Contribution à la réalisation d'un synthétiseur de la parole pour la langue Arabe

Aissa Amrouche, Leila Falek, Hocine Teffahi

الملخّـص

بالنّظر إلى خصائصها المورفولوجية والنّحوية، تطرح اللّغة العربيّة تحديًا كبيرا أمام إمكانيّة التّحكّم فيها آليًا، وإيجاد نظم تركيب لها، لهذا تندر الجهود العلمية في هذا الميدان. والهدف من عملنا هذا هو المساهمة في إنجاز نظام لتركيب الكلام العربيّ بانتقاء ديناميّ لسلسلة من الوحدات ضمن مدوّنة عربية منتقاة، من خلال قراءة آليّة للنّصّ العربيّ، وذلك باستعمال برنامج «الماتلاب» (Matlab) وقد توصلنا إلى نتائج تُعدّ مُرضية.

الكلمات المفاتيح : توليد الكلام، مدوّنة، اللّغة العربيّة، Bi-grams، TTS.

Contribution à la réalisation d'un synthétiseur de la parole pour la langue Arabe

Aissa Amrouche[1, 2], Leila Falek[2], Hocine Teffahi[2]

[1] Centre de Recherche Scientifique et Technique pour le Développement de la Langue Arabe (CRSTDLA), Alger, Algérie.
[2] Laboratoire de Communication Parlée et de Traitement du Signal LCPTS, Université des Sciences et de la Technologie Houari Boumediene (USTHB), Alger, Algérie.
amrouche_a@yahoo.fr, lfalek@hotmail.fr, hteffahi@gmail.com

Résumé

Par ses propriétés morphologiques et syntaxiques la langue arabe est considérée comme une langue très difficile à maîtriser dans le domaine du traitement automatique de la parole et les systèmes de synthèse à partir du texte arabe sont donc très peu nombreux. Le but de notre travail est une contribution à la réalisation d'un système de synthèse de la parole par concaténation d'unités dans un corpus pour la langue arabe. Le but de notre travail est une réalisation d'un système de synthèse de la parole par concaténation d'unités dans un corpus pour la langue arabe baptisé GArabic TTS sous l'environnement Matlab. Le texte à introduire est un texte non voyellé qui facilite l'utilisation du système, la sortie est disponible uniquement pour une voix masculine. L'évaluation finale de la qualité globale du système est jugée satisfaite.

Mots clés : TTS, Synthèse de la parole, Corpus, Bi-grams, Langue Arabe.

1. Introduction

Ces dernières années, une nouvelle génération de systèmes de synthèse par concaténation est apparue. L'augmentation des performances des ordinateurs, en termes de vitesse de calcul et surtout de quantité de mémoire vive disponible, rend désormais possible l'utilisation de dictionnaires de grande taille (plus d'une heure de parole). La sélection des unités, essentiellement statique dans les systèmes classiques de synthèse par concaténation, devient alors dynamique dans cette nouvelle génération de systèmes. En effet, afin de limiter la taille du dictionnaire, les systèmes classiques utilisent une seule réalisation acoustique de l'unité, laquelle est soigneusement choisie lors du processus de fabrication du dictionnaire, tandis que les nouveaux systèmes disposent en général de plusieurs réalisations acoustiques d'une même unité.

Dans un système classique de synthèse par concaténation, les unités acoustiques sont mono-représentées et choisies lors du processus de fabrication du dictionnaire.

Les diphones sont les unités les plus utilisées et sont en général enregistrés dans un contexte neutre, des mots isolés sans signification appelés logatomes. La seule instance (ou représentation) du diphone dans le dictionnaire ne permet pas de tenir compte des phénomènes de coarticulation (l'assimilation progressive ou régressive), ce qui provoque en général une discontinuité spectrale importante au point de concaténation [1].

