Boosting de bonzaïs pour la combinaison efficace de descripteurs : application à l'identification du rôle du locuteur

Antoine Laurent[1], Nathalie Camelin[2] Christian Raymond[3]
(1)LIMSI, Orsay, France
(2) LIUM, Le Mans, France
(3) IRISA-INSA de Rennes, France
antoine.laurent@limsi.fr,nathalie.camelin@lium.univ-lemans.fr
christian.raymond@irisa.fr

RÉSUMÉ

Dans ce travail, nous nous intéressons au problème de la détection du rôle du locuteur dans les émissions d'actualités radiotélévisées. Dans la littérature, les solutions proposées sont de combiner des indicateurs variés provenant de l'acoustique, de la transcription et/ou de son analyse par des méthodes d'apprentissage automatique. De nombreuses études font ressortir l'algorithme de boosting sur des règles de décision simples comme l'un des plus efficaces à combiner ces différents descripteurs. Nous proposons ici une modification de cet algorithme état-de-l'art en remplaçant ces règles de décision simples par des mini arbres de décision que nous appelons bonzaïs. Les expériences comparatives menées sur le corpus EPAC montrent que cette modification améliore largement les performances du système tout en réduisant le temps d'apprentissage de manière conséquente.

ABSTRACT

Boosting bonsai trees for efficient features combination : application to speaker role identification

In this article, we tackle the problem of speaker role detection from broadcast news shows. In the literature, many proposed solutions are based on the combination of various features coming from acoustic, lexical and semantic informations with a machine learning algorithm. Many previous studies mention the use of boosting over decision stumps to combine efficiently theses features. In this work, we propose a modification of this state-of-the-art machine learning algorithm changing the weak learner (decision stumps) by small decision trees, denoted bonzaï trees. Experiments show that using bonzaï trees as weak learners for the boosting algorithm largely improves both system error rate and learning time.

MOTS-CLÉS : identification rôle locuteur, boosting, arbre de décision.

KEYWORDS: Speaker role recognition, boosting, decision tree.

1 Introduction

Dans cet article nous nous intéressons à la détection du rôle des locuteurs dans les émissions radiotélévisées. Dans la littérature, ce problème est ramené à un problème de classification multi-classes où chaque locuteur d'une émission doit être associé avec un rôle. Suivant cette approche, les études précédentes ont attaqué le problème en utilisant des algorithmes de classification

supervisée utilisant comme descripteurs des indices lexicaux extraits de la transcription (Barzilay *et al.*, 2000; Damnati et Charlet, 2011; Liu, 2006), des indices acoustiques/prosodiques (Salamin *et al.*, 2009; Bigot *et al.*, 2010), ou les deux (Dufour *et al.*, 2012). Ces études ont mis en avant l'efficacité de l'algorithme de boosting sur des règles de décision à combiner ces différents descripteurs. Nous proposons ici une modification de cet algorithme de classification, en remplaçant les règles de décision simples par des mini arbres de décision que nous appelons bonzaïs afin d'améliorer l'efficacité de combinaison de ces différents descripteurs. Nous proposons alors un système d'identification de rôle de locuteur exploitant de multiples indices de descriptions comparable au système de (Dufour *et al.*, 2012) à l'exception notable de notre algorithme de classification. Nos expériences comparatives montrent que notre algorithme combine les indices de description de manière beaucoup plus performante que l'algorithme original tout en diminuant le temps nécessaire à l'apprentissage. Le taux de prédiction des rôles est ainsi amélioré relativement de 19,5% tandis que le temps d'apprentissage est diminué d'un facteur 3.

La suite du papier est organisée de la manière suivante : le chapitre 2 présente notre système de reconnaissance de rôle avec notamment les descripteurs utilisés, le chapitre 2.2 expose les modifications de l'algorithme de classification que nous proposons et les bénéfices que nous espérons en tirer, la section 3 présente les expériences comparatives effectuées sur le corpus EPAC.

