SAE S1.01
Projet "chatbot" - Partie 2

Le chatbot que nous avons développé dans la première partie est globalement en mesure de répondre aux questions de l'utilisateur dès lors que la question correspond à un format de question/réponse connu, que la réponse est dans sa base de réponses et qu'elle contient tous les mots non outil de la question. Cependant, il souffre d'un certain nombre de défauts/manquements auxquels nous nous proposons de remédier dans cette seconde partie.

1. Prise en compte des synonymes d'un mot et de ses différentes variantes

En l'état actuel, à la question « Qui a peint la Joconde ? », la réponse du chatbot est : « Je ne sais pas. » car bien que la réponse « La Joconde a été peinte par Léonard de Vinci. » soit présente dans la base des réponses, elle contient « peinte » et non « peint ». De même, à la question « Comment se nomment les habitants de Grenoble ? », la réponse est « Je ne sais pas. » bien que l'on trouve la réponse « Les habitants de Grenoble s'appellent les Grenoblois. » dans notre base de réponses, cette dernière utilisant le verbe « appeler » et non « nommer ». Par ailleurs, il pourrait être souhaitable de ne pas faire la différence entre les articles « le, la, les » pour que la question/réponse idéale : « Qui a inventé **le** téléphone ? **Le** téléphone a été inventé par Alexander Graham Bell. » puisse servir de forme modèle à « Qui a composé **la** Traviata ? ». Enfin, on peut imaginer pouvoir répondre à des questions comportant des fautes d'orthographe (« Quelle est la mo**n**aie officielle du Canada ? La mo**nn**aie officielle du Canada est le dollar canadien. »).

Pour toutes ces raisons nous proposons d'ajouter une nouvelle structure de données que l'on nommera **thésaurus** (même si elle va bien au-delà des relations de synonymie que l'on trouve habituellement dans un thésaurus) qui donnera pour différents mots, leur forme canonique, c'est-à-dire une forme unique qui permettra de d'établir le lien entre les différents synonymes ou variantes du mot. Le thésaurus sera pris en compte en remplaçant automatiquement (vous êtes guidés dans la suite) tous les mots par leur forme canonique (s'ils en ont une) dans nos différentes tables et index utilisés dans les étapes 1 et 2. Ce « thésaurus » sera sauvegardé dans le fichier `thesaurus.txt` et chargé en mémoire dès le lancement du chatbot. Chaque ligne de ce fichier (exemple ci-dessous) contiendra un mot suivi de « : », suivi de sa forme canonique.

```
appellent:nomme
nomment:nomme
s:se
la:le
l:le
les:le
monaie:monnaie
peinte:peint
…
```

a) *Créez votre propre fichier* `thesaurus.txt`.

b) *Développez les méthodes* `trierEntreesSorties`, `ajouterEntreeSortie`, `rechercherSortiePourEntree` *et le constructeur* `Thesaurus` *dans* **Thesaurus.java**

c) *Dans* **Utilitaire.java**, *modifiez (entêtes et contenu) des méthodes* `constructionIndexReponses`, `calculForme`, `constructionTableFormes`, `constructionIndexFormes`, `constructionReponsesCandidates`, `selectionReponsesCandidates` *pour qu'elles intègrent le thésaurus.*

d) *Modifiez* **Chatbot.java** *pour intégrer le thésaurus.*

Un exemple de trace que vous devriez pouvoir produire à ce stade :

```
> Comment se nomment les habitants de Grenoble ?
> Les habitants de Grenoble s'appellent les Grenoblois.
> Qui a peint la Joconde ?
> La Joconde a été peinte par Léonard de Vinci.
```

2. Réponses en contexte

Nous proposons de permettre la production de réponses en contexte, c'est-à-dire la possibilité d'utiliser le contexte thématique de la question précédente, pour permettre à l'utilisateur du chatbot de ne pas avoir à formuler une question complète reprenant le contexte thématique si ce contexte reste identique. Nous souhaitons pouvoir produire ce type de trace :

```
> Qui a peint La Joconde ?
> La Joconde a été peinte par Léonard de Vinci.
> En quelle année ?
> La Joconde a vraisemblablement été peinte entre 1503 et 1507.
> Où ?
> La Joconde aurait été peinte sur les bords du lac de Côme en Lombardie.
```

Dans cet objectif, nous allons écrire dans la classe `Chatbot` une méthode `repondreEnContexte` qui sera appelée à la place de `repondre` lorsque la question de l'utilisateur ne contient que des mots-outils.

