Ordonnancement de réponses dans les systèmes de dialogue basé sur une similarité contexte/réponse

Basma El Amel Boussaha Nicolas Hernandez Christine Jacquin
Emmanuel Morin
Laboratoire des Sciences du Numérique de Nantes (LS2N UMR 6004)
2 rue de la houssinière, BP 92208, 44322 Cedex 3 Nantes, France
prénom.nom@ls2n.fr

RÉSUMÉ

Construire des systèmes de dialogue qui conversent avec les humains afin de les aider dans leurs tâches quotidiennes est devenu une priorité. Certains de ces systèmes produisent des dialogues en cherchant le meilleur énoncé (réponse) parmi un ensemble d'énoncés candidats. Le choix de la réponse est conditionné par l'historique de la conversation appelé contexte. Ces systèmes ordonnent les énoncés candidats par leur adéquation au contexte, le meilleur est ensuite choisi. Les approches existantes à base de réseaux de neurones profonds sont performantes pour cette tâche. Dans cet article, nous améliorons une approche état de l'art à base d'un dual encodeur LSTM. En se basant sur la similarité sémantique entre le contexte et la réponse, notre approche apprend à mieux distinguer les bonnes réponses des mauvaises. Les résultats expérimentaux sur un large corpus de chats d'Ubuntu montrent une amélioration significative de 7, 6 et 2 points sur le Rappel@(1, 2 et 5) respectivement par rapport au meilleur système état de l'art.

ABSTRACT

Response ranking in dialogue systems based on context-response similarity

Building dialogue systems that converse with humans in order to help them in their daily tasks is becoming a priority. Some of these systems produce dialogues by finding the best response among a set of candidate responses. The choice of the best response is based on the history of the conversation called context. These systems rank the candidate responses by their relevance to the context, the best response is then chosen. Approaches based on deep neural networks performed well on this task. In this work, we improve a state of the art approach based on an LSTM dual encoder. By capturing semantic similarities between the context and the response, our approach learns to match the context with the best response. Experimental results on the Ubuntu Dialogue Corpus have shown a significant improvement of about 7, 6 and 2 points on Recall@(1, 2 and 5) compared to the best state of the art system.

MOTS-CLÉS : conversations écrites, dual encodeur, ordonnancement, agents conversationnels, apprentissage profond.

KEYWORDS: written conversations, dual encoder, ranking, chatbots, deep learning.

1 Introduction

Face au nombre croissant d'internautes, l'assistance automatique demeure la solution la plus adaptée pour les aider à résoudre leurs problèmes quotidiens. Grâce aux systèmes conversationnels, une assistance automatique peut être garantie pour chacun d'eux avec un coût minimal [1]. Ces systèmes appelés *chatbots* (agents conversationnels) sont capables de comprendre les besoins de l'utilisateur à travers des échanges textuels pour ensuite lui proposer une solution. Selon la nature de ces systèmes conversationnels, nous distinguons deux types (Figure 1) : les systèmes génératifs et les systèmes d'ordonnancement de réponses (Lowe *et al.*, 2017b). Les premiers systèmes génèrent des énoncés mot par mot, quant aux seconds, ils sélectionnent le bon énoncé parmi un ensemble d'énoncés candidats. En outre, et selon la tâche, nous distinguons deux catégories de systèmes conversationnels. La première catégorie regroupe les systèmes spécifiques à un domaine comme les systèmes de recommandation des restaurants (Wen *et al.*, 2017) et de réservation des tickets de cinéma (Li *et al.*, 2017). La deuxième catégorie comprend les systèmes de dialogue non spécifiques au domaine tels que SIRI [2], Alexa [3] et Replika [4]. Dans ce travail nous étudions les systèmes conversationnels qui sont à la fois des systèmes d'ordonnancement de réponses et spécifiques au domaine.

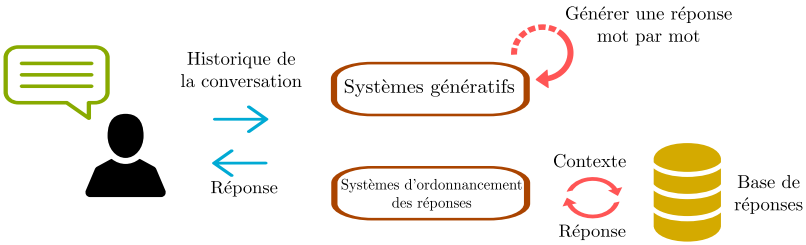

FIGURE 1: Les types des systèmes de dialogue.

Les conversations contiennent plusieurs tours de paroles entre deux ou plusieurs individus. Afin de produire une réponse adéquate à une conversation, il est important de considérer tous ces tours (nature multi-tours) ce qui rend la tâche plus complexe. Plusieurs travaux ont abordé le problème de la sélection du prochain tour de parole dans les conversations écrites. Certains exploitent tous les tours de parole de ces conversations pour sélectionner la réponse la plus adéquate à chacun d'eux (Lowe *et al.*, 2015; Kadlec *et al.*, 2015; Xu *et al.*, 2017; Wu *et al.*, 2017). D'autres négligent cette information pour ordonner les réponses candidates selon leur pertinence vis-à-vis du dernier tour de parole (Wang *et al.*, 2013; Wu *et al.*, 2016).

