nous intéressons à une catégorie très importante et très utilisée dans les administrations, à savoir, les noms propres.Un problème posé au niveau de l'administration algérienne, les banques et le centre des chèques postaux dû à la transcription des noms propres arabes en langue française. Ceci conduit à des anomalies qui peuvent être graves. Par exemple lors des virements avec l'absence d'outils de contrôle et de comparaison de la similarité des noms concernés par l'opération , notamment avec une base de données géante tel que celle du centre des chèques postaux.Il est important de signaler que ce problème d'écriture des noms propres arabes en français ne se limite pas seulement au niveau national, mais il touche tous les pays arabes francophones tels que les pays maghrébins.
Les recherches dans ce sous-domaine linguistique ont été très actives depuis la fin des années 80. En 1988 B. Van Berkel et K. De Smedt [1] se sont intéressés à la correction orthographique, y compris celle des noms propres.
En 2006, Peter Christen [1] s'est intéressé à l'étude des caractéristiques des noms propres dans la langue anglaise et ceci afin de détecter d'éventuelles «erreurs» d'orthographe sur les noms propres.
Les variations des écritures des noms propres peuvent être causée par la saisie manuelle ou automatique (vocal, OCR). Afin de résoudre le problème de variation d'écriture, Peter Christen [1] a proposé alors plusieurs mesures de correspondance entre les noms propres :

- Encodage Phonétique (soundex ,Phonex , Phonix, NYSIIS, Double-Metaphone, FuzzySoundex).
- Similaritité : en utilisant différentes mesures : Levenshtein, Damerau-Levenshtein,Smith, La plus longue sous-séquence commune (LCS) , n- grammes, Jaro , Winkler,Jaro-Winkler.
- La combinaison « Encodage Phonétique/Similarité »afin d'améliorer la qualité de l'appariement.

L'étude menée par Peter Christen [1] a montré que les trois techniques testées concernant la similarité de documents n'ont pas conduit à un résultat satisfaisant. Nous pensons que ces

résultats sont dûs à la longueur des noms propres et qui ne doivent dépasser en moyenne vingt caractères.
Par conséquent, il est judicieux de s'intéresser plutôt à une «égalité approximative» qui respecte l'ordre.
Dans ce mémoire nous proposons une approche basée sur les règles de phonétiques et l'alignement des séquences. Dans un premier temps, nous appliquons les règles de phonétique (fonction h) pour faire rapprocher les écritures des noms propres (x=Mostapha, y= mustafa) correspondant à un même nom écrit en arabe (o= مصطفى) par :

$h(x) = [mostafa]$ et $h(y) = [mustafa]$. Dans un second temps, nous utilisons l'alignement de séquences en proposant une nouvelle distance de similarité.
Notre mémoire est organisé comme suit :
Le premier chapitre introduit les notions de phonétique nécessaires à la compréhension de la suite du document puis nous proposons un nouveau modèle de correspondance phonétique.
Dans le deuxième chapitre nous commençons par rappeler les concepts de base des séquences, la similarité entre les séquences, et l'alignement de séquences. Ensuite, nous proposons une nouvelle mesure de similarité basé sur une fonction de score que nous avons défini, suite à une étude sur la phonétique.
Le chapitre 3 est dédié à la partie implémentation de la solution proposée dans les chapitres précédents.Nous présentons le Data Set que nous avons construit et utilisé dans les tests, puis une discussion des résultats obtenus.
En fin la conclusion clôture notre mémoire avec une synthèse de travail ainsi que des perspectives.

CHAPITRE 1

INTRODUCTION À LA PHONÉTIQUE ET SES APPLICATIONS AUX SÉQUENCES

"*Le coeur a ses raisons que la raison ne connaît point.*"

BLAISE PASCAL,
Extrait des Pensées

1.1 Introduction

DANS ce chapitre, nous nous intéressons à l'étude de la phonétique des deux langues reliées à notre thème, à savoir : la langue française et la langue arabe, dans le but faire la transcription entre les deux, et ainsi normaliser l'écriture des noms propres.
La phonétique est une méthode de représentation d'une prononciation commune pour différentes écritures. Dans ce qui suit nous présentons différentes notions et définitions ayant trait à la phonétique.

1.2 Phonétique

Définition 1 *La phonétique est une représentation symbolique des notes sonores (prononciations) des mots écrits dans une langue spécifique. D'après Jean-Michel Kalmbach [2] : «En simplifiant, on peut dire que la phonétique est la science qui s'occupe de décrire dans leur ensemble tous les phénomènes phonétiques d'une langue, tout le "matériau de production sonore" : sons, intonation, perception, organes etc..».*

Donc la phonétique traite les communications orales (phones), tel que mot phone représente le son de la voix humaine pour communiquer.

1.2.1 Transcription phonétique

La transcription phonétique est une méthode de représentation conventionnelle des phones par des signes (symboles). Cependant, elle ne permet pas de décrire les paroles d'une façon complète. Ceci est dû à la multi-variétés des transcriptions selon les individus, les langues,

les dialectes...etc. Ceci conduit à l'existence d'un nombre important de variantes.
Il est impérative de trouver les caractéristiques utiles et diminuer l'ensemble des signes (symboles) de la description pour qu'elle devienne une interprétation des sons qu'on puisse comprendre. On peut distinguer deux types de transcriptions :

1. La transcription phonétique : on essaye de se rapprocher le plus possible d'une description parfaite de différents sons, en exprimant des symboles (signes) supplémentaires (auxiliaires) appelés des *signes diacritiques* comme des petits o sous une lettre (o souscrit), un háček au-dessus de la lettre [ˇ], des accents indiquant la mouillure [r̝] etc. C'est une transcription difficile à maitriser (en écriture et lecture).
2. La transcription phonologique : tient compte des traits essentiels et distinctifs et ne s'intéresse pas aux petites variations [2].

L'exemple de la table 1.2.1 suivante montre la différence entre la transcription phonétique et la transcription phonologique.

Caractère	Transcription phonétique	Transcription phonologique
é très fermé	[e̝]	/e/
é moins fermé	[e]	/e/
é un peu ouvert	[e̞]	/e/

TABLE 1.1 – Exemple de la différence entre la transcription phonétique et phonologique de la lettre “é" suivant le cas.

1.2.2 Marquage par des crochets ou des barres obliques

Par convention dans la transcription phonétique on exploite les crochets [ ]. Les barres obliques / / sont utilisées pour la transcription phonologique. Habituellement, on utilise les crochets pour tout type de transcription. Souvent, on met des transcriptions phonologiques au lieu des «transcriptions phonétiques» pour des dictionnaires ou des manuels.
Il est nécessaire de noter qu'il est impossible de faire une représentation parfaite de la transcription par des symboles graphique. Cela implique que chaque représentation est une interprétation qui n'est pas nécessairement unique [2].

1.2.3 Alphabet de l'association phonétique internationale

Dans le domaine de la transcription phonétique, il existe divers systèmes de représentation des transcriptions. Citons par exemple : le système propre aux romanistes, et, en Finlande, un système dit finno-ougrien utilisé pour l'étude des langues finno-ougriennes et le système utilisé par l'association phonétique internationale (API), appelé «transcription API». La transcription API s'est imposée comme standard, malgré qu'elle inclut des signes issus du système finno-ougrien [2]. Le système API se base sur la classification de paroles ainsi que sur les organes sollicités lors de la prononciations d'un son, à travers l'appareil phonatoire (voir la Figure 1.5) inclut dans l'appareil respiratoire (voir la Figure 1.1) : classification des consonnes en pulmonaires et non pulmonaires, classification des voyelles.

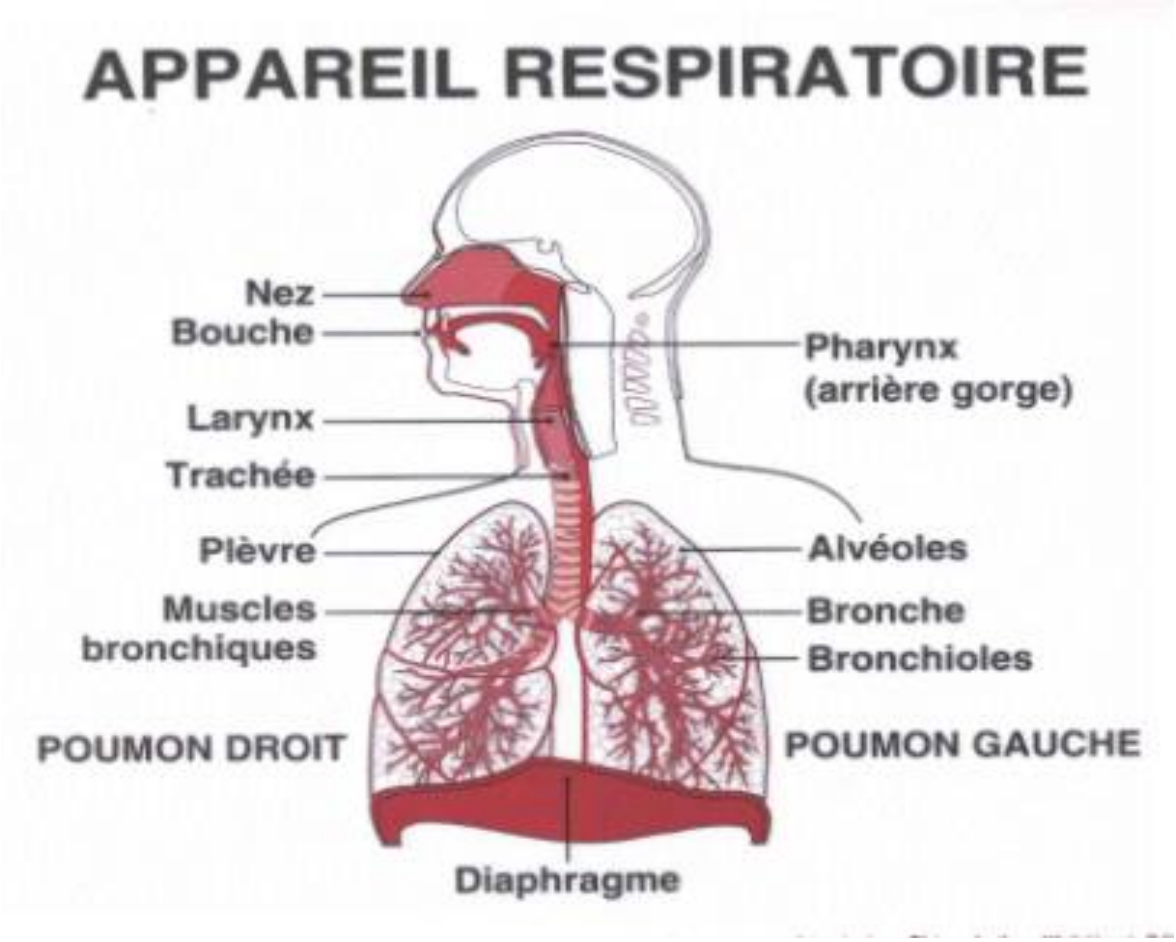
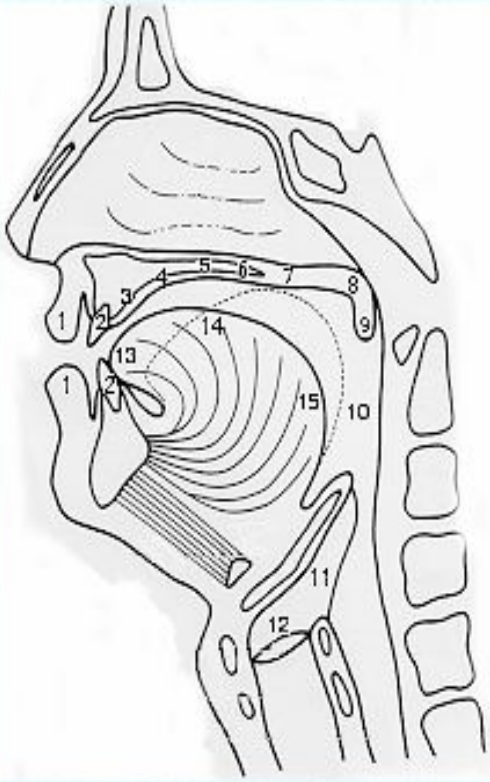


FIGURE 1.1 – Appareil respiratoire

1.2.4 Classification des consonnes

En plus de la classification des consonnes en pulmonaires et non pulmonaires, on utilise aussi une distinction selon l'articulation(voir la Figure 1.2) :

— les lèvres (articulations labiales ou bilabiales)
— les dents (articulations dentales)
— les lèvres et les dents (articulations labio-dentales)
— les alvéoles (articulations alvéolaires)
— le palais (articulations palatales)
— le voile du palais (articulations vélaires)
— la luette (articulations uvulaires)



	Organe anatomique		Nomenclature phonétique correspondante	
1	lèvres		labiales	
2	dents		dentales	
3	alvéoles		alvéolaires	
4	palais dur		pré-palatales	
5			médio-palatales	
6			post-palatales	
7	voile du palais		pré-vélaires	
8			post-vélaires	
9	luette (*uvula*)		uvulaires	
10	pharynx		pharyngales	
11	larynx		laryngales	
12	glotte		glottales	
13	apex	de la langue	apicales (pré-dorsales)	dorsales
14	dos		médio-dorsales	
15	racine		radicales (post-dorsales)	

FIGURE 1.2 – Point des articulations

1.2.4.1 Consonnes pulmonaires

On dit qu'une consonne est pulmonaire si elle est produite avec une obstruction de la glotte (l'espace entre les cordes vocales) ou de la cavité buccale (bouche), et avec une libération simultanée ou subséquente de l'air provenant des poumons. Les consonnes pulmonaires représentent la majorité des consonnes dans le système API.
La Figure 1.3 résume les détails de l'ensemble des consonnes pulmonaires.

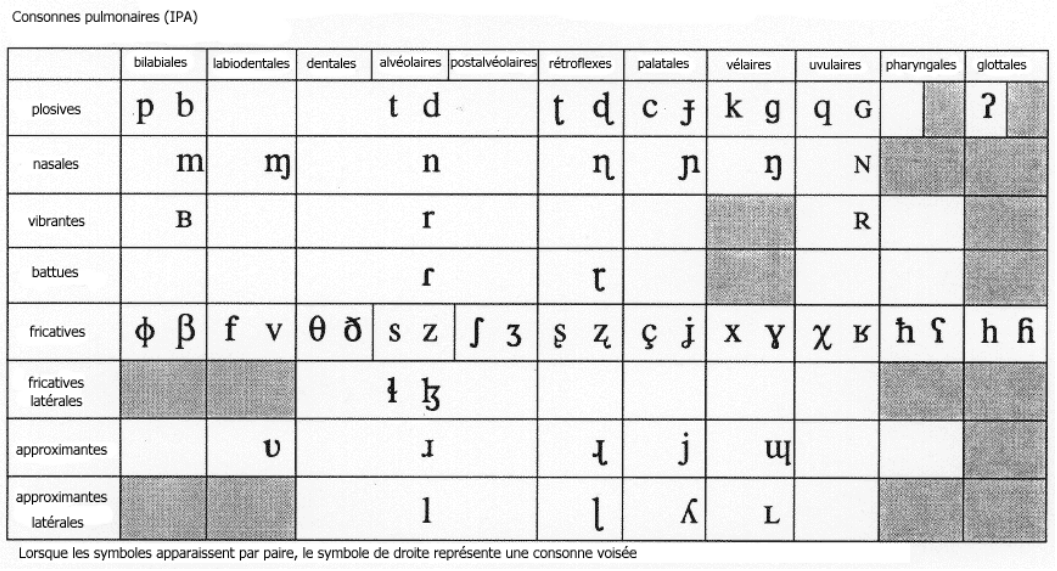


FIGURE 1.3 – consonnes pulmonaires(API Consonnes)

Lorsque les symboles apparaissent par paires, celui de droite représente une consonne exprimée. Les zones ombrées indiquent les articulations jugées impossibles.

1.2.4.2 Consonnes non pulmonaires

Les consonnes non pulmonaires sont représentées par la figure 1.4.

Clics		Implosifs vocaux		Éjectifs	
ʘ	Bilabial	ɓ	Bilabial	'	Exemples:
ǀ	Dentaire	ɗ	Dentaire / alvéolaire	p'	Bilabal
ǃ	(Post alvéolaire	ʄ	Palatal	t'	Dentaire / alvéolaire
ǂ	Palato-alvéolaire	ɠ	Vélaire	k'	Vélaire
ǁ	Latéral alvéolaire	ʛ	Uvulaire	s'	Fricative alvéolaire

FIGURE 1.4 – Consonnes non pulmonaires(API Consonnes)

Remarque Dans ce qui suit on note une lettre voyelle par V et une lettre consonnes par C. La Figure 1.5 expliquant l'appareil phonatoire chez les humains, pour faire apparaître la location de chaque élément fonctionnel au niveau de cette appareil.

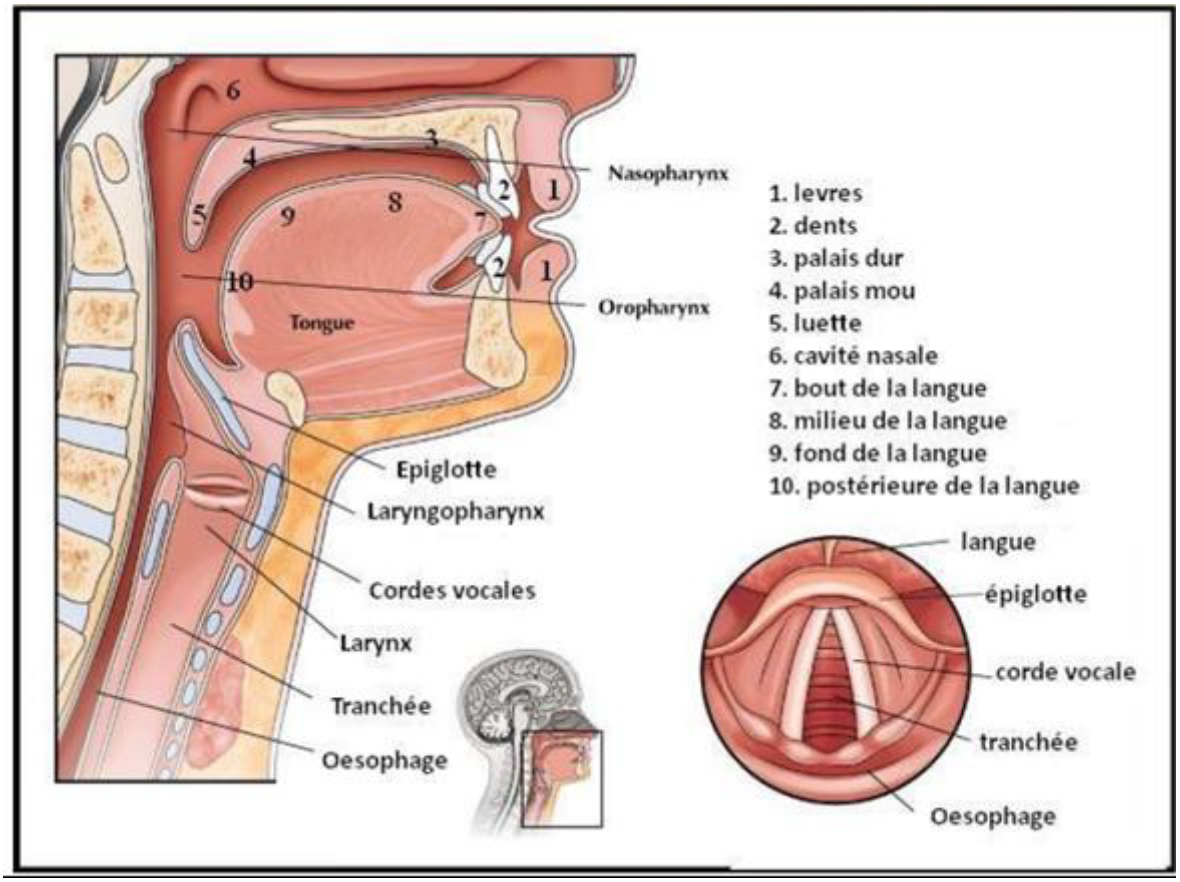


FIGURE 1.5 – l'appareil phonatoire [5]

1.3 Signes de l'alphabet API dans la transcription française

Rappelons que notre travail se base sur la langue française. Dans ce qui suit on s'intéresse aux signes de l'alphabet API pour la transcription du français. Notons l'absence en dans la langue française d'une correspondance parfaite entre la prononciation et l'orthographe de l'usage des mots. On peut résumer l'ensemble des règles comme suit :

1.3.1 Voyelles

Les voyelles sont classées en *voyelles orales* (voir la Figure 1.6) et *voyelles nasales* (voir la Figure 1.7). Les Figures : 1.6 et 1.7 donnent les écritures des différentes voyelles de la langue française ainsi que les phonèmes correspondants.