a) Dans **Utilitaire.java**, *développez la méthode* `entierementInclus` *qui permettra de vérifier que la question ne contient que des mots-outils.*

b) Dans **Chatbot.java**, *développez la méthode* `repondreEnContexte`, *qui s'appuie sur la question précédente posée par l'utilisateur (en fait, la dernière question avec des mots non outils) pour construire les réponses candidates (étape 1) et sur la nouvelle question (ne contenant donc que des mots-outils) pour sélectionner parmi ces réponses candidates celles dont la forme est cohérente avec celle de la question (étape 2).*

c) *Dans* **Chatbot.java**, *organisez le traitement pour appeler* `repondre` *ou* `repondreEnContexte` *selon que la question contient ou pas d'autres mots que des mots-outils et faites en sorte d'enregistrer dans une variable la dernière question contenant un contexte thématique. Vérifiez le bon fonctionnement en produisant la trace ci-dessus.*

3. Intégration des nombres dans la forme des réponses

On aimerait aussi pouvoir produire ce type de trace :

```
> En quelle année le stéthoscope a-t-il été inventé ?
> Le stéthoscope a été inventé en 1816.
> Par qui ?
> René Laennec a inventé le stéthoscope.
> Où ?
> Le stéthoscope a été inventé en France.
> Quand ?
> Le stéthoscope a été inventé en 1816.
```

La difficulté est que dans ce cas, en l'état actuel de nos développements, la forme de la réponse qui répond à la question « Où ? » et celle qui répond à la question « Quand ? » sont identiques. Afin de faire la différence, nous proposons de détecter les chaînes de caractères correspondant à des nombres et de les remplacer par la chaîne « NUM » lors du calcul de forme. La chaîne NUM sera considérée comme un mot-outil (donc à ajouter au fichier des mots-outils) et permettra dans de distinguer les formes de réponse « ... a … le ... en … » et « ... a … le ... en NUM ».

a) Dans **Utilitaire.java**, *développez la méthode* `estUnNombre` *qui permet de vérifier si une chaîne de caractères représente un nombre.*

b) Dans **Utilitaire.java**, *modifiez les méthodes* `calculForme`, `constructionIndexFormes` *et* `selectionReponsesCandidates` *de façon appropriée pour pouvoir produire la trace ci-dessus.*

4. Ajout de connaissances

Lorsque le chatbot ne sait pas répondre, on souhaite donner la possibilité à l'utilisateur d'ajouter la connaissance de façon interactive sans avoir besoin d'éditer les fichiers `reponses.txt` ou `question-reponses.txt` directement. On peut distinguer plusieurs cas :

- Cette connaissance correspond à une nouvelle forme de question/réponse mais la réponse existe déjà. Cette nouvelle forme de question/réponse pourra être réutilisée pour des questions similaires comme ici :

```
> Qui a inventé le téléphone ?
> Alexander Graham Bell a inventé le téléphone.
> Qui a inventé le stéthoscope ?
> René Laennec a inventé le stéthoscope.
> A qui doit-on l'invention du téléphone ?
> Je ne sais pas.
> A qui doit-on l'invention du stéthoscope ?
> Je ne sais pas.
> Je vais te l'apprendre.
> Je t'écoute.
> René Laennec a inventé le stéthoscope.
> Très bien, c'est noté.
> A qui doit-on l'invention du téléphone ?
> Alexander Graham Bell a inventé le téléphone.
```

Remarque : `invention:inventé` est dans notre thésaurus.

- Cette connaissance correspond à une nouvelle réponse mais la forme de question/réponse existe déjà :

```
> Quelle est la capitale des Etats-Unis ?
> La capitale des Etats-Unis est Washington.
> Quelle est la capitale des Bahamas ?
> Je ne sais pas.
> Je vais te l'apprendre.
> Je t'écoute.
> La capitale des Bahamas est Nassau.
> Très bien c'est noté.
> Quelle est la capitale des Bahamas ?
> La capitale des Bahamas est Nassau.
```

- Cette connaissance correspond à la fois à une nouvelle forme de question/réponse et à une nouvelle réponse.

a) Dans **Utilitaire.java**, *développez la méthode* `reponseExiste` *qui permet de vérifier si la réponse existe déjà en utilisant l'index sur le contenu (*`indexThemes`*) et la méthode* `IntegrerNouvelleReponse` *qui permet de l'ajouter à l'index sur le contenu.*

b) *Dans* **Utilitaire.java**, *développez la méthode* `formeQuestionReponseExiste` *qui permet de vérifier si la forme question/réponse existe déjà en utilisant l'index sur la forme (*`indexFormes`*) et la méthode* `IntegrerNouvelleQuestionReponse` *qui permet de l'ajouter à l'index sur la forme..*

c) *Modifiez* **Chatbot.java** *pour gérer les saisies et les modifications des index et des fichiers concernés (pour la persistance) pour tenir compte de la connaissance ajoutée.*