2 Système de reconnaissance de rôle du locuteur

Le problème d'identification du rôle du locuteur est ici vu comme un problème de classification multi-classes avec un jeu de descripteurs variés. Les descripteurs utilisés puis l'algorithme de classification sont présentés dans les deux sous-sections suivantes.

2.1 Ensembles de descripteurs

2.1.1 Caractérisation de la parole spontanée

Une méthode permettant de détecter automatiquement la parole spontanée dans des documents audio a été proposée par (Dufour *et al.*, 2009). L'objectif de cet outil est d'associer à chaque segment de parole une des trois classes de spontanéité : parole *préparée*, *faiblement spontanée* et *fortement spontanée*. Des caractéristiques acoustiques (durées des voyelles, durées phonémiques, pitch...) et linguistiques (nombre de répétitions et de noms propres, taille du découpage syntaxique...) sont extraites pour chaque segment à partir d'un système de transcription automatique de la parole (Deléglise *et al.*, 2009). De plus amples informations sur les caractéristiques peuvent être trouvées dans (Dufour *et al.*, 2009).

2.1.2 Analyse des réseaux sociaux (SNA)

Nous avons également intégré des indices prenant en compte la position de chaque locuteur dans le dialogue, c'est à dire permettant de savoir comment celui-ci interagit avec les autres locuteurs. Des indices SNA sont exploités notamment dans (Garg *et al.*, 2008; Wang *et al.*, 2011). L'efficacité de ces indices, combinés à ceux de « **SPONTA** », a été montrée dans le système (Dufour *et al.*, 2012). L'objectif de cette méthode est d'être capable de déterminer la *centralité* de chaque locuteur par rapport aux autres dans une émission, en considérant que le locuteur i dialogue

avec le locuteur j si j intervient juste après i dans la transcription. En s'inspirant de (Vinciarelli, 2007), la centralité se calcule selon l'équation :

$$C_i = \frac{\sum_{j=1}^{nb} \chi D_{i,j}}{\sum_{j=1}^{nb} D_{i,j}} \quad (1)$$

Avec $\chi = 1$ si $D_{i,j} = 1$, et $\chi = 0$ sinon, C_i la centralité de i, nb le nombre de locuteurs et $D_{i,j}$ la distance entre i et j. Cette distance est exprimée en nombre de liens (orientés) à parcourir pour atteindre chaque nœud.

En plus de la centralité, la *couverture du locuteur*, correspondant au temps écoulé entre la première et la dernière intervention du locuteur dans l'émission, est calulée puis normalisée par la durée de l'émission.

Dans la suite, « **SNA** » fera référence aux indices *centralité* et *couverture du locuteur*.

2.1.3 *Ngrammes* multi-niveaux

Parmi les indices les plus pertinents, se trouvent ceux extraits de la transcription elle-même. La méthode usuelle est l'extraction de *Ngramme* de mots. Parfois d'autres niveaux d'informations sont ajoutés, tels que les lemmes, les catégories syntaxiques (POS) des mots ou tout autre niveau qui est pertinent pour la tâche visée. Usuellement, les *Ngrammes* sont extraits d'un unique niveau, indépendamment des autres (Wang *et al.*, 2011). Ainsi, les *Ngrammes* extraits des mots sont très spécifiques et ont peu de pouvoir de généralisation, tandis que les *Ngrammes* extraits de la séquence d'étiquettes syntaxiques correspondante généralisent souvent trop et ne sont pas assez précis.

