UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de Valencia

Comparació de les eines informàtiques TLK i Kaldi per al desenvolupament de sistemes de reconeixement de la parla en català/valencià

Treball Fi de Grau

Grau en Enginyeria Informàtica

Autor: Pau Baquero Arnal

Tutor: Alfons Juan Ciscar
Cotutor: Albert Sanchis Navarro
Director Experimental: Adrià Giménez Pastor

Curs 2015-2016

Resum

En aquest treball es comparen les eines de reconeixement de la parla TLK i Kaldi per al desenvolupament de sistemes de reconeixement en català. Amb aquest objectiu, es construeixen i avaluen sistemes amb cada eina per fer-ne una valoració comparativa en termes de qualitat de resultats, eficiència temporal i facilitat d'ús. Es presenten els corpus de dades utilitzats. Es descriuen els passos d'entrenament dels sistemes i els models obtinguts al llarg del procés, així com l'ús que se'n fa per al reconeixement.

Paraules clau: Reconeixement de patrons, Reconeixement de la parla, ASR, català, comparació, TLK, Kaldi

Abstract

In this project the speech recognition toolkits TLK and Kaldi are compared for the development of Catalan speech recognition systems. To accomplish this, systems are constructed and evaluated with each toolkit for a comparative assessment to be made in terms of quality of results, temporal efficiency and ease of use. The used data corpora are presented. The system training steps and the models obtained throughout the process are described, as well as the use made of those models for recognition.

Key words: Pattern recognition, Speech recognition, ASR, catalan, comparison, TLK, Kaldi

Índex

CAPÍTOL 1

Introducció

Aquest treball compara dues eines informàtiques per a construir sistemes de reconeixement de la parla, TLK i Kaldi, en el desenvolupament de sistemes de reconeixement del català. Aquest capítol introductori serveix per a introduir la motivació del treball així com els conceptes bàsics necessaris del context del treball, i per explicar l'estructura de la memòria.

En aquest capítol, la secció 1.1 exposa la motivació del treball. En la secció 1.2 s'introdueix el reconeixement de patrons, per a centrar-nos en la secció 1.3 concretament en el reconeixement automàtic de la parla com a aplicació específica. En la secció 1.4 es presenten les mètriques utilitzades per avaluar els resultats dels sistemes i en la secció 1.5 s'explica com està estructurada la memòria.

1.1 Motivació

Les tecnologies de la parla tracten amb la forma més natural de comunicació en les persones: el llenguatge. Hi ha tres grans tipus de tecnologies de processament del llenguatge: el reconeixement, la traducció i la conversió de text a veu. Aquestes tecnologies es basen en models estadístics. Per construir tots aquests models, cal una gran quantitat de dades en la llengua a processar.

Actualment, el català es troba en una situació relativament bona pel que fa a recursos lingüístics i textuals. Aquests recursos han aprofitat per construir millors sistemes de reconeixement de la parla, de traducció automàtica i de síntesi de veu en català, encara que l'estat de la tecnologia de la parla en català no es troba al nivell d'altres llengües europees com l'anglés, el castellà, el francés, l'alemany o l'italià.

Per a més informació sobre la situació de la tecnologia de la parla en català i les seues perspectives de futur, podeu consultar *La tecnologia de la parla en català. Avenços i reptes* [1].

Els sistemes de reconeixement automàtic de la parla, o ASR (*Automatic Speech Recognition*), permeten generar transcripcions ràpides i barates per a vídeos i àudios, i això també pot servir per a altres aplicacions, com per exemple: subtitolació automàtica, extracció d'informació (resums, categorització, cerca de paraules) o traducció automàtica. En particular, l'ASR en català té interés en el marc de la Universitat Politècnica de València (UPV) perquè permet transcriure i traduir vídeos educatius amb una qualitat acceptable, ràpidament i amb baix cost.

Hi ha molts *toolkits* especialitzats en ASR, com poden ser HTK [2], RASR [3] o Sphinx [4]; però en aquest treball compararem les prestacions de TLK [5] i Kaldi [6].

TLK és l'eina d'ASR desenvolupada pel grup de recerca *Machine Learning and Language Processing* (MLLP) del Departament de Sistemes Informàtics i Computació (DSIC) que s'utilitza actualment per a la transcripció de vídeos educatius de la UPV en diferents llengües (principalment català i castellà), i que permet la seua traducció automàtica.

Kaldi és una de les eines més utilitzades en l'actualitat degut a la seua llicència lliure (Apache), i a que conceptualment ha estat creat amb la intenció d'oferir un programari flexible i extensible, a més de ser una de les primeres eines a incloure xarxes neurals profundes en ASR.

Encara que TLK està oferint resultats excel·lents en la transcripció automàtica de vídeos de la plataforma Polimedia, encara no hi ha cap comparació empírica amb Kaldi per fer-ne una valoració relativa. El principal objectiu d'aquest treball és la comparació de les prestacions d'aquestes dues eines informàtiques en el desenvolupament de sistemes de reconeixement de la parla en català, i concretament dins de la tasca de transcripció automàtica de vídeos educatius de la UPV en català.

1.2 Reconeixement de patrons

Amb aquesta secció, introduïm el reconeixement de patrons com a camp d'investigació que engloba, entre altres, el reconeixement de la parla. Aquesta secció i la següent serveixen per dotar al lector dels conceptes fonamentals del context del treball realitzat.

En la figura 1.1 podeu trobar una representació esquemàtica del funcionament genèric dels sistemes classificadors o reconeixedors, la construcció dels quals és l'aplicació més típica del reconeixement de patrons. "Reconeixement" i "classificació" fan de sinònims en aquest camp, i en endavant els podreu trobar utilitzats de forma indistinta.

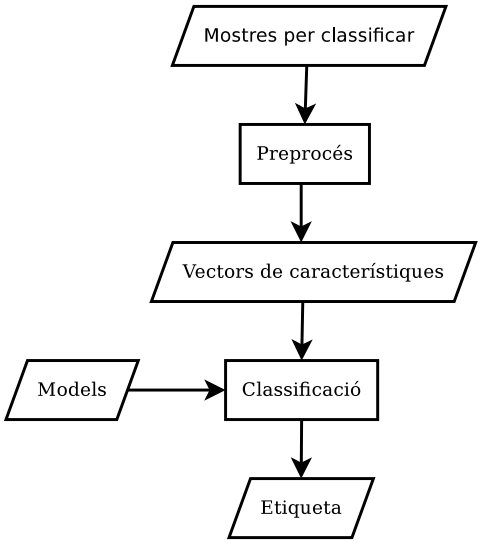

Figura 1.1: Sistema classificador genèric.

Segons Webb i Copsey, "el reconeixement de patrons estudia l'ús de tècniques estadístiques per a l'anàlisi de dades a fi d'extraure informació i prendre decisions justificades" [7]. El nom de "reconeixement de patrons" ve de l'anglés *Pattern Recognition* i també és conegut com a "reconeixement de formes".

L'objectiu d'un sistema de reconeixement és assignar una etiqueta de classe a cada mostra. Els models estadístics que s'utilitzen per a classificar s'obtenen gràcies a un conjunt de mostres d'entrenament. L'entrenament d'un classificador pot ser supervisat o no supervisat, en funció de si les mostres d'entrenament estan etiquetades o no ho estan.

En el cas del reconeixement de la parla, l'entrenament d'aquests sistemes es fa mitjançant aprenentatge supervisat, perquè les mostres d'entrenament estan etiquetades. Això vol dir que es disposa de les transcripcions del que hi ha pronunciat. L'objectiu d'aquests sistemes és transciure un senyal de veu acústic.

El procés de classificació es pot formalitzar com a l'obtenció de la classe c a partir de la mostra x que maximitze la probabilitat $p(c|x)$ calculada a partir dels models estadístics:

$$\hat{c} = \underset{c}{\operatorname{argmax}}\, p(c|x) \tag{1.1}$$

Indiquem $\hat{c}$ per referir-nos a l'estimació de la classe, contraposat amb c que indica la classe real a la que assumim que pertany la mostra. En molts casos per estimar al més acuradament possible aquesta eixida s'aplica el teorema de Bayes i es reescriu la fórmula així:

$$\hat{c} = \underset{c}{\operatorname{argmax}}\, \frac{p(x|c) \cdot p(c)}{p(x)} \tag{1.2}$$

Com que $p(x)$ no varia en funció de c, es pot eliminar del denominador i el resultat segueix sent el mateix. Per tant,

$$\hat{c} = \underset{c}{\operatorname{argmax}}\, p(x|c) \cdot p(c) \tag{1.3}$$

Tant per a l'entrenament com per a la classificació de les mostres, aquestes han de passar per un preprocés, la fase més important del qual és l'extracció de característiques, on cada mostra ha d'estar representada per les seues característiques, habitualment disposades en forma de vector. Un vector de característiques x representa la mostra i conté només la informació necessària per a diferenciar-la i classificar-la. De les característiques, volem que siguen tan discriminants com siga possible sense contindre més informació que la rellevant per a classificar la mostra.

A partir de les característiques —i les etiquetes, en el cas de l'aprenentatge supervisat— de les dades d'entrenament s'entrena el sistema reconeixedor. Aquest sistema es pot fer servir posteriorment amb mostres no etiquetades, i els assignarà una etiqueta de classe automàticament, basant-se en l'equació 1.3.

1.3 Reconeixement automàtic de la parla

L'ASR té com a objectiu obtindre automàticament la transcripció escrita a partir d'un senyal acústic. El problema s'interpreta com un de reconeixement de patrons on les classes possibles són totes les frases de la llengua i les mostres són la seqüència acústica a classificar. Per tant, es pot formalitzar com a l'obtenció de la seqüència de paraules $w = w_1 w_2 ... w_N$ a partir de la seqüència de característiques acústiques $x = x_1 x_2 ... x_T$ que maximitze la probabilitat $p(w|x)$ [8]:

$$\hat{w} = \underset{w}{\operatorname{argmax}}\, p(w|x) \tag{1.4}$$

Aplicant Bayes tal i com ho hem fet en la secció 1.2, el problema es converteix en el següent:

$$\hat{w} = \underset{w}{\operatorname{argmax}}\, p(x|w) \cdot p(w) \tag{1.5}$$

L'objectiu dels models estadístics en ASR és aproximar aquestes dues probabilitats: la probabilitat de la seqüència de paraules $p(w)$ i la probabilitat que eixa seqüència genere eixos sons $p(x|w)$. Per a aquesta tasca s'empren dos models, el model de llenguatge i el model acústic respectivament. La figura 1.2 representa el procés de classificació en un sistema de reconeixement automàtic de la parla.

En les següents subseccions s'explica en què consisteixen habitualment els models acústics i de llenguatge. Fonamentalment, es dóna una visió bàsica dels models ocults de Markov per al modelatge acústic i dels n-grames per al model de llenguatge.

1.3.1 Model acústic

El model acústic serveix per a estimar la probabilitat $p(x|w)$ per a tot x. La forma habitual que té aquest model és un conjunt de models ocults de Markov, o HMM (*Hidden Markov Model*), un per cada fonema (o trifonema, com veurem en la secció 4.3).

Cal també un diccionari de pronúncia, perquè per a poder aplicar els models de fonemes cal transformar la seqüència w a fonemes, i això es fa a partir d'aquest diccionari; que és simplement una llista de cada paraula del vocabulari amb la seua representació fonètica. El diccionari de pronúncia es pot considerar que forma part del model acústic perquè serveix conjuntament amb els HMMs per a estimar $p(x|w)$.

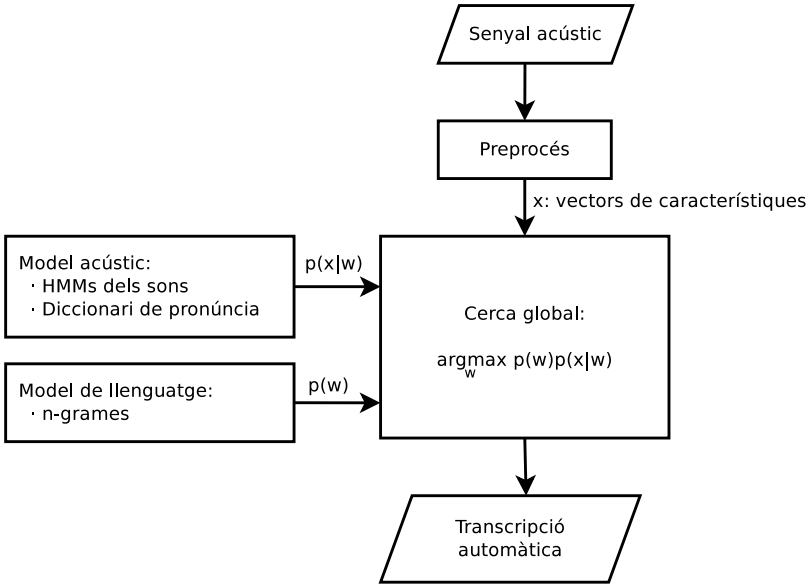

Figura 1.2: Esquema de la classificació en un sistema d'ASR

Els HMMs són una forma de definir distribucions de probabilitat sobre seqüències d'observacions. En el cas d'ASR aquestes observacions són els vectors de característiques acústiques. S'assumeix que les observacions es prenen en intervals equiespaciats, de tal forma que l'observació x_t ha estat presa en l'instant t.

Aquests models fan tres assumpcions característiques:

1. Que cada observació x_t la genera un estat S_t que és l'estat que està actiu en eixe instant i que roman ocult a l'observador.

2. Que l'estat actiu en el temps t, S_t, només depén de l'estat actiu en l'instant anterior $t-1$.

3. Que l'estat és una variable discreta: S_t pot prendre un valor concret entre els que ja estan definits al model.

Es pot veure una representació gràfica d'un HMM en la figura 1.3, on es posa com a exemple un model de quatre estats. També hi ha representats dos pseudoestats, l'inicial i el final, que no tenen emissió però que serveixen per representar les probabilitats inicials i finals de cada estat. En la figura 1.3 també estan representades les probabilitats de transició de cada estat al següent.

