Influence de la quantité de données sur une tâche de segmentation de phones fondée sur les réseaux de neurones

Céline MANENTI Thomas PELLEGRINI Julien PINQUIER
IRIT, Université de Toulouse, UPS, Toulouse, France
{celine.manenti, thomas.pellegrini, julien.pinquier}@irit.fr

RÉSUMÉ

Dans cet article, nous décrivons une étude expérimentale de segmentation de parole en unités acoustiques sous-lexicales (phones) à l'aide de réseaux de neurones. Sur le corpus de parole spontanée d'anglais américain BUCKEYE, une F-mesure de 68% a été obtenue à l'aide d'un réseau convolutif, en considérant une marge d'erreur de 10 ms. Cette performance est supérieure à celle d'un annotateur manuel, l'accord inter-annotateurs étant de 62%. Restreindre les données d'apprentissage à celles d'un unique locuteur, 30 minutes environ, a eu pour conséquence moins de 10% de perte et utiliser celles de 5 locuteurs a permis d'atteindre des résultats similaires à utiliser plus de données. Utiliser le modèle entraîné avec le corpus anglais sur un petit corpus d'une langue peu dotée a donné des résultats comparables à estimer un modèle avec des données de cette langue.

ABSTRACT

Phone-level speech segmentation with neural networks : influence of the amount of data

In this article, we describe speech segmentation experiments at phone level with neural networks, on a U.S. English speech corpus. Using filter banks and a ConvNet, a 68% F-measure was obtained, with an error margin of 10 ms, a figure close to the annotation agreement rate between human annotators. We then studied the impact of reducing the training data size : the decrease in performance was less than 10% only, when training with data from a single speaker, 30 min of speech, instead of data from 5 speakers. More data did not bring further improvements. Finally, we used a CNN trained on U.S. English to segment a small corpus in Xitsonga, a less-resourced language from South Africa. The English model led to a similar performance when compared to a model trained on a subset of the Xitsonga data.

MOTS-CLÉS : Réseaux de neurones, phonèmes, segmentation, langues peu dotées.

KEYWORDS: Neural Networks, phonemes, segmentation, under-resourced languages.

1 Introduction

La segmentation de parole est le processus, humain (cognitif) ou automatique (quand il est réalisé par une machine), qui vise à identifier des frontières entre des unités (mots, syllabes, phonèmes) dans un enregistrement ou un flux de parole. En traitement automatique de la parole, c'est un sous-problème qui a diverses applications en reconnaissance automatique de la parole (RAP). Actuellement, la recherche automatique de segments permettant d'identifier des mots ou des unités sous-lexicales est portée par l'intérêt pour l'apprentissage non-supervisé de ces unités, soit pour construire un lexique

de prononciation en identifiant les mots et l'inventaire de phones sans connaissance linguistique *a priori* (Lee *et al.*, 2015), soit pour faire des liens avec l'humain et l'acquisition du langage, en particulier par les enfants (Jansen *et al.*, 2013).

Dans ce contexte, nous pouvons mentionner l'intérêt croissant de la communauté scientifique pour le traitement automatique de langues dites peu-dotées, avec l'organisation de conférences et de sessions spéciales dédiées à ce thème chaque année, comme par exemple le Workshop sur les technologies de la parole pour les langues peu-dotées *SLTU*. À celles-ci s'ajoutent des défis, comme le *Zero Resource Speech Challenge* (Versteegh *et al.*, 2015), qui consistait à identifier des mots ou pseudo-mots, et des unités sous-lexicales à partir d'enregistrements sonores uniquement. Les données utilisées dans ce défi étaient le corpus de parole spontanée BUCKEYE, d'anglais américain, et également un petit corpus d'une langue peu dotée, le Xitsonga, une langue d'Afrique du Sud.

Les réseaux de neurones profonds (DNN) sont devenus populaires dans le monde du traitement du signal en raison de leurs excellentes performances, en particulier en RAP. Selon le problème considéré, ils donnent des résultats similaires ou supérieurs aux GMM. Par exemple, Joshi *et al.* (2015) obtiennent un gain absolu de 3% en classification de voyelles. Les réseaux de neurones ont la particularité de pouvoir s'adapter aux données et à la tâche demandée, s'approchant de la forme la plus adaptée au problème. Bhargava & Rose (2015) ont ainsi constaté que le réseau pouvait imiter des représentations proches des bancs de filtres lorsqu'ils prenaient directement des fenêtres du signal temporel en entrée.

