Stratégie de décodage conceptuel pour les applications de dialogue oral

Christian Raymond †, Frédéric Béchet †, Renato De Mori † , Géraldine Damnati ‡

† LIA-CNRS, Université d'Avignon, BP1228, 84911 Avignon Cedex 9, France
‡ France Télécom R&D - DIH/IPS/RVA 2 av. Pierre Marzin 22307 Lannion Cedex 07, France

{christian.raymond, frederic.bechet, renato.demori}@lia.univ-avignon.fr
geraldine.damnati@rd.francetelecom.com

ABSTRACT

The approach proposed in this paper is an alternative to the traditional sequential architecture of Spoken Dialogue Systems where transcribing and understanding a speech signal are two separate processes. By representing all the conceptual structures handled by the Dialogue Manager by Finite State Machines and by building a conceptual model that contains all the possible interpretations at a given dialogue state, we propose a decoding architecture that search first for the best conceptual interpretations before looking for the best strings of words. The output of this process is a structured n-best list of hypotheses, at the concept and word levels. Several confidence measures are then used in order to rescore and select a candidate from this list. This paper reports significant understanding error rate reduction on a tourist inquiry application developped by France Telecom R&D.

1. INTRODUCTION

Les systèmes de dialogue utilisent généralement une approche séquentielle dans la procédure menant à l'interprétation d'un signal de parole. Dans un premier temps un système de reconnaissance via des modèles acoustiques et linguistiques génère, à partir du signal, une hypothèse ou une liste d'hypothèses. L'hypothèse choisie sera ensuite soumise à un interpréteur sémantique qui en donnera une représentation conceptuelle. Cette approche séquentielle est limitée sur deux points : d'une part, si l'on considère une liste des meilleures hypothèses, il faut très souvent qu'elle soit de taille très importante pour avoir l'intégralité des interprétations possibles car beaucoup d'entre elles ne diffèrent que d'un ou quelques mots qui ne modifient pas leur interprétation sémantique (articles, préposition, etc) ; d'autre part, alors que dans un système de dialogue nous avons une connaissance exhaustive de la sémantique utilisée, elle n'est pas prise en compte durant la phase de reconnaissance, il serait alors pertinent de l'utiliser dans la recherche de la meilleure hypothèse, plutôt que de s'en remettre aux seuls critères acoustiques et linguistiques. Dans ce papier nous présentons une méthode permettant d'intégrer la sémantique dans le modèle de reconnaissance, pour pouvoir guider conceptuellement le décodage et obtenir directement la séquence de concepts associée à une hypothèse. Ceci nous permet de fournir une liste de N-meilleures hypothèses structurée conceptuellement. Exploitant cette liste, une stratégie apprise automatiquement au moyen de l'utilisation d'un arbre de décision sémantique est présentée. Cette stratégie s'appuie sur un ensemble de mesures de confiance pour faire un choix parmi les hypothèses de cette liste et permet de baisser significativement le taux d'erreur en compréhension.

2. MODÈLE CONCEPTUEL

L'intégration de modèles sémantiques dans des modèles de langage statistiques n'est pas une idée nouvelle [1, 2, 4]. La plupart de ces travaux utilisent ces modèles, *a posteriori*, pour réordonner une liste d'hypothèses ou pour étiqueter sémantiquement la meilleure hypothèse d'un système de reconnaissance (ASR). Dans notre approche, les connaissances sémantiques du système de dialogue sont intégrées dans le modèle statistique de reconnaissance de la parole. Des mesures de confiance sont ensuite utilisées pour vérifier l'intégrité des hypothèses obtenues. Le rôle du modèle conceptuel est de détecter l'ensemble des interprétations conceptuelles pour une intervention de l'utilisateur. Une interprétation est une séquence de concepts s_j qui sont liés au dialogue ou au domaine de l'application. Un concept s_j est représenté par un couple $<c_j, vj>$ où c_j est l'étiquette conceptuelle et v_j est sa valeur.

