

Graph2Bots: Assistance automatisée à la conception d'agents dialoguants

Jean-Leon Bouraoui*, Vincent Lemaire*

*2 Avenue Pierre Marzin, 22300 Lannion
{jeanleon.bouraoui,vincent.lemaire}@orange.com,
<http://vincentlemaire-labs.fr/>

Résumé. Nous décrivons la démonstration d'un prototype permettant la modélisation non supervisée de la structure de dialogues finalisés; ces dialogues appartiennent à un domaine donné (par exemple réservation de trains). Ceci présente de nombreux intérêts, le principal étant de servir de base à la conception de l'architecture d'un agent dialoguant. Un graphe modélise les principales étapes des dialogues et les transitions entre elles. La technique adoptée consiste à appliquer du CoClustering sur le corpus cible de dialogues, afin d'obtenir les principaux thèmes qui y figurent. On calcule ensuite les transitions entre thèmes dans chaque dialogue. Notre outil permet d'obtenir le graphe correspondant et de le manipuler de manière ergonomique. Nous présentons en détail les différentes fonctionnalités démontrées.

1 Introduction

En intelligence artificielle, les agents dialoguants connaissent un gain de popularité auprès du grand public; et ce d'autant plus qu'ils bénéficient des avancées dans la compréhension des contenus et du contexte. Cela est le fait notamment d'applications mobiles telles que Siri (Apple), Google Now (Google), ou Cortana (Microsoft) ou Alexa (Amazon). Cet intérêt grandissant pour la technologie des interfaces dialoguantes et des agents dialoguants en particulier est décrit par de nombreuses études telles que celle du cabinet d'analyse Gartner¹.

Une des tendances actuelles est de proposer des dispositifs logiciels de conception d'agents dialoguants, personnalisables selon les besoins et le domaine d'application (par exemple, réservation de voyages, commande de produits ou de services, etc.). L'un des enjeux de ces dispositifs est de pouvoir être mis en place rapidement, sachant qu'il n'existe actuellement pas de système générique, et qu'une adaptation de l'agent à un domaine d'application donné prend du temps.

Dans ce contexte nous présentons une solution d'assistance semi-automatique à la création ou l'adaptation d'un agent dialoguant pour un domaine applicatif donné. Dans un premier temps, nous décrivons la problématique abordée. Nous présentons ensuite notre solution, d'abord du côté backend, puis de l'interface utilisateur correspondante (frontend), telles qu'elles seront présentées à l'occasion de la démonstration.

1. <http://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017>

2 Description de la problématique

Dans le cadre de notre démonstration, nous appellerons dialogue un échange d’informations entre deux interlocuteurs (sachant qu’un dialogue peut faire intervenir plus de deux interlocuteurs). Un interlocuteur peut être un humain ou une machine (au sens large : un système artificiel, logiciel ou matériel). Nous nous intéressons aux dialogues finalisés, qui cherchent à atteindre un but : les interlocuteurs vont collaborer pour l’atteinte de ce but.

On appelle dans cet article “corpus textuel” un ensemble de n dialogues relatifs à un domaine particulier, (par exemple transcriptions de dialogues de réservations de trains, ou chats d’interactions entre un téléconseiller et un client). Chaque dialogue est composé de t tours de parole, un tour de parole correspondant à ce que dit l’un des interlocuteurs sans interruption (la plupart du temps, une ou plusieurs phrases).

Notre but est de déterminer automatiquement, et ce au sein de chaque dialogue : (i) les différentes phases du dialogue (incluant notamment les intentions exprimées ; nous les désignerons désormais par le terme “thème”, qui correspond aussi bien à des thématiques génériques qu’à des sous-buts du dialogue) ; (ii) les transitions entre les phases. Le but est d’obtenir une représentation du déroulement typique des dialogues du corpus. La représentation souhaitée est celle d’un graphe orienté montrant les principales transitions entre thèmes, comme celui représenté (et simplifié) sur la figure 1. Notre postulat est que, selon la position dans le dialogue, un tour de parole donné présente plus de probabilité d’appartenir à une phase donnée (i.e. un cluster), qu’un autre ; cette information est donc prise en compte lors du processus.

Le graphe ainsi obtenu présente de multiples intérêts. Le principal est l’initialisation de la conception de l’agent dialoguant : il pourra servir de base à la modélisation de l’architecture d’un agent dialoguant spécialisé sur le domaine cible, et ainsi en faciliter l’exécution. A l’heure actuelle, cette tâche est la plupart du temps effectuée manuellement : soit a priori, à partir de la représentation que le concepteur se fait des dialogues possibles portant sur une tâche et un domaine donné ; soit a posteriori, à partir de la consultation de corpus existants ; dans les deux cas, le processus est coûteux en temps.

