Construction et exploitation des réseaux de confusion dans le contexte d'une application de dialogue en langage naturel

Bogdan Minescu, Géraldine Damnati

Orange Labs - 2, av. Pierre Marzin, 22307 Lannion Cedex 07, France
{geraldine.damnati@orange-ftgroup.com,bogdan.minescu@gmail.com}

ABSTRACT

In the context of Spoken Language Understanding (SLU), post-processing rich Automatic Speech Recognition outputs such as word lattices rather than processing a single one-best solution has proven to be an efficient way of improving applicative performances but also to propagate uncertainty towards further applicative modules in order to delay the final decision. Confusion Networks consist in summarizing the information available in a word lattice into the form of a sequence of classes of local alternative hypotheses, while providing reliable confidence measures on these hypotheses. This study presents a strategy whose goal is to reject non-relevant messages as early as possible and to compute Confusion Networks only for relevant messages. On the basis of this strategy, an improved, SLU oriented, CN generation algorithm is also proposed that significantly reduces the size of the obtained CN while improving the recognition performances.

Keywords: Spoken Language Understanding, Confusion Networks, Spoken Dialogue Systems

1. Introduction

Des services téléphoniques permettant aux utilisateurs d'interagir en langage naturel avec des systèmes automatiques de dialogue sont désormais déployés auprès du grand public [1]. Les problèmes qui se posent à un tel système dépendent dans un premier temps de la reconnaissance automatique de parole (ASR) mais aussi des difficultés à modéliser les relations entre les concepts et la façon dont les utilisateurs les expriment. Dans ce type de système, le processus de compréhension passe par une succession d'étapes qui transforment, à l'aide de différents jeux de règles, la séquence de mots en entrée (la meilleure solution de l'ASR) en une interprétation. Celle-ci est ensuite fournie au gestionnaire du dialogue afin de construire une réponse adaptée. L'utilisation d'un espace de recherche plus grand en sortie d'ASR, notamment les graphes de mots, permet de retarder le processus de décision en maintenant actives plusieurs hypothèses. L'application sur ces hypothèses de post-traitements faisant appel à différentes sources d'information permet d'améliorer l'interprétation [4] [5].

Un CN est une séquence de "*classes*", contenant chacune une ou plusieurs hypothèses de mots en compétition. Chaque hypothèse est associée à sa probabilité *a posteriori*. La meilleure solution extraite du CN, appelée aussi *consensus hypothesis* (CH), est obtenue en choisissant dans chaque classe la transition ayant la meilleure probabilité *a posteriori*. Dans [3], un premier algorithme de génération des CNs a été proposé mais sa complexité élevée le rend difficilement utilisable dans des applications temps-réel. L'algorithme [2], que nous appelons *algorithme du pivot topologique*, se base sur une séquence de référence appelée *pivot*, puis regroupe itérativement les transitions du graphe de mots dans un ordre topologique. Dans un premier temps le *pivot* (plus court chemin du graphe dans cette étude) est segmenté en classes adjacentes. Les transitions sont ensuite insérées dans le réseau soit directement dans les classes existantes soit en créant de nouvelles classes. Ce processus est guidé par deux paramètres : le *recouvrement temporel* entre la transition et les classes candidates puis *l'ordre de précédence* de la transition avec les mots de la classe candidate (aucun des mots de la classe ne doit précéder sur un chemin du graphe la transition sinon une nouvelle classe est crée). A l'issue du processus, une transition nulle (*epsilon*) associée à une probabilité dite *d'omission* est ajoutée dans chaque classe afin que la somme avec les probabilités *a posteriori* des mots de la classe soit égale à un. Cet algorithme, de complexité réduite, sert de base aux développements et aux évolutions proposés dans cet article.

Comme nous l'avons montré dans [4], la recherche intégrée sur les graphes de mots et les réseaux de confusion (CN) améliore les performances du système moyennant cependant l'utilisation d'une stratégie adaptée. En effet, les messages à traiter dans le cadre d'applications réelles déployées sont de natures diverses. Les messages les plus bruités (bruits d'origine acoustique, parole hors-domaine...) ont pour effet de produire des graphes de taille importante et hautement ambigus. Dans ces conditions il est préférable de détecter au plus tôt les messages non-porteurs d'information et de privilégier les post-traitements sémantiques par exploration de graphes de mots pour les messages informatifs.