2.1. Modèle statistique

La contribution d'une séquence de mots W à une structure conceptuelle est évaluée par la probabilité *a posteriori* $P(\mid Y)$, où Y est la description des paramètres acoustiques. Cette probabilité est calculée comme suit :

$$P(\mid Y) = \frac{\sum_{W \in SW} P(Y \mid W) P(\mid W)^{\delta} P(W)^{\lambda}}{\sum_{W \in SW} P(Y \mid W) P(W)^{\lambda}} \quad (1)$$

où $P(Y \mid W)$ est donnée par le modèle acoustique, $P(W)$ est calculée par le modèle de langage. Les exposants δ et λ sont respectivement des coefficient de mélange (fudge) sémantique et syntaxique. SW correspond à l'ensemble des chaînes de mots qui peuvent être trouvées dans le graphe de mots. $P(\mid W)$ peut être calculée de plusieurs manières, mais dans l'étude préliminaire de ce papier, la probabilité $P(\mid BW)$ est simplement fixée à 0 pour une structure conceptuelle qui ne peut pas être inférée depuis W et à 1 si la structure conceptuelle peut être inférée sans ambiguïté à partir de W.

2.2. Entités conceptuelles

Dans cette étude, les entités conceptuelles sont les connaissances sémantiques du gestionnaire de dialogue. Elles sont en relation avec la gestion du dialogue (confirmation, contestation, ...) ou avec le domaine de l'appli-

cation (lieu, date, ...).
La section 5 présente des résultats obtenus sur une application de recherche de restaurants à Paris et dont les concepts les plus fréquents sont : *LIEU*, *PRIX* et *SPÉCIALITÉ* :
– *LIEU* : *près de Bastille* ;
– *PRIX* : *pour maximum vingt euros* ;
– *SPÉCIALITÉ* : *spécialité indienne*.
Une interprétation pour une intervention de l'utilisateur est la séquence de concepts qui peut être extraite d'une phrase. Par exemple, en utilisant ces unités conceptuelles, la transcription suivante : *Je recherche un restaurant italien près de Bastille pour maximum vingt euros* correspond à l'interprétation : <SPÉCIALITÉ,*italien*> <LIEU,*Bastille*> <PRIX,*vingt euros*>. Ces entités sont exprimées dans le corpus d'apprentissage par de courtes séquences de mots contenant trois type de mots : les Mots-Clefs comme *Bastille*, les mots spécifiques à un concept comme *spécialité* et des modifieurs explicitant comment interpréter le ou les Mots-Clefs comme *maximum*. Dans le but d'apprendre automatiquement des grammaires régulières spécifiques à chaque concept, le corpus d'apprentissage est étiqueté manuellement au niveau conceptuel. Les grammaires régulières sont générées à partir de ces exemples et généralisées par l'utilisation de critères syntaxiques et sémantiques.

2.3. *Transducteur Mots-Concepts*

Chaque concept C_k de l'application de dialogue est associé à une grammaire régulière. Ces grammaires sont représentées sous la forme d'Automates à États Finis appelés *accepteurs* (A_k pour le concept C_k). Dans le but de gérer les séquences de mots qui ne représentent aucun concept et que nous appellerons texte *background*, un modèle "Mange-Mots", appelé A_M est implémenté. Comme une même chaîne de mots ne peut pas être à la fois vue comme un concept et un background, tous les chemins des accepteurs A_k sont soustrait du modèle "Mange-Mots" A_F de la manière suivante :

$$A_F = \Sigma * \quad \bigcup_{k=1}^{m} A_k$$

où Σ est le lexique de l'application et m est le nombre de concepts utilisés. Tous ces accepteurs sont transformés en transducteurs qui prennent comme entrée des mots et comme sortie des balises de *début* et *fin* de concept. Tous les accepteurs A_k deviennent alors des transducteurs T_k où la première transition émet le symbole $<C_k>$ et la dernière le symbole $</C_k>$. Le modèle "Mange-Mots" devient le transducteur T_{bk} qui émet les symboles $<BCK>$ et $</BCK>$. Aucun autre symbole n'est émis : tous les mots des transducteurs émettent un symbole *epsilon*. Enfin tous ces transducteurs sont intégrés dans un seul modèle appelé $T_{concept}$ illustré dans la figure 1. Ce transducteur Mots-Concepts $T_{Concept}$ peut associer plusieurs interprétations (*chemins observés sur les sorties*) pour une même séquence de mots (*chemin observé sur les entrées*) mais une seule segmentation d'une chaîne de mots est possible pour une séquence donnée de concepts.

