République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Larbi Tébessi –Tébessa–
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département de mathématiques et Informatique

MEMOIRE DE MASTER
Domaine : Maths et Informatique
Filière : Informatique
Option : Système et multimédia

Thème :

Modèle d'indexation automatique pour la recherche sur le contenu des documents multimédias

Présenté par :
Bouguessa Brahim & Djedouani Nafaa

Sous la direction de M. Bendib Issam

Devant le jury :

Aouine Mohamed	MA classe A	Université de Tébessa	Président
Gattal Abdeldjalil	MA classe A	Université de Tébessa	Examinateur

Date de soutenance : 30/05/2016

Note : ……. **Mention :** ……..……

Remerciement

En premier lieu, nous souhaitons remercier chaleureusement notre encadreur, Bendib Issam. Nous tenons à le remercier pour sa disponibilité, ses conseils judicieux, sa réactivité et son aisance à faciliter les différentes étapes de cette thèse.

Nous sommes tout particulièrement reconnaissant aux membres de jury qui ont accepter de juger notre travail

Et Pour terminer, un grand merci tout particulier à toutes personnes qui à contribuer dans la réalisation de ce mémoire.

ملخص

إن استخدام التقنية في عالمنا المعاصر اليوم سمح بظهور وتطور عدة وسائل وتقنيات رقمية من برامج ووسائل اتصالات وإذاعات ومكتبات رقمية، مما أدى إلى توفر كميات هائلة من المعلومات والمعارف، مخزنة في الحواسيب ووسائط التخزين المتعددة. إن استغلال هذه الموارد الرقمية وخاصة الملفات الصوتية تحتاج إلى توفير مناهج وتقنيات للبحث، تتلاءم مع طبيعة وحجم هذه المعطيات. بالمقابل إن استغلال هذه الموارد الرقمية يحتاج إلى عمليات فهرسة وترتيب وفق محتوياتها. هذه الأخيرة، تزداد تعقيدا كلما تنوعت مجالات الموارد الرقمية وطبيعتها. في هذا الإطار، سنناقش في هذه المذكرة تصور طريقة أوتوماتيكية لفهرسة محتويات الموارد الالكترونية وخاصة منها الصوتية. سنتطرق في الجزء الأول إلى المبادئ الأساسية لأنظمة البحث النصية والرقمية وفي الجزء الثاني نعرض مختلف التقنيات الخاصة بمعالجة الملفات الصوتية بالإضافة إلى دراسة بيبليوغرافية على الأبحاث المنجزة في هذا المجال. بينما نقدم في الجزء الأخير النظام المقترح لفهرسة الموارد الرقمية مع البرمجيات الخاصة بها مع عرض وجيز لأهداف هذا المشروع مع التصورات المستقبلية.

ABSTRACT

The use of the technologies in our modern life day allowed the emergence and development of several techniques and tools as broadcasts, digital libraries and communications systems, which led to provide enormous amounts of information and knowledge stored in computers and multiple storage devices.
However, to use those digital resources and especially audio files, we need to provide approaches and research techniques adapted at the complicated structure and the size of these data. In other hand, to exploit those digital resources, we need to build indexes and techniques of search in their cotenants. The latter are becoming increasingly complex as digital resources type and varied nature. In this context, we will discuss in this work a proposed approach for automatic content indexing of multimedia resources including speech.
We present in the first part the fundamental techniques, principals of textual research systems and digital search systems. In the second part, we describe various techniques for speech recognition process of spoken documents. In addition, we present a state of art of the studies and researches performed in this field. Finally, we describe in the last part our proposed system of automatic contents of multimedia resources with their simulation and we conclude the future prospects.

Keywords : Speech recognition; documents multimedia; indexing; search contents

RESUME

L'utilisation de la technologie dans la vie quotidienne, a permis l'émergence et le développement de plusieurs médias numériques et des programmes de technologie et des moyens de communication et de stations radio et les bibliothèques numérique, qui ont conduit à la disponibilité d'énormes quantités d'informations et de connaissances stockées dans plusieurs ordinateurs et supports de stockage. L'exploitation de ces ressources numériques et plus particulièrement des fichiers audio nécessite des méthodes et techniques qui correspond à la nature et la taille de ces données. D'autre part, l'exploitation de ces ressources nécessite l'indexation numérique qui fait correspondre des termes (index) avec le contenu. Celle-ci devient plus complexe vu la variété des domaines et de la nature des ressources numériques. Dans ce contexte, nous allons discuter dans ce mémoire de proposer une approche d'indexation automatique des contenus des ressources multimédias, en particulier les documents multimédias audio. Nous allons expliquer dans la première partie des principes de base des systèmes de recherche d'information, texte et vocale. Dans la deuxième partie, nous présentons diverses techniques spéciales pour traiter les fichiers audio en plus d'un état de l'art sur les recherches menées dans ce domaine. Alors que nous proposons dans la dernière partie l'architecture de système proposé pour l'indexation des ressources audio avec une brève présentation des objectifs de ce travail et les perspectives futurs.

Dédicace

Je dédie ce modeste travail à

Mon adorable petite famille

Ma femme **Abir** *Mon fils* **Firas**

Ma chère grande famille :

Ma mère source de tendresse…

Mon père source de courage…

Mes propres frères

Bilel *et sa famille et* **Taki**

et mes amis

Brahim Bouguessa

Dédicace,

A la mémoire de mon père

puisse dieu les accueillir dans son infini e miséricorde

A la plus belle perle du monde ...ma tendre mère

A mes frère

je leur souhaitant tout le succès ... tout le bonheur

A toute ma famille pour l'amour et le respect qu'ils m'ont toujours accordé

A mon binôme pour le frère agréable qu'il était et qu'il restera pour moi

A tout mes amis

pour une sincérité si merveilleuse ... jamais oubliable, en leur souhaitant tout le succès ... tout le bonheur

A toute personne

qui m'a aide a franchir un horizon dans ma vie.

A mes collègue de travail

Aimablement

DJEDOUANI Nafaa

Je dédie ce modeste travail

Sommaire

Liste des figures

Liste des tableaux

Introduction générale

L'humanité a connu le premier enregistrement sonore depuis un siècle et demi, exactement en 1860 par le typographe français, Edouard-Léon Scott[1], la masse de documents audio a connu depuis cette année une énorme croissance à travers le monde.

Ces documents audio sont de plus en plus répandus au travers les différents médias : radio, télévision et internet. Le web a d'ailleurs provoqué ces dernières années un véritable raz de marée au niveau de la production de contenus multimédias avec des podcasts ou avec des sites comme YouTube[2] ou Dailymotion[3]. Le contenu de ces documents peut-être de la parole, de la musique ou d'autres sons divers, en général ils contiennent de l'information qui est d'une importance cruciale pour les individus et les sociétés. Les gens comptent sur diverses informations pour prendre des décisions, favoriser de nouvelles idées, et même organiser des activités sociales…etc. L'information désirée est généralement souillée par du bruit et confuse avec des détails qui sont insignifiants. La récupération fiable et efficace des informations demandées à partir de diverses sources est devenue un sujet de recherche important.

La recherche d'information (IR) est le domaine par excellence qui répond à ce besoin, au moins pour les documents texte, il a atteint un niveau de progression vertigineux au cours des dernières années. Néanmoins, la récupération des informations provenant d'autres médias reste un problème difficile comme les documents parlés.

Les productions numériques des contenus multimédias sont la plupart du temps enrichies de métadonnées décrivant le contenu pour en permettre une classification sommaire et un accès aux contenus. Celles-ci, définies manuellement, peuvent être constituées d'un titre, de mots-clés, un nom d'auteur, un résumé…etc. Néanmoins ces informations, s'avèrent souvent insuffisantes pour une classification efficace et retrouver les documents a posteriori. En outre, le nombre de documents nouveaux ou archivés et non annotés, associé au temps nécessaire de traitement de l'annotation rend l'indexation manuelle fastidieuse. Aujourd'hui, près de 600 radios et télévisions couvrant 40 langues différentes sont indexées à travers le monde de façon manuelle ou semi manuelle. Les annotations

[1] http://www.anecdote-du-jour.com/le-premier-enregistrement-de-voix-au-monde-date-de-1860/
[2] https://www.youtube.com
[3] https://www.dailymotion.com

manuelles de ces contenus, que ce soit les transcriptions ou les traductions, prennent au minimum cinq fois la durée des contenus considérés. C'est pour faciliter et accélérer ces opérations que de nombreux travaux sont menés dans le domaine de l'indexation automatique des documents parlés.

La nécessité de trouver des solutions d'indexation sur le contenu des ressources multimédias est fortement sollicitée et surtout lors de la manipulation des documents multimédia de grandes tailles. Dans ce contexte, on vise à proposer une démarche qui fournisse des index extraits automatiquement du contenu des documents multimédias pour améliorer la pertinence des résultats de recherches sur ce contenu.

L'ensemble de chapitres composant ce mémoire sont organisés en trois grandes parties : La première partie consiste à présenter des notions générales sur le domaine de la recherche d'information : principe, techniques et outils, organisés dans trois chapitres, le premier est une présentation générale de la RI, les concepts de base de RI, les processus reliés au domaine de la RI et en fin les différents modèles de recherche d'information. Le deuxième chapitre présente les systèmes de recherche d'information, leurs composantes : l'indexation et ces phases, l'interrogation et la recherche. Dans le troisième chapitre en présente une sorte de comparaison entre les documents texte et les documents parlés, avec une présentation des travaux reliés au traitement de la parole.

Dans la deuxième partie, nous présentons un état de l'art des travaux existants, organisé en trois chapitres. Le chapitre n° 1 présente les principales stratégies de recherche dans les contenus parlés. Dans le chapitre n° 2, nous présentons les fondements théoriques sur les SRAP, la modélisation acoustique, la modélisation du langage, les techniques de décodage et en fin l'évaluation des systèmes de reconnaissance de la parole. Le chapitre n° 3 présente les approches générales des STDs, les recherches actuelles menais sur les STDs, et en fin nous terminons avec un récapitulatif, et une synthèse à propos des travaux de recherche existants dans la littérature.

Dans la partie 3, nous présentons notre modèle d'indexation proposé comme solution à la problématique proposée dans le chapitre 1. Dans le chapitre 2, nous présentons la plateforme utilisée au cours de la réalisation de notre mémoire, en en fin nous terminons dans le chapitre 3 par une simulation et une validation de l'approche proposée.

Partie 1

-

La recherche d'information : principes, techniques et outils

Chapitre 1 – La recherche d'information

Introduction

La Recherche d'Information (RI) est un domaine qui s'intéresse à la structuration, l'analyse, le stockage, et à la recherche de l'information. Le défi est de trouver les documents pertinents par rapport au besoin de l'utilisateur dans un volume très important de documents disponibles.

La réalisation de tel défi se fait par des outils informatiques appelés Systèmes de Recherche d'Information (SRI) qui ont pour but de mettre en correspondance une représentation du besoin de l'utilisateur – qu'on peut appeler requête - avec une représentation du contenu des documents – qu'on peut appeler résultat de requête – à l'aide d'une fonction de correspondance.

Dans ce chapitre en a essayé de présenter en premier lieu le principe et l'idée générale du domaine de recherche d'information, les travaux reliés à ce domaine ainsi que les modèles existants.

Recherche d'information

Le terme « recherche d'information » a été utilisé pour décrire un vaste domaine de recherche qui est « concerné par la représentation, le stockage, l'organisation, et l'accès à des éléments d'information ». [1].

Le scénario typique associé à la recherche d'information consiste à localiser un ensemble d'éléments d'information ou de « documents » au sein d'une grande collection ou source qui correspondent le mieux à une « requête » d'un utilisateur ; cette requête est généralement une spécification d'information incomplète de ces besoins. L'utilisateur ne cherche pas un fait précis mais il est intéressé par un sujet général et veut en savoir plus à ce sujet, avec des résultats pertinents à sa demande tout en comprenant la mesure dans laquelle les documents mentionnés correspondent à la demande.

Concepts de base de RI

La recherche d'information selon une synthèse des travaux de [2] et [3] s'articule sur les concepts de base suivants :

Collection de documents : la collection de documents (fonds documentaire) constitue l'ensemble de documents exploitables et accessibles.

Document : c'est l'information élémentaire d'une collection de documents. L'information élémentaire, peut représenter tout ou une partie d'un document.

Besoin d'information : la notion de besoin en information est souvent représentée par le besoin de l'utilisateur. Et selon [4] il y a trois types de besoin d'information :

- **Besoin vérificatif :** l'usager souhaite vérifier une information ou retrouver des éléments d'informations aux caractéristiques connues (le nom de l'auteur d'un article, la revue dans laquelle il a été publié, la date de publication). Il veut, par exemple, retrouver des références bibliographiques précises, un article déjà lu ou cité par un autre auteur. Il sait que l'information existe, et parfois où il va la retrouver. La précision des recherches est alors déterminante.
- **Besoin conscient concernant un sujet (dirigé) :** l'utilisateur veut clarifier, passer en revue ou approfondir certains aspects d'un sujet connu. Il possède des données relatives au sujet, comme des termes, des concepts, des représentations imagées, etc.

- **Besoin flou sur un sujet :** l'utilisateur veut explorer de nouveaux concepts ou relations en dehors des domaines qu'il connaît, ou les données qu'il connaît sont vagues et incomplètes. Les SRI classiques ne sont pas du tout adaptés à ce type de besoin : l'usager ne dispose souvent pas, par exemple, du vocabulaire adéquat pour formuler sa demande. Souvent, il ne connaît pas non plus les sources qui pourraient l'aider.

Requête : la requête constitue l'expression du besoin en information de l'utilisateur. Elle représente l'interface entre le SRI et l'utilisateur. Divers types de langages d'interrogation sont proposés dans la littérature. Une requête est un ensemble de mots clés, mais elle peut être exprimée en langage naturel, booléen ou graphique.

Modèle de représentation : c'est un processus qui permet l'extraction d'une représentation paramétrée qui couvre au mieux son contenu sémantique depuis un document. Ce processus est appelé l'indexation, qui a comme résultat un descripteur du document, constitué d'une liste de termes auxquels sont associés généralement des poids pour différencier leurs degrés de représentativité du contenu sémantique de l'unité textuelle correspondante. L'ensemble de termes reconnu par le SRI est rangé dans un dictionnaire constituant le langage d'indexation.

Modèle de recherche : c'est le cœur du SRI, lié au modèle de représentation. Il comprend la fonction de mise en correspondance entre une requête et l'ensemble de documents pertinents à restituer.

Processus reliés à la recherche d'information

Il y a plusieurs processus de traitement de l'information qui sont étroitement liés à la recherche d'information. Ces processus sont listés dans le tableau 1 [5], ainsi qu'une brève description de leurs objectifs et des caractéristiques relatives à la recherche et récupération de documents. Dans cette section, nous décrivons brièvement ces processus et mentionnons leurs similitudes et leurs différences pour la récupération de documents.

Processus de traitement de l'information	Buts	Documents	Requêtes	Données d'apprentissage
Récupération de l'information	Rechercher et récupérer des documents dans une collection qui se rapportent à une requête de l'utilisateur	Statique, non structurées	Dynamique, incomplète	-
Récupération de base de données	Rechercher une base de données et retour des faits particulier ou des enregistrements spécifiques qui répondent à une requête de l'utilisateur	Statique, structurées	Dynamique, complète	-
Filtrage de l'information	Identifier les documents pertinents depuis des flux entrants de documents	Dynamique, non structurées	Statique, incomplète	+
Catégorisation de documents	Classer les documents dans un ou plusieurs catégories prédéfinies	Non structurées	Pas de requêtes, catégories connues	+
Regroupement de documents	Découvrir automatiquement la structure dans une collection de documents non étiquetés	Statique, non structurées	Pas de requêtes, catégories inconnues	-
Extraction d'informations	Automatiquement trouver et extraire domaine, des caractéristiques ou des faits précis	Non structurées	Pas de requêtes	+
Résumé de documents	Dériver automatiquement une représentation de sens concise du document	Non structurées	Pas de requêtes	-

Tableau 1: Liste des processus de traitement d'information liés à la recherche d'information.

Récupération de l'information

Le but de ce processus est de rechercher et de récupérer des documents pertinents à la demande d'un utilisateur depuis une collection (donnée ou disponible). Cette tâche est caractérisée par un besoin dynamique de l'information dans le sens que les demandes des utilisateurs peuvent changer d'une session à l'autre, et on ne sait pas a priori ce que l'utilisateur demandera. La demande est également en général une spécification incomplète (imprécise) des besoins en informations de l'utilisateur. Une autre caractéristique est que la collection de documents est relativement statique. Il peut y avoir des ajouts et/ou suppressions de la collection, mais ces opérations ont généralement un effet très faible sur toute la collection. En outre, aucune donnée d'apprentissage supervisé n'est disponible lors de la création de collections [5], car en ne sait pas ce que l'utilisateur demandera et, par conséquent, quels sont les documents pertinents à sa demande et qui ne sont pas.

Récupération de base de données

Dans la récupération de base de données, le but est de rechercher une base de données et de retourner des faits précis ou des enregistrements particuliers qui répondent ou correspondent exactement à une demande d'utilisateur donnée. La structure des enregistrements est généralement bien définie (par exemple, les enregistrements sont constitués de champs spécifiques remplis de types particuliers de valeurs telles que les codes postaux, les dates, etc.) et la demande est une spécification complète (précise) des besoins en informations de l'utilisateur.

Filtrage / routage d'information

Dans le filtrage / routage de l'information, l'objectif est d'identifier les documents pertinents depuis un flux entrant de documents. Le filtrage est généralement basé sur la description des préférences à long terme de l'information appelé « profile » au lieu des requêtes dynamiques. Les documents qui correspondent sont ensuite transférés ou routés vers les utilisateurs associés au profil ; ceux qui ne correspondent pas sont rejetés. Puisque le besoin en information est relativement statique, les documents qui ont été traités et évalués par l'utilisateur peuvent servir de données d'apprentissage pour améliorer le profil [5].

Catégorisation des documents

Le but de catégorisation de documents est de classer les documents dans un ou plusieurs ensembles de catégories prédéfinies. Les documents peuvent être dans une collection statique ou peut arriver dans un flux de données (par exemple, news-wire (fil d'infos)). Il y a habituellement un ensemble de données étiquetées (document : catégorie paire) qui peuvent être utilisés pour faire l'apprentissage aux classificateurs pour les différentes catégories [5].

Regroupement de documents

Dans le clustering de document, il n'y a pas de données d'apprentissage, et le but est de découvrir automatiquement la structure dans une collection de documents non étiquetés. Le Clustering a été utilisé pour organiser des collections de documents pour améliorer l'efficacité et la performance de la recherche et récupération d'information [6] ; [7].

Extraction d'information

Dans l'extraction de l'information, l'objectif est de trouver automatiquement et extraire des caractéristiques ou des faits spécifiques d'un domaine comme des entités, des attributs et des relations d'un document [8]. Parmi les types d'informations habituellement extraites en trouve des noms d'organisations, les gens, les lieux et les dates.

Résumé de documents

L'objectif dans le résumé de document est de dériver automatiquement une représentation du document qui capture ses caractéristiques importantes brièvement. Il y a eu un peu de travail pour essayer de tirer automatiquement des résumés de documents de texte pour une utilisation dans la recherche d'information [9].

Modèles de recherche d'information

Le rôle d'un modèle de RI est de fournir une formalisation du processus de RI avec un cadre théorique pour la modélisation de la mesure de pertinence. Dans la littérature, il y a beaucoup de modèles de RI textuelle développé, qui ont en commun le vocabulaire d'indexation basé sur les mots clés ; et ce différent principalement par le modèle d'appariement requête-documents (modèle de correspondance). Généralement le vocabulaire d'indexation est défini par l'ensemble :

$$V = \{t_i\}, i \in \{1, \dots, n\}$$

Où n représente le nombre de mots qui apparaissent dans les documents.

Selon les travaux de [10] un modèle de RI est défini par un quadruplet (D, Q, F, R(q,d)) : où

- D est l'ensemble de documents
- Q est l'ensemble de requêtes
- F est le schéma du modèle théorique de représentation des documents et des requêtes
- R(q,d) est la fonction de pertinence du document d à la requête q

Dans cette section qui suit, on va présenter les principaux modèles de RI qui sont : le modèle booléen, le modèle vectoriel et le modèle probabiliste.

Modèle booléen

Le modèle booléen selon [11] se base sur la théorie des ensembles. Les documents et les requêtes sont représentés par des ensembles de mots clés. Chaque document est représenté par une conjonction logique des termes non pondérés qui constitue l'index du document. A cet effet, le document est représenté par la formule suivante :

$$d = t_1 \wedge t_2 \wedge t_3 \ldots \wedge t_n.$$

Une requête peut être représentée par une expression booléenne dont les termes sont reliés par des opérateurs logiques (OR, AND, NOT) permettant d'effectuer des opérations d'union, d'intersection et de différence entre les ensembles de résultats associés à chaque terme. L'expression suivante représente un exemple de requête :

$$q = \left(t_1 \bigwedge t_2\right) \bigvee \left(t_3 \bigwedge t_4\right)$$

La fonction de correspondance est basée sur l'hypothèse de présence/absence des termes de la requête dans le document et vérifie si l'index de chaque document d_j implique l'expression logique de la requête q. Le résultat de cette fonction est donc binaire, est décrit comme suit :

$$RSV(q, d) = \{1,0\}.$$

Modèle vectoriel

Dans ces modèles, et selon [11], la pertinence d'un document vis-à-vis à une requête est définie par des mesures de distance dans un espace vectoriel. Le modèle vectoriel représente les documents et les requêtes par des vecteurs d'un espace à n dimensions, les dimensions étant constituées par les termes du vocabulaire d'indexation. L'index d'un document dj est le vecteur

$$\overrightarrow{d_j} = \left(w_{1,j}, w_{2,j}, w_{3,j}, \ldots, w_{n,j}\right),$$

où $w_{k,j} \in [0\ 1]$ dénote le poids du terme t_k dans le document d_j.

Une requête est également représentée par un vecteur

$$\vec{v} = \left(w_{1,q}, w_{2,q}, w_{3,q}, \ldots, w_{n,q}\right),$$

où $w_{k,q}$est le poids du terme t_k dans la requête q.

La fonction de correspondance mesure la similarité entre le vecteur requête et les vecteurs documents. Une mesure classique utilisée dans le modèle vectoriel est le cosinus de l'angle formé par les deux vecteurs :

$$RSV_{q,d} = \cos\overrightarrow{}_{q},\overrightarrow{}_{d}$$

Si l'angle formé est petit alors les vecteurs sont similaires, et donc le cosinus de cet angle est grand. A l'inverse du modèle booléen, la fonction de correspondance évalue une correspondance partielle entre un document et une requête, ce qui permet de retrouver des documents qui ne reflètent pas la requête qu'approximativement. Les résultats peuvent donc être ordonnés par ordre de pertinence décroissante.

La figure 1 présente un exemple du modèle vectoriel de deux document d1 et d2 avec une requête q, les deux ongles formés sont respectivement α et θ.

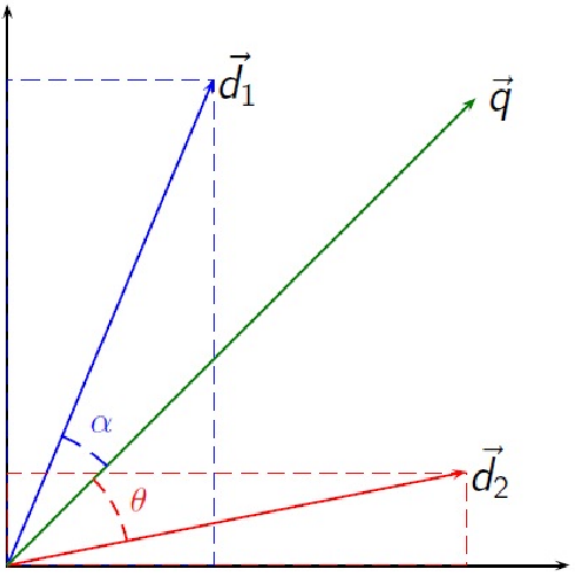


Figure 1 Représentation de deux documents (d1 et d2) et d'une requête (q) dans un espace vectoriel. La proximité de la requête aux documents est représentée par les angles et entre les vecteurs.