I. VOYELLES ORALES			
SONS – APHI	SIGNES GRAPHIQUES		EXEMPLES
	Minuscules	Majuscules	
1 [a]	a, à, e	A, À, E	*patte, là, femme, solennel*
2 [ɑ]	a, â,	A, Â	*tas, pâte*
3 [ɛ]	e, è, ê, ë, ai, aî, ei, eî, ea, æ	E, È, Ê, Ë AI, AÎ, EI, EÎ, EA, AE	*nette, père, fête, Noël, laine, naître, peine, reître, steak, et cætera*
4 [e]	é, er, ez, es, et, œ	E, É, Œ	*thé, parler, nez, les, et, fœtus*
5 [i]	i, î, ï, y, ee	I, Î, Ï, Y, EE	*il, île, naïf, style, meeting*
6 [ɔ]	o, u, au, oo	O, AU, U, OO	*comme, radium, Paul, alcool*
7 [o]	o, ô, au, eau, aô	O, Ô, AU, EAU, AO, AÔ	*dos, tôt, chaud, eau, Saône*
8 [œ]	eu, eui, œi, œu, uei, u, i	EU, EUI, ŒI, ŒU, UEI, U, I	*peur, fauteuil, œil, bœuf, cœur, recueil, club, girl*
9 [ø]	eu, eû, œu, ueue, oe	EU, EÛ, ŒU, UEUE, OE	*feu, jeûne, nœud, queue, foehn*
10 [ə]	e, on, ai	E, ON, AI	*le, serai, monsieur, faisons*
11 [u]	ou, où, oû, aou, aoû, oo	OU, OÙ, OÛ, AOU, AOÛ, OO	*ou, où, saoul, août, football*
12 [y]	u, û, eu, ü, ue, üe, uë	U, Û, EU, Ü, ÛE, UË	*mur, mûr, il a eu, Saül, bossue, aigüe, aiguë*

FIGURE 1.6 – les voyelles orales [10]

II. VOYELLES NASALES			
SONS – APHI	SIGNES GRAPHIQUES		EXEMPLES
	Minuscules	Majuscules	
1 [ɑ̃]	an, am, en, em, aon	AN, AM, EN, EM, AON	*an, champ, lent, emballer, paon*
2 [ɛ̃]	in, im, ym, yn, ain, aim, ein, în	IN, IM, YM, YM, AIN, IM, EIN, IN, ÎN	*fin, simple, thym, syntaxe, main, faim, sein, vînt*
3 [õ]	on, om, un	ON, OM, UN	*non, nom, punch*
4 [œ̃]	um, un, eun	UM, UN, EUN	*parfum, un, à jeun **

FIGURE 1.7 – les voyelles nasales [10]

1.3.2 Consonnes

La figure 1.8 regroupe les différentes consonnes de la langue française ainsi que leurs phonèmes.

IV. CONSONNES			
SONS – APHI	SIGNES GRAPHIQUES		EXEMPLES
	Minuscules	Majuscules	
1 [b]	b, bb	B, BB	*balle, abbé*
2 [d]	d, dd, dh	D, DD, DH	*dos, additionner, Bouddha*
3 [f]	f, ff, ph	F, FF, PH	*fort, bouffer, phonème*
4 [g]	g, c, gh	G, C, GH	*gant, second, ghetto*
5 [k]	c, ch, cch, ck, k, q, qu, cqu	C, CH, CCH, CK, K, Q, QU, CQU	*corps, chaos, saccharine, ticket, kilo, coq, qui, grecque*
6 [l]	l, ll	L, LL	*calcul, Lille*
7 [m]	m, mm	M, MM	*maman, grammaire*
8 [n]	n, nn	N, NN	*noir, panne*
9 [ɲ]	gn	GN	*agneau, gnagnan*
10 [p]	p, pp,	P, PP	*papa, apporter*
11 [ʀ]	r, rr, rh	R, RR, RH	*rare, arracher, rhume*
12 [s]	s, ss, sc, c, ç, t, x	S, SS, SC, C, Ç, T, X	*soir, assez, fascicule, ciel, ça, nation, soixante*
13 [ʃ]	ch, sch, sh	CH, SCH, SH	*chose, schéma, shampooing*
14 [t]	t, tt, th	T, TT, TH	*tu, attendre, thé*
15 [v]	v, w	V, W	*voir, wagon*
16 [z]	z, zz, s, x	Z, ZZ, S, X	*zigzag, mezzanine, rose, dixième*
17 [ʒ]	j, g	J, G	*jour, neige*

FIGURE 1.8 – les consonnes [10]

1.3.3 Semi-consonnes

Il existe aussi des séquences de lettre qui jouent en même temps le rôle d'une voyelle et contiennent des consonnes. Ces séquences sont appelées *semi-consonnes* ou *semi-voyelles*. Ces séquences sont représentées dans la figure 1.9.

III. SEMI-VOYELLES / SEMI-CONSONNES			
SONS – APHI	SIGNES GRAPHIQUES		EXEMPLES
	Minuscules	Majuscules	
1 [j]	i, ï, ill, y	I, Ï, Y	*bien, ïambe, fille, voyons, yeux*
2 [w]	oi oî oy ou (+ Voyelle) oe oê ua	OI, OÎ OY OU (+ Voyelle) OE, OÊ UA	*loi* *croît* *Troyes* *oui* *moelle* *poêle* *jaguar, équateur*
3 [ɥ]	u (+ Voyelle)	U (+ Voyelle)	*lui, buée, tuons*

FIGURE 1.9 – les semi - voyelles [10]

1.4 Syllabation

En langue française, une *syllabe* est une voyelle ou un groupe de lettres qui se prononcent d'une seule émission de voix.
La *syllabation* est le processus de découpage d'un mot en syllabes. La syllabation en français est de deux types :

Syllabe ouverte : elle se termine par une voyelle (majoritaire). Elle est la base de la structure rythmique du français parlé.

Syllabe fermée : elle se termine par une consonne.

En français (méridional ou autres), il y a une implication entre les types des voyelles et les types des syllabes comme suit : Syllabe ouverte : $\rightarrow$ voyelle fermée [e, ø, o] et Syllabe fermée : $\rightarrow$ voyelle ouverte [E, ø, ç].Mais, pour cette règle, il existe des exceptions en français standard. la liste des groupes consonantiques possibles est la suiante : [pr], [pl], [br], [bl], [kr], [kl], [gr], [gl], [fr], [fl], [vr], [tr], [dr], [sp],[st], [sk], [sf], [ps], [pn], [tS], [str], [spl], [srk]. Les consonnes se combinent aussi avec les semi-voyelles : [vw], [drw], [trw], [prw], [krw], [frw], [blw], [glw].

Types de syllabes		Fréquence	Exemples
Syllabes ouvertes	CV	55%	nous [nu]
	CCV	14%	prend [prã]
	V	6%	ou [u]
	CCSV	1%	croit [krwa]
Syllabes fermées	CVC	17%	donne [dçn]
	CVCC	4%	sorte [sçrt]
	VC	2%	hors [çr]
	CVCC	1%	muscle [myskl]

TABLE 1.2 – Statistiques sur les types de syllabations [9]

Le processus de syllabation divise le mot en syllabes et non pas en sous-mots. Entre les syllabes d'un groupe rythmique, il n'existe pas des ruptures : dans ses prononciations, on a un séquencement des syllabes une après l'autre appelé enchaînement. Il existe deux types d'enchaînement : l'enchainement consonantique est exprimé phonologiquement par prononciation d'une consonne finale suivie par une voyelle initiale du mot suivant.

Exemple 1 *Notre hôtel anglais se trouve au centre ville [nç tro tE lã gle s' tru vo sã tr' vil] et non pas : [nçtr o tEl ã gle s' truv o sã tr' vil]*

L'enchaînement vocalique est exprimé phonologiquement par prononciation d'une séquence des voyelles consécutives.

Exemple 2 *Elle a haï Haïti ici aussi [E la a i a i ti i si o si]*

1.4.1 Liaison

selon Dr.Ayoun [9],«la liaison est un procédé phonologique par lequel une voyelle initiale fait apparaître un son consonantique final généralement muet». Il y a trois types de liaison :

Les liaisons obligatoires : qui se font toujours.

Les liaisons interdites : qui ne se font jamais.

Les liaisons facultatives : qui peuvent se faire ou non, selon le locuteur.

1.5 Signes de l'alphabet API dans la transcription arabe

1.5.1 Phonèmes

Selon Kouloughli [6], la langue arabe se caractérise par un consonantisme riche et un vocalisme pauvre. On peut classer les phonèmes arabes comme suit :

1. Labiales

Caractère	Prononciation	Observation
ب	b	
ف	f	comme le son de la lettre f du français
م	m	

2. Inter-dentales :

Caractère	Prononciation	Observation
ث	þ	fricative sourde, se prononce comme le th en anglais ex : think.

Dans les dialectes :

(a) réalisée dans les parlers bédouins et les parlers "conservateurs" (Tunisie, Irak‘) ;

(b) ‘t en Algérie et au Maroc.

(c) t ou s en Orient (s généralement dans les emprunts récents au standard).

caractère	prononciation	observation
ذ	th	fricative sonore, se prononce comme le th en anglais ex : there.

Dans les dialectes, même distribution que : ث

(a) dans les parlers bédouins et les parlers conservateurs ;

(b) d en Algérie et au Maroc ;

(c) d ou z en Orient (avec la même distribution que ci dessus).

caractère	prononciation	observation
ظ		fricative sonore emphatique.

Presque toujours réalisée, en standard et en dialectal :

(a) comme un (Maroc) ;

(b) comme un (emphatique du z) en Orient.

3. Dentales :

caractère	prononciation	observation
ت	t	parfois prononcé comme le ts de tsé-tsé au Maroc
د	d	
ل	l	comme le sons de la lettre l du français
ن	n	
ط		occlusive sourde emphatique.
ض		occlusive sonore emphatique.

Totalement confondue, en arabe standard (accent régional) et en arabe dialectal, avec le (Irak, est du Maghreb, parlers bédouins)

caractère	prononciation	observation
ر	r	vibrante sonore.

r toujours roulé comme en espagnol ou en italien.

4. Sifflantes :

caractère	prononciation	observation
س	s	
ز	z	comme les sons correspondants du français
ص	S	fricative sourde emphatique (emphatique du s)

Parfois réalisée comme z dans certains dialectes d'Orient (mais jamais en arabe standard).

5. Palatales :

caractère	prononciation	observation
ش		comme ch français de chat.
ج		comme j français de jamais.

Dans les dialectes et en standard (accent régional) :

(a) réalisé dj (affriquée sonore) en Algérie et dans certaines régions d'Orient ;

(b) réalisé g (comme dans gâteau) en Egypte.

caractère	prononciation	observation
ي	y	comme y français de payé.

6. ي Vélaires :

caractère	prononciation	observation
ك	k	comme k français.

Réalisé comme tch de atchoum dans les dialectes bédouins d'Orient.

caractère	prononciation	observation
خ		fricative sourde, comme le j espagnol ou le ch allemand .
غ		fricative sonore, comme le r parisien grasseyé, différente de r roulé.

7. ق Uvulaire :

caractère	prononciation	observation
ق	q	occlusive sourde .

Dans les dialectes :

(a) au Maghreb souvent réalisé g comme le g de gâteau (parlers ruraux, ou influencés par les parlers ruraux) ;

(b) en Orient réalisé g ou dans les parlers bédouins, q dans les parlers ruraux, dans les parlers citadins.

8. ث Pharyngales :

caractère	prononciation	observation
ح		fricative sourde.
ع	e	fricative sonore.

9. Glottales :

caractère	prononciation	observation
ه	h	fricative sonore.

h expiré, ,proche du h anglais de have.

caractère	prononciation	observation
ء		occlusive sourde.

Ressemble à l'attaque vocalique en allemand (Atem).
ci-dessous un tableau résume la phonétique arabe(voir Figure 1.10) :

lettre	nom	fin	milieu	Début	phonétique
ا	alif	ـا	ـا	ا	a:
ب	ba	ـب	ـبـ	بـ	b
ت	ta	ـت	ـتـ	تـ	t
ث	ṯa (tha)	ـث	ـثـ	ثـ	θ
ج	ǧim (jim)	ـج	ـجـ	جـ	ʤ , ʒ , g
ح	Ḥa	ـح	ـحـ	حـ	ħ
خ	ḫa (kha)	ـخ	ـخـ	خـ	x
د	dal	ـد	ـدـ	دـ	d
ذ	ḏal (dhal)	ـذ	ـذـ	ذـ	ð
ر	ra	ـر	ـرـ	رـ	r
ز	zay	ـز	ـزـ	زـ	z
س	sin	ـس	ـسـ	سـ	s
ش	šin (shin)	ـش	ـشـ	شـ	ʃ
ص	Ṣad	ـص	ـصـ	صـ	$s^ʕ$
ض	Ḍad	ـض	ـضـ	ضـ	$d^ʕ$, $ð^ʕ$
ط	Ṭa	ـط	ـطـ	طـ	$t^ʕ$
ظ	Ẓa	ـظ	ـظـ	ظـ	$z^ʕ$, $ð^ʕ$
ع	ʿayn	ـع	ـعـ	عـ	$ʔ^ʕ$
غ	ġayn (ghayn)	ـغ	ـغـ	غـ	ɣ
ف	fa	ـف	ـفـ	فـ	f
ق	qaf	ـق	ـقـ	قـ	q
ك	kaf	ـك	ـكـ	كـ	k
ل	lam	ـل	ـلـ	لـ	l
م	mim	ـم	ـمـ	مـ	m
ن	nun	ـن	ـنـ	نـ	n
ه	ha	ـه	ـهـ	هـ	h
و	waw	ـو	ـوـ	وـ	w , u:
ي	ya	ـي	ـيـ	يـ	j , i:
ء	hamza	أ ؤ إ ئ ـئـ			ʔ

FIGURE 1.10 – Les lettres arabes transcrits en phonèmes latins [8]

1.5.2 Translittération des noms arabes en écriture latine

L'alphabet de la langue arabe est constitué de 28 lettres. Elle contient 25 consonnes et 3 voyelles longues et il existe d'autres voyelles courtes ; que l'on peut considérer comme l'apostrophe ou l'accent du latin, utilisés beaucoup dans l'écriture du Coran el-karim et el-hadith el-charif.
On peut les résumer dans la table suivante :

Voyelle courte	Transcription	Nom
	a	fatha (فتحة)
ـِ	i	Kasra(كسرة)
	u	Damma(ضمة)
ـّ	Double consonne	Chadda(شدة)
ـً	aa	Fathataine(فتحتين)
ـٍ	ii	Kasrataine(كسرتين)
ـٌ	uu	Dammataine(ضمتين)
ـْ	e	Sukune(سكون)

TABLE 1.3 – Les voyelles courtes en arabe [8]

1.5.3 Normes de translittération pour l'arabe

Il existe divers normes de translittération, comme : EI (1960), ISO/R 233 (International Organization for Standardization, 1961), UN (United Nations Group of Experts on Geographical names, 1972), DIN-31635 (Deutsches Institut für Normung, 1982), ISO 233 (International Organization for Standardization, 1984) , la norme ALA-LC (America Library Association, 1997). La normes DIN-31635 utilisée internationalement par la communauté scientifique et la norme adoptée par l'Encyclopédie de l'Islam (EI)[8].

1.6 Correspondance proposée pour la translittération des lettres arabes vers le français

Dans cette section nous proposons une correspondance pour la translittération des lettres arabes vers le latin. Cette correspondance a été utilisée dans notre implémentation du processus de phonétisation.

Lettre	Équivalents en écriture latine	Phonème français	Lettre	Équivalents en écriture latine	Phonème français
ء	’, a	a	غ	Gh, gh, Ğ, ğ, ḡ	gh
ا	A, a ä, â, á, ā, e, ê	a	ف	F, f, ph	f
ب	B, b	b	ق	Q, q, C, c, K, k	k
ت	T, t	t	ك	K, k, C, c	k
ث	Th, th, t, t̲,	t	ل	L, l	l
ج	J, j, Dj, dj, g, Ğ, ğ	g	م	M, m	m
ح	H, h, Ḥ, ḥ, ẖ, 7	h	ن	N, n	n
خ	Kh, kh, ẖ, h̲	kh	ه	H, h	h
د	D, d	d	و	W, w, ou, o, u, ô, û, ū, ú, ü	w
ذ	Dh, dh, D, d, D̲, d̲, Ḍ, ḍ	dh	ي	I, i, y, ï, î, ī,	i
ر	R, r	r	آ	A, a, ā, ’ā, ’â	a
ز	Z, z, Z̧, z	z	ة	A, a, ā, ’ā, ’â	a
س	S, s	s	ى	H, h, T, t, at, a, ẗ	a
ش	Ch, ch, Sh, sh, Š, š	ʃ	أ	A, a, á, à, ā, ỳ	a
ص	S, s, Ş, ş, Ṣ, ṣ	s	ؤ	A, a, á, à, ā, ỳ	a
ض	D, d, Ḍ, ḍ, Ḍ, ḍ	d	إ	i	a
ط	T, t, Ṭ, ṭ, Ţ, ţ	t	ئ	U, u, Ou, ou, Û, û	a
ظ	Z, z, Ẓ, ẓ, 6’, Dh, dh, D, d	dh	گ	G, g	k
ع	‘, ’, ‘, 3, a, â	a		’, (Blanc)	

FIGURE 1.11 – Les lettres arabes transcrits en phonèmes français

Parfois quelques lettres arabes transcrites en chiffres, comme celle de la norme SMS Européen ou du Moyen Orient.

Lettre	ق	غ	ع	ظ	ط	ض	ص	خ	ح	ء
transcription alphanumérique	2	7	7’	9	9’	6	6’	3	3’	8

TABLE 1.4 – Équivalences alphanumériques dans les textes écrits en alphabet latin [8]

1.6.1 Transit de la phonétique arabe vers la phonétique française

D’après la lecture de l’automate(voir la Figure 1.12), la réponse à la question de vérification si le mot (nom) est voyellé ? Si la réponse est oui, alors on fait la suppression des

voyelles courtes, sinon on passe directement à la translittération. A la fin du processus on obtient une liste de noms propres arabes écrits en latin.