Un classifieur peut tirer avantage des deux informations mais les considérer de manière séparée n'est pas optimal. Ici nous proposons d'extraire des *Ngrammes* multi-niveaux. Ces *Ngrammes* sont construits en générant la liste exhaustive des combinaisons possibles entre ces différents niveaux d'informations. L'idée est de capter les avantages des différents niveaux en composant des *Ngrammes* à partir de ces différents niveaux. Ainsi, on espère combiner la précision des *Ngrammes* de mots avec le pouvoir généralisant des *Ngrammes* de composants plus haut niveau. Dans ce travail, 3 niveaux d'informations associés à la transcription seront utilisés :

1. la transcription en mot
2. la séquence d'étiquette syntaxique correspondante (POS) prédite par TreeTagger (Schmid, 1994)
3. une combinaison de 2 éléments : l'entité nommée du mot si il en existe une, son lemme sinon. L'entité nommée est prédite selon le système (Raymond et Fayolle, 2010)

La présence de ces indices de descriptions dans notre système sera référencée par « **TEXT** ».

Sur la base de ces indices, 4 sortes de *Ngrammes* pourront être créés. Des *Ngrammes* uniquement composés de mots (W), des *Ngrammes* uniquement composés de POS (P), des *Ngrammes* uniquement extraits à partir du niveau EN/lemme (LNE) et également des *Ngrammes* multi-niveaux ($WPLNE$), *ie.* composés des 3 niveaux à la fois .

Lors de nos expériences, nous testerons le système avec différente jeu de descripteurs *Ngrammes* et notamment le jeu de descripteurs *Ngrammes* extrait à partir de tous les niveaux indépendamment, que nous noterons $W - P - LNE$, ou alors générés à partir de tous les niveaux (*Ngrammes* multi-niveaux) que nous noterons $WPLNE$.

Il est à noter que dans le cas d'utilisation d'*uni*-grammes, les *Ngrammes* multi-niveaux sont identiques à ceux extraits séparément.

2.2 Boosting de bonzaïs

De nombreuses études sur la reconnaissance de rôle du locuteur (Garg *et al.*, 2008; Barzilay *et al.*, 2000; Damnati et Charlet, 2011; Dufour *et al.*, 2009; Wang *et al.*, 2011) ont révélé l'efficacité de l'algorithme AdaBoost sur des règles de décision simples (*ie.* arbres de décision à deux feuilles) qui combinent différents descripteurs. Effectivement, les règles de décision sont induites directement à partir des différents indices tandis que l'algorithme de boosting permet de les combiner afin de produire un classifieur final très efficace.

En plus de cela, parmi les variantes d'algorithmes de boosting, certains d'entre eux sont multi-classes, comme AdaBoost.MH (Schapire et Singer, 2000). C'est notamment celui ci qui est appliqué dans de précédents travaux (Dufour *et al.*, 2012; Barzilay *et al.*, 2000). Bien que cet algorithme a montré de bons résultats, la combinaison linéaire des règles de décisions utilisée peut être insuffisante pour capturer certaines structures dans les données (*eg.* impossible de résoudre le problème du XOR). Afin de palier ce problème, l'idée appliquée ici consiste à implémenter l'algorithme de boosting sur des arbres de décision.

Nous avons décidé d'expérimenter le boosting sur de petits arbres de décision, que nous appelons *bonzaïs*, contraints par leur profondeur, suffisamment profond pour pouvoir capturer des structures dans les données mais pas trop pour conserver des règles générales et éviter le sur-apprentissage.

Dans les situations comme la notre exploitant des descripteurs de type *Ngramme*, le nombre de descripteurs s'élèvent très facilement à plusieurs millions même avec un N relativement faible. De ce fait, nous attendons 3 bénéfices de l'implémentation du boosting de bonzaïs :

1. un bonzaï étant plus complexe qu'une règle de décision, il est capable de modéliser et donc de capturer des informations plus complexes à partir des données d'apprentissage, nous espérons alors un gain de performance ;
2. pour les mêmes raisons, nous espérons que notre algorithme de boosting produise un classifieur aussi performant que l'original avec moins d'itérations que l'algorithme original n'en nécessite. La construction du bonzaï étant intrinsèquement parallélisable, alors que l'algorithme de boosting ne l'étant pas, la réduction du nombre d'itération permet d'espérer un gain de temps important sur l'apprentissage ;
3. avec l'utilisation de bonzaïs en tant que classifieur de base, nous nous demandons si le bonzaï n'est pas capable d'implicitement créer des *Ngrammes* à partir d'observations isolées telles que le sac de mots (*uni*-grammes). Si cela était le cas, il ne serait plus utile de générer explicitement les *Ngrammes* mais de se contenter de générer des *uni*-grammes, ce qui diminuerait drastiquement le nombre de descripteurs et par conséquent le temps d'apprentissage nécessaire à l'algorithme.