3. ARCHITECTURE GÉNÉRALE

Pour l'implémentation des automates présentés dans la section 2.3 nous avons utilisé la librairie d'AT&T [6] qui fournit des algorithmes pratiques et performants pour manipuler les accepteurs ainsi que les transducteurs.

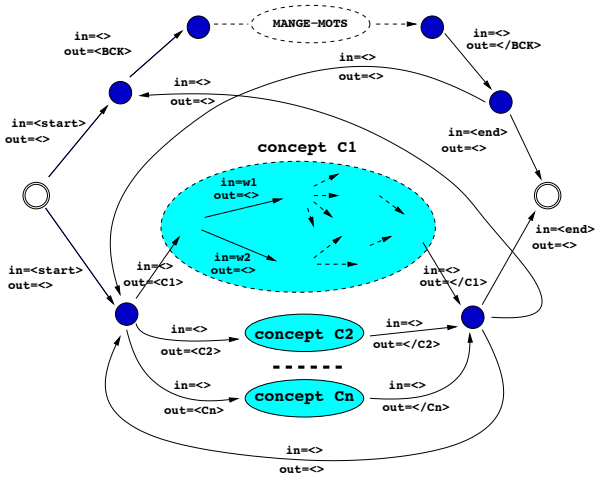

FIG. 1: Transducteur Mots-Concepts

Nous allons maintenant présenter les différentes étapes du processus qui utilise le modèle conceptuel dans le but de générer une liste réduite d'hypothèses qui sont sémantiquement différentes et validées par différentes mesures de confiance.

3.1. *Graphe de mots enrichi conceptuellement*

Un premier décodage est effectué en utilisant des modèles statistiques acoustiques et linguistiques. Ceci génère un graphe de mots qui est transformé en un transducteur stochastique T_{ASR}. Les poids associés aux transitions dans ce transducteur sont la combinaison des scores acoustiques et linguistiques. Les entrées et sorties du transducteur (qui sont les mots du lexique) sont identiques. En appliquant une opération de *composition* entre T_{ASR} et le modèle conceptuel $T_{Concept}$ nous obtenons un nouveau transducteur T_{Decod} qui intègre, dans le graphe de mots de l'ASR, les connaissances conceptuelles :

$$T_{Decod} = T_{ASR} \circ T_{Concept}$$

Dans T_{Decod} les entrées sont les mots et les sorties sont les étiquettes conceptuelles. Un chemin dans T_{Decod} correspond à une chaîne de mots si seuls les symboles d'entrée sont considérés ; de la même manière en considérant seulement les symboles de sortie, ce chemin correspond à une séquence d'étiquettes conceptuelles.

3.2. *N-meilleures interprétations conceptuelles*

Le transducteur T_{Decod} est converti en un accepteur par une projection sur ses sorties. Cette projection crée un accepteur dont les entrées sont les étiquettes conceptuelles et les chemins dans cet automate représentent toutes les séquences conceptuelles qui peuvent être émises par T_{Decod}. En y appliquant un algorithme de recherche des meilleurs chemins, nous obtenons la liste des meilleures interprétations, qui existent dans le graphe de mots. En pratique, toutes les interprétations qui existent dans T_{Decod} peuvent être exprimées avec une liste relativement petite[1]. Il est à noter que le score associé à une séquence de concepts est la somme des probabilités de toutes les séquences de mots de T_{Decod} qui engendre cette séquence de concepts conformément à l'équation 1.