Dans ce travail, nous avons abordé la segmentation automatique en phones en modélisant les frontières de segments plutôt que les segments eux-mêmes. Nous comparons les performances, sur le corpus BUCKEYE, obtenues par des réseaux à couches cachées denses (*Multilayer Perceptron, MLP*) et par des réseaux convolutifs (*Convolutional Neural Networks, CNN*). Après une brève description de notre système dans la section 2, des corpus et métriques d'évaluation en section 3, nous comparons dans la section 4 différentes configurations des modèles (nombre de neurones, de filtres), et illustrons l'influence des données utilisées (faible quantité, langue différente) lors de l'apprentissage des réseaux de neurones.

2 Description du système

Le schéma 1 représente les différentes étapes de notre système permettant d'obtenir la segmentation en phones du signal de la parole. Les trois étapes sont détaillées dans les sous-sections suivantes.

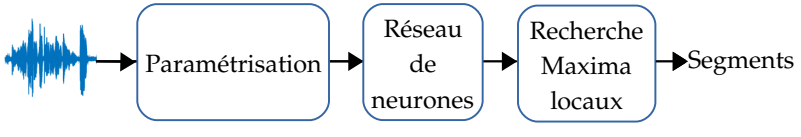

FIGURE 1 – Schéma du système de segmentation en phones

2.1 Paramétrisation

Suite à différents essais de paramètres temporels et fréquentiels, nous avons opté pour des bancs de filtres, calculés sur le signal découpé en fenêtres de 16 ms, avec un pas de 4 ms. Nous extrayons les coefficients FBANK, que nous donnons en entrée du réseau de neurones. Il est rappelé que le processus d'extraction des FBANK est fondé sur la transformation de l'amplitude spectrale grâce à un banc de filtres. Celui-ci est caractérisé par des filtres triangulaires, répartis de manière linéaire selon l'échelle Mel. Le logarithme des énergies obtenu après filtrage est calculé pour obtenir les FBANK.

2.2 Réseaux de neurones

Les CNN sont très efficaces en reconnaissance de formes : plus de 99% de reconnaissance correcte sur des chiffres manuscrits (MNIST) (LeCun *et al.*, 1998), par exemple. Le MLP peut parvenir à des résultats similaires avec davantage de couches : 12 couches totalement connectées contre 6 (1 couche de convolution et 5 totalement connectées) pour un CNN (Golik *et al.*, 2015). Dans cet article, nous comparons ces deux types de réseaux de neurones (CNN et MLP), avec comme objectif de reconnaître les variations dans le spectrogramme marquant un changement de phones, changement qui varie selon les phones considérés.

Le réseau de neurones voit la tâche de segmentation comme une tâche de classification binaire : présence / absence de frontière. Classiquement, lors de l'attribution d'une classe à un individu, il calcule d'abord pour chaque classe la probabilité que l'individu étudié lui appartienne, puis il indique en sortie la classe la plus probable. Cependant, cette dernière étape rencontre deux difficultés : les deux classes de frontière (présence, absence) étant réparties en des proportions inégales (i.e. 1/5, 4/5), les probabilités en sortie sont plus difficilement favorables à la présence de frontière. De plus, lorsqu'une fenêtre a une probabilité élevée d'être une frontière, alors ses voisines ont de grandes chances de l'être elles aussi. Pour éviter cela, il faudrait que le réseau considère les probabilités des fenêtres voisines avant de prendre une décision, ce qui est davantage envisageable avec un réseau de type récurrent. Nous avons choisi de traiter nous-même les probabilités en sortie du réseau pour en déduire les bornes des segments des phones à l'aide d'une méthode de recherche de maxima locaux.