Nous montrons dans la présente étude que la génération de CN permet d'améliorer les performances du système en termes de taux d'erreurs interprétation (IER) moyennant une telle stratégie. Nous proposons par ailleurs un nouvel algorithme de génération des CN. Le cadre applicatif est présenté dans la section 2. Nous décrivons la stratégie proposée en 3 et le nouvel algorithme de génération des CNs en 4. Les résultats obtenus sont présentés en 5.

2. Cadre applicatif et caractérisation des énoncés

Cette étude est réalisée sur des données de l'application **3000** [1] et dans le contexte du projet européen LUNA [1] traitant notamment de l'amélioration de la robustesse des méthodes de compréhension automatique de la parole. Le **3000**, premier service vocal utilisant le language naturel déployé par France Telecom, permet aux utilisateurs d'obtenir des informations sur une trentaine de services différents liés à la ligne fixe, de souscrire à ces services et d'effectuer des opérations en ligne comme le paiement de la facture et plus généralement la gestion des services auxquels ils ont déjà souscrit. Le système de compréhension de la parole [1] procède en deux étapes successives. La première étape qui consiste à passer d'une séquence de mots à une séquence de concepts élémentaires utilise approximativement 1200 règles écrites de façon à avoir la meilleure couverture possible du domaine de l'application. Ces règles font correspondre une séquence d'un ou plusieurs mots à un concept, tout en sachant que le même concept peut être activé par des séquences de mots différentes. Le système utilise environ 400 concepts différents. Seule une partie des mots du lexique est utilisée pour la construction de ces règles. On distingue ainsi les mots **non-vides** qui entrent dans la composition d'au moins une règle ("*abonnement*", "*informations*") des mots **vides** ("*allo*" "*effectivement*"). Le lexique de l'application contient 2548 mots. 56% sont des mots non-vides, ce qui correspond à 65% des occurrences dans le corpus de test.

La deuxième étape consistant à transformer la séquence de concepts en une interprétation utilise 3200 règles triées par ordre de priorité et pouvant conduire à 2030 interprétations différentes. Une règle d'interprétation correspond à une combinaison logique de concepts. Une séquence de concepts donnée peut activer plusieurs règles d'interprétation, mais seule la règle ayant le meilleur rang dans la liste ordonnée sera choisie. Dans une application réelle, on doit prendre en considération toutes les détections de parole qui sont soumises au moteur de reconnaissance. Outre les détections non-parole insérées à tort par le module de Détection Bruit/Parole, le système doit également composer avec de la parole hors-domaine (HD), à savoir des commentaires (l'utilisateur se parle à lui même ou insulte le système : "*oh non, j'en ai marre de ce truc*") ou des apartés (l'utilisateur s'adresse à une tierce personne : "*va ranger ta chambre*"). Les messages restants sont considérés comme contenant de la parole Dans-le-Domaine.

L'ASR utilisé en première passe s'appuie sur un modèle de language de type bigram appris à partir d'un corpus de 44k énoncés. Il inclue un modèle de rejet dédié aux messages Non-Parole ainsi qu'un sous-modèle de langage dédié à la détection des commentaires [1]. Le corpus de test contient 3200 dialogues (6501 énoncés) collectés en conditions réelles.

Le tableau 1 donne la répartition des trois catégories d'énoncés dans le corpus de test. La deuxième colonne du tableau montre la taille des graphes de mots, exprimée en nombre moyen de transitions par graphe. Le moteur de reconnaissance à tendance à être sur-génératif pour $C1$ et $C2$. Non seulement la génération de ces graphes mais aussi les éventuels post-traitements sont coûteux en temps de calcul.

[1] www.ist-luna.eu

Tab. 1: Description du corpus de test par catégorie

Catégorie	# énoncés	# transitions par graphe
$C1$: Détections non-parole	1333	24000
$C2$: Parole Hors-Domaine	674	23000
$C3$: Parole Dans-le-Domaine	4494	14000
Total	6501	17000