El que aquest tipus de models aconsegueixen és distribuir una massa de probabilitats unitària entre totes les possibles seqüències de característiques, en el nostre

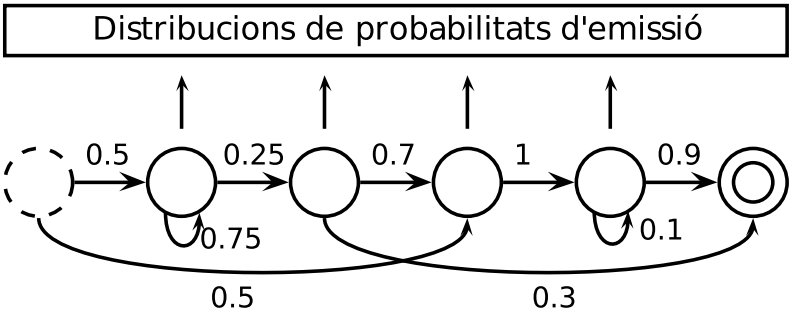

Figura 1.3: Exemple d'HMM. Cada estat té una distribució de probabilitats pròpia.

cas característiques acústiques. La probabilitat d'una seqüència d'observacions es pot factoritzar de la següent manera:

$$p(x) = \sum_{s_1,\ldots,s_T} \prod_{t=1}^{T+1} p(s_t \mid s_{t-1}) \prod_{t=1}^{T} p(x_t \mid s_t) \tag{1.6}$$

Per a veure més informació sobre els HMMs i la seua relació amb les xarxes bayesianes, un altre tipus de model estadístic més general, es pot consultar [9].

Tant TLK com Kaldi utilitzen HMMs per al modelatge acústic, però la topologia dels models —és a dir, les transicions des de cada estat— és lleugerament diferent. La forma en que cadascuna d'aquestes dues eines gasta els HMMs està descrita en els capítols 4 i 5 respectivament.

1.3.2 Model de llenguatge

El model emprat per aproximar la probabilitat $p(w)$ d'una seqüència de paraules rep el nom de model de llenguatge. S'assumeix que el llenguatge és un conjunt de seqüències de paraules, on cada seqüència té una probabilitat associada. Els models de llenguatge ajuden a guiar i a restringir la cerca entre hipòtesis alternatives de paraules. Aquests models assignen una probabilitat $p(w)$ a les seqüències de paraules $W = w_1, ..., w_n$ subjecte a:

$$\sum_{w} P(w) = 1 \tag{1.7}$$

Un model àmpliament emprat és el d'n-grames. L'assumpció que es fa és que la probabilitat d'una determinada paraula depén de les paraules que la precedeixen,

i en major grau de les immediatament anteriors. Els models d'n-grames utilitzen les $n-1$ paraules prèvies per representar la història, i les probabilitats de la paraula estan basades en recomptes de freqüència que es fan sobre unes dades d'entrenament:

$$p(w_1^T) = \prod_{t=1}^{T+1} p(w_t \mid w_{\max\{t-n+1\}}^{t-1}) \tag{1.8}$$

On w_0 és la paraula especial inicial i w_{T+1} la paraula especial final. Per a l'exemple de trigrames, la probabilitat de la paraula w_3 donat que la història —les dues paraules anteriors— és $w_1 w_2$ seria estimada de la següent manera:

$$p(w_3|w_1 w_2) = \frac{c(w_1 w_2 w_3)}{c(w_1 w_2)} \tag{1.9}$$

On $c(\bullet)$ és el recompte de vegades que apareix l'esdeveniment en les mostres d'entrenament del model de llenguatge.

La construcció de models d'n-grames a partir d'un corpus de dades escrit és relativament senzilla: només cal comptar la freqüència d'aparició de cada n-grama dins dels textos que el conformen.

Existeixen també altres tipus de model de llenguatge, com per exemple basats en xarxes neurals, en arbres de decisió o els models log-bilinials. Com veurem més endavant en els capítols 4 i 5, TLK fa ús de models de llenguatge d'n-grames en el format específic ARPA, mentre que Kaldi funciona amb transductors d'estats finits que s'obtenen a partir de models ARPA.

1.3.3 Cerca

La cerca és el reconeixement pròpiament dit. Amb els models acústics i de llenguatge, es fa una cerca dins de tot l'espai de possibilitats a partir de les dades, buscant la transcripció de màxima probabilitat. Es construeix una xarxa de reconeixement, que és un model que integra tant el model acústic com el de llenguatge i que permet computar les probabilitats de cada hipòtesi. Una d'aquestes xarxes la podeu trobar representada en la figura 1.4.

Aquesta xarxa de reconeixement en què es fa la cerca té forma d'arbre o de graf dirigit acíclic, i per tant se segueixen les mateixes estratègies que s'apliquen a aquestes estructures de dades, com poden ser el *look-ahead* o la poda. L'heurística que s'aplica en la cerca és de molta importància, ja que aquest graf pot ser considerablement gran: en principi permet la construcció de totes les frases possibles de la llengua.

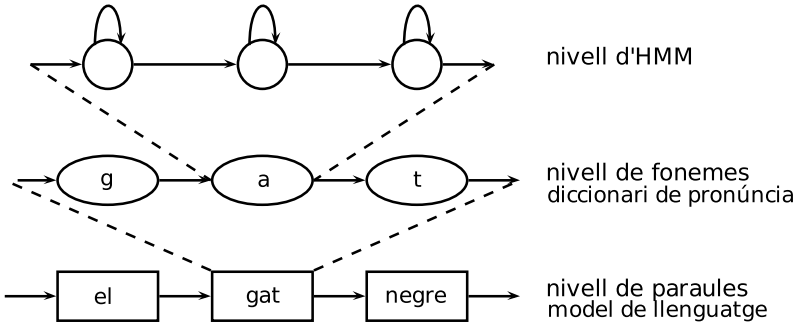

Figura 1.4: Esquema dels nivells de la xarxa de reconeixement

Per al reconeixement, es fa un aliniament de les mostres seguint l'algorisme de Viterbi. Així es calcula la probabilitat només del camí més probable — el mateix model podria emetre eixes característiques per un altre camí, però les probabilitats dels altres camins menys probables es desestimen en l'aproximació de Viterbi.

1.4 Avaluació dels resultats

L'avaluació dels resultats d'un sistema és fonamental per saber si el seu funcionament és correcte, i també per a comparar dos sistemes com és el cas d'aquest treball. Per poder comparar els sistemes cal que avaluem els seus resultats d'una forma rigorosa i sistemàtica. El que es fa és utilitzar mètriques automàtiques definides, com la *Taxa d'Error a nivell de Paraula*, o WER, de l'anglés *Word Error Rate*.

El WER és la mètrica utilitzada en aquest treball per a calcular l'error en el reconeixement. És una mètrica clàssica en l'avaluació de sistemes d'ASR, derivada de la distància mínima d'edició (o distància de Levenshtein), que quantifica la diferència entre la transcripció automàtica i una referència supervisada. La fórmula que defineix el WER és:

$$WER = \frac{I + D + S}{N} \tag{1.10}$$

On I, D i S és el nombre mínim d'operacions d'inserció, eliminació i substitució de paraules que cal fer per transformar la referència en l'eixida del sistema (*Insertions, Deletions, Substitutions* per les sigles); i N és la quantitat de paraules total de la referència. Per exemple, un WER del 23% vol dir que un 23% de les paraules estan mal reconegudes, ja siga perquè falten, perquè sobren, o perquè s'han detectat com a una altra paraula.

També és interessant en molts casos tindre en compte el temps que un sistema tarda en entrenar-se, o en reconéixer un vídeo. La mètrica que es fa servir en aquests casos és el *Factor de Temps Real*, o RTF, de l'anglés *Real-Time Factor*, que es pot definir de la següent manera:

$$RTF = \frac{t_p}{t_v} \tag{1.11}$$

On t_p és el temps que tarda l'àudio en processar-se, tenint en compte l'extracció de característiques i tots els passos de reconeixement, fins a obtindre la transcripció; i t_v és la durada de l'àudio. Per exemple, si la durada d'un àudio és d'un minut i la transcripció automàtica s'obté en dos minuts, el sistema d'ASR funcionaria a un RTF de 2.

1.5 Estructura del treball

Aquesta memòria consta de set capítols. Aquest capítol ha exposat la motivació del treball i ha introduït els conceptes bàsics necessaris de reconeixement de patrons i de reconeixement automàtic de la parla. També s'han presentat les principals mètriques que s'utilitzen per avaluar els resultats dels sistemes d'ASR.

El segon capítol tracta sobre les principals eines que s'han utilitzat en l'avaluació empírica. El capítol 3 aborda els conjunts de dades utilitzats en la construcció i avaluació dels sistemes i el seu preprocés. El capítol 4 explica la construcció del sistema i reconeixement amb TLK, pas per pas. El capítol 5 parteix dels passos esmentats en el capítol anterior per fer el mateix amb Kaldi, incidint en les principals diferències entre les dues eines. El capítol 6 compara les eines en quant a facilitat d'ús i els sistemes en quant a qualitat dels resultats i eficiència temporal. El capítol 7 conclou el treball amb un resum de tot el que s'hi ha abordat i possibles treballs futurs que se'n deriven.

L'ordre de lectura recomanat és el seqüencial. Això no obstant, hi ha capítols que poden ser obviats en cas que el lector ja hi estiga familiaritzat. És el cas del capítol 2 que exposa les principals eines utilitzades (TLK, Kaldi i SRILM) o el del capítol 4 si està familiaritzat amb la construcció i ús de sistemes d'ASR amb TLK.

CAPÍTOL 2

Eines informàtiques: TLK, KALDI i SRILM

2.1 Introducció

Aquest capítol presenta les principals eines informàtiques que s'han utilitzat al llarg del treball, amb una secció per a cadascuna d'elles. En les seccions 2.2 i 2.3 es descriuen TLK i Kaldi, les eines que es comparen en el treball per a la construcció i ús de sistemes de reconeixement de la parla. En la secció 2.4 es descriu SRILM, l'eina específica per construir els models de llenguatge. Finalment, la secció 2.5 recull les conclusions del capítol.

2.2 TLK

TLK és l'eina d'ASR desenvolupada pel grup MLLP del DSIC en el context del projecte europeu transLectures [10]. Té una llicència lliure (Apache 2.0) i es pot aconseguir des de la pàgina web `www.mllp.upv.es/tlk/`.

És una eina completa que permet fer totes les passes necessàries per a la creació i ús de sistemes ASR a partir de les dades: extracció de característiques, modelatge acústic amb HMMs de gaussianes i xarxes neurals profundes, adaptació al locutor i reconeixement. TLK es pot dividir en tres grans components: la biblioteca

(`libTLK`), que implementa el nucli de les funcionalitats; un conjunt de comandaments bàsics per a la terminal que en fa ús; i unes poques instruccions d'alt nivell que usen els comandaments bàsics per a preprocessar, entrenar i reconéixer amb el toolkit [5].

TLK ha estat emprat per a construir el sistema de transcripció automàtica de transLectures del repositori de vídeos educatius de la UPV poliMedia i del repositori VideoLectures.NET, així com per a tots els sistemes entrenats pel grup incloent sistemes ASR en català, castellà, anglés, portugués, francés, italià, alemany i eslové i altres tasques relacionades com adaptació dels models de llenguatge o revisió de transcripcions automàtiques [11] [12].

Per a l'ús de TLK en la construcció de sistemes d'ASR, s'inclouen tres eines d'alt nivell que fan ús de la resta d'eines per a preprocessar, entrenar i reconéixer les dades: tLtask-preprocess, tLtask-train i tLtask-recognise. Aquestes eines fan servir funcions de més baix nivell i són configurables, de tal forma que és més convenient utilitzar-les i no replicar la seua funcionalitat. A continuació es descriuen les funcions de cadascuna.

tLtask-preprocess serveix per preprocessar els corpus. Aquesta eina rep com a entrada senyals d'àudio amb les seues transcripcions i realitza les tasques d'extracció de característiques i d'anotació fonètica (veure l'apartat 3.5). També genera altres dades que es poden gastar per a processos com adaptació a locutor o a vídeo, o com el text original sense puntuació per a cada segment.

tLtask-train a partir de l'eixida de tLtask-preprocess, entrena els models acústics del sistema reconeixedor. En aquest treball, aquests models estan basats en HMMs amb probabilitats d'emissió modelitzades amb xarxes neurals. L'entrenament d'aquests models té diverses etapes que estan explicades en el capítol 4.

tLtask-recognise utilitza els models per a reconéixer. L'entrada que rep són els models entrenats —l'acústic obtingut amb tLtask-train i el model de llenguatge— i els àudios per transcriure; i l'eixida és la seua transcripció automàtica.

2.3 KALDI

Kaldi és un altre conjunt d'eines per a ASR de llicència lliure (Apache 2.0) que va nàixer el 2009 arran d'uns grups de treball internacionals organitzats per la universitat de Johns Hopkins. L'objectiu de Kaldi és aconseguir un codi fàcil d'entendre, de modificar i d'estendre per a la investigació en ASR [6].

Aquest altre *toolkit* també suporta extracció de característiques, entrenament de model acústic de HMMs amb gaussianes i xarxes neurals, adaptació del model acústic al locutor i reconeixement a partir dels models entrenats. Com funciona sobre transductors d'estats finits, també té la funcionalitat de convertir models de llenguatge de n-grames a transductors.

Els usos de Kaldi han estat nombrosos i diversos des de la seua aparició. Per donar alguns exemples, ha servit per a realitzar un reconeixedor *on-line* —s'anomena així el reconeixement al vol, on va reconeixent alhora que es va produint l'àudio— [13], per a fer un entrenament discriminatiu a nivell de seqüència amb xarxes neurals [14], per a aliniar àudio amb text [15] o per al reconeixement continu de la parla per al Serbi [16].

Kaldi fa ús de bibioteques externes, `OpenFST` i `BLAS/LAPACK`. La primera per a treballar amb els transductors d'estats finits i les altres dues per a fer operacions d'àlgebra lineal. Gastant aquestes biblioteques, hi ha una sèrie de mòduls propis de Kaldi i d'executables que fan servir els mòduls. Kaldi també inclou scripts que gasten aquests executables, i receptes en forma d'scripts que entrenen sistemes i els gasten per reconéixer; per donar exemples de com es pot fer servir el *toolkit* de moltes formes distintes.

2.4 SRILM

SRILM és un *toolkit* per a construir i aplicar models de llenguatge estadístics, sobretot per a gastar-los en reconeixement de la parla, etiquetatge i segmentació i traducció automàtica. Aquest ha estat desenvolupat en el *SRI Speech Technology and Research Laboratory* i està disponible lliurement per a ús no comercial. Funciona sobre Windows i sistemes operatius basats en UNIX.

Consisteix en una sèrie de components:

1. Un conjunt de classes de C++ que implementen els models de llenguatge, donen suport a estructures de dades i altres utilitats.

2. Un conjunt de programes executables que fan ús de les biblioteques C++ per a dur a terme accions com entrenar models de llenguatge i provar-los amb dades.

