

Rapport projet Traitement Automatique des Langues

Florent VOLLMER - Thibaut WITCZAK

05 Mai 2018

0.1 Introduction

Bob est un chatbot conspirationniste : Il croit à diverses théories du complot. Quelque soit le sujet de la conversation, il essaiera de l’orienter vers une de ces théories, afin d’exposer son point de vue à l’utilisateur. Il montre une attitude sceptique, voire condescendante, face aux réponses de l’utilisateur, qu’il juge “trop formatées”.

Il peut également paniquer si il pense que l’utilisateur est au centre d’une de ces théories (par exemple s’il croit être face à un reptilien), ou quitter la conversation si il s’ennuie (car on ne parle pas assez de sujet qui l’intéressent) ou encore si il trouve l’utilisateur trop “contrariant”, ne croyant pas à ses théories.

Nous l’avons codé anglophone, principalement pour éviter d’avoir affaire à une grammaire trop complexe, introduisant trop de pièges que nous aurions eu à éviter pour pouvoir espérer arriver à des résultats concluants.

0.2 Objectif du projet et motivations

Notre but en choisissant ce thème n’est pas de propager des théories du complots, mais au contraire de les tourner en dérision en les faisant défendre par un programme qui ne “comprend” pas vraiment ce qu’il dit. On a aussi pensé que ce thème serait amusant et donnerait lieu à des conversations atypiques avec l’utilisateur.

Cela a également amélioré nos compétences en Python et nous a appris le Latex. Nous n’avons en revanche pas eu beaucoup d’informations sur ce qui est considéré comme de “bonnes pratiques” en Python, et avons seulement une meilleure idée de comment faire du code fonctionnel dans ce langage.

Enfin, ce projet nous aura enfin permis de nous intéresser à un nouveau type de programme, en appliquant ce que nous avons vu en cours mais aussi en nous laissant une certaine liberté, aussi bien dans le thème de chatbot que dans son fonctionnement.

0.3 Fonctionnement du projet

Répartition du code dans les fichiers :

Main.py :

Sert d’interface entre l’utilisateur et Bob

Bob :

Regroupe toutes les fonctionnalités propres à Bob, telles que les modes 1, 2 et 3, ou encore la gestion des émotions.

LexField :

Regroupe tout ce qui touche à la gestion des champs lexicaux : reconnaître si un mot y appartient, fournir des réponses génériques... Les champs lexicaux sont regroupés en fichiers, contenant à la fois les mots, les groupes de mots, les réponses, les “influences” sur Bob et les liens avec les autres champs. Un mot peut faire partie de plusieurs champs lexicaux, mais on préfère généralement le mettre dans un minimum de champs lexicaux puis utiliser les liens de parenté.

Answer :

Consistant en une chaîne et un ID, a été créée pour rendre le code plus maintenable. En effet, quand on modifie un détail de formulation ou d’orthographe dans une réponse, pas besoin de la modifier partout : seul son ID est comparé avec celui d’autres réponses.

Fichiers texte :

Ces fichiers contiennent des listes de champ lexicaux. LexFields.txt est celui utilisé par le mode2, c’est le plus lourd et le plus générique. Les autres fichiers contiennent des mots-clés et des réponses qui ont un sens seulement dans certains contextes, identifiés dans le mode 3.

Comme demandé, le code a été organisé en 3 modes, correspondant aux méthodes ansBob1, ansBob2 et ansBob3.

Mode 1 :

Bob envoie des réponses comme “hmm...” ou “interesting...”, laissant penser qu’il prête attention à ce que l’utilisateur dit.

Mode 2 :

Bob se sert d’un tableau “subjects”, contenant une valeur réelle positive pour chaque sujet de conversation possible. La plus haute de ces valeurs correspond au sujet que Bob estime être le plus pertinent, en fonction des mots entrés par l’utilisateur dernièrement. Il pose ainsi des questions ou fait des remarques sur ces sujets afin d’alimenter la conversation, à la façon d’Elizia, à part que ses répliques ne visent pas à faire parler l’utilisateur de lui, mais de ce qu’il pense sur divers sujets.

Une autre différence avec Elizia est que les valeurs du tableau “subjects” ne sont pas remises à 0 à chaque réponse, mais sont multipliées par 0,6 et remises à 0 seulement si le résultat est en-dessous de 0,4. Cela fait qu’il restera une trace d’un sujet abordé pendant au moins 1 tour dans le tableau (et plus si on a insisté sur le sujet), ce qui permet de ne pas retomber immédiatement dans le mode 1 si l’utilisateur n’a pas entré de mot clé reconnu dans sa dernière réponse.

