Réordonnancement automatique d'hypothèses pour l'assistance à la transcription de la parole

Antoine Laurent[†§], *Sylvain Meignier*[†], *Paul Deléglise*[†]

LIUM[†] – Laboratoire d'Informatique de l'Université du Maine – Le Mans
prenom.nom@lium.univ-lemans.fr

Spécinov[§] – Trélazé
a.laurent@specinov.fr

ABSTRACT

Large vocabulary automatic speech recognition (ASR) technologies perform well in known, controlled contexts. However, some mistakes still have to be corrected. Human intervention is necessary to check and correct the results of such systems in order to make the output of ASR understandable. We propose a method for computer-assisted transcription of speech, based on automatic reordering confusion networks. It allows to significantly reduce the number of actions needed to correct the ASR outputs. WER computed before and after every network reordering shows an absolute gain of about 3.4%.

Keywords: Speech recognition, Automatic correction, Cache models, Confusion network

1. Introduction

Cet article présente une méthode d'assistance à la transcription automatique de la parole. Le transcripteur humain dispose de la meilleure hypothèse fournie par un système de reconnaissance automatique de la parole (SRAP). A chaque correction de sa part, le système propose une nouvelle transcription prenant en compte cette correction. Cette méthode permet au système et au correcteur de collaborer pour converger plus rapidement vers une transcription correcte (sans erreur).

Dans la littérature, peu d'articles sont consacrés à cette tâche. Dans l'article [9], les auteurs proposent de relancer le processus de décodage après chaque correction de l'utilisateur en se basant sur celle-ci. L'inconvénient majeur de cette méthode est qu'elle risque d'avoir un impact négatif sur le temps de réaction de l'interface homme machine.

D'autres travaux [3, 10, 6] ont été réalisés sur la traduction assistée par ordinateur (Computer-Assisted Translation - CAT). Ces types de systèmes proposent une traduction qui est lue par l'utilisateur. Dès qu'un mot erroné est corrigé, le système propose une traduction alternative. Certains travaux [1, 11, 2] présentent des méthodes de traduction assistées utilisant en entrée, non pas du texte, mais du langage parlé. L'idée générale consiste à utiliser un modèle de langage combinant un modèle de langage n-gramme avec les probabilités de traduction de chaque mot.

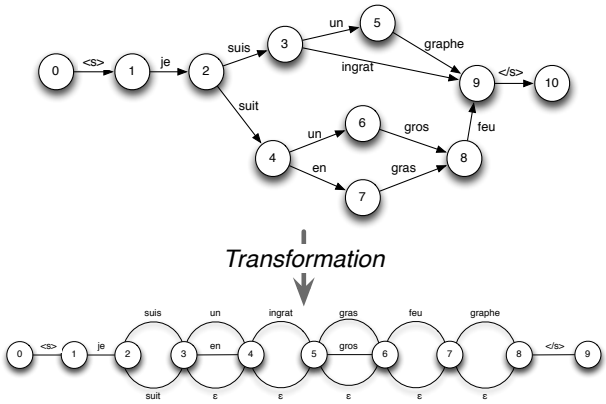

Fig. 1: Obtention d'un réseau de confusion

2. Méthode

La méthode proposée s'inspire des méthodes développées pour la tâche de CAT. Elle consiste à réévaluer la meilleure hypothèse d'un réseau de confusion en fonction des corrections apportées par l'utilisateur, sans nécessiter un décodage complet.

2.1. Réseau de confusion

Le réseau de confusion est obtenu à partir des treillis de mots construits lors du décodage. Les treillis de mots représentent, après élagage, tous les chemins hypothèses développés. Chaque état correspond à un instant dans l'enregistrement à transcrire, chaque lien représente un mot auquel est associé une probabilité.

Ce graphe est transformé en réseau de confusion (figure 1). La transformation consiste à fusionner les mots localement identiques, à regrouper sur des ensembles de confusion communs les mots temporellement proches, et à supprimer les chemins d'hypothèse trop faibles [8]. Chaque mot obtient un nouveau score qui est sa probabilité *a posteriori* (obtenue à partir du treillis de mots) divisée par la somme des probabilités *a posteriori* des mots en concurrence avec lui. Le symbole ϵ représente une transition vide (absence de mot).