2.3 Recherche de maxima locaux

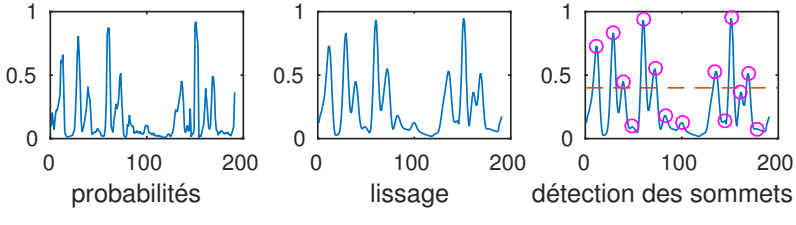

FIGURE 2 – Illustration de notre recherche de maxima locaux sur un enregistrement constitué de 200 fenêtres d'analyse

La figure 2 illustre le processus de recherche des maxima locaux. Pour chaque fenêtre d'analyse, le

réseau de neurones calcule une probabilité que celle-ci contienne une frontière (passage d'un phone à un autre). Chaque enregistrement donne lieu à une courbe de probabilités (200 valeurs sur la figure 2). Pour éviter de détecter des variations locales dues au bruit, nous lissons la courbe à l'aide d'une convolution avec une fenêtre de Hamming de petite taille (5, dans notre cas). Nous détectons ensuite les sommets (maxima locaux) et nous ne conservons que ceux supérieurs à un seuil. La valeur du seuil peut varier selon les besoins de privilégier la précision, le rappel ou la F-mesure. Le seuil maximisant la F-mesure correspond environ au seuil sélectionnant approximativement 12 phones par secondes pour le corpus de parole conversationnelle BUCKEYE et 9 pour le corpus lu de Xitsonga.

3 Corpora et métriques d'évaluation

Nous avons utilisé le corpus d'anglais américain appelé BUCKEYE (Pitt *et al.*, 2007), constitué de parole spontanée (enregistrements de radio) d'une quarantaine de locuteurs différents (hommes, femmes, jeunes, personnes âgées) avec environ 30 minutes de temps de parole pour chacun. Ce corpus est décrit en détails dans (Kiesling *et al.*, 2006). La qualité des annotations manuelles a été évaluée par les créateurs du corpus. Un accord inter-annotateurs a été calculé : il est de l'ordre de 62% de F-mesure pour la segmentation avec 10 ms de marge (décalage). Le pourcentage monte à 79% pour un décalage de 20 ms (Raymond *et al.*, 2002). La durée médiane des phonèmes est d'environ 70 ms, avec une soixantaine de phonèmes différents annotés, chiffre supérieur à la quarantaine habituellement référencée en anglais notamment à cause de prononciations particulières que les auteurs de BUCKEYE ont choisi d'isoler dans des classes différentes, pour les nasales en particulier. En nous basant sur le découpage du challenge *Zero Resource Speech*, nous avons divisé le sous-corpus d'apprentissage en deux parties : un sous-corpus d'entraînement (BUCKEYE-TRAIN, 75%, 10 heures, 20 locuteurs), un corpus de développement (BUCKEYE-DEV, 25%, 3 heures, 6 locuteurs), et nous avons conservé la partie officielle de test (BUCKEYE-TEST, 5 heures, 12 locuteurs) telle quelle.

Le corpus en langue Xitsonga (van Heerden *et al.*, 2013) est composé de courtes phrases lues, enregistrées sur Smartphone hors studio. Nous avons utilisé près de 500 phrases, avec en tout 10000 exemples de phonèmes annotés manuellement, issus de la base de données du même challenge *Zero Resource Speech*. La durée médiane des phones est d'environ 90 ms et il y a 49 phones différents.

Une certaine marge d'erreur est tolérée lors de l'attribution de la frontière. Nous avons utilisée deux marges différentes : la marge la plus courante dans la littérature (20 ms) et une marge plus petite de 10 ms parfois aussi trouvée. Pour évaluer les résultats, nous avons utilisé les métriques classiques de précision, rappel et F-mesure. Selon le seuil choisi pour la recherche des maxima locaux, nous repérons plus ou moins de frontières et obtenons des scores différents. Les courbes DET (*Detection Error Trade-off*), ayant en abscisse le taux de faux positifs et en ordonnée le taux de faux négatifs, nous permettront de visualiser les différents résultats selon les seuils testés (Martin *et al.*, 1997).