D'autre part, le diphone décrit très mal les phonèmes de transition et les phonèmes instables. Donc la sélection statique des unités est une approche essentiellement phonétique, qui ne tient pas assez compte des variantes acoustiques d'un même son. Pour répondre aux limitations intrinsèques des unités mono-représentées choisies de manière statique, la sélection statique des unités est remplacée par une sélection dynamique, dans un module qui précède le module de concaténation lors du processus de synthèse. Le dictionnaire est constitué de parole naturelle : des mots isolés dans les premiers systèmes, des paragraphes entiers dans les systèmes les plus récents. Les unités du dictionnaire doivent seulement assurer la plus grande couverture phonétique possible de la langue. Une méthode de sélection dynamique d'unités est proposée et appliquée pour la langue anglaise par T. Dutoit en 2005 [2]. Les résultats qu'il a obtenus sont très prometteurs et ont apporté un plus dans le domaine. Beaucoup d'autres travaux dans ce domaine concernent les langues latines, cependant, les applications pour la langue arabe ne sont pas très nombreuses.

Dans cet article, nous avons décrit un système de synthèse TTS pour la langue arabe que nous avons appelé GArabic. Nous avons commencé par décrire le corpus choisi comme base acoustique, puis l'utilisation de modèle bi-grams pour la sélection des unités. Enfin nous présentons une mise en œuvre de l'algorithme de sélection. Nous avons testé notre système de synthèse à base de l'écoute en faisant plusieurs tests pour juger la qualité de GArabic.

2. La base de données

2.1. Particularités de la langue arabe

L'arabe est une langue sémitique de la même famille que le syriaque, l'araméen et l'hébreu. Il est parlé aujourd'hui par plus de 300 millions d'habitants dans le monde et 22 pays. Par ses propriétés morphologiques et syntaxiques la langue arabe est considérée comme une langue difficile à maîtriser dans le domaine du traitement automatique des langues [3,4]. Les recherches pour le traitement automatique de l'arabe ont débuté vers les années 1970. Les premiers travaux concernaient notamment les lexiques et la morphologie. Nous allons citer quelques particularités de la langue arabe :

- L'alphabet arabe comporte 34 phonèmes: 6 voyelles et 28 consonnes (Table 1).
- L'arabe s'écrit et se lit de droite à gauche.
- L'alphabet arabe n'est constitué que de consonnes.
- Les lettres de l'alphabet arabe sont classées en deux groupes : les lettres solaires et lunaires.

Al	Sy	Tr	Al	Sy	Tr
ا	A	alif	ف	f	fā
ب	b	bā	ق	q	qāf
ت	t	tā	ك	k	kāf
ث	x	thā	ل	l	lām
ج	j	jīm	م	m	mīm
ح	H	ḥā	ن	n	nūn
خ	K	khā	ه	h	hā
د	d	dāl	و	w	wāw
ذ	D	dhal	ي	y	yā
ر	r	rā	◌َ	a	fatha
ز	z	zāy	◌ِ	i	kasra
س	s	sīn	◌ُ	u	dama
ش	c	chīn	◌ْ	-	soukoun
ص	S	ṣād	◌ٌ	U	tanwin
ض	@	ḍād	◌ً	à	tanwin
ط	T	ṭā	◌ٍ	I	tanwin
ظ	Q	ẓā	◌ّ	ublée	shadda
ع	X	ɛayn	~	:	madd
غ	G	ghayn			

Al : Alphabet, Sy : Symbole, Tr : Translittération

Table 1. L'alphabet arabe et symbole utilisé dans GArabic.

- Les voyelles sont rajoutées au-dessus et en dessous des consonnes comme des signes diacritiques.
- Les diacritiques sont : La fatha, La kasra, La dama, le soukoun, La shadda et Le madd.
- La césure des mots à la fin de la ligne est également inconnue.
- La forme d'une lettre peut dépendre de sa voyelle ou de celle de la lettre précédente.
- L'absence de règles de ponctuation universellement conventionnelles.
- La langue arabe comprend trois catégories de mots : Verbes, noms, particules que nous allons développer plus loin.

