

FICHE : PoC & CHOIX TECHNOLOGIQUES

=> Après avoir rempli cette fiche : convertir ce document en PDF et le téléverser dans votre dépôt Github dans un répertoire nommé 'doc'.

COMPTE RENDU DE LA VEILLE TECHNOLOGIQUE PoC

QUESTIONS DE RECHERCHE SUR LA PoC

Quel est l'opération la plus risquée de votre projet, l'élément qui risque de ne pas fonctionner ? Quelle est l'interaction entre deux technologies ? Quel est le problème technique à résoudre ?

- J'ai plutôt confiance sur l'affichage des réponses alors je pense que le seul problème sera de faire comprendre les questions que je pose à mon bot. Si je n'arrive pas à faire marcher les bibliothèques que je veux utiliser comme "train_chatbot.py" ou "chatgui.py" je devrais sans doute simplifier mon projet avec des questions prédéfinies ou alors trouver une autre solution.

Quelle est la question que vous vous posez et que vous demandez à internet de répondre ?

"How to code a chatbot with Python"

Il existe tant de façons de fonctionnement pour un chatbot qu'il en devient difficile de trouver la meilleure tout en la gardant réalisable dans le temps imparti.

POC = PREUVE DE CONCEPT

Quel genre de preuve de concept minimale pourrait valider que le problème n'existe pas ou qu'une solution a été trouvée ? Décrivez chaque élément du code requis.

Malheureusement je ne pense pas pouvoir régler ces problèmes de façon définitive. Même les meilleurs bot au monde ont des difficultés pour répondre à tous les types de questions, surtout quand elle devient très poussée.

LES MARQUE-PAGES IDENTIFIÉS LORS DE VOS RECHERCHES

Article pour créer un chatbot :

<https://www.upgrad.com/blog/how-to-make-chatbot-in-python/>

<https://towardsdatascience.com/how-to-create-a-chatbot-with-python-deep-learning-in-less-than-an-hour-56a063bdfc44>

Vidéo pour créer un chatbot :

<https://www.youtube.com/watch?v=1lwddP0KUEg>

<https://www.youtube.com/watch?v=tSjR7bk1Y9U>

Python et librairies :

<https://www.python.org>

<https://pypi.org/project/ChatterBot/>

<https://pythonhosted.org/chatbot/documentation.html>

<https://chatterbot.readthedocs.io/en/stable/>

<https://github.com/gunthercox/ChatterBot/tree/master/examples>

LES PREUVES DE CONCEPT

Pour chaque preuve de concept réalisée : identifier le but de la preuve de concept (ce qu'elle vérifie), le lien vers le sous-répertoire de votre dépôt GitHub qui contient le code de la preuve de concept ainsi que les résultats de votre expérimentation, puis, finalement, vos conclusions.

Au moins une preuve de concept doit être documentée et réalisée.

PREMIÈRE POC RÉALISÉE

Preuve : Première réponse du bot

URL Github : <https://github.com/cegepmatane/projet-specialise-2022-JolanThomassin>

EXPLIQUEZ VOTRE POC

Décrivez la POC en détail.

J'ai tout d'abord installé une bibliothèque en Python qui s'appelle [chatterbot](#). Chatterbot est une bibliothèque qui facilite la génération de réponses automatisées suite à une demande utilisateur. Elle utilise un ensemble d'algorithmes d'apprentissage automatique pour produire des réponses.

Néanmoins, je pense que je serais satisfait si le bot parvient à répondre à des questions simple comme par exemple celle-ci :

Bonjour / Comment vas-tu ? / Salut / Hey / ...

Quel est ton nom ? / Qui es tu ? / Que peux-tu faire ? / ...

Donne moi la météo

Donne moi le lien de la musique "Michael Jackson beat it"

Que PROUVE la Poc ?

Si le bot arrive à répondre à des questions basiques, alors il y a de forte chance qu'il puisse répondre à des questions plus complexes. Comme envoyer des liens ou des images à l'utilisateur, ou utiliser des recherches internet pour répondre à des questions, comme par exemple pour envoyer la météo sur demande.

Si le bot est capable d'apprendre de ces discussions avec l'utilisateur alors au fur et à mesure du temps il sera de plus en performant.

Et le bot arrive très bien à faire ces deux choses-là.

Que reste-t-il à prouver ?

Il reste à prouver que le bot puisse avoir des réelles discussions avec l'utilisateur en comprenant ce qu'il dit sans trop s'égarer.

Quels sont vos résultats de la Poc ?

J'ai réussi à avoir un bot qui répond à des questions simples comme : "Bonjour", "comment ça va ?", "Que fais-tu ?".

Et le bot a appris certaines de mes phrases qu'il cite malheureusement bien souvent hors contexte, et je ne sais pas si les fois où le contexte est approprié est un coup de chance ou non. Mais, je sens quand même une amélioration au fil du temps et des messages.

Pour l'instant j'ai simplement discuté avec lui via l'invite de commande Windows pour lui apprendre du vocabulaire, mais je pense que le faire discuter avec une autre IA qui ne possède pas de notion de self-learning peut être bénéfique. Le problème de cette technique serait que le bot serait sans doute amené à répéter les mêmes choses en boucle. Mais le problème d'avoir deux IA en self-learning serait qu'un être humain ne puisse plus les comprendre car elles se seraient créées leurs propres langages.

PREMIÈRE TECHNOLOGIE SÉLECTIONNÉE (LA NOUVELLE)

Technologie : Librairie Chatterbot

URL : <https://chatterbot.readthedocs.io/en/stable/#>

JUSTIFIER VOTRE CHOIX TECHNOLOGIQUE POUR CETTE TECHNOLOGIE

C'est la façon la plus efficace que j'ai trouvé pour créer un chatbot en python sans avoir une API ou une autre bibliothèque qui me donne un chatbot déjà fonctionnel

DEUXIÈME TECHNOLOGIE SÉLECTIONNÉE (LA CONNUE)

Technologie : Python

URL : <https://www.python.org>

JUSTIFIER VOTRE CHOIX TECHNOLOGIQUE POUR CETTE TECHNOLOGIE

Expliquer à l'aide d'une argumentation rationnelle votre choix technologique. Établir votre justification à l'aide de liens avec les fonctionnalités, contraintes et risques de votre projet. Un tableau comparatif permettant de synthétiser votre réflexion pourrait être un apport judicieux à vos explications.

J'ai choisi le Python car ce langage est réputé pour être le meilleur langage pour l'intelligence artificielle. C'est donc également le langage où l'on retrouve le plus de documentation et librairie sur l'IA. Python est un langage simple à prendre en main et il fonctionne rapidement.

Étant le langage que je maîtrise le plus il me permettra donc d'économiser du temps que j'aurais perdu sur l'apprentissage d'autres langues.

Python est très doué pour les algorithmes, il correspond donc très bien à l'IA qui se base principalement sur des algorithmes.