

Projet - Chatbot !

1 Le sujet : développement d'un chatbot

Un chatbot est un agent programmé pour entretenir un dialogue avec un utilisateur. À l'origine, les chatbots étaient destinés à une tâche spécifique (p. ex. réservation de billets de train, assistance pour les achats dans un magasin, psychologue, etc.), mais de nos jours, il existe aussi des chatbots destinés à la conversation en général, une tâche bien plus difficile.

Enjeux

- Le but d'un chatbot est de créer une situation de dialogue entre un système automatique et un utilisateur humain. Son langage doit donc essayer de refléter le mieux possible les énoncés attendus par un humain.
- Veillez à ce que les énoncés que vous produisez soient grammaticalement bien formés. Ceci peut être parfois difficile, mais fera une grande différence pour la qualité de votre chatbot !
- Plus votre chatbot prend en compte ce que dit l'utilisateur, plus il sera crédible. Toutefois, un système simple qui repère quelques mots clés peut être déjà assez efficace.

Travail demandé

Vous allez coder un chatbot qui parlera soit anglais, soit français. Votre chatbot aura trois modes différents. Le numéro du mode sera sélectionné par l'utilisateur sur la ligne de commande, au moment de lancer le chatbot dans le terminal. Il n'est pas nécessaire de développer une interface graphique, un chatbot fonctionnant entièrement dans le terminal conviendra parfaitement. Les trois modes sont décrits ci-dessous. Lorsque votre chatbot est lancé dans un mode donné, vous pourrez lui permettre de basculer temporairement vers un mode inférieur dans les cas où le mode d'origine, plus spécifique et plus complexe, ne trouve pas de réponse adaptée.

Mode 1 : Dans ce mode, votre chatbot répondra à chaque énoncé par ce qu'on appelle des « backchannels » (des mots comme « uh huh », « hmm », « continuez », « oui »...). Ce sont des mots ou phrases qui indiquent que le chatbot a entendu, mais qui sont totalement génériques (pensez à quelqu'un qui écoute au téléphone et qui répond très passivement). Vous définirez une liste d'au moins 5 backchannels et votre chatbot répondra avec un backchannel choisi aléatoirement. Mais faites en sorte que le backchannel soit toujours différent du dernier backchannel utilisé.

Mode 2 : Dans ce mode, votre chatbot répondra comme le chatbot psychologue Eliza. Ceci veut dire :

- Quand l'utilisateur dit « Je suis X » (p. ex. « Je suis content/triste/dans la cuisine », etc.) ou un équivalent en anglais, vous répondrez par quelque chose comme « Pourquoi est-ce que tu es X ? » Pensez à implémenter cela pour tous les temps du verbe « être » (« suis », « sera », « étais », etc.).
- Identifiez quelques mots-clés (« famille », « parents »...) qui correspondent à des sujets de discussion avec un psychologue (l'enfance en famille). Lorsque l'un de ces mots-clés est employé par l'utilisateur, le chatbot répondra par une question liée à ce sujet. Par exemple, si l'utilisateur écrit « On allait souvent en vacances avec mes **parents**. », le chatbot pourra répondre « Est-ce que

vous pensez que la **famille** est importante ? ». Définissez au moins 10 mots clés associés à au moins 5 sujets de discussion, et rédigez au moins 2 réponses possibles pour chaque sujet, parmi lesquelles le chatbot effectuera un choix aléatoire, sans répéter le choix précédent le cas échéant (comme pour les backchannels du mode 1).

Mode 3 : Ce mode est un mode « libre » : c'est à vous de définir la façon dont il ira au-delà du mode 2. Voici quelques directives et conseils :

- Vous pourrez choisir un domaine de compétence pour votre chatbot. Quel est son but ? Vous pouvez par exemple créer un chatbot pour interroger un site de réservation de billets de train, pour demander son chemin vers un lieu, ou pour entretenir une conversation. Si, à l'inverse, vous choisissez de développer un agent conversationnel général, vous vous concentrerez sur un ou deux thèmes de discussion que votre chatbot maîtrisera particulièrement bien.
- Vous pourrez utiliser la méthode de votre choix pour construire votre chatbot : heuristiques, motifs, mots-clés, grammaires (p. ex. **CFGs avec NLTK**), réponses génériques, etc.
- Les énoncés produits par votre chatbot devront être grammaticalement bien formés. Faites attention à l'accord, aux temps des verbes, etc. Évidemment, votre chatbot ne doit pas faire de fautes d'orthographe !
- Un dialogue est formé d'un locuteur (celui qui parle à un instant donné) mais aussi d'un interlocuteur (celui qui écoute). Pour créer un dialogue plus réaliste, pensez à inclure des énoncés qui montrent que l'agent prend en compte ce que dit l'utilisateur. Les « backchannels » sont des signaux verbaux ou non verbaux qui sont les réponses de l'interlocuteur (hochements de têtes pour les agents qui ont en ont une, des marqueurs de compréhension tels que « uh huh », « oui », des reprises de l'énoncé précédent en cas de manque de compréhension etc.). Vous pouvez utiliser le mode 1 pour cette fonctionnalité.
- Mettez en avant ce que votre agent arrive à faire, mais incluez dans votre rapport une section qui discute des limitations de votre chatbot : y a-t-il des choses qu'il ne peut pas gérer ? Quels types de solution proposeriez-vous pour l'avenir ?