4 Expériences

4.1 Comparaison de différentes configurations sur BUCKEYE-DEV

Dans le cadre de cet article, nous avons utilisé Theano (Bastien *et al.*, 2012) et Lasagne (Dieleman *et al.*, 2015) pour la mise en œuvre des modèles. Avant de chercher à optimiser les paramètres, nous

avons tout d'abord comparé le CNN et le MLP. En utilisant les bancs de filtres en entrée, le CNN s'est montré pertinent (taux d'apprentissage=0.007, coefficient de régularisation=0.9). Le MLP a eu, quant à lui, besoin de la dérivée des bancs de filtres pour obtenir des résultats intéressants avec un modèle peu profond. Cette dérivée n'a pas été pertinente pour le CNN et une étude a montré que de sa première couche de convolution effectuait elle-même plusieurs approximations d'une dérivée temporelle. Nous avons donc optimisés les paramètres pour chacun des deux réseaux (nombre de couches, de neurones, dimension des filtres de convolution) avant de faire le choix le plus pertinent.

Suite au choix d'ajouter la dérivée des bancs de filtres en entrée du MLP, celui-ci s'est montré peu sensible à son nombre de couches et nous avons restreint ce nombre à 3 couches cachées. Cependant, le nombre de neurones a eu beaucoup plus d'influence, avec un maximum de performance autour de 300 neurones (cf. figure 3), mais pour une augmentation de seulement 1% de la F-mesure, par rapport à un modèle uniquement constitué de 50 neurones. Le CNN est optimal, quant à lui, entre 50 et 400 neurones pour les couches totalement connectées. Le nombre de filtres de ses couches de convolution a un impact de l'ordre de 1% à 2%, en absolu. Passer de 15 à 120 filtres permet ainsi de gagner 1,2%.

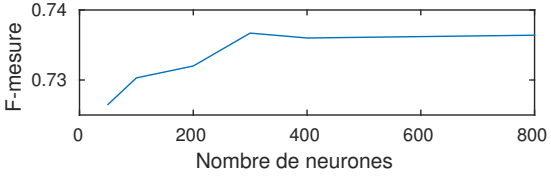

FIGURE 3 – Evolution de la F-mesure en fonction du nombre de neurones des couches cachées du MLP sur BUCKEYE-DEV

Le nombre de fenêtres voisines considérées s'est avéré être l'un des paramètres les plus importants : les changements de phones se repèrent notamment grâce au contexte. Le MLP supporte moins bien l'augmentation du nombre de données (allant avec l'augmentation du nombre de voisins) que le CNN, dont les couches de convolution effectuent visiblement un premier traitement pertinent et réducteur. La figure 4 illustre l'importance de la taille du contexte pour la tâche de segmentation : les résultats s'améliorent de manière visible avec l'augmentation du nombre de voisins. Nous avons choisi 18 voisins (84 ms), l'augmentation au-delà étant très faible par rapport à l'augmentation de la complexité.

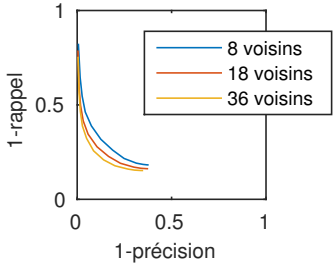

FIGURE 4 – Courbe DET avec un CNN en fonction du voisinage considéré sur BUCKEYE-DEV

La taille du voisinage étant un paramètre influent sur le réseau, nous comparons nos deux modèles (CNN et MLP) en fonction de son évolution. Sur la figure 5, nous voyons que le CNN s'avère plus efficace que le MLP : nous l'utiliserons donc uniquement celui-ci dans la suite du document.