Enfin, certains des sujets ont des “parents”, qui sont incrémentés de $0,6 \times \text{value}$ quand le sujet de base est incrémenté de valeur. Cela ne suffit donc pas à donner la priorité au sujet parent car ce n’est pas ce qu’on recherche, mais en cumulant plusieurs sujets qui ont le même parent, le parent peut effectivement devenir le sujet le plus pertinent.

Mode 3 :

On vérifie un ensemble de critères sur le tableau “subjects”, mais aussi sur l’humeur de Bob et on fait des vérifications plus précises sur la dernière phrase entrée par l’utilisateur. On prend également en compte la dernière question posée par Bob, afin qu’il sache que l’utilisateur répondait à l’une de ses

questions. Il peut alors envoyer des phrases plus cohérentes avec le contexte.

Pour analyser au mieux la dernière phrase de l'utilisateur, on a codé les méthodes `checkStunned` et `checkYesNo` de Bob. Ces méthodes font ce que fait le mode 2 mais de façon plus précise et plus au cas par cas, en utilisant notamment des poids pour chaque mot ou groupe de mot. Ces fonctions sont appelées occasionnellement par le mode 3 suite à certaines questions de Bob :

```
Bob : Hi I'm Bob! What could we talk about?
You : money
Bob : Anyway, money is just a political thing, isn't it?
You : What do you mean?
Bob : Well, it's quite obvious to me. In your mind, what is the purpose of money?
You : trade?
Bob : That's what it should be. But in my mind, it has never really been so.
You : But money is usefull isn't it ?
Bob : Don't you think this society attaches too much importance to money?
You : Yes maybe, but it's still usefull sometimes don't you think?
Bob : You seem to hesitate...
You : I mean, no I don't think so.
Bob : Maybe you should think again about that.
```

Ici, Bob a commencé par une réponse de mode 2 car le sujet money l'intéresse mais les conditions n'étaient pas remplies pour entrer en mode 3. En revanche, "Anyway, money is just a political thing, isn't it?" Est une des questions de Bob dont le mode 3 va analyser la réponse. Ici il a remarqué que l'utilisateur avait l'air surpris, il a alors enchaîné sur une autre question, afin de préciser son propos. Cette question "What is the purpose of money?" est également traitée par le mode 3, mais cette fois en faisant appel à une méthode appelée "miniMode2", car elle fait elle-même appel au mode 2, mais en lui fournissant une liste de champs lexicaux contenant des mot-clés et des réponses autres que ceux par défaut.

Après avoir donné son avis à l'utilisateur sur sa réponse à l'utilité de l'argent, Bob est sorti du mode 3. Mais comme l'utilisateur a continué à parler d'argent, il lui posé une autre question de mode 2, qui en l'occurrence menait également au mode 3 ("Don't you think this society attaches too much importance to money?"). La réponse à cette question a été analysée par la méthode `checkYesNo`. Malgré la présence du mot "yes", Celle-ci a considéré que l'utilisateur n'était pas assez clair, la méthode "askYesNo" a donc été appelée pour lui demander d'être plus clair (Elle n'aurait pas été appelée une deuxième fois si l'utilisateur avait encore hésité, afin de ne pas bloquer la conversation). La deuxième réponse a été considérée comme un "non", Bob a alors montré sa désapprobation via la méthode "disapprove".

L'humeur de Bob est codée sur 3 entiers, qui sont son intérêt pour la conversation, son stress et sa sympathie pour l'utilisateur.

L'intérêt est initialisé avec une valeur de 0, est décrémenté à chaque fois que le mode 1 est employé, et est remis à 0 à chaque fois que les modes 2 ou 3 sont employés. Si l'intérêt de Bob atteint la valeur -5, il quitte la conversation comme ci-dessous :

```
Please choose the mode you want to use : 3
Bob : Hi I'm Bob! What could we talk about?
You : nothing
interest = -1 ; stress = 0 ; sympathy = 0
Bob : Tell me more.
You : no
interest = -2 ; stress = 0 ; sympathy = 0
Bob : Interesting...
You :
interest = -3 ; stress = 0 ; sympathy = 0
Bob : Hmm...
You :
interest = -4 ; stress = 0 ; sympathy = 0
Bob : Tell me more.
You :
interest = -5 ; stress = 0 ; sympathy = 0
Bob : Hm, it's getting late, I should leave.
* Bob left the conversation because he got bored *
```

Dans cet exemple, on n'est jamais sorti du mode 1, alors que le mode 3 était activé. Bob a donc vite perdu patience.