Nos bonzaïs sont induits suivant le critère du pseudo-loss décrit dans (Schapire et Singer, 2000)

qui est recommandé pour construire les règles de décision simples dans l'algorithme original. Nous répétons alors de manière récursive l'opération pour développer l'arbre. La récursion est stoppée en fonction de deux critères :

1. la profondeur : le bonzaï ne peux plus se développer lorsque on atteint une profondeur fixée
2. le gain selon le critère de pseudo-loss : si ce gain n'est pas positif lors de la tentative de subdivision d'un nœud, ce nœud devient une feuille

Notre implémentation est disponible en ligne (Raymond, 2010).

3 Expériences comparatives

3.1 Corpus

Le projet EPAC (Estève *et al.*, 2010) concerne le traitement de données audio non structurées. L'objectif principal de ce projet a été de proposer des méthodes d'extraction d'information et de structuration de documents spécifiques aux données audio. Les données audio traitées durant le projet EPAC proviennent d'émissions radiophoniques enregistrées entre 2003 et 2004 au sein de trois radios françaises : France Info, France Culture et RFI. Au cours de ce projet, 100 heures de données audio ont été manuellement annotées, avec principalement des émissions contenant une forte proportion de parole spontanée (interviews, débats, talk shows...). Le corpus EPAC inclut des informations supplémentaires au niveau des locuteurs et des émissions transcrites. Plus précisément, le rôle, la fonction et la profession de chaque locuteur ont été manuellement annotés selon la disponibilité des informations. Ainsi, un même locuteur a un seul rôle général (par exemple *Invité*, *Interviewé*,...) mais qui peut être affiné avec un maximum de deux autres étiquettes (par exemple, pour un *Invité* : *politicien* / *premier ministre*). L'intégralité du corpus a été manuellement annotée en 10 rôles par un expert linguiste.

3.2 Évaluation

Tous les résultats sont présentés en terme de Taux d'erreur en rôle (TER) et ont été obtenus sur la totalité du corpus EPAC via une validation croisée sur 20 sous-parties du corpus. Le TER est la somme des rôles erronés divisée par le nombre de rôles dans la référence.

Plusieurs paramétrages de notre système sont testés en faisant varier :

1. la taille du *Ngramme* : utilisation soit d'*uni*-grammes *1g*, soit de *bi*-grammes *2g*
2. la profondeur du bonzaï utilisé dans l'algorithme de boosting : de 1 à 3. Notons que la profondeur 1 (prof.1) est équivalent à une règle de décision simple et donc à l'algorithme original utilisé dans les travaux antérieurs de la littérature sur le sujet. La profondeur 3 (prof.3) correspond quant à elle à un arbre à 8 feuilles maximum.
3. les descripteurs : $SPONTA$, SNA, $TEXT$ indépendamment les uns des autres ou en combinaison
4. les différents niveaux du descripteur $TEXT$: uniquement les mots W, ou alors à la fois les mots, les pos et la combinaison entité nommée/lemme de manière indépendante $W-P-LNE$ ou multi-niveaux $WPLNE$ (*cf.* 2.1.3)

Notre système paramétré avec la configuration *TEXT(2g W)+SPONTA+SNA prof.1* est équivalent au système publié dans (Dufour *et al.*, 2012). Ce système noté $REFERENT$ nous permettra par la suite de faire des comparaisons avec les différents résultats de notre système.