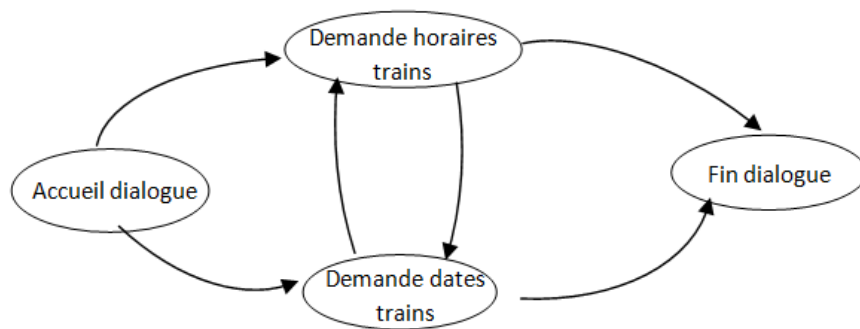


FIG. 1 – Représentation graphique des principales transitions entre thèmes

De plus, le graphe, ainsi que les étapes parcourues pour l’obtenir, permettra au concepteur, sans connaissance préalable du domaine d’application, d’avoir une première compréhension du contenu thématique des dialogues, de leur structuration, et plus généralement la connaissance des informations les plus pertinentes pour la réalisation de l’agent dialoguant.

3 Description de la démonstration

Un exemple prototypique d’utilisation de notre outil est le suivant : un concepteur souhaite mettre en place un agent dialoguant relatif pour un domaine d’application spécifique. Il dispose de corpus de dialogues relatifs à ce domaine. Dans un premier temps, il va utiliser le backend de notre outil pour identifier, de manière non supervisée et sans annotation préalable, les phases de dialogues sous-jacentes, et leurs transitions. Il va ensuite manipuler les graphes obtenus à l’aide de l’interface visuelle fournie. Nous décrivons ces différentes étapes ci-dessous.

3.1 Approche pour le backend

Au préalable, les mots du corpus sont filtrés pour supprimer les mots dits “outils” ou “creux”, a priori inutiles pour le traitement. Le filtre utilisé est la liste de “stopwords” français utilisés pour le stemming par la librairie NLTK². A moyen terme, nous envisageons d’autres prétraitements des mots du corpus, notamment leur lemmatisation.

Pour identifier les phases de dialogue, on utilise une technique de CoClustering qui permet d’obtenir une “copartition” de la matrice mots x tour de paroles. Étant données, deux (ou plus) variables catégorielles ou numériques, on réalise un partitionnement simultané des variables : les valeurs de variables catégorielles sont groupées en clusters et les variables numériques sont partitionnées en intervalles – ce qui revient à un problème de coclustering. La méthode employée est basée sur l’approche MODL décrite dans (Boullé et al., 2014). Il reste ensuite à déterminer les transitions entre les phases de dialogue.

Dans notre approche, une phase correspond à un cluster de tours de parole, homogènes par rapport à une thématique donnée. Une phase définie ainsi peut être reliée à une ou plusieurs autres, en fonction de la fréquence observée de leurs successions dans les dialogues du corpus.

Le CoClustering fait initialement perdre la séquentialité des dialogues, puisque les tours de paroles sont regroupés par thèmes, indépendamment de leur ordre dans le dialogue. Pour retrouver cet aspect temporel, le backend projette ensuite les identifiants de cluster sur chaque tour de parole correspondant. La représentation obtenue est un graphe orienté, dont les nœuds sont les clusters, et les arcs sont les successions entre clusters.

2. http://www.nltk.org/nltk_data

3.2 Présentation du frontend

L'interface utilisateur permet de visualiser interactivement et en temps réel les données traitées en backend sous la forme de graphes de dialogue. Nous en décrivons les principales caractéristiques ci-dessous.

3.2.1 Affichage interactif et en temps réel

Le concepteur de l'agent dialoguant peut observer et analyser le graphe obtenu. Les principales fonctionnalités appartenant à cette catégorie sont les suivantes :

- Choix de la granularité d'affichage du graphe, selon le nombre de clusters et/ou de leur relations, et modification dynamique du graphe correspondant ;
- Possibilité de manipuler les graphes avec le pointeur de la souris : par exemple pouvoir « tirer » un cluster à l'écart des autres, sélectionner un ou plusieurs clusters, zoomer et dézoomer, etc.
- Affichage des noms de clusters ³, de leur nombre, et de la fréquence de chaque relation (avec éventuellement affichage du pourcentage sur l'ensemble des dialogues)
- Possibilité de renommer les clusters, et de rajouter des noms aux relations entre clusters ;
- Possibilité d'afficher le contenu d'un cluster, composé de plusieurs tours de parole. Ceux-ci peuvent être affichés sous la forme d'une liste, avec éventuellement des informations supplémentaires.