Le stress est initialisé à 0, et peut être augmenté via le mode 2 ou le mode 3, à cause de certains mots clés (les questions sur lui, et les agences de renseignement parce qu'il est persuadé d'être traqué par celles-ci) ou des réponses de l'utilisateur à certaines questions. Si Bob est trop stressé, il fuira la conversation comme dans les deux cas ci-dessous :

```
Bob : Hi I'm Bob! What could we talk about?
You : NSA
interest = 0 ; stress = 2 ; sympathy = 0
Bob : The weather is great, isn't it? Let's talk about weather!
You : What's your name ?
interest = 0 ; stress = 3 ; sympathy = 0
Bob : Haha, I'm not such an interesting guy. Let's talk about what's going on in this world!
You : FBI
interest = 0 ; stress = 5 ; sympathy = 0
Bob : Yeah hm... I think it's time for me to go to... the swimming pool... in order to... walk
my pony or something like that...
* Bob ran away from you, convinced you are from the NSA *
```

Ici Bob a d'abord été stressé par le mot NSA. C'est le mode 2 qui est intervenu, aussi bien pour monter l'attribut stress que pour fournir la réponse de Bob, qui cherche à détourner la conversation. Il s'est passé la même chose quand l'utilisateur lui a demandé son nom. Quand on a enchaîné sur le mot FBI, le mode 2 a encore incrémenté le stress de 2 points, ce qui a déclenché la fuite de Bob, qui a trouvé un prétexte douteux pour s'en aller.

```
Please choose the mode you want to use : 3
Bob : Hi I'm Bob! What could we talk about?
You : NSA
interest = 0 ; stress = 2 ; sympathy = 0
Bob : Haha yeah sure... did I ever told you about my pony? His name is Charlie.
You : I prefer reptiles
interest = 0 ; stress = 2 ; sympathy = 0
Bob : Do you think reptiles are exceptional animals?
You : Yes. What about you?
interest = 0 ; stress = 3 ; sympathy = 0
Bob : Wait... are you a reptilian?
You : yes
interest = 0 ; stress = 3 ; sympathy = 0
Bob : Oh no, please don't kill me! I won't tell anyone!
* Bob ran away from you, convinced you are a reptilian *
```

Dans cet autre exemple, Bob a également été stressé par les questions sur lui et par le mot NSA, mais cette fois le tableau "subjects" indiquait qu'on l'on parlait aussi de reptiles. Bob a donc vu un

lien avec les reptiliens, et a demandé à l'utilisateur si il en était un. Lorsque celui-ci a répondu par l'affirmative, Bob a également pris la fuite.

Enfin, il est possible de contrarier Bob suffisamment pour qu'il quitte de lui-même la conversation. En effet, il aime qu'on soit d'accord avec lui, et ne voudra plus vous parler si son attribut sympathie atteint la valeur -5 ou en dessous. Cette variable est initialisée à 0, et peut être incrémentée ou décrétementée selon les réponses de l'utilisateur. Il est techniquement possible de la faire baisser directement via le mode 2, mais dans le projet seul le mode 3 la modifie, parfois via le mode 2 appelé par la méthode miniMode2 (on est donc toujours en mode 3).

```
Bob : Hi I'm Bob! What could we talk about?
You : terrorism
interest = 0 ; stress = 0 ; sympathy = 0
Bob : In your opinion, are our governments fighting terrorism efficiently?
You : yes
interest = 0 ; stress = 0 ; sympathy = -1
Bob : I hope one day you will understand.
You : terrorism
interest = 0 ; stress = 0 ; sympathy = -1
Bob : Anyway, most of terrorist attacks are in fact inside jobs.
You :
interest = 0 ; stress = 0 ; sympathy = -1
Bob : Don't you see that's all a distraction?
You : no
interest = 0 ; stress = 0 ; sympathy = -2
Bob : Oh god, no offense but you've been completely brainwashed.
You : terrorism
interest = 0 ; stress = 0 ; sympathy = -2
Bob : In your mind, who is really responsible for terrorism?
You : terrorists
interest = 0 ; stress = 0 ; sympathy = -4
Bob : Haha, reality is never as simple as that.
You : money
interest = 0 ; stress = 0 ; sympathy = -4
Bob : Do you consider yourself to be a materialistic person?
You : yes
interest = 0 ; stress = 0 ; sympathy = -4
Bob : Anyway, money is just a political thing, isn't it?
You : no
interest = 0 ; stress = 0 ; sympathy = -5
Bob : Well this conversation was... interesting, but maybe you're too young
to plainly understand these subjects.
* Bob left the conversation because you have been too annoying to him *
```