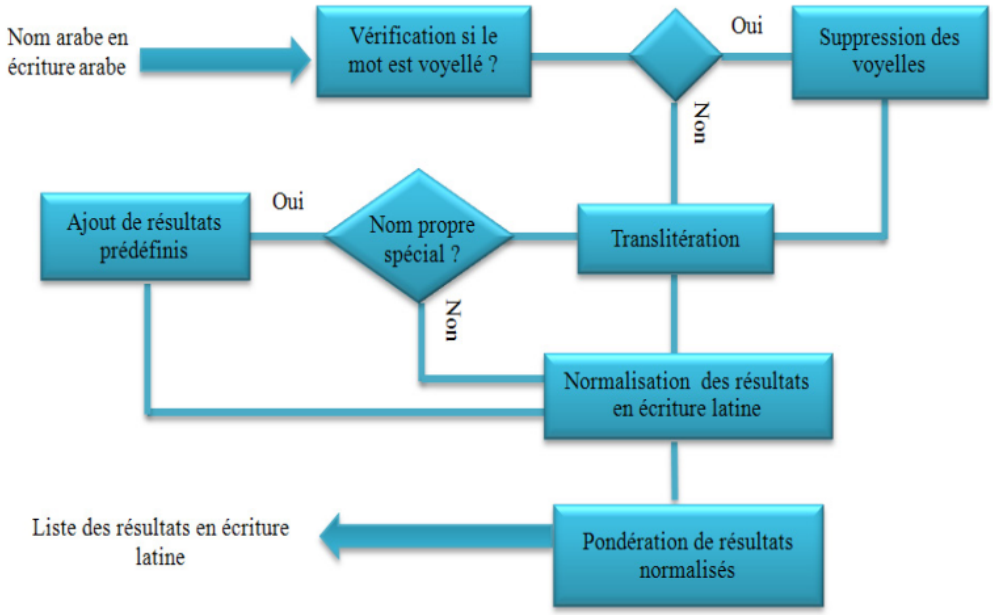


FIGURE 1.12 – Organigramme de fonctionnement translittérateur de l'arabe vers le latin [8]

Remarque : D'après le processus de translittération de noms propres arabes vers latin, on remarque que toutes les lettres écrites sont utiles, c.-à-d. sont prononcées. Ceci implique qu'elles sont toutes apparentes dans la transcription phonétique des noms (souvent pas de phonèmes muets [']), sauf quelques exceptions comme h à la fin du mot comme "MANSOURAH".

1.6.2 Conclusion

Dans de ce chapitre nous avons présenté la phonétique de la langue française et celle de la langue arabe. Puis nous avons exposé les étapes nécessaires au processus de phonétisation des deux langues, dans le but de diminuer le nombre des cas de traitement à cause de la normalisation de des différentes écritures, qu'elles ayant une grande partie commune entre eux deux à deux.
On peut conclure que via la phonétisation on réalise un gain de 41% des nombres des symboles (voir l'annexe C).
Ce processus de phonétisation a été implémenté et est considérer comme pré-traitement à notre algorithme de calcul de distance entre les noms propres.

CHAPITRE **2**

SIMILARITÉ DES SÉQUENCES

"Le coeur a ses raisons que la raison ne connaît point."

BLAISE PASCAL,
Extrait des Pensées

2.1 Introduction

Le mesure de la similarité entre deux séquences ou bien deux mots consiste à évaluer le niveau de ressemblance entre eux, à savoir quelles sont identiques.Ce dernier souvent exploiter dans plusieurs domaines essentiels, parmi eux on peut parler sur la recherche d'informations,le traitement de langue et de parole, la traduction automatique, la bioinformatique, ect.
Dans ce chapitre nous allons illustrer quelques méthodes et techniques de calcul de similarité et voir en bref leurs principes de fonctionnement afin de pouvoir les implémenter dans notre projet de fin d'étude. Parmi ces techniques nous avons explorer la technique de l'alignement, l'approche noyaux, et distances de similarité (Jaro, Jaro-Winkler et Levenshtein).

2.2 Alphabet, séquence et sous-séquence

On appelle un **alphabet** tout ensemble fini (non vide) Σ de symboles appelés **lettres** [11]. La taille des alphabets qu'on considère peut varier selon l'emploi. Dans notre étude on considère des alphabets finis. Un **mot** sur l'alphabet Σ est formé par concaténation d'une suite finie $(x_1, x_2, \ldots, x_n)$ de lettres x_i (appelée aussi séquence) et sera écrit sous la forme $x_1x_2 \cdots x_n$. [11]

La longueur d'un mot u, notée $|u|$, est le nombre de lettres qu'il comporte. On notera par ailleurs u_i la i-ème lettre du mot u. Il existe un unique mot de longueur égale à zéro : **le mot vide**, noté ε.

On définit l'opération de **concaténation** des mots : si u et v sont deux mots, alors uv est la concaténation de u et v. Le mot vide est l'élément neutre de cette opération : $\varepsilon w = w\varepsilon = w$, pour tout mot w.

Définition 2 *Soit $S = s_1s_2 \ldots s_n$ une séquence sur Σ. Une* **sous-séquence** *de S est tout mot $T = t_{i_1}t_{i_2} \ldots t_{i_m}$ tel que $1 \leq i_1 < i_2 < \ldots < i_m \leq n$. [11]*

L'exemple suivant illustre les différentes notions introduites précédemment.

Exemple 3 *Soit $\Sigma = \{A, C, G, T\}$ un alphabet de quatre symboles.*
— *Le mot $w = ACCTGT$ est une séquence sur Σ*
— *la longueur du mot w est $|w| = 6$*
— *$w_2 = w_3 =$ 'C'*
— *les mots ε, A, CTT, $ACGT$ sont des sous séquences de w*

2.3 Alignement de séquences

2.3.1 Principe de l'alignement

Aligner deux séquences définies sur un même alphabet consiste à quantifier et localiser la similarité dans la paire de deux séquences. L'idée est de superposer les zones identiques entre les deux séquences. Cette mise en correspondance des parties communes induira des brèches (gaps) dans les séquences, qui seront matérialisées par le symbôle "-" [12].

Définition 3 (Alignement) *Soit Σ un alphabet. Un alignement de deux séquences S et T sur Σ est un mot w sur l'alphabet $(\Sigma \cup \{-\} \times \Sigma \cup \{-\})\backslash\{(-,-)\}$*
tel que :

$$S = h(\Pi_1(w)) \quad \textit{et} \quad T = h(\Pi_2(w))$$

où Π_1 et Π_2 sont respectivement la première et la deuxième projection et h est la fonction qui remplace le symbole "-" par le mot vide ε c'est à dire $h('-') = \varepsilon$.

Exemple 4 *Soient $\Sigma = \{A, C, G, T\}$, $S = GACTGAG$ et $T = GATCGAAG$ deux sous-séquences sur Σ. Un alignement possible de S et T est :*

$$w = (G,G)(A,-)(C,A)(T,T)(G,C)(-,G)(-,A)(A,A)(G,G)$$

— $S' = \Pi_1(w) =$ `G A C T G - - A G`
— $T' = \Pi_2(w) =$ `G - A T C G A A G`

On peut vérifier que $h(S') = S$ et $h(T') = T$

Proposition 1 *Soient S, T deux séquences sur Σ et w un alignement de S et T. On a alors*

$$\max(|S|, |T|) \leq |w| \leq |S| + |T|$$

Définition 4 (Opérations d'édition) *Soit Σ un alphabet. Une opération d'édition est un symbole (x, y) de l'alphabet $(\Sigma \cup \{-\} \times \Sigma \cup \{-\})\backslash\{(-,-)\}$*
*il existe quatre types d'***opérations d'édition**

1. *Substitution : dans le cas où $x \neq y \neq$ '-' (par exemple (A, G)).*
2. *Délétion : dans le cas où $x \neq y$ et $y =$ '-' (par exemple $(T, -)$)*
3. *Insertion : dans le cas où $x \neq y$ et $x =$ '-' (par exemple $(-, A)$)*
4. *Identité : dans le cas où $x = y$ (par exemple (A, A)) [17]*

2.3.2 Évaluation d'un Alignement

Une mesure possible est la la longueur minimale de w. Dans ce qui suit nous présentons quelques mesures de distances d'alignement utilisées dans la pratique.

Afin de définir des mesures de distance entre séquences, nous attribuons des scores aux différentes symboles de l'alphabet des alignements. La fonction score est définie comme suit) [16] :

$$sc \quad : \quad (\Sigma \cup \{-\} \times \Sigma \cup \{-\}) \backslash \{(-,-)\} \to \mathbb{R} \tag{2.1}$$

La notion de score peut être étendue aux alignements comme suit : Soit $w = w_1 w_2 \dots w_n$ un alignement le score $s(w)$ est défini par

$$sc(w) = \sum_{i=1}^{n} sc(w_i)$$

Un cas particulier de la fonction score est celui utilisé en bio-informatique et qui est associé aux opérations d'édition [16]

1. Cas de la substitution $sc(x, y) = \alpha_S$: dans le cas où $x \neq y \neq$ '-'
2. Cas de la délétion $sc(x, y) = \alpha_D$ dans le cas où $x \neq y$ et $y =$ '-'
3. Cas de l'insertion $sc(x, y) = \alpha_I$ dans le cas où $x \neq y$ et $x =$ '-'
4. Cas de l'identité $sc(x, y) = \alpha_{Id}$ dans le cas où $x = y$

où $\alpha_S, \alpha_D, \alpha_I$ et α_{Id} sont des scalaires.

Comme il existe plusieurs alignements w possibles pour deux séquences S et T données, il est nécessaire de pouvoir déterminer quel est le meilleur alignement (meilleur score) ou plutôt le score optimal. Notons $W(S, T)$ l'ensemble des alignements des séquences S et T. Le score optimal est :

$$\min(sc(w))_{w \in W(S,T)}$$

Définition 5 (Distance d'édition) *La distance d'édition (parfois appellée distance de Levenshtein) $D(S, T)$ entre deux séquences S et T est le nombre minimal d'opérations d'insersion, de délétion, et substitution dans une suite d'operations qui transforme S en T i.e :*

$$D(S, T) = \min(sc(w))_{w \in W(S,T)}$$

où $\alpha_{Id} = 0$ et $\alpha_S = \alpha_I = \alpha_D = 1$.

Proposition 2 *La distance d'édition est une fonction symétrique :*

$$D(S, T) = D(T, S)$$

Notons que plusieurs alignement peuvent conduire à la même distance d'édition.

2.3.3 Choix de la fonction score

Le choix d'une «bonne» fonction de score dépend du contexte de l'application. En bio-informatique, il existe différentes fonctions de score. Pour $x \neq y \neq' -'$

— score identité : $sc(x, x) = 1$ et $sc(x, y) = 0$ pour $x \neq y$,
— score transition/transversion : $sc(x, x) = 3$, $sc(x, y) = 1$ et $sc(y, x) = 0$
— score Blast : $sc(x, x) = 5$ et $sc(x, y) = -4$

Reste le score associé aux brèche i.e. aux couples $(x,' -')$ et $('-', x)$. Si le score associé n'est pas assez pénalisant la fonction score favorisera les alignements contenant beaucoup de brèches. Inversement si les brèches sont fortement évaluer, les alignements ne comportant pas assez de brèches seront favorisés.

2.3.4 Evaluation des brèches(Gaps)

L'évaluation d'un alignement revient à l'évaluation des brèches $sc(x, -)$ et ça selon les valeurs attribuées aux brèches et leur nombre qui peut varier.
Si elles ne sont pas assez pénalisantes, la fonction de score risque de favoriser les alignements qui en contiennent beaucoup de brèches. Inversement, si les brèches $sc(x, -)$ sont trop évaluées, les alignements ne comportant pas assez des brèches sont favorises, empêchant ainsi d'avoir toutes les correspondances.
Il existe principalement deux modèles d'évaluation le coût généré par l'insertion d'une brèche, les valeurs reliés à une brèche pour un modèle peut être déduire d'après des fonctions qui emploie la longueur de la brèche et le cout de celle ci. [12]

2.3.5 Les brèches à cout constant

Ce modèle repose sur une évaluation simple, tels que la valeur du brèche est unique malgré la diversité de sa position, i.e : l'évaluation se faite de même façon de celle des paires de caractères.Pour la faire, il suffit de sommer toutes les valeurs sur la longueur de l'alignement.On peut considère le symbole '' peut alors être comme un caractère pour accomplir l'évaluation. [14]

2.3.6 Les brèches à cout affine

Son idée démarre du principe que la création de la brèche est beaucoup plus pénalisante que son élongation. Donc associer un cout différent suivant qu'il s'agit du début ou de l'extension de la brèche.
Les notions exploités généralement pour le coût d'ouverture d'une brèche sont K ou gop (opening penalty), mais on utilise fréquemment h ou gep (extending penalty) pour l'extension de la brèche . Le coût lie à une brèche de longueur l est exprimé par la fonction affine $gap(l) = K + h(l - 1)$.On trouve également cette fonction sous la forme $gap(l) = K' + h(l - 1)$ en posant $K' = K - h$ L'évaluation d'un alignement par cette méthode se faite d'une manière différente de ceux des brèches à cout constant. Il est possible de faire le calcul de deux méthodes :

— «La première consiste à n'évaluer que les paires des résidus, puis à y ajouter la somme des valeurs de toutes les brèches.

— La seconde méthode consiste à évaluer l'ensemble de l'alignement en faisant la somme de toutes les positions. Cette méthode nécessite lorsque l'on doit évaluer une brèche de savoir si la position précédente était également une brèche ou non. En pratique, seule cette méthode est utilisée pour la construction de l'alignement, même si elle génère plus de calculs que dans le cas des brèches à cout constant» [14].

2.3.7 Similarité proposé à base d'alignement

Pour chaque paire de séquence dans l'alignement, le calcul de la somme des scores de substitution ou gap entre elles permet de déterminer leur similarité.
Dans notre mémoire nous proposons une matrice de similarité entre alphabet sachant leurs types : voyelles, consonnes et semi-consonnes et même les brèches, et la ressemblance existante entre eux.
Donc nous avons donné des scores variant entres les caractères suivant leurs similarités et leur type, ces valeurs sont inclues dans l'ensemble {-30,-10,-7,-5,0,1,2,3,4,5}(la Table 2.1 présente la manière d'affectation des scores).
Le tableau 2.2 illustre quelques exemple :

Caractères comparés	scores
Lettre – brèche	-5
Voyelle - consonne	-10
Voyelle - voyelle	-1,-2,-3,0, 2,3,4 suivant la ressemblance des lettres
Consonne - consonne	-10,-7,-5,-4,-3,-2, 3, 2,5 suivant la ressemblance des lettres
Lettre complètement similaire	5

TABLE 2.1 – Principe de la construction de la table scores alphabétiques

Pou la totalité de la matrice voir l'annexe C .

Paires de caractères	scores
'-' , '-'	-30
'b','b'	5
'a','e'	0
'j','g'	2
'-','b'	-5
'i','q'	-10
'b','p'	-2
'o','i'	-1
'q','k'	3

TABLE 2.2 – exemples des quelques scores pondérés

et même chose pour les symboles I.P.A, nous avons donné des scores variants entres eux suivant leurs similarités lors de la prononciation, ces valeurs sont inclues dans l'ensemble -30,-10,-7,-5,-2,-1,0,1,2,3,4,5,, on affecte une de ces valeurs pour chaque paire de phonème I.P.A suivant :
le degré de leurs similarité phonétiques (voir la Table 2.3).
Le tableau 2.4 illustre quelques exemple :
Pou la totalité de la matrice voir l'annexe C .

Caractère phonétique comparés	scores
Caractère – brèche	-5
Voyelle - consonne	-10
Semi-Voyelle - Semi-voyelle	-1,-2,-3,0, 2,3,4 suivant la ressemblance des lettres
Voyelle - Semi-voyelle	-1,-2,-3,0, 2,3,4 suivant la ressemblance des lettres
Consonne - consonne	-10,-5,-4,-3,-2, 3, 2,5 suivant la ressemblance des lettres
Caractère complètement similaire	5

TABLE 2.3 – Principe de la construction de la table des scores phonétiques

Paires de caractères phonétiques	scores
'-' , '-'	-30
'b','b'	5
'œ','y'	0
'u','œ'	2
'-','b'	-5
'ə','k'	-10
'o','ɑ̃'	-2
'ɛ','ɔ'	-1
'e','ɑ'	3
'õ','ɔ'	4

TABLE 2.4 – exemples des quelques scores pondérés

2.3.8 Pondération des séquences

Si la pondération des séquences est à prendre en compte pour le calcul du score, la valeur de pondération pour chaque séquence est obtenue en fonction de la similarité de chaque séquence par rapport aux autres. Le calcul utilise la matrice de similarité précédente, avec le même principe. Annexe C

2.4 Conclusion

Nous avons présenté dans ce chapitre les notions de bases des séquences et le principe de la similarité des séquences à l'aide de l'alignement et nous avons proposé une matrice de similarité entre les caractères à la base des règles des phonétiques présentées dans le chapitre précédent.

CHAPITRE 3

IMPLÉMENTATION

amssymb

"*Dans une grande âme, tout est grand.*"

BLAISE PASCAL,
Extrait du discours sur les passions de l'amour

3.1 Introduction

DANS ce dernier chapitre réservé à la partie implémentation, nous commençons par un rappel des différentes similarité utilisées antérieurement par Peter Christen [1] afin de pouvoir comparer nos résultats avec les siens,nous allons présenter notre dataset qu'on vient de créer et l'utiliser,nous présenterons aussi notre application et discuter les résultats obtenus.

3.2 Similarité de Jaro-Winkler

Elle est posé de mesurer la similarité entre deux chaînes de caractères. Elle est définie comme une technique proposée en 1999 par William E.Winkler [23], provenant de la distance de Jaro [19] et considéré comme une amélioration de cette dernière qui est utilisée principalement dans la détection de doublons.
Cette mesure est normalisée d'une façon qu'elle est compris entre 0.0 et 1.0, tels que Plus sa valeur entre deux chaînes est élevée, implique qu' elles sont plus similaires. Elle est adaptée au traitement de chaînes courtes comme des noms ou des mots de passe, tels que la valeur zéro représentant qu'elles sont dissimilaires et celle de 1.0 implique qu'elles sont totalement similaires. [19]

3.2.1 Distance de Jaro

On définie la distance de [22] entre chaînes S_1 et S_2 par :

$$\mathrm{d}_j = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) \tag{3.1}$$

où :

- $|s_i|$ est la longueur de la chaîne de caractères S_i ;
- $|m|$ est le nombre de caractères correspondants ;
- $|t|$ est le nombre de transpositions ;

On considère deux caractères identiques de S_1 et S_2 comme correspondants si l'éloignement entre eux (i.e. la différence entre leurs positions dans leurs chaînes respectives) est inférieur ou égale($\leq$) :

$$\left[\frac{\max(|s_1|,|s_2|)}{2}\right]-1$$

Le nombre de transpositions est obtenu par comparaison du i-ème caractère correspondant de S_1 avec le i-ème caractère correspondant de S_2.On déduit le nombre de transpositions de tels manière qu'il est le nombre de fois où ces caractères sont différents, divisé par deux.