3. Uns quants scripts per a facilitar tasques relacionades amb els models de llenguatge.

Les funcionalitats de SRILM són la construcció i comprovació de models de llenguatge basats en n-grames, i es poden configurar moltes opcions sobre com construir-los. També pot estimar altres tipus de models, encara que la majoria estan basats sobre els n-grames com a punt de partida. Podeu veure una descripció de SRILM en [17].

2.5 Conclusions

En aquest capítol hem presentat les eines informàtiques més rellevants utilitzades al llarg del treball: TLK, Kaldi i SRILM. Les dues primeres són *toolkits* de construcció i ús de sistemes d'ASR mentre que la tercera serveix per a l'elaboració dels models de llenguatge per a ASR.

TLK és l'eina desenvolupada pel grup MLLP del DSIC, que ha estat gastada per a la transcripció dels repositoris poliMedia i VideoLectures.NET; i conté tres instruccions d'alt nivell per a abstraure l'ús de les tasques de preprocés, entrenament i reconeixement. També hem presentat Kaldi, una eina d'ASR en auge que va nàixer arran d'uns grups de treball internacionals amb l'objectiu de ser utilitzada en la investigació en ASR. Per últim, s'ha abordat SRILM, que serveix per a la construcció de models de llenguatge per a ASR.

CAPÍTOL 3

Dades i preprocés

3.1 Introducció

En aquest capítol es presenten els tres corpus que s'han utilitzat en la construcció dels sistemes i en la seua avaluació empírica, que són TECNOPARLA, Glissando i poliMedia-català; amb una secció per a cada corpus. També s'hi troba una secció que conté una descripció del preprocés que se n'ha fet en aquest treball, on principalment es detalla l'extracció de característiques acústiques, que són el mateix tipus de característiques tant en TLK com en Kaldi. L'última secció de conclusions recull les idees principals del capítol.

3.2 TECNOPARLA

Aquest corpus es va crear en la Universitat Politècnica de Catalunya (UPC) en el marc del projecte TECNOPARLA [18], un projecte amb l'objectiu de desenvolupar la tecnologia de la parla al voltant de la llengua catalana i concretament la seua aplicació a la traducció automàtica de veu.

Els vídeos que l'integren els va proporcionar la Corporació Catalana de Mitjans Audiovisuals i són 32 programes en directe del programa Àgora de la TV3. Hi

predomina el català central, una varietat del català oriental, molt per sobre de la resta. També hi ha molts participants de llengua castellana.

El corpus ha estat gastat prèviament en el context del projecte TECNOPARLA, on va aprofitar per provar una aproximació per a definir una estructura jeràrquica per a segmentació d'àudio, obtenint errors en la segmentació de 3.71% o 3.4% en funció de la mètrica [19]. També ha estat utilitzat per transcriure automàticament programes d'Àgora amb un WER del 30.2%, i del 25% entrenant amb un corpus addicional: el SPEECON-S [20].

La taula 3.1 mostra la distribució dels locutors de català en el corpus. La categoria "diversos locutors" fa referència als segments on diversos locutors parlen alhora. La distribució indica que hi ha un clar desequilibri de gènere en aquestes dades. Els segments de locutor desconegut són pocs i curts. Hi predomina el text espontani, amb característiques com frases incompletes, repeticions, falta de fluïdesa, etcètera. Per més informació sobre l'origen i la composició del corpus, podeu consultar [21].

Taula 3.1: Distribució dels locutors catalanoparlants en TECNOPARLA

Sexe	Locutors	Durada (h)	Segments
home	441	24:33	25335
dona	113	3:51	3848
diversos locutors	317	0:40	623
total	871	29:04	29806

3.3 Glissando

El corpus Glissando es va desenvolupar en el marc d'un projecte del *Plan Nacional de I+D* 2010-2012 dut a terme per tres grups de recerca: el Grup de Lingüística Computacional de la Universitat Pompeu Fabra, el Grup d'Estudis de Prosòdia de la Universitat Autònoma de Barcelona i el *Grupo de Entornos Computacionales Avanzados - Sistemas de Interacción Multimodal* de la Universidad de Valladolid. El seu objectiu era l'elaboració d'un corpus per a l'estudi de la prosòdia del català i el castellà [22].

Glissando té una llicència Creative Commons i està dividit en dos subcorpus: Glissando_ca, en català, i Glissando_es, en castellà. Cadascun d'aquests està enregistrat en estudis professionals per 28 locutors. Estan composats per 72 textos de notícies llegides per professionals de la ràdio i la publicitat i 42 textos de diàlegs. En la construcció dels sistemes amb TLK i Kaldi només s'ha utilitzat el subcorpus

de notícies. La distribució dels locutors d'aquest subcorpus està detallada en la taula 3.2.

Taula 3.2: Distribució dels locutors en el subcorpus de notícies de Glissando_ca

Id de locutor	Tipus de locutor	Sexe	Notícies	Durada
f01r	Ràdio	Dona	36	30' 16"
f02a	Publicitat	Dona	36	32' 30"
m04r	Ràdio	Home	36	28' 12"
m05a	Publicitat	Home	36	28' 20"
f07a	Publicitat	Dona	72	1h 8' 3"
m03r	Ràdio	Home	72	1h 3' 25"
m08a	Publicitat	Home	72	1h 7' 21"
Temps total				6h 23' 6"

En ser textos llegits, enregistrats en un bon entorn acústic (un estudi), amb bon material i llegits per professionals de la ràdio i la publicitat, és esperable que l'àudio siga de bona qualitat. En aquest corpus hi ha un equilibri entre els locutors masculins i femenins en haver-n'hi el mateix nombre i parlar durant aproximadament el mateix temps.

3.4 poliMedia-Català

El corpus poliMedia està extret a partir del servei de distribució de vídeos educatius de la UPV amb el mateix nom. El subcorpus format pels vídeos en català rep el nom de poliMedia-català. En aquest servei, els professors enregistren classes de més o menys 10 minuts sobre temes concrets. Aquestes classes són gravades a uns estudis especialitzats sota condicions controlades i homogènies. Aquest corpus evoluciona amb el temps, a mesura que es van enregistrant i supervisant les transcripcions dels vídeos. Una descripció de l'estat del corpus en el 2012 la podeu trobar en [23]. En la figura 3.1 es mostra una imatge d'un vídeo de la plataforma poliMedia.

D'aquest corpus s'han fet tres particions. La partició *train* contribueix a les dades d'entrenament juntament amb els altres dos corpus. La partició *dev* serveix per optimitzar paràmetres del sistema de reconeixement; i la partició *test* serveix per comprovar l'error del sistema. La taula 3.3 detalla els vídeos que formen part de cadascun d'estos conjunts.

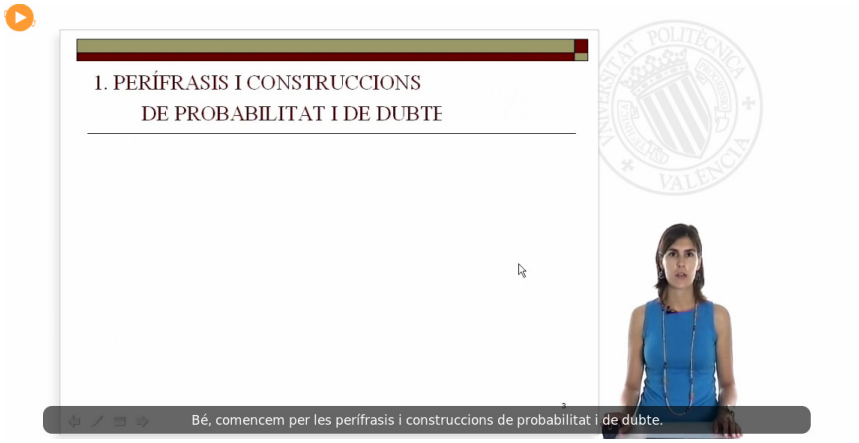

Figura 3.1: Imatge de la plataforma de vídeos educatius poliMedia

Taula 3.3: Distribució dels vídeos de poliMedia-català en els conjunts *train*, *dev* i *test*.

conjunt	vídeos	segments	durada (h)
train	177	11009	18:00:56
dev	17	1329	2:07:50
test	16	1254	1:50:28
total	210	13592	21:59:14

3.5 Preparació i preprocés de les dades

Abans de fer servir tant Kaldi com TLK, cal preparar les dades dels corpus per tal que estiguen en el format que cada eina necessita. Cada corpus està organitzat d'una forma diferent, i per això s'han de processar de forma específica. L'entrada que esperen TLK i Kaldi està descrita als capítols 4 i 5 respectivament.

Els objectius del preprocés de les dades són l'extracció de característiques de les mostres acústiques i la transcripció fonètica de les pronunciacions. A continuació s'explica en què consisteixen aquests dos processos.

Les característiques de les mostres acústiques s'extrauen a partir de l'àudio, de tal forma que per cada x mil·lisegons s'obté un vector de característiques. Aquest vector s'obté a partir de les ones de so, tenint en compte un fragment de temps (anomenat finestra) normalment de més longitud que el període entre vector i

vector. El procés descrit està representat en la figura 3.2. El resultat és una seqüència de vectors de característiques.

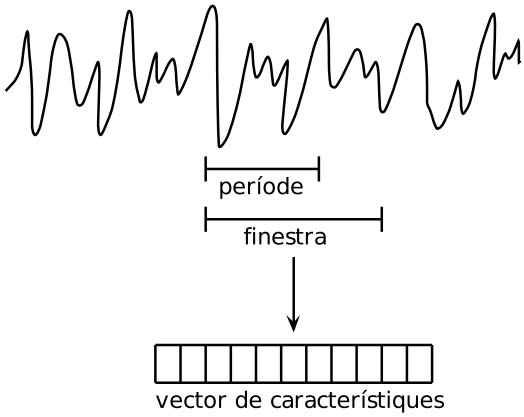

Figura 3.2: Extracció d'un vector de característiques per cada període

Per obtindre cada vector de característiques, es treballa sobre l'ona seguint una sèrie de passos. Els més rellevants són:

1. **Transformada de Fourier.** La senyal és convertida al domini de la freqüència per a treballar amb amplituds d'ona. Així s'obtindria un vector amb amplituds per a cada freqüència.

2. **Filtre de Mel.** Per a un millor reconeixement, empíricament s'ha demostrat que cal més resolució en freqüències baixes i menys en freqüències més altes. El filtre de Mel ho aconsegueix amb sumes ponderades i els coeficients que obté estan equiespaciats en una escala logarítmica, l'escala de Mel, definida per:

$$\mathrm{Mel}(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{3.1}$$

   La figura 3.3 representa com pondera cada freqüència en la suma que s'acumula en cada coeficient. Cada coeficient correspon a un dels triangles.

3. **Característiques cepstrals.** Les amplituds del banc de filtres tenen molta correlació i per tal d'obtindre característiques discriminatives amb coeficients el menys correlats possible cal aplicar-los una transformació cepstral. Als logaritmes de cada amplitud (m_j) se'ls aplica la transformada discreta del cosinus:

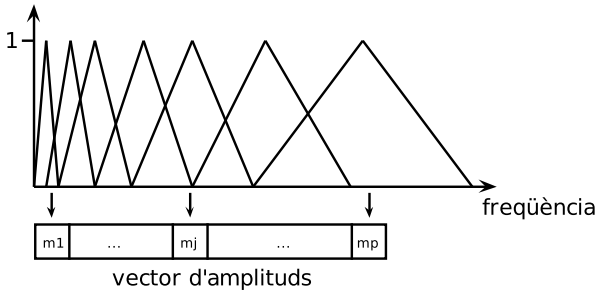

Figura 3.3: Banc de filtres de Mel

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} \log(m_j) \cos\left(\frac{\pi i}{N}(j - 0.5)\right) \tag{3.2}$$

El que s'obté finalment és un vector de característiques MFCC (*Mel-Frequency Cepstral Coefficients* en anglés). El resultat de l'extracció de característiques és un vector d'aquest tipus per cada període en la mostra d'àudio original.

4. **Normalització de mitjana i variància.** Una vegada s'han obtingut les característiques MFCCS, es pot normalitzar la seua mitjana i variància per cada cluster. En el nostre cas definim que un cluster és un locutor en un vídeo, i es normalitzen les mostres per cada locutor de cada vídeo; i així obtenim finalment les característiques que utilitzarem per entrenar el model acústic.

A més de l'extracció de característiques acústiques, el preprocés també ha de generar una transcripció del text original a nivell fonètic. Com el primer pas d'entrenament és construir un HMM de tres estats per cada fonema (veure secció 4.2), l'eixida del preprocés ha de ser, per cada segment, tant la seqüència de vectors de característiques acústiques com la seqüència de fonemes que s'hi pronuncien.

La transcripció a monofonemes és feta per un transliterador automàtic de català. Per posar un exemple del que genera aquest transliterador, a continuació tenim un segment de frase en text original i transcrit a monofonemes:

> inclou una introducció a la lectura d'una selecció de poemes i els comentaris anteriors i posteriors
>
> SP i n k l o u SP u n a SP i n t r o d u k s i o SP a SP l a SP l e k t u r a SP d u n a SP s e l e k s i o SP d e SP p o e m e s SP i SP e l s SP k o m e n t a r i s SP a n t e r i o r s SP i SP p o s t e r i o r s SP

3.6 Conclusions

En aquest capítol hem presentat els corpus de dades utilitzats en la construcció i avaluació dels sistemes de reconeixement de la parla comparats en el treball. Els tres corpus són TECNOPARLA, Glissando i poliMedia-català. TECNOPARLA és el corpus de la UPC construit a partir del programa Àgora de la TV3. Glissando és un projecte entre tres universitats per a construir un corpus per a l'estudi de la prosòdia del català i el castelllà. poliMedia-català són vídeos educatius de la UPV en català transcrits d'al voltant de 10 minuts.

També hem descrit el preprocés que se'n fa, centrant-nos en l'extracció de característiques a partir de l'ona de so. Hem vist que per cada període s'extrau un vector de característiques, passant per les fases de: transformada de Fourier, filtre de Mel, transformació cepstral; i que finalment es normalitzen en mitjana i variància per cada locutor de cada vídeo. Hem explicat que, a més, el preprocés inclou la transliteració a monofonemes de les transcripcions supervisades.