Modèle probabiliste

Selon les travaux de [12] ; [1] ; [13], ce modèle exprime une estimation de la probabilité de pertinence d'un document par rapport à une requête. Ainsi, il sert à classer une liste de documents dans l'ordre décroissant d'utilité probable pour l'utilisateur. Etant donné une requête utilisateur Q et un document D, il s'agit de calculer la probabilité de pertinence du document pour cette requête. Deux scénarios qui se présentent : D est pertinent pour Q et D n'est pas pertinent pour Q. Les documents et les requêtes sont représentés par des vecteurs

booléens dans un espace à n dimensions. L'exemple suivant représente un document dj et une requête q :

$$dj = (w_{1,j}, w_{2,j}, w_{3,j}, \dots, w_{n,j}),$$

$$q = (w_{1,q}, w_{2,q}, w_{3,q}, \dots, w_{n,q}).$$

Avec $w_{k,j} \in [0\ 1]$ et $w_{k,q} \in [0\ 1]$.

La valeur de $w_{k,j}(resp. w_{k,q})$ indique si le terme tk apparaît ou non dans le document $d_j(resp. q)$.

Le modèle probabiliste évalue la pertinence du document dj pour la requête q. Un document est sélectionné si la probabilité que le document d soit pertinent, notée $p(R/D)$, est supérieure à la probabilité que d soit non pertinent pour q, notée $p(\bar{R}/D)$ où R est l'événement de pertinence et $\bar{R}$ est l'événement de non pertinence. Le score d'appariement entre le document D et la requête Q, noté $RSV\ (Q, D)$ est donné par :

$$RSV(D|Q) = \frac{(R|D)}{(\overline{R}|D)}$$

Chapitre 2 – Systèmes de recherche d'information

Introduction

Selon [14], en peut définir un système de recherche d'information par trois composantes principales qui sont :

- Création de représentation de documents (Indexation) ;
- Création de représentation de requêtes (Formulation de requête) ;
- Comparaison des représentations des requêtes et des documents (Recherche).

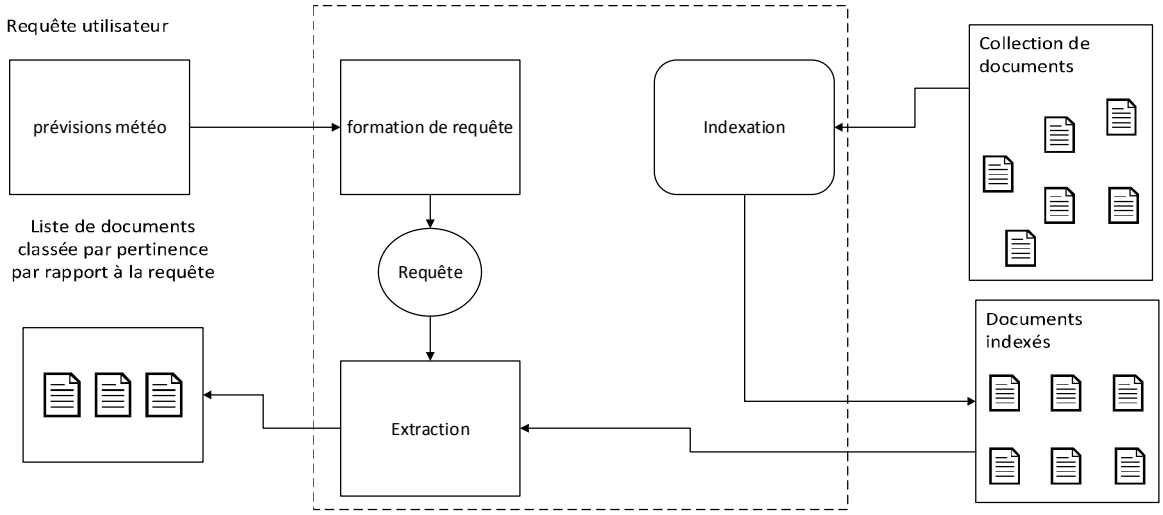


Figure 2 Schéma illustrant les principaux composants d'un système de recherche d'information.

Composantes d'un système de recherche d'information

Selon la définition citée précédemment, un SRI est composé de trois composante principales :

- L'indexation (représentation de document) ;
- La requête ou interrogation (représentation des besoins d'utilisateur) ;
- L'appariement ou la comparaison (recherche).

L'indexation

L'indexation est un processus qui permet de capturer les informations importantes contenues dans le document original sous forme des indexes, qui lui permet d'être comparé à des représentations d'autres documents et représentations des requêtes utilisateur. Cette première tâche est généralement effectuée au profil du processus de recherche car, la

construction des index est une tache fastidieuse et couteuse vis-à-vis le nombre de documents de la collection ainsi que de la taille de la collection.

En générale, la représentation sous forme d'indexes a un caractère réducteur, car tous les termes d'un document ne sont pas importants à prendre en compte pour la recherche. On peut dire que le document va être présenté sous forme de liste de termes qui sont les plus représentatifs et les plus marquant du sujet du document.

Dans la littérature, il y a trois formes d'indexation [3] :

- Indexation manuelle (Réalisé par expert dans le domaine ou les bibliothéconomie) ;
- Indexation semi-automatique (effectué par des systèmes informatiques assistés par des experts) ;
- Automatique (Réalisé purement par des systèmes informatiques).

Bien que les techniques d'indexation se différent, en peut trouver plusieurs indexations différentes d'une même ressource, aussi valables les unes que les autres, en fonction de l'usage qui doit en être fait et du public auquel elles s'adressent.

Phases d'indexation :

L'indexation se décompose en trois phases (voir figure 3) :

- L'extraction des termes du document.
- La sélection des termes discriminatifs pour un document.
- La pondération des termes.

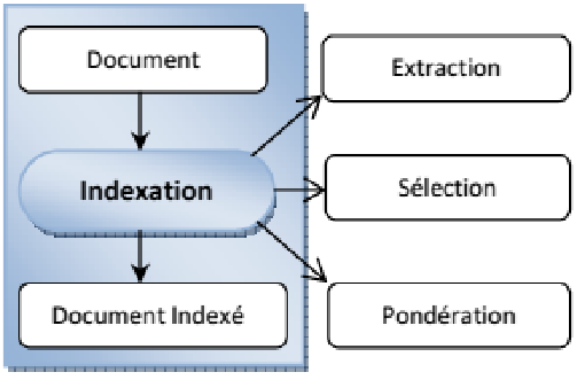


Figure 3 Phases d'indexation d'un document [14]

L'interrogation (Requête) :

L'utilisateur exprime sons besoin en information par une requête dans la forme imposée par le système (SRI), ce processus est connu par la formulation de la requête. Comme dans le cas des représentations de documents, la requête doit être capable de capturer les informations importantes contenues dans la requête d'origine sous une forme qui lui permet d'être comparée à la représentation du document. La requête est utilisée par le système de recherche pour sélectionner les documents pertinents de la collection.

La réponse du système est un ensemble de références triées à des documents qui obtiennent une valeur de correspondance élevée. Cet ensemble est généralement présenté sous la forme d'une liste ordonnée suivant la valeur de correspondance [15].

La recherche (L'appariement) :

Le processus crucial d'un système de recherche d'information est la comparaison de la représentation de la requête avec les représentations de documents. Ceci est le processus de recherche ou de récupération. En général, une fonction de correspondance est celle qui sélectionne les documents pertinents de la collection sur la base des représentations des requêtes et des documents. En principe, chaque document de la collection est comparé à la requête afin de déterminer sa pertinence.

Les Performance des SRI

La performance d'un système de recherche et de récupération d'information, peut être mesurée sur de nombreuses dimensions. Dans les applications du monde réel (application mise en service dans les entreprises), des facteurs tels que le coût de mise en œuvre et la maintenance, la facilité d'indexation de nouveaux documents, et la vitesse de récupération sont importants. Néanmoins les critères de performance les plus populaires sont ceux de l'efficacité de récupération qui sont habituellement composés de deux mesures : rappel et précision. Le tableau suivant indique pour chaque mesure son nom, ce qu'elle représente et sa définition [16] :

Mesure	Description	Définition
Rappel	Rappel exact par rapport à l'ensemble des documents retrouvés. Le rappel mesure la capacité du système à restituer l'ensemble des documents	$R = \frac{Nb\ de\ Doc\ pertinents\ retrouvés}{Nb\ de\ Doc\ pertinents}$ Nombre totale de documents retrouvés limité aux 1000 premiers (valeur fixée par les programmes d'évaluation utilisés).

	pertinents (en lien avec le silence documentaire).	
Précision	Précision moyenne non interpolée par rapport à l'ensemble des documents pertinents. Mesure la capacité du système à ne restituer que des documents pertinents (en lien avec le bruit documentaire). Comme les listes de résultats qui sont évaluées sont limitées à 1000 documents, précision exacte et P1000 (voir plus bas) sont proches.	$P = \frac{Nb\ de\ Doc\ pertinents\ retrouvés}{Nb\ de\ Doc\ pertinents}$ Nombre total de documents retrouvés, limité aux 1000 premiers (valeur fixée par les programmes d'évaluation utilisés).

Tableau 2 Présentation des deux mesures de performance les plus utilisées d'un SRI

Le rappel est la fraction de tous les documents pertinents dans l'ensemble de la collection qui sont récupérés en réponse à une requête. La précision est la fraction des documents récupérés qui sont pertinents. En balayant la liste des documents classés, une courbe de précision rappel peut être tracée. La figure 4 présente un exemple de cette courbe.

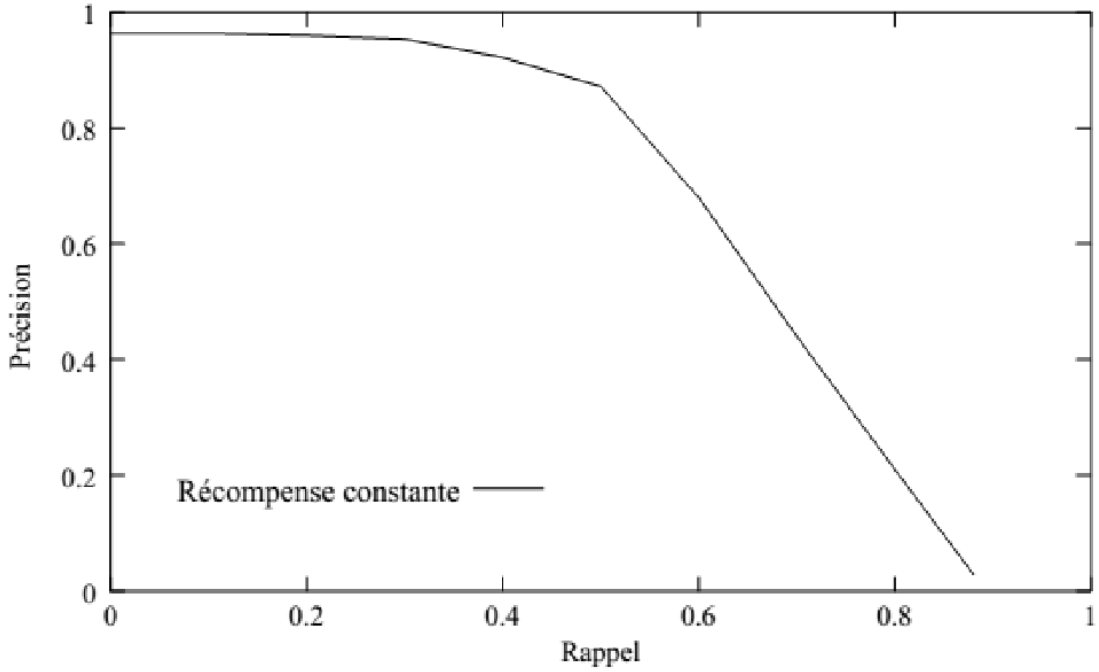


Figure 4 Courbe rappel/précision de la méthode à récompense constante. [17]

Ce graphique indique l'ensemble des points de fonctionnements possibles qui peuvent être obtenus par seuillage de la liste des documents trouvés à divers points. Les facteurs rappel et précision, varient généralement inversement l'un avec l'autre.

Chapitre 3 – Documents texte Vs. Documents parlés

Introduction

De point de vue structurel, il existe de nombreuses différences entre les documents texte et les documents parlés qui soulèvent de nouveaux challenges qui doivent être abordées dans le processus de recherche d'information.

Parmi ces challenges, nous citons que la parole est un support plus riche et plus expressive que le texte [18]; il contient plus d'informations que les mots. Avec la parole, des informations telles que l'identité de la langue parlée, l'identité du locuteur, et l'attitude ou le ton du locuteur, exprimées en indices prosodiques, sont capturées en plus de mots parlés.

Cependant, ces informations supplémentaires peuvent être utiles dans le développement des systèmes de recherche et de récupération des informations ; elles offrent de nouvelles voies d'indexation. Le type d'information commun entre les deux types de documents texte et parlé, est le concept du sujet ou le sujet du document. Dans ce contexte, nous intéressons dans ce mémoire sur le contenu des documents parlés ; l'utilisation des autres informations acoustiques contenues dans le signal de parole telle que : l'identité de locuteur ou ses caractéristiques sont au-delà de la portée de ce travail.

Le deuxième point est de savoir comment extraire et représenter le contenu d'un document parlé sous une forme qui peut être efficacement stockée et recherchée avec précision. Bien qu'une tâche similaire doive être faite avec des documents de texte, le passage du texte à la parole ajoute une couche supplémentaire de complexité et d'incertitude. Il y a beaucoup de défis, y compris être capable de gérer plusieurs locuteurs, la parole bruyante, la parole conversationnelle ou spontanée et en plus de traiter un langage de très grands vocabulaires. Dans ce contexte, nous détaillons dans la deuxième partie les techniques et les outils qui permettent de surmontés ces problèmes, en introduisant les concepts liés au langage elle-même comme les syllabes et la modélisation phonétique.

Le troisième point est la robustesse des modèles de recherche dans le contenu des documents multimédia au milieu bruité et aux erreurs de transcription. Dans ce contexte, la plupart des méthodes d'indexation et de recherche qui ont été développés pour les documents texte suppose implicitement que les transcriptions se génèrent sans erreur. Avec le texte, les mots dans les documents sont supposés être connus avec une certitude élevée. Par conséquent, il n'y a pas de mécanisme explicite dans les modèles pour le traitement des

erreurs dans la représentation du document. Cependant, avec la parole, il n'y a pas actuellement des méthodes de transcription automatique parfaite et il y aura probablement des erreurs dans les transcriptions générées par le système de reconnaissance vocale. L'utilisation de dictionnaires pour vérifier l'orthographe des mots, des thesaurus pour élargir les mots ou des ontologies avec la combinaison de mots à leurs radicaux peuvent tous être considérés comme des approches qui abordent la question de la robustesse à des degrés divers. La modification du modèle d'indexation et de recherche en utilisant une fonction de correspondance plus complexe pour permettre la correspondance approximative des termes d'indexation, ainsi que d'autres techniques pour traiter les erreurs des représentations de documents.

Travaux reliés au traitement de la parole

Depuis plusieurs années, le domaine de la reconnaissance automatique de la parole attire l'intérêt des chercheurs afin de développer des applications dans de nombreux domaines. Parmi les nombreux travaux de recherche liés à ce domaine, la détection de mots-clés qui constitue une branche importante. Cette technique consiste à détecter dans un flux audio continu certaines parties de la parole où des mots-clés ont été susceptibles d'avoir été prononcés.

Dans cette section en va se limiter à quelques travaux à titre d'exemple seulement, et ce point sera détaillé plus loin dans la deuxième partie de ce mémoire.

Dans les littératures, on trouve quelques applications importantes pour l'exploitation des documents parlés en utilisant la technique de recherche de mots clés « word spotting » telles que la reconnaissance de mots-clés appliquée à la téléphonie [19], le tri des messages vocaux [20] ou encore la numérotation téléphonique dans les GSMs [21].

Reconnaissance de mots-clés appliquée à la téléphonie

Parmi les chercheurs qui ont travaillé dans le domaine de reconnaissance de mots-clés « *Keyword spotting* », on trouve Wilpon et ses collègues [19]. Ils ont réalisé une étude approfondie sur la reconnaissance de mots isolés appliquée à la téléphonie. La base de données qui ont travaillé dessus et celle construite par AT&T en 1988 qui contient environ 70000 appels, où les personnes étaient censées dire un des cinq mots prédéfinis « *'collect', 'calling-card', 'third-number', 'person', 'operator'* » de façon à obtenir le service désiré.

Les résultats obtenus ont montrés que 17% des personnes ne citaient pas un des cinq mots clés, mais ajoutaient également dans leurs messages d'autres mots comme *« 'collect call please'»*. Entre d'autre, on trouve les travaux de Bossemeyer qui ont montrés que le taux de reconnaissance par mot isolé se décroît de 97% à 90% lorsque ces mots parasites étaient présents [22]. Pour remédier à ce problème, ils ont proposé d'utiliser un modèle poubelle qui permit d'améliorer ces taux respectivement à 99.3% et 95.1%.

Reconnaissance des mots-clés appliquée au tri de messages vocaux

R.C. Rose, [23], a proposé en 1991 d'utiliser la reconnaissance de mots clés pour la classification des messages vocaux en 6 classes différentes en fonction de leur contenu. En se basant sur 120 mots clés. Ils ont obtenu un taux moyen de détection de 69% avec un taux de fausses alarmes de 5.4% par mot clé et par heure. Ces détections subissaient un post-traitement en fonction du taux de reconnaissance de chaque mot clé. En associant chacune de ces 6 classes avec 20 mots clés, il parvenait à classifier les phrases, d'une durée moyenne de 30 secondes, avec un taux de réussite moyen de 82.4%.

Reconnaissance des mots-clés appliquée à la numérotation téléphonique

S. Nakamura, [21], a proposé lui aussi, l'utilisation de la reconnaissance de mots-clés, pour la numérotation téléphonique dans les GSMs, l'idée est que chaque nom propre sur l'annuaire du GSM correspondant à un numéro à composer. Les contraintes étaient fortes, vu que le système de reconnaissance devait pouvoir fonctionner sur le DSP[1]

du téléphone et dans des conditions d'environnements très bruités qui étaient la voiture, mais il pouvait cependant être dépendant du locuteur. Il a utilisé une méthode de pré-segmentation basée sur la puissance acoustique. Puis, il applique un algorithme de programmation dynamique pour comparer les mots-clés. Les résultats obtenus étaient de l'ordre de 90% de détections correctes pour un vocabulaire de 100 mots.

[1] Un DSP est l'un des éléments clés dans un téléphone cellulaire numérique.

Conclusion

Dans cette partie, on a essayé de donner une vue d'ensemble sur les systèmes de recherche d'information on ordre général : principe, architecture et fonctionnalité. Certainement ces systèmes sont conçus essentiellement pour la fouille dans les documents textes. Néanmoins, ils peuvent être utilisés pour des documents plus complexes tels que les documents multimédias et les documents parlés. En d'autres termes, les systèmes de recherches dans le contenu multimédia sont celle utilisés pour les documents texte. Cependant, il faut introduire les techniques de traitements des documents parlés « les techniques de traitements de la parole » dans ces systèmes afin qu'ils puissent les traiter.

A cet effet, dans la deuxième partie, on va détailler d'une façon approfondi les caractéristiques des systèmes de recherches dans les documents multimédia, ainsi que les techniques utilisées.

Partie 2

-

Concepts, techniques et état de l’art

Introduction

La première finalité des systèmes de reconnaissance automatique de la parole est la transcription entière du signal de parole en mot. Cependant, pour traiter les signaux de parole à large vocabulaire, il faut utiliser des unités phonétiques autres que le mot. Dans ce contexte, il existe de nombreuses autres unités possibles pour la segmentation de la parole, parmi eux on trouve :

- Les phones, ou unités sous-phonèmes, qui, fusionnées entre elles, permettent d'obtenir des unités plus longues ;
- Les phonèmes, ou l'unité la plus courte qu'un être humain est capable d'identifier dans de la parole continue ;
- Les allophones, ou les différentes réalisations sonores possibles d'un phonème ;
- Les diphones, demi-syllabes, et syllabes [24] qui permettent d'incorporer les phénomènes Co articulatoires ;
- Les graphèmes, ou unité fondamentale d'une écriture donnée ;
- Les morphèmes, ou les plus petites unités porteuses de sens qu'il soit possible d'isoler dans un énoncé ;
- Les subwords, ou unités sous-mots, terme générique regroupant les différentes unités évoquées ci-dessus.

L'unité qui nous intéresse plus particulièrement est le subwords (tri-phonèmes ou phonème). Le phonème est considéré comme étant la plus petite unité qui permet de faire la distinction entre les caractères d'un mot. Donc les phonèmes d'une langue peuvent être identifiés par appariement des mots de sens différents, mais ne différant que par un seul son. Ces mots sont appelés paires minimales, à l'exemple des mots balle et palle, mot de sens différents dont la forme sonore ne se distingue que par le /b/ et le /p/.

Dans la section suivante, on va éclaircir l'obscurité autour du domaine de reconnaissance automatique de la parole (Automatic Speech Recognition - *ASR*) et les concepts théoriques liés de manière directe ou indirecte à ce domaine. Puis, nous présentons les principales stratégies de recherche existantes dans les contenus parlés, Puis, nous détaillons les approches générales du domaine de la détection des termes parlés (Spoken Term Detection *STD*). Finalement, nous traçons un état de l'art sur les principaux travaux

existants dans la littérature sur le domaine de détection de mots clés (keyword spotting) où on va réaliser une synthèse globale des approches et travaux existants qui nous amène vers notre solution proposée plus loin dans la troisième partie de ce mémoire.

Chapitre 1 – Stratégies de recherche dans le contenu parlés

Introduction

Dans cette section, nous discutons la recherche et la consultation des documents parlés. Nous nous concentrons principalement sur l'application de la détection de mots clés dans les documents parlés où un utilisateur fournit une requête et le système renvoie un ensemble de documents audio qui contient les termes correspondant le mieux représentatifs à la requête.

Stratégies principales de recherche dans le contenu parlé

Recherche de documents parlés (SDR)

La recherche de documents parlés consiste à récupérer des documents audio qui répondent aux requêtes des utilisateurs à partir d'une grande collection de ressource de données multimédia. Le scénario de base est de supposer qu'un utilisateur fournira une requête et le système retournera une liste de documents audio qui contiennent les concepts recherchés triés selon leur degré de pertinence par rapport à la requête exécutée.

La requête est généralement supposée être sous forme d'une chaine de mots à base de texte, bien que les requêtes orales puissent être utilisées à la place du texte dans certaines applications.

Recherche d'énoncés parlés (SUR)

Pour la consultation de documents audio de grande taille, il devient important de localiser les parties spécifiques de ces documents qui sont réellement pertinentes à la requête. Dans certains scénarios de SDR, cela peut être réalisé en segmentant les longs documents en segments plus petits se rapportant à des sujets spécifiques et en localisant ces segments thématiques. Alternativement, un système pourrait fonctionner sur les segments aussi courts que les énoncés oraux individuels (à savoir, des segments de parole presque équivalents à des phrases prononcées). Lorsque la tâche nécessite la récupération des petits segments au lieu des documents audio intégraux, la tâche est connue comme recherche d'énoncés parlés (SUR). Dans ce cas, le but de SUR est de trouver tous les énoncés relatifs à la requête même lorsque plusieurs énoncés relatifs à la requête existent dans un document unique.

