Diploma Programme
Programme du diplôme
Programa del Diploma

Informatique
Étude de cas : Le chatbot parfait

A utiliser en mai et novembre 2025

Instructions destinées aux candidats

- Ce livret d'étude de cas est indispensable pour l'épreuve 3 du niveau supérieur.

5525 – 6402

8 pages

Scénario

La compagnie d'assurance *RAKT* a récemment mis en place un chatbot (agent de dialogue) utilisant un modèle linguistique afin de répondre aux questions des client·es. Toutefois, les performances du chatbot sont mauvaises et nombre de client·es ne sont pas satisfait·es des réponses reçues.

La compagnie vous a recruté·e en tant qu'étudiant·e stagiaire pour étudier le problème et formuler des recommandations pour améliorer le chatbot. Votre maître de stage a déjà fait l'analyse des plaintes des client·es et identifié la cause fondamentale du problème. Elle souhaite que vous examiniez six aspects principaux susceptibles d'améliorer les performances du chatbot.

Problèmes à résoudre

1. **Latence** : le temps de réponse du chatbot est long, ce qui nuit à l'expérience client.
2. **Nuances linguistiques** : le modèle de langage de l'agent a du mal à répondre de manière appropriée aux formulations ambigües.
3. **Architecture** : l'architecture du chatbot est trop rudimentaire et incapable de gérer un langage complexe.
4. ***Jeu de données*** : le jeu de données d'entraînement du chatbot n'est pas assez diversifié. Il en résulte une mauvaise compréhension des questions occasionnant de mauvaises réponses.
5. **Puissance de traitement** : les capacités de traitement du système informatique constituent un facteur limitant.
6. **Enjeux éthiques** : le chatbot ne donne pas toujours de conseils appropriés et a tendance à divulguer les données personnelles présentes dans le jeu de données d'entraînement.

Latence

Les client·es de *RAKT* attendent du chatbot qu'il réponde rapidement et correctement à leurs questions. Un modèle complexe de *traitement du langage naturel* a été développé pour reconnaître les nuances dans la communication humaine. Cependant, ce modèle a augmenté le temps de réponse du chatbot, en particulier lorsque le volume de questions est élevé.

L'intelligence artificielle (IA) conversationnelle utilise un vaste réseau de modèles d'apprentissage automatique pour obtenir une réponse, chacun d'entre eux constituant un petit morceau du puzzle. Un modèle peut par exemple prendre en compte l'emplacement de la personne utilisatrice, tandis qu'un autre peut analyser l'historique des messages des utilisateurs et utilisatrices, y compris les retours d'informations sur des réponses similaires. Chaque modèle affine en principe la réponse mais au détriment de l'augmentation de la latence du système.

La latence est produite par un algorithme de décision appelé « chemin critique », qui est la séquence de modèles d'apprentissage automatique la plus courte et la plus efficace pour qu'à partir du message de la personne utilisatrice, le chatbot apporte une réponse. On sait que les modifications apportées à un modèle peuvent avoir une incidence sur l'ensemble d'un réseau d'apprentissage automatique s'il présente des dépendances.

Un moyen de surmonter les dépendances est de transformer du texte non structuré en données informatiques exploitables. La *compréhension du langage naturel* (en anglais *NLU*, acronyme de *natural language understanding*) est un pipeline créé à l'aide de nombreux modèles d'apprentissage automatique effectuant diverses fonctions dans le but d'améliorer la compréhension par le chatbot de l'entrée utilisateur. Elle permet d'identifier et de filtrer les modèles inutiles, réduisant ainsi la latence.

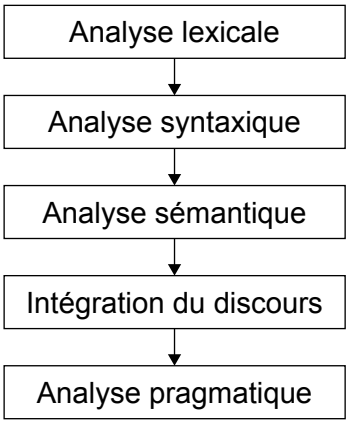
L'entraînement du chatbot contribue également à améliorer ses capacités à comprendre le texte. Pour réduire le temps de réponse, le jeu de données d'entraînement doit être volumineux, exact, classé, lisible, spécifique à un domaine et pertinent.