Les systèmes d'ordonnancement de réponses classent un ensemble de réponses candidates en se basant sur leur cohérence par rapport au contexte de la conversation. Dans la table 1, un exemple d'une conversation technique entre deux internautes extraite du corpus de dialogue d'Ubuntu (Lowe *et al.*, 2015) est illustré. Nous appelons *contexte*, l'historique de la conversation dans lequel nous concaténons les tours de parole de ces deux internautes. Dans cet exemple, un système d'ordonnancement de réponses doit classer la première réponse avant la deuxième. Il est important que le système capture les informations en commun (portées par les mots en gras) entre le contexte et chacune des réponses candidates. Selon Wu *et al.* (2017), les difficultés de la tâche d'ordonnancement de réponses

1. Comparé aux coûts engendrés par les humains travaillant comme assistants.
2. https://www.apple.com/ios/siri/
3. https://developer.amazon.com/alexa
4. http://replika.ai/

proviennent d'une part, de comment identifier les informations importantes (mots, phrases et énoncés) dans le contexte et de comment apparier ces informations avec celles présentes dans la réponse et d'autre part, de comment modéliser les relations entre les énoncés du contexte.

	Contexte
Tour 1	Hi, I can not longer access the graphical **login screen** on ubuntu 12.04
Tour 2	What exactly happen ?
Tour 3	I can't remember the error message, would it have auto-logged to a file or should I reboot quick ?
Tour 4	You mean it won't **automaticaly start** and what happen then ?
Tour 5	It just stop at a text **screen**, but I can access the command line **login** via alt F1-6, and **start** x **manually** there. I think it might me **lightdm** that's break but I'm not sure
	Réponses candidates
Réponse 1	For me **lightdm** often won't start **automatically** either. It show me console tty1 instead and I have to **start lightdm manually** ✓
Réponse 2	What about sources.list ? ✗

TABLE 1: Exemple d'une conversation technique entre deux participants extraite du corpus de dialogue d'Ubuntu (Lowe *et al.*, 2015). La première réponse candidate est la bonne réponse, tandis que la deuxième réponse ne peut être le prochain tour de parole.

Les travaux récents soit utilisent des architectures complexes pour capturer des similarités entre le contexte et la réponse ou nécessitent des modules externes afin de produire des informations complémentaires utiles à la tâche (Lowe *et al.*, 2015; Kadlec *et al.*, 2015; Wu *et al.*, 2016; Xu *et al.*, 2017). Certains de ces modules requièrent en plus, des connaissances externes collectées manuellement et qui sont fortement liées à un domaine d'application spécifique.

Dans ce travail, nous améliorons un système d'ordonnancement de réponses à base d'un dual encodeur (Lowe *et al.*, 2015). Pour cela, nous proposons une architecture simple, qui ne requiert pas de modules externes et entraînée de bout en bout. Notre système s'appuie sur la similarité entre le contexte de la conversation et la réponse candidate. À partir des vecteurs du contexte et de la réponse encodés par un dual encodeur LSTM (*Long Short-Term Mermory*) (Hochreiter & Schmidhuber, 1997), nous calculons leur produit vectoriel. Le vecteur résultant mesure la similarité entre le contexte et la réponse. Nous transformons ce vecteur de similarité en une probabilité avec une couche entièrement connectée et une fonction sigmoïde. Cette probabilité est utilisée par la suite pour ordonner les réponses candidates selon leur adéquation au contexte. Cette nouvelle méthode de calcul de similarité permet de capturer des propriétés sémantiques communes entre le contexte et la réponse. Nous avons évalué notre approche sur un large corpus de dialogues issus du canal *#Ubuntu* sur le Freenode IRC et nous avons suivi Lowe *et al.* (2015), Wu *et al.* (2016), Xu *et al.* (2017) et Wu *et al.* (2017) pour le choix des métriques d'évaluation : le Rappel@k et le Mean Recall Rank (MRR). Les résultats expérimentaux montrent des améliorations significatives [5].

La suite de cet article est organisée comme suit : nous résumons les travaux autour des systèmes conversationnels dans la section 2. Ensuite nous formalisons le problème et décrivons l'architecture de notre système dans la section 3. Les détails d'implémentation et les résultats expérimentaux sont donnés en section 4. Nous concluons dans la section 5 avec quelques perspectives.

5. Validées à l'aide d'un test de significativité.

2 État de l'art

Récemment, plusieurs travaux se sont orientés vers la construction de systèmes conversationnels à base de réseaux de neurones profonds. Dans ce cadre, la plupart des systèmes génératifs se basent sur l'architecture *séquence à séquence* de Sutskever *et al.* (2014) pour générer des dialogues (Vinyals & Le, 2015; Serban *et al.*, 2016; Sordoni *et al.*, 2015). Bien que ces systèmes génèrent des énoncés personnalisés pour chaque contexte de conversation, ils ont tendance à générer des réponses courtes et générales (Shao *et al.*, 2017; Li *et al.*, 2016). Ceci est dû essentiellement à la complexité de la tâche et au choix des fonctions objectifs qui entraîne un manque de diversité dans les réponses générées (Li *et al.*, 2016). En revanche, les systèmes d'ordonnancement de réponses sont capables de trouver des énoncés plus précis et syntaxiquement corrects dans le cas où ces énoncés figurent dans l'ensemble des réponses candidates. Ce type de système est au centre de nos intérêts dans le cadre de ce travail.