Détection de termes parlés (STD)

Dans les deux cas de SDR et SUR, le but est de trouver les documents parlés qui sont pertinents pour une requête utilisateur. En général, il est supposé que des documents ou des énoncés contenant des mots dans la requête seront pertinents. Cependant, la pertinence perçue d'un document est souvent une notion subjective qui n'est pas facile à définir ou à évaluer, car les utilisateurs peuvent avoir chacun leur propre satisfaction sur la qualité et l'ordre de pertinence des ressources audio retournées par rapport à une requête donnée. Cependant, la détection de mots clés est un domaine d'application lié à la recherche de documents (SDR) qui fournit une mesure plus concrète de l'évaluation. Dans les applications de STD, le but est de trouver tous les occurrences phonétiques d'un mot spécifique d'une requête. Dans ce cas, les résultats retournés sont soit correcte soit incorrecte et aucune détermination subjective de la pertinence n'est nécessaire. La figure 5 illustre le cadre standard d'un système STD.

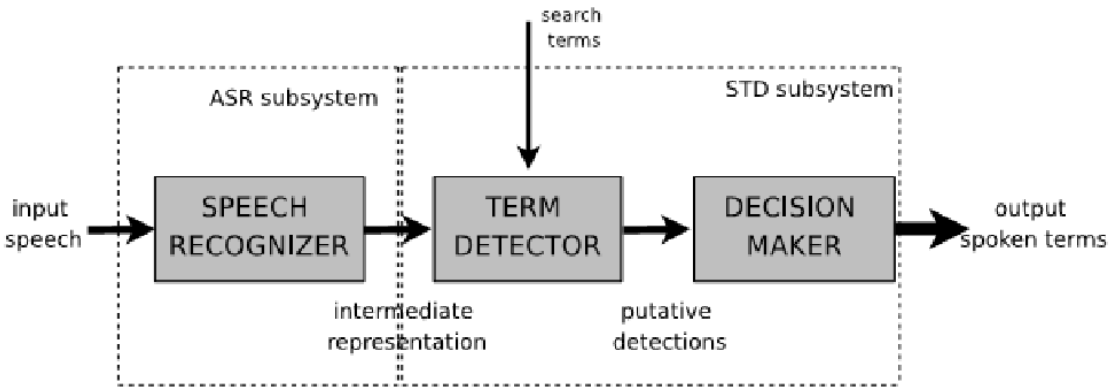


Figure 5 Architecture standard d'un système STD [25]

On trouve qu'il existe trois éléments dans cette architecture : un système de reconnaissance de la parole utilisé pour transcrire la parole à des représentations plus élémentaires tel que la représentation syllabique ou par phonèmes ; un noyau de détection de terme utilisé pour les rechercher selon leurs présentations phonétiques afin de trouver les occurrences correspondant aux de termes de la requête ; et un outil de décision qui permet d'affirmer les détections fiables et rejettent les fausses alarmes. Un système STD fonctionne en deux phases : lors de la phase d'indexation (qui est hors ligne), les documents vocaux sont transcrits et archivés ; lors de la phase de détection (qui est en ligne), les requêtes sous la forme de séquences courtes de mots sont recherchées dans les ressources multimédias et les occurrences possibles des requêtes sont repérées.

Relation de STD avec d'autres tâches

Relation de STD avec le domaine de recherche des documents parlés

Dans l'architecture standard illustrée par la figure 4, on trouve que le STD repose sur la reconnaissance automatique de la parole (ASR) ; en ce sens, STD appartient à la famille du domaine de recherches de la reconnaissance automatique de la parole -RAP. Ces systèmes ont été développés depuis plus d'un demi-siècle, et ont obtenu des succès significatifs de point de vue précision et efficacité [26]. La tâche de base des RAP est la transcription automatique de parole, qui a évolué à un système de reconnaissance vocale à un grand vocabulaire continue (LVCSR), et a obtenu une grande précision sur le discours lu et même elle est étendue vers la parole spontanée. Outre la transcription, d'autres tâches sont également basées sur les techniques de RAP y compris l'évaluation de la qualité de la voix, le résumé des documents et les STD.

Dans cette section, nous comparons STD avec quelques d'autres tâches qui sont les plus pertinents, y compris la transcription de parole, détection de mot-clé et SDR.

Relation de STD avec la transcription de parole

La transcription de la parole est la tâche essentielle des ASR et le fondement de STD. Selon la figure 5, le système de STD utilise un système de reconnaissance de parole pour convertir la parole d'entrée à des représentations intermédiaires, ce qui signifie que la précision de la transcription influe directement sur la performance de l'ensemble des STD. D'autre part, la performance de STD n'est pas entièrement déterminée par la précision de la transcription, car ils ont des objectifs différents : pour la transcription, l'objectif est d'arriver à taux d'erreur de reconnaissance de mot (WER) faible, de sorte que tous les mots sont traités de manière égale ; cependant, pour STD, seuls les termes de recherche sont importants. Par conséquent, pour obtenir une bonne performance de STD, un transcripteur de la parole idéale dans le sous-système ASR devrait être précis sur les termes de recherche, mais insensible à d'autres termes, tels que les mots d'outils de langage comme « le », « que », « leur » et d'autres mots de fonctions.

Relation de STD avec la détection de mots-clés

Le but de la détection de mot-clé et STD sont similaires : les deux ont pour objectif de détecter des mots ou des séquences de mots pertinents dans le signal de la parole. La principale différence est que la détection de mots-clés traditionnels détecte les mots-clés depuis un flux audio, tandis que STD détecte les termes de recherche depuis une

représentation intermédiaire ; mais ceci n'est pas absolu, car de plus en plus de systèmes de détection de mots clés utilisent des modèles transcriptions phonétiques comme des représentations intermédiaires.

D'autres différence est qu'un système de détection de mots clés a généralement un vocabulaire fixe tandis qu'un système de STD a un vocabulaire ouvert, ce qui signifie qu'un système STD est en mesure de rechercher un terme sans passer par la retranscription du signal de la parole. Encore une fois, ce n'est pas absolu, puisque de nombreux chercheurs présentent des systèmes de détection de mots clés à un vocabulaire ouvert aussi. La dernière différence est dans les paramètres d'évaluation : STD utilise des nouvelles mesures d'évaluation définies par le NIST[1], en particulier (ATWV), alors que la détection de mots clés utilise des paramètres classiques tels que le facteur de mérite (FOM). Alors en peut dire que STD n'est qu'un « nom moderne » de la détection de mots clés, mais un système suivant une architecture standard et des paramètres d'évaluation standard définis par NIST.

Relation de STD avec SDR

SDR et les STD sont similaires de point de vue qu'ils visent à récupérer certains segments intéressants de la parole, de sorte qu'ils utilisent certaines techniques d'extraction, par exemple, la recherche des représentations phonétiques, la modélisation de sous mot, etc. La différence est que les systèmes SDR ne sont pas intéressés par les positions exactes d'occurrences de termes particuliers, mais par contre il récupère des segments de parole entiers. Par conséquent, SDR est moins touché par la précision de transcription que STD, car il peut utiliser des informations de haut niveau, par exemple, la fréquence des mots dans un document, des modèles linguistiques de longue, les structures sémantiques, etc. D'autre part, STD est une approche importante pour SDR [27].

[1] NIST : Nationl Institute of Standard and Technology

Chapitre 2 – Fondements théoriques sur les SRAP

Introduction

Dans ce chapitre, on va présenter les concepts et les techniques des systèmes de reconnaissance automatique de la parole (SRAP). Nous détaillons leur principe général en introduit les concepts de base nécessaires à leur principe de fonctionnement, notamment les modélisations acoustiques et linguistiques ainsi que les algorithmes de décodage.

Système de reconnaissance de la parole

L'objectif des systèmes de reconnaissance automatique de la parole (SRAP), est la transcription automatique de la langue parlée en texte lisible en temps réel. Le but ultime de la recherche RAP est de permettre à un ordinateur de reconnaître en temps réel, avec un degré de précision proche de l'idéal, quelques mots qui sont intelligiblement parlés par toute personne, indépendamment de la taille du vocabulaire, le bruit, les caractéristiques du locuteur ou l'accent. Les principales applications utilisant des SRAP sont la transcription automatique, l'indexation de documents multimédias et le dialogue homme-machine. La figure 6 présente la structure générale d'un système SRAP.

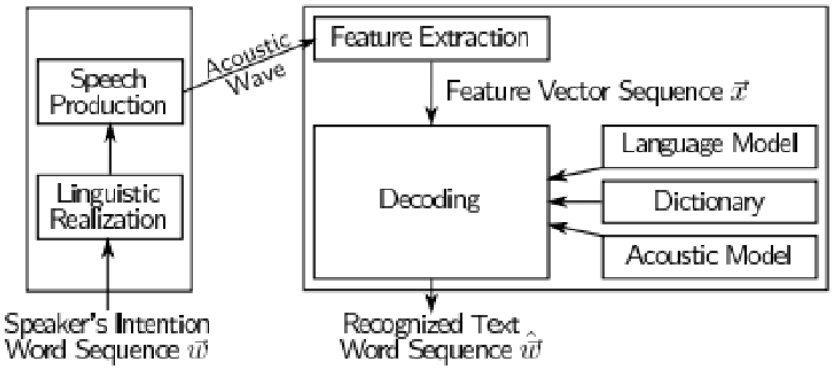


Figure 6 Structure générale d'un SRAP

Un tel système SRAP à trois fonctions principales :

- Le traitement acoustique qui nous permet d'extraire les caractéristiques acoustiques qui représentent le mieux le signal de la parole, en convertissons la forme brute de la parole qui est sous forme d'onde vocale en vecteurs de paramètres fréquentiels et temporels discrète [28] . Ces vecteurs sont appelés les vecteurs acoustiques.

- l'apprentissage automatique qui est réalisé par une association entre les segments élémentaires de la parole et les éléments lexicaux. Elle fait appel aux techniques de modélisation stochastiques comme les modèles de Markov cachés MMC (en Anglais : HMM - Hidden Markov Models) et/ou par les réseaux de neurones artificiels (ANN, Artificial Neural Networks). On note que les modèles de Markov cachés (MMC) sont les plus utilisés à ce jour [28].
- Le décodage qui est la fonction élémentaire dans les SRAP, consiste à faire des correspondances acoustiques entre le vecteur acoustique et les modèles sources d'information pour détecter la séquence la plus probable.

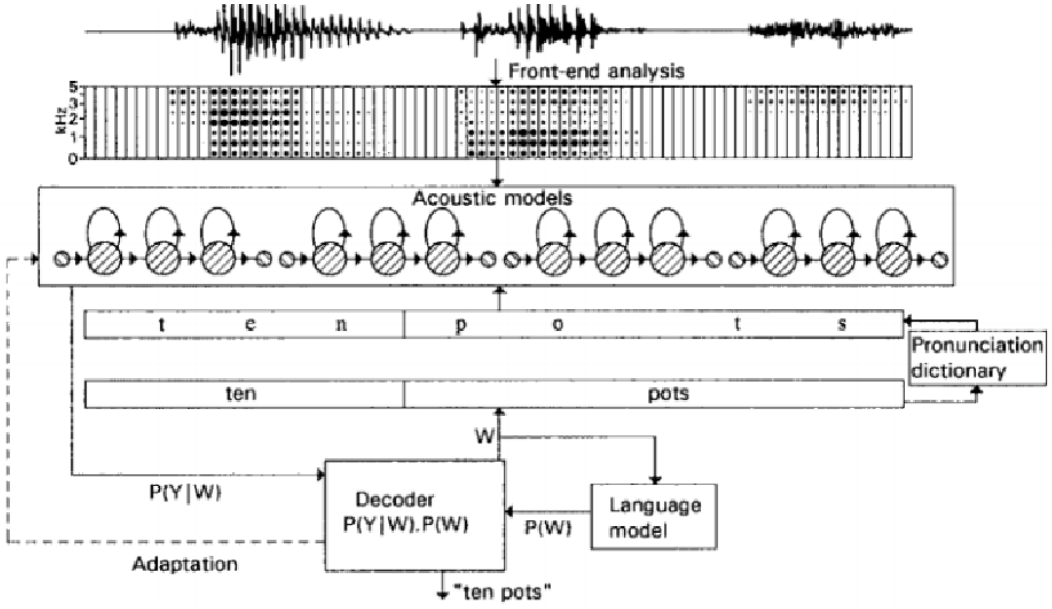


Figure 7 Exemple de décodage d'un signal de parole en calculant la probabilité d'une séquence de mots en termes des probabilités du modèle de langage et du modèle acoustique, montré pour la reconnaissance de l'expression «ten pots». [29]

Cependant, pour la réalisation d'une transcription des documents parlés, les SRAP s'appuient sur les modèles suivants :

- le modèle de langage qui calcule la probabilité *P(W)* pour chaque suite de mots *W* dans le langage cible.
- le modèle acoustique de prononciation qui donne pour chaque suite de mots *W*, la ou les prononciations possibles *H* avec une probabilité $P(H|W)$.
- le modèle acoustico-phonétique qui estime la probabilité $P(X|H)$ de la séquence observée de vecteurs acoustiques *X* étant donnée une prononciation possible *H* d'une séquence de mots donnés.

Donc le SRAP doit calculer la probabilité de toute suite de mot W étant donné une observation de signal de parole X comme présenté dans la formule suivante [30] :

$$\widehat{W} = \underset{W}{\operatorname{argmax}} P(W/X)$$

En appliquant la théorie de Bayes, l'équation devient :

$$\widehat{W} = \underset{W}{\operatorname{argmax}} \left(\frac{P(X/W)P(W)}{P(X)} \right)$$

A savoir que la séquence de mots les plus probables $P(W)$ est indépendante de la probabilité de séquence d'observation $P(X)$, donc $P(X)$ peut être sortie de la formule :

$$\widehat{W} = \underset{W}{\operatorname{argmax}} (P\,X/W)\,P(W)$$

Modélisation acoustique

La modélisation acoustique la plus efficace est basée sur une structure dénommée modèles de Markov cachés (MMC). Depuis que la technique a été introduite en 1970, elle est rapidement devenue la forme dominante de la modélisation acoustique et a été appliquée à toutes sortes de tâches de reconnaissance de la parole. Ceci est en partie à cause de l'existence de paramètres efficaces de formation et des algorithmes de reconnaissance pour HMM [28]

Avant d'aller plus loin, il existe plusieurs aspects qui doivent être définis avant l'utilisation des HMM. On cite les aspects suivants : la fonction de discrimination, le choix d'unité de la parole, la topologie du modèle, le modèle de distribution de l'observation, l'initialisation des paramètres et quelques techniques d'adaptation.

Fonction discriminative

On considère le modèle HMM λ défini par : $\lambda_{MV} = \underset{\lambda}{argmax}\, P(O/\lambda)$, le maximum de vraisemblance (MV) cherche à maximiser la probabilité d'observation d'une séquence O. Cependant, le problème de reconnaissance de la parole peut être défini comme étant une tâche de classification où on définit pour chaque classe acoustique $c \in 1..C$ un modèle HMM correspondant λ_c. Donc le MV permet d'estimer le modèle λ_c par rapport à la séquence acoustique O^c de la classe C, par la formule suivante :

$$(\lambda_c)_{MV} = \underset{\lambda}{argmax}\, P(O^c/\lambda)$$

En effet, le critère MV estime chaque modèle HMM à part, donc il ne garantit pas une solution optimale pour minimiser l'erreur de reconnaissance. Il ne prend pas en compte

la capacité de discrimination de chaque modèle (la capacité de distinguer les observations générées par le modèle correct de ceux générés par les autres modèles). Un autre critère qui permet de maximiser cette dernière est le critère de Maximum d'information Mutuel (MIM) [31]. L'information mutuelle entre l'observation de la séquence O^c et la classe c, paramétré par : $\Lambda = \{\lambda_c\}$, $c = 1,2,\dots,C$ est

$$I_\Lambda(O^c, c) = log \frac{P(O^c/\lambda_c)}{\sum_{w=1}^{C} P(O^c/\lambda_w, w)P(w)}$$

$$= log\, P(O^c/\lambda_c) - log \sum_{w=1}^{C} P(O^c/\lambda_w, w)P(w)$$

Le critère MIM cherche à trouver l'ensemble de modèles Λ qui maximise l'information mutuelle.

$$\Lambda_{MIM} = \max_{\Lambda} \left\{ \sum_{c=1}^{C} I_\Lambda(O^c/c) \right\}$$

En pratique, le MIM est basée sur une variante de l'algorithme Baum-Welch appelé Extended Baum-Welch qui maximise ce critère. En bref, l'algorithme calcule les probabilités avant-arrière pour les séquences d'apprentissage. Puis, un autre passage avant-arrière calcule les probabilités sur toutes les autres expressions possibles.

Le choix d'unité de parole

Plusieurs aspects doivent être pris en compte dans le choix d'une unité de modélisation de la parole parmi eux en trouve la taille du vocabulaire utilisé. Si en prend un vocabulaire de petite taille (<1K mots), l'unité de modélisation utilisée et le mot mais il est nécessaire d'avoir un grand nombre de mots présents dans le corpus d'apprentissage. Si la taille du vocabulaire et moyenne (>10K mots) ou grande (<10K mots), il devient difficile ou impossible de modéliser ce grand nombre de mots. Pour résoudre ce problème, en fait appel à une unité de modélisation plus petite, qui est le sous-mot (subword), et on note selon Joseph Mariani que « plus l'unité est petite, plus elle sera présente dans le corpus d'apprentissage, et meilleurs seront les paramètres du modèle » [32]. Cependant, le phonème qui est le plus petit élément dans le signal vocal, est aussi l'unité de modélisation largement utilisée, vu la disponibilité des standards et des règles de passage du phonème vers le mot, qui favorisent son utilisation pour la modélisation de la voix. Entre temps, pour utiliser les phonèmes comme unité de modélisation, il faut utiliser un modèle lexical (dictionnaire) pour le passage des mots vers leurs transcriptions phonétiques. L'apprentissage et la reconnaissance se font

au niveau phonétique. À la fin du processus, les séquences phonétiques détectées sont reconverties en séquence de mots.

Du point de vue modélisation phonétique, il y a deux types : mono phonème indépendant du contexte et phonème dépendant du contexte. Dans le premier, chaque phonème est indépendant par rapport aux phonèmes adjacents, et ne tient pas en compte du problème d'articulation qui influe sur la prononciation de ce phonème. Ainsi, l'utilisation de mono phonème indépendant du contexte dans le processus de reconnaissance ne donne pas des résultats encouragent [76]. Dans le deuxième type qui est la modélisation en phonème dépendant du contexte, en prend en considération les phonèmes adjacents (les tri-phonèmes qui sont largement utilisés dans les systèmes de reconnaissance de la parole, le précèdent et le suivant). Ainsi, on trouve des modèles qui intègrent des informations du contexte plus élargi (les deux précédents et le deux suivants), appelé cinq-phonème (quinphones) [77], [78]. Cependant, il y a deux possibilités pour segmenter des mots d'une phrase en tri-phonèmes. Soit, on permet le croisement des tri-phonèmes entre deux mots : tri-phonème croisé (Cross word triphones) ou, on ne permet pas le croisement entre deux mots. Donc, on utilise les bi-phonèmes pour marquer le début et la fin d'un mot.

Par conséquent, l'utilisation des modèles tri-phonèmes accrois le nombre d'unité acoustique à modéliser. Par exemple pour les 44 phonèmes de l'anglaise, le nombre de tri-phonèmes croisés est environ 100000. D'où, il est très difficile d'avoir les corpus nécessaires pour le processus d'apprentissage. Pour résoudre ce problème, des techniques d'attachement (tying) ou de segmentation (clustring) sont utilisées, [79], [28]. Leurs principes sont de chercher un ensemble d'unités qui partagent les mêmes valeurs spectrales. En apprentissage, tout l'ensemble est utilisé pour l'estimation des paramètres partagés. L'approche la plus utilisée est appelée « state clustring » [79] ou la distribution d'émission est partagée pour tous les éléments de l'ensemble, comme la montre la figure 8.

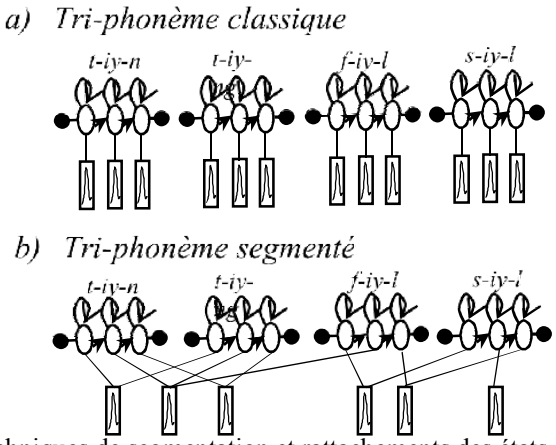


Figure 7 Illustration des techniques de segmentation et rattachements des états. (a) l'état initial où chaque état a sa propre distribution. (b) Montre les états après attachements et segmentation ou quelques états partagent les mêmes distributions. D'après [79]

Topologie de modèle

L'un des défis majeurs de l'utilisation des HHMs., est le choix optimal du nombre d'états et les transitions entre eux. Sachant que le flux vocal est un signal temporel non stationnaire, l'utilisation de la topologie dite gauche-droite, qui permet de capturer mieux la dynamique temporelle. Un HMM gauche-droite peut être décrit comme un automate probabiliste à états finis comportant deux processus : un processus caché -non observable- responsable des changements d'état et un processus responsable des émissions. La transition du modèle dans un état au cours du second processus génère une observation. Un exemple de modèle de Markov gauche-droite à trois états, est présenté en Figure 9.

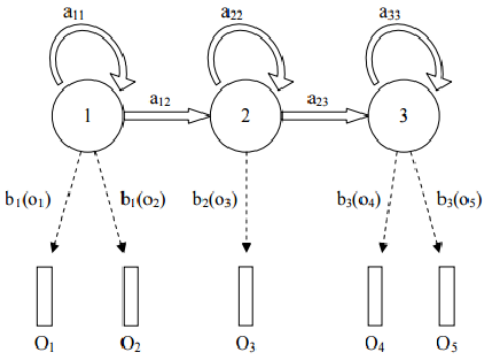


Figure 8 HMM gauche-droite à 3 états

La réalisation d'un processus d'un HHM se traduit par l'existence d'une séquence $Q = (q_0, \dots, q_T)$ d'états de l'automate. Le processus d'émission du modèle du HMM associe à la séquence Q une séquence $O = (o_0, \dots, o_T)$ de T observations.

D'une façon formelle, il peut être décrit par :

- Un ensemble fini d'états $Q = \{q_1, \dots, Q_N\}$;
- Les probabilités des transitions $a(i,j) = P(q_j/q_i)$, qui peuvent êtres écrites sous forme d'une matrice $A[N \times N]$;
- Un ensemble χ des symboles d'émission possibles x (discret ou continu) ;
- Les probabilités d'émission $b(t,j) = P(x_j/q_t)$;

Modèle de distribution de l'observation

Généralement, les probabilités d'émission sont soit des valeurs discrètes obtenues à l'aide des techniques de la quantification vectorielle, ou sont des valeurs continues calculées avec une fonction de densité continue. Les observations discrètes sont rarement utilisées, vu la nature continue et multidimensionnelle du signal vocal. Cependant, les observations continues avec des fonctions de densités gaussiennes ou même avec les réseaux de neurones sont largement utilisées. La majorité des systèmes de reconnaissance vocale utilisent des observations générées par des mixtures gaussiennes, qui permet de modéliser l'observation et estimer la probabilité d'émission pour chaque état par :

$$b_i(o) = \sum_{k=1}^{M} C_{jk} \mathrm{N}(o, \mu_{ik} \Sigma_{ik})$$

Tel que o est le vecteur d'observation cible, $\mathrm{N}(o, \mu_{ik}\Sigma_{ik})$ est une fonction de gaussienne simple avec le vecteur moyenne μ_{ik} et la matrice de covariance Σ_{ik} pour l'état i, M représente le nombre de gaussienne et C_{ik} est le poids de la k^{éme} gaussienne.