Nuances linguistiques

Les modèles de langage des chatbots gagnent à incorporer des traits plus humains, comme la capacité à détecter les émotions, le ton et le contexte, et à y répondre. L'amélioration du modèle permet au chatbot de générer des réponses plus personnalisées et plus adaptées aux besoins individuels de la clientèle, conduisant à une meilleure expérience globale.

Le traitement du langage naturel suit cinq étapes pour assurer la personnalisation des réponses d'un chatbot (voir **figure 1**).

Figure 1 : Les cinq étapes du traitement du langage naturel

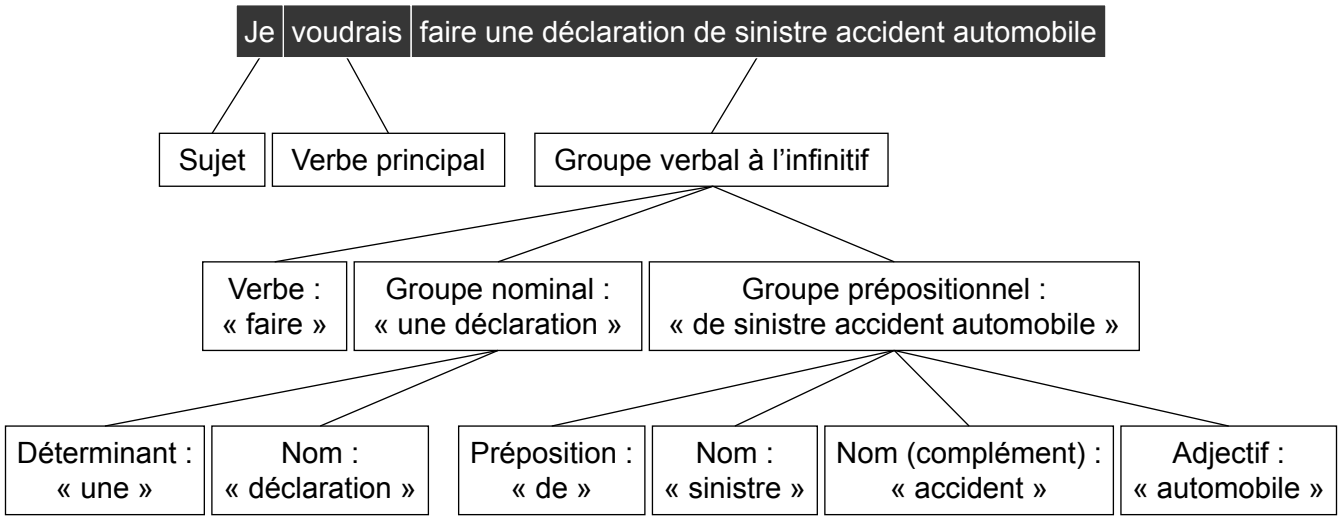


Les analyses lexicale et syntaxique se concentrent sur les aspects structuraux du langage, notamment le découpage d'un texte en mots, l'identification des catégories grammaticales et l'analyse du rapport entre mots et expressions. L'analyse sémantique, l'intégration du discours et l'analyse pragmatique portent essentiellement sur la signification du langage allant au-delà de la structure superficielle.

