Méthodologie de pilotage des performances du modèle en production

Kévin Duranty | OpenClassrooms - P10

CONTENTS OF THIS TEMPLATE

Fly Me est une agence qui propose des voyages clé en main pour les particuliers ou les professionnels.

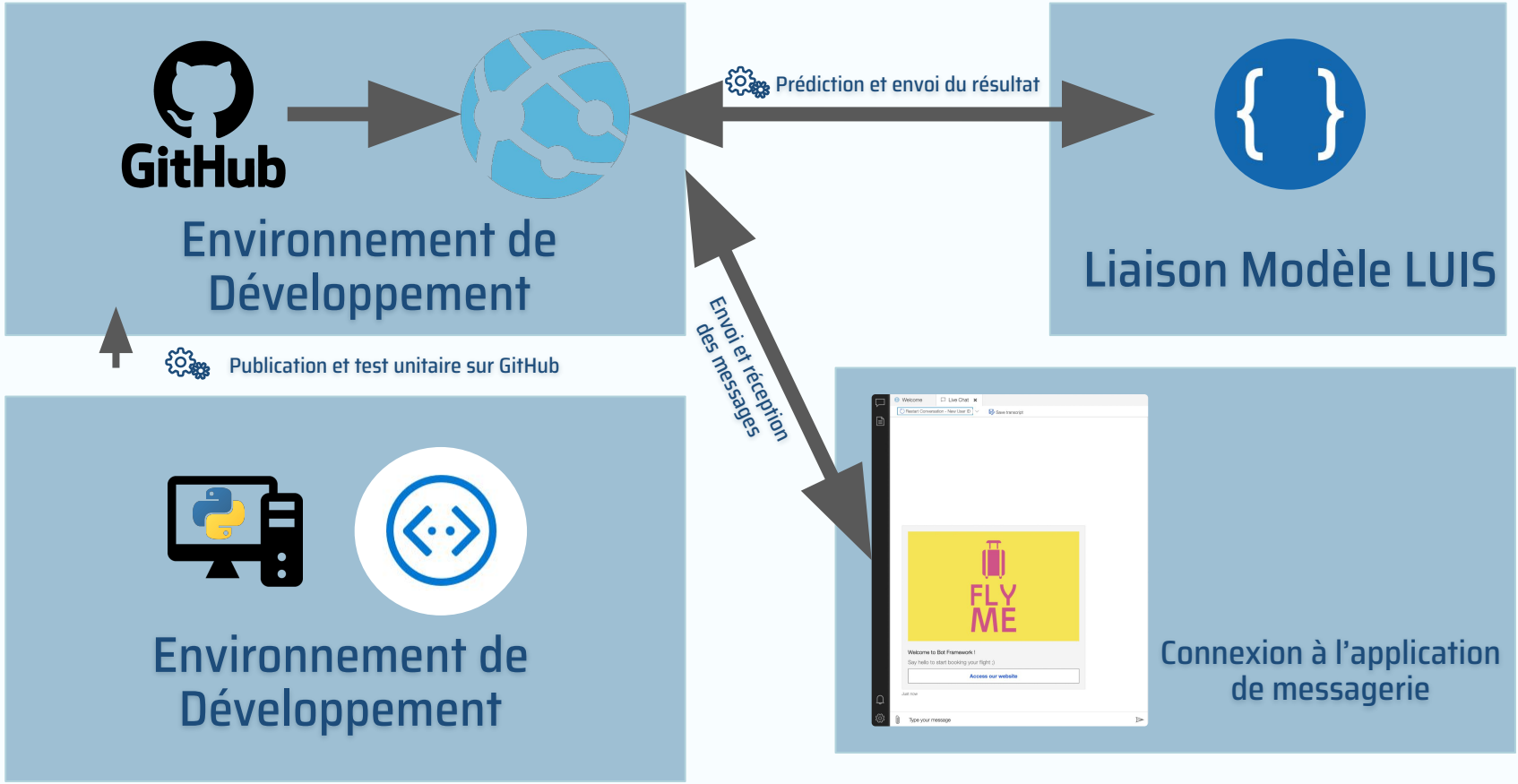
Fly Me a lancé un projet ambitieux de développement d'un chatbot pour aider les utilisateurs à choisir une offre de voyage.