3. Stratégie d'exploitation des CNs

3.1. Analyse du comportement de l'algorithme du pivot *topologique*

Le tableau 2 montre le détail de la contribution de chacune des trois catégories d'énoncés au taux d'erreur mot global (WER) pour la *1-best* et la *consensus hypothesis* (CH) extraite à partir de l'algorithme du *pivot* topologique. Est également indiquée la taille des CNs obtenus avec l'algorithme du *pivot* topologique (*baseline*) sur les trois catégories d'énoncés. Le nombre moyen de classes par mot de la référence représente la largeur du CN (ex : égale à 2 pour un énoncé de référence de 3 mots et un CN associé de 6 classes) et le nombre moyen de mots par classe représente la profondeur du CN. Les différences entre la taille des CNs sur les trois catégories sont directement liées aux graphes de mots, avec des CNs deux à trois fois plus grands pour $C1$ et $C2$ par rapport à $C3$. La largeur importante des CNs obtenus avec cet algorithme incombe en grande partie au nombre élevé de classes où la transition portant l'omission a la probabilité *a posteriori* la plus élevée. Nous proposons donc une étape d'élagage qui élimine du CN de telles classes lorsque le rapport entre la probabilité de l'omission et celle de la deuxième meilleure transition est supérieur à un seuil (optimisé empiriquement pour cette étude à une valeur de 10). Cette étape n'a aucune influence sur la *consensus hypothesis*. En revanche elle permet de réduire la largeur des CNs à 2.7 classes par mots de la référence. La profondeur moyenne s'en trouve augmentée à 13 mots par classes.

Tab. 2: Evaluation WER de la 1*best* et de la CH *baseline* issue de l'algorithme du *pivot* topologique

		WER	# classes par mots de ref.	# mots par classe
C1	1-best	6.8%	–	–
1333 én.	CH baseline	11.6%	65.6	6.3
C2	1-best	10.9%	–	–
674 én.	CH baseline	12.8%	51.8	7.0
C3	1-best	24.3%	–	–
4494 én.	CH baseline	26.9%	23.0	5.6
Total	1-best	42%	–	–
6501 én.	CH baseline	51.3%	34.8	5.9

Par ailleurs, si les performances sont globalement dégradées avec les CNs issus de l'algorithme du *pivot* topologique, la dégradation est plus significative sur $C1$ et $C2$ qui ne représentent que 30% des énoncés mais qui contribuent pour près de la moitié aux erreurs observées avec les CNs. Ceci illustre l'inadéquation des CNs pour rejeter des entrées non-valides. En effet, les graphes de taille importante et hautement ambigus génèrent à leur tour des CNs bruités pour lesquels la CH est source d'insertions et de substitutions. Ceci pourrait être compensé en utilisant des scores de confiance dérivés des probabilités *a posteriori* pour filtrer la CH. Néammoins, le souci d'efficacité lié à l'utilisation en temps réel pour une application de dialogue nous pousse à privilégier une stratégie de rejet performante dès la première passe de reconnaissance. Par ailleurs, à la section 4, nous proposerons un nouvel algorithme visant à améliorer les performances des CNs mais visant également à réduire leur taille.

3.2. Stratégie proposée

Comme mentionné à la section précédente, nous devons envisager un moyen efficace et rapide qui permette au système de rejeter les énoncés non-valides sans avoir besoin de générer les graphes de mots ni par conséquent les CNs correspondants. La *1-best* présente une précision de l'ordre de 95% pour la détection des énoncés à rejeter. Elle constitue ainsi un moyen rapide et peu coûteux en temps de calcul. En conséquence, nous avons décidé d'utiliser la *1-best* afin de construire une stratégie qui vise à rejeter les messages non-valides (appartenant à $C1$ et $C2$) et à ne générer les CNs que sur les énoncés contenant de la parole dans le domaine de l'application. Les CNs ne sont ainsi générés que si le modèle de première passe émet une hypothèse qui ne soit ni un rejet non-parole ni une détection de commentaire.

4. Algorithme de génération des CNs multi-niveaux

Comme nous l'avons montré dans [4], une analyse approfondie de l'algorithme du *pivot* topologique montre une influence négative de mots courts sur l'alignement et l'insertion des transitions dans le réseau. Ceci se traduit à la fois par une taille des CNs trop grande ainsi que par de moins bonnes performances en terme de WER. Ceci nous a conduit à proposer un algorithme du *pivot* modifié, à plusieurs niveaux, dans le but de construire des CNs plus petits, ayant de meilleures performances. L'algorithme du *pivot* initial analyse et traite les transitions dans un ordre topologique. Nous proposons une nouvelle technique d'insertion des transitions dans le réseau de confusion qui tient compte du mot porté par la transition.

Le nouvel algorithme de génération des CNs est constitué de quatre étapes. Les transitions sont analysées dans un ordre topologique mais ne sont insérées dans le CN que si elles respectent les critères pour chaque étape. Un traitement particulier est réservé aux mots non-vides et leur insertion est guidée par l'analyse en concepts de la séquence de mots constituant le *pivot*. L'objectif est de rendre plus fiable la construction dans le CN des hypothèses portant sur des mots significatifs pour l'application.