[1] cela dépend du nombre de concepts utilisé par l'application et de la taille du graphe de mots généré durant la première passe de décodage

3.3. *N-meilleures hypothèses*

Chaque séquence de concepts *i* obtenue précédemment est transformée en un transducteur T_{inter_i} avec des entrées et sorties identiques : les symboles *début* et *fin* de chaque concept. Avec une opération de *composition* entre le transducteur T_{Decod} et le transducteur T_{inter_i}, nous obtenons un transducteur qui représente un sous-graphe de mots contenant toutes et seulement les hypothèses qui génèrent la séquence de concepts correspondant à l'interprétation *i*. Une liste des meilleures hypothèses pour chaque interprétation *i* est obtenue en appliquant un algorithme de recherche sur le transducteur $T_{Decod} \circ T_{inter_i}$.

3.4. *Mesures de Confiance*

À chaque hypothèse de la liste sont associées différentes mesures de confiance. Une mesure linguistique LC qui calcule la proportion de replis effectués par le modèle de langage durant le décodage. Son calcul est très rapide et les scores de confiance obtenus donnent des résultats intéressants présentés dans [3]. Une mesure de confiance acoustique AC, détaillée dans [7], calculée sur toute la phrase ou sur des portions de signal repérées comme étant des concepts. Une mesure sémantique SC, détaillée dans [7], qui estime les non-détections ou les détections à tort de concept. Cette mesure utilise des arbres de classification sémantique introduit pour la tache ATIS par [5]. Aux précédentes mesures de confiance est ajouté le rang de chaque hypothèse dans la liste. Dans la liste standard c'est simplement sa position. Si c'est une liste structurée, le rang comporte deux nombres, le rang de l'interprétation et la position de l'hypothèse parmi les hypothèses de cette interprétation.

4. STRATÉGIE PAR ARBRE DE DÉCISION

4.1. *Utilisation des mesures de confiance*

La première étape pour la mise en œuvre de notre stratégie de choix est de construire un corpus d'apprentissage. Ce corpus est composé de transcriptions automatiques de notre corpus de développement. À ces transcriptions sont associées les concepts et leurs valeurs détectés par notre modèle sémantique, ainsi que toutes les mesures de confiance précédemment définies. Le principal avantage d'une stratégie par arbre de décision est qu'il n'est pas nécessaire d'avoir une connaissance *a priori* sur l'efficacité des critères choisis : c'est l'arbre de décision lui même qui choisira les plus pertinents. Les scores de confiance étant des valeurs numériques, nous les avons convertis en représentation discrète. Nous avons choisi 3 étiquettes pour chaque critère AC, LC and SC : `H`, `N` et `F` respectivement pour une confiance Haute, Neutre et Faible. Pour chaque exemple du corpus d'apprentissage nous avons les éléments suivants : AC_{global}, la confiance acoustique sur toute la phrase ; AC_H, AC_N et AC_F, le nombre de concepts détectés étiquetés avec respectivement une confiance haute, neutre et faible. De la même façon, nous avons : LC_{global}, SC_H, SC_N, SC_F et le rang R. Il est à noter que les intervalles de valeurs associés aux étiquettes de confiance ont été obtenus avec un corpus de développement et basés sur la différence du pourcentage d'acceptations correctes *vs*. le pourcentage de fausses acceptations considérant la mesure en question.

4.2. *Sélection de la meilleure hypothèse*

Pour entraîner l'arbre de décision nous avons associé une étiquette à chaque exemple dans le corpus d'apprentissage. Deux étiquettes sont définies : `ALL_OK` et `NOT_OK`. La première est associée aux exemples dont les concepts et leur valeur sont corrects. Le second aux exemples contenant des concepts/valeurs incorrectes. L'arbre est alors entraîné pour minimiser l'impureté, au moyen du critère de Gini, de la distribution des étiquettes `ALL_OK` et `NOT_OK` dans l'ensemble d'apprentissage. Le processus s'arrête lorsque il n'y a plus de gain en impureté ou lorsque le nombre d'exemple attaché à une feuille de l'arbre est inférieur à un seuil fixé. Les questions de l'arbre posées à chaque nœud sont extraites de la liste des étiquettes de confiance. À la fin du processus d'entraînement, le score attaché à chaque feuille de l'arbre est la proportion entre le nombre d'exemples `ALL_OK` et le nombre total d'exemples dans la feuille. Ce score représente la confiance donnée par la classification qu'un exemple ne contienne que des concepts/valeurs corrects. Une fois l'arbre construit, le processus de choix d'une hypothèse dans la liste *L* des N-meilleures est le suivant :