Les réseaux de confusion contiennent, en plus de l'ensemble des mots en concurrence, des temps approximatifs obtenus à partir du graphe de mot. Cet ajout est une extension par rapport à [8]. L'instant de départ d'un état du réseau de confusion correspond au plus petit instant des mots associés à cet état ; le

temps de fin correspond à l'instant de fin du dernier de ces mots. La meilleure hypothèse de reconnaissance qui sera donnée à corriger au transcripteur est la séquence de mots qui maximise chacune des probabilités *a posteriori* du réseau de confusion.

2.2. Principe

A partir d'une séquence d'observations acoustiques X, l'objectif du SRAP est de trouver la séquence de mots $\hat{W}$ la plus probable parmi l'ensemble des séquences possibles W. La recherche de $\hat{W}$ maximisant la probabilité d'émission de W sachant X correspond à l'équation suivante après application du théorème de Bayes et simplification :

$$\hat{W} = \arg\max_{W} P(X|W)P(W) \quad (1)$$

$P(W)$ est fourni par le modèle de langage, la probabilité $P(X|W)$ correspond à la probabilité attribuée par le modèle acoustique. Dans notre cas, la séquence de mots possibles W est séparée en deux : un préfixe p qui a été validé et/ou corrigé par l'utilisateur et un suffixe s à déterminer en fonction du préfixe p. Nous cherchons donc la séquence de mots $\hat{s}$, parmi toutes les séquences de suffixes possibles s maximisant l'équation suivante :

$$\hat{s} = \arg\max_{s} P(X|s,p)P(s|p) \quad (2)$$

La méthode présentée ne remet pas en cause l'acoustique, $P(X|s,p)$ est constant. $P(s|p)$ sera calculé à partir de la combinaison linéaire entre un modèle de langage quadrigramme P_{4G} et un modèle cache P_{cache} [4]. Nous cherchons donc, parmi tous les suffixes candidats $\tilde{s}$ dans le réseau de confusion, le suffixe $\hat{s}$ maximisant la probabilité suivante :

$$\hat{s} = \arg\max_{\tilde{s}}((1-\lambda)P_{4G}(s|p) + \lambda P_{cache}(s|p)) \quad (3)$$

Le modèle cache est construit à partir des mots contenus dans le préfixe p. Il permet de renforcer la probabilité des mots récemment rencontrés, on suppose ici qu'ils ont une plus forte chance d'apparaître dans le futur (dans s).

Plusieurs types de modèles caches sont proposés dans la littérature. Dans l'application proposée, les meilleurs résultats en terme de perplexité ont été obtenus à partir de la méthode proposée dans [4]. La probabilité d'apparition du mot w_i est exponentiellement proportionnelle à la distance entre la position actuelle et les apparitions précédentes du mot w_i dans l'historique h_i :

$$P_{cache}(w_i|h_i) = \beta \sum_{j=1}^{i-1} I_{\{w_i=w_j\}} e^{-\alpha(i-j)} \quad (4)$$

avec α le coût du décalage dans le cache, $I_{w_i=w_j} = 1$ si $w_i = w_j$ et 0 sinon, et β est une constante de normalisation calculée comme suit :

$$\beta = \frac{1}{\sum_{j=1}^{i-1} e^{-\alpha j}} \quad (5)$$

2.3. Application

A partir du réseau de confusion et du préfixe corrigé et/ou validé, la recherche des suffixes candidats dans le réseau de confusion est effectuée de la manière suivante. Soit t l'instant de fin du dernier mot validé. Le principe va être de rechercher, parmi tous les états suivants du réseau de confusion ceux ayant un instant de début supérieur ou égal à t. La recherche est récursive. Pour chacun des états concurrents, nous recherchons à nouveau les états pouvant lui succéder et ainsi de suite. L'utilisation de ces temps, bien qu'approximatifs, permet d'éviter de choisir des séquences de mots dans lesquelles certains mots se chevaucheraient. La méthode proposée ne se déclenche que lorsque l'utilisateur remplace un mot par un autre (erreur de substitution).

Le réordonnancement automatique s'arrête dès que le dernier mot proposé correspond à un mot qui était présent dans l'hypothèse initiale. Si l'utilisateur fait le choix de supprimer un mot (mot incorrect glissé entre deux mots corrects) ou d'en insérer un (mot manquant entre deux mots corrects) plutôt que de faire une correction (substitution), nous avons fait l'hypothèse que le mot suivant était juste.