Lowe *et al.* (2015) ont proposé un système d'ordonnancement de réponses à base de *dual encodeur*. Le principe de ce système consiste à encoder le contexte et la réponse candidate séparément dans deux vecteurs. Le contexte consiste en la concaténation des tours de parole successifs dans l'historique de la conversation. Ensuite un score de similarité est calculé comme étant un produit de ces deux vecteurs et d'une matrice de paramètres appris par le système. Ce score est utilisé pour ordonner les réponses candidates. De plus, différentes variantes de cette approche à base de LSTM et de RNN (*Recurrent Neural Network*) ont été étudiées dans le même travail. Une extension de cette étude a été réalisée dans le travail de Kadlec *et al.* (2015) dans laquelle une approche ensembliste à base du dual encodeur de Lowe *et al.* (2015) a été déployée regroupant 11 LSTMs, 7 Bi-LSTMs et 10 CNNs. Une moyenne des scores de ces systèmes est calculée pour obtenir le score final de la réponse candidate.

Inspirés par le fonctionnement du cerveaux humain, Xu *et al.* (2017) ont incorporé dans leur travail des connaissances sur le domaine pour mieux modéliser le contexte et la réponse. Ils ont introduit pour la première fois une nouvelle cellule r-LSTM qui a une porte supplémentaire appelée *"Recall Gate"*. Cette cellule, comme son nom l'indique, sert à mémoriser les connaissances sur le domaine. D'abord ces connaissances sont obtenues grâce à une base de connaissance construite manuellement et qui permet d'obtenir des mots liés au domaine à partir du contexte et de la réponse candidate. En plus du contexte et de la réponse candidate, les informations liées au domaine sont encodées grâce à un encodeur r-LSTM en un vecteur qui résume toute la conversation. Ce vecteur est transformé en une probabilité utilisée comme score d'ordonnancement de réponses.

Wu *et al.* (2017) ont développé un système qui considère cette fois-ci les tours de parole séparément. Ils ont extrait deux types d'information de chaque tour de parole sous forme de deux matrices : la similarité au niveau des mots et des tours de parole. Grâce à une succession de convolution et de max-pooling, ces matrices de similarité ont été transformées en des vecteurs. Ensuite, ces vecteurs ont été accumulés grâce à un réseau de neurones récurrents à base d'unité GRU (Gated Recurrent Unit) (Chung *et al.*, 2014) afin d'obtenir un score de correspondance entre le contexte et la réponse.

Contrairement à tous ces travaux, Wang *et al.* (2013); Wu *et al.* (2016) se sont limités au dernier tour de parole. Wu *et al.* (2016) ont exploité le sujet de la conversation comme information supplémentaire afin d'améliorer la qualité de la réponse sélectionnée. Ils ont utilisé un modèle de sujets : le Twitter LDA (Zhao *et al.*, 2011) afin de générer un sujet pour le contexte et la réponse candidate. Le contexte, la réponse et leurs sujets respectifs ont été représentés par des plongements de mots et transformés en des vecteurs grâce à une convolution et au max-pooling. Ensuite ces vecteurs ont été appariés deux à deux grâce à des Réseaux de Tenseurs Neuronaux (*Neural Tensor Networks NTNs*) (Socher *et al.*,

2013) afin d'obtenir le score de la réponse. Bien que d'autres types d'information ont été pris en compte, cette restriction au dernier tour de parole est une hypothèse forte qu'il faudrait lever.

Dans ce travail, nous adoptons le premier système qui a abordé le problème d'ordonnancement de réponses avec une architecture neuronale : le dual encodeur de Lowe *et al.* (2015). Nous proposons une nouvelle approche de calcul du score de la réponse candidate. Contrairement à l'approche ensembliste de Kadlec *et al.* (2015) qui génère plusieurs paramètres à raffiner dans le cas où nous changeons de domaine d'application, notre approche est simple et facilement portable. Notre système ne requiert pas d'informations externes liées au domaine contrairement au système de Xu *et al.* (2017) ce qui favorise encore plus son adaptation à d'autres domaines. De plus le problème de reproductibilité des résultats de Wu *et al.* (2017), comme expliqué dans la section 4, ne nous a pas permis d'exploiter leur système et a motivé notre choix de nous appuyer sur l'architecture du système initial.

3 Modèle

Dans cette section, nous formalisons le problème auquel nous nous intéressons et nous décrivons l'architecture de notre système d'ordonnancement de réponses.

3.1 Formalisation du problème

Étant donné un contexte C de conversation entre deux utilisateurs sous forme d'une succession de n tours de paroles t_i tel que $C = \{t_1, t_2, t_3, ...t_n\}$. Le problème consiste à sélectionner le prochain tour de parole t_{n+1} appelé la réponse à ce contexte parmi un ensemble de m réponses possibles $t_{n+1} \in \{r_1, r_2, r_3, ..., r_m\}$. Nous définissons le problème comme étant un problème d'ordonnancement dans lequel nous classons les réponses candidates dans un ordre croissant de leurs pertinences vis-à-vis du contexte de la conversation. La réponse ayant le plus grand score est choisie comme étant le prochain tour de parole dans la conversation.

3.2 Architecture du modèle

Inspirés par le système de Lowe *et al.* (2015), nous proposons une architecture améliorée du dual encodeur à base de LSTM entraînée de bout en bout (Figure 2). Tout d'abord nous concaténons les tours de parole du contexte en gardant un simple marqueur de fin de tour. Nous ne mettons aucune restriction sur la taille du contexte en termes du nombre de tours de parole présents. L'idée de base consiste à représenter le contexte C et la réponse R d'abord en utilisant les plongements des mots. Ensuite ces plongements $e_1, e_2, .., e_j$ sont fournis dans l'ordre chronologique des mots à un encodeur. Cet encodeur consiste en un réseau de neurones récurrents à base de cellules LSTM, dont la couche cachée est mise à jour à chaque fois qu'un plongement de mots est donné en entrée. Ce processus est modélisé dans le cadre en Figure 2, il est similaire à celui déployé dans le système de Lowe *et al.* (2015). En sortie, nous récupérons la dernière couche cachée de l'encodeur C' et R' qui représente dans ce cas le contexte dans son ensemble et la réponse respectivement.