- premièrement, tous les scores de confiance pour chaque hypothèse dans *L* sont calculés ;
- les étiquettes correspondant aux différents niveaux de confiance leur sont associées ;
- l'arbre est parcouru par chaque hypothèse qui reçoit le score associé à la feuille terminale ;
- l'hypothèse choisie dans *L* est la première qui possède un score d'être `ALL_OK` supérieur à un seuil donné ;
- finalement, si aucune hypothèse n'a un score au dessus de ce seuil, l'hypothèse ayant le score le plus élevé est choisie (dans le cadre d'une stratégie sans rejet), sinon elle est rejetée (dans le cadre d'un stratégie avec rejet).

5. ÉVALUATION DE LA STRATÉGIE

5.1. *Données expérimentales*

Les expériences ont été menées sur un corpus de dialogue fourni par France Telecom R&D. Le vocabulaire est de 2200 mots. L'application de dialogue considérée est une recherche de restaurant à Paris via le téléphone. Le corpus a été coupé en deux : une partie de développement de 511 interventions utilisateur et une partie de test de 419 interventions. Le développement a été utilisé pour déterminer les différents seuils pour les mesures de confiance ainsi que pour l'entraînement de l'arbre de décision stratégique. Le taux d'erreur mots sur le corpus de test est de 22.7%. Les expériences ont été menées sur les interventions du corpus de test contenant au moins un concept.

5.2. *Critères d'évaluation*

La mesure considérée est le taux d'erreur en compréhension (*UER*). Le *UER* est associé au couple `<étiquette conceptuelle,valeur>`. Les valeurs sont obtenues par un ensemble de règles qui transforme la séquence de mots détectées comme concept en valeurs significatives. Par exemple la transcription : *un restaurant à Bastille* est associée à `<LIEU,BASTILLE>`. Le corpus de référence est alors construit en filtrant les transcriptions pour ne garder que les étiquettes conceptuelles et leur valeur pour chaque intervention.

Avec en tant que structure conceptuelle de chaque hypothèse du corpus de test, l'*UER* est défini comme suit :

$$UER = \frac{S_v + D_c + I_c}{T} \times 100$$

où S_v la substitution d'une valeur d'un concept dans , D_c indique la suppression d'un attribut I_c indique une insertion. T est le nombre total de concepts dans la référence.

5.3. Évaluation

Les courbes de la figure 2 présentent les valeurs du UER en fonction du pourcentage de phrases rejetées sur le corpus de développement et de test. On peut y observer que notre stratégie arrive bien à détecter les cas où la reconnaissance n'a pas été satisfaisante : avec un taux de rejet de seulement 8% nous obtenons une réduction relative de 43.1% du UER sur le test. Il est à noter que l'arbre de décision tire avantage de la liste structurée pour améliorer le découpage entre les exemples `ALL_OK` et `NOT_OK`. Le tableau 1 montre les résultats obtenus avec la stratégie sans rejet sur le corpus de développement et de test. Deux conditions sont examinées : le choix dans une liste standard des N-meilleures hypothèses et dans une liste structurée sémantiquement. La taille des listes a été fixée à 12 candidats : les 12 premiers pour la liste standard et les 4 premiers candidats des 3 premières interprétations pour la liste structurée. Dans les deux cas, le gain obtenu est très significatif. Le gain ne peut être comparé avec celui obtenu sur le taux d'erreur mots : il chute de 21.6% à 20.7% sur le développement et de 22.7% à 22.5% sur le test. Il apparaît clairement que le taux d'erreur mots n'est pas une mesure pertinente dans un contexte de dialogue oral : en effet une forte réduction du UER n'a que peu d'effet sur celui-ci.