Supposons qu'un·e client·e saisisse la phrase suivante dans le chatbot : « Je voudrais faire une déclaration de sinistre accident automobile. » Le traitement du langage naturel en cinq étapes interprétera cette phrase comme suit :

1. **Analyse lexicale** : décomposition du texte d'entrée en mots et phrases, par exemple, [« Je », « voudrais », « faire », « une », « déclaration », « de », « sinistre », « accident », « automobile »].
2. **Analyse syntaxique** : interprétation de la grammaire et de la structure de la phrase, par exemple, détermination que « Je » est le sujet de la phrase, « voudrais » le verbe et « déclaration » l'objet (voir **figure 2**).
3. **Analyse sémantique** : analyse du sens de la phrase et de chacun de ses mots, par exemple, que la phrase veut dire que le client ou la cliente souhaite déclarer un sinistre concernant un accident automobile.
4. **Intégration du discours** : intégration du sens de la phrase dans le contexte de la conversation, par exemple, comprendre que l'utilisateur ou utilisatrice est un·e client·e qui veut se renseigner sur la déclaration d'un sinistre.
5. **Analyse pragmatique** : analyse du contexte social, juridique et culturel de la phrase, par exemple, comprendre que le client ou la cliente est un conducteur ou une conductrice qui a eu un accident et qui pourrait être blessé·e ou dont le véhicule est endommagé.

Figure 2 : Analyse syntaxique de la grammaire et de la structure de la phrase par le chatbot



Architecture

Le moteur de traitement du langage naturel est un composant essentiel de l'architecture d'un chatbot. Il utilise des algorithmes d'apprentissage automatique pour déterminer l'intention de l'utilisateur ou de l'utilisatrice et y faire correspondre les actions du logiciel.

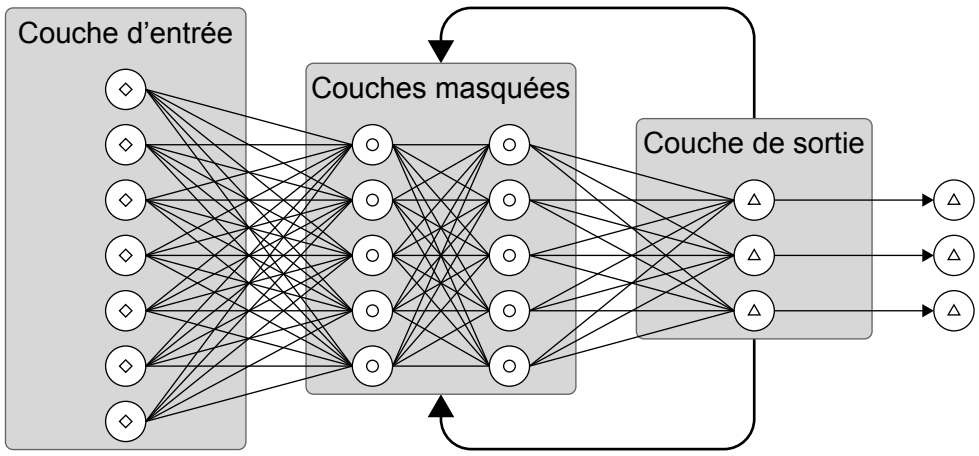
Il existe deux types de modèles d'*apprentissage profond* couramment utilisés dans le cadre de problème de traitement du langage naturel : le *réseau de neurones récurrent* (en anglais *RNN*, acronyme de *recurrent neural network*) et le *réseau de neurones transformeur* (ou transformeur).

Réseaux de neurones récurrents (RNN)

Le chatbot de la compagnie d'assurance utilise un RNN, un réseau de neurones artificiels conçu pour traiter les données séquentielles.

Un RNN comporte trois couches. La couche d'entrée traite la séquence de données en introduisant chacun des éléments de la séquence à tour de rôle dans le réseau. Les couches masquées gardent en mémoire les mots préalablement traités. La couche de sortie produit le résultat final, qui peut être une prédiction ou une classification (voir **figure 3**).