3.2.2 Distance de Jaro-Winkler

La méthode de Winkler utilise un coefficient de préfixe p qui est destiné surtout pour les chaînes commençant par un préfixe de longueur l (avec $l < 4$).En considérant deux chaînes S_1 et S_2 leur distance de Jaro-Winkler d_w est [20] :

$$\mathrm{d}_w = d_j + (l_p(1-d_j)) \tag{3.2}$$

Où :

- d_j est la distance de Jaro entre S_1 et S_2 ;
- l est la longueur du préfixe commun (maximum 4 caractères) ;
- p est un coefficient qui permet de favoriser les chaînes avec un préfixe commun. Winkler propose pour valeur $p = 0.1$;

On peut résumer le principe de la cette distance par l'organigramme suivant :

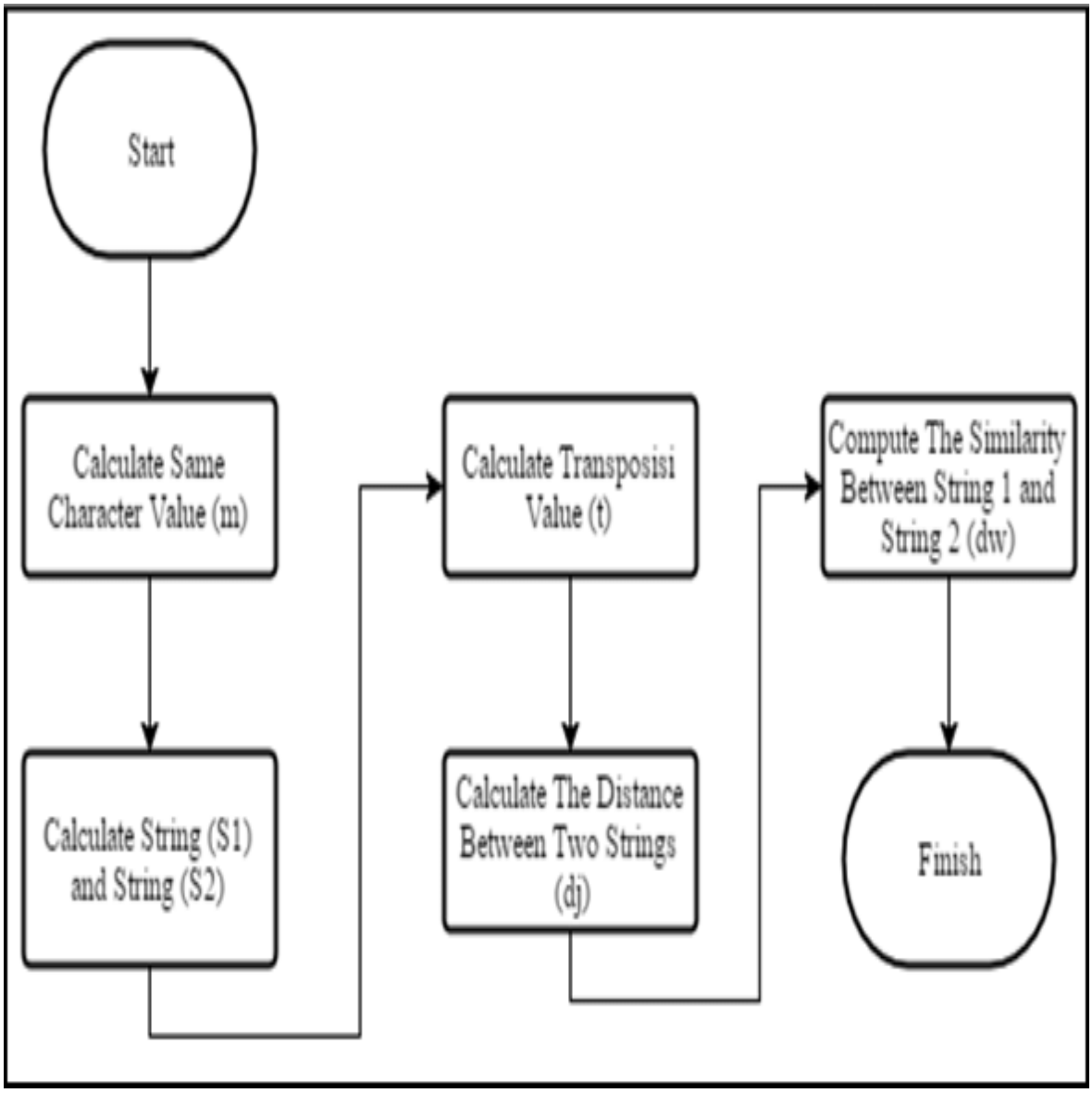


FIGURE 3.1 – Algorithme de distance Jaro-Winkler [20]

Exemples Soit deux chaînes S_1 = FARTHA et S_2 = FARHTA. ; la Table 3.2.2 de correspondance est :

$m = 6$ (nombre de 1 dans la table)

$|s_1| = 6$

$|s_2| = 6$

Les caractères correspondants sont {F,A,R,T,H,A} pour s_1 et {F,A,R,H,T,A} pour s_2. En considérant ces ensembles ordonnés, on a donc 2 couples (T/H et H/T) de caractères

	F	A	R	T	H	A
F	1	0	0	0	0	0
A	0	1	0	0	0	0
R	0	0	1	0	0	0
H	0	0	0	0	1	0
T	0	0	0	1	0	0
A	0	0	0	0	0	1

TABLE 3.1 – Exemple de table de correspondance entre deux mots : FARTHA et FARHTA

correspondants différents, soit deux demi-transpositions. D'où :

$$\text{t} = \frac{2}{2} = 1$$

La distance de Jaro est :

$$\text{d}_j = \frac{1}{3}(\frac{6}{|6|} + \frac{6}{|6|} + \frac{6-1}{6}) = 0.944$$

La distance de Jaro-Winkler avec $p = 0.1$ avec un préfixe de longueur $l = 3$ devient

$$\text{d}_w = 0.944 + (3 \times 0.1 \times (1 - 0.944)) = 0.961$$

Avec les chaînes $s_1 =$ MARWANE et $s_2 =$ MAROUANE on trouve :

$m = 6$
$|s_1| = 7$
$|s_2| = 8$
La distance de Jaro est :

$$\text{d}_j = \frac{1}{3}(\frac{6}{7} + \frac{6}{8} + \frac{6-2}{6}) = 0.758$$

Celle de Jaro-Winkler avec $l = 3$:

$$\text{d}_w = 0.758 + (3 \times 0.1 \times (1 - 0.758)) = 0.831$$

Avec les chaînes $s_1 =$ BIXON et$s_2 =$ BICKSONX, on obtient :
On calcule l'éloignement maximum pour le critère de correspondance

$$\left[\frac{\max(|s_1|, |s_2|)}{2}\right] - 1 = [\left[\frac{8}{2}\right] - 1 = 3$$

	B	I	X	O	N
B	1	0	0	0	0
I	0	1	0	0	0
C	0	0	0	0	0
K	0	0	0	0	0
S	0	0	0	0	0
O	0	0	0	1	0
N	0	0	0	0	1
X	0	0	1	0	0

TABLE 3.2 – Exemple de table de correspondance entre deux mots : BIXON et BICKSONX

$m = 4$ (les deux X ne correspondent pas, car ils sont éloignés de plus de 3 caractères)
$|s_1| = 5$
$|s_2| = 8$
La distance de Jaro :

$$\mathrm{d}_j = \frac{1}{3}(\frac{4}{|5|} + \frac{4}{|8|} + \frac{4-0}{4}) = 0.767$$

La distance de Jaro-Winkler avec avec $l = 2$:

$$\mathrm{d}_w = 0.767 + (2 \times 0.1 \times (1 - 0.767)) = 0.813$$

3.3 Similarité de Levenshtein

La distance de Levenshtein entre mots ou chaînes de caractères est donnée par un calcul assez simple des indications sur le degré de ressemblance de ces chaînes. Sa définition est la suivante :
Si s_1, s_2 sont deux mots, la distance de Levenshtein d est le nombre minimal de remplacements, ajouts et suppressions de lettres pour passer du mot s_1 au mot s_2.
d satisfait bien à la définition des distances [21] :
s_1, s_2 et s_3 étant trois mots quelconques (éventuellement vides),

- $d(S_1, S_2)$ est un réel positif ou nul ;
- $d(S_1, S_2) = 0$ si et seulement si $S_1 = S_2$;
- $d(S_1, S_2) = d(S_2, S_1)$ (symétrie) ;
- $d(S_1, S_3)$ est inférieur ou égal à $d(S_1, S_2) + d(S_2, S_3)$ (inégalité triangulaire) ;

On peut remarquer en outre que $d(S_1, S_2)$ est un entier. [21]

3.3.1 Algorithme de distance de Levenshtein [21]

Algorithm 1 entier DistanceDeLevenshtein$((caractere chaine1[1..longueurChaine1], chaine2[1..longueurChaine2])

algorithm.1

Require: déclarer entier d[0..longueurChaine1, 0..longueurChaine2] {// d est un tableau de longueurChaine1+1 rangées et longueurChaine2+1 colonnes // d est indexé à partir de 0, les chaînes à partir de 1}

Require: déclarer entier i, j, coûtSubstitution {// // i et j itèrent sur chaine1 et chaine2} {}

1: **for** i = 0 to $longueurChaine1$ **do**
2: $d[i, 0] \leftarrow i$
3: **end for**
4: **for** j = 0 to $longueurChaine2$ **do**
5: $d[0, j] \leftarrow j$
6: **end for**
7: **for** i = 0 to $longueurChaine1$ **do**
8: **for** j = 0 to $longueurChaine2$ **do**
9: **if** chaine1[i-1] = chaine2[j-1] **then**
10: $coûtSubstitution \leftarrow$0
11: **else**
12: $coûtSubstitution \leftarrow$1
13: **end if**
14: **end for**
15: **end for**
16: $d[i, j]\leftarrow$minimum(d[i-1, j] + 1, d[i, j-1] + 1, d[i-1, j-1] + coûtSubstitution)
17: **return** $d[longueurChaine1, longueurChaine2]$

Exemple d'utilisation possible :

Lorsque vous recherchez le mot A dans un lexique L,
- soit A se trouve dans L : $d(A, A) = 0$
- soit A n'est pas dans L et vous suspectez une erreur d'écriture, en utilisant la distance de Levenshtein, vous pouvez rechercher les mots B de L les plus proches de A, tels que par exemple $d(A, B) < k$ (k est petit). L'un de ces mots sera peut-être la bonne orthographe de A.

3.4 Construction du Dataset

pour tester notre implémentation nous avons créer un data set à l'aide des bases de données créés et saisies au niveaux local de quelques administrations de la wilaya de Ghardaïa et nous pensons à amplifier ces données par des noms propres des différentes régions de notre pays et même des pays maghrébins. notre data set comportes enivrent dix-neuf milles noms propres arabes avec les différentes transcriptions en français, d'une telle façon de donner les différentes écritures en français du même noms en arabe, par exemple(Figure 3.2) :

ID	nom_phon	nom_aqui_f	nom_aqui
1	.a..b..a..s.	AABASS	عباس
2	.a..b..a..z..a.	AABAZA	اعبازة
3	.a..b..d..ɛ..l..a..w. .k..h..ã..i.r.a.	AABDELAOUI KHEMIRA	عبد اللاوي خميرة
4	.a..b..ɛ..d.	AABED	عابد
5	.a..b..u..d.	AABOUD	عبود
6	.a..d.	AAD	عاد
7	.a..d..i.	AADI	عادي
8	.a..d..o..õ..ɛ.	AADOUNE	عدون
9	.a..f..i..a.	AAFIA	عافية
10	.a..i..a..d.	AAIAD	عياد
11	.a..i..b..i.	AAIBI	عايبي
12	.a..i..b..i.	AAIBI	عيبي
13	.a..k..ɛ..b.	AAKEB	عاقب
14	.a..l. .h..a..k..ɛ̃.	AAL HAKIM	آل حكيم
15	.a..l. .h..a..m..u.	AAL HAMOU	ال حمو
16	.a..l..ã.	AALEM	عالم
17	.a..l..q..a.	AALGA	علقة
18	.a..l..m..i.	AALMI	لعلى
19	.a..l..m..i.	AALMI	لعلمي
20	.a..l..w..ã..i.	AALWANI	العلواني
21	.a..m..a..i..a.r	AAMAIAR	عمير
22	.a..m..a.r.a.	AAMARA	عمارة
23	.a..m..ɛ.r	AAMER	أعمر
24	.a..m..i. .m..u..s..a.	AAMI MOUSSA	عمي موسى
25	.a..m..i. .s..a..i..d.	AAMI SAID	عمي سعيد
26	.a..ã..i..l.	AANIL	عنيل
27	.a.r.u..s..i.	AAROUSI	عروصي
28	.a..z..a..l.	AASAL	عسال
29	.a..s..a..l..i.	AASSALI	عسالي
30	.a..t..a. .a..l..a.h	AATA ALLAH	أعطاة
31	.a..z..i.	AAZI	عزي
32	.a..z..i..z	AAZIZ	

FIGURE 3.2 – structure de la liste des noms personnes écrits en français étiquetés en ses écritures arbes et phonétiques(corpus)

3.5 Implémentattion de la solution

Dans cette section nous allons présenter la solutions informatique implémentée pour traiter la problématique posées précédemment. pour ce la nous avons utilisé le C# qui est un langage de programmation orientée objet, commercialisé par Microsoft depuis 2002 et destiné à développer sur la plateforme Microsoft .NET, sa Syntaxe est très proche de C++ et Java.

FIGURE 3.3 – Logo du langage C#.net

3.5.1 Présentation de notre application

Dans cette sous - section , nous présentons notre application crée, i.e : quelques formes avec une petite explication : première forme c'est la forme de menu principale :

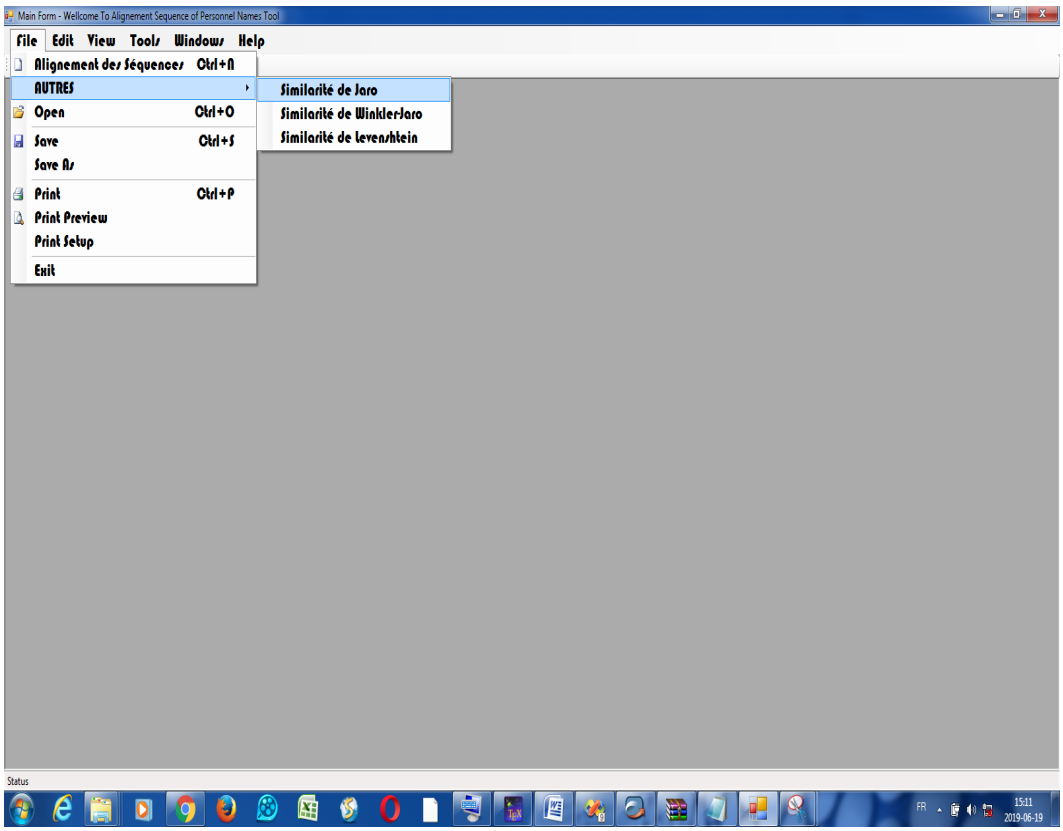


FIGURE 3.4 – Forme de menu principale

la forme principale c'est la forme d'accueil dés que le chargement de l'application, elle contient une bar de menu principale qui sert à gérer l'accès aux différentes fonctionnalités de l'application, tels que les différentes formes des similarités et ses paramétrages.
puis, les deux formes des paramétrages,comme suite :
la forme des paramètres d'alignement des séquences des alphabets françaises(des scores initiaux)