CAPÍTOL 4

Sistema TLK de reconeixement de la parla en català

4.1 Introducció

Aquest capítol descriu l'avaluació empírica realitzada amb TLK. Com que la intenció és que els processos i els resultats siguen comparables, s'han fet els mateixos passos en TLK i en Kaldi. Per tant, es descriu la construcció d'un sistema d'ASR i el seu ús per a reconeixement amb TLK i en el capítol 5, sobre Kaldi, es fa referència al que es diu ací per a les explicacions compartides.

Per a parlar sobre la construcció i ús d'un sistema d'ASR, val la pena recordar la figura 1.2 vista en la introducció. Hi intervenen uns quants processos i models: el preprocés, els models acústics, el diccionari de pronúncia, el model de llenguatge i la cerca. Així, aquest capítol conté una secció per al preprocés (4.2), tres per a l'entrenament dels models (4.3, 4.4 i 4.5) i una per al reconeixement (4.6). La secció 4.7 conté una descripció de l'avaluació empírica realitzada amb TLK. Per finalitzar, la secció 4.8 tanca el capítol amb un resum de les idees principals exposades.

4.2 Preprocés de les dades

En TLK, el preprocés de les dades es fa mitjançant `tLtask-preprocess`. Aquesta ordre realitza les fases del preprocés explicades en la secció 3.5: extracció de característiques i transcripció de la part escrita a monofonemes, i també altres processos secundaris. `tLtask-preprocess` treballa sobre un directori d'entrada que conté tots els àudios en format .wav i les transcripcions respectives amb fitxers .trs o .dfxp; cada transcripció ha de tindre el mateix nom que el seu àudio, i totes les transcripcions de la carpeta han d'estar en el mateix format, ja siga trs o dfxp.

En la carpeta d'eixida genera les mostres del preprocés i també llistes dels fitxers que contenen aquestes dades per poder ser llegides posteriorment. La figura 4.1 representa l'estructura bàsica de l'entrada i l'eixida de `tLtask-preprocess`, ometent alguns elements secundaris.

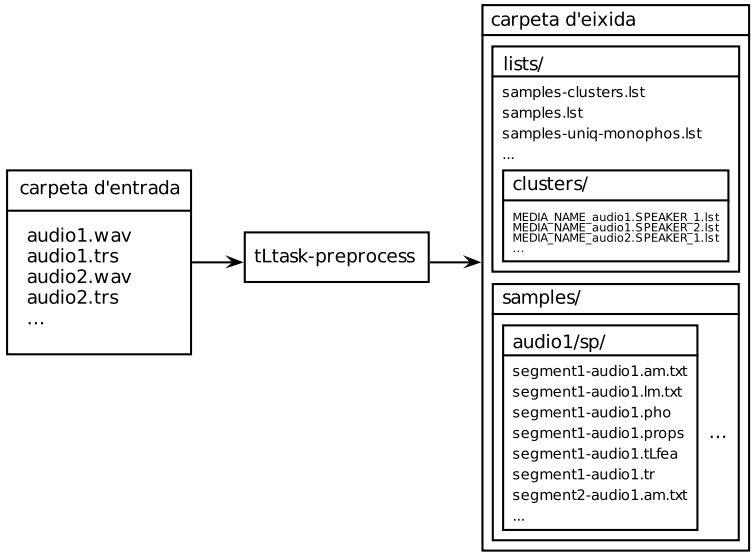

Figura 4.1: Esquema de l'entrada i l'eixida de `tLtask-preprocess`

Dins de samples, per cada vídeo crea un directori del vídeo, on per cada segment hi genera 6 fitxers, tots amb el mateix nom identificatiu del segment però amb extensions diferents. Els rellevants per a l'entrenament del model acústic són els d'extensió .tLfea, que contenen les característiques (*features* en anglés), i els d'extensió .pho, que contenen la transliteració del segment a monofonemes.

En l'avaluació empírica s'ha cridat a `tLtask-preprocess` una vegada per al corpus Tecnoparla, una altra per a Glissando i tres per a PoliMedia-Català — una per

cada partició: train, dev i test. Per posar un exemple de crida a aquesta funció, a continuació podeu veure la que ha servit per preprocessar Tecnoparla:

```
$ EXTRA_TLEXTRACT_OPTS="-C 15" tlk161/bin/tLtask-preprocess \
                                          -norm spk ca trs trs mfccs48
```

Aquesta crida indica que s'han utilitzat 15 MFCCs, que s'han normalitzat les mostres MFCC per mitja i variança per cada locutor (`-norm spk`), que la llengua és el català (`ca`), que el format d'entrada de les transcripcions és trs (`trs`) i que les carpetes d'entrada i d'eixida són trs i mfccs48 respectivament. La mateixa configuració han tingut els preprocessos de Glissando i PoliMedia-Català, exceptuant el format de les transcripcions d'aquest darrer que és dfxp. Per a l'extracció de característiques s'han utilitzat els valors de període i de finestra per defecte de TLK: 10ms i 25ms respectivament.

4.3 Entrenament d'un sistema estàndard d'ASR

En la construcció d'un sistema d'ASR és fonamental la construcció de models acústics, que permeten aproximar a partir de les característiques acústiques quins fonemes s'estan pronunciant en cada moment.

L'entrenament del sistema consisteix a determinar els paràmetres dels models a partir de les dades d'entrenament. Generalment, com més dades d'entrenament i de més qualitat es gasten, més acuradament s'estimen els paràmetres i més bons són els models resultants. Aquest entrenament passa per una sèrie de fases, on cadascuna genera un model i es treballa sobre l'anterior per construir el següent de forma iterativa.

La modelització acústica en totes les fases és un conjunt de HMMs de tres estats; un per cada fonema. Podem veure una representació d'un d'aquests models en la figura 4.2, on també es pot veure la seua topologia (les transicions definides entre els estats). Es pot interpretar com a que el primer estat representa l'inici de la pronúncia del fonema, el segon estat l'emissió del fonema i el tercer estat el final del fonema. L'únic model que té una topologia distinta és el del silenci, també representat en la figura 4.2, que només consta d'un estat i té definida una transició directa a l'estat final que evitaria l'emissió de cap característica acústica.

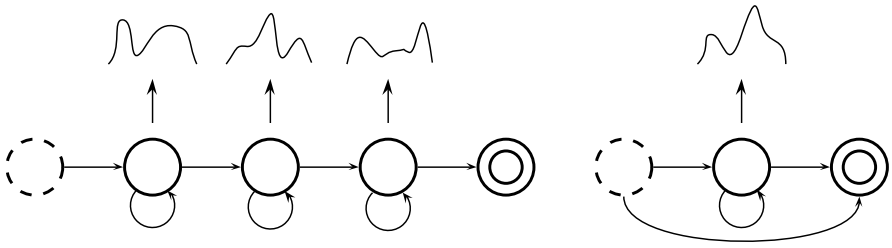

Figura 4.2: A l'esquerra, HMM de tres estats per a un fonema. A la dreta, HMM d'un estat per al silenci.

Cada estat té una distribució de probabilitats d'emissió associada que està representada per una mixtura de gaussianes. La probabilitat $b_j(o_t)$ que un estat j genere una observació (vector) o_t ve donada per la següent fórmula:

$$b_j(o_t) = \sum_{m=1}^{M_{js}} c_{jsm} \mathcal{N}(o_{st}; \mu_{jsm}, \Sigma_{jsm}) \tag{4.1}$$

on M_{js} és la quantitat de components de la mixtura, c_{jsm} és el pes de la component m-èsima i $\mathcal{N}(\cdot\,; \mu, \Sigma)$ és la distribució gaussiana de mitjana μ i matriu de covariàncies Σ.

Els paràmetres a estimar durant l'entrenament són els de la distribució de probabilitats d'emissió (mitjana i variància) de cada gaussiana de les mixtures i també les probabilitats de transició entre els estats.

Al llarg de l'entrenament d'un model acústic, es construeixen iterativament una sèrie de models d'HMMs, cadascun més refinat que l'anterior i partint de l'anterior. A continuació es descriu el procediment d'entrenament d'un sistema estàndard d'ASR.

1. **Entrenament del model de monofonemes.** El model de monofonemes és un conjunt de HMMs on cadascun modelitza un únic fonema. Per entrenar els HMMs, TLK segueix un procés iteratiu on a cada iteració s'hi aplica l'algorisme Baum-Welch, del qual s'expliquen les idees fonamentals a continuació:

   A partir d'un segment del qual es té l'àudio i la transcripció a monofonemes es crea un model de Markov sintètic que és la concatenació dels models de cadascun dels fonemes. Amb aquest model, el model de la frase, es calcula l'ocupació de cada estat per a cada vector MFCC. Aquesta ocupació és la

probabilitat que té eixe vector MFCC d'haver estat emés per eixe estat, i es calcula amb un algorisme de programació dinàmica anomenat *forward-backward*. D'aquesta forma es calcula l'ocupació de cada estat per cada vector MFCC de totes les dades d'entrenament.

Les mixtures de gaussianes d'aquest model només tenen una component, és a dir que la distribució de probabilitats d'emissió és una única gaussiana. L'estimació dels paràmetres de cada gaussiana és molt senzilla: la mitjana és la de les seues mostres, i la variància també; i les mostres estan ponderades per la seua ocupació d'eixe estat, és a dir, per la probabilitat que tenen d'haver-hi estat emeses.

Es pot repetir el procés fins aconseguir la convergència en algun moment, i està garantit que a cada iteració la versemblança del model —la probabilitat que eixe model haja emés eixes mostres— serà major o igual que en la iteració anterior; però en la pràctica es fixa un nombre d'iteracions que en el cas de TLK són 8 per tal que acabe en un termini raonable.

2. **Entrenament del model de trifonemes.** La pronúncia d'un fonema sovint es troba contaminada pels que l'envolten. Els models de trifonemes tenen en compte cada fonema per quin altre va precedit i quin altre li succeeix, d'ací el seu nom: té en compte tres fonemes.

   Les probabilitats de transició de tots els HMMs que tenen com a centre el mateix fonema són compartides; per exemple els trifonemes de la "a" tindrien tots les mateixes transicions però distintes emissions en els estats.

   Per a l'entrenament dels trifonemes, s'inicialitzen cadascun com al seu monofonema corresponent (entrenat en la fase anterior) i se'ls aplica també l'entrenament forward-backward; i les mixtures del model de trifonemes tenen també només una component.

3. **Entrenament del model de fonemes lligats.** Com que les possibilitats de trifonemes són enormes, hi ha trifonemes que apareixen molt poques vegades en les dades d'entrenament i n'hi haurà d'altres que ni tan sols s'hi trobaran. Per tant cal crear un model més reduït que puga entrenar correctament els trifonemes que són molt poc freqüents, i això es fa compartint dades dels models per a distints trifonemes.

   Ja havíem dit que en el cas dels trifonemes tots els models que comparteixen el mateix fonema central tenien les mateixes probabilitats de transició entre estats. Amb els fonemes lligats, poden també compartir estats i fins i tot compartir el mateix HMM sencer. Un algorisme heurístic s'encarrega de "nugar" els estats semblants en primera posició, en segona i en tercera per a cada fonema central. En la figura 4.3 podeu veure una representació d'aquesta idea.

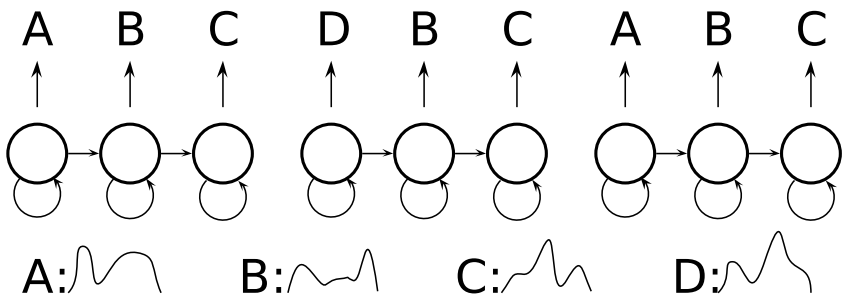

Figura 4.3: Models de fonemes lligats: amb estats compartits per al mateix fonema central. Es representen tres models del mateix fonema central, dos són idèntics a pesar de representar trifonemes distints i l'altre hi comparteix dos estats.

Tot això permet que els trifonemes poc freqüents s'entrenen en conjunt amb altres que són semblants a ells, i amb més dades s'obtenen models més acurats; i que els trifonemes que no figuren en les dades d'entrenament tinguen un model entrenat a partir dels altres igualment.

En el cas dels fonemes lligats, com és el model acústic final, s'incrementa el nombre de components, aplicant un altre algorisme que té en compte les aparicions de cada gaussiana de la mixtura per a duplicar-les iterativament fins arribar a 64 gaussianes per cada mixtura.

Qualsevol d'aquests models pot servir per reconéixer segments d'àudio i transcruire automàticament el que s'hi diu. El model de trifonemes funciona millor que el de monofonemes, i el de fonemes lligats amb mixtures de gaussianes encara millor. No obstant això, aquests models serveixen sobretot per inicialitzar l'aprenentatge d'altres models. La següent secció parla de la construcció de models acústics adaptats a la veu del locutor, i l'altra de com gastar xarxes neurals profundes com a models de distribució de probabilitats d'emissió dels vectors de característiques.

L'entrenament del model estàndard amb TLK es fa mitjançant tLtask-train, que fa ús de les utilitats de més baix nivell que inclou el toolkit per a construir els models. Per a gastar tLtask-train, cal crear primer un fitxer de configuració per a definir les opcions de l'entrenament. La forma més directa és obtindre un fitxer per defecte i editar-lo, fent ús del comandament:

```
$ tLtask-train --write-example-config-file > train.conf
```

tLtask-train crea tres directoris: `models`, on guarda tots els models acústics que va generant; `log`, on emmagatzem els registres del procés; i `tmp`, un directori temporal que s'esborra en acabar l'execució.

Si en el fitxer de configuració indiquem que només volem l'entrenament del model estàndard, en la carpeta `models` ens trobarem els fitxers:

standard.monophone_I01.model Model acústic de monofonemes amb una única gaussiana per estat.

standard.triphoneme_I01.model Model acústic de trifonemes amb una única gaussiana per estat.

standard.triphoneme_I01.occs Recompte de aparicions de cada estat del model de trifonemes en les dades. Aquest recompte serveix per a la transformació que crea el model de fonemes lligats.

standard.tied_init_I01.model Model acústic de fonemes lligats obtingut a partir de la transformació.

standard.tiedphoneme_I<number>.model Model acústic de fonemes lligats on cada estat emet probabilitats amb una mixtura de gaussianes. El número que hi ha darrere de la I indica quantes gaussianes conté la mixtura.

standard.tlist Llista de correspondències entre trifonemes i fonemes lligats. Cada línia indica quin fonema lligat modelitza cada trifonema.