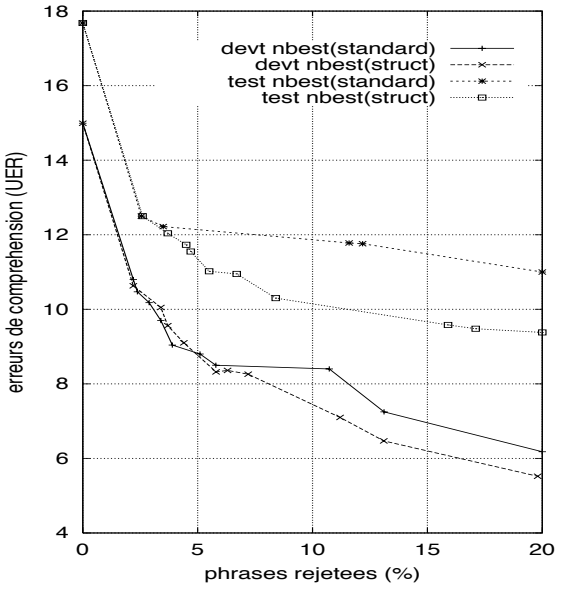

FIG. 2: UER en fonction du pourcentage des phrases rejetées

6. CONCLUSION

L'approche proposée est une alternative à la traditionnelle architecture séquentielle des systèmes de dialogue où les connaissances sémantiques de l'application ne sont pas prises en compte pendant le processus de décodage. En représentant toutes les structures conceptuelles utilisées par le gestionnaire de dialogue par des Automates à États Finis et en implémentant un modèle sémantique qui contient toutes les interprétations possibles, nous proposons une architecture de décodage qui cherche en premier lieu les meilleures interprétations avant de chercher les meilleures chaînes de mots. Une nouvelle méthode pour construire une liste de *N*-meilleurs candidats est proposée. À l'aide d'une série de mesures de confiance associé à ces candidats, nous proposons une stratégie automatique de sélection par arbre de décision qui permet de faire baisser significativement le taux d'erreur en compréhension.

TAB. 1: Taux d'erreurs de compréhension avec et sans utilisation de la stratégie sur une liste des N-meilleures standard et structurée (N=12 et stratégie sans rejet)

Liste des N-meilleures standard			
Corpus	baseline	rescoring	UER reduction %
Devt.	15.0	13.0	13.3%
Test	17.7	14.7	16.9%
Liste structurée des N-meilleures			
Corpus	baseline	rescoring	UER reduction %
Devt.	15.0	12.4	17.3%
Test	17.7	14.5	18%

RÉFÉRENCES

[1] Helene Bonneau-Maynard and Fabrice Lefevre. Investigating stochastic speech understanding. In *Proceedings of Automatic Speech Recognition Understanding Workshop*, Trento, Italy, 2001.

[2] Hakan Erdogan, Ruhi Sarikaya, Yuqing Gao, and Michael Picheny. Semantic structured language models. In *Proceedings of International Conference on Spoken Language Processing*, Denver, USA, 2002.

[3] Yannick Estève, Christian Raymond, Renato De Mori, and David Janiszek. On the use of linguistic consistency in systems for human-computer dialogs. *IEEE Transactions on Speech and Audio Processing*, 11 :746–756, Novembre 2003.

[4] Kadri Hacioglu and Wayne Ward. Dialog-dependent language modeling combining n-grams and stochastic context free grammars. In *Proceedings of International Conference on Acoustics Speech and Signal Processing*, Salt Lake City, USA, 2001.

[5] Roland Kuhn and Renato De Mori. The application of semantic classification trees to natural language understanding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(449-460), 1995.

[6] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer, Speech and Language*, 16(1) :69–88, 2002.

[7] Christian Raymond, Frédéric Béchet, Renato De Mori, Géraldine Damnati, and Yannick Estève. Automatic learning of interpretation strategies for spoken dialogue systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Montreal, Canada, 2004.