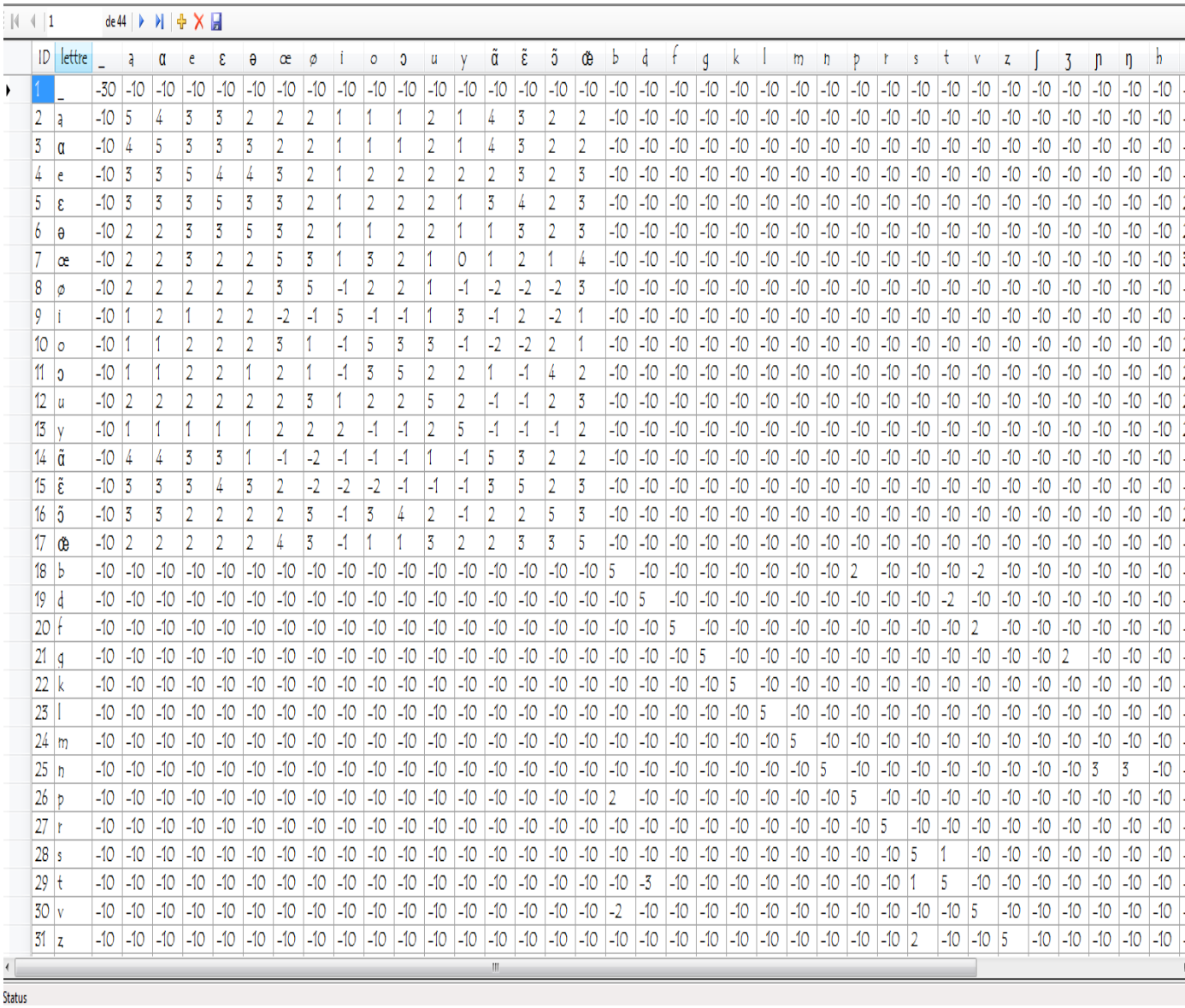
Fr Alphabet Score Parametres

1 de 33

ID	lettre	_	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	é	è	ê	ë	ï	â
1	_	-30	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
2	a	-5	5	-10	-10	-10	-2	-10	-10	0	-3	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-1	-10	-10	-10	-5	-10	-2	-2	-2	-2	-3	5
3	b	-5	-10	5	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-7	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
4	c	-5	-10	-10	5	-10	-10	-10	-5	-5	-10	-10	2	-10	-10	-10	-10	-10	-2	-10	3	-4	-10	-10	-10	2	-10	-10	-10	-10	-10	-10	-10	-10
5	d	-5	-10	-10	-10	5	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
6	e	-5	0	-10	-10	-10	5	-10	-10	0	-2	-10	-10	-10	-10	-10	-1	-10	-10	-10	-10	-10	-5	-10	-10	-10	-3	-10	2	2	2	3	-2	-2
7	f	-5	-10	-10	-10	-10	-10	5	-10	0	-10	-10	-10	-10	-10	-10	-10	3	-10	-10	-10	-10	-10	-7	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
8	g	-5	-10	-10	-5	-10	-10	-10	5	0	-10	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	-10	-10	-10	-10	-10	-10
9	h	0	0	0	-5	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	i	-5	-2	-10	-10	-10	-10	-10	-10	0	5	-10	-10	-10	-10	-10	-7	-10	-10	-10	-10	-10	-2	-10	-10	-10	3	-10	-1	-1	-1	-1	5	-2
11	j	-5	-10	-10	-10	-10	-10	-10	2	0	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
12	k	-5	-10	-10	2	-10	-10	-10	-10	0	-10	-10	5	-10	-10	-10	-10	-10	3	-10	-10	-10	-10	-10	-10	3	-10	-10	-10	-10	-10	-10	-10	-10
13	l	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
14	m	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
15	n	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
16	o	-5	-2	-10	-10	-10	-1	-10	-10	0	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-10	-10	-10	-5	-10	-2
17	p	-5	-10	-2	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
18	q	-5	-10	-10	-2	-10	-10	-10	-10	0	-10	-10	3	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
19	r	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
20	s	-5	-10	-10	3	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-2	-10	-10	-10	-10	-10	-1	-10	-10	-10	-10	-10	-10
21	t	-5	-10	-10	-4	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-2	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
22	u	-5	-2	-10	-10	1	-10	-10	-10	0	3	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	3	2
23	v	-5	-10	-7	-10	-10	-10	2	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	2	-10	-10	-10	-10	-10	-10	-10	-10	-10
24	w	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	5	-10	-10	-10	-10	-10	-10	-10	-10	-10
25	x	-5	-10	-10	2	-10	-10	-10	1	0	-10	-10	3	-10	-10	-10	-10	-10	3	-10	3	-10	-10	-10	-10	5	-10	2	-10	-10	-10	-10	-10	-10
26	y	-5	-5	-10	-10	-10	-10	-10	-10	0	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-5	-10	-10	-10	5	-10	-10	-10	-10	-10	4	2
27	z	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10	-10	1	-10	5	-10	-10	-10	-10	-10	-10
28	é	-5	-2	-10	-10	-10	3	-10	-10	0	-1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	-10	2	-10	5	4	4	3	3	-2
29	è	-5	-2	-10	-10	-10	3	-10	-10	0	-1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	-10	2	-10	4	5	4	3	3	-2
30	ê	-5	-2	-10	-10	-10	3	-10	-10	0	-1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	-10	2	-10	4	4	5	3	3	-2

FIGURE 3.5 – Forme des paramètres de l'alignement des séquences des alphabets françaises

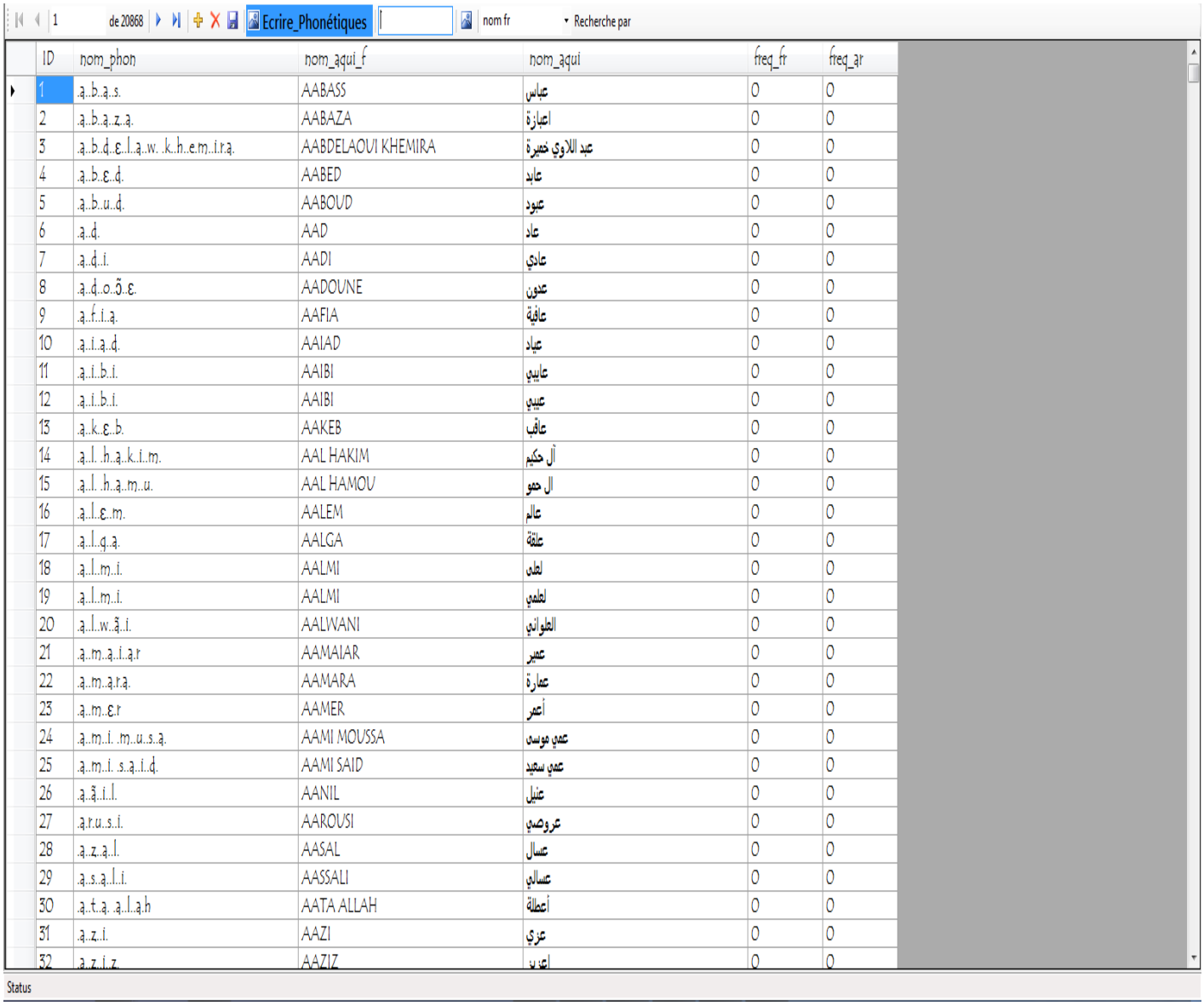
et la forme des paramètres d'alignement des séquences des symboles de l'I.P.A françaises(des scores initiaux)



ID	lettre	_	a	ɑ	e	ɛ	ə	œ	ø	i	o	ɔ	u	y	ɑ̃	ɛ̃	ɔ̃	œ̃	b	d	f	g	k	l	m	n	p	r	s	t	v	z	ʃ	ʒ	ɲ	ŋ	h
1	_	-30	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
2	a	-10	5	4	3	3	2	2	2	1	1	1	2	1	4	3	2	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
3	ɑ	-10	4	5	3	3	3	2	2	1	1	1	2	1	4	3	2	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
4	e	-10	3	3	5	4	4	3	2	1	2	2	2	2	2	3	2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
5	ɛ	-10	3	3	3	5	3	3	2	1	2	2	2	1	3	4	2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
6	ə	-10	2	2	3	3	5	3	2	1	1	2	2	1	1	3	2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
7	œ	-10	2	2	3	2	2	5	3	1	3	2	1	0	1	2	1	4	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
8	ø	-10	2	2	2	2	2	3	5	-1	2	2	1	-1	-2	-2	-2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
9	i	-10	1	2	1	2	2	-2	-1	5	-1	-1	1	3	-1	2	-2	1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
10	o	-10	1	1	2	2	2	3	1	-1	5	3	3	-1	-2	-2	2	1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
11	ɔ	-10	1	1	2	2	1	2	1	-1	3	5	2	2	1	-1	4	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
12	u	-10	2	2	2	2	2	2	3	1	2	2	5	2	-1	-1	2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
13	y	-10	1	1	1	1	1	2	2	2	-1	-1	2	5	-1	-1	-1	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
14	ɑ̃	-10	4	4	3	3	1	-1	-2	-1	-1	-1	1	-1	5	3	2	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
15	ɛ̃	-10	3	3	3	4	3	2	-2	-2	-2	-1	-1	-1	3	5	2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
16	ɔ̃	-10	3	3	2	2	2	2	3	-1	3	4	2	-1	2	2	5	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
17	œ̃	-10	2	2	2	2	2	4	3	-1	1	1	3	2	2	3	3	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
18	b	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10	-2	-10	-10	-10	-10	-10	-10
19	d	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-10	-10
20	f	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10	-10	-10	-10
21	g	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10
22	k	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
23	l	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
24	m	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
25	n	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	3	3	-10
26	p	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
27	r	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10
28	s	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	1	-10	-10	-10	-10	-10	-10	-10
29	t	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-3	-10	-10	-10	-10	-10	-10	-10	-10	1	5	-10	-10	-10	-10	-10	-10	-10
30	v	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10
31	z	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	5	-10	-10	-10	-10	-10

FIGURE 3.6 – Forme des paramètres de l'alignement des séquences des symboles I.P.A françaises

Ainsi la forme de la liste des noms algériens(locaux) écrits en français et étiquetés par ses écritures en arabes et, puis ses représentations phonétiques, comme suite(Figure 3.7) :



1 de 20868 Ecrire_Phonétiques nom fr Recherche par

ID	nom_phon	nom_aqui_f	nom_aqui	freq_fr	freq_ar
1	.a..b..a..s.	AABASS	عباس	0	0
2	.a..b..a..z..a.	AABAZA	اعبازة	0	0
3	.a..b..d..ɛ..l..a..w. ..k..h..e.m..i.r.a.	AABDELAOUI KHEMIRA	عبد اللاوي خميرة	0	0
4	.a..b..ɛ..d.	AABED	عابد	0	0
5	.a..b..u..d.	AABOUD	عبود	0	0
6	.a..d.	AAD	عاد	0	0
7	.a..d..i.	AADI	عادي	0	0
8	.a..d..o..õ..ɛ.	AADOUNE	عدون	0	0
9	.a..f..i..a.	AAFIA	عافية	0	0
10	.a..i..a..d.	AAIAD	عياد	0	0
11	.a..i..b..i.	AAIBI	عايبي	0	0
12	.a..i..b..i.	AAIBI	عيبي	0	0
13	.a..k..ɛ..b.	AAKEB	عاقب	0	0
14	.a..l. .h..a..k..i..m.	AAL HAKIM	ألحكيم	0	0
15	.a..l. .h..a..m..u.	AAL HAMOU	ال حمو	0	0
16	.a..l..ɛ..m.	AALEM	عالم	0	0
17	.a..l..g..a.	AALGA	علقة	0	0
18	.a..l..m..i.	AALMI	لعلمي	0	0
19	.a..l..m..i.	AALMI	لعلمي	0	0
20	.a..l..w..ã..i.	AALWANI	العواني	0	0
21	.a..m..a..i..a.r	AAMAIAR	عمير	0	0
22	.a..m..a.r.a.	AAMARA	عمارة	0	0
23	.a..m..ɛ.r	AAMER	أعمر	0	0
24	.a..m..i. .m..u..s..a.	AAMI MOUSSA	عمي موسى	0	0
25	.a..m..i. .s..a..i..d.	AAMI SAID	عمي سعيد	0	0
26	.a..ã..i..l.	AANIL	عنيل	0	0
27	.a.r.u..s..i.	AAROUSI	عروصي	0	0
28	.a..z..a..l.	AASAL	عسال	0	0
29	.a..s..a..l..i.	AASSALI	عسالي	0	0
30	.a..t..a. .a..l..a.h	AATA ALLAH	أعطاة	0	0
31	.a..z..i.	AAZI	عزي	0	0
32	.a..z..i..z.	AAZIZ	اعزين	0	0

Status

FIGURE 3.7 – Liste des noms algériens écrits en français étiquetés par ses écritures en arabes (corpus)

Ainsi, les formes des différentes similarités utilisées :
la forme commune des divers similarités comme suite(Figure 3.8) :

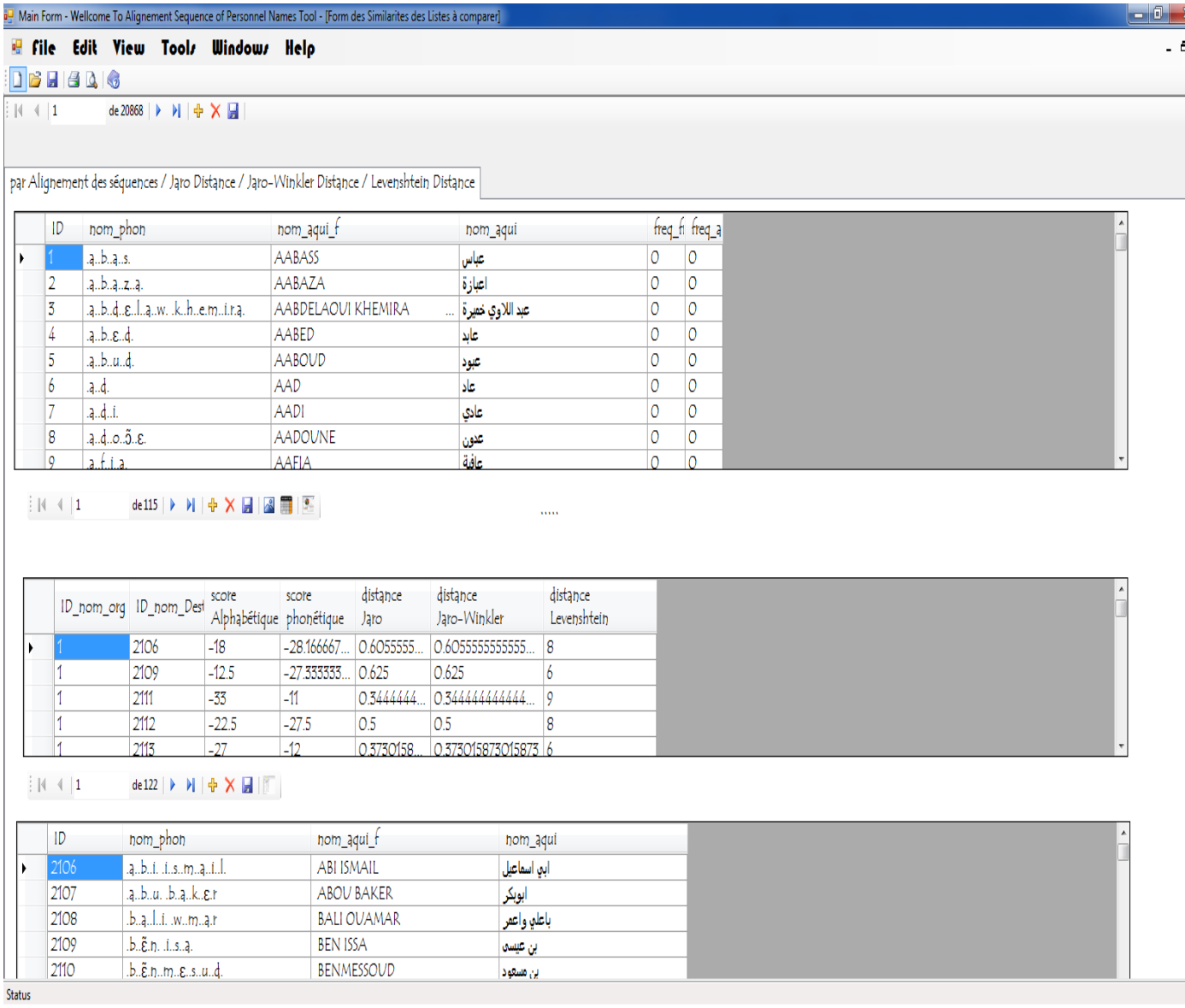


FIGURE 3.8 – Forme commune des divers similarités utilisées

la forme l'alignement des séquences comme suite(Figure 3.9) :