<name>.mixcounts Recompte de vegades que cada gaussiana ha ocorregut a l'hora d'estimar el seu model acústic corresponent. Serveix per a l'algorisme que duplica el nombre de gaussianes d'un model.

4.4 Adaptació del model acústic al locutor

Aquesta secció tracta de com TLK pot tindre en compte que cada locutor té un registre de veu diferent per construir models adaptats al locutor amb la tècnica CMLLR (*Constrained Maximum Likelihood Linear Regression*).

Aquest procediment es basa en la idea d'aplicar transformacions lineals per reduir el desajust entre el model i les dades d'un locutor en particular. Com són transformacions lineals, es poden implementar com una multiplicació de matriu per vector per aplicar-la als vectors de característiques i no als models. Per cada locutor de cada vídeo es calcula una matriu CMLLR pròpia i se'n transformen els vectors de característiques premultiplicant-la-hi.

Així s'obtenen a partir dels vectors MFCC les característiques CMLLR, que conceptualment es poden entendre com a parcialment lliures del soroll introduït pels trets singulars de la veu de cada locutor, és a dir una mena de normalització de les mostres.

Més formalment, l'efecte de les transformacions és un canvi en els vectors de característiques tal que cada estat del HMM té més probabilitats de generar aquestes dades adaptades. El procediment és una regressió lineal de màxima versemblança, tal i com diu el nom CMLLR, i per fer aquesta regressió s'entrena una matriu específica per cada locutor de cada vídeo.

L'entrenament de la matriu CMLLR d'un locutor usa l'algorisme Baum-Welch (explicat en la secció 4.3) per modificar paràmetres a cada iteració. Els paràmetres que s'entrenen són els d'una matriu que premultiplica les mitjanes i variàncies de les distribucions de probabilitats d'emissió de cada estat. S'empra el model de fonemes lligats amb estats d'una gaussiana per entrenar aquests paràmetres.

La C (*Constrained*) de CMLLR fa referència al fet que és la mateixa matriu la que premultiplica les mitjanes i les variàncies. I és això el que permet que una vegada entrenada aquesta matriu es puguen premultiplicar les mostres per la seua inversa obtenint exactament els mateixos resultats.

Quan ja s'han obtingut les característiques CMLLR de cada locutor, es torna a repetir el procés descrit per al model estàndard: entrenament de monofonemes, de trifonemes i de fonemes lligats (secció 4.3); amb aquestes característiques. El resultat és un model reconeixedor per a les característiques CMLLR.

Aquests models, com els estàndards, també es poden utilitzar per reconéixer dades noves; però abans caldria transformar-ne les característiques. Per a fer la transformació CMLLR cal tindre la transcripció, per tant es fa el que anomenem "reconeixement en dos passos": el primer pas reconeix utilitzant models estàndards per obtindre una primera transcripció aproximada, i el segon utilitza aquesta aproximació per fer la transformació CMLLR i reconéixer amb models adaptats. Així i tot, en aquest cas aquests models no són tampoc els finals: en la pròxima secció es descriu la introducció de xarxes neurals en els models acústics.

Amb TLK, aquest entrenament també es fa amb tLtask-train. De fet, la configuració estàndard d'aquesta eina per defecte fa entrenament del model estàndard i CMLLR. En finalitzar l'execució, en la carpeta `models` trobarem, a més dels fitxers descrits en la secció anterior, els models CMLLR:

cmllr.monophone_I01.model

cmllr.triphoneme_I01.model

cmllr.triphoneme_I01.occs

...

4.5 Xarxes neurals profundes

Les xarxes neurals profundes o DNN (*Deep Neural Networks* en anglés) han demostrat empíricament ser models que funcionen millor en el reconeixement comparats amb utilitzar solament les mixtures de gaussianes. TLK ofereix la possibilitat d'entrenar el que s'anomenen models híbrids: HMMs on les probabilitats d'emissió dels estats estan modelitzades per xarxes neurals en compte de mixtures de gaussianes.

L'entrenament d'aquest tipus de model híbrid es fa substituint les mixtures de gaussianes dels estats dels models de fonemes lligats per xarxes neurals profundes; i tornant a entrenar el model. Les xarxes neurals tenen el mateix paper que les mixtures de gaussianes: definir les probabilitats d'emissió dels estats.

En l'entrenament de les xarxes neurals TLK entrena a la Viterbi: en compte de calcular l'ocupació de cada estat, per a cada mostra es pren només l'estat més probable d'haver-la emés com a 100% de probabilitat. Això vol dir que no s'hi calcula per cada mostra l'ocupació de cada estat, sinó que es fa el que es diu un aliniament, on a cada mostra se li assigna l'estat que més probabilitats té d'haver-la emés. El primer d'aquests aliniaments es fa a partir dels models de fonemes lligats de mixtures de gaussianes, i a partir de la primera iteració de l'algorisme de Viterbi ja es gasten xarxes neurals.

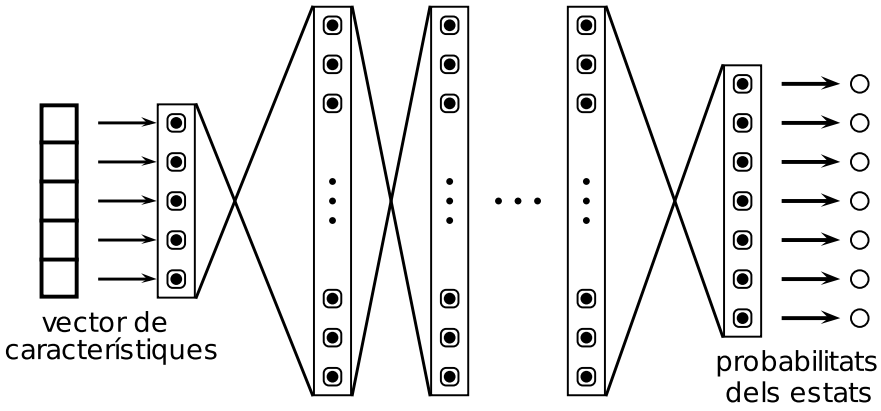

Figura 4.4: Xarxa neural profunda que modelitza $P(q|x)$

És important el fet que les mixtures de gaussianes modelitzen directament $P(x|q)$, la probabilitat de la mostra x donat un estat q; mentre que les xarxes donen $P(q|x)$, la probabilitat de l'estat q donada la mostra x, tal i com queda reflectit en la figura 4.4. Com que cal l'altra probabilitat per operar amb els HMMs, es pot aplicar Bayes:

$$P(x|q) = \frac{P(q|x)P(x)}{P(q)} \tag{4.2}$$

En aquesta equació, $P(x|q)$ ve donada per la xarxa neural, i $P(q)$ es pot calcular obtenint les voltes per les que es passa per l'estat q en totes les mostres d'entrenament. La complicació la tindríem amb $P(x)$, però no cal tindre-la en compte: com que volem el camí de màxima probabilitat per a una seqüència, i $P(x)$ és constant per a cada mostra de la seqüencia, multiplicar per la seua probabilitat no afecta al resultat i per tant es pot ignorar. Per tant, podem actuar com si:

$$P(x|q) \sim \frac{P(q|x)}{P(q)} \tag{4.3}$$

Si volem treballar amb CMLLR, cal entrenar dos d'aquests models híbrids: un per al model estàndard i un altre per a l'adaptat; perquè per a poder transformar les característiques originals en CMLLR cal disposar d'una transcripció, i el model estàndard pot gastar-se per obtindre'n una aproximació primera. Aquest procés està explicat en la secció 4.4.

Aquests últims models són els gastats en el que és el sistema de reconeixement final. En acabar l'entrenament, tenim els models amb gaussianes de monofonemes, trifonemes i fonemes lligats, i els que gasten xarxes neurals profundes de fonemes lligats; tant per a característiques MFCC com CMLLR.

L'entrenament realitzat amb TLK, amb tLtask-train, ha seguit els passos descrits en aquestes tres seccions (4.3, 4.4 i 4.5) sobre entrenament dels models acústics: construcció del model estàndard, de l'adaptació al locutor i la integració de les xarxes neurals profundes en els models acústics d'HMMs. Els tres models d'eixida de l'entrenament són:

1. El model estàndard amb xarxes neurals profundes, per fer una primera aproximació a la transcripció.

2. El model de fonemes lligats d'una sola gaussiana, per a la transformació de les característiques en CMLLR.

3. El model CMLLR amb xarxes neurals profundes, per a fer la transcripció definitiva.

Els canvis més rellevants sobre la configuració estàndard de l'entrenament són:

- La quantitat de coeficients de cada vector de característiques és 48, en compte dels 39 per defecte.

- El nombre de gaussianes de la mixtura del model de fonemes lligats és 64, i no els 128 per defecte. Això és perquè en el model final les probabilitats d'emissió seran modelitzades amb xarxes i no amb mixtures de gaussianes i per tant no cal un model tan precís amb gaussianes.
- La topologia de la xarxa neural té 4 capes i no les 5 per defecte.

Amb el fitxer de configuració creat, cridant a tLtask-train s'ha entrenat el sistema classificador a partir de les dades obtingudes en el preprocés. La crida a tLtask-train ha sigut com aquesta:

```
$ tLtask-train train.conf --log-folder log
```

Amb els models entrenats, passem a la següent etapa de l'experiment: el reconeixement automàtic de la parla utilitzant aquests models que han estat construïts.

4.6 Reconeixement

Ja entrenats els models acústics i si tenim també el model de llenguatge, podem provar a reconéixer dades noves. Els models acústics són els HMMs dels fonemes que s'han obtingut amb l'entrenament, en aquest cas per mitjà de tLtask-train, i el diccionari de pronúncia.

El diccionari de pronúncia s'ha obtingut amb el transliterador del que es parlava en la secció 3.5 sobre preprocés de les dades; i el model de llenguatge són les probabilitats de cada n-grama que s'han obtingut amb el recompte d'aparicions en els textos que han servit d'entrenament, tal i com està descrit en 1.3.2.

El reconeixement emprat és un reconeixement de dos passos: primer es reconeixen les dades mitjançant el model estàndard, i amb la transcripció aconseguida es transformen les característiques a CMLLR per a reconéixer amb el model adaptat.

Per a optimitzar el sistema, primer es reconeixen les dades del conjunt dev (de *development*) fent una exploració de paràmetres i escollint els que minimitzen l'error. Amb el sistema optimitzat es reconeixen les dades de test, i l'error del sistema es mesura sobre els resultats obtinguts en aquest conjunt de dades.

En aquesta secció es descriu el procés de la cerca —que és un pas de reconeixement pròpiament dit— en els models; els dos passos de reconeixement que s'han gastat en aquest treball; com s'ha utilitzat TLK per a fer el reconeixement i els resultats que s'han obtingut.

Dins d'un pas de reconeixement el procés està guiat per una xarxa de reconeixement, esquematitzada en la figura 1.4 en la introducció de la memòria. Aquesta

xarxa es construeix a partir del model de llenguatge, el diccionari de pronúncia i els models acústics; i consisteix en un conjunt de nodes connectats per arcs, on cada node és bé un HMM o bé un node especial de final de paraula. Aquests nodes i arcs formen una estructura que es pot veure a nivell d'estat d'HMM, de fonema, o de paraula.

Tot això serveix per fer un pas de reconeixement. Adés hem dit que el sistema construit és un sistema de dos passos. Això és perquè per tal de transformar les característiques MFCC a CMLLR l'algorisme que ho fa necessita tindre la transcripció dels segments: es fa el primer pas de reconeixement gastant els models estàndards i un segon pas amb els models adaptats. El procediment és el següent:

1. **Primer pas:** reconeixement amb models acústics no adaptats al locutor. Es gasten els models estàndards de xarxes neurals profundes per a obtindre una transcripció a partir de la qual poder fer la transformació CMLLR.

   En aquest cas es pot dir que l'aprenentatge de les matrius d'aquesta transformació és un aprenentatge no supervisat, donat que inicialment no es coneix la transcripció del segment. La transcripció automàtica que s'obté conté errors. Així i tot, adaptar-se al locutor amb una transcripció automàtica, si no conté massa errors, funciona millor que no adaptar-se.

2. **Transformació de les característiques MFCC en CMLLR.** Aquest procés és idèntic al que es fa per a l'entrenament, només que amb transcripcions automàtiques que poden contindre errors. Es gasten els models de trifonemes de gaussianes amb una única component per mixtura tal i com s'ha descrit en la secció 4.4, aplicant exactament el mateix procediment.

3. **Segon pas:** reconeixement amb models acústics adaptats al locutor. Aquests models són els que estan entrenats amb característiques CMLLR. Es gasten també els models de xarxes neurals profundes, però aquesta vegada els que han estat entrenats amb les característiques adaptades.

En TLK, la instrucció que permet fer el reconeixement és tLtask-recognise. Aquest comandament fa el primer pas amb els models estàndards, estima les matrius i característiques CMLLR amb la transcripció obtinguda, i reconeix amb els models adaptats gastant aquestes característiques. Aquesta instrucció crida a les utilitats de TLK de més baix nivell per a reconéixer.

tLtask-recognise crea dos directoris: `log` i `recognition`. En la primera emmagatzema els registres dels comandaments que executa, i en la segona guarda la transcripció, les eixides i altres fitxers auxiliars. Per a gastar tLtask-recognise, cal crear un fitxer de configuració i editar-ne les opcions, tal i com es fa en tLtask-preprocess.

Hi ha dos paràmetres clau que determinen com es puntuen les hipòtesis en la xarxa de reconeixement: el GSF o *Grammar Scale Factor* que pondera el pes del model de llenguatge, i el WIP o *Word Insertion Penalty* per a regular la quantitat de paraules, ja que sense penalització els sistemes tendeixen a inserir massa paraules.

Per a escollir els millors paràmetres GSF i WIP es gasta el conjunt `dev` (veure secció 3.4) per a explorar els valors dels paràmetres; i amb els valors que proporcionen un menor error es fa la comprovació definitiva amb el conjunt `test`. Es fa així per tal de comprovar l'error del sistema reconeixedor amb els paràmetres GSF i WIP ja establerts.

4.7 Experiments

Per a l'avaluació empírica de TLK s'ha realitzat un estudi experimental. Aquest estudi ha consistit en l'entrenament d'un conjunt de models a partir de les dades descrites en el capítol 3: els corpus TECNOPARLA, Glissando i una partició del poliMedia-català.