Figure 3 : La couche d'entrée, les couches masquées et la couche de sortie d'un réseau de neurones récurrent (RNN)



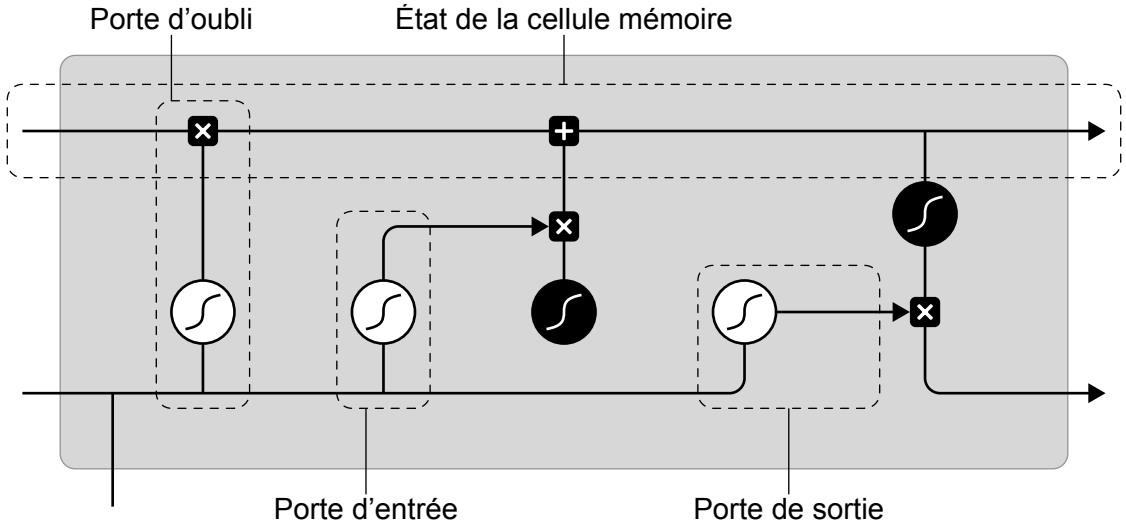
Un RNN peut traiter des données séquentielles de taille variable en conservant une mémoire contenant les entrées précédentes sous forme d'états cachés.

Tous les RNN ont besoin d'être entraînés à l'aide d'un jeu de données prétraitées approprié et divisé en données d'entraînement, de validation et de test. Les hyperparamètres, notamment le taux d'apprentissage et le nombre de couches masquées, sont définis. L'*optimisation des hyperparamètres* consiste à déterminer les valeurs optimales des paramètres.

Un RNN module la *pondération* des couches d'entrée et de sortie et celle des couches masquées lors de la phase d'entraînement au moyen de la *rétropropagation dans le temps* (en anglais *BPTT*, acronyme de *backpropragation through time*). Cette technique calcule les gradients de la *fonction de perte* pour chacun des poids à chaque étape, et ajuste la pondération du modèle de manière appropriée. Néanmoins, l'entraînement des RNN à l'aide de la BPTT peut conduire au problème de *disparition du gradient*, qui rend difficile l'apprentissage des dépendances à long terme dans les données.

Un réseau *de neurones récurrents à mémoire court terme et long terme* (en anglais *LSTM*, acronyme de *long short-term memory*) est un type de RNN conçu pour répondre au problème de disparition du gradient. Il utilise pour ce faire un système de trois portes qui contrôle le flux d'informations. L'*état de la cellule mémoire*, la porte d'entrée, la porte d'oubli et la porte de sortie permettent aux LSTM de retenir ou d'oublier sélectivement les informations au fil du temps (voir **figure 4**).

Figure 4 : Unité LSTM



Transformeurs

Un modèle de type transformeur constitue une alternative au réseau LSTM. Son originalité réside dans l'application d'un *mécanisme d'auto-attention*. Celui-ci exprime les relations entre les mots de la séquence d'entrée en calculant la pondération d'attention pour chacun des mots selon sa ressemblance aux autres mots de la séquence.

GPT-3 (acronyme de Generative Pretrained Transformer 3), créé par Open AI, est un exemple de *grand modèle de langage* (en anglais *LLM*, acronyme de *large language model*) utilisant un mécanisme d'auto-attention.