La première étape va consister à rechercher à quel état du réseau de confusion le mot venant d'être corrigé peut être rattaché. Si ce mot est présent dans le graphe à l'endroit de la correction, il sera rattaché à l'état correspondant, s'il n'est pas présent, le mot sera ajouté à l'état du mot substitué. Cette première étape réalisée, toutes les séquences possibles de mots pouvant succéder à l'état sélectionné du graphe sont recherchées, en respectant les indices temporels.

Le score calculé grâce à l'équation (3) permet de départager les différentes séquences possibles. Si deux séquences de mots ont la même probabilité (cas exceptionnel), la probabilité *a posteriori* moyenne de la séquence de mots devient l'élément discriminant. Cette stratégie d'auto-réordonnancement s'arrête dès que le dernier mot proposé automatiquement est identique à celui qui était dans l'hypothèse précédente (figure 2).

2.4. Cas des mots hors vocabulaire

La liste de mots pouvant être proposée de façon automatique est limitée aux mots se trouvant dans le réseau de confusion. Si l'utilisateur saisit un mot inconnu du SRAP dans le préfixe, la recherche du suffixe parmi les différents suffixes candidats fera intervenir la probabilité du mot inconnu. Ce mot sera ajouté au modèle cache, mais il n'est pas possible, en l'état actuel, que ce mot nouveau réapparaisse de façon automatique dans la suite des corrections.

3. Expériences

Les expériences menées simulent le comportement d'un transcripteur corrigeant la meilleure hypothèse du réseau de confusion généré par le SRAP.

3.1. Corpus & SRAP

L'optimisation des coefficients du modèle cache a été réalisée sur le corpus de test d'ESTER 1. Les expériences ont été effectuées sur le corpus de test d'ESTER 2 [7]. Il s'agit d'émissions radiophoniques francophones, complétées par des articles provenant du journal "Le Monde". Les réseaux de confusion sont créés à partir du graphe d'hypothèse des mots généré par le SRAP du LIUM développé lors de la campagne ESTER 2. Le décodage s'effectue en 5 passes détaillées dans l'article [5]. Sans l'ajout de traitements particuliers pour les segments provenant de la radio africaine, le taux d'erreur mot du système sur le corpus de test de la campagne d'ESTER 2 est de 19,2%.

3.2. Métriques

La méthode est évaluée selon deux métriques. Le taux d'erreur mot classique (WER) et le KSR (*Keystroke Saving Rate*) [12].

Le KSR a été mis en place dans les systèmes de communication assistés destinés aux handicapés. Il se calcule de la façon suivante :

$$KSR = (1 - \frac{k_p}{k_a}) \times 100 \qquad (6)$$

Où k_p est le nombre d'appuis effectivement réalisés par l'utilisateur lors de la saisie d'un message et k_a le nombre d'appuis qui auraient été nécessaires sans aide à la composition de mots. Ces appuis peuvent être des appuis sur un clavier ou sur un dispositif particulier mis en place pour la gestion de l'handicap de l'utilisateur : joystick, clignement d'un oeil, etc.

Dans notre cas, k_p représentera le nombre d'actions réalisées par l'utilisateur pour corriger l'hypothèse du SRAP, en utilisant un clavier, et k_a le nombre d'actions qui auraient été nécessaires en partant d'une hypothèse vide (ne contenant pas de mots).

Pour calculer le KSR, il est supposé que l'utilisateur appuiera sur le moins de touches possible pour obtenir la transcription corrigée. Deux stratégies sont retenues pour minimiser le nombre d'actions : soit tous les mots de la zone erronée sont supprimés puis remplacés par les mots corrects, soit le maximum de lettres de l'hypothèse sont conservées et les actions ne portent que sur les lettres erronées.

Un alignement entre l'hypothèse générée par le système de reconnaissance automatique de la parole et la référence de transcription est effectué. Cet alignement sera réalisé au niveau des mots et au niveau des lettres correspondant aux zones en erreur.

Les coûts des actions sont les suivants :
- Les coûts de déplacement à l'intérieur du texte de mot en mot ne sont pas pris en compte. L'application d'aide à la correction, comme celle de [9], présente les mots un à un lors du procédé d'aide à la transcription, et chaque mot est corrigé lors de son apparition.
- La suppression d'un mot coûte 1 (raccourci clavier permettant de supprimer un mot entier).
- L'appui sur une touche du clavier coûte une action.