S'ha passat per tots els passos d'entrenament descrits en aquest capítol, i per tant s'han obtingut tots els models que s'hi han descrit: el model de monofonemes, el de trifonemes, el de fonemes lligats i el model híbrid; tant per al sistema estàndard com per a l'adaptat. En la valoració inicial s'han fet servir els sistemes híbrids estàndard i adaptat per a fer els dos passos de reconeixement descrits en la secció anterior.

Per a l'exploració de GSF i WIP s'ha utilitzat la partició `dev` del corpus poliMedia-català descrita en l'apartat 3.4. Amb la combinació òptima en `dev` s'hauria de realitzar el mateix reconeixement sobre el conjunt `test`. Això no obstant, s'ha replicat l'exploració també amb aquest conjunt per observar el grau de correlació entre els resultats en els dos conjunts. En la gràfica de la figura 4.5 es representen aquests resultats.

Es pot observar una forta correlació entre les dues exploracions, però no és una correlació total. La combinació òptima en `dev` és GSF=11 i WIP=18, amb un WER del 24.1%. Amb aquests paràmetres, el reconeixement en `test` produeix una transcripció amb un WER del 22.2%, i aquest és el resultat que es pren com a vàlid del sistema, encara que no és l'òptim de la seua exploració.

Podreu veure més resultats en el capítol 6 de comparació dels sistemes, juntament amb els resultats d'una bateria de proves dirigida a establir les causes de les diferències de resultats entre els sistemes entrenats amb TLK i Kaldi.

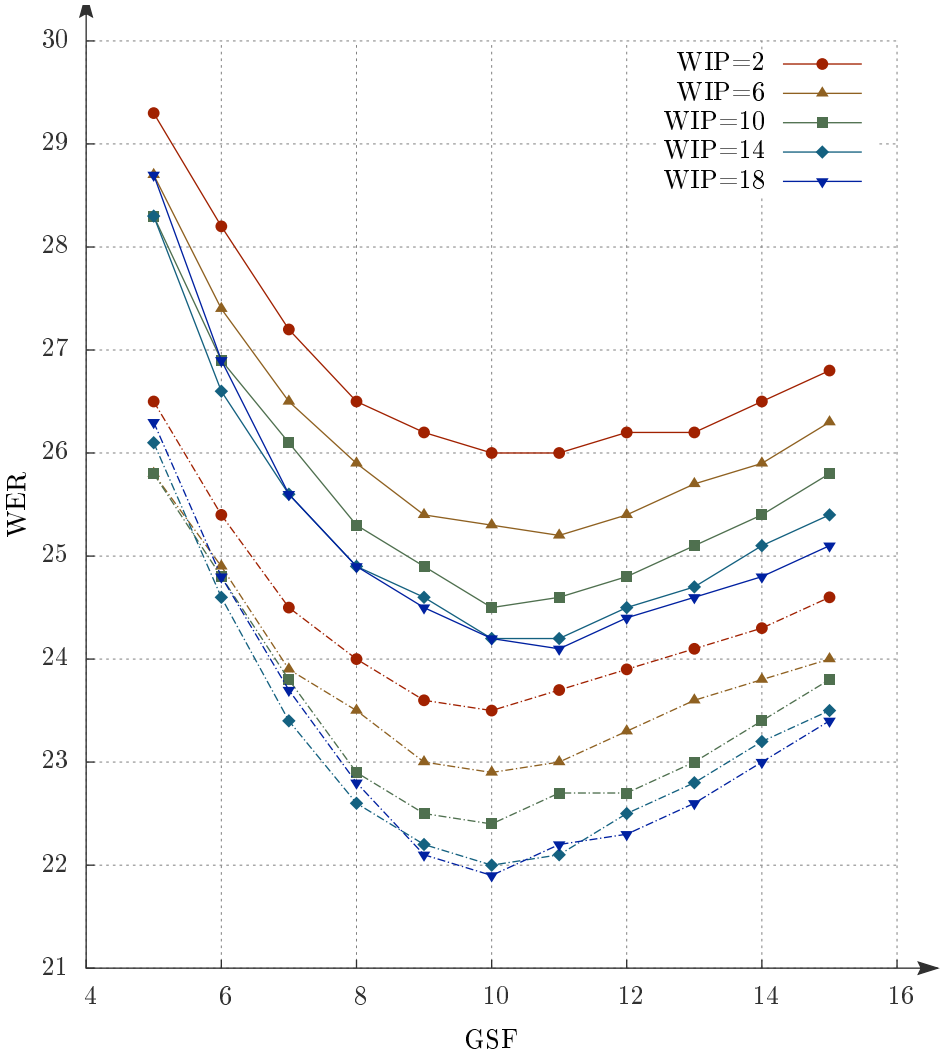

Figura 4.5: WER (%) en funció del GSF i WIP, per al `dev` (traç continu) i el `test` (traç discontinu).

4.8 Conclusions

En aquest capítol s'ha descrit el procés d'ús de TLK en la construcció i ús de sistemes d'ASR. Hem vist que TLK té tres comandaments d'alt nivell que permeten fer el treball amb l'ajuda de fitxers de configuració. El preprocés es fa mitjançant `tLtask-preprocess`, l'entrenament mitjanánt `tLtask-train` i el reconeixement mitjançant `tLtask-recognise`.

El preprocés realitza les tasques d'extracció de característiques i transliteració a monofonemes de la transcripció. L'entrenament construeix i determina els paràmetres dels models acústics, que consisteixen en HMMs. Aquest entrenament segueix unes etapes en l'entrenament del sistema estàndard, i posteriorment del sistema adaptat; que tenen un ordre de construcció. Dins de les etapes d'entrenament dels sistemes estàndard i adaptat amb gaussianes, s'entrenen tres models: el de monofonemes, el de trifonemes i el de fonemes lligats; i a continuació s'han substituït les gaussianes per xarxes neurals per obtindre sistemes híbrids.

Hem vist que l'ús del sistema entrenat té lloc en el reconeixement. Amb els models acústics entrenats, el model de llenguatge i el diccionari de pronúncia, es construeix una xarxa de reconeixement en la qual fer la cerca de la hipòtesi amb major versemblança. El reconeixement consta de dos passos: el primer amb el model estàndard per obtindre una primera aproximació que es gasta per fer la transformació a CMLLR de les característiques i passar al segon pas, amb el model adaptat. Es reconeix primer el conjunt `dev` per a definir quins són els millors paràmetres de reconeixement GSF i WIP, i posteriorment el conjunt `test` per a provar el sistema i calcular-ne l'error.

En l'experiment realitzat, la combinació òptima de paràmetres de reconeixement era GSF=11 i WIP=18; i el WER obtingut amb eixos paràmetres sobre el conjunt `test` ha sigut del 22.2%.

CAPÍTOL 5

Sistema KALDI de reconeixement de la parla en català

5.1 Introducció

En aquest capitol es descriuen els procediments que s'han seguit en el treball per a l'avaluació empírica de Kaldi en la tasca proposada de sistema de reconeixement de català. Com que els conceptes de preprocés, entrenament i reconeixement són els mateixos que en TLK, i s'han executat els mateixos en el mateix ordre, en aquest capítol ens limitarem a descriure les diferències entre els dos.

El capítol conté una primera secció de preparació i preprocés de les dades, que explica com preparar les dades a falta d'un comandament equivalent a `tLtask-preprocess`. Li segueix una altra secció d'entrenament del sistema, exposant algunes diferències com la topologia dels HMM del silenci o el tipus d'entrenament amb l'aproximació de Viterbi. A continuació s'hi troba una secció de reconeixement, on s'introdueixen conceptes com el *rescoring* i s'explica com Kaldi fa ús de la biblioteca `openfst` per a construir i cercar en la xarxa de reconeixement. Li segueix una secció de descripció de l'experiment realitzat; i per acabar una secció de conclusions on es recullen totes les idees més importants del capítol.

5.2 Preparació i preprocés de les dades

Kaldi no inclou l'equivalent al tLtask-preprocess de TLK, i per tant cal gastar les eines de més baix nivell que sí que inclou a més de fer manualent alguns scripts per a preparar alguna part de les dades. El procés de preparació de les dades està publicat en la documentació, i es pot trobar en `http://kaldi-asr.org/doc2/data_prep.html`.

Els fitxers necessaris en cada carpeta de dades són els següents: `cmvn.scp`, `feats.scp`, `reco2file_and_channel`, `segments`, `spk2utt`, `text`, `utt2spk` i `wav.scp`. Alguns d'aquests poden ser creats automàticament per Kaldi a partir d'uns altres. Els arxius que cal crear "a mà" són els següents:

- `text`: conté les transcripcions de cada seqüència
- `wav.scp`: conté una llista amb identificadors dels àudios i la ruta on estan
- `segments`: és una llista amb identificadors de segments, identificadors d'àudios i instant d'inici i de final de cada segment
- `reco2file_and_channel`: és un fitxer que serveix per a mesurar l'error en `dev` i `test`.
- `utt2spk`: aquest fitxer indica qui és el locutor de cada segment.

El fitxer `spk2utt` és creat per l'*script* `utt2spk_to_spk2utt.pl` a partir de `utt2spk`. L'extracció de característiques MFCC la fa un altre *script* de Kaldi a partir de les llistes anteriors: `make_mfcc.sh`, que també genera una llista que en fa referència: `feats.scp`. Per últim, la normalització de les mostres per mitjana i variància es fa per mitjà de `compute_cmvn_stats.sh`. Per comprovar que la carpeta està en un format correcte, es pot fer servir `validate_data_dir.sh` que és un altre *script* de Kaldi.

A més de la preparació de les dades (`train`, `dev` i `test`), també cal disposar del model de llenguatge en forma de transductors d'estats finits. Els fitxers que han de formar part del directori `data/lang/`, que són molts, es poden generar amb un *script* de Kaldi que a partir d'un conjunt reduït d'informació genera la carpeta dels models de llenguatge.

Aquest conjunt reduït està format per 5 arxius: `extra_questions.txt`, `lexicon.txt`, `nonsilence_phones.txt`, `optional_silence.txt` i `silence_phones.txt`. Tots aquests fitxers són llistes: `lexicon.txt` és el diccionari de pronúncia, mentre que els altres són llistes de possibles fonemes i altres llistes secundàries.

Això crea tota la carpeta `data/lang` excepte el model de llenguatge en forma de transductors de què parlàvem adés, l'equivalent al d'n-grames. Aquest model

de llenguatge s'ha obtingut convertint un fitxer de format ARPA a transductors d'estats finits, utilitzant utilitats de Kaldi i d'openfst (que ve amb la distribució de Kaldi).

Com veurem en la secció 5.4, Kaldi gasta biblioteques externes d'estats finits per a treballar amb aquests transductors. Aquestes biblioteques no estan preparades per a treballar amb un model massa gran com és la conversió directa del model de llenguatge ARPA, i cal podar el model de llenguatge eliminant els n-grames poc freqüents (per sota d'un llindar determinat). En la secció 5.4 veurem l'estratègia que se segueix per a no perdre informació del model de llenguatge en l'estimació de la millor hipòtesi de transcripció.

Kaldi inclou un *script* que serveix per comprovar que `data/lang` està correctament creada: `validate_data_dir.sh`; en el nostre cas hi havia un error amb 7 segments que s'ha solucionat fent ús d'un altre *script* de Kaldi, `fix_data_dir`, i aquest *script* els ha detectat i eliminat de les mostres.

5.3 Entrenament del sistema d'ASR

Amb Kaldi, també se segueixen els mateixos passos per a entrenar el reconeixedor que aplicàvem amb TLK descrits en el capítol 4: entrenament del sistema estàndard, transformació de les característiques a CMLLR i entrenament del sistema adaptat al locutor. Els passos que s'han seguit han estat els mateixos per tal que els sistemes obtinguts foren comparables. No obstant això, hi ha algunes diferències que cal remarcar i que es comenten a continuació.

La topologia dels HMMs que utilitza Kaldi és configurable, i s'ha utilitzat la que té per defecte: tres estats per als fonemes/trifonemes amb so i quatre estats per als silencis. La topologia dels HMMs es defineix en `data/lang/topo`. La topologia dels fonemes amb so és la mateixa que s'ha descrit per a TLK, i la dels silencis està representada en la figura 5.1.

L'entrenament dels paràmetres dels models en Kaldi segueix un algorisme diferent al descrit en la secció 4.3 per a TLK. També es construeix un HMM gran concatenant tots els fonemes de la frase, però en compte de calcular l'ocupació de cada estat per a cada vector de característiques s'aplica l'aproximació de Viterbi, que consisteix a calcular l'estat més probable que haja emés eixa característica i assignar-li el 100% de probabilitat. Aquest procés rep el nom d'"alineament" perquè alinea cada vector de característiques amb un estat. Les mitjanes i variàncies de les gaussianes es calculen amb els vectors que ha acumulat cada estat al llarg de totes les dades d'entrenament.

Kaldi no disposa d'un programa d'alt nivell de l'estil de TLK que permeta fer l'entrenament simplement amb una crida i un fitxer de configuració; cal crear un

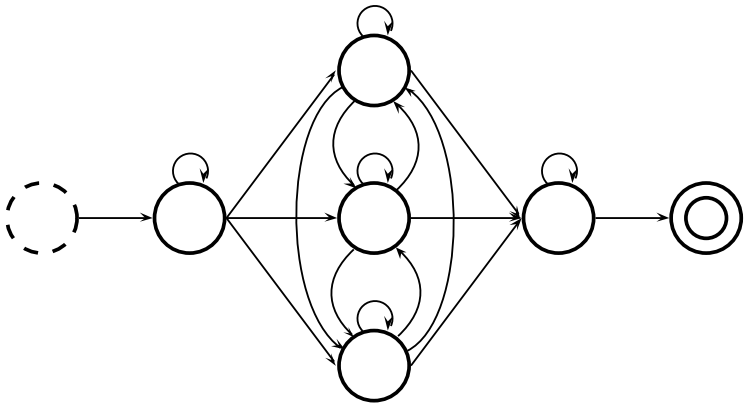

Figura 5.1: Topologia dels HMMs del silenci en Kaldi. Les emissions no hi estan representades per simplificar l'esquema.

script propi que faça ús de les utilitats de Kaldi per entrenar el sistema. S'entrena primer el model de monofonemes, després el de trifonemes i fonemes lligats (és la mateixa crida en Kaldi), a continuació els CMLLR i per acabar les xarxes neurals. Entre cada model i el següent cal fer una aliniació per Viterbi. Tot seguit hi ha una descripció d'aquest procés:

1. Entrenament del model de monofonemes (`train_mono.sh`)
2. Aliniament de les dades amb el model de monofonemes (`align_si.sh`)
3. Entrenament del model de trifonemes i fonemes lligats (`train_deltas.sh`)
4. Aliniament de les dades amb el model de fonemes lligats (`align_si.sh`)
5. Entrenament del model adaptat (`train_sat.sh`)
6. Aliniament de les dades, ara amb el model adaptat (`align_fmllr.sh`)

Una diferència notable entre TLK i Kaldi és la forma en què decideixen quins estats dels trifonemes han de ser lligats. TLK utilitza una sèrie de preguntes binàries sobre els trifonemes que estan predeterminades, mentre que Kaldi les crea basant-se en la distància dels vectors acústics.