2.2. Constitution de la base de données

La base de données doit être suffisamment grande pour contenir toutes sortes de séquences phonétiques qui peuvent apparaitre dans différents contextes linguistiques, de sorte que les segments de parole qui en résultent soient disponibles dans différente formes prosodiques. Dans cette étude, pour plus de simplification, nous avons choisi 45 phrases issues du logiciel Rosetta Stone [6] (c'est un logiciel d'apprentissage des langues assisté par ordinateur (CALL) exclusivement publié par Rosetta Stone) prononcées par un locuteur masculin. Le vocabulaire total est de 410 mots. Elle est constituée de :

```
GAGArabic_corpus = {
        %sentence number 1 in database.
    % الصيِّن هيَ البَلَدُ الآسْيَوي الْمُلوَّنُ بِالأَحْمَر فِي هَذِهِ
الْخَرِيطَة
    'الصين'          'noun'            'AS_in'
    'هي'             'pronoun'         'hy'
    'البلد'          'noun'            'l_bld'
    'الاسيوي'        'adjective'       'lA_sy_w'
    'الملون'         'adjective'       'Almlwn'
    'بالأحمر'        'adjective'       'bl_AHmr'
    'في'             'preposition'     'fi'
    'هذه'            'demonstrative'   'hDh'
    'الخريطة'        'noun'            'l_K_r_T'
    '.'              'punctuation'     '_'
    %sentence number 2 in database.
    %     رُوسْيَا مُلَوَّنَة بِالأَحْمَر فِي هَذِهِ الْخَرِيطَة.
    %     رُوسْيَافِي أُورُوبَّا وَآسْيَا.
    'روسيا'          'noun'            'ru_sy'
    'ملونة'          'adjective'       'ml_wn'
    'بالأحمر'        'adjective'       'bl_AHmr'
    'في'             'preposition'     'fi'
    'هذه'            'demonstrative'   'hDh'
    'الخريطة'        'adjective'       'l_K_r_T'
    ','             'punctuation'     '_'
    'روسيا'          'noun'            'ru_sy'
    'في'             'preposition'     'fi'
    'اوروبا'         'noun'            'o_rB_a'
    'و'              'coordinator'     'W'
    'اسيا'           'noun'            'A_sy'
    '.'            'punctuation'     '_'
```

Figure 1 : Exemple de fichier (.m) extrait du corpus "GArabic corpus file"

- Un dossier constitué de toutes les phrases du corpus avec l'extension (.wav). Chaque phrase est recensée par un numéro tel que : 1.wav, 2.wav,…., 45.wav,
- Un fichier (.m) qui contient la transcription orthographique, parties du discours, et la transcription phonétique de tous les mots prononcés par le locuteur en langue arabe " GArabic_corpus " Figure 1.

3. Mise en œuvre de la méthode

Les différentes étapes de mise en œuvre de la méthode sont :

3.1. Création du lexique

La création du lexique est faite à partir du corpus **GArabic** à l'aide de la fonction **«Garabic_load_corpus.m»**. Ce script Matlab contient toutes les phrases de corpus ainsi que la partie du discours et la transcription phonétique de chaque mot. Les trois premiers mots seront représentés comme suit :

```
>>corpus=GArabic_corpus(1:3,:)
corpus =
'الصين'   'noun'        'AS_in'
'هي'      'pronoun'     'hy'
'البلد'   'adjective'   'l_bld'
```

3.2. Prétraitement du corpus: pre-proceesing

Notre corpus bien que volumineux, se distingue par sa simplicité : il ne contient ni nombres, ni acronymes ni noms propres complexes. Pair ailleurs, les phrases à synthétiser ne doivent contenir aucune faute d'orthographe. Ainsi, la seule tâche qui nous reste à faire est de décomposer le texte en entrée, en ensemble d'états (des mots et des ponctuations). L'implémentation sous Matlab de cette stratégie est rendue facile grâce à la fonction «strtok». Le résultat de cette décomposition est illustré par cet exemple :

```
>>  Phrase='الصين هي البلد الاسيوي الملون بالأحمر في هذه الخريطة'
        Phrase =
          الصين هي البلد الاسيو الملون بالأحمر في هذه الخريطة'

>> sentence=decompose(phrase)
   sentence=

'الصين'
'هي'
'البلد'
'الاسيوي'
'الملون'
'بالأحمر'
'في'
'هذه'
'الخريطة'
```

3.3. Analyse phonologique du corpus GArabic

Nous avons présenté précédemment les avantages d'utilisation du corpus GArabic comme base acoustique et nous avons montré que les points forts de cette base résidaient dans la composition hiérarchique de ses fichiers. N'oublions pas, bien sûr, les possibilités offertes par les fichiers des transcriptions phonétiques. De ce fait, l'analyse phonologique de la base GArabic consiste à présenter une liste des mots du corpus associés à leurs parties de discours. Ainsi, nous parcourons tous les répertoires et les fichiers «.wav» et «.seg» de notre corpus pour créer cette liste.