La première étape de ce projet est de construire un MVP qui aidera les employés de Fly Me à réserver facilement un billet d'avion pour leurs vacances.

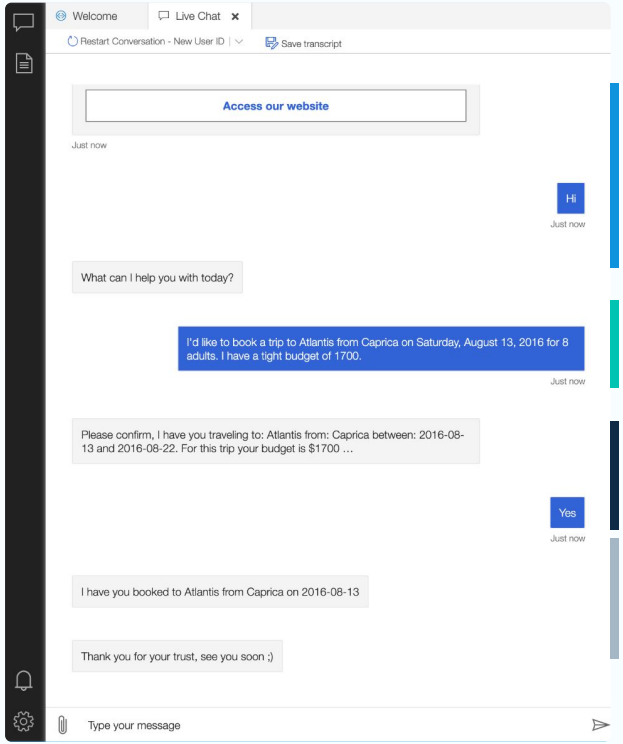
Comme ce projet est itératif, nous avons limité les fonctionnalités de la V1 du chatbot. La V1 devra pouvoir identifier dans la demande de l'utilisateur les cinq éléments suivants :

- Ville de départ
- Ville de destination
- Date aller souhaitée du vol
- Date retour souhaitée du vol
- Budget maximum pour le prix total des billets

1. Principe de fonctionnement



1. Principe de fonctionnement



STEP 1 : Démarrage de l'application de messagerie

STEP 2 : Détection des entité du premier message
(Demandes individuelles des entités non détectées)

STEP 3 : Demande de confirmation
(Renvoi au début de conversation si réponse négative)

STEP4 : Fin de conversation avec récapitulatifs des informations envoyées

1. Principe de fonctionnement

STEP 1 : Dans un premier temps l'application se lance avec un message d'accueil invitant l'utilisateur à taper un message salutation dans l'objectif de démarrer l'échange.

STEP 2 : A cette étape le message envoyé par l'utilisateur est analysé par le modèle LUIS afin de détecter les entités présentes dans le message. Si l'intention de réservation n'est pas détectée, le bot demande à l'utilisateur de formuler sa demande.
A noter que l'analyse sémantique par le modèle LUIS s'effectue uniquement à cette étape. Dans une prochaine version nous envisageons d'appliquer le modèle sur tous les messages envoyés par l'utilisateur.

STEP 3 : Le bot rentre dans une boucle où il est amené à demander à l'utilisateur les entités non détectées à l'étape précédente. Une procédure particulière se lance pour la détection des dates.

STEP4 : Une fois les entités récoltées l'utilisateur est amené à confirmer l'ensemble des informations saisies, s'il refuse le bot relance la conversation à partir du **STEP 2** en effaçant toutes les réponses de l'utilisateur.

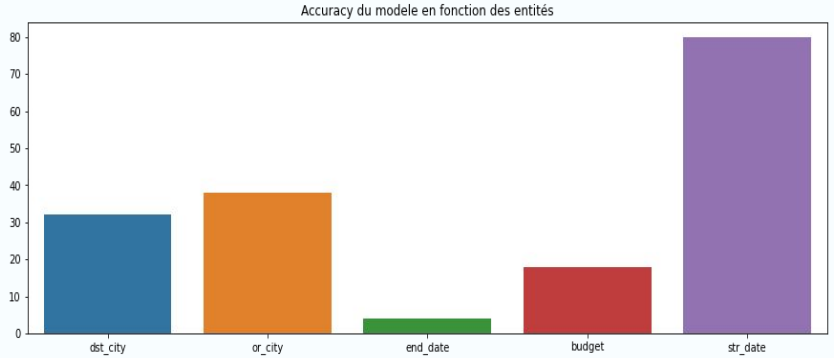
2. Critères d'évaluation de la performance du modèle