Cadascun dels scripts gastats té una entrada i una eixida pròpies que es poden indicar en la crida. Les carpetes gastades han estat les següents:

- `exp/mono` per als models de monofonemes
- `exp/mono_ali` per a l'aliniament amb monofonemes

- `exp/tri1` per als models de fonemes lligats
- `exp/tri1_ali` per a l'aliniament amb fonemes lligats
- `exp/tri2` per als models CMLLR de fonemes lligats
- `exp/tri2_ali` per a l'aliniament amb CMLLR

L'entrenament dels models híbrids es fa mitjançant un *script* de Kaldi, `run_dnn.sh`, que fa tant l'entrenament com l'exploració GSF i WIP amb els models entrenats per a `dev` i `test`.

5.4 Reconeixement

El procés de cerca en Kaldi es basa en els mateixos conceptes que TLK. A partir dels HMMs, el diccionari de pronúncia i el model de llenguatge es construeix una xarxa de reconeixement. Hi ha una diferència clara respecte a la construcció d'aquesta xarxa, perquè tant el diccionari de pronúncia com el model de llenguatge estan en forma de transductors d'estats finits.

En treballar amb models d'aquest tipus, la construcció de la xarxa de reconeixement es fa mitjançant operacions estàndard de composició, determinització i miniminització de transductors d'estats finits [24], i per a realitzar aquestes operacions Kaldi fa ús de la biblioteca especialitzada `openfst` [25].

Aquesta biblioteca també dóna suport a la cerca optimitzada en transductors d'estats finits amb pesos. Es pot fer servir aquesta utilitat per a implementar la cerca, siguent els pesos les probabilitats dels models. Com ja s'ha dit en la secció 5.2, el model que es gasta en aquesta cerca és un model podat per tal que `openfst` puga treballar amb ell.

Per a calcular correctament les probabilitats de cada hipòtesi, en la primera cerca es crea un graf que representa hipòtesis de frases (*wordgraph*). Sobre aquest graf es fa el que es diu *rescoring*, que és una re-estimació de les probabilitats de transició entre les paraules del graf utilitzant el model sense podar. Seguidament es torna a fer una cerca sobre aquest graf i es pren la hipòtesi més probable.

Kaldi gasta el *rescoring* intensivament. Els passos de reconeixement són els següents:

1. Genera el *wordgraph* amb el model de llenguatge podat i el model estàndard de gaussianes com a model acústic.
2. Primer pas de reconeixement per a convertir les característiques a CMLLR.
3. Fa un *rescoring* acústic sobre el *wordgraph* amb el model híbrid adaptat.

4. Fa un segon *rescoring* amb el model de llenguatge sense podar.

Sobre els *wordgraphs* es fa l'exploració de GSF i WIP descrita en la secció 4.6. Els paràmetres GSF i WIP afecten als pesos en la re-estimació de les probabilitats, de forma que afecta el resultat de la hipòtesi més probable sense que siga necessari tornar a generar el graf cada vegada. Cal tindre en compte que encara que els conceptes darrere dels paràmetres GSF i WIP de Kaldi són els mateixos que els que hi ha darrere dels paràmetres de TLK, la implementació és diferent i no és estrany trobar-se amb un resultat òptim subtancialment diferent.

Com ja s'ha dit al final de la secció 5.3, Kaldi conté un *script* que fa l'entrenament i la classificació amb les xarxes neurals profundes en un context d'avaluació empírica de construcció i prova de sistemes d'ASR. Aquest *script*, `run_dnn.sh`, també fa l'exploració de GSF i WIP però no fa el *rescoring*, que és l'últim pas de l'avaluació empírica realitzada de Kaldi.

Aquest últim pas es fa mitjançant `lmrescore.sh` sobre cada resultat en forma de *wordgraph* de l'exploració de GSF i WIP de `dev` i com a mínim del millor resultat de GSF i WIP en `test`; en aquest cas s'ha fet *rescoring* sobre tots els resultats de l'exploració en els dos conjunts de dades.

5.5 Experiments

Amb Kaldi s'han realitzat proves experimentals per avaluar l'eina amb condicions similars a TLK. S'ha entrenat el sistema a partir de les mateixes dades d'entrenament. Els sistemes finals han consistit de dos models: l'estàndard i l'adaptat. Hi ha una diferència rellevant respecte als mateixos models entrenats amb TLK: només el model adaptat és híbrid, el primer pas de reconeixement es fa amb models de gaussianes.

En el reconeixement, Kaldi genera automàticament l'exploració de GSF i WIP tant per al conjunt `dev` com per al `test`. Els resultats estan en la taula 5.1. A partir d'aquesta exploració, hem de prendre els valors de GSF i WIP que han funcionat millor amb el conjunt `dev`, és a dir 12 i 0.0 respectivament. En `test`, el WER amb aquests paràmetres és del 21.5%.

Aquests errors són calculats per Kaldi i per tant no són directament comparables als calculats per TLK, ja que hi ha detalls que canvien a l'hora de computar-los. L'últim pas per tal que els errors siguen comparables ha sigut calcular l'error amb TLK a partir de la transcripció feta per Kaldi per al conjunt `test` amb GSF=12 i WIP=0.0. El WER del sistema de Kaldi queda, així, amb un valor del 21.1%.

Taula 5.1: Exploració de GSF i WIP en `dev` i `test`, amb WERs (%) calculats per Kaldi.

	WIP					
	`dev`			`test`		
GSF	0.0	0.5	1.0	0.0	0.5	1.0
10	24.1	24.8	25.8	21.8	22.2	22.9
11	24.1	24.8	25.8	21.4	22.1	23.0
12	**24.0**	24.9	26.0	**21.5**	22.3	23.2
13	24.1	25.1	26.4	21.7	22.5	23.6
14	24.4	25.5	26.8	21.9	22.8	23.8
15	24.7	25.9	27.1	22.2	23.0	24.2
16	25.0	26.1	27.5	22.5	23.5	24.5
17	25.3	26.4	27.9	22.9	23.8	24.9
18	25.5	26.8	28.2	23.1	24.2	25.3
19	25.9	27.2	28.5	23.4	24.5	25.7
20	25.2	27.6	28.8	23.7	24.8	26.1

5.6 Conclusions

Aquest capítol ha servit per exposar els passos de construcció i utilització de sistemes d'ASR amb Kaldi. Hem vist que Kaldi, al contrari que TLK, no conté grans comandaments d'alt nivell que simplifiquen els passos sinó que cal fer ús d'utilitats més concretes escrivint *scripts* que les facen servir.

S'ha mostrat la preparació i preprocés de les dades, on destaca la poda i conversió del model de llenguatge com a característics de Kaldi. S'ha explicat el procediment d'entrenament, que seguia les mateixes etapes que TLK amb algunes diferències, com la topologia dels HMMs dels silencis o l'entrenament amb l'aproximació de Viterbi. S'ha exposat el procediment de reconeixement de Kaldi, que es basa en *rescoring* sobre el *wordgraph* de la cerca del primer pas de reconeixement. Per últim, s'han mostrat els resultats obtinguts de l'avaluació empírica.

CAPÍTOL 6

Comparació dels sistemes TLK i KALDI

6.1 Introducció

En aquest capítol es fa una comparació de les eines TLK i Kaldi després d'haver construït i provat sistemes ASR en català amb les dues. Hi ha tres vessants d'aquesta comparativa, i cadascuna compta amb la seua secció. La primera secció compara la facilitat d'ús de les eines, la segona fa un balanç dels resultats obtinguts i la tercera compara l'eficiència temporal dels sistemes en el reconeixement. Finalment, hi ha una secció de recapitulació de la comparació on es remarquen les conclusions més rellevants.

6.2 Qualitat dels resultats

La qualitat dels resultats es mesura amb la mètrica del WER introduïda en la secció 1.4 sobre el conjunt `test` definit, comparant la transcripció realitzada pel sistema amb la transcripció supervisada que conté el corpus. Com ja hem vist anteriorment, l'error del sistema construit amb TLK ha estat del 22.2% i l'error del construit amb Kaldi del 21.1%.

La diferència en WER en termes absoluts és de 1.1 punts percentuals en favor de Kaldi. En termes relatius, el WER del sistema de Kaldi és un 5% millor que el de TLK en condicions semblants. Aquesta diferència no és desproporcionada però sí que es considera significativa.

A més de mesurar el WER, a l'hora de mesurar l'error de la transcripció també es calcula el detall del nombre i percentatge de paraules de la referència correctament transcrites, així com nombre i percentatge de substitucions, eliminacions i insercions en la transcripció automàtica respecte de la supervisada. Els resultats estan representats en la taula 6.1.

Taula 6.1: Comparació dels resultats dels dos sistemes

Sistema	Encerts	Substitucions	Eliminacions	Insercions	WER
Kaldi	80.2% (15316)	13.1% (2495)	6.7% (1284)	1.3% (249)	21.1%
TLK	80.4% (15358)	14.3% (2735)	5.2% (1002)	2.6% (502)	22.2%

Podem observar que els resultats són prou semblants en tots els tipus d'error. El sistema de TLK tendeix a fer més substitucions (1.2 punts de diferència) i insercions (1.3 punts), mentre que el de Kaldi tendeix a fer més eliminacions (1.5 punts). Per a posar aquesta informació en context, és interessant saber la quantitat de paraules que conté cadascuna de les transcripcions del conjunt, que es pot veure en la taula 6.2.

Taula 6.2: Quantitat de paraules en la referència i en les transcripcions

Transcripció	Paraules
Referència	19095
Sistema TLK	18595
Sistema Kaldi	18060

Podem veure que el sistema TLK té més tendència a introduir paraules: 535 paraules més, que representa un 2.8% respecte de la quantitat de paraules de la referència; una quantitat no menyspreable quan la diferència entre els errors ha estat de l'1.1%. Una altra forma de veure l'impacte d'aquesta circumstància és comparar aquesta xifra amb les 191 paraules que representarien 1 punt de WER. El fet que la transcripció del sistema de TLK conté més paraules fa que siga esperable que continga més errors d'insercions i menys d'eliminacions que el sistema de Kaldi, com de fet ocorre.

Per explicar la diferència d'error entre TLK i Kaldi, s'han fet experiments addicionals. El primer d'aquests experiments ha estat el reconeixement amb sistemes

preliminars del conjunt `test` amb l'objectiu d'esbrinar en quina etapa funciona millor cada eina. La taula 6.3 mostra el WER de cada model dels entrenats en el treball. Els sistemes de gaussianes són els de tied-fonemes amb mixtures de fins a 64 components.

Taula 6.3: Comparació de l'error del reconeixement en cada etapa

	Gaussianes		Híbrid	
Eina	Estàndard	Adaptat	Estàndard	Adaptat
Kaldi	37.3%	28.8%	—	21.1%
TLK	43.6%	34.4%	25.3%	22.2%

Podem veure que Kaldi té uns resultats significativament més bons en el reconeixement amb els models de gaussianes, però no hi ha tanta diferència en els sistemes finals.

Les altres proves han estat l'entrenament i reconeixement de sistemes TLK amb components que s'han identificat en Kaldi, concretament les característiques extretes i la topologia del model de silenci, per veure la rellevància relativa d'aquestes dues diferències. Els resultats d'aquestes avaluacions experimentals addicionals es troben en la taula 6.4.

Taula 6.4: WER de TLK amb el sistema sense modificar, amb característiques extretes per Kaldi, i amb la topologia del silenci de Kaldi.

Sistema base TLK	Característiques Kaldi	Silenci Kaldi
22.2%	22.9%	22.7%

Les característiques MFCC extretes per Kaldi han funcionat lleugerament pitjor que les de TLK. Com que el preprocés i l'entrenament són processos ben diferenciats, aquest resultat pot indicar que les característiques extretes per TLK són més discriminatives que les extretes per Kaldi. Això no contribuiria a explicar per què el sistema de Kaldi funciona més bé, més aviat al contrari.

Els resultats del sistema de TLK amb la topologia dels HMMs del silenci pròpia de Kaldi han estat lleugerament pitjors respecte del sistema base. Això no necessàriament indica que la topologia de Kaldi funciona pitjor: és probable que hi haja altres elements de com es fa l'entrenament o la cerca en cada *toolkit* que fan que la topologia no siga directament transferible a TLK sense penalitzar altres procediments. Per tant, no es pot afirmar a partir dels resultats que la topologia del silenci de Kaldi siga una modelització més encertada que la de TLK però tampoc que siga una pitjor modelització. Així i tot, es pot assumir que la topologia del

silenci no és una de les causes principals dels resultats millors de Kaldi respecte a TLK.

En síntesi, es pot dir que els resultats del sistema Kaldi són comparativament més bons que els de TLK per un marge relativament escàs però significatiu donat que són els mateixos passos els seguits amb les dues eines, que TLK té més tendència a inserir paraules, i que els sistemes de gaussianes de Kaldi funcionen considerablement més bé que els de TLK encara que la distància es redueix de forma notable amb els sistemes híbrids. Les característiques de TLK semblen ser lleugerament més discriminatives, i no hem pogut concloure que la topologia del silenci de Kaldi siga part de l'explicació del seus millors resultats que TLK.

6.3 Eficiència temporal

Com ja s'ha dit en la secció 1.4, la mètrica que s'utilitza per a l'avaluació de l'eficiència temporal d'un sistema d'ASR és el RTF, que es definia com a:

$$RTF = \frac{t_p}{t_v} \tag{6.1}$$

Per calcular l'RTF cal conéixer la durada dels vídeos i el temps utilitzat per cada sistema en el reconeixement. Per a mesurar aquests temps s'han llançat processos de reconeixement específicament seqüencialitzats i executats en la mateixa màquina. Això és important perquè si es llançen en el *cluster* com s'han llançat la resta de processos les condicions en què es mesuraren els temps no serien necessàriament comparables. Les mesures estan indicades en la taula 6.5. El temps del sistema Kaldi està separat en el temps *inicial*, que fa referència a la primera cerca i generació dels *wordgraphs*, i el temps de *rescoring* del model de llenguatge.