Ceci est réalisé par script Matlab, à l'aide de la fonction «corpus_to_list.m», qui va créer une matrice «word_list» présentant les mots de base et les transcriptions phonétiques correspondantes:

```
      >>[word_list
P_of_S]=corpus_to_directory(GArabic_corpus);
   word_list =
        ''                  {1x1 cell}
        ','                 {1x1 cell}
        '.'                 {1x1 cell}
      'اسبانيا'             {1x1 cell}
       'اسيا'               {1x2 cell}
     'الاسيوي'              {1x1 cell}
    'الاوروبي'              {1x1 cell}
       'البلد'              {1x2 cell}
     'الخريطة'              {1x2 cell}
      'الصين'               {1x1 cell}
      'الملون'              {1x1 cell}
       'اوروبا'             {1x2 cell}
```

Par exemple, sur la table 3, nous avons obtenu pour le mot «**البلد**» deux parties de discours telles que :

```
>>  w=word_list{7,2}
w    w =
     'adjective'   'noun'
```

La même fonction« **corpus_to_list.m** », nous permet d'avoir aussi toutes les parties du discours :

```
>> P_of_S
P_of_S =
    'adjective'
    'adverb'
    'coordinator'
    'demonstrative'
    'noun'
   'verbe'
    'preposition'
    'pronoun'
    'propername'
    'punctuation'
    'subordinator'
```

Nous avons par la suite créé une fonction Matlab, «directory_search.m», qui nous permet de trouver les parties du discours pour un mot donné tel que :

```
>>possible_word=directory_search('البلد',word_list)
 possible_word =
      'adjective'    'noun'
```

Cette fonction est utilisée par le script Matlab « tts_morph_using_directory », pour trouver toutes les parties du discours possibles correspondantes aux mots constituants la phrase à synthétiser.

```
>>possible_tags=tts_morph_using_directory({'هي';'البلد'},
   word_list)
   possible_tags =
   {1x1 cell}    {1x2 cell}
    >>possible_tags{:,:}
   ans =
     'pronoun'
   ans =
      'adjective'    'noun'
```

3.4. Présélection des unités

A ce stade, il est possible d'associer à chaque mot la partie du discours qui lui correspond. Ceci implique que nous devons faire une sélection sur l'ensemble des parties du discours proposées par les modules précédents. La stratégie standard était d'utiliser le modèle linguistique des n-grams [2, 7] que nous allons rappeler dans le paragraphe suivant. Le modèle n-grams est apparu dans le contexte de la reconnaissance de la parole pour estimer une séquence de mots $W_1, W_2 \dots, W_N$ dans une langue donnée.

Dans le contexte de présélection pour une phrase donnée $W = (w_1, w_2, \dots, w_N)$, nous cherchons la meilleure séquence d'étiquette$\hat{T}$ sur toutes les séquences possibles $T = (t_1, t_{,2} \dots, t_N)$, effectuées sur l'ensemble des étiquettes de partie du discours $\{c_1, c_2, \dots, c_M\}$:

$$\hat{T} = arg_T max P(T|W) \quad (1)$$

Selon le théorème de Bayes, ceci est équivalent à retrouver :

$$\hat{T} = arg_T \, max \frac{P(T,W)}{P(W)} = arg_T max \frac{(W|T)P(T)}{P(W)} \quad (2)$$