3.2.1 Analyse sur l'apport des *Ngrammes* multi-niveaux

La table 1 montre les résultats obtenus par notre système utilisant l'ensemble des descripteurs décrits en 2.1 et faisant varier la profondeur du bonzaï et la façon dont les *Ngrammes* sont extraits à partir de $TEXT$. Les résultats sont donnés pour 2000 itérations, valeur où chaque système semble avoir atteint sa performance maximale.

TER pour 2000 itérations		**prof. 1**	**prof. 2**	**prof. 3**
1g	*W*	27.1	23.8	22.3
	W-P-LNE	26.1	24.5	22.3
	WPLNE	26.2	23.4	22.9
2g	*W*	26.1*	22.2	22.3
	W-P-LNE	25.7	**21.7**	21.5
	WPLNE	**24.7**	22.8	**21.0**

TABLE 1 – Résultats des système appris avec tous les descripteurs (TEXT+SNA+SPONTA) en fonction de la profondeur de l'arbre et des types de *Ngrammes* $TEXT$ utilisés. Le système $REFERENT$ est noté avec une étoile.

Les résultats présentés dans le tableau 1 montrent qu'avec l'utilisation de règles simples ($prof.1$) les *Ngrammes* multi-niveaux ($WPLNE$) améliorent les performances en comparaison avec les *Ngrammes* extraits séparément ($W-P-LNE$). En effet, la figure 1 montre que les *Ngrammes* multi-niveaux obtiennent rapidement de bien meilleurs résultats que les *Ngrammes* $W-P-LNE$, même si dans le cas de cette tâche, les *Ngrammes* de mots seuls montrent de bons résultats.

Il est à noter que la configuration $W-P-LNE$ et $W-PLNE$ en *uni*-grammes sont équivalentes et ne peuvent se distinguer que pour $N > 1$. La différence de 0.1% entre ces configurations en *1g* s'explique par le fait que parmi les millions de descripteurs *Ngrammes* beaucoup d'entre eux ont le même pouvoir discriminant et l'algorithme de construction décide arbitrairement de conserver le premier de la liste. Ainsi l'ordre des *Ngrammes* étant présentés différemment selon la configuration, le classifieur obtenu est donc légèrement différent et cela explique cette instabilité mineure des performances obtenues.

3.2.2 Analyse sur l'apport des bonzaïs

L'utilisation des bonzaïs améliore très nettement la performance de l'algorithme de boosting, le tableau 1 montre bien que chaque niveau supplémentaire dans la profondeur des arbres améliore le système de manière très significative. La figure 2 montre que la profondeur nécessaire à ce gain est petite : un bonzaï de profondeur 2 (4 feuilles max) améliore grandement les résultats obtenus avec une règle simple (équivalent à un bonzaï prof.1, deux feuilles). Le bonzaï prof.3 améliore sensiblement, ce qui n'est plus le cas pour des profondeurs supérieures.

La figure 2 montre que l'augmentation de la profondeur de l'arbre permet de converger avec moins d'itérations vers le taux d'erreur minimum. Ceci est particulièrement intéressant sur les tâches faisant intervenir des millions d'indices textuels (*Ngrammes*). En effet l'algorithme d'induction d'arbre fait des évaluations exhaustives de chaque descripteur tandis que l'algorithme de boosting est itératif et non parallélisable. En effet, la figure 2 révèle que le système configuré avec des *2g* et une profondeur 2 fait jeu égal avec le système configuré avec des *1g* et une