Certains états peuvent partagés des densités d'observation similaires. Afin d'améliorer l'estimation des paramètres, les distributions des états similaires peuvent être attachées ou regroupées selon une règle. Parmi les techniques utilisées pour sélectionner les états à attacher on cite les arbres de décision à base des modèles de tri-phonèmes [79].

Un arbre de décision est un arbre binaire dans lequel une question est attachée à chaque nœud. Les questions sont liées au contexte phonétique adjacent à gauche ou à droite.

Un arbre de décision est construit pour chaque phonème pour regrouper tous les états correspondants de tous les tri-phonèmes. Chaque groupe d'état (dans les nœuds feuilles de l'arbre) formera un seul état. Comme présenté à la figure 10, la première question dans l'arborescence (nœud racine) est : Est-ce le phonème contextuel gauche est nasale ? Un arbre de décision est construit pour chaque phonème pour regrouper tous les états correspondants

de tous les tri-phonèmes. Chaque groupe d'état (dans les nœuds feuilles de l'arbre) formera un seul état.

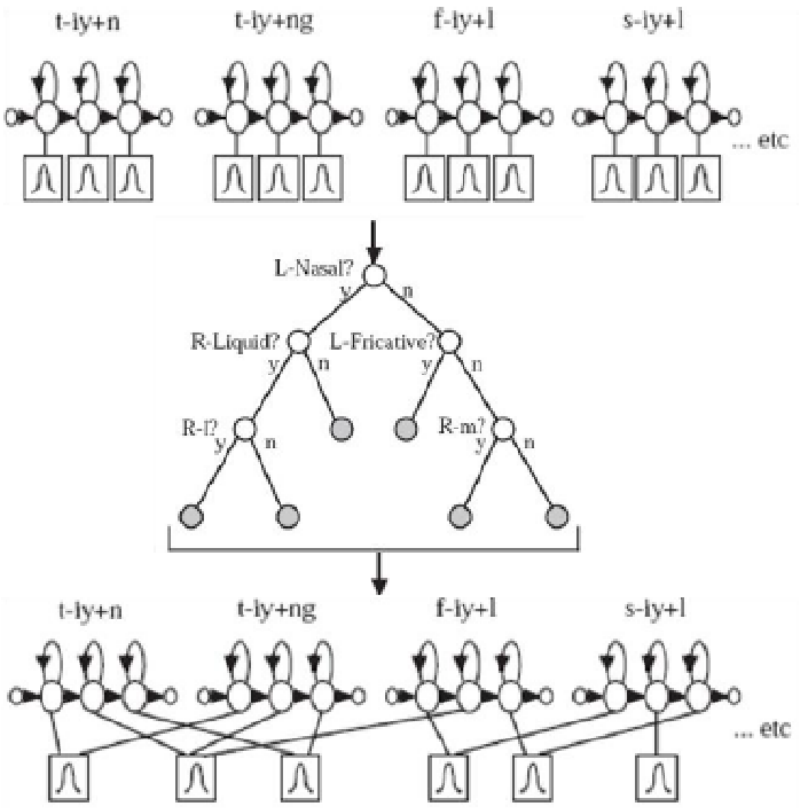


Figure 9 Exemple des états attachés d'un HMM avec un arbre de décision phonétique

Entre autres, les réseaux de neurones artificiels (ANN) peuvent être une autre variante pour estimer les probabilités d'émission. Dans la littérature, on note que la sortie du classifieur ANN peut être interprétée comme l'estimation de la probabilité a posteriori de la classe de sortie relative aux données d'entrées [33]. Ensuite, la probabilité de sortie d'état peut être estimée en appliquant la règle de Bayes aux sorties [34]. De nombreux systèmes de reconnaissance utilisent l'approche hybride HMM / ANN [7].

Estimation initiale

Il n'existe aucune approche analytique permettant d'obtenir l'ensemble des paramètres optimaux assurant un maximum global de MV. Cependant, l'algorithme itératif de Baum Welch développé à la fin des années 60, [11], tend à un maximum local, il est important de sélectionner une estimation initiale la plus proche possible du maximum global de la fonction de vraisemblance.

Dans le cas des HMM discrets, on peut utiliser des estimations initiales aléatoires ou uniformes [12]. Cependant, lorsque les observations sont dans l'espace continue, on peut appliquer des méthodes plus développées pour calculer l'estimation initiale. Tel que la

technique de segmentation des données avec k-means afin d'extraire les paramètres de la fonction de densité de probabilité pour chaque état (la moyenne et la covariance) [45], et la technique de démarrage plat, qui initialise toutes les probabilités des transitions équiprobables et initialise les paramètres de densité pour chaque état avec les paramètres estimés pour ce modèle [35]. Les modèles de mixture de gaussiennes peuvent être estimées avec une division incrémentale des densités gaussiennes pour chaque itération.

Modélisation du langage

Le modèle de langage nous permet de caractériser, de capturer et d'exploiter les régularités de la langue traitée [35]. Deux types de modèles sont principalement utilisés : l'approche à grammaire formelle mises au point par des experts en linguistique et les approches probabilistes qui sont des modèles stochastiques qui utilisent un corpus pour estimer des probabilités d'une suite de mots d'un langage de manière automatique.

L'approche formelle

Les approches à syntaxe formelle sont des ensembles de règles exprimées à l'aide d'une grammaire non contextuelle (GNC). Leur traduction en graphes détermine les séquences de mots valides. Cependant, la génération d'un ensemble de règles décrivant un langage est un processus long et difficile qui nécessite l'intervention des experts en linguistique. L'inconvénient le plus évident de ces approches est l'impossibilité de décrire de façon exhaustive une langue au travers d'une grammaire à base de règles [10]**.** Aussi, la limitation importante des approches formelle réside dans leur incapacité à reconnaître des messages grammaticalement approximatifs ou même faux comme la parole spontanée avec ses faux départs, hésitations…

L'approche probabiliste

Le principe d'un modèle de langage probabiliste est de capturer les régularités parmi les suites de mots. Cette modélisation consiste à donner la probabilité d'un mot à partir de la séquence de mots qui le précède. La probabilité d'une séquence de N mots, $P(M)$ est le produit des probabilités conditionnelles d'un mot sachant les mots qui précèdent. Elle s'écrit comme suit

$$P(m_1 \dots m_N) = P(m_1) \prod_{i=2}^{N} P(m_i / m_1 \dots m_{i-1})$$

En pratique, la manipulation des probabilités de toutes les suites de mots possibles est irréalisable, et l'historique est tronqué à quelques mots. Les modèles trigrammes ou quadri grammes, qui correspondent respectivement à un historique de deux mots et de trois mots sont les plus utilisés.

Modèle n-grammes

Les modèles de langage probabilistes, qui reposent sur l'application de la règle des probabilités conditionnelles, définissent la vraisemblance d'une suite de mots $P(M)$ est approximée avec le modèle *n*-gram par la formule suivante :

$$P(M) = P(m_1\, m_2 \ldots m_N) \approx \prod_{i=1}^{N} P(m_i / m_{i-n+1} \ldots m_{i-1})$$

La valeur de $\boldsymbol{n}$ est un compromis entre la stabilité de l'estimation et sa justesse. Le tri-gram (n=3) est un choix commun pour les grands corpus d'apprentissage. Alors que, le bi-gram (n=2) est souvent utilisé avec des corpus de petite taille. Cependant, l'augmentation de la valeur de $\boldsymbol{n}$ accroitre la difficulté de l'estimation de la probabilité a priori. Cette probabilité peut être estimée par l'approche de fréquence relative.

$$P(m_i / m_{i-n+1} \ldots m_{i-1}) = \frac{F(m_{i-n+1} \ldots m_{i-1} m_i)}{F(m_{i-n+1} \ldots m_{i-1})}$$

Les fréquences sont estimées sur de grands corpus de texte. En principe, les n-grammes peuvent être estimés par les fréquences d'apparition. Le problème est que le nombre potentiel de *n*-grammes est la puissance N de la taille du vocabulaire (soit pour seulement 1.000 mots, 1 milliard de trigrammes distincts).

Lissage

Le lissage consiste à prendre de la masse de probabilité des n-grammes observés, pour donner une valeur non-nulle aux probabilités des n-grammes non-observés ou peu observés. L'une des techniques de lissage la plus utilisée est la technique dite de Kneser-Ney modifiée [22]. Avec cette technique, les probabilités des n-grammes peu observés sont estimées comme avec les autres techniques de lissage, en faisant un repliement («backoff ») sur un historique d'ordre moins grand. Pour un trigramme par exemple, le bigramme puis l'unigramme si nécessaire sont utilisés. L'originalité de la technique Kneser-Ney modifié est de ne pas prendre la même distribution de probabilité pour les ordres plus petits que n. Au lieu de prendre la fréquence de l'historique d'ordre n−1 à savoir m_{i-n+1}^{i-1} , c'est le nombre de contextes différents dans lesquels se produit m_{i-n+1}^{i-1} qui est consulté. L'idée est que si

ce nombre est faible alors la probabilité accordée au modèle d'ordre (n-1) doit être petite et ce, même si m_{i-n+1}^{i-1} est fréquent. Ainsi le biais potentiel introduit par la fréquence de l'historique est évité [30].

Evaluation des modèles de langage

Une question primordiale est de savoir comment deux modèles de langage peuvent être comparés en termes de performances d'un système de reconnaissance global. La réponse ç cette question est que les modèles de langage se distingueront entre eux par les hypothèses choisies pour réduire la complexité combinatoire et améliorer leur capacité de généralisation. Par l'estimation de la probabilité de séquences de mots qui ne font pas partie du corpus d'apprentissage du modèle. La probabilité d'un texte $M = m_1 m_2 \dots m_n$ appelée « vraisemblance » en français, « likelihood» en anglais et notée lh.

$$lh(M) = \hat{P}(m_1 m_2 \dots m_n)$$

Plus la vraisemblance est grande, plus le modèle est capable de prédire les mots contenus dans le corpus. Le chapeau de $\widehat{P}$ est là pour rappeler que nous ne pouvons qu'estimer cette probabilité par des modèles (les modèles n-grammes en général) [30].

La grandeur la plus utilisée pour caractériser les performances d'un modèle de langage est la perplexité, souvent notée pp, définit comme suit :

$$pp = {}^{1}/_{\hat{P}(m_1^N)^{1/N}}$$

Elle est équivalente à la vraisemblance mais fait intervenir une normalisation sur le nombre de mots du corpus de test. Plus la probabilité de la séquence de mots est grande, plus la vraisemblance est grande, plus la perplexité est petite. Ainsi, maximiser la vraisemblance est équivalent à minimiser la perplexité.

Une interprétation courante de la perplexité consiste à voir cette grandeur comme le facteur de branchement moyen pondéré d'une langue. Le facteur de branchement moyen d'une langue est le taux moyen de mots qui peuvent suivre un mot donné de manière équiprobable. Une perplexité de 200 signifie qu'en moyenne, chaque mot du texte sur lequel elle a été mesurée, peut être suivi par 200 mots distincts de manière équiprobable.

Enfin, voici deux remarques à prendre en considération lorsque l'on compare des modèles de langage :

- Une réduction de perplexité (ou une augmentation de vraisemblance) n'implique pas toujours un gain de performances d'un système de reconnaissance,

- En général, la perplexité de deux modèles n'est comparable que s'ils utilisent le même vocabulaire. Sinon, il faut utiliser une perplexité normalisée qui simule un nombre de mots identique.

Bien que des modèles de langage avec des mesures de perplexité qui diminuent tendent à améliorer les performances d'un système de reconnaissance, il existe dans la littérature des études qui reportent des diminutions importantes de perplexité n'ayant peu ou pas apporté de gain de performance [37].

Techniques de décodage

Le module qui effectue en pratique la reconnaissance dans le système de reconnaissance de la parole, est appelé décodeur, qui a pour rôle de chercher dans un espace d'hypothèses très grand, le meilleur chemin qui amène à la séquence de mots la plus probable. En général, un module de décodage dans un SRAP passe par deux étapes primordiales, la première consiste à l'identification des parties du signal vocale qui sont effectivement des termes parlés. La deuxième consiste au processus de décodage lui-même. Il existe de nombreuses stratégies de décodage, et le choix d'une technique ou autres dépendent des contraintes de temps réel et de taille du vocabulaire utilisé [30]. Le point commun entre ses différentes stratégies est le compromis nécessaire entre la taille des modèles (en nombre de paramètres), et les réductions de l'espace de recherche (élagage ou « pruning » en anglais) [42].

Espace de recherche

La définition de l'espace de recherche est effectuée par la combinaison du modèle de langage avec le modèle acoustique pour toutes les séquences possibles des mots du vocabulaire. Généralement, l'espace de recherche est modélisé par des HMM. La complexité de décodage est liée étroitement à la complexité de l'espace de recherche.

Décodage avec l'Algorithme de Viterbi à une passe

Cet algorithme a pour but de trouver la séquence d'états la plus probable ayant produit la séquence mesurée. Dans un SRAP à grand vocabulaire, L'espace de recherche de l'algorithme de Viterbi doit contenir tous les chemins possibles des phonèmes construisant les différents mots de l'automate du modèle du langage utilisé. Il peut être représenté comme une arboréssance qui est établie dynamiquement au besoin, où l'on partage les modèles avec

les différentes hypothèses qui commencent par les mêmes ordres des sous-mots (triphonème).

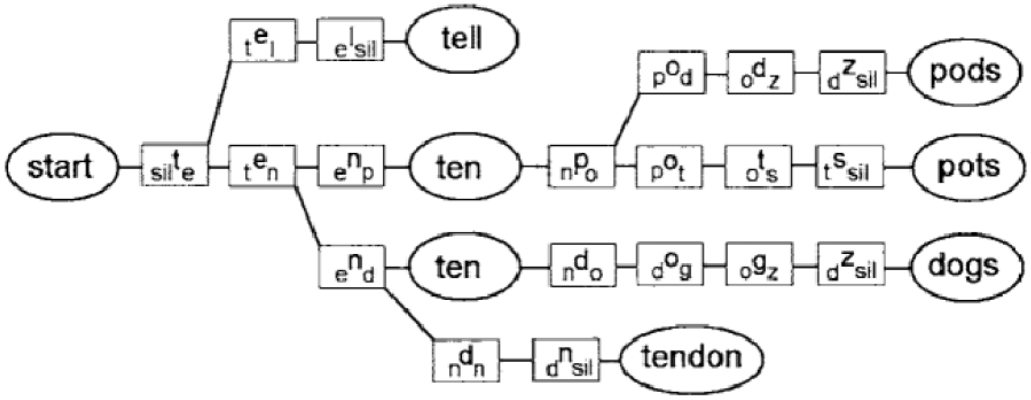


Figure 10 Un fragment très petit d'un réseau pour décodage dans une automate de modèle de langage triphonème. Ce fragment présente la séquence « ten pots » avec quelques chemins possibles dans le réseau de décodage. Noter que différents nœuds dans le réseau, selon si le mot suivant est des « pots » ou des « dogs », représentent le mot « ten ».

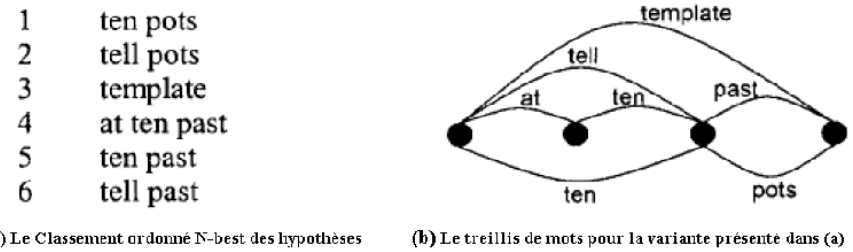
Cependant, l'élagage efficace de l'arboréssance est primordial pour les systèmes à base de LVCSR. On utilise la recherche vectorielle comme stratégie habituelle, par lequel à un instant t de la séquence recherchée, on élimine tous les chemins qui ont une probabilité qui n'appartient pas aux probabilités de meilleur chemin (best-scoring path). Ainsi il est possible de concentrer la recherche sur les segments étroits des séquences possibles [29].

Entre-temps, les contraintes de langue cernent à limiter l'ensemble de mots qui sont probables à n'importe quel point donné dans une expression. Il est donc avantageux d'employer le modèle de langue pour élaguer les possibilités peu probables aussitôt que possible en décodant une expression, mais dans Viterbi conventionnel la probabilité de décodage ne sera connue qu'à la fin de la séquence cible. Cependant, si on garde les probabilités des modèles triphonèmes des mots possibles, il est possible d'employer ce dernier pour élaguer les hypothèses peu probables.

Décodage avec l'Algorithme Viterbi à passes multiples

Afin d'accroitre la qualité de reconnaissance avec le décodage de Viterbi à un seul passe est de l'utiliser on plusieurs passe (multi passe). À cet effet, l'idée est d'exécuter un décodage simple pour extraire un nombre limité d'hypothèses probables, puis on exécuter un autre décodage plus approfondi pour trouver l'hypothèse la plus probable. Par exemple, la première passe utilise une modélisation en tri-phonème simple pour les mots avec un

modèle de langage bi-gram. Tandis que, la deuxième passe utilise une modélisation en tri-phonème interconnecté pour les mots avec un modèle de langage tri-gram.



(a) Le Classement ordonné N-best des hypothèses (b) Le treillis de mots pour la variante présente dans (a)

Figure 11 Les alternatives possibles qui peuvent être générés pour un court énoncé [29]

La sortie de la première passe de décodage est souvent exprimée par classement ordonné *N-best* de toutes les séquences possibles du mot, ou par un réseau de graphe ou treillis de mots qui contient toutes les possibilités probables. La figure 11 présente un exemple de N-best séquences possibles et le treillis de mots correspondant.

Dans la littérature, plusieurs algorithmes peuvent être adaptés pour donner la liste des N meilleures hypothèses [56], [45], [52]. L'algorithme de décodage à pile « stack-decoding » fournit une phrase complète en choisissant la meilleure hypothèse partielle, au lieu de sélectionner uniquement la meilleure hypothèse partielle, l'algorithme pourrait choisir, selon la même fonction objective, les N meilleures hypothèses. Un autre algorithme qui peut-être étendu est l'algorithme « Forward- Backward », qui utilise une recherche approximative temporelle synchrone dans la direction vers l'avant pour faciliter une recherche plus complexe et coûteuse dans la direction arrière. On utilise le modèle acoustique ou de langage simplifié pour effectuer une recherche passe-avant « Forward » rapide et efficace, dans laquelle on stocke le score de tous les chemins partiels qui surmontent le seuil d'élagage défini. Ensuite, on effectue une recherche en arrière « Backward » pour générer la liste de N meilleurs hypothèses « N-best ». La recherche arrière donne un score élevé sur une hypothèse seulement s'il existe aussi une bonne séquence conduisant à la fin d'un mot à cet instant du temps.

Evaluation des systèmes de reconnaissance de la parole à LVCS

Mesure des erreurs

Lors de la reconnaissance vocale de la parole connectée, il existe trois types d'erreurs de reconnaissance : substitution (le mauvais mot est reconnu), la suppression (un mot est omis) et d'insertion (un mot supplémentaire est reconnu). Le taux d'erreur de mot (WER) est donné par :

$$WER = \frac{C(substitution) + C(suppression) + C(insertion)}{N}\%$$

Où N est le nombre total de mots dans le discours de test, et $C\ (x)$ est le nombre d'erreurs de type x. Cependant, on trouve dans la littérature d'autres métriques plus spécifiques comme celle introduite pour estimer la fidélité sémantique des transcriptions réalisées [62], pour des systèmes d'interprétation de dialogue, d'indexation...

Aussi, on trouve que pour évaluer la fiabilité de ces mesures statistiques, il convient de calculer un intervalle de confiance relatif au nombre d'échantillons et d'erreurs. Cet intervalle de confiance est calculé en considérant que l'apparition d'une erreur de reconnaissance sur un mot ou sa non-reconnaissance est associée à une variable aléatoire binomiale, dont la distribution dépend des couples (mot reconnu, mot prononcé) [31].

Chapitre 3 – Recherches actuelle sur STD

Introduction

En règle générale, l'indexation de la parole est effectuée sur la base de l'identité du locuteur ou du contenu ou les deux. Alors que la reconnaissance du locuteur est utilisée pour l'indexation de la parole basée sur l'identité du locuteur, des variantes de techniques de reconnaissance de la parole telle que la détection de mot-clé et les STD sont utilisés pour l'indexation basée sur le contenu.

La détection de mots-clés consiste à trouver les occurrences phonétiques des mots prononcés dans le contenu des documents parlés. Néanmoins, le STD étend même en trouvant des termes (qui peuvent y aller d'un mot à une séquence de ces mots) dans les documents parlés. Cependant, la détection de mot-clé est considérée comme une partie de STD.

Dans ce contexte, on peut résumer les défis importants dans le contexte des STD dans les points suivants :

1. L'amélioration de la performance de processus de détection.
2. L'amélioration de la qualité de l'indexation.
3. L'accélération du temps de recherche.
4. L'amélioration du pouvoir de manipulation des vocabulaires sans restriction, y compris les mots hors vocabulaire (HV).
5. Le traitement des différentes variantes de prononciation acoustiques.

Dans les littératures, on trouve que ces défis ont été abordés en utilisant des approches différentes, qui seront décrites dans les sections suivantes.

Approches générales de STD

La détection de termes parlés (STD) est considérée comme une variante du problème de la reconnaissance de la parole. Néanmoins, un bon nombre d'approches destinées à la reconnaissance de la parole trouvent leur applicabilité dans les systèmes STD avec des modifications appropriées. Dans les sections qui se suivent, nous présenterons une classification générale des différentes approches utilisées pour la détection de termes parlés,

et nous nous concentrerons sur les trois premières approches supervisées : approche par détection de mots clés ; approche à base de LVCSR ; approche à base de sous mot.

1. **Approches supervisées**

Dans le volet des approches qui sont basées sur la notion de l'apprentissage et de classifications supervisées, on trouve :

- Les approches basées détection de mot clé acoustique dans le contenu des documents parlés.
- Les approches basées sur les systèmes de reconnaissance à base de mot pour le contenu parlé.
- Les approches basées sub-word dans les systèmes de reconnaissance de la parole continue à large vocabulaire
- Les approches de détection des termes parlés à base des exemples de requêtes « *Query-by-Example : text based STD* »
- Les approches discriminatives de détection des mots clés.

2. **Approches non supervisées : QBE (Query-by-exemple en utilisant template matching)**

- Les approches basées sur l'alignement dynamique de modèle des trames « Frame based template matching ».
- Les approches sur la segmentation basé sur l'alignement dynamique des modèles « Segment based template matching ».

Dans les littératures, on trouve qu'une grande partie des recherches antérieures effectuées dans ce domaine sont basées sur la détection acoustique des mots clés proposée par *Rose* et ses collaborateurs [36]. On note aussi, que dans ce mémoire on a focalisé seulement sur les approches supervisées à base acoustique. Dans ce contexte, dans la section 3, on va présenter un résumé des recherches effectuées dans le domaine de la détection acoustique de mot-clé.

Beaucoup de systèmes STD actuelle utilisent les techniques de reconnaissance de la parole continue à large vocabulaire (LVCSR). Cette dernière, nécessite un apprentissage supervisé de ses modèles, qui sont souvent des modèles probabilistes stochastiques : HMM-GMM. Cependant, les systèmes STD peuvent être appliqués aussi sur la majorité des langues

humaines utilisées voire même les dialectes. Dans ce contexte, des algorithmes de décodage sont développés pour permettre de gérer les langues qui ne disposent pas des ressources suffisantes pour le processus d'apprentissage de ses modèles de représentation [37].