- Le dénominateur de l'équation (2) est indépendant de T, nous pouvons donc le négliger ;
- Le modèle n-grams pour la présélection assume les approximations suivantes :
  - La probabilité d'un mot, connaissant celle qui lui précède, dépend de l'étiquette ;
  - La probabilité d'une étiquette, connaissant celle qui lui précède, dépend des n-1 étiquettes précédentes. Nous obtenons alors le résultat suivant :

$$P(W|T) = P(w_1, w_2, \dots, w_N|t_1, t_2, \dots, t_N)$$
$$= P(w_1|t_1, t_2, \dots, t_N)P(w_2|w_1, t_1, t_2, \dots, t_N)$$
$$\dots P(w_N|w_1, \dots w_{N-1}t_1, t_2, \dots, t_N) \quad (3)$$
$$\approx \prod_{1}^{N} P(w_i|t_i) \quad (4)$$

$$P(T) = P(t_1, t_2, \dots, t_N)P(t_1)P(t_2|t_1)$$
$$\dots P(t_N|t_1, t_2, \dots, t_{N-1}) \approx$$
$$\prod_{1}^{N} P(t_i|t_{i-1}, t_{i-2}, \dots, t_{i-N+1}) \quad (5)$$

Il est clair, que nous pouvons modéliser le problème par un automate à états finis. Cet automate représente un modèle bi-grams où n=1 [2, 7].

Le modèle bi-grams considéré est représenté par des états associés aux parties du discours possibles (un état par partie du discours). A chaque transition nous associons une probabilité $p\ (c_i \mid c_j)$ qui représente la probabilité qu'un mot avec la catégorie «c_j» sera suivi par un mot avec la catégorie «c_i». Les probabilités d'émission $p(w_i \mid c_j)$ représentent la probabilité que la catégorie «c_j» correspond au mot «w_i». Un exemple du modèle bi-grams est donné par la figure 2.

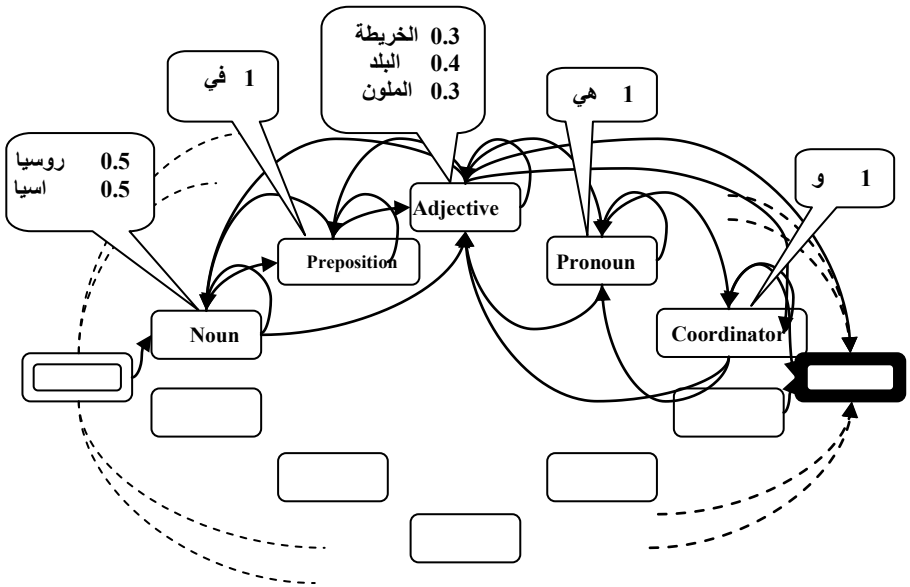

Figure 2: le modèle bi-grams.

Le calcul de la probabilité d'émission est simple avec Matlab. En effet, cette probabilité représente le nombre de fois que le mot *«w_i»* apparaît avec comme *«c_i»* divisé par le nombre total des mots avec les parties du discours *«c_i»* :

$$P(w_i|c_j) = \frac{\#(w_i|c_j)}{\#(c_j)} \qquad (6)$$

De même, les probabilités transitionnelles entre deux catégories *«cj»* et *«ci»* représentent le nombre de fois que la catégorie *«ci»* est précédée par *«cj»* divisé par le nombre total des mots avec sa partie du discours *«cj»* :

$$P(c_i|c_j) = \frac{\#(c_i|c_j)}{\#(c_j)} \qquad (7)$$

Le calcul de ces probabilités a été assuré par la fonction Matlab « copus_ to_bigrams.m », qui retourne les valeurs de probabilités d'émission et de transition:

```
>>[emission_probs, transition_probs]= corpus
_to_bigrams(GArabic_corpus);
>>emission_probs(:,3)        >>transition_probs(:,5)
   0                            0.5102
   0                              0
   0                              0
   0                              0
   0                            0.4082
   0                              0
   0                              0
   1                            0.0816
   0                              0
   0                              0
```