- **Étape 1 :** Les transitions portant un mot du *pivot* sont insérées dans les classes correspondantes.
- **Étape 2 :** Une transition portant un mot non-vide est insérée dans le CN si elle correspond au même concept qu'un des mots du *pivot*.
- **Étape 3 :** Toutes les transitions restantes portant des mots non-vides sont insérées dans le CN.
- **Étape 4 :** Les transitions restantes, qui ne portent que des mots vides, sont insérées dans le réseau de la même manière qu'à l'étape 3.

Pour chaque étape les conditions de recouvrement temporel et de précédence doivent être respectées. Seules l'étape 3 et l'étape 4 peuvent conduire à la création de nouvelles classes. L'étape 1 permet d'éviter la dispersion des hypothèses des mots du pivot sur plusieurs classes en largeur, et donc de rendre plus représentative leur probabilité *a posteriori*. L'étape 2 permet de favoriser le regroupement de mots portant le même concept et de fiabiliser ainsi les hypothèses conceptuelles. Le cas des règles faisant correspondre un concept à une séquence de plusieurs mots n'est pas traité et les mots se trouvant dans ce cas sont traités à l'étape suivante. La séparation des étapes 3 et 4 permet d'envisager d'interrompre la construction des CN dès l'étape 3, en effet l'ajout de mots vides supplémentaires (seul ceux présents dans le pivot étant conservés à l'étape 1) augmente la taille des réseaux alors que ces mots sont sans effet sur les traitements applicatifs en aval. De plus, on observe une sur-génération des mots vides dans les graphes de mots (55% des occurrences) par rapport à seulement 35% des occurrences dans le corpus de test. Enfin l'élagage *a posteriori* des classes pour lesquels l'omission est la plus probable est appliqué comme décrit à la section 3.1.

5. Résultats expérimentaux

Les performances du système en termes de WER ne garantissent pas en elles mêmes de bonnes performances du point de vue de l'utilisateur. Afin d'avoir une évaluation qui reflète au mieux ce point de vue nous évaluons le taux d'erreur d'interprétation (IER). L'interprétation générée par le module de compréhension est représenté par une composition de paires attribut-valeurs (ex : *Gest(Desactiver,TransfertAppel)*). Une interprétation est considérée comme étant correcte si tous les éléments qui la composent sont corrects. Les trois différents type d'erreurs sont les *Fausses Alarmes (FA)* quand une interprétation est trouvée pour un énoncé à rejeter, les *Substitutions (Sub)* et les *Faux Rejets (FR)*. L'*IER* est obtenu en sommant les différents types d'erreurs (*FA+Sub+FR*) et en divisant par le nombre d'interprétations de référence non-vides.

Le tableau 3 montre l'impact des modifications apportées sur l'algorithme de construction sur la taille des CNs générés. La dernière ligne du tableau représente la variante où l'algorithme s'arrête à l'étape 3. L'omission de la dernière étape permet de réduire notablement la largeur et la profondeur des CNs générés. Un premier jeu d'expériences vise à montrer les performances de la stratégie décrite dans la partie 3 en terme de WER. Suite à l'utilisation de la stratégie, 17% des énoncés ont été rejetés par la *1-best*. Le

Tab. 3: Taille des CNs générés

	# classes par mot de ref.	# mots par classe
pivot topologique	2.7	13
μ-niveaux	2.6	9.2
μ-niveaux sans vides	1.8	7.5

tableau 4 montre le détail sur les trois catégories du WER. En le comparant avec le tableau 2 on observe un gain de près de 10% sur Alg. du *pivot* topologique grâce à l'utilisation de la stratégie. Si le WER de la *1-best* est toujours légèrement meilleur que celui de l'algorithme topologique (la différence provenant essentiellement des messages appartenant à $C3$), ce n'est plus le cas pour l'algorithme multi-niveaux proposé. La variante consistant à écarter les mots vides de la phase de regroupement des classes permet d'améliorer encore plus les performances, avec une réduction absolue de 3.5% du WER par rapport à la *1-best*. En ne conservant que les mots vides présents initialement dans le pivot, on évite d'augmenter de façon trop importante le nombre d'insertions de ces mots vides. Ceci a d'autant plus d'effet sur les catégories C1 et C2 qui gérèrent les graphes les plus bruités

Tab. 4: Evaluation stratégie en terme de WER

WER Stratégie	Total	C1	C2	C3
pivot topologique	44.3%	6.6%	10.8%	26.9%
μ-niveaux	42.3%	6.3%	10.2%	25.8%
μ-niveaux sans vides	38.6%	5.3%	8.7%	24.6%