Même pour les langues bien dotées, les techniques de LVCSR souffrent de plusieurs limitations comme celles qui concernent les tâches STD. En plus de ces derniers, on trouve le problème de reconnaissance de mots hors vocabulaire (MHV) car ces mots ne sont pas traités par LVCSRs. Un autre problème avec l'approche LVCSR est que la précision pour les termes des concepts de haut niveau est faible et ceci due à l'effet du modèle de langage [38]. Cet aspect rend ces systèmes moins efficaces pour les tâches de STD dans les domaines pour lesquels un modèle de langage approprié n'est pas très riche pendant l'apprentissage. Ce problème a été abordée par les systèmes de détection de mots clés qui ont fait l'usage de systèmes de reconnaissance à base de phonétiques [39] ; [40] ; [41] ; [42] au lieu de système de reconnaissance à base de mots.

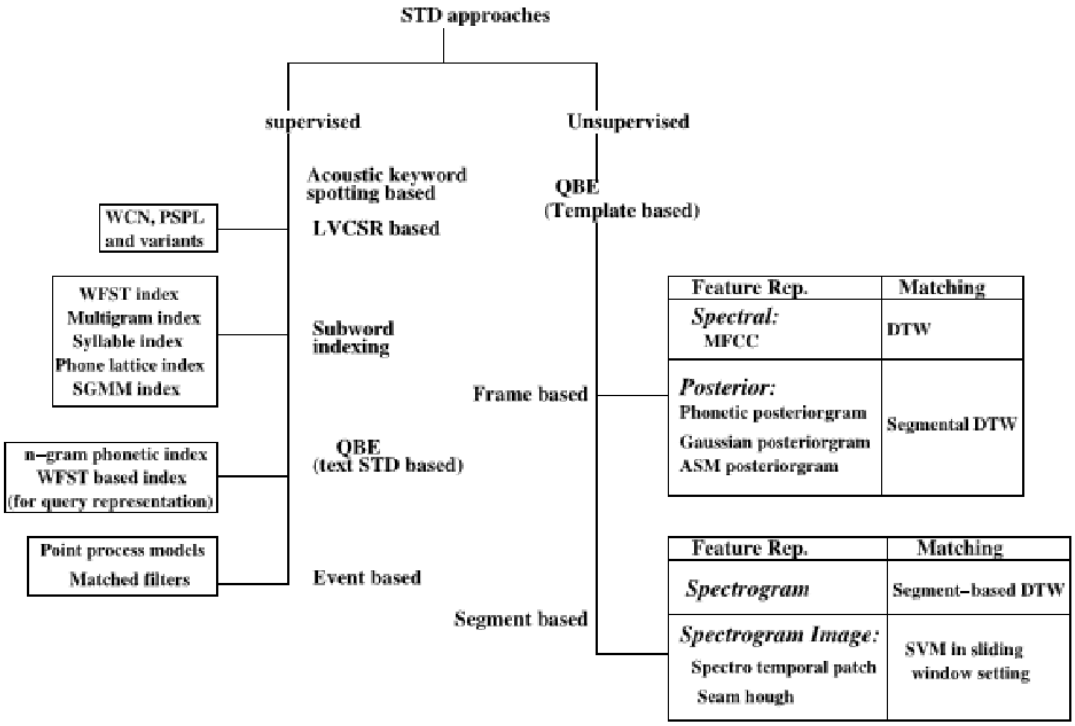


Figure 12 Taxonomie des approches de STD

Travaux existants sur les approches de STD

Détection de mots clés acoustiques

Le principe de base des systèmes de détection de mot-clé acoustique sur une architecture a deux composants : mots clés et mots poubelle. Donc, il utilise un réseau parallèle de modèles des mots clés et un réseau des modèles poubelles [43]. Dans ce contexte, le modèle d'un mot clé est construit par la concaténation des modèles de phonèmes constitutifs. Les modèles poubelle sont construits en utilisant des boucles de phonèmes. Chaque phonème est modélisé comme un *HMM/GMM* formés en utilisant des techniques probabilistes stochastiques.

Les réseaux de neurones sont également utilisés pour la modélisation de phonèmes [44]. Le score de vraisemblance (log-likelihood) correspondant aux résultats obtenus est calculé en utilisant les mots clés avec les modèles poubelles en arrière-plan « *background filler model* ». Des variantes pour l'évaluation du score de vraisemblance sont utilisées pour la décision sur les mots clés obtenus (accepté ou rejeté).

L'apprentissage des modèles HMM / GMM est généralement fait en utilisant des techniques d'apprentissage impliquant la maximisation de probabilité. Cependant, on trouve dans ces travaux que l'objectif d'apprentissage vise à maximiser la qualité de la transcription de paroles et non pas celle de la performance de détection de ces mots clés [45]. Cette question est abordée par des approches d'apprentissage discriminant. Ces approches maximisent au cours d'apprentissage, les différents critères qui ont un impact direct sur la performance de la détection des mots clés.

Cependant, l'approche proposée dans les travaux de Sukkur [46] vise à maximiser le rapport de vraisemblance (log-likelihood) entre les modèles de mots clés et les modèles poubelles pour les énoncés de mots clés et le minimiser sur un ensemble de fausses alarmes générées par les modèles des mots clés. En revanche, Sandness ont proposé d'appliquer la technique de minimum classification erreur (MCE) au problème de détection de mot-clé [47].

D'autre part, les approches proposées dans les travaux de [48] et [34] combinent des différents modèles de détection des mots clés à base des modèles HMM. Dans le premier travail, ils ont utilisé des réseaux de neurones pour combiner les rapports de vraisemblance (log-likelihood) des différents modèles ; tandis que dans la deuxième, ils ont utilisé les

SVMs pour combiner la moyenne des différents scores de rapport de vraisemblances (log-likelihood) des phonèmes.

Aussi, dans les travaux de Keshet, on trouve qu'ils ont proposé un algorithme d'apprentissage afin d'optimiser directement le critère facteur de mérite (FOM) généralement utilisé pour évaluer les performances des systèmes de détection de mots clés [49].

Une limitation majeure des systèmes acoustiques basés sur l'approche détection de mots clés est la difficulté rencontrée dans le traitement de nouveaux mots clés. Le système doit décoder l'énoncé cible avec la nouvelle liste de mots clés une fois de plus, chaque fois qu'un nouveau mot est entré. Il en résulte un temps de recherche excessivement élevées. Cette limitation est traitée dans les systèmes STD basé sur la technique du LVCSRs et de reconnaissance de sous-mots comme décrit ci-après.

STD utilisant LVCSRs

On trouve dans les littératures des efforts de recherches considérables dans le domaine de détection des termes parlés dans les ressources vocales. Ils ont porté sur l'extension des techniques de récupération d'informations disponibles pour le texte aux documents parlé. Certains d'entre eux sont décrits dans [50]. Ils ont construit un système à base de LVCSR et il est utilisé pour générer la transcription (niveau de mot) correspondant à la parole d'entrée. Ceux-ci sont ensuite indexés à l'aide des techniques de récupération d'informations disponibles pour le texte. Ces indices sont recherchés pour la présence de termes de la requête. Cependant, souvent la transcription du mot généré par 1-Best de l'LVCSR contient des erreurs qui affectent la performance du système STD. Par conséquent, l'utilisation des arbres de mots qui sont appelés souvent : les treillis de mots *« words lattices »* dans le processus de l'indexation au lieu de la sortie 1-Best du LVCSR est fortement sollicité. Les treillis de mots sont des graphes acycliques dirigés. Chaque noeud dans le treillis est associé à un *« timestamp »*. Chaque branche (u, v) est marquée avec un mot ou une hypothèse de phonème et la probabilité apriori qui est la probabilité du signal délimité par les timestamps des nœuds u e tv, qui forme l'hypothèse.

Une représentation similaire mais plus compact d'un treillis de mot est appelé les réseaux de confusion des mots « Word Confusion Network – WCN » [51] ; [52]. Chaque branche (u, v), est marquée avec une hypothèse de mot et sa probabilité postérieure, qui est la probabilité du mot donné par le signal. La construction d'un WCN est basée sur des arcs

de mots. Tous les mots arcs qui se chevauchent dans le temps sont regroupés dans des chemins respectifs, quelles que soient les positions de ces arcs. Ainsi les WCNs fournissent un alignement strict dans le temps de tous les mots dans le treillis.

Cependant, on trouve les travaux de Chelba [53]. Ils ont proposé une représentation plus compacte que celles de treillis de mot et des réseaux de confusion. Elle est définie par le calcul des probabilités postérieur des positions d'un mot dans le treillis des mots, ils ont appelé PSPL « *Posterieur Specific Position Lattice* ». Le PSPL calcule la probabilité postérieure d'un mot W à une position spécifique dans un treillis. Tous les chemins dans le treillis sont énumérés, chacun avec son propre poids de chemin. La probabilité postérieure d'un mot donné à une position donnée est calculée en additionnant tous les poids de chemin qui incluent le mot indiqué à une position indiquée et puis divisé par la somme des poids dans le treillis. La figure 13 montre un treillis de mot et sa représentation correspondante de PSPL et de WCN.

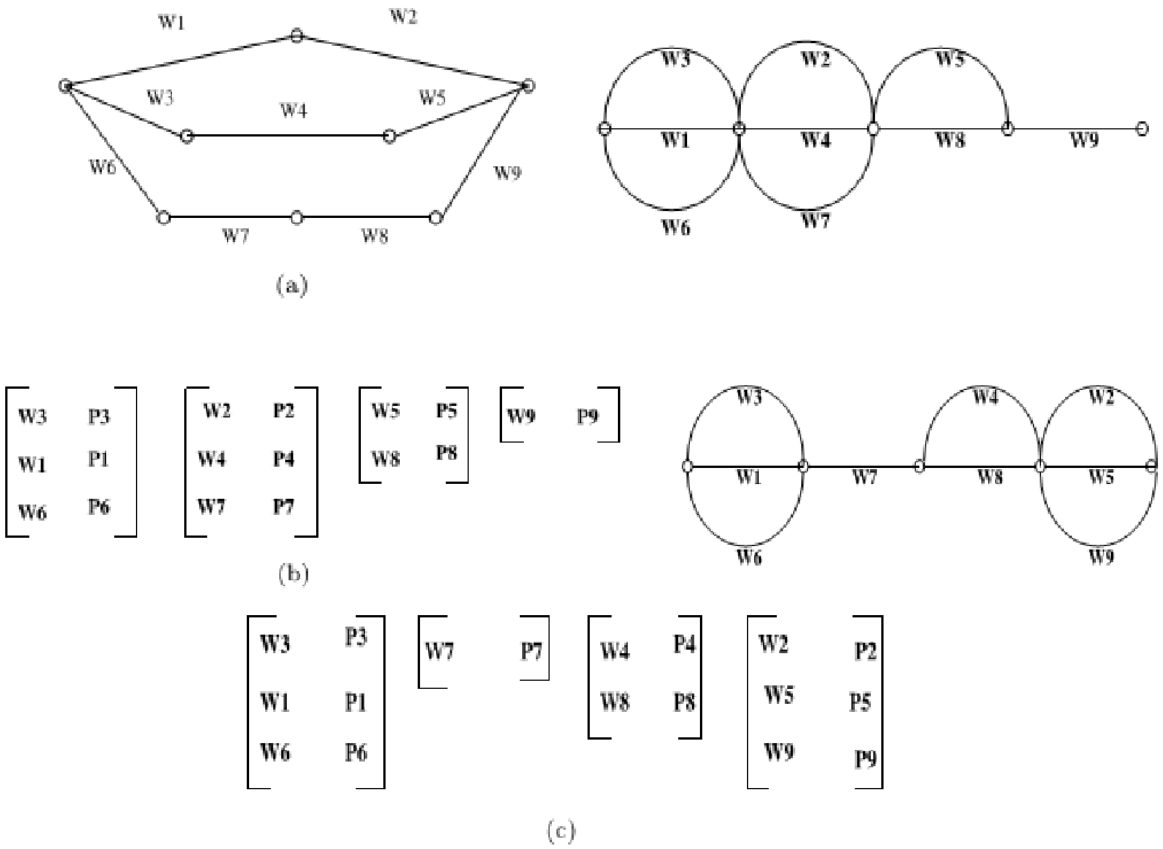


Figure 13 (a) Treillis de mot avec sept mots et leurs (b) PSPL et (c) WCN correspondants respectivement. Les W_i représentent des mots-clés et $p_i{}' s$ représentent les probabilités postérieures correspondantes liées à chaque mot.

Les treillis de mot et ses variantes ont été employés avec succès pour améliorer le taux de détection des mots de vocabulaire *« In vocabulry (IV) »*. Cependant, ils ne peuvent

pas traiter les mots hors vocabulaire (OOV). Une comparaison détaillée des performances de PSPLs et WCNs était discutées dans [54]. On voit que le PSPLs toujours donne une meilleure performance que WCNs, mais exige plus d'espace pour stocker les index. L'utilisation des unités de subword est également explorée pour améliorer la technique à base de mot de PSPL/WCN à PSPLs à base de subword (S-PSPL) et WCNs à base subword (S-WCN). On le constate que S-PSPLs/S-WCNs, rapportent toujours une précision moyenne de performance (MAP) bien mieux pour les deux types de requête (OOV et IV) tout en consommant beaucoup moins d'espace mémoire que PSPLs/WCNs à base de mot.

STD utilisant des systèmes de reconnaissance à base de sous mot (subword)

Les approches dans cette catégorie impliquent l'utilisation des index avec différentes unités de sous-mots tels que des n-grammes, des phonèmes, des multi-grammes, des syllabes, des segments ou des représentations de treillis des unités précédemment mentionnées [55] ; [56]. Les index sont établis sur des transcriptions d'un système de reconnaissance approprié de sous mots en utilisant des techniques existants de reconnaissance et récupération de texte [57]. Les transcriptions phonétiques d'un document parlé sont obtenues par un système de reconnaissance à base de phonèmes. Les transcriptions de niveau de phonèmes sont alors employées pour obtenir des unités de sous mots de complexité variable en matière de leurs longueurs. On l'observe qu'en termes de précision moyen (MAP), la meilleure performance est obtenue par l'utilisation d'index de tri-phonèmes (valeur de MAP est de 0.86) ; suivi par 5 multi-grams (valeur de MAP est de 0.81) [55]. L'utilisation des multi-grammes [58] comme unités de sous mots pour traiter des mots hors vocabulaire (OOV) est explorée dans [56]. L'impact des paramètres de multi-gramme ; à savoir sa longueur et le facteur d'élagage sur la taille de l'index et l'exactitude de STD sont étudiés. La performance de détection la plus élevée est obtenue avec des unités de multi-gramme de la longueur cinq. Cependant, le facteur d'élagage n'a pas beaucoup d'impact sur la performance de détection de phonème qui est autour d'une valeur de 50 %. Deux méthodes d'apprentissage de multi-grammes sont proposés, et qui ont amélioré la performance de phonème de 9 % et la performance de STD de 7 %. On le constate également que l'incorporation du modèle standard de langue de n-gramme sur des unités de multi-gramme est bénéfique, et avec le modèle de langue de trigramme donne la meilleure performance.

Les treillis phonétiques ont été les plus utiles pour accueillir à des taux d'erreur élevés dans les transcriptions et en permettant des requêtes de mots OOV [42] ; [33] ; [59] ; [60] ; [61]. Dans Saraclar (2004), ils présentent une amélioration de la précision de détection de

mot exprimée par F-scores pour les requêtes du type IV et OOV par l'utilisation des deux types de treillis : phonétique et mot. Dans ce contexte, trois stratégies de récupération ont été proposés. La première consiste à combiner les résultats après une recherche du mot et l'index phonétique. Dans la deuxième, ils suggèrent la recherche de l'index de mot pour les requêtes IV et l'index phonétique pour les requêtes de OOV. La troisième stratégie consiste à la recherche de l'index phonétique seulement si la recherche de l'index de mot ne renvoie aucun résultat.

Entre autres, le problème des requêtes hybrides est abordé dans [42]. Leur approche emploie deux index, un WCN pour le stockage d'index de mot et un index phonétique établi du treillis de phonème. Pour chaque unité de l'indexation (mot et phonème), les timestamps correspondant au début et fin de l'unité sont stockés. Pendant la recherche d'une requête sur un terme IV, une liste d'émission est extraite à partir de l'index de mot. Pour une requête sur un terme OOV, le terme est converti en une séquence de phonèmes en utilisant un maximum d'entropie de modèles N-gramme commun. La d'émission de chacun des phonèmes est alors extrait à partir de l'index phonétique. Pour une requête hybride de mot-clé impliquant les deux types de termes IV et OOV, l'index de mot pour les termes IV et l'index phonétique pour les termes OOV sont employés. Dans ce cas, les listes d'émission des termes IV extraites de l'index de mots sont fusionnées avec les listes d'émission des termes OOV obtenus à partir de l'index phonétique. Le résultat final de la requête hybride est obtenu par l'union ou l'intersection (AND ou OR) des résultats des requêtes individuelles en fonction de la relation entre les termes de la requête. Cette approche est plus performante que les méthodes basées uniquement sur l'index de mot ou l'index phonétique en réalisant une valeur de précision de 0,89 et une valeur de rappel de 0,83.

L'approche de Thambiratnam et Sridharan (2005) prétend de diminuer le taux d'omission et d'augmenter la vitesse de recherche pour la détection de mot-clé dans vocabulaire illimité. Il sert à la recherche de correspondance dynamique de treillis de phonèmes (DMPLS), une extension de la recherche du treillis de phonèmes (PLS) qui peut gérer l'insertion, la suppression et la substitution des erreurs d'un système de reconnaissance à base de phonèmes. L'approche utilise une représentation phonétique de la parole en utilisant un décodage Viterbi multi passe N-best. Le treillis est ensuite décodé pour la séquence de phonèmes constituant le mot-clé. Pendant la recherche, des scores de pénalités de coût appropriés sont imposées pour les erreurs de la reconnaissance de phonèmes, suivant. Le système proposé a atteint un taux d'omission de 10,2% et un taux de fausses alarmes de

18,5%. Ces résultats sont beaucoup plus bas que ceux des systèmes classiques de détection de mot-clé basés sur les HMM [62]. Cependant, la vitesse de recherche est d'environ 300 fois en temps réel. Il est à noter que la portée de la recherche est limitée à l'appariement de la représentation textuelle des mots-clés contre le treillis de phonèmes et n'inclut pas la génération du treillis depuis le contenu parlé.

Un aspect important lié à l'utilisation de treillis des ASR, est de construire l'index d'une manière efficace de façon à réduire au minimum les exigences de stockage et le temps de la recherche. Dans ce contexte, les treillis des ASR actuellement sont prétraités en utilisant la technique de Weighted Finite State Transducers (WFST) et les informations de synchronisation sont poussées sur l'étiquette de sortie de chaque arc dans le treillis. Les poids sont convertis en probabilités postérieures à travers une étape de normalisation supplémentaire [63]. Aussi, on trouve les travaux de Allauzen qui décrit un algorithme pour créer un index complet représenté comme un WFST qui mappe chaque sous-chaîne *x* à l'ensemble des indices dans les automates dans lesquels *x* apparaît [33]. L'index créé représente un transducteur de facteur qui est un index inversé des facteurs de parcours dans l'automate WFST. Pendant la recherche, la requête est représentée comme un transducteur pondéré et en utilisant une opération de composition unique avec l'index, l'automate contenant la requête est récupéré.

Dans [60], le facteur de transducteur « FT » tient à jour une entrée d'index unique pour toutes les occurrences d'un facteur dans un énoncé « utterence » et est adapté pour une tâche de récupération d'énoncé parlé. Une variante de la même structure d'index nommé comme timed transducer factor (TFT) qui stocke des informations de synchronisation sur le poids de l'arc est proposé par Can [64] pour la tâche STD. L'idée principale derrière TFT est que l'indice temporel est représenté par un WFST mappant chaque facteur *x* dans l'intervalle où x apparaît dans chacun des automates et les probabilités postérieures *x* qui se produisent réellement dans chaque automate pendant l'intervalle de temps correspondants. Les autres considérations pour FT sont conservées dans TFT. L'avantage de cette approche est que la complexité de la recherche est linéaire par apport à la longueur de la requête, et par conséquent elle est utile pour les requêtes longues. En outre, elle est très souple et prend en charge plusieurs autres fonctions, en plus des STD, tel que récupération de toutes les relations à états finis de l'indice, la recherche de relations complexes entre les mots de la requête et aussi la recherche de permutations arbitraires de mots de la requête sans modifier l'index.

Cependant, on trouve des approches qui exigent moins de ressources par rapport aux techniques décrites ci-dessus comme celle présentée dans les travaux de Garcia [65]. Dans cette approche, une petite quantité de données vocales (15 minutes de discours -mots-transcrit) est utilisée pour l'apprentissage d'un système de reconnaissance auto-organisationnelle, qui définit ses propres unités sonores pour un domaine spécifique. Les transcriptions sont utilisées pour l'apprentissage d'un convertisseur graphème-à-unité-sonore. La parole d'entrée est segmentée automatiquement et d'une manière non supervisée en fonction des discontinuités spectrales. Les segments ainsi obtenus sont ensuite modélisés en utilisant des modèles de segmentation des mixtures de gaussiennes (SGMMs). Le sous-ensemble des enregistrements vocaux pour lesquels il y a des transcriptions de niveau mot, et ils sont décodés en termes d'indices de SGMM. En utilisant les transcriptions parallèles, un modèle de multi-grammes commun est utilisé pour obtenir une cartographie probabiliste entre des séquences de lettres dans les transcriptions au niveau mot et des séquences d'indices de SGMM. Ce modèle est ensuite utilisé pour prédire la prononciation d'un mot-clé donné en termes des unités de SGMM, éliminant ainsi la nécessité d'un dictionnaire de prononciation. Enfin, une recherche de programmation dynamique, qui minimise la distance entre la prononciation prédite d'un mot-clé et la transcription automatique obtenu, est utilisée pour trouver les occurrences putatives du mot-clé. La Figure-de-Mérite moyenne obtenu (FOM) est de l'ordre de 0,34 en utilisant 15 minutes de données transcrites de l'ensemble d'apprentissage.

Autour des idées similaires, une autre approche est décrite dans [66], en vue d'améliorer la performance d'un LVCSR en utilisant des techniques non supervisées pour améliorer l'apprentissage des modèles acoustiques et modèles de langue. Le modèle acoustique résultant récupère 50% du gain par rapport à son équivalent supervisé. L'apprentissage de modèle de langage impliquant la multiplication des ensembles confidences de mots, pourrait atteindre une réduction de 2% du taux d'erreur (WER) sur la ligne de base et de 0,5% en valeur absolue sur les transcriptions non pondérées.

Cependant, dans toutes ces approches, le format de la requête est présenté sous la forme d'un texte. Par conséquent, ces méthodes supposent la disponibilité de l'expansion phonétique des mots clés, soit en utilisant des règles graphème-phonème ou par d'autres moyens.

Synthèse récapitulative des travaux existants

Dans cette section, on propose une synthèse récapitulative des travaux sur STD existants dans la littérature, cités dans les sections précédentes.