Lowe *et al.* (2015) calculent le score de la réponse candidate R par rapport au contexte C en multipliant C' par R' et par une matrice M de paramètres appris par le modèle. Dans notre approche, le score est calculé à partir d'un produit vectoriel P entre C' et R' qui reflète la similarité entre le

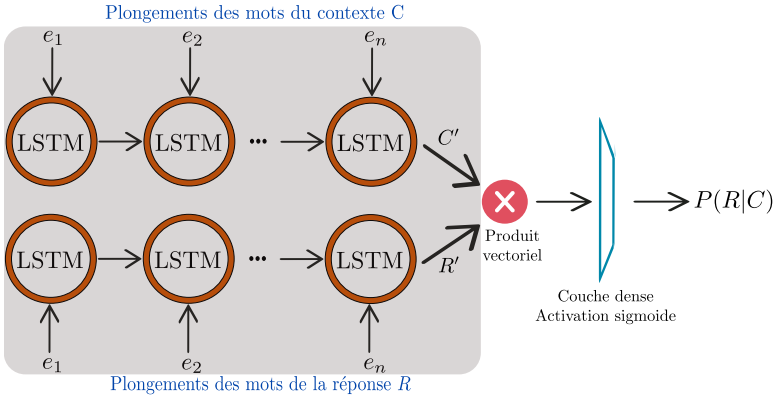

FIGURE 2: Architecture de notre système à base de dual encodeur

contexte et la réponse. Le résultat est transformé en une probabilité grâce à une fonction sigmoïde. Notre motivation réside dans le fait que dans une conversation le contexte et la réponse partagent des notions communes. Ces notions sont capturées d'abord par les plongements de mots et ensuite, grâce aux encodeurs et au calcul de la similarité, nous capturons des similarités sémantiques entre le contexte et la réponse.

Dans l'approche de base, Lowe *et al.* (2015) définissent la tâche de recherche du prochain tour de parole comme une tâche de génération. En plus des vecteurs C' et R', leur dual encodeur apprend une matrice M de paramètres qui sera utilisée pour générer implicitement une réponse R'' en multipliant C' par M (Équation 1).

$$R'' = C'^T \cdot M \tag{1}$$

Ensuite cette réponse R'' générée à partir du contexte de la conversation est comparée à la réponse que le système devrait prédire R'. La comparaison est effectuée grâce à un produit scalaire entre R'' et R' (Équation 2). Ce score de similarité est utilisé par la suite pour ordonner les réponses candidates.

$$Score = R'' \cdot R' \tag{2}$$

Dans l'approche que nous proposons dans ce travail, nous ordonnons les réponses candidates selon leur similarité sémantique directe avec le contexte de la conversation. Nous calculons cette similarité via un produit vectoriel entre le vecteur du contexte et le vecteur de la réponse que nous obtenons à partir de l'encodeur. Ce produit mesure l'intensité du lien entre chacune des réponses candidates et le contexte de la conversation. En conséquence, notre système apprend à ordonner les réponses en prenant en compte la sémantique partagée entre celles-ci et le contexte. Ceci explique l'amélioration des résultats que nous obtenons par rapport aux systèmes état de l'art de Lowe *et al.* (2015) et Kadlec *et al.* (2015).

4 Expériences et résultats

Dans cette section, nous présentons notre environnement expérimental ainsi que les résultats d'évaluation. Nous commençons par décrire le corpus sur lequel nous avons évalué notre système. Ensuite nous définissons les métriques d'évaluation que nous avons utilisées. Enfin nous discutons les résultats expérimentaux ainsi que les paramètres de notre système.

4.1 Corpus de dialogues d'Ubuntu

Dans le travail de Lowe *et al.* (2015) un large corpus de dialogues provenant de chat d'Ubuntu a été construit. Ce corpus comprend environ un million de conversations entre deux utilisateurs ayant au moins trois tours de parole. Ces conversations sont issues des logs du canal *#Ubuntu* sur le Freenode IRC (Internet Relay Chat) [6]. Les conversations sont du chat en anglais et traitent des sujets techniques divers. La première version (V1) de ce corpus comprenait quelques lacunes qui ont été corrigées plus tard dans la deuxième version (V2) [7]. La table 2 présente quelques statistiques et propriétés de la version 2 de ce corpus.

# énoncés (au total)	7 100 000
# tours de parole (au total)	5 139 574
# mots (au total)	100 000 000
Min. # tours de parole par dialogue	3
Moy. # tours de parole par dialogue	4,94
Moy. # mots par énoncé	10,34
# dialogues d'entraînement	1 000 000
# dialogues de test	18 920
# dialogues de validation	19 560

TABLE 2: Propriétés du corpus d'Ubuntu V2

Le corpus contient un million de dialogues pour l'entraînement, 19 560 pour la validation et 18 920 pour le test. Chaque élément d'entraînement est un triplet *(contexte, réponse, étiquette)*. L'étiquette est à "1" dans le cas où la réponse est le prochain tour de parole, "0" dans le cas contraire. Dans les ensembles de validation et de test, chaque élément est composé d'un contexte, d'une bonne réponse et de neuf mauvaises réponses (extraites aléatoirement d'autres conversations).