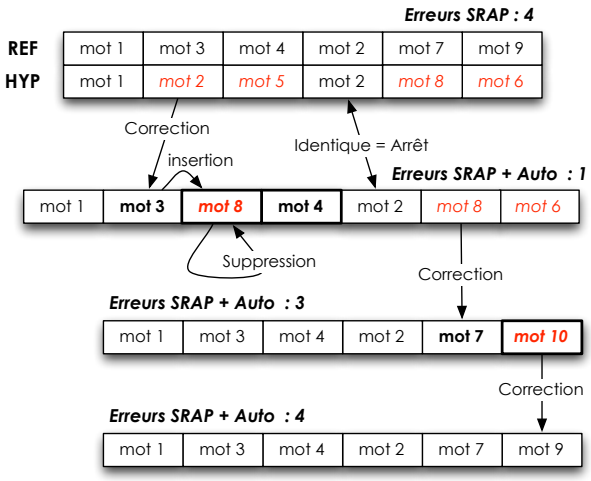

Fig. 2: Calcul du taux d'erreur mot de la méthode de réordonnancement automatique

Cette méthode de calcul étant relativement subjective, puisqu'elle ne prend pas en compte la pénibilité de chaque action pour l'utilisateur, la méthode a également été évaluée en terme de WER. Le nombre d'erreurs (Insertion + Substitution + Suppression) a été calculé pour chaque segment avant et après chaque procédure de correction automatique. Quand la méthode automatique ne se déclenche pas, son coût est considéré identique à celui de la méthode manuelle (voir figure 2).

La figure 2 présente un cas où la méthode de réordonnancement ne permet pas d'observer de gain en terme de taux d'erreur mot.

3.3. Résultats

Dans un premier temps, le nombre d'actions à réaliser par l'utilisateur a été calculé en ne lui proposant que la sortie du système de reconnaissance automatique à corriger, sans autre aide. La possibilité de remplacer un mot par un autre par simple sélection d'un mot concurrent dans une liste de mots a ensuite été évaluée. La liste est constituée des mots présents dans les réseaux de confusion à cet instant. Nous supposons que la liste des concurrents est toujours visible à l'écran, un coût de 1 est attribué pour le remplacement d'un mot de cette manière.

Le tableau 1 présente un résumé des résultats obtenus dans différentes configurations. La référence est composée de 435 005 lettres (caractères espace compris). Il faut donc appuyer 435 005 fois sur les touches du clavier pour la saisir dans sa totalité (Manuelle dans le tableau). En utilisant les sorties du système de reconnaissance automatique de la parole (ligne SRAP), ce nombre d'actions est de 53 492. L'utilisation du système de reconnaissance automatique de la parole a donc permis de réaliser un gain en terme de *KSR* de 87,7%, pour un taux d'erreur mot de 19,2%. Avec les listes déroulantes (ligne SRAP+liste), le nombre d'actions à effectuer par l'utilisateur diminue et passe à 52 003, soit un *KSR* de 88%. Les sorties du SRAP associées à la mise en oeuvre de la méthode de réordonnancement automatique permet d'observer un *KSR* de 89,2% (46 795 actions), pour un taux d'er-

reur mot de 17% (ligne SRAP+auto). L'ajout de la sélection des mots dans une liste déroulante permet de diminuer ce nombre d'actions à 44 732, soit un *KSR* de 89,7% (SRAP+auto+liste). Enfin, le système complet (SRAP+auto+liste+cache) utilisant le modèle cache, la technique de réordonnancement automatique et la sélection des mots dans la liste déroulante permet d'observer un *KSR* de 90,3% (41 992 actions), et un WER de 15,8%.

Tab. 1: *KSR et WER sur le corpus de test ESTER 2*

Méthode	Nb actions	KSR	WER
Manuelle	435005	0%	–
SRAP	53492	87,7%	19,2%
SRAP+liste	52003	88,0%	19,2%
SRAP+auto	46795	89,2%	17,0%
SRAP+auto+liste	44732	89,7%	17,0%
SRAP+auto+liste+cache	41992	90,3%	15,8%

En terme de taux d'erreur mot, la méthode proposée permet d'obtenir un gain d'environ 3,4% sur le corpus de test d'ESTER 2.