Ici Bob n'a pas été content parce que l'utilisateur a toujours répondu une des "pires" réponses possibles à chaque question de Bob, aussi bien sur le terrorisme que sur l'argent. Bien que les variables affichées ici sont normalement masquées, L'utilisateur peut deviner s'il gagne la sympathie de Bob ou non en regardant les réponses fournies par le mode 3, plus précisément par "miniMode2" ("Haha, reality is never as simple as that") ou par les méthodes "Approve" et "Disapprove" ("Oh god, no offense but you've been completely brainwashed", "I hope one day you will understand"). Ces réponses sont ici toutes très condescendantes, mais peuvent être plus amicales quand il faut montrer que la sympathie de Bob augmente.

0.4 Limites et améliorations possibles du projet

Nous avons choisi de faire un chatbot axé sur les champs lexicaux, et non pas sur la grammaire et les constructions de phrases. Bob est donc incapable de reconnaître des choses trop compliquées, telles que des doubles négations. Il ne sait pas non plus construire ses propres phrases, et pioche

toujours parmi une liste de phrases toutes faites.

De plus, le nombre de théories et de champs lexicaux que Bob connaît est relativement limité, ainsi même le mode 2, censé être relativement robuste, ne trouve souvent rien de particulier à dire sur la phrase entrée par l'utilisateur, et on doit faire appel au mode 1. Une amélioration possible serait d'agrandir lesdits champs lexicaux et d'augmenter leur nombre en faisant plus de recherches sur les différentes théories.

On pourrait aussi générer les champs lexicaux automatiquement en analysant plusieurs articles sur différents sujets, et donner un "poids" à chaque mot ou groupe de mots pour chaque catégorie, correspondant à sa fréquence d'apparition dans ce sujet, à la manière de l'exercice sur la détection de spam fait en classe, et ainsi avoir une analyse à la fois plus fine et plus robuste.

Si l'on voulait mettre en application les améliorations précédemment citées, on pourrait rencontrer des problèmes de performances et il faudrait alors sans doute revoir la structure des fichiers et des données, afin d'avoir quelque chose de plus optimisé. Par exemple, on pourrait stocker tous les mots et groupes de mots par ordre alphabétique afin de vérifier plus rapidement s'ils existent dans au moins 1 champ lexical, puis d'accéder à une liste de nombres correspondant à leurs poids dans différents sujets. Dans le cas du mot "sugar" par exemple, on chercherait "sugar" par bijection dans une unique liste de mots, et une fois trouvé on incrémenterait la pertinence du sujet "alimentation" de 10, celle de "addiction" de 7, et celle de "IT" de 1 (pour "syntactic sugar"). On vérifierait également s'il fait partie d'un quelconque groupe de mots, pour lesquels il faudrait trouver une solution.

Il y a sans doute également des progrès à faire dans les relations entre les champs lexicaux. Pour cela il faudrait trouver un compromis entre la "réalité" analysée dans des articles de presse par exemple, et le caractère complotiste de Bob, qui va voir des liens là où le commun des mortels n'en voit pas.

Une autre limite de Bob est l'aspect un peu brouillon du mode 3, qui ne supporterait sans doute pas d'être complexifié beaucoup plus. Il faudrait donc trouver un moyen de rendre son code plus générique, et de mettre les variantes au cas par cas dans des fichiers .txt ou .csv par exemple, qui pourraient être bien plus nombreux et seraient lus par le mode 3, un peu à la manière des fichiers de champs lexicaux lus par le mode 2.

0.5 Contributions des membres au projet

Florent a essentiellement contribué au développement du code du chatbot. Ainsi, il aura davantage travaillé sur les modes 2 et 3 du programme.

Thibaut, quant à lui, a créé le mode 1 et a contribué au code au début du projet. Il s'est ensuite principalement occupé du rapport et de la maîtrise du latex, et a grandement participé au contenu des fichiers de champs lexicaux.