La tâche sur ce corpus consiste à ordonner la bonne réponse au rang supérieur par rapport aux neuf mauvaises réponses. Nous avons choisi d'évaluer notre système sur ce corpus pour essentiellement deux raisons. La première est liée à notre objectif de construire un système d'ordonnancement de réponses spécifique à un domaine qui est ici, l'assistance technique autour d'Ubuntu. La deuxième raison est le fait que plusieurs systèmes d'ordonnancement de réponses ont été évalués sur ce corpus, ce qui nous a aidé à préparer notre environnement d'évaluation et permis de comparer les résultats obtenus.

6. Sur la période 2004-2015 disponible sur `https://irclogs.ubuntu.com/`
7. Disponible sur `https://github.com/rkadlec/ubuntu-ranking-dataset-creator`

4.2 Métriques d'évaluation

L'évaluation des systèmes conversationnels est un domaine de recherche ouvert dans lequel il n'y a pas de métriques standards (Lowe *et al.*, 2017a; Liu *et al.*, 2016). Nous avons suivi Lowe *et al.* (2015), Wu *et al.* (2016), Xu *et al.* (2017) et Wu *et al.* (2017) dans l'utilisation du *Rappel@k*, du *Mean Recall Rank* (MRR) (Voorhees, 2001) comme métriques d'évaluation de notre système d'ordonnancement de réponses. Ces deux métriques mesurent la capacité du système à ordonner la bonne réponse avant les mauvaises réponses. Notons qu'en raison de la présence d'une seule bonne réponse pour chaque contexte dans notre corpus de dialogue d'Ubuntu, la *Mean Average Precision* (MAP) (Baeza-Yates & Ribeiro-Neto, 1999) et la *Précision@1* sont équivalentes aux MRR et Rappel@1 respectivement.

4.3 Systèmes état de l'art

Nous rapportons les résultats de quatre systèmes état de l'art auxquels nous comparons notre système. Nous avons vérifié la version du corpus d'Ubuntu sur laquelle chacun de ces systèmes a été évalué et nous rapportons les résultats dans le cas où il s'agit de la version V2. Dans le cas où le système a été évalué sur la V1 uniquement, nous l'avons ré-évalué en utilisant le code source des auteurs (si disponible) sur la V2. Les systèmes auxquels nous avons comparé notre système sont les suivants :

TF-IDF Nous rapportons les résultats de l'approche Term Frequency-Inverse Document Frequency présentée comme système état de l'art dans le travail de Lowe *et al.* (2015) et ré-évalué sur la V2 du corpus d'Ubuntu par Lowe *et al.* (2017b). Le contexte et chacune des réponses candidates sont représentés par un vecteur des scores TF-IDF des mots. Ensuite une similarité cosinus est calculée entre le vecteur du contexte et de la réponse afin d'obtenir un score de réponses.

RNN/LSTM dual encodeur Ces deux modèles ont été introduits dans le travail de Lowe *et al.* (2015) et ré-évalués sur la V2 du corpus par la suite dans le travail de Lowe *et al.* (2017b).

BiLSTM dual encodeur Ce système a été évalué par Kadlec *et al.* (2015) sur la V1 du corpus. Grâce au code source de Lowe *et al.* (2015), nous avons ré-évalué ce même système sur la V2 en utilisant les paramètres que les auteurs avaient décrits dans leur article.

4.4 Résultats

Dans la table 3, les trois premières lignes rapportent les résultats des systèmes de Lowe *et al.* (2017b). Le BiLSTM dans la ligne 4 est le système de Kadlec *et al.* (2015) que nous avons ré-évalué sur la V2. Notons bien que notre système surpasse les autres systèmes état de l'art et ceci sur toutes les métriques Rappel@k. Le Rappel@1 est une mesure forte de la capacité du système à ordonner la bonne réponse en premier parmi les 10 réponses candidates. De plus, nous remarquons que l'utilisation des cellules LSTM Bidirectionnelles (BiLSTM) dans notre système permet d'améliorer les résultats. Dans cette approche, nous obtenons deux vecteurs pour le contexte et la réponse grâce aux BiLSTM que nous concaténons pour obtenir un vecteur représentatif. Les résultats de la table 3 sont obtenus à partir d'une seule exécution des programmes pour laquelle l'entraînement converge sur l'ensemble de validation.

Comme expliqué dans la section 3, la différence dans la manière dont nous calculons la similarité entre le contexte et la réponse améliore significativement les résultats et nous permet de gagner environ 7, 6 et 2 points sur le Rappel@(1, 2 et 5) respectivement. Nous avons réalisé le test T de

Méthode	Rappel@1	Rappel@2	Rappel@5	MRR
TF-IDF (Lowe *et al.*, 2017b)	48,8 %	58,7 %	76,3 %	-
RNN Dual Encodeur (Lowe *et al.*, 2017b)	37,9 %	56,1 %	83,6 %	-
LSTM Dual Encodeur (Lowe *et al.*, 2017b)	55,2 %	72,1 %	92,4 %	-
BiLSTM Dual Encodeur (Kadlec *et al.*, 2015)	54,2 %	71,6 %	91,9 %	-
Similarité LSTM Dual Encodeur *(Sim LSTM DE)*	**62,9** † %	**78,5** † %	**95,2** † %	**76,1** † %
Similarité BiLSTM Dual Encodeur *(Sim BiLSTM DE)*	**63,7** † %	**79,1** † %	**95,2** † %	**76,7** † %

TABLE 3: Résultats de l'évaluation en utilisant les métriques Rappel@k et MRR.

Student (Student, 1908) pour évaluer la significativité des résultats. Les résultats de ce test montrent une nette significativité des écarts entre notre approche et celle de base que nous avons améliorée.