Il est à noter que lorsque le WER diminue, le nombre d'actions à réaliser suit la même tendance. En effet, le WER passe de 19,2% à 15,8%, soit un gain relatif de 17,7%. Le nombre d'actions, quant à lui, chute de 53 492 à 41 992, soit un gain relatif de 21,5%. Lorsque l'on compare le nombre d'actions nécessaires à la correction de la sortie du SRAP seul, avec l'utilisation du SRAP complété de la méthode de réordonnancement automatique, là encore, le nombre d'actions et le WER diminuent tous les deux : 12,5% de gain relatif en terme de nombre d'actions et 11,4% de gain relatif en terme de WER.

4. Conclusion

Cet article présente une technique de réordonnancement automatique des hypothèses du SRAP. Cette technique permet d'observer un gain de 3,4% absolu en terme de WER et de diminuer le nombre d'actions de 21,5% (relatif) par rapport à l'utilisation seule des sorties du système de reconnaissance automatique de la parole. Certaines améliorations peuvent encore être apportées à cette méthode, puisqu'elle ne permet pas, pour l'instant, l'ajout automatique de nouveaux mots (*ie* de mots qui ne seraient pas présents dans le réseau de confusion). La méthode pourrait également être améliorée en propageant les corrections apportées par l'utilisateur. Pour l'instant, la correction d'un mot déclenche le réordonnancement automatique des hypothèses du réseau de confusion pour la fin du segment en cours de correction. Cette correction pourrait avoir un impact sur d'autres segments se trouvant plus éloignés dans la transcription.

Références

[1] J. C. Amengual, J. M. Benedí, A. Castaño, A. Castellanos, V. M. Jiménez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, and J.M. Vilar. The EuTrans-I speech translation system. *Machine Translation*, 14(3) :941–951, 2000.

[2] F. Casacuberta, E. Vidal, A. Sanchis, and J.M. Vilar. Pattern recognition approaches for speech-to-speech translation. *Cybernetic and Systems : an International Journal*, 35(1) :3–17, 2004.

[3] J. Civera, J.M. Vilar, E. Cubel, A.L. Lagarda, S. Barrachina, F. Casacuberta, and E. Vidal. A novel approach to computer assisted translation based on finite-state transducters. *Proceedings of Finite-State Methods and Natural Language Processing (FSMNLP)*, 4002 :32–42, 2006.

[4] P.R. Clarkson and A. J. Robinson. Language model adaptation using mixtures and an exponentially decaying cache. In *Proceedings of International Conference on Acoustics Speech and Signal Processing (IEEE, ICASSP 97)*, pages 799–802, 1997.

[5] P. Deléglise, Y. Estève, and S. Meignier. Improvements to the lium french asr system based on cmu sphinx : what helps to significantly reduce the word error rate ? In *Proceedings of International Conference on Spoken Language Processing (ISCA, Interspeech 2009)*, Brighton, UK, 2009.

[6] G. Foster. *Text Prediction for Translators*. PhD thesis, Université de Montréal, 2002.

[7] S. Galliano, G. Gravier, and L. Chaubard. The ESTER 2 Evaluation Campaign for the Rich Transcription of French Radio Broadcasts. In *Proceedings of International Conference on Spoken Language Processing (ISCA, Interspeech 2009)*, Brighton, UK, September 2009.

[8] H. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition : Word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4) :373–400, 2000.

[9] L. Rodríguez, F. Casacuberta, and E. Vidal. Computer Assisted Transcription of Speech. *Lecture Notes In Computer Science, Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, Part I*, 4477 :241–248, 2007.

[10] J. Tomás and F. Casacuberta. Statistical phrase-based models for interactive computer-assisted translation. In *Proceedings of Coling/Association for Computational Linguistics*, pages 835–841, Sydney, Australia, 2006.

[11] E. Vidal, F. Casacuberta, L. Rodríduez, J. Civera, and C. Martnez. Computer-assisted translation using speech recognition. *IEEE Transaction on Audio, Speech and Language Processing*, 14(3) :941–951, 2006.

[12] M. Wood and E. Lewis. Windmill - the use of a parsing algorithm to produce predictions for disabled persons. In *Proceedings of the 1996 Autumn Conference on Speech and Hearing*, pages 315–322, 1996.