2 Consignes générales

Le but de ce projet est de montrer que vous pouvez mener un projet jusqu'au bout. Faites donc du travail incrémental, en commençant par les choses les plus simples. Une partie importante de la note sera consacrée au rapport, la raison étant que, dans le monde de la recherche comme dans celui de l'entreprise, la façon de présenter et de vendre son travail est aussi (voire plus) importante que le travail lui-même. Le projet est à réaliser en groupes de deux ou trois.

La date de rendu de votre projet est le **samedi le 5 mai 2018 à 23h59** (-5pts par jour de retard¹).

Vous utiliserez un dépôt **git** (<https://github.com>) que vous partagerez entre les membres de votre groupe pour assurer la compatibilité entre les modifications que vous faites. Ce répertoire **git** devra impérativement être structuré comme détaillé à la section suivante. Vous m'enverrez un lien vers votre projet (bawden@limsi.fr) dès que vous avez créé le répertoire. Le rendu (avant la date limite) consistera donc en un simple mail, dont le sujet sera de la forme : **[TAL] nom1, nom2, (nom3)** qui m'indiquera que le répertoire est prêt pour être noté. La date d'arrivée du mail dans ma boîte mail fera foi. Vous pouvez envoyer plusieurs fois le mail avant la date limite. Tout mail de rendu arrivé en retard induira une retenue sur la note finale comme indiqué ci-dessus.

3 À rendre (dans le répertoire **git**)

1. Un fichier texte nommé **README** contenant les indications nécessaires pour exécuter votre programme. Vous indiquerez notamment les bibliothèques à installer (à l'exception des bibliothèques standard), les options pour la ligne de commande, ainsi que des exemples d'utilisation.
2. Dans un dossier **src**, le code source de votre chatbot, écrit en Python 3, séparé en fichiers et en fonctions de façon logique. La lisibilité du code sera l'un des critères de notation (y compris la présence de commentaires précis, concis, clairs et bien orthographiés).

1. Un jour de retard est compté à partir d'une minute après minuit.

3. Dans un dossier `doc`, votre rapport, écrit en \LaTeX et compilé en un PDF de 5 à 7 pages nommé `nom1-nom2-nom3.pdf` qui décrit votre travail (police et marges de taille raisonnable). Vous y décrirez l'objectif de votre projet, la motivation derrière ce que vous avez fait, les méthodes que vous avez utilisées, à quel point cela fonctionne, et les limitations de votre chatbot (c'est-à-dire les cas qui restent à traiter ou qui fonctionnent mal). Vous devez également inclure une petite section qui décrit de façon brève la contribution respective de chaque membre du groupe. Votre rapport peut être écrit en anglais ou en français. Le rapport ne doit pas inclure de code source, ni de portions de texte copiées-collées du présent sujet de TP.

4 Notation

Le barème ci-dessous n'est donné qu'à titre indicatif et pourra être modifié ultérieurement. Vous pouvez néanmoins l'utiliser comme guide général.

1. **Code et README : 7 points**
 - 2 pts pour la lisibilité et clarté de votre code
 - 5 pts pour le bon fonctionnement du programme et ses fonctionnalités, telles que décrites dans le rapport.
2. **Rapport : 13 points**
 - 4 pts pour l'orthographe, la grammaire et la bonne formulation (style professionnel) de votre rapport. Donc relisez-vous bien et prenez du plaisir à « vendre » votre travail !
 - 9 pts pour le contenu du rapport : est-ce que vous avez bien présenté la motivation du projet, ce que votre programme fait et ce qui resterait à faire dans l'avenir ? Évidemment, si vous avez fait réalisé du bon code, vous aurez plus d'idées et de détails pour le rapport, ce qui amènera à plus de points ici.

Enfin, le non-respect des consignes détaillées dans ce sujet conduira à des retraits de points.

5 Conseils

- Committez souvent votre travail vers le répertoire `git` et communiquez bien entre vous. Ceci évitera les conflits entre versions différentes du code. Votre contribution au projet pourra être vérifiée sur `github`, en regardant les commits que vous avez faits. Les notes seront différentes entre les membres d'une même équipe si le volume de travail de tel ou tel est clairement différent de celui des autres. Une bonne manière d'assurer la participation de tous est de bien diviser le travail en tâches à l'avance.
- Mettez-vous d'accord sur les espaces et les tabulations pour l'indentation de votre code. Par exemple, décidez que toute tabulation sera remplacée automatiquement par 4 espaces (en modifiant les paramètres de votre éditeur de texte), ou soyez sûrs que tout le monde utilise les tabulations ou, plus simplement encore, le même éditeur de texte paramétré de la même façon.
- Le rapport est une opportunité pour vendre votre travail. Soyez-en fiers ! Si quelque chose n'est pas mentionné dans le rapport, il ne sera peut-être pas vu et donc pas noté. Mettez en avant ce que votre programme gère bien. Les limitations de votre programme sont également importantes à mentionner. Présentez-les plutôt comme des points à améliorer plutôt que comme des failles. Si vous savez comment vous auriez pu résoudre ces limitations si vous aviez eu plus de temps, mentionnez-le et décrivez vos idées en les présentant comme des pistes pour du travail futur.