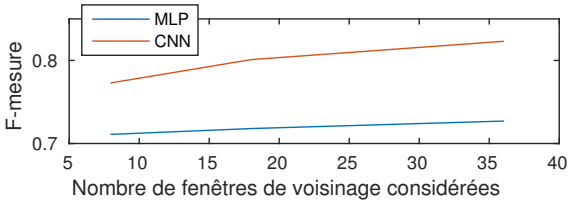

FIGURE 5 – Comparaison des F-mesures du MLP et du CNN selon le nombre de fenêtres de voisinage sur BUCKEYE-DEV

4.2 Résultats sur BUCKEYE-TEST

Avec un CNN composé de 2 couches de convolution dotées de 60 filtres et d'une couche totalement connectée de 200 neurones, nous avons obtenu une F-mesure de 68% pour une tolérance de 10 ms. Nous pouvons obtenir une précision proche de 90%, si nous acceptons de ne trouver qu'un tiers des frontières, ou bien un rappel de 72% avec la moitié des détections erronées (cf. table 1).

Précision	Rappel	F-mesure
0.52	**0.72**	0.61
0.71	0.65	**0.68**
0.94	0.16	0.27

TABLE 1 – Résultats sur BUCKEYE-TEST pour 4 valeurs de seuil et 10 ms de tolérance

La figure 6 est un exemple de résultat obtenu par le réseau de neurones, montrant la courbe des probabilités superposée au spectrogramme du signal. Les valeurs élevées de la courbe correspondent effectivement à des variations dans le spectre et sont corrélées avec les frontières.

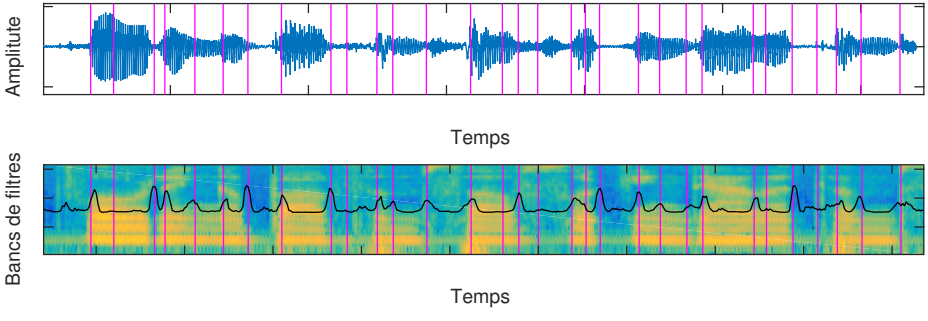

FIGURE 6 – Probabilités en sortie du CNN pour la segmentation sur BUCKEYE-TEST

Nous avons analysé les taux de détection des frontières de quelques phones parmi les plus fréquents (cf. table 2). Nous constatons que les frontières des phones courts avec une forte attaque tels que /g/ ou /k/ sont souvent trouvées alors que le réseau rencontre davantage de difficultés pour le /l/ et le /r/. Ainsi par exemple pour 10 ms de marge, la frontière entre /ao/ et /l/ n'est trouvées que dans 9% des

cas, 15% pour celle entre /aw/ et /r/. Les frontières entre deux voyelles ont aussi de mauvais scores : 8% entre /ow/ et /ay/, 11% entre /er/ et /ay/.

	/r/	/l/	/ao/	/uh/	/g/
% débuts segments détectés	46	45	73	62	81
% fins segments détectées	49	51	39	76	81

TABLE 2 – Analyse des résultats pour 5 phones différents – BUCKEYE-TEST, 10 ms de tolérance

Les résultats en segmentation automatique sont proches de l'erreur constatée entre les annotateurs humains. La table 3 montre même que notre système est plus précis lorsqu'il localise une frontière juste : nous avons une meilleure F-mesure pour 10 ms de tolérance d'erreur et son augmentation entre 10 ms et 20 ms est plus faible que pour celle observée entre les annotateurs.

	annotateurs	CNN
10 ms	0.62	0.68
20 ms	0.79	0.79

TABLE 3 – Comparaison de F-mesures entre l'accord inter-annotateurs et le CNN – BUCKEYE-TEST

4.3 Segmentation avec peu de données d'apprentissage

La segmentation ne séparant les données qu'en deux classes différentes, nous pouvons espérer que peu d'échantillons suffisent pour l'apprentissage, et qu'utiliser un modèle appris sur une langue avec beaucoup de données peut quand même détecter des frontières si nous l'utilisons pour une autre langue, pour laquelle peu de données sont disponibles.