Par exemple la troisième colonne de "**emission_probs**" à une valeur non nulle, ce qui expliquent le fait que la troisième catégorie de partie du discours **P_of_S** (coordinator) peut apparaitre comme coordinateur "و" avec sa probabilité d'émission égal à 1.

Pareillement, la cinquième colonne de "**transition_probs**" a trois valeurs non nulles. Ce qui explique le fait que la cinquième catégorie de partie du discours **P_of_S** (noun) est suivie par un adjectif, un nom et un nom propre dans le corpus d'apprentissage, avec les probabilités 0.5102, 0.4082, 0.0816 respectivement. Nous ne sommes pas sûrs de couvrir toutes les probabilités du corpus. Ainsi nous sommes face à un problème de lissage. Nous adoptons pour cela, la stratégie adoptée par T. Dutoit dans [2], qui consiste à remplacer les valeurs nulles par des 1e-8.

(7)

3.5. Sélection des unités

Une fois les probabilités calculées, il reste à trouver la meilleure séquence ayant la plus haute probabilité. Par analogie avec la procédure d'optimisation des coûts [8, 9, 10, 11, 12, 13, 14, 15, 16], nous pouvons estimer le coût cible par l'inverse de la probabilité d'émission et le coût de concaténation par l'inverse de la probabilité transitionnelle. Ainsi, trouver la meilleure séquence minimisant le coût revient à trouver la séquence maximisant la somme des probabilités. Ceci correspond à trouver le meilleur chemin dans le réseau des états.

Un schéma modélisant le problème est présenté par la figure 3.

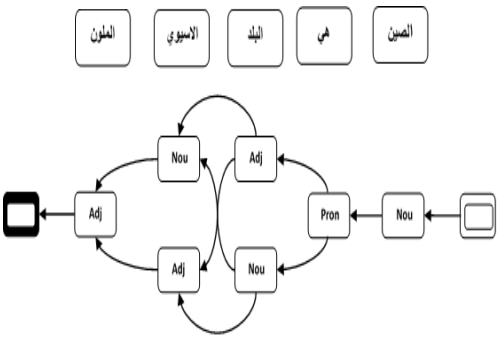

Figure 3: Sélection des unités

Pour obtenir la meilleure séquence, nous avons utilisé les deux fonctions «lattice_get_paths.m» et **«tts_tag_using_bigrams.m»** que nous avons appliqué sur le modèle bi-grams.

Un exemple d'utilisation est présenté comme suit :

```
sentence={'الملون';'الاسيوي';' البلد';' هي';' الصين'} ;
possible_tags=tts_morph_using_directory( sentence, word_list);
tags=tts_tag_using_bigrams(emission_probs,transition_probs,word_list,P_of_S,sentence,possible_tags);
tags =
'noun'
'pronoun'
'noun'
'adjective'
'adjective'
```

3.6. Synthèse de la parole

Une fois les unités sont sélectionnées depuis la base GArabic, il nous reste à reconstruire le signal par concaténation. L'algorithme utilisé ici est basé sur la méthode TD-PSOLA [17].

4. Résultats et discussions

Le résultat de cette étude est présenté sous forme d'une interface graphique interactive, qui permet à l'utilisateur une utilisation simple de notre système de synthèse (figure4).

Figure 4 : Interface graphique de SAT.

Par ailleurs, pour tester l'évaluation de la qualité globale du système développé GArabic, nous avons appliqué deux tests pour l'intelligibilité et les aspects naturels de la voix de synthèse obtenue.

Ces tests consistent à sélectionner un groupe de personnes au hasard pour leur permettre d'essayer le programme en passant par un questionnaire qui sera utilisé pour évaluer le rendement du projet.