4.5 Évaluation de l'impact des plongement de mots

Dans ce travail, nous nous sommes intéressés à l'étude de l'impact des plongements de mots sur les performances de notre système. Pour cela, nous avons réalisé un ensemble d'expérimentations que nous résumons dans la table 4. Nous avons comparé 3 variantes de notre système. Pour chacune des variantes, nous avons changé les poids initiaux de la couche des plongements des mots. Nous avons fixé la taille de ces vecteurs à 300 et nous avons exploré l'impact de l'affinement des poids de la couche des plongements de mots comparé au non affinement (gèle) des poids durant la phase d'entraînement.

Les modèles de plongements de mots utilisés dans cette étude pour initialiser les poids de la couche des plongements de mots sont les suivants :

— **Word2Vec** : Nous avons entraîné word2vec (Mikolov *et al.*, 2013) sur l'ensemble d'entraînement en utilisant Gensim (Řehůřek & Sojka, 2010). Taille du vocabulaire = 770k.

— **FastText** : Nous avons utilisé des vecteurs de plongement de mots pré-entraînés sur Wikipedia en utilisant FastText [8] (Bojanowski *et al.*, 2017). Taille du vocabulaire = 2.5M.

— **Glove** : Nous avons utilisé des vecteurs pré-entraînés avec Glove (Pennington *et al.*, 2014) [9] sur Common Crawl Corpus [10]. Taille du vocabulaire = 2.2M.

Système	Rappel@1	Rappel@2	Rappel@5	MRR
Word2Vec-gelé	62,2 %	77,8 %	94,6 %	75,5 %
Word2Vec-affiné	63,3 %	78,4 %	94,9 %	76,2 %
FastText-gelé	58,9 %	75,1 %	94,2 %	73,2 %
FastText-affiné	61,7 %	77,8 %	94,7 %	75,2 %
Glove-gelé	62,5 %	78,0 %	94,8 %	75,7 %
Glove-affiné	62,9 %	78,5 %	95,2 %	76,1 %

TABLE 4: Évaluation de l'impact des plongements de mots sur les performances de notre système.

†. Scores significativement différents du dual encodeur de base au seuil de confiance de 0,01 selon le test T de Student.

8. https://s3-us-west-1.amazonaws.com/fasttext-vectors/wiki.en.vec
9. http://nlp.stanford.edu/data/glove.840B.300d.zip
10. http://commoncrawl.org/the-data/

En se basant sur les résultats, nous observons que pour tous les systèmes, l'affinement des poids de la couche des plongements de mots permet d'améliorer les résultats. De plus, ces plongements pré-entraînés avec Glove donnent les meilleurs résultats. Nous avons remarqué que l'entraînement des plongements de mots au niveau des mots dans le cas de Glove et Word2Vec donnent de meilleurs résultats en comparaison à ceux obtenus au niveau des caractères dans le cas de FastText.

4.6 Évaluation qualitative et quantitative des résultats

En plus de l'évaluation de notre système via les métriques spécifiques aux systèmes d'ordonnancement de réponse, nous avons mené une étude qualitative et quantitative pour analyser les résultats obtenus. Dans ce but, nous avons entraîné le système de Lowe *et al.* (2015) à l'aide de leur code source jusqu'à obtenir des scores similaires à ceux rapportés dans leur publication. Nous avons ensuite comparé les prédictions obtenues par leur système et le nôtre.

Contexte	- Hello .. Is it possible to disable GPG check for a specific APT repository ? - Why would you ever need to do that - It's for a custom repository in enterprise environment. But that's unimportant isn't it. Was that a statement that it's not possible ?		
LSTM DE	Sim LSTM DE	Étiquette	Réponse
0,06	**0,87**	1	3rd party repo ? PPA ? what is it ?
0,29	0,25	0	Find it sticky edge
0,17	0,40	0	That response doesn't help me in the slightest

Contexte	- How can I remount a drive as read/write ? - mount -o rw /dev/whatever /whever I belive theres a remount option i think - Thanks - I'd say check the mount man page also. I forgot the syntax for the remount option		
LSTM DE	Sim LSTM DE	Étiquette	Réponse
0,96	0,49	1	Okay
0,62	**0,88**	0	Thats sound like a good idea find out I'm missing authz_hos somehow
0,14	0,87	0	Thanks I will read that

Contexte	- Is there a length limitation on the hostname in SSH ? - 255 char for the FQDN - FQDN ?		
LSTM DE	Sim LSTM DE	Étiquette	Réponse
0,99	**0,94**	1	Full Qualify Domain Name : mycomputer.kitchen.myhouse.com
0,01	0,27	0	Alright good luck
0,01	0,08	0	You have to do it once the bios hand off to grub

Contexte	- Is there a script that can generate a live cd iso of your currently run ubuntu hdd install ? - Remastersys - Have you use it ?		
LSTM DE	Sim LSTM DE	Étiquette	Réponse
0,05	0,18	1	I hadn't have much luck with it, but that is a while ago (2 years)
0,88	**0,71**	0	It can
0,91	0,56	0	Not for me I doubt I could figure it out to be honest, but any theme installations come to mind as a guess

TABLE 5: Exemples d'accord et de désaccord de prédictions entre notre système (Sim LSTM DE) et le système de base de Lowe *et al.* (2015) (LSTM DE). Les scores en gras sont les scores les plus élevés attribués par le système.