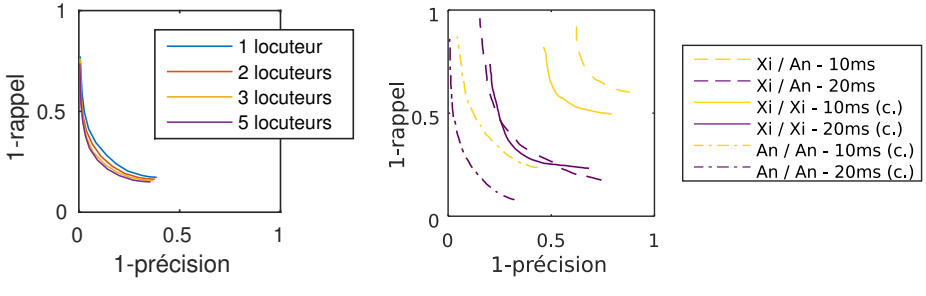

FIGURE 7 – Courbes DET sur des cas difficiles : **à gauche** : peu de locuteurs, **à droite** : apprentissage sur une langue mieux dotée. Notations : (c.) résultats obtenus en validation croisée sur peu de données selon la marge de tolérance (ms), Xi pour Xitsonga, An pour Anglais, [corpus de test]/[corpus d'apprentissage]

La figure 7 montre la bonne adaptation du CNN à des cas peu dotés. Sur le graphique de gauche, nous remarquons qu'apprendre sur un unique locuteur donne bien évidemment des résultats légèrement inférieurs à ceux obtenus à l'aide de plusieurs locuteurs. En utilisant les données de 3 ou 5 locuteurs, nous avons obtenu des résultats très proches et l'amélioration au-delà de 5 locuteurs est presque inexistante.

Sur le graphique de droite, se trouvent différents résultats du CNN dans le cas du Xitsonga, langue peu dotée sur laquelle nous avons effectuée les tests. Les résultats sur le corpus de Xitsonga sont moins bons que sur le corpus d'anglais dans des conditions d'apprentissage similaires (en validation croisée sur le même nombre d'échantillons, de 4 locuteurs différents). Ceci peut s'expliquer par les conditions d'enregistrement différentes : les enregistrements de Xitsonga ayant été réalisés avec des smartphones, hors studio. Un résultat intéressant qui apparaît est que le modèle appris sur le petit corpus de Xitsonga donne des résultats meilleurs avec une tolérance de 10 ms d'erreur que le modèle appris sur BUCKEYE-TRAIN, mais similaires pour une tolérance de 20 ms. Nous pouvons donc en conclure que le modèle appris sur le grand corpus d'anglais se montre surtout moins précis lors de l'attribution des frontières des phones du Xitsonga.

Afin de mieux appréhender les résultats, nous avons affiché un exemple sur la figure 8.

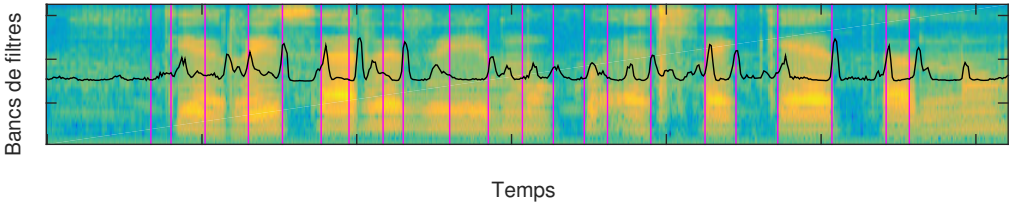

FIGURE 8 – Probabilités de frontières en sortie du CNN appris sur BUCKEYE-TRAIN pour le corpus Xitsonga