Travaux	Approches	Techniques utilisées	Corpus utilisé dans l'expérimentation	Critères d'évaluation	Remarques
Szöke et al 2005	Approche basée KWS	HMM - GMM	TRAP_NN0477	FOM = 64.46	
	Approche basée LVCSR	HMM-GMM Likelihood ratio confidence	ICST meeting	FOM = 66.95	
	Approche basée Subword	Treillis de phonème	TRAP_NN0477	FOM = 58.90	
Szöke et al 2008	Approche basée LVCSR	HMM ML	NIST STD06 dev-set VTS data	UBTWV = 63.0	
	Approche basée Subword			UBTWV = 65.0	
Grangier et al 2009	Approche basée KWS	Discriminative	TIMIT	AUC = 0.99	Le même travail de Keshet et al 2009
		HMM-GMM		AUC = 0.96	
Sandness 2000	Approche basée KWS	Discriminative MCE Training algorithme	JUPITER	ML_{INV} = 6.0 ML_{OOV} = 13.9	
				MCE_{INV} = 5.7 MCE_{OOV} = 13.6	
				KB_{INV} = 5.2 KB_{OOV} = 13.1	
Benayed et al 2003	Approche base KWS	HMM – GMM SVM	Speech Dat	EER = 26.7%	
Keshet et al 2009	Approche basée KWS	Discriminative	TIMIT	AUC = 0.99	Le même travail de Grangier et al 2009
		HMM-GMM		AUC = 0.96	
Hakkani et al 2003	Approche basée LVCSR	Segmentation de treillis et pivot alignement	$HMIHY^{SM}$	WER = 33.8%	
Mangu et al 2000	Approche basée LVCSR	Alignement de treillis HMM	Switch Board Speech	WER = 37.3	
			Broadcast news (DARPA Hub-4)	WER = 32.5%	
Chelba et Acero	Approche basée LVCSR	PSPL Treillis de phonèmes	MIT iCompus Data	MAP = 0.62	
				R-Precision = 0.58	
				WER_{PSPL}= 22%	
				$WER_{1\text{-best}}$ = 45%	

Mamou et al 2007	**Approche basée LVCSR**	**Discriminative**	**NIST06 Broadcast News**	**MAP_{OOV} = 0.48**	
	Approche Basé Subword	Fuzzy phonetic search			
Can et al 2009	Approche basée LVCSR	WFST CN	Broadcast news (DARPA Hub-4)	ATW = 0.453	
	Approche Basé Subword				
Vergyri et al 2006	Approche basée LVCSR	HMM 4-gram LM Cross word	Broadcast news	WER = 10.7%	
			CTS	WER = 17.0%	
			ICST meeting	WER = 37.0%	
Akbacak et al 2006	Approche basée LVCSR	À base de graphone Term Weighted Value (TWV)	English Broadcast News	WER = 14.8%	
	Approche bassée Subword			ATWV-(GL-TH) = 0.842	
				ATWV-(TERM-TH) = 0.889	
Wallace et al 2010	Approche basée KWS	Discriminative training	Fisher CTS Corpus	FOM = 0.606	
	Approche basée Subword				
Motlicek et al 2010	Approche basée LVCSR		Klewel Lecture Recording	EER = 3.6%	
				MTWV = 0.81	
Thambiratnam et al 2007	Approche basée KWS	Dynamic Match Lattices Spoting (DMLS)	TIMIT	Miss rate = 10.2	
			CTS	Miss rate = 13.9	
Parada et al 2010	Approche basée LVCSR		NIST STD06	ATWV = 0.8597	
			OOV Corp	ATWV = 0.415	
Chan et Lee 2010	Approche basée LVCSR	Frame-Based and Segment-Based DTW	Mandarin broadcast news	MAP = 48.6%	
Chen et Lee 2010	Approche bassée LVCSR	Feature Space Pseudo-Relevance Feedback	recorded lectures of a course offered in National Taiwan University	MAP_{SI} = 50.44%	
				MAP_{MLLR}=61.43%	
				MAP_{SD} = 75.17%	
		Partial index	DARPA BCN	f-measure = 86.1	

Alouazen et al 2004	Approche basée LVCSR & Subword	WFST	Switch Board Corpus	f-measure=60.8	
			Teleconferences	f-measure=52.7	
Saraclar et Sproat 2004	Approche basée Subword	Transducer factor	DARPA BCN	f-measure=86.0	
			Switch Board Corpus	f-measure=60.5	
			Teleconferences	f-measure=52.8	
Garcia et Guish 2006	Approche basée subword				
Can et saraclar 2011	Approche basée Subword	Timed Transducer Factor	Turkish broadcast news		
			Nist06 English broadcast news		
Pan et Lee 2010	Approche basée LVCSR	S-PSPL S-CN	Mandarin Broadcast news	AUC = 86.12%	
	Approche basée subword				

Tableau 3 Synthèse récapitulative des travaux existant sur STD

Synthèse

Dans cette partie, on a présenté les outils et les techniques nécessaires pour la construction d'un système d'indexation des ressources vocales. Ces derniers utilisent des modèles probabilistes stochastiques pour la modélisation des structures spectrales fréquentielles complexes du signal de la parole telle que les GMM/HMM. En revanche, pour le décodage et le décryptage de ces ressources, on utilise des algorithmes récursifs comme Viiterbi pour le parcours dans des arbres représentatifs du contenu du signal de la parole telle que les treillis de mots ou phonèmes, les réseaux de confusion et surtout les automates à états finis WFst.

Cependant, notre objectif dans ce mémoire est de proposer un modèle d'exploitation et d'indexation automatique pour les documents parlés. A cet effet, on cherche à utiliser les techniques de la littérature qui permettent de gérer les problèmes de passage de manuelle vers l'automatique ou tout simplement gérer le pouvoir de généralisation de ces derniers. Notre système est censé d'être capable de connaitre des termes référentiels de la langue du document parlé d'une façon efficace, mais le plus important qu'il doit être capable de gérer les nouveaux termes et les termes techniques du domaine des documents parlés cible. Cette dernière est traitée dans le domaine du traitement de la parole comme un phénomène de mot hors vocabulaire « OOV ». En pratique, ces mots hors vocabulaire ont une grande tendance d'être des indexes discriminant.

Dans l'étude bibliographique réalisée dans le troisième chapitre de cette partie, on constate qu'il existe deux philosophies pour la recherche et la détection de termes dans le contenu parlé. La première est basée sur la modélisation des termes elles-même pour accroitre le taux de détection. Certes, elle donne des résultats encourageants mais elle n'est pas évolutive car l'ajout des nouveaux termes mot clés nécessite reconstruction du modèle. Donc son utilisation pour la construction de notre système d'indexation automatique du contenu multimédia n'est pas prometteuse.

En revanche, deuxième philosophie repose sur l'utilisation des systèmes de reconnaissance LVCSR au profil de l'indexation et la recherche. Donc son objectif est l'amélioration de la qualité des systèmes de reconnaissance de la parole continue pour améliorer la détection et l'indexation de ces ressources multimédias. Le développement est réalisé autour de la capacité de gérer le maximum des termes de la langue même celles moins doté et de gérer les mots hors vocabulaire (OOV). Ils ont basé sur des termes de modélisation

plus élémentaire telle que les phonèmes. Cependant, ces unités élémentaires rendent la structure arbres représentatifs du contenu du signal de la parole telle que les treillis de phonèmes, les réseaux de confusion à base de phonèmes très complexe. Pour remédier à ce problème, on trouve l'utilisation des techniques Wfst dans les dernières recherches comme il est mentionné dans notre tableau de synthèse récapitulatif présenté précédemment. On note aussi que les plateformes de développements actuels tels que « KALDI » intègrent des bibliothèques de cette technique.

En conclusion, et vue les résultats obtenus dans les littératures, on a décidé d'utiliser la deuxième philosophie comme un noyau de notre modèle d'indexation automatique du contenu parlé. Cependant, on intègre des outils des modules de traitement linguistique et de gestion de connaissances pour soulever les problèmes d'omissions et de fausse d'alarmes des systèmes LVCSR ainsi que la détection de mots hors vocabulaire. Dans la partie suivante, on va présenter notre modèle d'indexation proposé.

Partie 3

-

Approche proposée et validation

Introduction

Jusqu'à présent, nous avons présentés les techniques et les stratégies utilisées dans la reconnaissance automatique de la parole ; ainsi que les méthodes existantes dans la littérature en vue de l'indexation en exploitant ces techniques de détection de mots clés.

Dans cette partie, nous présenteront tout d'abord la proposition d'un modèle d'indexation sur le contenu pour la recherche et la détection de mots clés dans un document multimédia parlé.

Par la suite, nous présenteront dans le chapitre 2 le framework Kaldi speech recognition toolkit, par une présentation brève de l'outil, un aperçu général des fonctions inclus dans l'outil (extraction de caractéristiques, modélisation acoustique, modélisation du langage, décodage …).

En fin nous termineront cette partie par une simulation de cette proposée avec des expérimentations sur des corpus libres tel que : *TED-LIUM corpus*, qu'il a été développé dans le contexte de la participation de l'LIUM à la campagne d'évaluation IWSLT 2011. Toutes ses données brutes (signaux acoustiques et ses annotations) ont été extrait du site TED et les transcriptions automatiques obtenues à partir d'un décodeur acoustique avec un alignement avec ses transcriptions. Afin d'obtenir des références alignées automatiquement pour les données audio. Ainsi que, on a utilisé IARPA *Babel Project,* qui a pour but de développer des méthodes pour construire la technologie de reconnaissance vocale pour un ensemble beaucoup plus vaste de langues que jusqu'à présent été abordée.

Chapitre 1 – Modèle d'indexation proposé

Problématique

Les progrès de la technologie numérique et de l'informatique sont à l'origine d'une croissance explosive dans l'utilisation des machines dans le domaine de traitement de l'information, qui provient d'un être humain et à être utilisé par un être humain.

De plus, ce développement crucial des technologies a généré un raz de marée d'informations multidisciplinaires et gigantesques qui se mesure en milliards de ressources (texte, audio, vidéo…).

Dans le cadre de notre travail, nous nous intéressons aux données numériques multimédia et en particulier les documents parlés, qui peuvent contenir des informations nécessaires et utiles. Cependant, afin d'exploiter ces ressources, ces derniers sont enrichis avec des annotations manuelles constituées de titre, mots-clés, nom d'auteur et un résumé décrivant le contenu, pour permettre une classification et indexation sommaire afin de facilité l'accès à ces informations. Néanmoins ces annotations, souvent réduites au strict minimum et elle dépend étroitement par les compétences de l'expert humain (l'indexeur). A cet effet, elles s'avèrent souvent insuffisantes pour classer les informations d'une manière efficace et retrouver les documents pertinents.

D'autre part, la nécessité de trouver des solutions d'indexation sur le contenu des ressources multimédia est fortement sollicitée et surtout lors de la manipulation des documents multimédia de grandes tailles. Dans ce contexte, on vise de trouver des solutions pour des systèmes de recherches qui fournissent des segments du document numérique et non son intégralité pour améliorer la pertinence des résultats de recherches

En outre, le volume des documents multimédias disponibles : nouveaux ou archivés et non annoté est très important, en plus de temps nécessaires de traitement de l'annotation rend l'indexation manuelle fastidieuse. C'est pour ces causes-là, qu'on fait appel à proposer une conception d'une démarche d'indexation semi-automatique ou même automatique sur le contenu, via les techniques de gestion du contenu numérique et de gestion des connaissances. A cet effet, notre problématique se résume dans les questions suivantes :

- ⇨ Comment accéder et gérer le contenu des ressources multimédia d'une façon efficace et précise ?

- ⇨ Comment accéder et repérer des segments spécifiques pour une requête donnée et non pas la totalité du document parlé ?
- ⇨ Comment extraire les indexe (ou des annotations) discriminants du contenu des ressources multimédia d'une façon automatique sans fait recours aux experts des domaines.
- ⇨ Comment remédier les problèmes liés au processus de reconnaissance automatique de la parole telle que les fausses alarmes, les fausses acceptations et les mots hors vocabulaire – les termes techniques du domaine ? On note dans ce travail, qu'on s'intéresse seulement sur les mots hors vocabulaire de point de vue documents et non pas du de point de vue requêtes.

Architecture proposée

Dans le cadre de réalisation d'une indexation sur le contenu des documents multimédias, bous rencontrons deux problèmes : comment connaitre les termes discriminants du domaine du document et comment repérée ces termes dans le document numérique en tenant compte de la complexité de sa structure.

Pour remédier le premier problème, nous proposant dans notre solution une démarche pour l'extraction de ces termes sachant qu'on vise de passer d'une simple tâche manuelle basée sur les connaissances des experts du domaine vers une tâche semi-automatique voir automatique a base des techniques de gestion de connaissance à base des ontologies légères telle que « WordNet ».

Tandis que pour le deuxième problème, on a utilisé dans notre système d'indexation les techniques du domaine de traitement de la parole dans le volet de la détection des mots clés à base de flux phonétique de ces termes. On note aussi, que la technique qui vient à l'esprit de réaliser une transcription intégrale du document multimédia n'est pas valide de point de vue charge de calculs et des erreurs de reconnaissances et surtout pour les termes étranges du vocabulaire qui sont appelés souvent mots *hors vocabulaire*. Dans ce contexte, La figure 16 présente un schéma bloc de l'architecture proposée pour l'indexation sur le contenu des documents multimédias.

Description de système d'indexation proposé

On a proposé une architecture d'indexation présentée dans la figure 15 sur trois phase : la première sert à trouver de termes d'indexation sur des segments du document multimédia cible susceptibles d'être des indexes pour la totalité du document multimédia cible. Ensuite, dans la deuxième phase, on va convertir ces indexes candidats sous forme phonétiques pour accéder au contenu physique du document multimédia à l'aide de la technique de détection des mots clés sur un flux phonétique. Dans la dernière phase, on valide les indexes phonétiques candidats dans le document cible pour tester le pouvoir de généralisation de ces indexes pour le document multimédia entier.

Phase 1 : pré traitement et extraction des index candidats

Dans cette phase de l'architecture de système d'indexation proposée, on propose d'utiliser des segments du document parlé. Dans ce contexte, sachant que les documents multimédias sont généralement structurés en paragraphes et pour la structure physique de document multimédia, on trouve les silences qui permettent de cerner les paragraphes du document. Entre temps, on suppose que le contenu de chaque paragraphe est homogène de point de vue sémantique. A cet effet, on propose d'utiliser des segments pour chaque paragraphe du document pour extraire les indexes candidats du document afin de surmonter le problème de localité vers la globalité et pour atteindre un pouvoir de généralisation acceptable.

Du point de vue technique, la boite à outils Kaldi, dispose de toutes les techniques disponibles pour réaliser une transcription de ses segments en utilisant un système de reconnaissance automatique de la parole continue à large vocabulaire.

Cependant, les résultats fournis par ce système sont très encourageants et surtout pour les langues bien dotées comme l'anglais (disponibilité des ressources pour l'apprentissage de modèle de langage 3-gram ou même 4-gram), ce qui réduit les problèmes d'erreurs de reconnaissance comme les fausses alarmes, les fausses acceptations et la détection des mots hors vocabulaire. Entre temps, pour traiter le problème des mots hors vocabulaire (OOV), on fait appel à un outil de traitement syntaxique et sémantique qui permet l'enrichissement

de ces termes par des termes de plus haut niveau en exploitant la ressource linguistique sémantique de l'ontologie légère « WordNet »[1] présenté dans la figure 15.

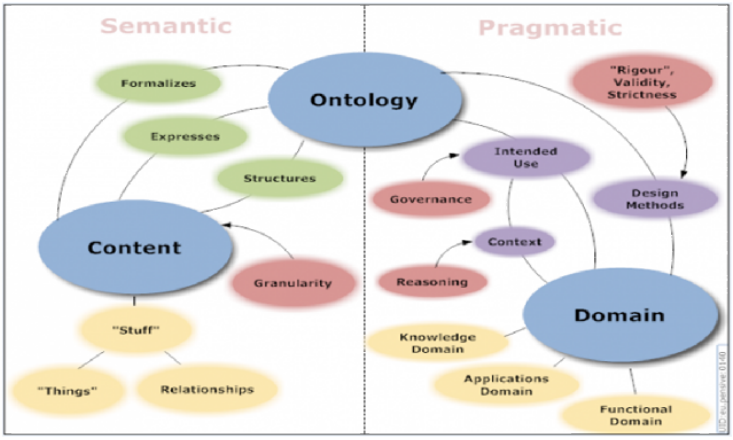


Figure 14 Principe d'ontologie, grâce à une ontologies, la connaissance est obtenue pour décrire des concepts et de leurs relations pour un domaine donné [67]

[1] Cet outil est le sujet d'un travail de Master réalisé dans le département MI de l'université l'Arbi Tebessi – Tébessa.

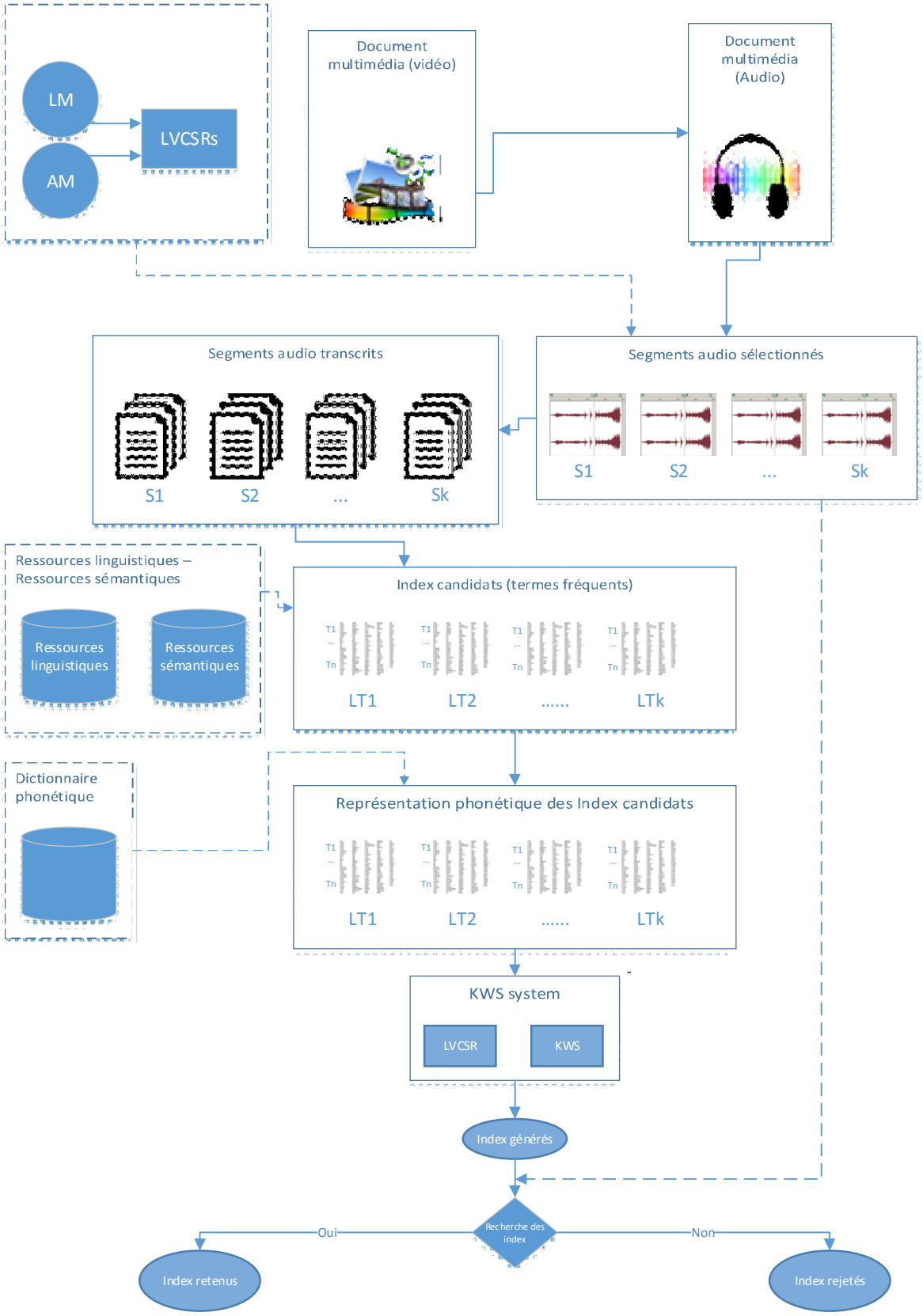


Figure 15 Architecture du système d'indexation proposé

Phase 2 : représentation phonétique des index candidats

Les résultats intermédiaires de la première phase sont des termes susceptibles d'indexation pour le document cible, qu'on a appelé « termes d'indexation candidats ». Ces termes sont obtenus à partir de la transcription des segments du document cible avec un enrichissement sémantique de ces derniers pour surmonter le problème de termes hors vocabulaire ainsi que l'élimination des fausses acceptations.

Dans cette phase, on va construire le modèle phonétique d'indexation de ces termes candidat, en utilisant le dictionnaire phonétique du langage. Dans la figure 17, on illustre des exemples de l'écriture phonétique des termes « Liste des termes en Anglais ».

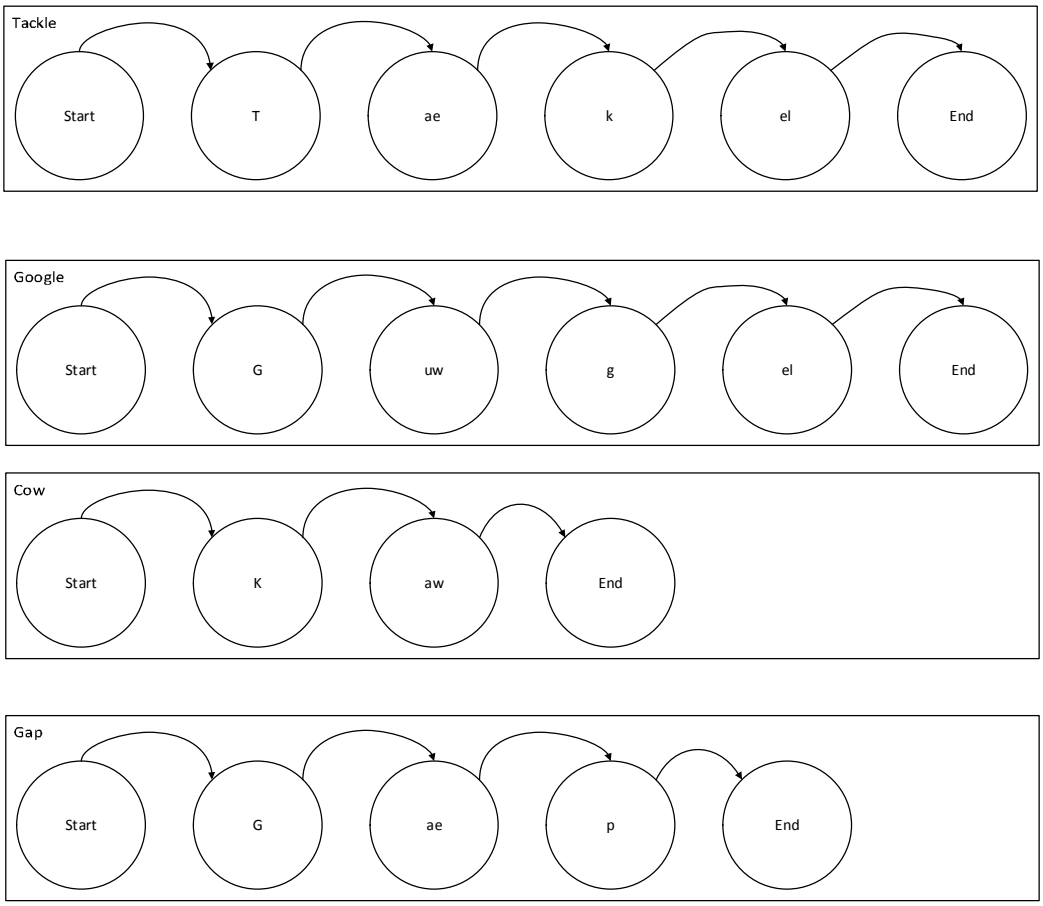


Figure 16 Exemple de représentation phonétique des termes

Cependant, en tenant compte de l'étude comparative présentée dans la partie précédente de notre mémoire, on a trouvé que les techniques d'indexation à base de sub-word sont les plus utilisées et fournissent des résultats encourageants.

Dans ce contexte, on propose pour notre architecture d'utiliser un dictionnaire phonétique pour faire la représentation phonétique des index candidats, afin de les passés à un système KWS qui se compose de deux parties : un module de LVCSR qui décode la

collection de recherche et génère le treillis correspondant présentée dans la figure 17 et un module de KWS qui permet de créer les index pour les treillis générer et recherche les mots-clés depuis les index générés.