Métrique d'évaluation du modèle :

Pourcentage d'entité détecter par le modèle lors du premier message

- La capacité du modèle à reconnaître les entités dans un texte est évaluée en phase de production. Le modèle est testé sur un jeu de données test contenant des exemples textuels de réservation de vol incluant les 5 entités à reconnaître. **Lors de la phase d'entraînement le score du modèle était de 73,33% d'entités détectées**.

- D'autre part, nous évaluons pour chaque entité la capacité du modèle à les reconnaître individuellement. Pour cela nous évaluons le modèle à partir d'un jeu de données test contenant des exemples incluant au maximum une entité, nous pouvons ainsi évaluer la détection de chaque entité individuellement. (Ci-joint le graphique de l'accuracy du modèle en fonction des entités lors de la phase d'entraînement).



2. Critères d'évaluation de la performance du modèle

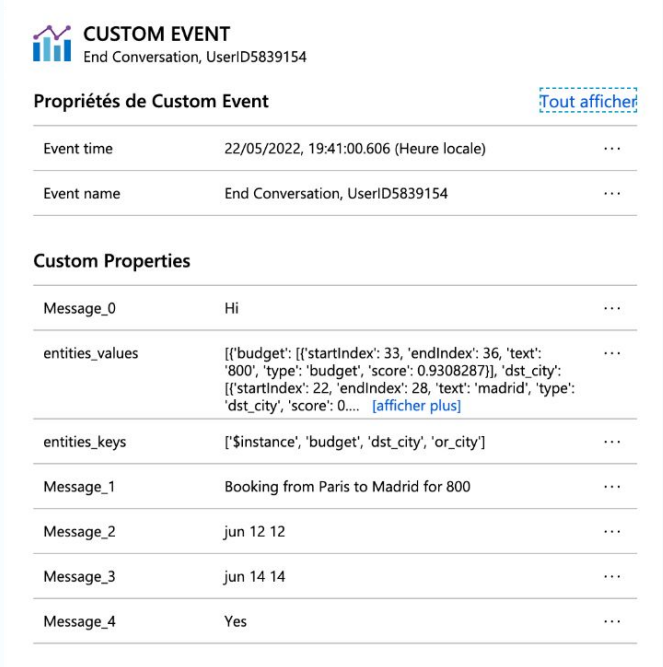
Métrique d'évaluation du modèle :

1. **Nombre moyen de message envoyé par l'utilisateur**
   Nous comptabilisons le nombre d'échange moyen entre un utilisateur et l'agent conversationnel. Le nombre d'informations étant de 5, nous estimons qu'il faudrait 5 échanges pour obtenir l'ensemble des données nécessaire à la réservation. L'évolution de cette moyenne permettra de suivre l'efficience du programme en production.

2. **Nombre moyen d'erreur par conversation**
   Nous verrons par la suite quelles sont les erreurs pouvant être remontées par le programme. La moyenne des erreurs par conversation permet également de suivre les performances du modèle.

Ces métriques et leurs seuils seront amenés à évoluer lors de la première phase de vie du bot afin d'adapter au mieux le déclenchement des alertes ainsi que de la procédure de mise à jour du modèle.

3. Seuil d'alerte du modèle



CUSTOM EVENT
End Conversation, UserID5839154

Propriétés de Custom Event — Tout afficher

Event time	22/05/2022, 19:41:00.606 (Heure locale)	···
Event name	End Conversation, UserID5839154	···

Custom Properties

Message_0	Hi	···
entities_values	[{'budget': [{'startIndex': 33, 'endIndex': 36, 'text': '800', 'type': 'budget', 'score': 0.9308287}], 'dst_city': [{'startIndex': 22, 'endIndex': 28, 'text': 'madrid', 'type': 'dst_city', 'score': 0.... [afficher plus]	···
entities_keys	['$instance', 'budget', 'dst_city', 'or_city']	···
Message_1	Booking from Paris to Madrid for 800	···
Message_2	jun 12 12	···
Message_3	jun 14 14	···
Message_4	Yes	···

L'application App Insights permet d'avoir un suivi détaillé des interactions utilisateurs.