La table 5 représente quelques exemples extraits à partir de l'ensemble de test. Chaque exemple est composé d'un contexte qui contient entre trois et quatre tours de parole, trois réponses candidates

avec leurs étiquettes (1 : bonne réponse, 0 : mauvaise réponse) et le score de prédiction obtenu par chacun des systèmes. De plus, les statistiques en nombre de cas d'accord et de désaccord entre les deux systèmes sont données dans la table 6. Dans le premier exemple, notre système attribue le score le plus élevé à la bonne réponse contrairement au dual encodeur de base. Malgré la difficulté du choix de la réponse car aucune des réponses ne partage explicitement des mots avec le contexte, notre système a pu capturer des relations sémantiques entre *repo* et *repository*, *PPT* et *APT*.

Sim LSTM DE \ LSTM DE	Réussite	Échec
Réussite	7437	4476
Échec	2143	4864

TABLE 6: Statistiques sur le nombre de cas d'accord et de désaccord entre les deux systèmes testés. Rappelons que la taille de l'ensemble de test est de 18 920 dialogues.

Le deuxième exemple représente le cas où le dual encodeur de Lowe *et al.* (2015) a pu retrouver la bonne réponse contrairement à notre système. Notons bien que même si notre système n'a pas réussi à retrouver la bonne réponse en premier, il a attribué le deuxième meilleur score à la troisième réponse. Cette réponse peut très bien remplacer la bonne réponse sans altérer le sens. Dans le troisième exemple, les deux systèmes retrouvent la bonne réponse avec des scores très élevés. Le dernier exemple représente un cas de figure dans lequel les deux systèmes ont échoué. Toutefois nous pensons que les deux réponses marquées avec une étiquette *"0"* sont des réponses potentielles au contexte.

Nous voyons donc dans la table 6 que notre système obtient des meilleurs résultats que LSTM DE (ce qui corrobore les résultats donnés dans la table 3). Mais il reste quand même 2143 cas où LSTM DE a trouvé la bonne réponse alors que le nôtre en a donné une mauvaise. Réaliser l'analyse de ces cas pourrait permettre de comprendre les lacunes de notre système afin de l'améliorer. Dans le but d'obtenir une analyse plus précise, il serait aussi intéressant de prendre en compte parmi les mauvaises réponses données par les systèmes, celles qui pourraient être considérées complétement cohérentes par rapport au contexte.

4.7 Paramètres du système

Les plongements de mots ont été initialisés avec Glove (Pennington *et al.*, 2014) préalablement entraînés sur Common Crawl Corpus puis affinés durant l'entraînement [11]. Les seuls prétraitements effectués sont la tokenisation, la racinisation et ensuite la lemmatisation (dans l'ordre) disponibles au moment du téléchargement du corpus à l'aide de NLTK (Loper & Bird, 2002). Les paramètres du système ont été mis à jour avec un gradient de descente stochastique avec l'algorithme Adam (Kingma & Ba, 2015). Tout le modèle a été entraîné sur un seul GPU Titan X.

Le taux d'apprentissage initial est de 0,001 et les paramètres d'Adam β_1 et β_2 sont 0,9 et 0,999 respectivement. Comme stratégie de régularisation, nous avons utilisé le *"early-stopping"* et pour entraîner nos modèles, nous avons utilisé un mini-batch de taille 256. La taille des plongements de mots de la couche cachée du LSTM est de 300. La taille du contexte et de la réponse a été réduite à 160 mots. Nous avons implémenté notre système avec Keras (Chollet *et al.*, 2015) ayant Tensorflow (Abadi *et al.*, 2015) comme backend [12].

11. Notons que nous avons entraîné les plongements de mots sur l'ensemble d'entraînement sans amélioration des résultats.
12. Nous mettons en ligne le code source qui permet de reproduire nos résultats sur `https://github.com/basma-b/`

5 Conclusion et perspectives

Nous avons proposé dans ce travail une approche d'ordonnancement de réponses dans les conversations écrites à base de dual encodeur. Les résultats expérimentaux montrent que notre approche apporte des améliorations significatives en comparaison aux approches de l'état de l'art. La nouvelle méthode basée sur la similarité sémantique entre le contexte et la réponse pour le calcul du score de pertinence de chaque réponse permet en particulier de mieux associer le contexte à la bonne réponse.

Par la suite, nous souhaitons d'abord ré-évaluer les autres approches état de l'art (Wu *et al.*, 2016, 2017; Xu *et al.*, 2017) sur le même jeu de données. Nous avons aussi comme objectif d'améliorer la représentation du contexte de la conversation en considérant, cette fois-ci, les tours de paroles de manière distincte au lieu de les concaténer simplement. Nous analyserons en détail les statistiques obtenus pour mieux comprendre les raisons de réussite et d'échec de notre système et proposer d'éventuelles améliorations. De plus, nous introduirons le mécanisme d'attention pour apprendre une meilleure représentation du contexte. Nous souhaitons aussi évaluer l'impact des prétraitements tels que l'élimination des mots outils et le filtrage des urls, numéros, etc. sur cette approche. En outre, une évaluation de nos méthodes sur de plus grands corpus de différentes langues tels que Baidu Tieba (Wu *et al.*, 2016) et Douban (Wu *et al.*, 2017) est prévue.

6 Remerciements

Ce travail a été partiellement financé par le projet ANR 2016 PASTEL CE33-0007 [13]. Nous remercions les relecteurs anonymes pour leurs nombreuses remarques qui ont été utiles pour l'amélioration du contenu de l'article.