La probabilité *a posteriori* des mots dans le CN représente une mesure de confiance très efficace [2]. Un deuxième jeu d'expériences vise à montrer les performances de la stratégie en fonction d'un seuil appliqué sur cette mesure de confiance : les mots de la CH sont filtrés selon un seuil sur leur mesure de confiance et l'interprétation est calculée à partir de la séquence de mots filtrée. Dans la figure 1, les courbes ont été construites en variant le seuil sur la mesure de confiance. Le point le plus à droite sur chaque courbe correspond à un seuil égal à 0 (pas de filtrage). Le point le plus à droite dans la figure correspond à l'interprétation de la *1-best*. On peut ainsi observer une amélioration de l'IER en utilisant la stratégie avec l'algorithme du *pivot* topologique. Le point de fonctionnement est déplacé vers la gauche, avec une réduction importante de nombre de SUB et FA. Les deux variantes de l'algorithme multi-niveaux permettent une amélioration importante de l'IER avec un déplacement du point de fonctionnement vers la gauche et une augmentation moindre du taux de FR. Les performances équivalentes des deux variantes de l'algorithme multi-niveaux s'expliquent, d'un coté, par le fait que l'interpréteur n'utilise pas les mots vides et de l'autre par le fait que l'interpréteur est peu sensible aux insertions au niveau mots, qui contribuent principalement à la différence du WER entre les deux algorithmes. L'utilisation des mesures de confiance permet d'améliorer l'IER et d'obtenir un point de fonctionnement différent pour chaque seuil.

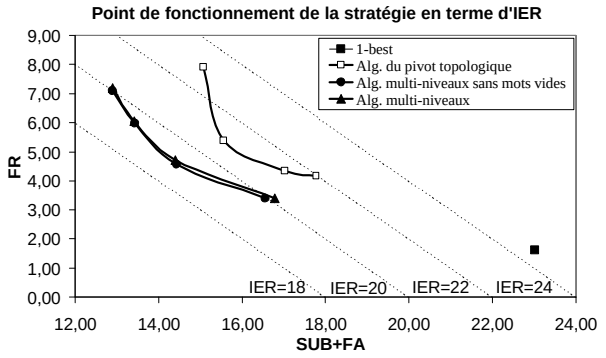

Fig. 1: *Le point de fonctionnement en fonction du seuil sur la mesure de confiance*

La différence entre les deux variantes de l'algorithme réside donc dans le choix des objectifs de l'application choisie. Ainsi, si on a besoin d'un espace de recherche plus grand avec de bonnes performances au niveau interprétation, on choisi l'*Alg. multi-niveaux*. En revanche, si on se trouve dans un contexte avec de fortes contraintes de temps de calcul et de taille des données, l'*Alg. multi-niveaux sans mots vides* peut constituer un meilleur choix.

6. Conclusions

Dans cette étude nous avons présenté une nouvelle stratégie pour l'exploitation des CNs dans le contexte d'un service de dialogue en language naturel. Nous avons également proposé deux variantes d'un algorithme de génération des CNs à multi-niveaux, basé sur l'algorithme du *pivot* topologique. En incluant l'utilisation des mesures de confiance que nous avons présentées, on obtient un éventail de méthodes qui permettent, en fonction du domaine applicatif et des objectifs choisis, d'optimiser différents aspects comme le temps de calcul, la taille des CNs, le taux d'erreur mots ou le taux d'erreur d'interprétation.

Références

[1] G. Damnati, F. Bechet, and R. de Mori. Spoken language understanding strategies on the france telecom 3000 voice agency corpus. In *Proc. ICASSP*, pages 9–12, 2007.

[2] D. Hakkani-Tur, F. Béchet, G. Riccardi, and G. Tur. Beyond asr 1-best : Using word confusion networks for spoken language understanding. *CSL*, 20(4) :495–514, 2006.

[3] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition : Word error minimization and other applications of confusion networks. *CSL*, 14(4) :373–400, 2000.

[4] B. Minescu, G. Damnati, F. Bechet, and R. De Mori. Conditional use of word lattices, confusion networks and 1-best string hypotheses in a sequential interpretation strategy. In *Proc. ICSLP*, 2007.

[5] Christophe Servan, Christian Raymond, Frederic Bechet, and Pascal Nocera. Conceptual decoding from word lattices : Application to the spoken dialogue corpus MEDIA. In *Proc. ICSLP*, pages 1614–1617, 2006.