3.2.2 Manipulation de l'architecture du dialogue

Ces fonctionnalités permettent de modifier l'architecture locale ou globale du graphe. Le concepteur de l'agent dialoguant peut ainsi adapter et raffiner l'architecture obtenue en fonction de ses besoins. Il est ainsi possible de modifier :

- Le contenu d'un cluster donné. Notamment en supprimant un ou plusieurs tours de parole qui ne seraient pas homogènes thématiquement avec le cluster.
- L'architecture elle-même. Deux principales fonctionnalités sont utilisables. D'une part la fusion de deux clusters (par exemple si ces clusters sont similaires thématiquement et donc redondants). D'autre part la possibilité de sélectionner plusieurs tours de paroles d'un cluster donné, pour créer un nouveau cluster (ce qui revient à scinder en deux le cluster courant); cela est utile dans le cas où les tours de parole en question sont similaires sémantiquement, mais hétérogènes par rapport à la principale thématique exprimée dans le cluster. Si l'une de ces fonctionnalités est utilisée, l'affichage du nombre de clusters et de leurs relations est mis à jour.

Nous avons utilisé ce prototype pour plusieurs usescase réels, dont celui décrit dans (Bou-raoui et Lemaire, 2017). La figure 2 présente la version actuelle du frontend du prototype. Il sera présenté plus en détails lors de la démonstration.

3. Le nom attribué initialement à un cluster par le backend est une suite non significative de caractères.

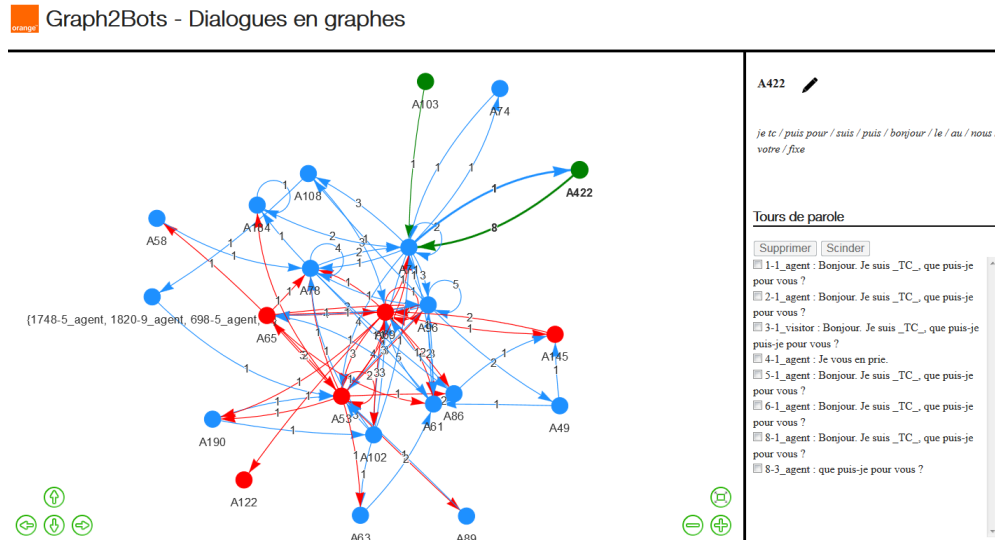


FIG. 2 – Graphe pour la conception d'architecture

Références

- Boullé, M., R. Guigourès, et F. Rossi (2014). Analyse exploratoire par k-coclustering avec khlops coviz. In *Advances in Knowledge Discovery and Management*, Volume 527, pp. 15–35.
- Bouraoui, J.-L. et V. Lemaire (2017). Cluster-based graphs for conceiving dialog systems. In *Workshop DMNLP at European Conference on Machine Learning (ECML)*.

Summary

We demonstrate a prototype allowing the unsupervised modeling of the structure of task-oriented dialogues. The dialogues are related to a given domain (for instance, train booking). This presents several advantages; notably its use as a basis for conceiving a conversational agent architecture. The modeling is represented by a graph. It displays the main stages of the dialogues and the transitions between them. Our approach consists in applying the coclustering to the targeted dialogue corpus. Thus we obtain the main themes that appear in the corpus. We then compute the theme transitions within each dialogue. Our tool allows to obtain and manipulate the corresponding graph. We detail the various functionalities demonstrated.