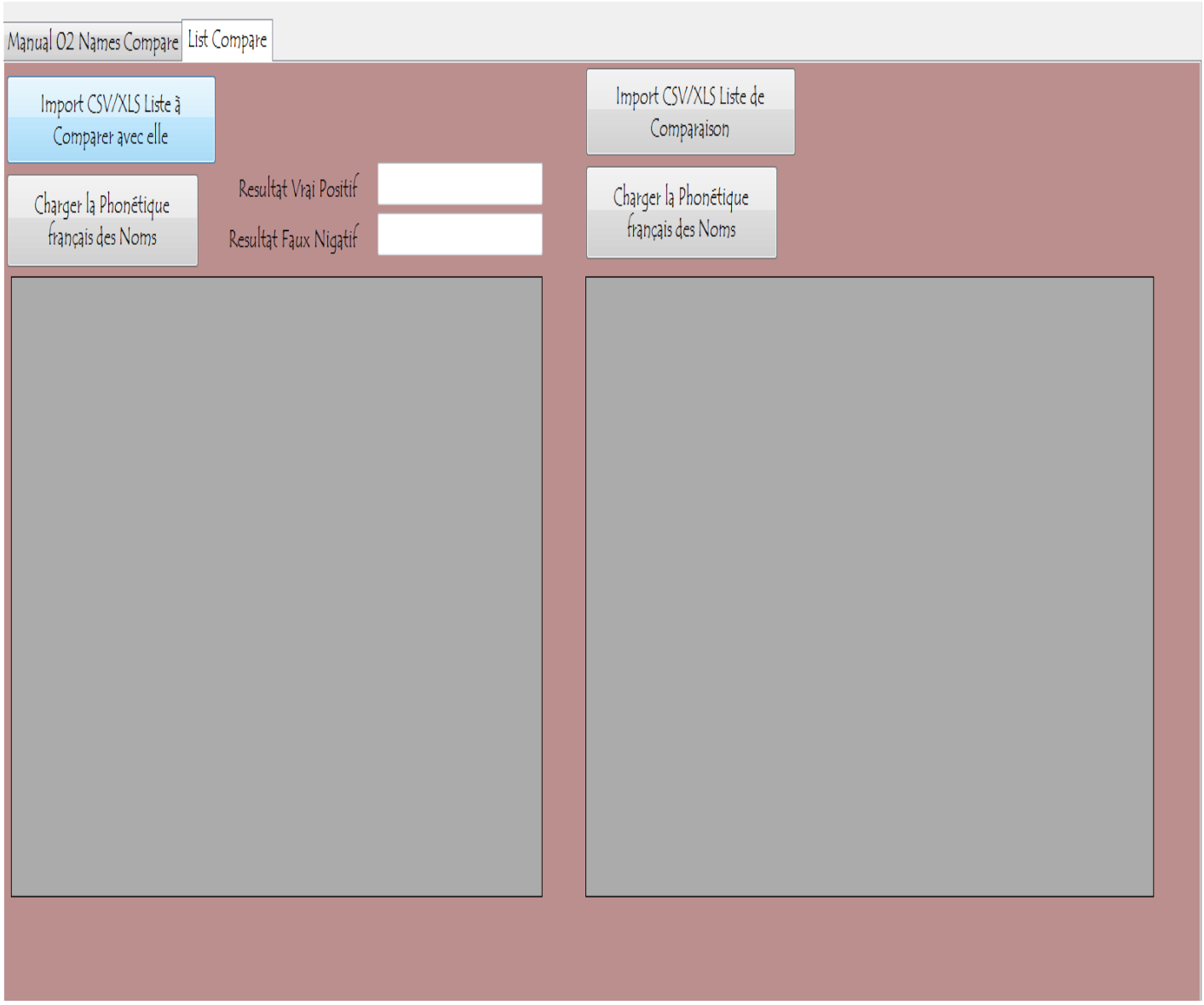


FIGURE 3.9 – Forme l'alignement des séquences

la forme de similarité de distance jaro comme suite(Figure 3.10) :

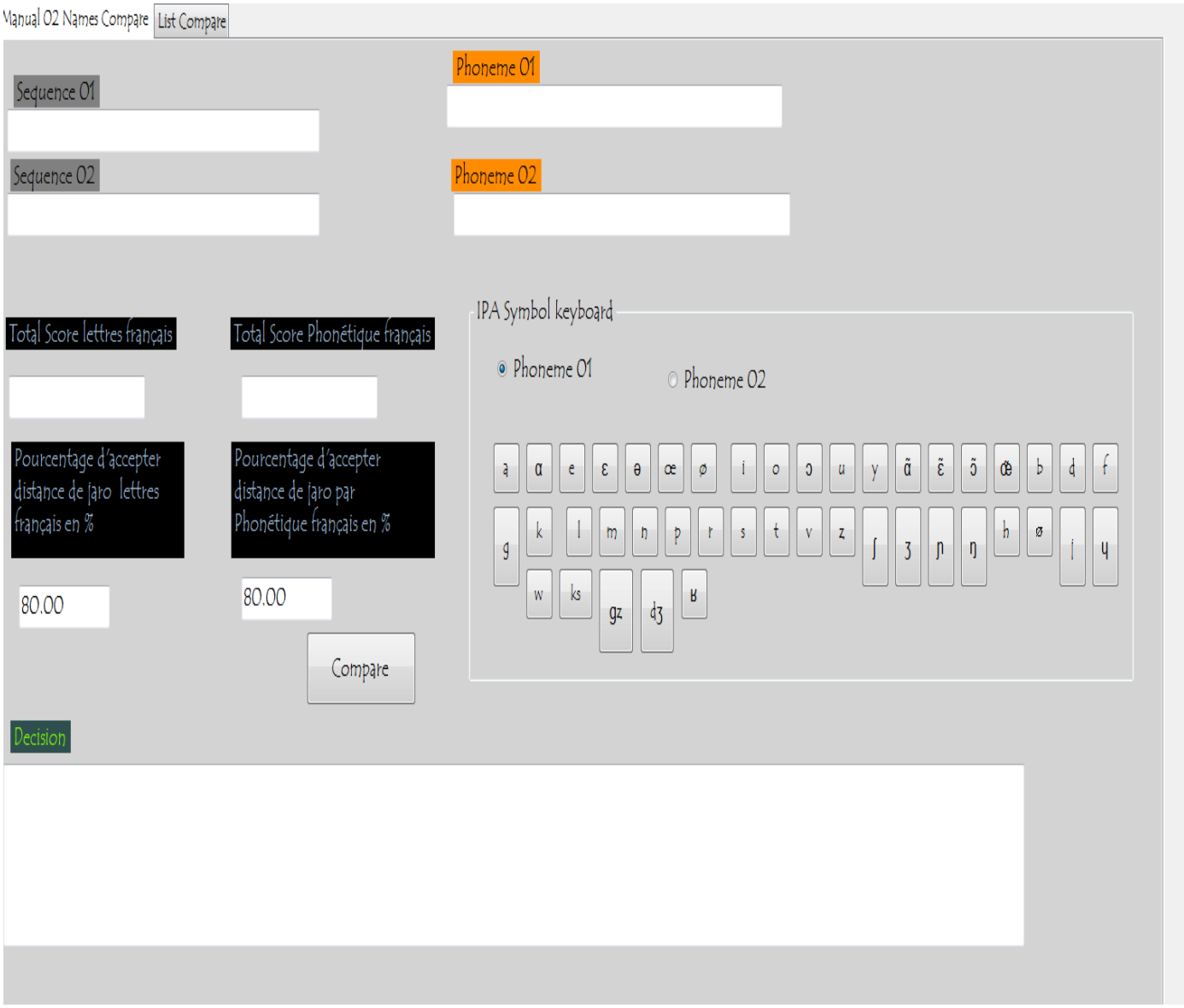


FIGURE 3.10 – Forme de similarité de distance jaro

la forme de similarité de distance jaro-winkler comme suite(Figure 3.11) :

FIGURE 3.11 – Forme de similarité de distance jaro-winkler

la forme de similarité de distance Levenshtein comme suite(Figure 3.12) :

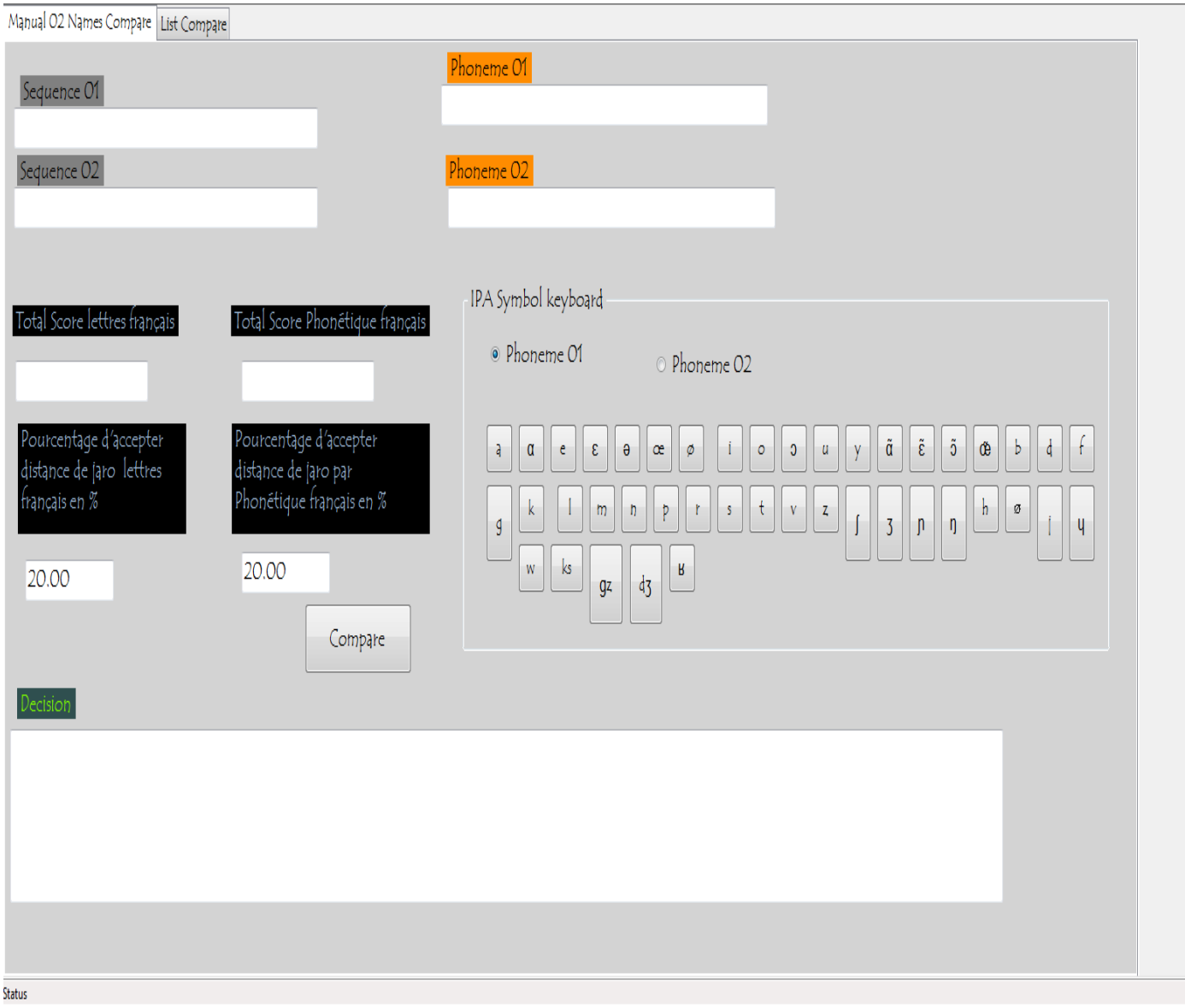
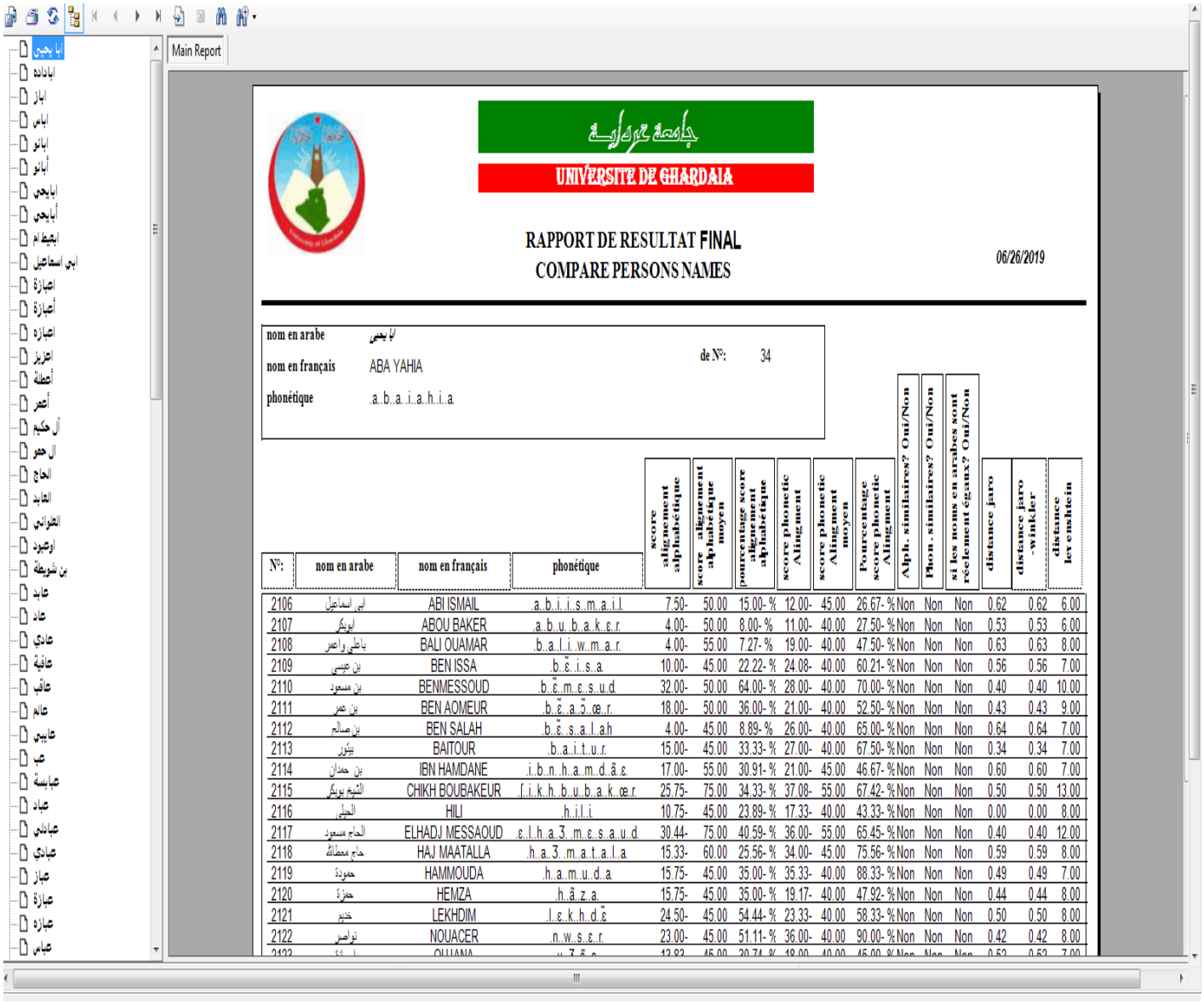


FIGURE 3.12 – Forme de similarité de distance Levenshtein

Finalement, les états de sorties des statistiques, représentent les résultats des calcules des différentes scores des alignements des lettres alphabétiques et des symboles phonétiques I.P.A, ainsi les distances des différentes similarités : jaro, jaro - winkler et levenshtein.
la forme de (la Figure 3.13) , donne une aidé sur ceux qui nous avons cité précédemment.



N°:	nom en arabe	nom en français	phonétique	score alignement alphabétique	score alignement alphabétique moyen	pourcentage score alignement alphabétique	score phonetic Alingment	score phonetic Alingment moyen	Pourcentage score phonetic Alingment	Alph. similaires? Oui/Non	Phon. similaires? Oui/Non	si les noms en arabes sont réellement égaux? Oui/Non	distance jaro	distance jaro -winkler	distance levenshtein
2106	ابي اسماعيل	ABI ISMAIL	.a..b..i..i..s..m..a..i..l	7.50-	50.00	15.00- %	12.00-	45.00	26.67- %	Non	Non	Non	0.62	0.62	6.00
2107	ابوبكر	ABOU BAKER	.a..b..u..b..a..k..ɛ..r.	4.00-	50.00	8.00- %	11.00-	40.00	27.50- %	Non	Non	Non	0.53	0.53	6.00
2108	باعلي واعمر	BALI OUAMAR	.b..a..l..i..w..m..a..r.	4.00-	55.00	7.27- %	19.00-	40.00	47.50- %	Non	Non	Non	0.63	0.63	8.00
2109	بن عيسى	BEN ISSA	.b..ɛ̃..i..s..a.	10.00-	45.00	22.22- %	24.08-	40.00	60.21- %	Non	Non	Non	0.56	0.56	7.00
2110	بن مسعود	BENMESSOUD	.b..ɛ̃..m..ɛ..s..u..d.	32.00-	50.00	64.00- %	28.00-	40.00	70.00- %	Non	Non	Non	0.40	0.40	10.00
2111	بن عمر	BEN AOMEUR	.b..ɛ̃..a..ɔ̃..œ..r.	18.00-	50.00	36.00- %	21.00-	40.00	52.50- %	Non	Non	Non	0.43	0.43	9.00
2112	بن صالح	BEN SALAH	.b..ɛ̃..s..a..l..a.h	4.00-	45.00	8.89- %	26.00-	40.00	65.00- %	Non	Non	Non	0.64	0.64	7.00
2113	بيتور	BAITOUR	.b..a..i..t..u..r.	15.00-	45.00	33.33- %	27.00-	40.00	67.50- %	Non	Non	Non	0.34	0.34	7.00
2114	بن حمدان	IBN HAMDANE	.i..b..n..h..a..m..d..â..ɛ.	17.00-	55.00	30.91- %	21.00-	45.00	46.67- %	Non	Non	Non	0.60	0.60	7.00
2115	الشيخ بوبكر	CHIKH BOUBAKEUR	.[..i..k..h..b..u..b..a..k..œ..r.	25.75-	75.00	34.33- %	37.08-	55.00	67.42- %	Non	Non	Non	0.50	0.50	13.00
2116	الحيلي	HILI	.h..i..l..i.	10.75-	45.00	23.89- %	17.33-	40.00	43.33- %	Non	Non	Non	0.00	0.00	8.00
2117	الحاج مسعود	ELHADJ MESSAOUD	.ɛ..l..h..a..ʒ..m..ɛ..s..a..u..d.	30.44-	75.00	40.59- %	36.00-	55.00	65.45- %	Non	Non	Non	0.40	0.40	12.00
2118	حاج معطاله	HAJ MAATALLA	.h..a..ʒ..m..a..t..a..l..a	15.33-	60.00	25.56- %	34.00-	45.00	75.56- %	Non	Non	Non	0.59	0.59	8.00
2119	حمودة	HAMMOUDA	.h..a..m..u..d..a	15.75-	45.00	35.00- %	35.33-	40.00	88.33- %	Non	Non	Non	0.49	0.49	7.00
2120	حمزة	HEMZA	.h..ã..z..a.	15.75-	45.00	35.00- %	19.17-	40.00	47.92- %	Non	Non	Non	0.44	0.44	8.00
2121	خديم	LEKHDIM	.l..ɛ..k..h..d..ɛ̃	24.50-	45.00	54.44- %	23.33-	40.00	58.33- %	Non	Non	Non	0.50	0.50	8.00
2122	نواصر	NOUACER	.n..w..s..ɛ..r.	23.00-	45.00	51.11- %	36.00-	40.00	90.00- %	Non	Non	Non	0.42	0.42	8.00