Ce questionnaire a été conçu de manière aléatoire pour évaluer l'intelligibilité (clarté), le naturel, la qualité du son et la prononciation au niveau de phrases et de mots.

Concernant les auditeurs, ils sont au nombre de 50, de différentes professions

et de connaissance de la langue arabe afin d'obtenir une bonne évaluation.

- Dans le premier test nous avons demandé aux auditeurs de cocher les mots entendus d'une manière aléatoire Test 1 ;
- Au second, nous avons divisé le test en deux parties :
  - la première partie consiste à écouter phrase par phrase et cocher sur le questionnaire la phrase entendue Test 2A ;
  - la deuxième partie est la plus difficile. Pour cela, nous avons pris plusieurs groupes de phrases. Dans chaque groupe de phrases, nous avons répété les mêmes mots constituants la phrase avec un petit changement au niveau de la phrase Test 2B.

À la fin des tests nous avons demandé aux auditeurs de juger selon l'écoute du programme en donnant une mesure de la qualité comme suit : (5 - Excellent, 4 - Très bon, 3-bon, 2- moyen, 1 -mauvais).

En analysant les résultats de ces tests, on a laissé entendre que, quand il s'agit de l'intelligibilité du système, le système **GArabic TTS** est réussi (Donne de bons résultats).

Les résultats de ces tests ont montré que les participants peuvent entendre ce qui se dit et reconnaître les changements à la parole synthétisée. La majorité des mots et les phrases ont été correctement reconnue et perçue par la majorité des auditeurs et l'évaluation de la qualité globale du système est satisfaisante.
La figure 5 montre le signal original et le signal synthétique de la phrase :

« الصين هي البلد الاسيوي الملون بالأحمر في هذه الخريطة »

5. Conclusion

Dans cet article, nous avons développé un système SAT baptisé « GArabic TTS », pour la langue arabe basé sur la méthode de synthèse par sélection d'unité dans un corpus. La conception d'une base de données « corpus » et les différentes étapes nécessaires à la mise en œuvre de la méthode développée ont été décrites.
La première étape est réservée à la création d'un lexique contenant les mots et les transcriptions phonétiques existant dans le corpus. Ensuite, un modèle bi-grams a été créé en se basant sur le lexique dégagé. Lors de la sélection, les coûts cibles et les coûts de concaténation ont été remplacés par les valeurs des probabilités d'émission et de transition. En effet, minimiser le coût revient à maximiser la somme des probabilités. Comme dernière étape, nous avons concaténé les diphones extraits à partir de la base de données actuelle et les résultats obtenus sont satisfaisants.
L'intelligibilité globale du système GArabic TTS, a été jugée de qualité, naturelle et acceptable par les auditeurs.
Des perspectives et des améliorations restent à envisager :

- Au niveau de corpus il faut élargir le corpus pour l'obtention d'une meilleure qualité de la voix.
- Tester et réaliser un autre corpus pour un texte arabe diacritique.