Références

ABADI M., AGARWAL A., BARHAM P., BREVDO E., CHEN Z., CITRO C., CORRADO G. S., DAVIS A., DEAN J., DEVIN M., GHEMAWAT S., GOODFELLOW I., HARP A., IRVING G., ISARD M., JIA Y., JOZEFOWICZ R., KAISER L., KUDLUR M., LEVENBERG J., MANÉ D., MONGA R., MOORE S., MURRAY D., OLAH C., SCHUSTER M., SHLENS J., STEINER B., SUTSKEVER I., TALWAR K., TUCKER P., VANHOUCKE V., VASUDEVAN V., VIÉGAS F., VINYALS O., WARDEN P., WATTENBERG M., WICKE M., YU Y. & ZHENG X. (2015). TensorFlow : Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

BAEZA-YATES R. A. & RIBEIRO-NETO B. (1999). *Modern Information Retrieval*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics (TACL)*, **5**, 135–146.

CHOLLET F. *et al.* (2015). Keras. `https://github.com/keras-team/keras`.

`dual_encoder_udc`

13. `http://www.agence-nationale-recherche.fr/?Projet=ANR-16-CE33-0007`

CHUNG J., GULCEHRE C., CHO K. & BENGIO Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Workshop on Deep Learning and Representation Learning at the 28th Annual conference on Advances in Neural Information Processing Systems (NIPS'14)*, Montreal, Canada.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

KADLEC R., SCHMID M. & KLEINDIENST J. (2015). Improved deep learning baselines for ubuntu corpus dialogs. In *Workshop on Machine Learning for Spoken Language Understanding and Interaction at the 29th Annual Conference on Neural Information Processing Systems (NIPS'15)*, Montreal, Canada.

KINGMA D. & BA J. (2015). Adam : A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR'15)*, San Diego, CA, USA.

LI J., GALLEY M., BROCKETT C., GAO J. & DOLAN B. (2016). A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL'16)*, p. 110–119, San Diego, CA, USA.

LI X., CHEN Y.-N., LI L., GAO J. & CELIKYILMAZ A. (2017). End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (AFNLP'17)*, p. 733–743, Taipei, Taiwan.

LIU C.-W., LOWE R., SERBAN I., NOSEWORTHY M., CHARLIN L. & PINEAU J. (2016). How not to evaluate your dialogue system : An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, p. 2122–2132, Austin, Texas.

LOPER E. & BIRD S. (2002). Nltk : The natural language toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics (ETMTNLP'02)*, p. 63–70, Stroudsburg, PA, USA.

LOWE R., NOSEWORTHY M., SERBAN I. V., ANGELARD-GONTIER N., BENGIO Y. & PINEAU J. (2017a). Towards an automatic turing test : Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, p. 1116–1126, Vancouver, Canada.

LOWE R., POW N., SERBAN I. & PINEAU J. (2015). The ubuntu dialogue corpus : A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL'15)*, p. 285–294, Prague, Czech Republic.

LOWE R. T., POW N., SERBAN I. V., CHARLIN L., LIU C.-W. & PINEAU J. (2017b). Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue & Discourse*, **8**(1), 31–65.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR'13)*, p. 1–12, Scottsdale, Arizona.

PENNINGTON J., SOCHER R. & MANNING C. (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, p. 1532–1543, Doha, Qatar.

ŘEHŮŘEK R. & SOJKA P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Worshop on New Challenges for NLP Frameworks at the 7th edition of the Language Resources and Evaluation Conference (LREC'10)* , p. 45–50, Valletta, Malta.

SERBAN I. V., SORDONI A., BENGIO Y., COURVILLE A. & PINEAU J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, p. 3776–3783, Phoenix, AZ, USA.

SHAO Y., GOUWS S., BRITZ D., GOLDIE A., STROPE B. & KURZWEIL R. (2017). Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*, p. 2210–2219, Copenhagen, Denmark.

SOCHER R., CHEN D., MANNING C. D. & NG A. (2013). Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 26th international conference on Advances in Neural Information Processing Systems (NIPS'13)*, p. 926–934. Lake Tahoe, NV, USA.

SORDONI A., BENGIO Y., VAHABI H., LIOMA C., GRUE SIMONSEN J. & NIE J.-Y. (2015). A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM'15)*, p. 553–562, Melbourne, Australia.

STUDENT (1908). The probable error of a mean. *Biometrika*, p. 1–25.

SUTSKEVER I., VINYALS O. & LE Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 2014 conference on Advances in Neural Information Processing Systems (NIPS'14)*, p. 3104–3112. Montreal, Canada.

VINYALS O. & LE Q. (2015). A neural conversational model. In *Workshop on Deep Learning at the 31 st International Conference on Machine Learning (ICML'15)*, Lille, France.

VOORHEES E. M. (2001). The trec question answering track. *Natural Language Engineering*, **7**(4), 361–378.

WANG H., LU Z., LI H. & CHEN E. (2013). A dataset for research on short-text conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*, p. 935–945, Seattle, WA, USA.

WEN T.-H., VANDYKE D., MRKŠIĆ N., GASIC M., ROJAS BARAHONA L. M., SU P.-H., ULTES S. & YOUNG S. (2017). A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL'17)*, p. 438–449, Valencia, Spain.

WU Y., WU W., LI Z. & ZHOU M. (2016). Response selection with topic clues for retrieval-based chatbots. *arXiv preprint arXiv :1605.00090*.

WU Y., WU W., XING C., ZHOU M. & LI Z. (2017). Sequential matching network : A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, p. 496–505, Vancouver, Canada.

XU Z., LIU B., WANG B., SUN C. & WANG X. (2017). Incorporating loose-structured knowledge into conversation modeling via recall-gate lstm. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'17)*, p. 3506–3513, Anchorage, AK, USA.

ZHAO W. X., JIANG J., WENG J., HE J., LIM E.-P., YAN H. & LI X. (2011). Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR'11)*, p. 338–349, Berlin, Heidelberg.