Jeux de données

Il semblerait que le jeu de données du chatbot soit biaisé et de mauvaise qualité. Les jeux de données doivent être en rapport avec le domaine dans lequel le chatbot évolue. Il peut s'agir de données concernant les déclarations de sinistre, le service à la clientèle, les polices d'assurance, ainsi que des données médicales. On peut créer des jeux de données à partir de données réelles et/ou de *données synthétiques*.

Il est important de s'assurer que les jeux de données sont objectifs et diversifiés afin d'éviter de perpétuer tout *biais* existant du secteur.

Il existe plusieurs types de biais présents dans les jeux de données. Ceux-ci peuvent avoir une incidence sur l'exactitude et l'équité des modèles de traitement du langage naturel, notamment les biais *de confirmation*, *historique*, *d'étiquetage*, *linguistique*, *d'échantillonnage* et *de sélection*.

- **Biais de confirmation** : Ce type de biais a lieu lorsque le jeu de données est biaisé en faveur d'un point de vue particulier, par exemple des données d'entraînement qui incluent uniquement des questions posées par la clientèle sur certains types de police.
- **Biais historique** : Cette forme de biais se produit lorsque les données d'entraînement ne tiennent pas compte des modifications dans le temps. Par exemple, si le modèle de traitement du langage naturel a été entraîné avec des données datant de plusieurs années, il ne sera peut-être pas en mesure de prédire les questions récentes de manière fiable.
- **Biais d'étiquetage** : Cette forme de biais a lieu lorsque les étiquettes appliquées aux données sont subjectives, inexactes ou incomplètes. Par exemple, si les étiquettes attribuées aux questions posées par la clientèle sont trop génériques, le modèle ne sera éventuellement pas capable de prédire avec exactitude l'intention des client·es.
- **Biais linguistique** : Ce type de biais se produit lorsque le jeu de données est biaisé en faveur de certaines spécificités linguistiques comme un dialecte ou un certain vocabulaire. Par exemple, si le jeu de données est élaboré à partir d'un langage officiel écrit, le modèle ne pourra peut-être pas interpréter correctement le langage familier.
- **Biais d'échantillonnage** : Cette forme de biais a lieu lorsque le jeu de données d'entraînement n'est pas représentatif de l'ensemble de la population, notamment s'il contient des questions provenant uniquement d'un certain groupe démographique.
- **Biais de sélection** : Ce type de biais se produit lorsque les données d'entraînement ne sont pas sélectionnées au hasard mais au contraire choisies en fonction d'un critère particulier. Un modèle de langage entraîné sur des données suggérant que certains groupes démographiques ont plus tendance à déclarer un sinistre est biaisé en faveur des personnes qui entrent dans cette catégorie.

Puissance de traitement

Les étapes de préparation d'un modèle d'apprentissage automatique sont :

1. le *prétraitement* des données d'entrée ;
2. l'entraînement du modèle ;
3. le déploiement du modèle.

Prétraitement des données d'entrée

Le prétraitement est la première manipulation des données brutes. Il les rend compréhensibles aux algorithmes d'apprentissage automatique. Il comprend le nettoyage, la sélection, la transformation et la réduction des données afin d'en améliorer leur qualité et leur exactitude.

Actuellement, le modèle de langage du chatbot utilise l'algorithme des *sacs de mots* pour comprendre l'entrée du client ou de la cliente et d'en extraire les informations pertinentes. Pour ce faire, il applique une représentation vectorielle du texte basée sur la fréquence des mots. Cet algorithme peut solliciter de manière intensive le processeur lorsqu'il traite des jeux de données volumineux.

Entraînement du modèle

L'entraînement du modèle d'apprentissage automatique est la tâche qui demande le plus de calculs, étant donné que d'énormes quantités de données textuelles sont utilisées. Le processus d'entraînement peut prendre plusieurs semaines, voire plusieurs mois et nécessite de matériel puissant pour accélérer les calculs, comme des clusters de *processeurs graphiques* (en anglais *GPU*, acronyme de *graphical processing unit*) ou d'*unités de traitement de tenseur* (en anglais *TPU*, acronyme de *tensor processing unit*).