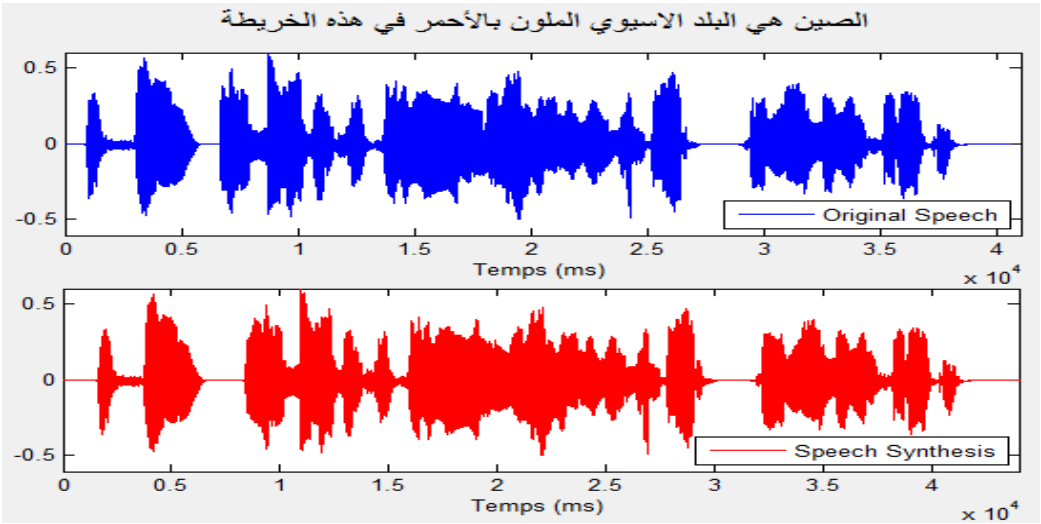

Figure 5: Signal original et signal synthétisé pour la phrase " الصين هي البلد الاسيوي الملون بالأحمر في هذه الخريطة".

6. Références

[1] Boite, R., Bourlard, H., Dutoit, T., Hancq J. and Leich, H., "Traitement de la parole, chapitre : Synthèse de la parole à partir d'un texte", Collection Electricité, Presses polytechniques et universitaires romandes, 345-441, 2000.

[2] Dutoit, T. and Cernăk, M.,"TTSBOX: A Matlab toolbox for teaching Text-To-Speech Synthesis", IEEEICASSP, 2005.

[3] Aljlayl, M. and Frieder, O., "On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach", In 11th International Conference on Information and Knowledge Management (CIKM), Virginia (USA), 340-347, 2002.

[4] Larkey, L. S., Ballesteros, L. and Connell M., "Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis", In Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR), Tampere, Finland, 275-282, 2002.

[5] Boula, P., "Etude linguistique appliquée à la synthèse de la parole à partir du texte", thèse de doctorat, Université de Paris XI, Orsay, 1997.

[6] http://www.rosettastone.fr

[7] Chen, S.F. and Goodman, J.,"An empirical study of smoothing techniques for language modeling", Center for Research in Computing Technology, Harvard University, Cambridge, Massachusetts, 1998.

[8] Toda, T., Kawai, H., Tsuzaki, M. and Shikano, K., "Unit Selection for Japanese Speech Synthesis Based on Both Phoneme Unit and Diphone Unit", In Proc. of ICASSP, 1: 465–468, 2002.

[9] Breen, A. and Jackson, P.,"Non-Uniform Unit Selection and the Similarity Metric within BT's LAUREATE TTS System", 3rd ESCA Int. Workshop, November 1998.

[10] Prudon, R. and Alessandro, C., "A Selection/Concatenation Test-to-Speech System: Databases Development, System Design, Comparative Evaluation", 4th ISCA Tutorial and Research Workshop on Speech Synthesis, 2001.
[11] Yi, G.R.W. and Glass, J., "Information-Theoretic Criteria for Unit Selection Synthesis", In Proc. of ICSLP, 2617–2620, 2002.
[12] Lee, M., Lopresti, D.P. and Olive, J.P., "A Text-to- Speech Platform for Variable Length Optimal Unit Searching Using Perceptual Cost Functions", 4th ISCA Tutorial and Research Workshop on Speech Synthesis, September 2001.
[13] Peng, H., Zhao, Y. and Chu, M., "Perceptually Optimizing the Cost Function for Unit Selection in TTS System With one Single Run of MOS Evaluation", In Proc. of ICSLP, 2613–2616, September 2002.
[14] Donovan, R.E. and Eide, E.M., "The IBM Trainable Speech Synthesis System", In Proc. of ICSLP, 1998.
[15] Nomura, T., Mizuno, H. and Sato, H., "Speech Synthesis by Optimum Concatenation of Phoneme Segments", 1st ESCA-IEEE Tutorial and Research Workshop on Speech Synthesis, 39–42, 1990.
[16] Pantazis, Y., Stylianou, Y. and Klabbers, E., "Discontinuity Detection in Concatenated Speech Synthesis Based on Nonlinear Speech Analysis", In Proc. of Inter speech, 2005.
[17] Charpentier, F.J. and Stella, M.G. "Diphones synthesis using an overlap-add technique for speech waveforms concatenation", In Proceedings of ICASSP, 1986.