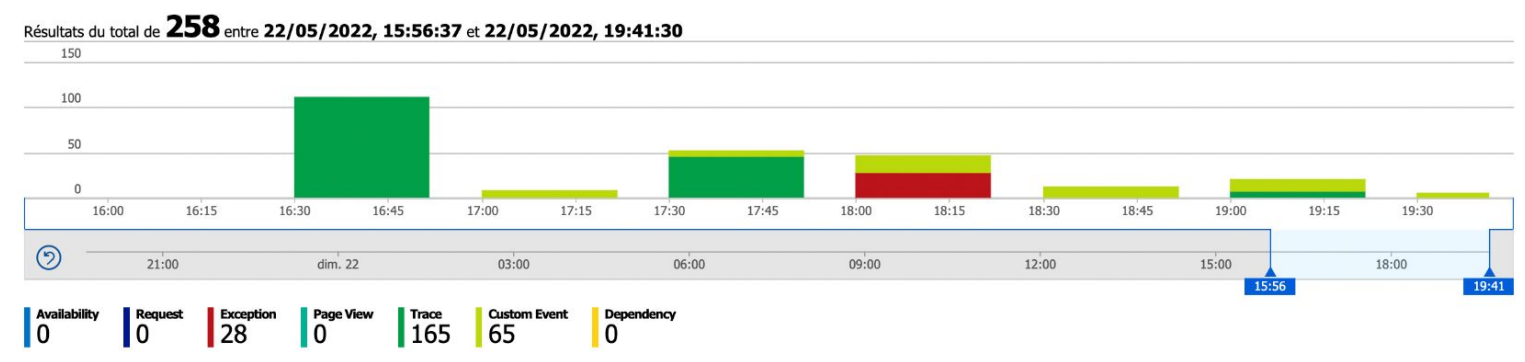
Nous relevons au terme de la conversation, qu'elle soit validée ou non par l'utilisateur :

- **L'ensemble des messages par ordre chronologique**
- **Les entités détectées par le modèle LUIS**
- **Le nombre de message envoyé par l'utilisateur**
- **Le nombre d'erreur relevé survenue lors des échanges**

Les seuils d'alerte déclenche des événements sur la plateforme Microsoft Azure Insights. Voici la liste des événements déclencheurs d'alerte en fonction de leur niveau d'intérêt :

- Démarrage et fin de conversation **| Niveau Info**
  Appel au model LUIS **| Niveau info**
- Un nombre de message échangé supérieur à 6 **| Niveau Warning**
- Une réponse négative à l'étape de validation des données renseignées **| Niveau Warning**
- 3 Erreur du chatbot **| Niveau Error**
- Un nombre de message échangé supérieur à 10 **| Niveau Critical**

3. Seuil d'alerte du modèle



Sur cette capture d'écran provenant de l'application Insights de Microsoft Azure nous apercevons les données de l'application flybot :

- **En rouge nous pouvons apercevoir les exceptions levées par les seuils d'alerte de niveau "error" et "critical".**
- **En vert foncé les traces des alertes de niveau warning.**
- **En vert clair les alertes de niveau "info".**

4. Méthodologie de mise à jour du modèle

Afin d'améliorer les performances du modèle, il sera nécessaire d'effectuer une évaluation du modèle par fréquence de 10 conversations invalidées par utilisateurs ou toutes les 15 conversations validées par l'utilisateur si l'accuracy par entité est dégradée par rapport aux valeurs des scores lors de la mise en production (cf. *Critères d'évaluation de la performance du modèle* page 6).

L'évaluation portera sur la capacité du modèle LUIS d'extraire les entités présentes dans les premiers messages utilisateurs.

L'équipe de développement en charge de l'entretien du modèle se chargera de labéliser à la main les données textuelles envoyées par les utilisateurs, concernant à la fois les conversions ayant été à leur terme ainsi que celles abandonnées par les utilisateurs pendant la période des 10 conversations validées.

Nous déterminerons l'accuracy du modèle sur ces données en fonction des entités.

En fonction des scores obtenus il sera nécessaire d'inscrire l'entraînement du modèle avec les nouvelles données utilisateurs en fonction des scores par entitées.