Déploiement du modèle

Une fois entraîné, un LLM peut être déployé sur diverses plate-formes matérielles. Cependant, pour obtenir les meilleures performances possibles, il est recommandé d'utiliser un matériel spécialisé, comme des TPU. En plus du matériel requis, l'exécution des LLM nécessite également d'espaces de stockage et de mémoire importants afin de stocker le modèle et ses données associées.

Enjeux éthiques

L'équipe de *RAKT* a identifié plusieurs enjeux éthiques qu'elle devra examiner. Ceux-ci incluent :

- confidentialité et sécurité des données – protéger les données utilisateur d'un usage abusif ;
- biais et équité – éviter les réponses discriminatoires ;
- responsabilité – décider qui est responsable des conseils donnés ;
- transparence – expliquer clairement le processus décisionnel ;
- désinformation et manipulation – éviter la propagation de fausses informations.

Défis rencontrés

Les défis suivants sont à étudier de manière plus approfondie :

- évaluation des méthodes de réduction de la latence ;
- compréhension des cinq étapes du traitement du langage naturel ;
- enquête pour déterminer si un transformeur peut améliorer les performances d'un RNN dans le cadre du traitement du langage naturel ;
- compréhension de ce qui est nécessaire à la création d'un jeu de données de grande taille, non biaisé et de haute qualité ;
- analyse de la puissance de traitement exigée par un modèle de traitement du langage naturel ;
- réflexion sur l'éthique de l'utilisation d'un chatbot pour donner des conseils.

Les candidats ne sont pas tenus de comprendre les termes linguistiques, par exemple le modèle sujet-verbe-objet (langue SVO).

Terminologie supplémentaire

Apprentissage profond (*deep learning*)
Biais (*biases*)
- de confirmation (*confirmation*)
- d'échantillonnage (*sampling*)
- d'étiquetage (*labelling*)
- historique (*historical*)
- linguistique (*linguistic*)
- de sélection (*selection*)

Compréhension du langage naturel (NLU, *natural language understanding*)
Disparition du gradient (*vanishing gradient*)
Données synthétiques (*synthetic data*)
État de la cellule mémoire (*memory cell state*)
Fonction de perte (*loss function*)
Grand modèle de langage (LLM, *large language model*)
Jeu de données (*dataset*)
Latence (*latency*)
Mécanisme d'auto-attention (*self-attention mechanism*)
Neurones récurrents à mémoire court terme et long terme (LSTM, *long short-term memory*)
Optimisation des hyperparamètres (*hyperparameter tuning*)
Pondération (*weights*)
Prétraitement (*pre-processing*)
Processeur graphique (GPU, *graphical processing unit*)
Réseau de neurones récurrent (RNN, *recurrent neural network*)
Réseau de neurones transformeur (ou transformeur, *transformer NN*)
Rétropropagation dans le temps (BPTT, *backpropagation through time*)
Sac de mots (*bag-of-words*)
Traitement du langage naturel (*natural language processing*)
- Analyse lexicale (*lexical analysis*)
- Analyse pragmatique (*pragmatic analysis*)
- Analyse sémantique (*semantic analysis*)
- Analyse syntaxique (*syntactical analysis (parsing)*)
- Intégration du discours (*discourse integration*)

Unité de traitement de tenseur (TPU, *tensor processing unit*)

Certains produits, sociétés et individus mentionnés dans cette étude de cas sont fictifs. Toute ressemblance avec des entités réelles ne saurait être que fortuite.

Références :

Figure 3 Avec la permission de Vinod Sharma. https://vinodsblog.com/2019/01/07/deep-learning-introduction-to-recurrent-neural-networks/.

Figure 4 Rengasamy, D., Jafari, M., Rothwell, B., Chen, X. et Figueredo, G. P., 2020. Deep Learning with Dynamically Weighted Loss Function for Sensor-Based Prognostics and Health Management. *Sensors*, 20(723), pages 1–21. doi:10.3390/s20030723. Article en libre accès. Source adaptée.