FIGURE 3.13 – Rapport final des résultat : page 1

3.6 Discussion des résultats obtenues

Dans cette section, nous allons discuter les paramètres choisis et les résultats obtenus comme suite :
les paramètres sont arbitrairement choisis, suivant notre expérience au travail de la saisie des noms propres , au cours des taches quotidiennes à l'administration publique, nous estimons un score sur un échelle de 5 point de deux sens i.e : le sens positif : un gain qui présente la possibilité de substituer un caractère (resp.phonème au phonétique) par un autre caractère dans quelques fois reliées par sa prononciation et suivant le nombre d'apparition de cette substitution,et l'autre sens i.e : le sens négatif la pénalisation de substitution d'un caractère par un autre suivant une distance estimée de l'effet de remplacement par ce caractère, risque de changer l'entité du nom propre . Ces paramètres nous avons déjà discuter sur la section de **'Similarité proposé à base d'alignement'** du **chapitre 2 : Similarité des séquences**,(l'annexe c détaille les valeurs des scores choisis pour l'alignement alphabétique et phonétique).
avant de voir le résultat, il faut faire entrer les pourcentages des seuils d'accepter les résultats affichés, inclut des statistiques sur eux.
les formes des seuils(voir Figure 3.14 et la 3.15)

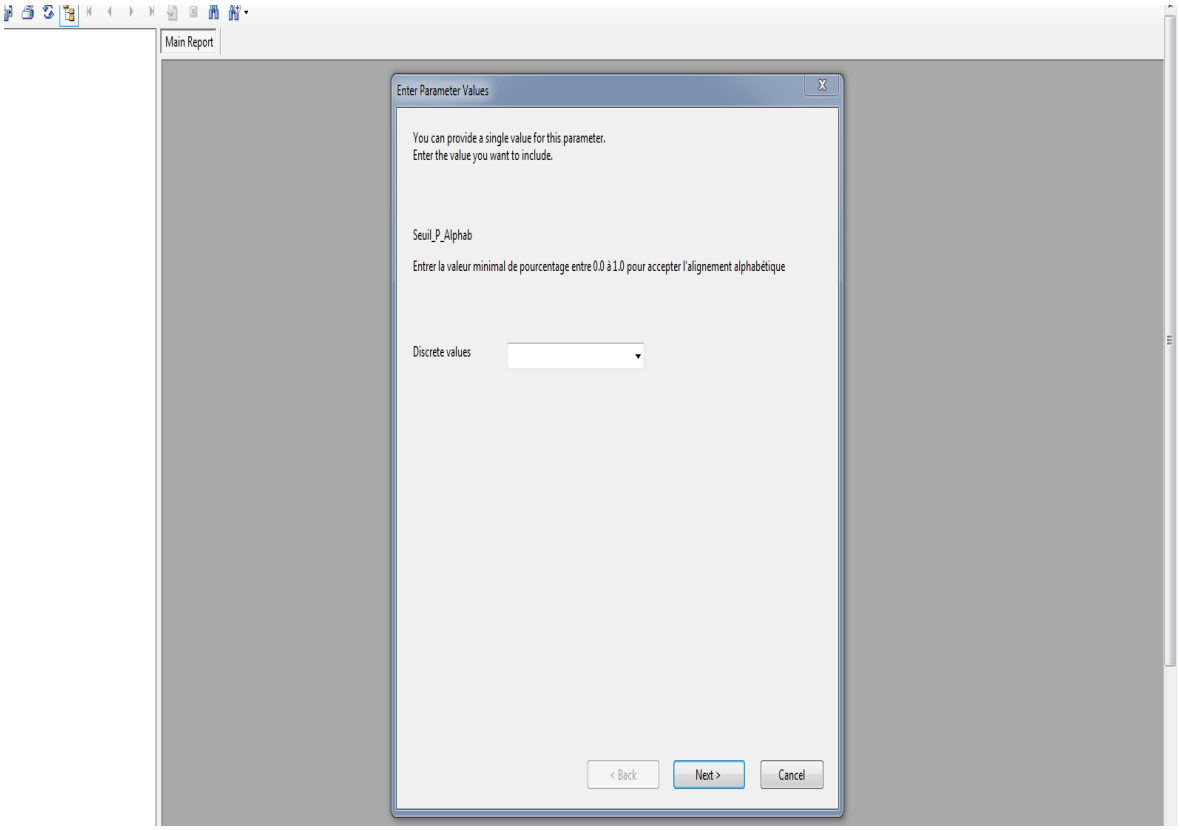


FIGURE 3.14 – paramètre de seuil d'accepter le résultat l'alignement alphabétique

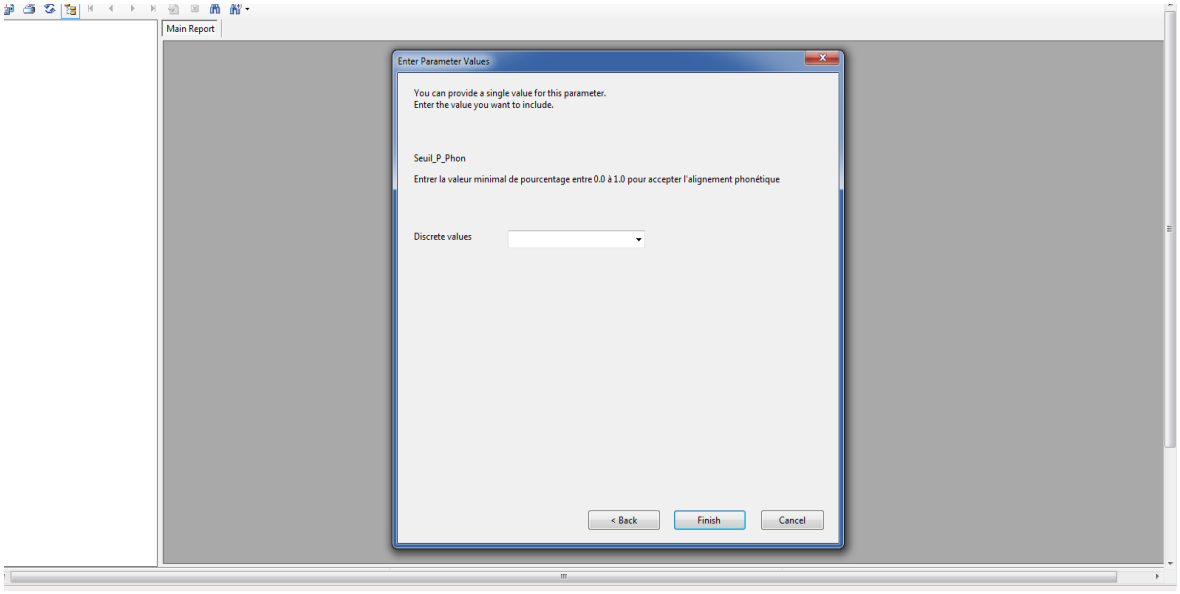


FIGURE 3.15 – les paramètres de seuil d'accepter le résultat de l'alignement phonétique

Nous avons choisit 40.00 % comme seuil d'accepter les résultats de l'alignement alphabétique (idem pour les résultats de l'alignement phonétique).

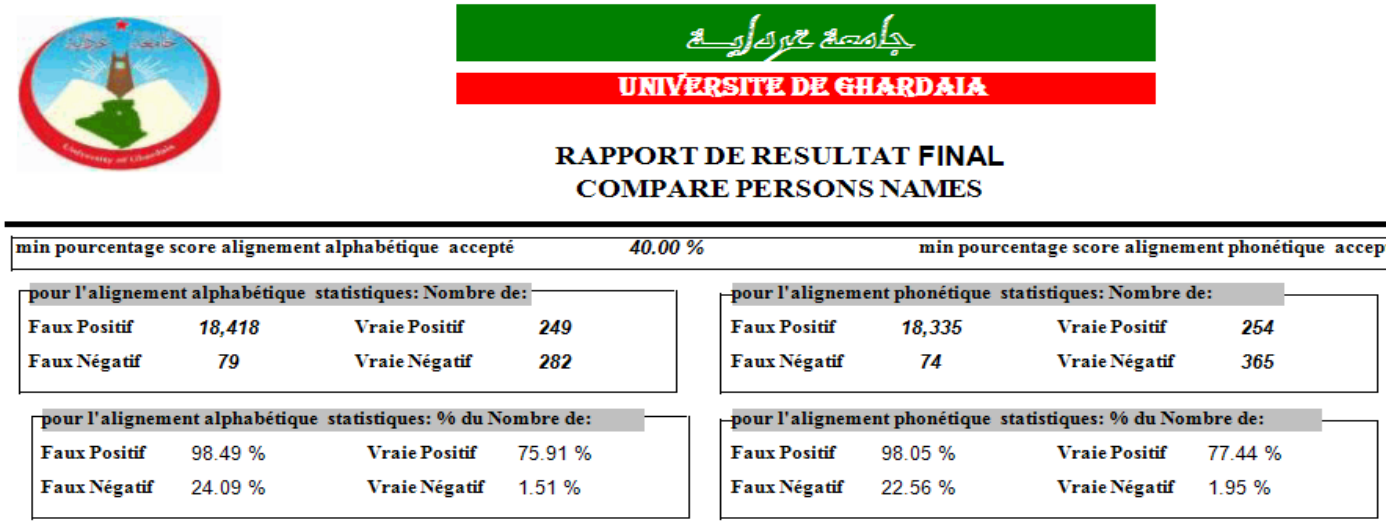


FIGURE 3.16 – Page du résultat final

les résultats obtenus(voire Figure 3.16), précédemment , surtout les erreurs au vraie négatif et faux négatif, sont dûs aux erreurs d'étiquetage en arabe (exemple à la Figure 3.17)

جامعة غرداية

UNIVERSITE DE GHARDAIA

RAPPORT DE RESULTAT FINAL
COMPARE PERSONS NAMES

nom en arabe ابا يحيى
nom en français ABA YAHIA
phonétique .a..b..a..ɪ..a..h..i..a
de N°: 34

N°:	nom en arabe	nom en français	phonétique	score alignement alphabétique	score alignement alphabétique moyen	pourcentage score alignement alphabétique	score phonetic Alingment	score phonetic Alingment moyen	Pourcentage score phonetic Alingment	Alph. similaires? Oui/Non
2150	عزى	AZZI	.a..z..i.	12.75-	45.00	28.33- %	18.00-	40.00	45.00- %	Non
2151	اعزيز	AZIZ	.a..z..i..z.	13.00-	45.00	28.89- %	28.00-	40.00	70.00- %	Non
2152	عبد المجيد	ABDEL MADJID	.a..b..d..ɛ..l..m..a..ʒ..i..d.	12.83-	60.00	21.39- %	22.00-	50.00	44.00- %	Non
2153	ابا يحيى	ABOU YAHIA	.a..b..u..i..a..h..i..a.	29.00	50.00	58.00 %	37.00	40.00	92.50 %	Oui
2154	ابايحى	ABA YAHIA	.a..b..a..i..a..h..i..a.	40.00	45.00	88.89 %	40.00	40.00	100.00 %	Oui
2155	أبايحى	ABA YAHIA	.a..b..a..i..a..h..i..a.	40.00	45.00	88.89 %	40.00	40.00	100.00 %	Oui
2156	عباد	ABAADE	.a..b..a..d..ɛ.	0.50	45.00	1.11 %	22.00-	40.00	55.00- %	Non
2157	عبد الجبار	ABADELJABAR	.a..b..a..d..ɛ..l..ʒ..a..b..a..r.	4.00-	55.00	7.27- %	17.00-	55.00	30.91- %	Non
2158	عبايد	ABAID	.a..b..a..i..d.	2.00-	45.00	4.44- %	27.00-	40.00	67.50- %	Non
2159	عبيد الله	ABID ALLAH	.a..b..i..d. .a..l..a.h	5.00	50.00	10.00 %	3.00-	40.00	7.50- %	Non
2160	عبيدى	ABIDI	.a..b..i..d..i.	5.33-	45.00	11.85- %	24.00-	40.00	60.00- %	Non
2161	ابانو	ABANOU	.a..b..ã..u.	1.00-	45.00	2.22- %	18.00-	40.00	45.00- %	Non
2162	أبانو	ABANO	.a..b..ã..o.	5.00-	45.00	11.11- %	19.00-	40.00	47.50- %	Non
2163	أبانو	ABANOU	.a..b..ã..u.	1.00-	45.00	2.22- %	18.00-	40.00	45.00- %	Non
2164	عباس	ABAS	.a..b..a.s	10.00-	45.00	22.22- %	28.00-	40.00	70.00- %	Non
2165	عباسى	ABASSI	.a..b..a..s..i.	2.50	45.00	5.56 %	20.00-	40.00	50.00- %	Non
2166	ابايحى	ABAYAHIA	.a..b..a..i..a..h..i..a.	35.00	45.00	77.78 %	30.00	40.00	75.00 %	Oui

FIGURE 3.17 – Exemple d'une erreur d'étiquetage

3.7 Conclusion

Dans ce chapitre consacré à l'implémentation des études précédentes, et des algorithmes, nous les exerçons dans une application informatique, quelle nous la considéré comme un outil de comparaison des résultats obtenus par l'expérimentation des ces algorithmes .

Conclusion Générale

TOUT au long de la préparation de notre projet de fin d'études, nous avons essayé de mettre en pratique les connaissances acquises durant nos études universitaires et cela dans le but de répondre à la problématique posée concernant la similarité des séquences courtes le cas des noms propres et de réaliser une application pour aboutir à la solution voulue. Au cours de cette mémoire, nous avons étudié les notions phonétiques de la langue française que nous avons utilisée comme un pré-traitement pour standardiser la transcription des noms propres algériens en écriture française, en basant de ces notions nous avons proposé une correspondance pour la translittération des lettres arabes en français. Après nous avons présenté l'alignement des séquences comme une technique de similarité des séquences courtes et nous avons proposé des tables de score pour l'alphabet et la phonétique.
Dans la partie implémentation nous avons créé un dataset composé de plus de dix-neuf milles noms arabes transcris en français, ces noms sont obtenus des administrations de la wilaya de Ghardaïa, et en C# nous avons réalisé une application qui répond au problème posé, nous avons obtenu des résultats qui sont affectés par les scores choisis et en même temps nous avons comparé l'alignement avec d'autre techniques tel que : jaro, jaro-winkler et levenshtein.
. Comme perspective, nous souhaitons pour les prochains projets de fin d'étude de bien évaluer notre travail et approche proposés, élargir notre dataset vers un dataset national et pourquoi pas maghrébin, étudier l'influence des valeurs des scores sur les résultats obtenus, et de passer à la comparaison arabe arabe et arabe français.

Bibliographie

Bibliographie

[1] P. Christen. Joint Computer Science Technical Report Series. A Comparison of Personal Name Matching : Techniques and Practical Issues. Department of Computer Science Faculty of Engineering and Information Technology. Sciences Laboratory Research School of Information Sciences and Engineering. Sciences Laboratory Research School of Information Sciences and Engineering. publisher : THE AUSTRALIAN NATIONAL UNIVERSITY (A.N.U), September 2006

[2] J-M. Kalmbach, Phonétique et prononciation du français pour apprenants. *http ://research.jyu.fi/phonfr/Manuel*$_2011.html, Kieltenlaitos,$
$Jyväskylänyliopisto, UniversitédeJyväskylä, Finlande, 2017.$
$ISBN978 - 951 - 39 - 4424 - 7$
$Version1.1.(9/2013)$

[3] J, Shawe-Taylor and N, Cristianini Kernel Methods for Pattern Analysis. *book* , Cambridge University Press, 2004. New York, NY, USA.

[4] S, Bellaouar cours module ELT. *document* , université Ghardaia, 2019.

[5] A. ELSAADANI, PHONÉTIQUE DU FRANÇAIS POUR LES DÉBUTANTS ARABOPHONES (F.114). *Rapport pour la premiéres année de la langue française*, Université de de MANSOURA, 2018.

[6] D. E.KOULOUGHLI, Grammaire de l'arabe d'aujourd'hui Poche. *livre de poche*, LIBRARIE dialogues, 1994.

[7] H. Saadane, N. Semmar. Transcription des noms arabes en écriture latine. *Thèse*, Université Stendhal - Grenoble III, 2013.

[8] H. Saadane, N. Semmar. Revue de l'Information Scientifique et Technique. Transcription des noms arabes en écriture latine url de publisher : www.asjp.cerist.dz. *pages 51-62*.

[9] D. Ayoun. Introduction à la phonétique française. University of Arizona, 2010 *pages 01-10*.

[10] G. Straka. Aide-Mémoire phonétique. EUGENE IONESCO,2019 *pages 01-36*.

[11] V. Derrien . Heuristiques pour la résolution du problème d'alignement multiple publisher : Université d'Angers. *Thèse de doctorat* Nd'ordre 885, 2008.

[12] K. Mezhoud . Alignement de séquences Principes et méthodes Karim Mezhoud Ir. publisher : Centre national des Sciences et Technologies Nucléaires Sidi Thabet – Tunis. *Agronome PhD, Toxicologie, Protéomique, Bioinformatique,2019* .

[13] A. Layeb. Approche quantique évolutionnaire pour l'alignement multiple de séquences en bioinformatique. publisher :Université Mentouri de Constantine . *Mémoire du diplôme de Magistère en Informatique : Information et Computation* , 2005.

[14] Kh. Djerbouai . Alignement multiple des séquences protéiques par l'algorithme de recherche tabou. publisher : UNIVERSITE MOHAMED BOUDIAF - M'SILA . *Mémoire Du diplôme de Master Académique*, 2017 /2018.

[15] V. Derrien, J-M. Richer,J-K. Hao. Plasma, un nouvel algorithme progressif pour l'alignement multiple de séquences publisher : LERIA – Université d'Angers, 2 Bd Lavoisier, 49045 Angers, France . *Premières Journées Francophones de Programmation par Contraintes, CRIL - CNRS FRE 2499, Jun 2005, Lens, pp.39-48.*

[16] J. Shawe-Taylor, N. Cristianini. Kernel Methods for Pattern Analysis. publisher : Cambridge University Press, New York,NY, USA. *Livre, 2004.*

[17] N. Benlahrache. Optimisation Multi-Objectif Pour l'Alignement Multiple de Séquences. publisher : Université Mentouri de Constantine. *Mémoire du diplôme de Magistère en Informatique , 2007.*

[18] C. Leslie , E. Eskin, W-S. NOBLE. The spectrum kernel : a string kernel for svm protein classification. publisher : Department of Computer Science, Columbia University, New York, NY 10027 . *Proceedings of the Pacific Symposium on Biocomputing, 2002. pp. 564–575. , .*

[19] M-A. Jaro. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. publisher : Journal of the American Statistical Association . *http ://dictionnaire.sensagent.leparisien.fr/Distance%20de%20Jaro-Winkler/fr-fr/ .Juin 2019*

[20] Leonardo, Brinardi and Hansun, Seng. Text documents plagiarism detection using Rabin-Karp and Jaro-Winkler distance algorithms. url publisher : www.researchgate.net . *https ://www.researchgate.net/figure/Jaro-Winkler-Distance-Algorithm*$_{fig2_3}16681173, Juin2019.$

[21] J-P. Davalan. Distance de Levenshtein. publisher : © (Copyright) Jean-Paul Davalan 2002-2014 . *http ://jeux-et-mathematiques.davalan.org/lang/algo/lev/index.html* .

[22] M-A. Jaro. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. publisher : Research and Development , System Automation Corporation , Silver Spring , MD , 20910 , USA Published online : 12 Mar 2012. *.To link to this article :http ://dx.doi.org/10.1080/01621459.1989.10478785* .

[23] W-E. Winkler. Overview of Record Linkage and Current Research Directions. publisher : Statistical Research Division U.S. Census Bureau Washington, DC 20233. . *RESEARCH REPORT SERIES (Statistics 2006-2)* .