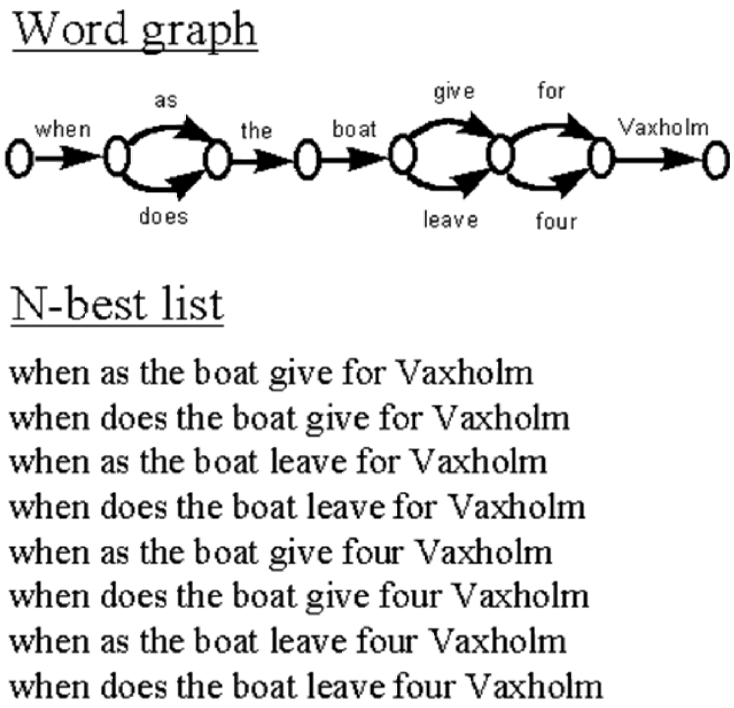


Figure 17 Exemple de Treillis de mots

Phase 3 : Validation des index

Cette phase a pour objectif de valider les indexes extraient à partir des segments du document multimédia. On note, qu'on a travaillé avec des segments du document et non pas avec l'intégralité du document. Donc, il faut vérifier si ces indexes sont réellement des indexes globales du document et non pas local pour les segments. A cet effet, on exécute une tâche de détection de mots clés a base phonétique sur des segments aléatoire test du document cible afin de valider le pouvoir de généralisation de ces indexes candidats. Les résultats de cette phase peuvent être classés dans les quatre propositions suivantes :

- Index candidat extraits à partir des segments du document et il est localisé dans les segments tests. Donc, index candidat retenu.
- Index candidat extrait à partir des segments du document et il n'est pas localisé dans les segments tests. Donc, index candidat rejeté.

- Index candidat déduit à partir des distances sémantiques avec les indexes extraits à partir des segments du document et il est localisé dans les segments tests. Donc, index candidat retenu.
- Index candidat déduit à partir des distances sémantiques avec les indexes extraits à partir des segments du document et il n'est pas localisé dans les segments tests. Donc, index candidat rejeté.

Conclusion

Dans ce chapitre, on a présenté la modélisation de notre solution pour l'indexation automatique sur le contenu des documents multimédias. Dans notre démarche, on a essayé de trouver une solution d'indexation automatique traitée sur deux aspects : aspect de sélection des termes discriminants du document multimédia et l'aspect de la recherche sur le contenu dans les documents de structure complexe tel que les documents multimédias parlés. Cependant, pour la simulation et la validation de notre architecture, on s'intéresse dans ce mémoire seulement sur le deuxième aspect. Dans ce contexte, on va présenter dans le chapitre suivant l'outil « *KALDI ASR* » qu'il sera utilisé ultérieurement.

Chapitre 2 – Kaldi speech recognition toolkit

Introduction

Kaldi[1] est une boîte à outils open-source pour la reconnaissance vocale écrite en C++ et sous licence Apache License v2.0. Le but de *Kaldi* est d'avoir un code moderne et flexible qui est facile à comprendre, modifier et d'étendre. *Kaldi* est disponible sur *SourceForge*[2] . Cet outil est compilé sur les systèmes *Unix* ; couramment utilisés ; ainsi que les autres systèmes d'exploitation tel que Microsoft *Windows*. Cependant, les chercheurs sur la reconnaissance automatique de la parole (ASR) ont plusieurs choix possibles de boîtes à outils open-source pour la construction d'un système de reconnaissance. Parmi eux on peut citer : HTK [68], Julius [69] (écrit en C), Sphinx-4 [70] (écrit en Java), et la boîte à outils RWTH ASR [71] (écrit en C ++).

Notre choix de la boite à outils *Kaldi* est basé sur l'étude comparative de [72], qui consiste à décrire une évaluation à grand échelle des toolkits de reconnaissance vocale open source. Les expériences effectuées par ces derniers ont montré un ordre des toolkits évalués en ce qui concerne le rapport effort/performance. Les résultats ont prouvé que Kaldi est plus performant que toutes les autres boites à outils de reconnaissance.

Recognizer	**VM1[3]**	**WSJ1[4]**
HDecode v3.4.1	22.9	19.8
Julius v4.3	27.2	23.1
pocketsphinx v0.8	23.9	21.4
Sphinx4	26.9	22.7
Kaldi	**12.7**	**6.5**

Tableau 4 Taux d'erreur de mots (WER) sur l'ensemble de test VM1 et l'ensemble WSJ 1 (novembre '93). [72]

[1] Selon la légende, Kaldi était le chevrier éthiopien qui a découvert la plante de café.
[2] http://kaldi.sf.net/
[3] Corpus issus du projet German Verbmobil project (1993), inclus des dialogues vocaux en 3 langues (Anglais, Japonais et l'allemand).
[4] Corpus issus en 1994, inclus des lectures des articles de Wall Street Journal News.

Les caractéristiques importantes de Kaldi :

La plateforme Kaldi est une plateforme complète pour le domaine de reconnaissance automatique de la parole et qui intègre des scripts de plusieurs projets. Elle construite autour des caractéristiques suivantes :

- ⇨ **Intégration avec Transducteurs à 'Etat Fini :** Kaldi utilise la boite à outil OpenFst toolkit[1] comme une bibliothèque.
- ⇨ **Vaste soutien de l'algèbre linéaire :** Kaldi inclut une bibliothèque de gestion des calculs matriciels qui enveloppe les routines BLAS et LAPACK standard.
- ⇨ **Conception extensible :** Kaldi essaye de fournir ces algorithmes sous la forme la plus générique possible.
- ⇨ **Licence Open :** Le code est distribué sous licence Apache v2.0, qui est l'une des licences moins restrictives disponibles.
- ⇨ **Direction complète :** Kaldi offre une direction complète pour la construction de systèmes de reconnaissance vocale qui fonctionnent à partir des bases de données largement disponibles telles que celles fournies par le Linguistic Data Consortium (LDC).
- ⇨ **Test complet :** Le but est pour que presque tout le code ait des routines correspondantes d'essai.

Aperçu de la boite à outil Kaldi

Dans cette section, on présente un aperçu schématique de la boîte à outils Kaldi schématisé dans la figure18. La boîte à outils dépend de deux bibliothèques externes qui sont également disponible gratuitement : l'une est OpenFst [73], et l'autre est une collection de bibliothèques d'algèbre numérique. Il utilise la norme « *Basic Linear Algebra Subroutines (BLAS)* » et « *Algèbre linéaire PACKage" (LAPACK)* »[2] pour ces routines. Les modules de bibliothèque peuvent être regroupés en deux moitiés distinctes, chacune en fonction de

[1] OpenFst est une bibliothèque pour la construction, la combinaison, l'optimisation et la recherche des transducteurs à états finis pondérés (FSTs). Les transducteurs à états finis pondérées sont des automates où chaque transition a une étiquette d'entrée, une étiquette de sortie, et un poids.
[2] http://www.netlib.org/blas/ et http://www.netlib.org/lapack/ respectivement.

seulement une des bibliothèques externes (voir figure 18). Un seul module qui est l'interface décodables relie ces deux moitiés.

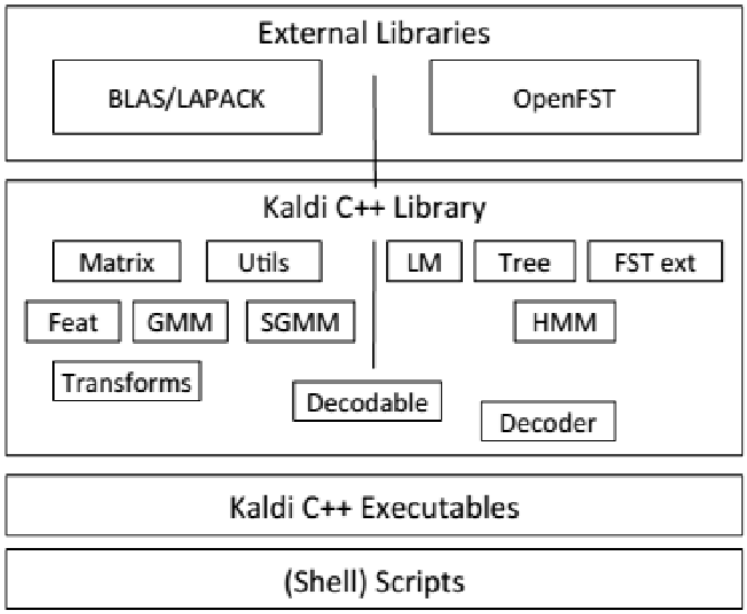


Figure 18 Une vue simplifiée des différentes composantes de Kaldi. Les modules de la bibliothèque peuvent être regroupés en ceux qui dépendent de bibliothèques d'algèbre linéaire et ceux qui dépendent de OpenFst. La classe décodable comble ces deux moitiés. Les modules qui sont plus bas dans le schéma dépendent d'un ou plusieurs modules qui sont plus haut. [74]

L'accès aux fonctionnalités de la bibliothèque est fourni grâce à des outils de ligne de commande écrit en C ++, qui sont ensuite appelés à partir d'un langage de script pour la construction et l'exécution d'un logiciel de reconnaissance vocale. Chaque outil a une fonctionnalité très spécifique avec un petit ensemble d'arguments de ligne de commande : par exemple, il y a des exécutables distincts pour : accumuler des statistiques, la mise à jour d'un modèle acoustique à base de GMM en utilisant une estimation du maximum de vraisemblance.

Extraction de caractéristiques

Dans la phase d'extraction des paramètres des fichiers sonores ; la plateforme Kaldi utilise le code d'extraction de caractéristiques et le code waveforme-reading. Ces routines ont pour but d'extraire les caractéristiques acoustiques standard MFCC et PLP.

Modélisation acoustique

L'objectif de Kaldi est pour soutenir les modèles classiques (à savoir diagonale GMMs) et modèles subspatial Mixture de gaussiennes (SGMM), mais pour être aussi facilement extensible à de nouveaux types de modèle.

A. Modèle de mixture gaussienne

Kaldi supporte les modèles GMMs, avec des structures de covariance diagonales et complètes. Plutôt que de représenter les densités gaussiennes individuelles séparément, Kaldi met en œuvre directement une classe de GMM qui est paramétrée par les paramètres naturels.

B. Modèle acoustique à base de GMM

Le « modèle acoustique » de la classe AmDiagGmm, définit dans Kaldi représente une collection d'objets DiagGmm, indexé par « pdf-ids » qui correspondent à des états HMM dépendant du contexte. Cette classe ne représente pas une structure de HMM, mais juste une collection de densités (à savoir GMM). Il y a des classes séparées qui représentent le HMM, principalement le code de la topologie et de la transition de modélisation et le code responsable de la compilation de décodage dans l'arbre de décodage, qui fournissent un mappage entre les états HMM et l'indice de pdf de la classe modèle acoustique.

C. Topologie de HMM

Il est possible dans Kaldi, de spécifier séparément la topologie HMM pour chaque phonème indépendant du contexte. Le format de la topologie permet aux états non émetteurs, et aux utilisateurs de pré spécifier le lissage des graphes de décodage pour les différents états HMM.

Arbres de décision phonétiques

L'objectif de la construction du code d'arbre de décision phonétique était pour le rendre efficace pour les tailles de contexte arbitraires, et aussi pour le rendre suffisant pour soutenir un large éventail d'approches. L'approche classique est, dans chaque état-HMM de chaque monophone, d'avoir un arbre de décision qui pose des questions au sujet des phonèmes gauches et des phonèmes droites. Les racines des arbres de décision peuvent être partagés entre les phonèmes et les états des phonèmes, et des questions peuvent être posées à propos de tout phonème dans la fenêtre de contexte, et sur l'état de HMM. Des questions phonétiques peuvent être fournis sur la base de connaissances linguistiques, mais dans la boite à outils Kaldi, les questions sont générées automatiquement basées sur un arbre de regroupement des phonèmes.

Modélisation du langage

La plateforme Kaldi utilise un framework basé sur les FST (Finit States transducer), il est possible, en principe, d'utiliser un modèle de langage qui peut être représenté comme un FST. Kaldi fournis des outils pour convertir LMs dans le format ARPA standard au format FST. Kaldi utilise la boîte à outils IRSTLM[1] à des fins telles que taille de LM. Pour la construction des LMs du texte brut. Les utilisateurs peuvent utiliser la boîte à outils de IRSTLM, pour laquelle Kaldi fournis une aide d'installation, ou une boîte à outils des fonctionnalités plus complètes telles que SRILM[2].

Décodage

La boite à outils Kaldi propose plusieurs décodeurs. Par "décodeur", on entend une classe C ++ qui implémente l'algorithme de décodage de base. Les décodeurs ne nécessitent pas un type particulier de modèle acoustique : ils ont besoin d'un objet satisfaisant à une interface très simple avec une fonction qui fournit une sorte de pointage de modèle acoustique pour une (entrée-symbole et frame) en particulier.

Les programmes de décodage qui peuvent être appelés depuis la ligne de commande sont tous très simple, le principe général est de faire un seul passage de décodage, et sont spécialisés pour un décodeur et un type de modèle acoustique. Le décodage multi-passe est mis en œuvre au niveau du script.

Tous les algorithmes d'apprentissage de décodage utilisent des transducteurs pondérés à états finis (WFSTs).

[1] Disponible à partir de : http://hlt.fbk.eu/en/irstlm
[2] Disponible à partir de : http://www.speech.sri.com/projects/srilm/

Chapitre 3 – Validation de l'approche proposée

Introduction

Dans ce chapitre, nous présenterons une simulation de cette architecture proposée avec des expérimentations sur des corpus libres tel que : *TED-LIUM corpus*[1], qu'il a été développé dans le contexte de la participation de l'LIUM à la campagne d'évaluation IWSLT 2011. Toutes ses données brutes (signaux acoustiques et ses annotations) ont été extrait du site TED et les transcriptions automatiques obtenues à partir d'un décodeur acoustique avec un alignement avec ses transcriptions. Ainsi qu'on a utilisé IARPA *Babel Project*[2], qui a pour but de construire un système de détection de mots clés avec les techniques de reconnaissance vocale pour un ensemble beaucoup plus vaste des différentes langues.

Environnement de travail

Pour réaliser notre simulation, il y a des prés requis qui doivent être présents.

Règle numéro 1 – l'utilisation d'un système d'exploitation Linux. Bien qu'il soit possible d'utiliser Kaldi sur Windows, nous avons choisi Ubuntu 16.04 Lts Desktop. Une version de Linux riche et stable, qui assure la mise en œuvre de la boite à outils Kaldi avec le moins de problèmes possibles. Puis, on vérifie la disponibilité des outils et ressources requise par Kaldi avec la commande **extras/check_dependencies.sh**. Cette étape permet d'installer les outils « package » suivants :

- **atlas** - automatisation et l'optimisation des calculs dans le domaine de algèbre linéaire,
- **autoconf** - compilation automatique de logiciel sur exploitation différent systèmes,
- **automake** – compilation et création de fichiers « Makefile » portables,
- **git**[3] - est un logiciel de gestion de versions décentralisé. Il est conçu pour être efficace tant avec les petits projets, que les plus importants,
- **libtool** - création de bibliothèques statiques et dynamiques,
- **svn** - système de contrôle de révision (Subversion),

[1] http://www-lium.univ-lemans.fr/en/content/ted-lium-corpus
[2] https://www.iarpa.gov/index.php/research-programs/babel
[3] ***Git*** est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de deux millions de personnes

- **wget** - transfert de données via HTTP, HTTPS et FTP, nécessaire à Kaldi pour le téléchargement et l'installation,
- **zlib** - pour la compression des données,

Cependant, dans la phase de simulation avec le projet Kaldi et la programmation, des composants et ressources systèmes doivent être installés :

- **awk** - langage de programmation, utilisé pour la recherche et le traitement des modèles dans les fichiers et les flux de données,
- **bash** - shell Unix et le langage de programmation de script,
- **grep** utilitaire de ligne de commande pour la recherche des ensembles de données en texte brut pour correspondant à une expression régulière,
- **perl** - langage de programmation dynamique, parfait pour les fichiers texte En traitement.

Après l'installation de tous les composants systèmes avec tous les outils nécessaires. On peut télécharger et compiler et construire la boîte à outil Kaldi avec la commande : « **make** ». Après l'installation de la boîte à outils Kaldi, on procède à faire notre simulation.

Structure des répertoires de Kaldi

Après l'installation de la boîte à outils Kaldi, on trouve que son répertoire contient l'arborescence suivante :

'Kaldi-tronc' : répertoire principal de Kaldi qui contient les répertoires suivants :

- **'Egs'** – répertoire plein d'exemples des scripts qui vous permet de construire rapidement des systèmes ASR depuis plus de 30 corpus vocaux populaires. On note que la majorité des corpus appartiennent à : Linguistic Data Consortium « LDC » et qui ne sont pas gratuits ;
- **'Misc'** – répertoire des outils supplémentaires, il est optionnel pour le fonctionnement correct Kaldi ;
- **'Src'** – répertoire du code source de Kaldi ;
- **'outils'** – répertoire des composants utiles et des outils externes ;
- **'windows'** – répertoire des outils pour exécuter Kaldi sous Windows.

Description du corpus

Pour réaliser notre simulation, nous avons choisi d'utiliser TED-LIUM corpus, sous licence Creative Commons BY-NC-ND 3.0 « http://creativecommons.org/licenses/by-nc-nd/3.0/deed.en ».

Ce corpus a été créé dans le cadre de la participation de l'équipe du laboratoire LIUM de l'université TED, à la campagne d'évaluation IWSLT 2011. Ainsi, toutes les données de ce corpus, ont été extraites des discussions vidéo disponibles gratuitement sur le site TED 1. Cela nous a conduit à disposer dans ce corpus de :

- 1495 discussions audio au format NIST sphère (SPH) ;
- 1495 transcriptions au format STM ;
- Dictionnaire avec la prononciation (159848 entrées) ;
- Les données monolingues sélectionnées pour la modélisation du langage de WMT12 public corpora ;

Les fichiers SPH ont les caractéristiques suivantes :

- Canaux : 1 ;
- Sample Rate : 16000 ;
- Précision : 16 bits ;
- Bitrate : 256k ;

Simulation et validation de l'approche proposée

Dans le cadre de réalisation de ce mémoire, on à procéder à faire une simulation afin de valider notre approche proposée, la simulation se divise en 3 parties : **partie 1-** reconnaissance vocale à base de vocabulaire large. **Partie 2** consiste à la création d'index et en fin la **partie 3** consiste à faire la recherche à base des index crées dans la partie 2 de la simulation.

Partie 1 : LVCSR avec Kaldi

Dans cette partie en réalise une tache de reconnaissance vocale à base du vocabulaire large du corpus TED-LIUM.

La tâche de reconnaissance vocale est exécutée selon les étapes suivantes :

1. Préparation des données : la figure suivante, présente une partie du script qui consiste à la tâche de préparation des données vocale (préparer les segments d'apprentissage et de test, traiter les fichiers .STM, préparer les fichiers texte, préparer les 'segments', la liaison entre les transcriptions et le contenu des signaux audio « utt2spk » et « spk2utt ». Puis la préparation des fichiers de liaison de localisation et des chemins « wav.scp », « reco2file_and_channel ».

```
#!/bin/bash
. path.sh
export LC_ALL=C
# Prepare: test, train,
for set in dev test train; do
  dir=data/$set
  mkdir -p $dir
  { # Add STM header, so sclite can prepare the '.lur' file
    echo ';;
;; LABEL "o" "Overall" "Overall results" ;; LABEL "f0" "f0" "Wideband channel" ;; LABEL "f2" "f2" "Telephone channel";; LABEL "m
"female" "Female" "Female Talkers";;'
    # Process the STMs
    cat db/TEDLIUM_release1/$set/stm/*.stm | sort -k1,1 -k2,2 -k4,4n | \
      sed -e 's:<F0_M>:<o,f0,male>:' \
          -e 's:<F0_F>:<o,f0,female>:' \
          -e 's:([0-9]):::g' \
          -e 's:<sil>::g' \
          -e 's:([^ ]*)$::' | \
      awk '{ $2 = "A"; print $0; }'
  } | local/join_suffix.py db/TEDLIUM_release1/TEDLIUM.150K.dic > data/$set/stm
  # Prepare 'text' file
  # - {NOISE} -> [NOISE] : map the tags to match symbols in dictionary
  cat $dir/stm | grep -v -e 'ignore_time_segment_in_scoring' -e ';;' | \
    awk '{ printf ("%s-%07d-%07d", $1, $4*100, $5*100);
           for (i=7;i<=NF;i++) { printf(" %s", $i); }
           printf("\n");
         }' | tr '{}' '[]' | sort -k1,1 > $dir/text || exit 1
  # Prepare 'segments', 'utt2spk', 'spk2utt'
  cat $dir/text | cut -d" " -f 1 | awk -F"-" '{printf("%s %s %07.2f %07.2f\n", $0, $1, $2/100.0, $3/100.0)}' > $dir/segments
  cat $dir/segments | awk '{print $1, $2}' > $dir/utt2spk
  cat $dir/utt2spk | utils/utt2spk_to_spk2utt.pl > $dir/spk2utt
  # Prepare 'wav.scp', 'reco2file_and_channel'
  cat $dir/spk2utt | awk -v set=$set -v pwd=$PWD '{ printf("%s sph2pipe -f wav -p %s/db/TEDLIUM_release1/%s/sph/%s.sph |\n", $1,
  cat $dir/wav.scp | awk '{ print $1, $1, "A"; }' > $dir/reco2file_and_channel
  # Create empty 'glm' file
  echo ';; empty.glm
  [FAKE]     =>  %HESITATION     / [ ] __ [ ] ;; hesitation token
  ' > data/$set/glm
  # Check that data dirs are okay!
```

Figure 19 Partie du script de préparation des données - prepar-data.sh

2. L'extraction des paramètres acoustiques des ressources audio. Les caractéristiques MFCC sont utilisés à l'aide des classes définies dans la bibliothèque « Kaldi/outils ».

```
fi

# Feature extraction
feat_dir=$pwd/data/mfcc_features
if [ $stage -le 1 ]; then
  for set in test dev train; do
    dir=data/$set
    steps/make_mfcc.sh --nj 20 --cmd "$train_cmd" $dir $dir/log $dir/data || exit 1
    steps/compute_cmvn_stats.sh $dir $dir/log $dir/data || exit 1
  done
fi
```

Figure 20 Partie du script qui consiste à l'extraction des caractéristiques - run.sh-

3. Dans cette étape, on segmente les données d'apprentissage (118 heures) vers des segments plus cours (à savoir 10000 segments) pour faciliter la tâche

```
if [ $stage -le 2 ]; then
  utils/subset_data_dir.sh --shortest data/train 10000 data/train_10kshort || exit 1
  local/remove_dup_utts.sh 10 data/train_10kshort data/train_10kshort_nodup || exit 1
fi
```

Figure 21 Partie du script pour créer des segments cours pour faciliter l'apprentissage

d'apprentissage.