Taula 6.5: durada dels vídeos del `test` i temps de cada sistema en el reconeixement

Sistema	Temps
Referència	1h 50' 28"
TLK	16h 43' 50"
Kaldi (inicial)	1h 28' 39"
Kaldi (rescoring)	2h 30' 51"
Kaldi (total)	3h 59' 30"

Els RTFs resultants són de 9.1 per al sistema de TLK i de 2.2 per al sistema de Kaldi. Aquesta és una gran distància que pot tindre una explicació en les diferències en el procediment de reconeixement: TLK fa una cerca completa amb xarxes neurals profundes en els dos passos de reconeixement, mentre que Kaldi

només fa una cerca completa amb gaussianes per al primer pas per a generar el *wordgraph* i el segon pas és un *rescoring* acústic amb les xarxes neurals profundes. El fet que per al primer pas s'utilitze un model de llenguatge podat també pot explicar aquesta diferència en l'eficiència temporal.

6.4 Facilitat d'ús

Els sistemes d'ASR són complexos i per tant les eines que serveixen per a construir-los i utilitzar-los han de ser necessàriament complexes també. Per això és important la facilitat del seu ús, per a no afegir complicacions innecessàries a l'hora de fer-les servir i que no siga necessari un coneixement de molts detalls de la seua implementació o estructura.

TLK disposa de tres eines d'alt nivell: tLtask-preprocess, tLtask-train i tLtask-recognise; que a partir d'una entrada simple (àudios i transcripcions en `trs` o `dfxp`) poden fàcilment entrenar un sistema ASR, gastar-lo per a reconéixer i mesurar-ne l'error de classificació.

Per a aprendre a utilitzar TLK, en la seua pàgina web (`http://www.mllp.upv.es/tlk/`) està la seua documentació i tutorials. Per a configurar l'entrenament dels models i la classificació es defineixen fitxers de configuració que també estan detallats en la documentació amb el significat de cada opció.

Kaldi no té utilitats d'alt nivell que permeten un ús ràpid de l'eina, cal aprendre un cert detall del seu funcionament per a poder construir-hi sistemes i escriure scripts que criden als scripts i executables de Kaldi en l'ordre i amb els paràmetres correctes. Per sort, compta amb una extensa documentació que es pot consultar a `http://kaldi-asr.org/doc2/` i una gran quantitat d'exemples de construcció i ús de sistemes a partir de diferents corpus que vénen amb la pròpia distribució oficial de Kaldi.

La part més tediosa és la preparació de les dades a partir del format en què es troben en els corpus. Fer els scripts necessaris per a obtindre les nombroses llistes que calen i que estiguen en l'ordre correcte no és una tasca trivial, i encara que hi ha exemples és molt fàcil fallar amb algun detall. Per sort la documentació i els exemples són molt bons i tot es pot fer sense grans impediments fins i tot per a algú amb coneixements limitats d'ASR una volta familiaritzat amb l'entorn.

Si es compara la facilitat d'ús per a construir sistemes ASR, TLK és significativament més fàcil d'utilitzar que Kaldi. Així i tot, per a l'objectiu de crear dos sistemes comparables, fent els mateixos passos amb les dues eines ha fet falta un cert grau de comprensió del funcionament de les dues.

En resum, si l'objectiu és construir i utilitzar un sistema d'ASR, TLK és l'eina més fàcil d'utilitzar; però Kaldi compta amb una extensa documentació i exemples que faciliten l'aprenentatge del seu funcionament amb un major grau de detall, ja que l'objectiu de Kaldi no és la facilitat en la construcció de sistemes sinó la seua utilització per a investigació i potser la versatilitat afegida fa que siga una eina menys destinada a la producció i complica la seua utilització.

6.5 Conclusions

Aquest capítol ha servit per a comparar les eines Kaldi i TLK des del punt de vista de la facilitat d'ús, de la qualitat dels resultats obtinguts i de l'eficiència temporal dels sistemes construïts en el reconeixement.

Quant a la qualitat dels resultats, el sistema de TLK té uns resultats lleugerament més roïns que els de Kaldi però la diferència (1.1 punts de WER) és significativa donat que els passos seguits en el seu entrenament han estat els mateixos. Hem vist que TLK té més tendència a la inserció de paraules que Kaldi, i que en gaussianes els sistemes de Kaldi donen uns resultats notablement més bons que els de TLK, diferència que es veu atenuada amb els sistemes híbrids.

De la facilitat d'ús, hem dit que TLK és més fàcil d'utilitzar ja que el preprocés, l'entrenament i el reconeixement tenen tres grans comandaments configurables que faciliten el procés. En canvi, per realitzar aquestes tasques amb Kaldi, l'ús és més complicat ja que cal fer servir utilitzats més especialitzades basant-se amb els scripts d'exemple que inclou i consultant la documentació. Aquesta dificultat afegida es pot deure a que Kaldi està dissenyat pensant en el seu ús en investigació i s'haja primat la flexibilitat per sobre de la usabilitat.

CAPÍTOL 7

Conclusions

L'ASR en català té interés en el marc de la Universitat Politècnica de València perquè permet transcriure i traduir vídeos educatius amb una qualitat acceptable, ràpidament i a baix cost. Al llarg d'aquest treball, s'han construit i avaluat sistemes de reconeixement de la parla en català amb TLK i amb Kaldi per a fer-ne una valoració comparativa.

En aquesta memòria han estat introduïts els conceptes fonamentals de l'ASR, amb una visió general del reconeixement de patrons i amb les mètriques utilitzades per a l'avaluació dels sistemes d'ASR. S'han presentat les eines utilitzades, tant les comparades en el treball (TLK i Kaldi) com l'eina específica per a construir models de llenguatge (SRILM). També s'han vist els tres corpus de dades utilitzats per a entrenar i avaluar els sistemes d'ASR: TECNOPARLA, Glissando i poliMedia-català; i el preprocés que se'ls ha aplicat per extraure'n característiques i obtindre'n transcripcions en monofonemes.

Hem vist les fases de l'entrenament dels sistemes, amb cadascun dels passos iteratius i models intermedis generats, que s'aplica tant per a la construcció del model estàndard, que utilitza les característiques MFCC, com per a la del model adaptat, que utilitza les característiques CMLLR. Les principals diferències que s'han identificat entre TLK i Kaldi en l'entrenament han estat l'algorisme d'aprenentatge

(Baum-Welch i Viterbi, respectivament) i la topologia del model de silenci, però poden haver-hi més factors distintius.

S'ha explicat el concepte de la xarxa de reconeixement en la qual té lloc la cerca de la hipòtesi de major versemblança. El reconeixement emprat en el treball ha constat de dos passos: un primer amb el model estàndard, i un segon amb el model adaptat. S'ha fet una exploració en GSF i WIP tant en `dev` com en `test` amb els sistemes entrenats en les dues eines. S'han vist les diferències en l'enfocament del reconeixement: TLK fa dos reconeixements complets, mentre que Kaldi realitza un reconeixement que genera un *wordgraph* i fa servir *rescorings* per obtindre la transcripció final.

Quant a la qualitat dels resultats obtinguts, el sistema entrenat amb Kaldi ha obtingut un WER de 1.1 punts menys d'error, el que representa una millora relativa del 5%, que no és molta diferència però sí que és significativa. Una bateria de proves ha demostrat que els sistemes preliminars amb gaussianes funcionen considerablement millor amb Kaldi, amb diferències de fins a 6.3 punts de WER. També s'ha conclòs a partir d'aquestes proves addicionals que les característiques extretes per TLK semblen ser lleugerament més discriminatives. Ni l'extracció de característiques ni la topologia del model de silenci semblen ser motius dels millors resultats de Kaldi respecte a TLK.

A més dels resultats obtinguts, s'han avaluat les eines pel que fa a eficiència temporal i facilitat d'ús. Pel que fa a l'eficiència temporal, el sistema de TLK ha aconseguit un RTF de 9.1 mentres que el de Kaldi n'ha obtingut un de 2.2. Aquesta diferència és significativament gran, i es pot explicar degut a l'ús que fa Kaldi del *rescoring*. Quant a facilitat d'ús, TLK ha resultat relativament més senzill d'utilitzar degut a que compta amb els tres comandaments d'alt nivell per al preprocés, entrenament i reconeixement.

El treball obri la porta a treballs futurs en base al que s'ha exposat en aquesta memòria. Una via a explorar és l'aprofundiment de la identificació de les causes de les diferències de resultats entre TLK i Kaldi. Els resultats de la comparació amb sistemes de gaussianes indiquen que un focus d'interés es troba en els algorismes que entrenen les mixtures de gaussianes. Altres processos que poden ser rellevants per al seu estudi són: el CART (el procediment d'obtenció de fonemes lligats), la transformació CMLLR o l'entrenament de xarxes neurals profundes.

Una altra via de treball futur és la millora de TLK mitjançant tècniques que en Kaldi hagen demostrat millorar els resultats. Una possibilitat de millora que es desprén de les proves realitzades en aquest treball és l'ús del *rescoring* per guanyar en eficiència temporal.

Bibliografia

[1] José A. R. Fonollosa. "La tecnologia de la parla en català. Avenços i reptes". A: *Llengua i ús. Revista Tècnica de Política Lingüística* 48 (2010) (v. la pàg. 2).

[2] S Young et al. "The HTK book (v3. 4)". A: *Cambridge University* (2006) (v. la pàg. 2).

[3] David Rybach et al. "The RWTH aachen university open source speech recognition system." A: *Interspeech*. 2009, pàg. 2111 - 2114 (v. la pàg. 2).

[4] Willie Walker et al. "Sphinx-4: A flexible open source framework for speech recognition". A: (2004) (v. la pàg. 2).

[5] MA del-Agua et al. "The translectures-UPV toolkit". A: *Advances in Speech and Language Technologies for Iberian Languages*. Springer, 2014, pàg. 269 - 278 (v. les pàg. 2, 12).

[6] Daniel Povey et al. "The Kaldi speech recognition toolkit". A: *IEEE 2011 workshop on automatic speech recognition and understanding*. EPFL-CONF-192584. IEEE Signal Processing Society. 2011 (v. les pàg. 2, 12).

[7] A. R. Webb i Keith D. Copsey. *Statistical pattern recognition*. Wiley, 2011 (v. la pàg. 3).

[8] Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1997 (v. la pàg. 5).

[9] Zoubin Ghahramani. "An Introduction to Hidden Markov Models and Bayesian Networks". A: *International Journal of Pattern Recognition and Artificial Intelligence* (2001), pàg. 9 - 42 (v. la pàg. 7).

[10] Joan A. Silvestre Cerdà et al. "transLectures". A: IberSPEECH 2012. 2012, pàg. 345 - 351 (v. la pàg. 11).

[11] Miguel Angel del-Agua et al. "The MLLP ASR Systems for IWSLT 2015". A: (2015) (v. la pàg. 12).

[12] A. Martínez-Villaronga et al. "Advances in Speech and Language Technologies for Iberian Languages: Second International Conference, IberSPEECH 2014, Las Palmas de Gran Canaria, Spain, November 19-21, 2014. Proceedings". A: ed. de Juan Luis Navarro Mesa et al. Cham: Springer International Publishing, 2014. Cap. Language Model Adaptation for Lecture Transcription by Document Retrieval, pàg. 129 - 137. ISBN: 978-3-319-13623-3. DOI: 10.1007/978-3-319-13623-3_14 (v. la pàg. 12).

[13] Ondrej Plátek i Filip Jurcıcek. "Free on-line speech recogniser based on Kaldi ASR toolkit producing word posterior lattices". A: *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. 2014, pàg. 108 - 112 (v. la pàg. 13).

[14] Karel Veselỳ et al. "Sequence-discriminative training of deep neural networks." A: *INTERSPEECH*. 2013, pàg. 2345 - 2349 (v. la pàg. 13).

[15] Xavier Anguera, Jordi Luque i Ciro Gracia. "Audio-to-text alignment for speech recognition with very limited resources." A: *INTERSPEECH*. 2014, pàg. 1405 - 1409 (v. la pàg. 13).

[16] B Popović et al. "Large vocabulary continuous speech recognition for Serbian using the Kaldi toolkit". A: *Proceedings of the 10th DOGS Conference*. 2014, pàg. 31 - 34 (v. la pàg. 13).

[17] Andreas Stolcke et al. "SRILM-an extensible language modeling toolkit." A: *INTERSPEECH*. Vol. 2002. 2002, pàg. 2002 (v. la pàg. 14).

[18] Henrik Schulz, M. Costa-Jussà i José A. R. Fonollosa. "TECNOPARLA - Speech technologies for Catalan and its application to Speech-to-speech

Translation". A: *Procesamiento del lenguaje Natural* 41 (2008), pàg. 319-320. ISSN: 1135-5948 (v. la pàg. 15).

[19] Mateu Aguiló et al. "A Hierarchical Architecture for Audio Segmentation in a Broadcast News Task". A: *Proceedings Workshop on Speech and Language Technologies for Iberian Languages ((Porto Salvo, Portugal, 2009), pp)* (2009), pàg. 17-20 (v. la pàg. 16).

[20] Henrik Schulz, José A. R. Fonollosa i David Rybach. "Transcription of Catalan Broadcast Conversation". A: *TSD*. 2009, pàg. 154-161 (v. la pàg. 16).

[21] Henrik Schulz i José A. R. Fonollosa. "A Catalan Broadcast Conversational Speech Database". A: 2009, pàg. 27-30 (v. la pàg. 16).

[22] J. Garrido et al. "Glissando: a corpus for multidisciplinary prosodic studies in Spanish and Catalan". A: *Language resources and evaluation* 47.4 (des. de 2013), pàg. 945-971. DOI: 10.1007/s10579-012-9213-0 (v. la pàg. 16).

[23] Juan Daniel Valor Miró et al. "Integrating a State-of-the-Art ASR System into the Opencast Matterhorn Platform". A: 2012, pàg. 237-246 (v. la pàg. 17).

[24] Mehryar Mohri, Fernando Pereira i Michael Riley. "Speech recognition with weighted finite-state transducers". A: *Springer Handbook of Speech Processing*. Springer, 2008, pàg. 559-584 (v. la pàg. 43).

[25] Cyril Allauzen et al. "OpenFst: A general and efficient weighted finite-state transducer library". A: *Implementation and Application of Automata*. Springer, 2007, pàg. 11-23 (v. la pàg. 43).

Índex de figures

Índex de taules