[24] V-T. Nguyen, Ch. Sallaberry, M. Gaio Mesure de la similarité entre termes et labels de concepts ontologiques. publisher : CORIA 2013, Apr 2013, Neufchâtel, Suisse. . *pp.415-430. ffhal-00847528f* .

Annexes

Annexe A : Représentation des classes d'Alignement et Transcription Phonétique

```
public class AlignementSequences
{
  public string motAligned1, motAligned2;

  public static Dictionary<char,int> fr_alphabet;
  public static Dictionary<String, int> fr_ipa;
  public static int[,] Mscore, Mscore_Ph;
  public static float[,] DPi, DP, DPi_Ph, DP_Ph;
  public static string[] alphabet
  {
    get
    {
      return alphabet;
  }
    set{

    } }
  public AlignementSequences()
  {
    initialiserMscores();
  }
  public static void initialiserMscores()
  {
    //================== Alphabetic Part
    Mscore = new int[33, 33];


    }// chargement de la matrice de score alphabtétique
    //================== Phonetic Part
    //_     a     ɑ     e     ɛ     ə     œ     ø     i     o     ɔ     u
     y     ɑ̃     ɛ̃     ɔ̃     œ̃     b     d     f     g     k     l
     m     n     p     r     s
    // t     v     z     ʃ     ʒ     ɲ     ŋ     h     ø     j     ɥ
     w     ks    gz    dʒ    ʁ

    Mscore_Ph = new int[44, 44];

    }// chargement de la matrice de score phonétique
  }

  public AlignementSequences(int[,] Msc, bool IsPhonetic)
  {
    //-     a     b     c     d     e     f     g     h     i     j     k
     l     m     n     o     p     q     r     s     t     u     v
     w     x     y     z     é     è     ê     ë     ï     â

    fr_alphabet = new Dictionary< char,int>();

            k = 0;
    fr_ipa = new Dictionary< String,int>();
```

```
    }
    public  float getScorePhonValue(String x, String y, int posx, int posy)
    {

    }
    public  float getScoreAlphValue(char x, char y, int posx, int posy)
    {

    }
    //alignement sequence alphabetic
    public float alignSeqAlph(String x, String y)
    {

                DP[i, j] = Math.Max(DP[i - 1, j - 1] + DPi[i, j], Math.Max(DP[i, j - 1] + DPi[0, j], DP[i
- 1, j] + DPi[i, 0]));
            }

        }
       // solutionAlign(x, y, DP, DPi);
        return DP[m, n];
    }
    //alignement sequence phonetic
    public float alignSeqPhon(String x1, String y1)
    {
                DP_Ph[i, j] = Math.Max(DP_Ph[i - 1, j - 1] + DPi_Ph[i, j], Math.Max(DP_Ph[i, j - 1]
+ DPi_Ph[0, j], DP_Ph[i - 1, j] + DPi_Ph[i, 0]));
            }

        }
        return DP_Ph[m, n];

    }


}
```

```
public class PhoneticTranscription
{
    public Dictionary<String, String> phonetic_arab;
    public Dictionary<String, String[]> phonetic_fr;
    public Dictionary<String, String[]> sequence_phoneme_fr;
    public PhoneticTranscription()
    {
        phonetic_arab = new Dictionary<String, String>();

        phonetic_fr = new Dictionary<String, String[]>();

        //      séquence transcrit au phonéme
        //---------------------------------- >3 caractères
```

```
    //----------------------------------- 3 caractères
    sequence_phoneme_fr = new Dictionary<String, String[]>();
    sequence_phoneme_fr.Add("onn", new string[] { "ɔ̃n" });
    sequence_phoneme_fr.Add("omm", new string[] { "ɔ̃m" });
    //**************

}
//////// la fonction d'extraction de la phonétique du mot:
public string Fr_phone(string mot)
{
    string m = mot.Trim();
    int n, i = 1;
   // listBox1.Items.Clear();
    List<String> listBox1 = new List<String>();
    Dictionary<int, String> phonetic_seq = new Dictionary<int, String>();

    return m;
}


//
public string ReplaceFirst(string mot, string replaced, string replaced_by)
{
    if (mot.IndexOf(replaced) < 0)
    {
        return mot;
    }
    int Place = mot.IndexOf(replaced);
    return mot.Remove(Place, replaced.Length).Insert(Place, replaced_by);
}
//////// la fonction de spécifie si un caractère est une voyelle forte:
public bool IsStrongVowel(char c)
{
    switch (c)
    {
        //case "i","e","é","i","è": return false;
        case 'e': return false;
        case 'é': return false;
        case 'è': return false;
        case 'ê': return false;
        case 'ë': return false;
        case 'i': return false;
        case 'ï': return false;
        case 'î': return false;
        case 'y': return false;
        case 'a': return true;
        case 'à': return true;
        case 'â': return true;
        case 'ä': return true;
        case 'o': return true;
        case 'ô': return true;
        case 'ö': return true;
```

```
        case 'u': return true;
        case 'ù': return true;
        case 'û': return true;
        case 'ü': return true;
        default: return false;
    }
}
//////// la fonction de spécifie si un caractère est une voyelle faible:
public bool IsWeakVowel(char c)
{
    switch (c)
    {
        //case "i","e","é","i","è": return false;
        case 'e': return true;
        case 'é': return true;
        case 'è': return true;
        case 'ê': return true;
        case 'ë': return true;
        case 'i': return true;
        case 'ï': return true;
        case 'î': return true;
        case 'y': return true;
        case 'a': return false;
        case 'à': return false;
        case 'â': return false;
        case 'ä': return false;
        case 'o': return false;
        case 'ô': return false;
        case 'ö': return false;
        case 'u': return false;
        case 'ù': return false;
        case 'û': return false;
        case 'ü': return false;
        default: return false;
    }
}
//////// la fonction de spécifie si un caractère est une consonne :
public bool IsConsonant(char c)
{
    switch (c)
    {
        //case "i","e","é","i","è": return false;
        case 'e': return false;
        case 'é': return false;
        case 'è': return false;
        case 'ê': return false;
        case 'ë': return false;
        case 'i': return false;
        case 'ï': return false;
        case 'î': return false;
        case 'y': return false;
        case 'a': return false;
        case 'à': return false;
```

```
                case 'â': return false;
                case 'ä': return false;
                case 'o': return false;
                case 'ô': return false;
                case 'ö': return false;
                case 'u': return false;
                case 'ù': return false;
                case 'û': return false;
                case 'ü': return false;
                default: return true;
            }
        }
        // iif function like of iif vb
        public T Iif<T>(bool expression, T truePart, T falsePart)
        { return expression ? truePart : falsePart; }
        //.....

    }
}
```

Annexe B : Représentation des structures des classes de distance Jaro, Jaro-Winkler et Levenshtein

```
      public static class JaroDistance

  {
     /* The Winkler modification will not be applied unless the
      * percent match was at or above the mWeightThreshold percent
      * without the modification.
      * Winkler's paper used a default value of 0.7
      */
     private static readonly double mWeightThreshold = 0.7;

     /* Size of the prefix to be concidered by the Winkler modification.
      * Winkler's paper used a default value of 4
      */
     private static readonly int mNumChars = 4;
     /// <summary>
     /// Returns the Jaro-Winkler distance between the specified
     /// strings. The distance is symmetric and will fall in the
     /// range 0 (no match) to 1 (perfect match).
     /// </summary>
     /// <param name="aString1">First String</param>
     /// <param name="aString2">Second String</param>
     /// <returns></returns>
     public static double proximity(string aString1, string aString2)
     {
}
public static class JaroWinklerDistance
  {
     /* The Winkler modification will not be applied unless the
      * percent match was at or above the mWeightThreshold percent
      * without the modification.
      * Winkler's paper used a default value of 0.7
      */
     private static readonly double mWeightThreshold = 0.7;

     /* Size of the prefix to be concidered by the Winkler modification.
      * Winkler's paper used a default value of 4
      */
     private static readonly int mNumChars = 4;

     public static double distance(string aString1, string aString2)
     {
        return 1.0 - proximity(aString1, aString2);
     }

     public static double proximity(string aString1, string aString2)
     {

        }
```

```
public static partial class ComparisonMetrics
  {

      public static double LevenshteinDistance(this string source, string target)
      {
          }

}
```

Annexe C : Représentation des Tables : des statistiques sur les gains en nombres des symbols par les transcriptions phonétiques , puis les matrices initiaux des scores alphabétiques et phonétiques

phonème	sous-mot1	sous-mot2	sous-mot3	sous-mot4	sous-mot5	sous-mot6	sous-mot7	sous-mot8	sous-mot9	sous-mot10	sous-mot11	sous-mot12	moyenne de nombres des car. Des cellules non vides	gain moyenne de nombres des car. Transpcrits
[a]	a	à	e	ea									1.25	0.25
[ɑ]	a	â											1.00	0.00
[e]	e	é	ay	æ	œ	ey	er	ez	ë				1.44	0.44
[ɛ]	e	è	ê	ai	ei	aî	ë	eî	ay	ey			1.60	0.60
[ə]	e	o	ai										1.33	0.33
[œ]	eu	œu	œ	u	ue								1.60	0.60
[ø]	eu	œu	eû	œ	ö								1.60	0.60
[i]	i	î	y	ï	hi	ee	ea	ie					1.50	0.50
[o]	o	au	ô	eau	aô	ho	a	ow	aw				1.78	0.78
[ɔ]	o	oi	um	au									1.75	0.75
[y]	ou	oû	aou	aoû	où	ew	oo	ow					2.25	1.25
[y]	u	û	ü										1.00	0.00
[ɑ̃]	en	an	em	am	aon	aën	aen						2.43	0.43
[ɛ̃]	ein	in	ain	im	aim	en	ein	yn	ym	în	ën	em	2.50	0.50
[ɔ̃]	om	on	un										2.00	0.00
[œ̃]	un	um	eun										2.33	0.33
[b]	b	p											1.00	0.00
[d]	d	dh	dd										1.67	0.67
[f]	f	ff	ph										1.67	0.67
[g]	g	c	gg	gh									1.50	0.50
[k]	c	q	k	ch	cch	cc							1.67	0.67
[l]	l	ll											1.50	0.50
[m]	m	mm											1.50	0.50
[n]	n	nn											1.50	0.50
[p]	p	pp											1.50	0.50
[ʀ]	r	rr	rh										1.67	0.67
[s]	s	ç	c	ss	ti								1.40	0.40
[t]	t	th	tt										1.67	0.67
[v]	v	w											1.00	0.00
[z]	s	z	zz	x									1.25	0.25

[ʃ]	ch	tch	sh	sch									2.50	1.50
[ʒ]	g	j	jj										1.33	0.33
[ɲ]	gn												2.00	1.00
[ŋ]	ng												2.00	1.00
[h]	h	hh											1.50	0.50
[j]	ie	ille	il	y									2.25	1.25
[ɥ]	ui	ué											2.00	1.00
[w]	oui	oi	oue	wh	w								2.20	1.20
[ks]	x												1.00	-1.00
[gz]	x												1.00	-1.00
[s]	x												1.00	0.00
[z]	x												1.00	0.00
[Ø]	x												1.00	0.00
[ʒ]	j												1.00	0.00
[dʒ]	j												1.00	-1.00
[ʁ]	j												1.00	0.00
													1.56	**40.51%**
														gain moyenne de nombres des car. Transpcrits

Matrice des scores initiaux pour l'alignement des séquences à la base des lettres d'alphabet française

ID	lettre	_	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	é	è	ê	ë	ï	â
1	**_**	-30	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
2	**a**	-5	5	-10	-10	-10	-2	-10	-10	0	-3	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-1	-10	-10	-10	-5	-10	-2	-2	-2	-2	-3	5
3	**b**	-5	-10	5	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-7	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
4	**c**	-5	-10	-10	5	-10	-10	-10	-5	-5	-10	-10	2	-10	-10	-10	-10	-10	-2	-10	3	-4	-10	-10	-10	2	-10	-10	-10	-10	-10	-10	-10	-10
5	**d**	-5	-10	-10	-10	5	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
6	**e**	-5	0	-10	-10	-10	5	-10	-10	0	-2	-10	-10	-10	-10	-10	-1	-10	-10	-10	-10	-10	-5	-10	-10	-10	-3	-10	2	2	2	3	-2	-2
7	**f**	-5	-10	-10	-10	-10	-10	5	-10	0	-10	-10	-10	-10	-10	-10	-10	3	-10	-10	-10	-10	-10	-7	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
8	**g**	-5	-10	-10	-5	-10	-10	-10	5	0	-10	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	-10	-10	-10	-10	-10	-10
9	**h**	0	0	0	-5	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	**i**	-5	-2	-10	-10	-10	-10	-10	-10	0	5	-10	-10	-10	-10	-10	-7	-10	-10	-10	-10	-10	-2	-10	-10	-10	3	-10	-1	-1	-1	-1	5	-2
11	**j**	-5	-10	-10	-10	-10	-10	-10	2	0	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
12	**k**	-5	-10	-10	2	-10	-10	-10	-10	0	-10	-10	5	-10	-10	-10	-10	-10	3	-10	-10	-10	-10	-10	-10	3	-10	-10	-10	-10	-10	-10	-10	-10
13	**l**	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
14	**m**	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
15	**n**	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
16	**o**	-5	-2	-10	-10	-10	-1	-10	-10	0	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-10	-10	-10	-5	-10	-2
17	**p**	-5	-10	-2	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
18	**q**	-5	-10	-10	-2	-10	-10	-10	-10	0	-10	-10	3	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
19	**r**	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
20	**s**	-5	-10	-10	3	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-2	-10	-10	-10	-10	-10	-1	-10	-10	-10	-10	-10	-10
21	**t**	-5	-10	-10	-4	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-2	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
22	**u**	-5	-2	-10	-10	1	-10	-10	-10	0	3	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	3	2
23	**v**	-5	-10	-7	-10	-10	-10	2	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	2	-10	-10	-10	-10	-10	-10	-10	-10	-10
24	**w**	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	5	-10	-10	-10	-10	-10	-10	-10	-10	-10
25	**x**	-5	-10	-10	2	-10	-10	-10	1	0	-10	-10	3	-10	-10	-10	-10	-10	3	-10	3	-10	-10	-10	-10	5	-10	2	-10	-10	-10	-10	-10	-10
26	**y**	-5	-5	-10	-10	-10	-10	-10	-10	0	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-5	-10	-10	-10	5	-10	-10	-10	-10	-10	4	2
27	**z**	-5	-10	-10	-10	-10	-10	-10	-10	0	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10	-10	1	-10	5	-10	-10	-10	-10	-10	-10
28	**é**	-5	-2	-10	-10	-10	3	-10	-10	0	-1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	2	-10	5	4	4	3	3	-2
29	**è**	-5	-2	-10	-10	-10	3	-10	-10	0	-1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	-10	2	-10	4	5	4	3	3	-2

30	**ê**	-5	-2	-10	-10	-10	3	-10	-10	0	-1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	-10	2	-10	4	4	5	3	3	-2
31	**ë**	-5	-2	-10	-10	-10	4	-10	-10	0	-1	-10	-10	-10	-10	-10	-5	-10	-10	-10	-10	-10	1	-10	-10	-10	2	-10	3	3	3	5	3	-2
32	**ï**	-5	-3	-10	-10	-10	-2	-10	-10	0	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	4	-10	3	3	3	3	5	-2
33	**â**	-5	5	-10	-10	-10	-2	-10	-10	0	-2	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-4	-10	-2	-2	-2	-2	-2	5

Matrice des scores initiaux pour l'alignement des séquences à la base des symboles phonétiques

ID	lettre	_	a	ɑ	e	ɛ	ə	œ	ø	i	o	ɔ	u	y	ɑ̃	ɛ̃	ɔ̃	œ̃	b	d	f	g	k	l	m	n	p	r	s	t	v	z	ʃ	ʒ	ɲ	ŋ	h	ø	j	ɥ	w	ks	gz	dʒ	ʁ
1	_	-30	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
2	a	-10	5	4	3	3	2	2	2	1	1	1	2	1	4	3	2	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	1	-10	-10	-10	-10
3	ɑ	-10	4	5	3	3	3	2	2	1	1	1	2	1	4	3	2	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	1	-10	-10	-10	-10
4	e	-10	3	3	5	4	4	3	2	1	2	2	2	2	2	3	2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	1	-10	-10	-10	-10
5	ɛ	-10	3	3	3	5	3	3	2	1	2	2	2	1	3	4	2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	1	1	1	-10	-10	-10
6	ə	-10	2	2	3	3	5	3	2	1	1	2	2	1	1	3	2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	1	1	1	-10	-10	-10
7	œ	-10	2	2	3	2	2	5	3	1	3	2	1	0	1	2	1	4	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	3	-10	-10	3	-10	-10	-10
8	ø	-10	2	2	2	2	2	3	5	-1	2	2	1	-1	-2	-2	-2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	2	-10	-10	-10	-10
9	i	-10	1	2	1	2	2	-2	-1	5	-1	-1	1	3	-1	2	-2	1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	3	2	1	-10	-10	-10
10	o	-10	1	1	2	2	2	3	1	-1	5	3	3	-1	-2	-2	2	1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	3	-10	-10	-10
11	ɔ	-10	1	1	2	2	1	2	1	-1	3	5	2	2	1	-1	4	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	2	-10	-10	-10
12	u	-10	2	2	2	2	2	2	3	1	2	2	5	2	-1	-1	2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	1	1	2	-10	-10	-10
13	y	-10	1	1	1	1	1	0	2	2	-1	-1	2	5	-1	-1	-1	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	3	2	-10	-10	-10
14	ɑ̃	-10	4	4	3	3	1	-1	-2	-1	-1	-1	1	-1	5	3	2	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	-10
15	ɛ̃	-10	3	3	3	4	3	2	-2	-2	-2	-1	-1	-1	3	5	2	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
16	ɔ̃	-10	3	3	2	2	2	2	3	-1	3	4	2	-1	2	2	5	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	2	-10	-10	-10	-10
17	œ̃	-10	2	2	2	2	2	4	3	-1	1	1	3	2	2	3	3	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10
18	b	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10	-2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
19	d	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10
20	f	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
21	g	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10	-10	-10	-10	-10	-10	2	1	
22	k	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10
23	l	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
24	m	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
25	n	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	3	3	-10	-10	-10	-10	-10	-10	-10	-10	-10
26	p	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
27	r	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
28	s	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	3	-10	-10	-10
29	t	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-3	-10	-10	-10	-10	-10	-10	-10	-10	1	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	-10
30	v	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
31	z	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	3	-10	-10
32	ʃ	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10

33	**ʒ**	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	4
34	**ɲ**	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
35	**ŋ**	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10	-10
36	**h**	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	-10
37	**ø**	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	2	3	-10	-10	-10	-10
38	**j**	-10	-10	-10	-2	-10	-10	-10	-10	2	-10	-10	1	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	2	-10	-10	-10	-10	-10
39	**ɥ**	-10	1	1	1	1	1	1	2	2	-10	-10	-10	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	5	2	-10	-10	-10	-10
40	**w**	-10	1	1	1	1	1	2	2	-10	2	2	3	2	2	1	2	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	3	-10	3	5	-10	-10	-10	-10
41	**ks**	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	3	-10	-10	-2	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	2	2	-10
42	**gz**	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	3	-10	-10	-10	-10	-10	-10	1	-10	-10	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	2	5	2	-10
43	**dʒ**	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-3	-10	2	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5	-10	-10	-10	-10	-10	-10	-10	2	2	5	-10
44	**ʁ**	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	1	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	3	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	5