4. L'exécution des routines d'apprentissage pour la construction des modèles à partir des ressources définies dans les étapes précédentes.

```
# Train
if [ $stage -le 3 ]; then
  steps/train_mono.sh --nj 20 --cmd "$train_cmd" \
    data/train_10kshort_nodup data/lang_nosp exp/mono0a || exit 1

  steps/align_si.sh --nj $nj --cmd "$train_cmd" \
    data/train data/lang_nosp exp/mono0a exp/mono0a_ali || exit 1

  steps/train_deltas.sh --cmd "$train_cmd" \
    2500 30000 data/train data/lang_nosp exp/mono0a_ali exp/tri1 || exit 1

  utils/mkgraph.sh data/lang_nosp_test exp/tri1 exp/tri1/graph_nosp || exit 1

  steps/decode.sh --nj $decode_nj --cmd "$decode_cmd" \
    --num-threads 4 \
    exp/tri1/graph_nosp data/dev exp/tri1/decode_nosp_dev || exit 1
  steps/decode.sh --nj $decode_nj --cmd "$decode_cmd" \
    --num-threads 4 \
    exp/tri1/graph_nosp data/test exp/tri1/decode_nosp_test || exit 1
fi

if [ $stage -le 4 ]; then
  steps/align_si.sh --nj $nj --cmd "$train_cmd" \
    data/train data/lang_nosp exp/tri1 exp/tri1_ali || exit 1

  steps/train_lda_mllt.sh --cmd "$train_cmd" \
    4000 50000 data/train data/lang_nosp exp/tri1_ali exp/tri2 || exit 1

  utils/mkgraph.sh data/lang_nosp_test exp/tri2 exp/tri2/graph_nosp || exit 1

  steps/decode.sh --nj $decode_nj --cmd "$decode_cmd" \
```

Figure 22 Partie du script assurant l'apprentissage - run.sh-

Partie 2 : Création d'index

Description du système KWS à base OpenFST

Les treillis générés par les modèles BMMI sont traités en utilisant la technique d'indexation à base de treillis présenté dans la deuxième partie de ce mémoire. Les treillis de tous les énoncés dans l'ensemble d'évaluation sont convertis à partir de transducteurs à états finis « FST ».

Étant donné un mot clé ou une phrase, on crée une simple machine à états finis qui accepte le mot-clé / phrase et compose avec le transducteur de facteur pour obtenir toutes les occurrences du mot-clé / phrase dans le jeu de treillis de l'ensemble d'évaluation, ainsi que l'ID de conversation, le temps de début et de fin et la probabilité postérieur de treillis de chaque occurrence. Chaque exemple présumé d'un mot-clé ainsi obtenu sont triés en fonction de leurs probabilités a posteriori. En outre, une décision OUI / NON est attribué à chaque instance. Plus précisément, pour chaque mot clé, son nombre prévu dans l'ensemble d'évaluation est estimée en additionnant les probabilités a posteriori de tous ses résultats

présumés, et un seuil de décision qui maximise la valeur pondérée attendue du terme est calculé pour chaque mot-clé. Tous les mots clés avec des probabilités a posteriori au-dessus de ce seuil spécifique à ce mot clé sont marqués OUI. La figure suivante présente la partie du script pour créer les index.

Partie 3 : Recherche d'index

Nous avons la méthode de proxy keyword pour résoudre le problème des mots hors vocabulaire (OOV) du treillis des mots. Cette approche consiste à trouver les mots acoustiques semblables dans le vocabulaire (IV), au mot-clé d'OOV, et les utiliser comme proxy de mots-clés au lieu du mot-clé OOV originale.

```
if [ -z "$model" ]; then # if --model <mdl> was not specified on the command line...
  model=$srcdir/final.mdl;
fi

for f in $word_boundary $model $decodedir/lat.1.gz; do
  [ ! -f $f ] && echo "make_index.sh: no such file $f" && exit 1;
done

echo "Using model: $model"

if [ ! -z $silence_word ]; then
  silence_int=`grep -w $silence_word $langdir/words.txt | awk '{print $2}'`
  [ -z $silence_int ] && \
    echo "Error: could not find integer representation of silence word $silence_word" && exit 1;
  silence_opt="--silence-label=$silence_int"
fi

$cmd JOB=1:$nj $kwsdir/log/index.JOB.log \
  lattice-add-penalty --word-ins-penalty=$word_ins_penalty "ark:gzip -cdf $decodedir/lat.JOB.gz|" ark:- \| \
    lattice-align-words $silence_opt --max-expand=$max_expand $word_boundary $model  ark:- ark:- \| \
    lattice-scale --acoustic-scale=$acwt --lm-scale=$lmwt ark:- ark:- \| \
    lattice-to-kws-index --max-states-scale=$max_states_scale --allow-partial=true \
    --max-silence-frames=$max_silence_frames --strict=$strict ark:$utter_id ark:- ark:- \| \
    kws-index-union --skip-optimization=$skip_optimization --strict=$strict --max-states=$max_states \
    ark:- "ark:|gzip -c > $kwsdir/index.JOB.gz" || exit 1
```

Figure 23 Partie du script pour créer les index - make_index.sh-

Le processus général de génération de mots clés proxy peut être formulé comme suit[79]:

$$K' = Project(ShortestPath((L_1^*)^{-1} o\ E\ o\ (L_2^*)\ o\ K))$$

Plus précisément, si K représente un accepteur d'états finis pour un mot-clé qui est OOV par rapport à un lexique de référence $L1$, mais IV par rapport à un lexique augmentée $L2$, où les deux $L1$ et $L2$ sont des transducteurs à états finis qui acceptent des séquences de phonèmes et les mots de sortie, et si E est un "edit-distance" transducteur qui mappe toute séquence de phonèmes à toute autre séquence de phonèmes avec un coût égal à leur 'distance de Levenshtein', représente alors le mot-clé / phrase K' en vocabulaire qui est le plus proche de K. On peut utiliser K' comme un proxy pour K à la recherche des treillis générés en utilisant $L1$.

```
# Begin configuration section.
cmd=run.pl
nbest=-1
strict=true
indices_dir=
# End configuration section.
echo "$0 $@"  # Print the command line for logging
[ -f ./path.sh ] && . ./path.sh; # source the path.
. parse_options.sh || exit 1;
if [ $# != 2 ]; then
   echo "Usage: steps/search_index.sh [options] <kws-data-dir> <kws-dir>"
   echo " e.g.: steps/search_index.sh data/kws exp/sgmm2_5a_mmi/decode/kws/"
   echo ""
   echo "main options (for others, see top of script file)"
   echo "  --cmd (utils/run.pl|utils/queue.pl <queue opts>) # how to run jobs."
   echo "  --nbest <int>                                    # return n best results.
   echo "  --indices-dir <path>                             # where the indices shoul
   <kws-dir>"
   exit 1;
fi
kwsdatadir=$1;
kwsdir=$2;
if [ -z $indices_dir ] ; then
  indices_dir=$kwsdir
fi
mkdir -p $kwsdir/log;
nj=`cat $indices_dir/num_jobs` || exit 1;
keywords=$kwsdatadir/keywords.fsts;
for f in $indices_dir/index.1.gz $keywords; do
  [ ! -f $f ] && echo "make_index.sh: no such file $f" && exit 1;
done
$cmd JOB=1:$nj $kwsdir/log/search.JOB.log \
  kws-search --strict=$strict --negative-tolerance=-1 \
```

Figure 24 Partie du script assurant la recherche des index - earch_index.sh-

Conclusion

Dans cette partie, nous avons essayé de proposer une approche d'indexation automatique des documents parlés multimédia, pour la recherche et la détection des mots clés, avec l'utilisation de la boite à outils Kaldi, comme plateforme d'expérimentation et de génération des systèmes ASR, afin de prouver notre approche proposée dans le chapitre1.

Tout en présentant quelques figures illustrant des parties des scripts utilisés tout au long de la réalisation de notre mémoire.

Conclusion et perspectives

Le travail réalisé dans le cadre de ce mémoire s'inscrit dans le domaine d'indexation sur le contenu des ressources multimédias. Nous nous sommes particulièrement intéressés à là à proposer une démarche qui fournisse des indexes extraits automatiquement des contenus des documents multimédias pour améliorer la pertinence des résultats de recherches sur ce contenu.

Dans le volet de validation et de simulation de notre approche proposée, on a essayé d'utiliser la plateforme « Kaldi ». Cette dernière est une plateforme complète du domaine de traitements de la parole. Cependant, la majorité de ces ressources ne sont pas gratuites et ceci a influé sur la phase de l'apprentissage et à la suite des étapes de simulation.

Dans ce contexte, nos perspectives futures sont de préparé des ressources parlé avec le format utilisé dans la plateforme de test afin que nous puissions valider notre approche.

Bibliographie

[1] G. Salton et M. J. McGill, «Introduction to Moderne Information Retrieval,» New York, 1983.

[2] M. Baziz, M. Boughanem, N. Aussenac-Gilles et C. Chrisment, «Semantic cores for representing documents in IR,» chez *ACM Symposuim on Applied Computing*, 2005.

[3] N. Kompaoré, «Fusion de systèmes et analyse des caractéristiques linguistiques, Thèse de doctorat en informatique.,» Université Paul Sabatier de Toulouse, Toulouse, 2008.

[4] P. Ingwersen, «Polyrepresentation of information needs and semantic entities: elements of a cognitive theory for information retrieval interaction,» chez *SIGIR'94*, London, Springer, 1994, pp. 101-110.

[5] K. Ng, «Subword-based Approaches for Spoken Document Retrieval,» Department of Electrical Engineering and Computer Science, Massachusetts , 2000.

[6] W. B. Croft, «Model of cluster searching based on classification,» chez *Information Systems*, 5 éd., vol. 5, 1980, pp. 189-195.

[7] D. D. Lewis, «Representation and learning in information retrieval (Doctoral dissertation),» University of Massachusetts, Massachusetts, 1992.

[8] N. A. Chinchor et Sundheim, B, «Message Understanding Conference (MUC) tests of discourse processing,» chez *Proc. AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, 1995, pp. 21-26.

[9] J. Kupiec, J. Pedersen et F. Chen, «A trainable document summarizer.,» chez *InProceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, USA, 1995, pp. 68-73.

[10] R. Baeza-Yates, . B. Ribeiro-Neto et others, Modern information retrieval, vol. 463, ACM press New York, 1999.

[11] G. Salton, «A comparison between manual and automatic indexing methods,» *American Documentation,* vol. 20, n° %11, pp. 61-71, 1969.

[12] S. E. Robertson et Jones, K Sparck, «Relevance weighting of search terms,» *Journal of the American Society for Information science,* vol. 27, n° %13, pp. 129-146, 1976.

[13] M. E. Maron et Kuhns, John L, «On relevance, probabilistic indexing and information retrieval,» *Journal of the ACM (JACM),* vol. 7, n° %13, pp. 216-244, 1960.

[14] C. Tambellini, «Un système de recherche d'information adapté aux données incertaines: adaptation du modèle de langue,» Grenoble I - France, 2007.

[15] M. Charhad, «Modèles de Documents Vidéo basés sur le Formalisme des Graphes Conceptuels pour l'Indexation et la Recherche par le Contenu Sémantique,» UNIVERSITE JOSEPH FOURIER-Grenoble, Grenoble, 2005.

[16] Alain Baccini , Sébastien Déjean, Désiré Kompaoré et Josiane Mothe, «Analyse des critères d'évaluation des,» 2006.

[17] Y. B. Ayed, «Détection de mots clés dans un flux de parole,» Télécom Paris Tech, 2003.

[18] C. Schmandt, «Voice Communication with Computers (Conversational Systems),» New York, 1994.

[19] J. G. Wilpon, L.R. Rabiner, C.H. Lee et E.R. Goldman, «Automatic recognition of keywords in unconstrained speech using hidden Markov models,» *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 38, n° %111, pp. 1870-1878, 1990.

[20] R. C. Rose, E. I. Chang et R. P. Lippmann, «Techniques for information retrieval from voice messages,» chez *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, 1991, pp. 317-320.

[21] S. Nakamura, T. Akabane et S. Hamaguchi, «Robust word spotting in adverse car environments.,» chez *EUROSPEECH*, 1993.

[22] R. Bossemeyer, J. Wilpon, C. Lee et L. Rabiner, «Automatic speech recognition of small vocabularies within the context of unconstrained input,» *The Journal of the Acoustical Society of America,* vol. 84, n° %1S1, pp. S212--S212, 1988.

[23] R. C. Rose, E. I. Chang et R. P. Lippmann, «Techniques for information retrieval from voice messages,» chez *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, 1991, pp. 317--320.

[24] O. Fujimura, «Syllables as concatenated demisyllables and affixes,» *The Journal of the Acoustical Society of America,* vol. 59, n° %1S1, pp. S55--S55, 1976.

[25] D. Wang, «Data resources used in thesis of D. WANG,» Decembre 2009. [En ligne]. Available: http://data.cstr.ed.ac.uk/dwang2/thesis-res.html. [Accès le 24 Avril 2016].

[26] H. Sakoe, S. Chiba, A. Waibel et K. Lee, «Dynamic programming algorithm optimization for spoken word recognition,» *Readings in speech recognition,* vol. 159, 1990.

[27] G. J. Jones, J. T. Foote, K. S. Jones et S. J. Young, «Retrieving spoken documents by combining multiple index sources,» chez *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 1996, pp. 30--38.

[28] K. Yu, «Adaptive T raining for Large V ocabulary Continuous Speech Recognition,» University of Cambridge, Cambridge, 2006.

[29] J. Holmes et W. Holmes, Speech Synthesis and Recognition, Second Edition éd., London: Taylor & Francis, 2001.

[30] T. Pellegrini, «Transcription automatique de langues peu dotées.,» Université Paris Sud-Paris XI, Paris, 2008.

[31] B. LECOUTEUX, «Reconnaissance automatique de la parole guidée par des transcriptions a priori,» UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE, 2008.

[32] J. MARIANI, Reconnaissance automatique de la parole : progrès et tendances, vol. 7, ORSAY, 2002.

[33] C. Allauzen, M. Mohri et M. Saraclar, «General indexation of weighted automata: application to spoken utterance retrieval,» chez *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004*, Association for Computational Linguistics, 2004, pp. 33--40.

[34] Y. a. F. D. a. H. J.-P. a. C. G. Benayed, «Confidence measures for keyword spotting using support vector machines,» chez *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, IEEE, 2003, pp. I--588.

[35] I. Zitouni, «Modélisation du langage pour les systèmes de reconnaissance de la parole destinés aux grands vocabulaires: application à MAUD,» université Henri Poincaré - Nancy 1, 2000.

[36] R. C. Rose, «Word spotting from continuous speech utterances,» chez *Automatic speech and speaker recognition*, Springer, 1996, pp. 303--329.

[37] L. Boves, R. Carlson, E. W. Hinrichs, D. House, S. Krauwer, L. Lemnitzer, M. Vainio et P. Wittenburg, «Resources for speech research: present and future infrastructure needs,» chez *INTERSPEECH*, 2009, pp. 1803--1806.

[38] S. Novotney, R. Schwartz et J. Ma, «Unsupervised acoustic and language model training with small amounts of labelled data,» chez *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, IEEE, 2009, pp. 4297--4300.

[39] D. A. James et S. J. Young, «A fast lattice-based approach to vocabulary independent wordspotting,» chez *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, IEEE, 1994, pp. I--377.

[40] K. Thambiratnam et S. Sridharan, «Dynamic Match Phone-Lattice Searches For Very Fast And Accurate Unrestricted Vocabulary Keyword Spotting.,» chez *ICASSP (1)*, 2005, pp. 465--468.

[41] D. Vergyri, I. Shafran, A. Stolcke, V. R. R. Gadde, M. Akbacak, B. Roark et W. Wang, «The SRI/OGI 2006 spoken term detection system.,» chez *INTERSPEECH*, Citeseer, 2007, pp. 2393--2396.

[42] J. Mamou, B. Ramabhadran et O. Siohan, «Vocabulary independent spoken term detection,» chez *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, New York USA, 2007, pp. 615--622.

[43] R. C. Rose et D. B. Paul, «A hidden Markov model based keyword recognition system,» chez *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, 1990, pp. 129--132.

[44] I. Szoke, P. Schwarz, P. Patejka, L. Burget, M. Karafiat, M. Fapso et J. Cernocky, «Comparison of keyword spotting approaches for informal continuous speech.,» chez *Interspeech*, Lisbon, Portugal, 2005, pp. 633--636.

[45] D. Grangier, J. Keshet et S. Bengio, «Chapter on discriminative keyword spotting.,» chez *In Automatic speech and speaker recognition: large margin and kernel methods.*, NewYork:Wiley, 2009.

[46] R. A. Sukkar, A. R. Setlur, M. G. Rahim et C.-H. Lee, «Utterance verification of keyword strings using word-based minimum verification error (WB-MVE) training,» chez *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, IEEE, 1996, pp. 518--521.

[47] E. D. Sandness et I. L. Hetherington, «Keyword-based discriminative training of acoustic models.,» chez *INTERSPEECH*, 2000, pp. 135--138.

[48] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig et A. Stolcke, «Neural-network based measures of confidence for word recognition,» chez *icassp*, IEEE, 1997, p. 887.

[49] J. Keshet, D. Grangier et S. Bengio, «Discriminative keyword spotting,» *Speech Communication,* vol. 51, n° %14, pp. 317--329, 2009.

[50] J. S. Garofolo, C. G. Auzanne et E. M. Voorhees, «The TREC spoken document retrieval track: A success story,» chez *Content-Based Multimedia Information Access-Volume 1*, LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2000, pp. 1--20.

[51] D. Hakkani-Tür et G. Riccardi, «A general algorithm for word graph matrix decomposition,» chez *Acoustics, Speech, and Signal Processing, 2003.*

Proceedings.(ICASSP'03). 2003 IEEE International Conference on, IEEE, 2003, pp. I--596.

[52] L. Mangu, E. Brill et A. Stolcke, «Finding consensus in speech recognition: word error minimization and other applications of confusion networks,» *Computer Speech \& Language,* vol. 14, n° %14, pp. 373--400, 2000.

[53] C. Chelba et A. Acero, «Position specific posterior lattices for indexing speech,» chez *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2005, pp. 443--450.

[54] Y.-C. Pan et L.-s. Lee, «Performance analysis for lattice-based speech indexing approaches using words and subword units,» *Audio, Speech, and Language Processing, IEEE Transactions on,* vol. 18, n° %16, pp. 1562--1574, 2010.

[55] K. Ng, «Subword-based Approaches for Spoken Document Retrieval,» Massachusetts Institute of Technology, Massachusetts, 2000.

[56] I. Szoke, L. Burget, J. Cernocky et M. Fapso, «Sub-word modeling of out of vocabulary words in spoken term detection,» chez *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*, Goa, India, IEEE, 2008, pp. 273--276.

[57] K. Ng et V. W. Zue, «Subword-based approaches for spoken document retrieval,» *Speech Communication,* vol. 32, n° %13, pp. 157--186, 2000.

[58] S. Deligne et F. Bimbot, «Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams,» chez *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, Michigan, USA, IEEE, 1995, pp. 169--172.

[59] P. Can, E. Cooper, A. Sethy, C. White, B. Ramabhadran et M. Saraclar, «Effect of pronunciations on oov queries in spoken term,» chez *InProc. int. conf. acoustics, speech and signal processing,*, Taipei, Taiwan, IEEE, 2009, pp. 3957--3960.

[60] M. Saraclar et R. Sproat, «Lattice based search for spoken utterance retrieval,» *Urbana,* vol. 51, p. 61801, 2004.

[61] K. Thambiratnam et S. Sridharan, «Dynamic match phonelattice searches for very fast and accurate unrestricted vocabulary,» chez *In Proc. int. conf. acoustics, speech and signal processing*, Philadelphia, USA, 2005, pp. 465--468.

[62] J. R. Rohlicek, Chapter on word spotting. InModern methods of speech processing, Norwell: Kluwer Academic., 1995.

[63] M. Mohri, F. Pereira et M. Riley, «Weighted automata in text and speech processing,» *arXiv preprint cs/0503077,* 2005.

[64] D. Can et M. Saraclar, «Lattice indexing for spoken term detection,» *Audio, Speech, and Language Processing, IEEE Transactions on,* pp. 2338--2347, 2011.

[65] A. Garcia et H. Gish, «Keyword spotting of arbitrary words using minimal speech resources,» chez *InProc. int. conf. Acoustics, Speech and Signal.*, IEEE, 2006, pp. I--I.

[66] S. S. R. &. M. J. Novotney, «Unsupervised acoustic and language model training with small amounts of labelled data,» chez *InProc. int. conf. acoustics, speech and signal processing.*, Taipei, Taiwan, IEEE., 2009, pp. pp. 4297-4300.

[67] http://wiki.opensemanticframework.org, «A_Basic_Guide_to_Ontologies,» 2014. [En ligne]. Available: http://wiki.opensemanticframework.org/index.php/A_Basic_Guide_to_Ontologies. [Accès le 20 04 2016].

[68] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey et others, «The HTK book (v3. 4),» *Cambridge University,* 2006.

[69] A. Lee, T. Kawahara et K. Shikano, «Julius – an open source realtime large vocabulary recognition engine,» chez *EUROSPEECH*, 2001, p. 1691–1694.

[70] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf et J. Woelfel, «Sphinx-4: A flexible open source framework for speech recognition,» 2004.

[71] D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Loof, R. Schluter et H. Ney, «The RWTH Aachen University Open Source Speech Recognition System,» chez *INTERSPEECH*, 2009, p. 2111–2114.

[72] C. Gaida, P. Lange, R. Petrick, P. Proba, A. Malatawy et D. Suendermann-Oeft, «Comparing open-source speech recognition toolkits.,» 2014.

[73] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut et M. Mohri, «OpenFst: A general and efficient weighted finite-state transducer library,» chez *Implementation and Application of Automata*, Springer, 2007, pp. 11--23.

[74] D. Povey, G. Arnab, B. Gilles, B. Lukas, O. Glembek, G. Nagendra, H. Mirko, c. Petr Motlı, Q. Yanmin, S. Petr, S. Jan, S. Georg et V. Karel, «The Kaldi Speech Recognition Toolkit».

[75] S. Young et P. Woodland, «The use of state tying in continuous speech recognition,» chez *Eurospeech*, 19993.

[76] T. Hain et P. W. e. al, «Automatic transcription of conversational telephone speech,» *IEEE transactions,* 2004.

[77] R. Prasad, S. Mastocas et e. al, «the 2004 BBN/LIMSI 20xRT english conversational telephone speech recongnition,» chez *Interspeech*, 2005.

[78] S. Young et P. Woodland, «State clustring in hiden Markov Model - based continuous speech recognition,» chez *Computer speech and language* , 1994.

[79] G. Chen, O. Yilmaz, J. Trmal, D. Povey et S. Khudanpur, «USING PROXIES FOR OOV KEYWORDS IN THE KEYWORD SEARCH TASK,» chez *ASRU2013*, 2013.

[80] C. Van Rijsbergen, «Information retrieval: theory and practice,» chez *Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems*, London, 1979, pp. 1-14.

[81] P. Gelin, «Détection de mots clés dans un flux de parole: Application à l'indexation de documents multimédia,» Université de liège, Liège, 1997.

[82] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal et R. C. Rose, «The subspace Gaussian mixture model—A structured model for speech recognition,» *Computer Speech & Language,* vol. 25, n° %12, pp. 404--439, 2011.

[83] D. R. H. Miller, K. M., K. C., K. O., C. T., L. S., S. R. M. et G. H., «Rapid and accurate spoken term detection,» *in Proc. of Interspeech 2007,* vol. vol. 7, p. pp. 314–317, 2007.

[84] G. Chen, O. Yilmaz, J. Trmal, D. Povey et S. Khudanpur, «USING PROXIES FOR OOV KEYWORDS IN THE KEYWORD SEARCH TASK,» chez *ASRU2013*, 2013.