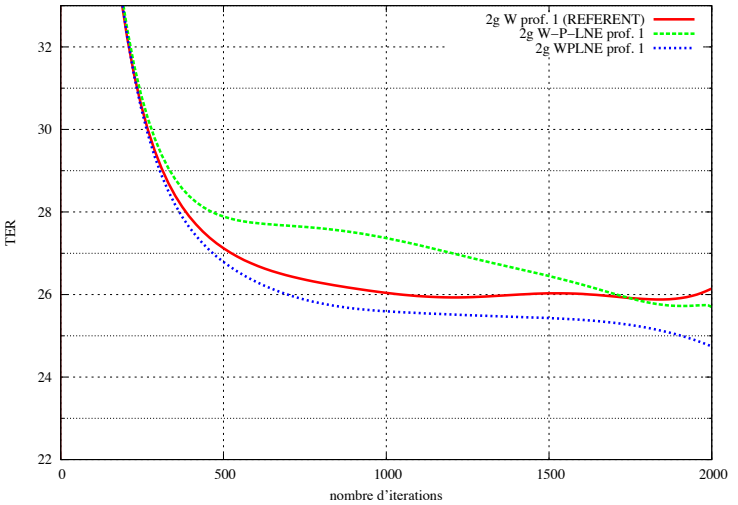

FIGURE 1 – Comparaison des perfomances des différents niveaux *Ngrammes TEXT* pour le système *TEXT(2g)+SNA+SPONTA*

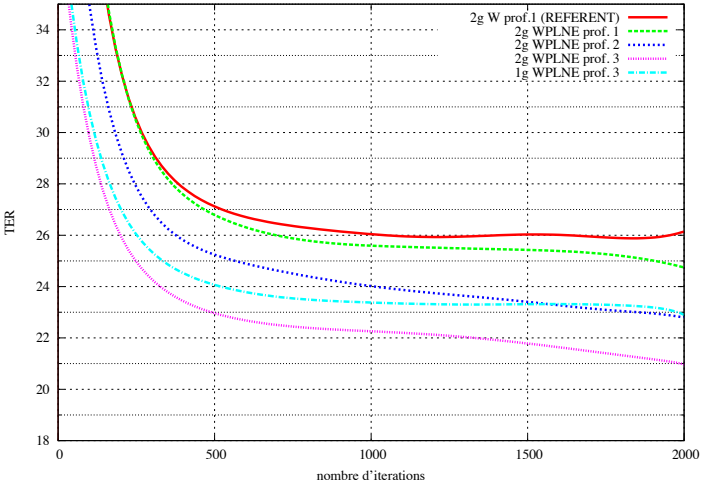

FIGURE 2 – Courbes de résultats pour différentes configurations de profondeur de bonzaïs et de descripteurs en fonction du nombre d'itération de boosting).

profondeur 3, la complexité générée par un arbre plus profond est largement compensée par l'économie faite par la réduction du nombre de descripteurs à évaluer : le gain de temps est d'un facteur 5. Ce même système comparé avec le système référent permet d'atteindre la même performance que celui ci avec 5 fois moins d'itérations ce qui lui permet un gain de temps d'un facteur 3.

Le tableau 2 présente la contribution de chaque catégorie d'indices isolée ou combinée avec les autres pour les deux systèmes : notre meilleure configuration *TEXT(2g WPLNE) prof.3* et la configuration référent (Dufour *et al.*, 2012) *TEXT(2g W) prof.1*. Il apparaît très clairement que l'utilisation des bonzaïs permet systématiquement d'améliorer les performances par rapport à l'utilisation de règles simples. Le système référent présente un TER de 26,1%, l'utilisation de bonzaïs permet avec exactement le même jeu de descripteurs de réduire le TER de 5,1% absolu.

Il est intéressant de noter que chacun des types de descripteurs apporte un gain. Notre meilleure configuration a choisi 65% de features de type $TEXT$, 27% de type $SPONTA$ et 8% de type *SNA*. Les bonzaïs permettent d'utiliser plusieurs types de features à chaque itération. Ainsi, quasiment 50% des itérations utilisent des features de types $TEXT$ et $SPONTA$ et près d'un tiers combinent les trois types de descripteurs.

indices/système	*proposé*	*référent*
TEXT	29.8	35.5
+ SPONTA	23.7	28.4
+ SNA	**21.0**	**26.1**
+ SNA	22.9	29.1
SPONTA	32.1	34.8
+ SNA	25.9	29.1
SNA	40.0	45.4