5 Conclusion

Dans cet article, nous avons décrit des résultats expérimentaux de segmentation automatique de la parole en phones à l'aide de différents réseaux de neurones (CNN et MLP). Sur les enregistrements d'anglais américain issu du corpus BUCKEYE, nos réseaux de neurones ont obtenu des résultats assez remarquables : 68% de F-mesure pour notre meilleur système de segmentation automatique contre 62% pour l'accord inter-annotateurs, avec une tolérance de 10 ms sur la localisation des frontières de phones. De plus, les modèles ont fait preuve d'une bonne adaptation à des cas particuliers difficiles : peu de données d'apprentissage et application à une langue différente de celle de l'apprentissage. En particulier, des performances similaires ont été obtenues sur un petit corpus de la langue peu dotée Xitsonga en utilisant : 1) un modèle entraîné sur l'anglais américain, 2) un modèle entraîné sur un sous-corpus de petite taille de la langue peu dotée. Ce résultat nous fait supposer que des modèles entraînés sur des langues disposant de grandes quantités de données peuvent être utilisés avec des langues peu dotées en première approche (Renshaw *et al.*, 2015). Pour la segmentation de langues peu dotées, telles que le Xitsonga, nous envisageons de réaliser des expériences d'apprentissage semi-supervisé comme le *bootstrap*, en utilisant des modèles entraînés sur l'anglais américain et d'autres grands corpora.

Références

BASTIEN F., LAMBLIN P., PASCANU R., BERGSTRA J., GOODFELLOW I. J., BERGERON A., BOUCHARD N. & BENGIO Y. (2012). Theano : new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

BHARGAVA M. & ROSE R. (2015). Architectures for deep neural network based acoustic models defined over windowed speech waveforms. p. 6–10.

DIELEMAN S., SCHLÜTER J., RAFFEL C., OLSON E., SØNDERBY S. K., NOURI D., MATURANA D., THOMA M., BATTENBERG E., KELLY J., FAUW J. D., HEILMAN M., DIOGO149, MCFEE B., WEIDEMAN H., TAKACSG84, PETERDERIVAZ, JON, INSTAGIBBS, RASUL D. K., CONGLIU, BRITEFURY & DEGRAVE J. (2015). Lasagne : First release.

GOLIK P., TÜSKE Z., SCHLÜTER R. & NEY H. (2015). Convolutional neural networks for acoustic modeling of raw time signal in lvcsr. p. 26–30.

JANSEN A., DUPOUX E., GOLDWATER S., JOHNSON M., KHUDANPUR S., CHURCH K., FELDMAN N., HERMANSKY H., METZE F., ROSE R. *et al.* (2013). A summary of the 2012 jhu clsp workshop on zero resource speech technologies and models of early language acquisition.

JOSHI S., DEO N. & RAO P. (2015). Vowel mispronunciation detection using dnn acoustic models with cross-lingual training. p. 697–701.

KIESLING S., DILLEY L. & RAYMOND W. D. (2006). The variation in conversation (vic) project : Creation of the buckeye corpus of conversational speech. p. 55–97.

LECUN Y., BOTTOU L., BENGIO Y. & HAFFNER P. (1998). Gradient-based learning applied to document recognition. p. 2278–2324.

LEE C.-Y., O'DONNELL T. J. & GLASS J. (2015). Unsupervised lexicon discovery from acoustic input. *Transactions of the Association for Computational Linguistics*, **3**, 389–403.

MARTIN A., DODDINGTON G., KAMM T., ORDOWSKI M. & PRZYBOCKI M. (1997). *The DET curve in assessment of detection task performance*. Rapport interne, DTIC Document.

PITT M., DILLEY L., JOHNSON K., KIESLING S., RAYMOND W., HUME E. & FOSLER-LUSSIER E. (2007). Buckeye corpus of conversational speech (2nd release). www.buckeyecorpus.osu.edu.

RAYMOND W. D., PITT M., JOHNSON K., HUME E., MAKASHAY M., DAUTRICOURT R. & HILTS C. (2002). An analysis of transcription consistency in spontaneous speech from the buckeye corpus.

RENSHAW D., KAMPER H., JANSEN A. & GOLDWATER S. (2015). A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge. p. 3199–6303.

VAN HEERDEN C., DAVEL M. & BARNARD E. (2013). The semi-automated creation of stratified speech corpora.

VERSTEEGH M., THIOLLIÈRE R., SCHATZ T., CAO X. N., ANGUERA X., JANSEN A. & DUPOUX E. (2015). The zero resource speech challenge 2015. p. 3169–3173.