5. Enrichissements ou spécialisations

Notre chatbot est maintenant suffisamment avancé pour nous permettre de chercher à l'utiliser véritablement. Nous proposons de l'enrichir en lui apprenant à répondre à de nouvelles questions de culture générale ou de le spécialiser sur un sujet précis. Ce sujet peut être par exemple le cours de R1.01, la FAQ d'un site web comme celui de la SNCF, ou tout autre sujet qui vous intéresse particulièrement. Il faudra tenir compte des particularités de notre chatbot qui impose de formuler des questions/réponses simples dont les mots de la question sont dans la réponse (en s'appuyant sur un thésaurus pour contourner cette contrainte dans certains cas et on pourrait aussi tolérer pour des questions un peu longue, l'absence d'un mot en modifiant la valeur du seuil passé à `maxOccurences`). Essayez de couvrir le plus de formulations possibles de questions. Faites-le utiliser par une personne extérieure à votre binôme pour découvrir de nouvelles réponses ou formes de question à ajouter. Vous pouvez aussi si vous le souhaitez, ajoutez des « méta » questions concernant le fonctionnement du chatbot lui-même comme « Quelles questions puis-je te poser ? », « A quelles questions peux-tu répondre ? », « Comment dois-je formuler mes questions ? », etc. L'objectif est de donner l'illusion à l'utilisateur que le chatbot comprend véritablement ses demandes comme un humain le ferait.

6. Quelques mots sur les limites de notre chatbot et des chatbots en général

Les principes de fonctionnement de notre chatbot impliquent certaines limites. Nous imposons par exemple que les mots de la question soient présents dans la réponse. Même si notre thésaurus permet de lever partiellement cette contrainte, encore faut-il le remplir. Nous nous appuyons aussi sur la distinction entre les mots-outils et les mots qui ne le sont pas ce qui induit inévitablement des problèmes lorsque le mot-outil correspond au thème de la question (*« Qui a écrit « Ça » ?»*). Quant à notre prise en compte du contexte, elle se limite à la question précédente et ne s'effectue que lorsque notre question n'est composée que de mots-outils. Enfin et surtout, nos réponses se limitent à des phrases simples que notre chatbot choisit directement dans une base de réponses existantes. C'est une différence importante avec ChatGPT qui, bien que les éléments de réponse lui soient aussi donnés préalablement, génère des réponses (on parle d'IA générative) construites par choix successifs de mots à ajouter à la réponse sachant : la question, le contexte de la discussion et les mots présents dans le début de la réponse généré.

Nous avons néanmoins traité des problèmes inhérents aux chatbots : traitement des synonymes/variantes, prise en compte du contexte, apprentissage à partir d'exemples de question/réponses idéales, choix d'une alternative dans un grand ensemble de réponses/mots possibles dans des délais acceptables. Bien que ChatGPT utilise des techniques plus élaborées (basées sur les réseaux de neurones) pour faire face à ces problèmes et que la qualité des réponses puisse faire illusion, il ne comprend pas non plus véritablement les questions posées comme un humain le ferait (il n'a pas de représentation du monde autres que les chaînes de caractères qu'il manipule). Nous vous invitons à lire les articles Wikipédia décrivant le test de Turing (https://fr.wikipedia.org/wiki/Test_de_Turing) et l'expérience de la chambre chinoise (https://fr.wikipedia.org/wiki/Chambre_chinoise) qui posent la question de « l'intelligence » d'un chatbot.

7. Evaluation

.

7.1 Dossier à rendre

A l'issue de ce projet vous devez rendre :

- un rapport (format pdf) de 4 à 5 pages **rédigé en anglais** comportant :
  - une introduction présentant le sujet
  - un point sur ce que vous avez réalisé (ce que vous avez fait, ce que vous n'avez pas fait)
  - une comparaison de la **complexité asymptotique dans le pire des cas** de l'approche dite naïve (**sans index**) avec notre approche (**avec index**) pour **l'étape 1 de construction des réponses candidates**. On cherchera dans les 2 cas, une formule exprimant **le nombre de comparaisons de mots ou d'identifiants** de réponses en fonction du nombre de réponses dans la base (`nr`), du nombre de mots non outils distincts dans toutes les réponses (`nmv`), du nombre de mots non outils dans les réponses (`nmr`), du nombre de mots non outils dans les questions (`nmq`). De plus, afin de simplifier les calculs, on supposera une répartition équilibrée des mots dans les réponses de telle sorte que le nombre de réponses contenant un mot (`nrpm`) soit constant et égal à `nr*nmr/nmv`. Appliquez les formules trouvées pour calculer le facteur d'accélération lié à l'utilisation d'un index dans le cas où `nr=100000, nmv=10000, nmr=5, nmq=3`.
- vos codes sources et vos fichiers `reponses.txt`, `questions-reponses.txt`, `mots-outils.txt`, `thesaurus.txt`

Il suffira de placer tout cela dans votre dossier ***projet-sae-s1-01*** et de taper la commande ***fin-sae-s1-01.***

7.2 Présentation Orale

Lors de la dernière séance vous ferez une présentation orale de votre projet (5-10 minutes). Prévoyez une suite de questions posées à votre chatbot montrant l'étendue de vos développements. Votre enseignant testera ensuite par quelques questions votre maîtrise du projet. Vous rendrez votre projet à la fin de votre présentation orale.