TABLE 2 – Résultats en terme de TER du système *proposé* et *référent* entraîné avec différentes combinaison d'indices

4 Conclusion

Nous avons proposé dans ce travail une modification de l'algorithme de boosting qui est très largement utilisé dans la tâche de la reconnaissance du rôle du locuteur. Nous proposons de remplacer les règles simples utilisées systématiquement dans la littérature en tant que classifieur faible dans l'algorithme de boosting par des mini arbres de décision appelés bonzaïs. Un système de reconnaissance de rôle à base de ce classifieur est évalué et comparé à un système état-de-l'art publié dans (Dufour *et al.*, 2012). Les expériences comparatives montrent qu'avec un jeu de descripteurs identique notre algorithme permet de mieux les exploiter en proposant une amélioration de 19,5% des performances tout en réduisant drastiquement le temps d'apprentissage du système.

Références

Barzilay, R., Collins, M., Hirschberg, J. et Whittaker, S. (2000). The rules behind roles : Identifying speaker role in radio broadcasts. *In AAAI*, pages 679–684.

Bigot, B., Ferrané, I., Pinquier, J. et André-Obrecht, R. (2010). Speaker role recognition to help spontaneous conversational speech detection. *In Searching Spontaneous Conversational Speech*, pages 5–10, Firenze, Italie.

Damnati, G. et Charlet, D. (2011). Robust speaker turn role labeling of tv broadcast news shows. *In ICASSP*, Prague, République Tchêque.

Deléglise, P., Estève, Y., Meignier, S. et Merlin, T. (2009). Improvements to the LIUM French ASR system based on CMU Sphinx : what helps to significantly reduce the word error rate ? *In Interspeech*, pages 2123–2126, Brighton, Angleterre.

Dufour, R., Estève, Y., Deléglise, P. et Béchet, F. (2009). Local and global models for spontaneous speech segment detection and characterization. *In ASRU*, Merano, Italie.

Dufour, R., Laurent, A. et Estève, Y. (2012). Combinaison d'approches pour la reconnaissance du rôle des locuteurs. *In JEP*, Grenoble, France.

Estève, Y., Bazillon, T., Antoine, J.-Y., Béchet, F. et Farinas, J. (2010). The EPAC corpus : manual and automatic annotations of conversational speech in French broadcast news. *In LREC*, pages 1686–1689, Valletta, Malte.

Garg, P. N., Favre, S., Salamin, H., Hakkani-Tür, D. et Vinciarelli, A. (2008). Role recognition for meeting participants : an approach based on lexical information and social network analysis. *In ACM Multimedia Conference (MM'08)*, pages 693–696, Vancouver, Canada.

Liu, Y. (2006). Initial study on automatic identification of speaker role in broadcast news speech. *In Human Language Technology Conference of the NAACL*, pages 81–84, New York, USA.

Raymond, C. (2010). bonzaiboost. http ://bonzaiboost.gforge.inria.fr/.

Raymond, C. et Fayolle, J. (2010). Reconnaissance robuste d'entités nommées sur de la parole transcrite automatiquement. *In Traitement Automatique des Langues Naturelles*, Montréal, Canada.

Salamin, H., Favre, S. et Vinciarelli, A. (2009). Automatic role recognition in multiparty recordings : Using social affiliation networks for feature extraction. *In IEEE Transactions on Multimedia*, volume 11, pages 1373–1380.

Schapire, R. E. et Singer, Y. (2000). BoosTexter : A boosting-based system for text categorization. *Machine Learning*, 39:135–168.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. *In Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Vinciarelli, A. (2007). Speakers role recognition in multiparty audio recordings using social network analysis and duration distribution modeling. *IEEE Transaction on Multimedia*, 9(6): 1215–1226.

Wang, W., Yaman, S., Precoda, K. et Richey, C. (2011). Automatic identification of speaker role and agreement/disagreement in broadcast conversation. *In ICASSP*, pages 5556–5559.