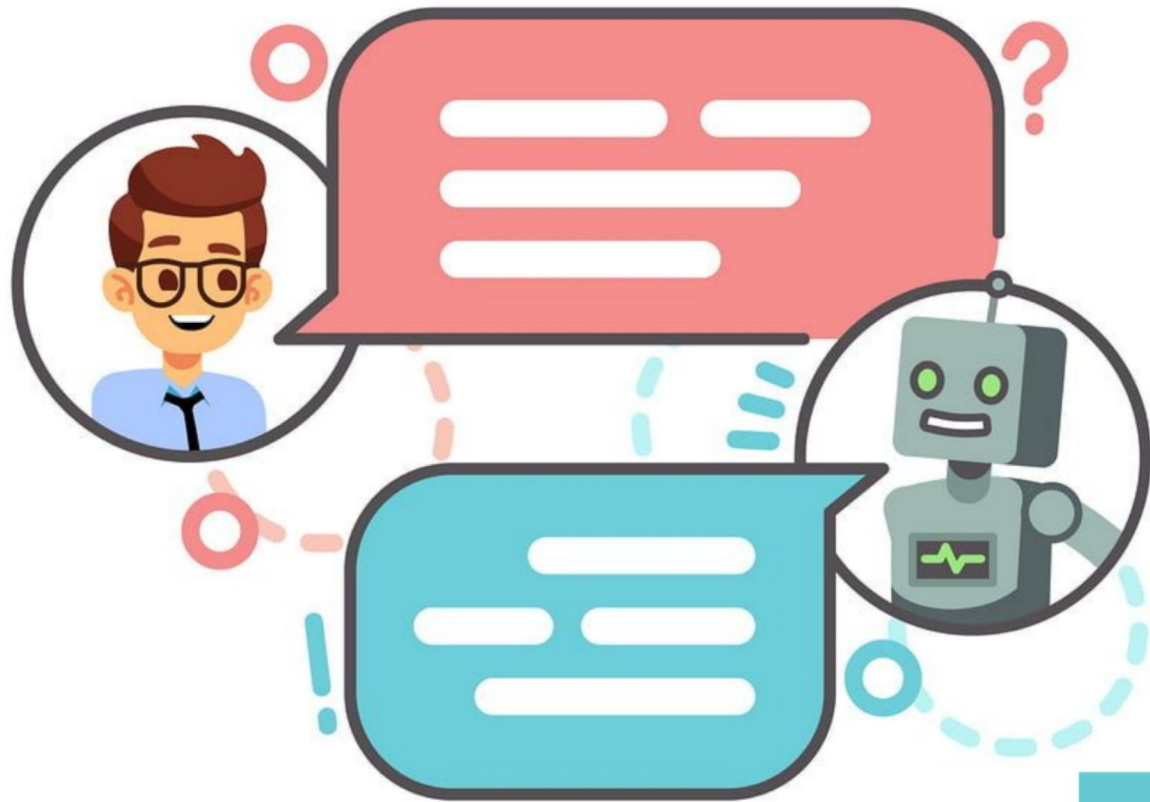


Réinventons l'expérience client

Emmanuel Letremble

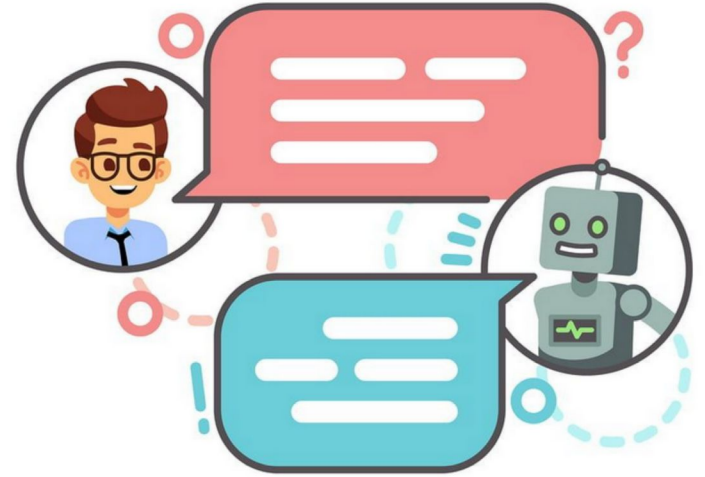


Réinventons l'expérience client

Credit

Concevoir un chatbot permettant de fluidifier l'expérience des utilisateurs grâce à des réponses ciblées et rapides ...

Le résultat qui doit être **Polyglotte, Fiable et Pertinent**, permettra de s'adapter aux usages numériques pour conserver et conquérir de nouveaux marchés.

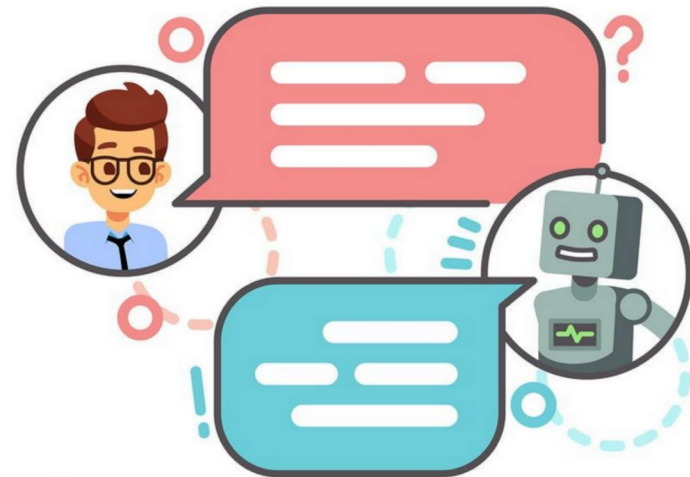


L'objectif

Développer la fonctionnalité polyglotte permettant d'identifier la langue utilisée par l'utilisateur.

Première étape: développer un outil pour évaluer la qualité du modèle pré-entraîné d' sur les 5 langues les plus fréquentes.

(Mandarin, Espagnol, Anglais, Arabe, Hindi)



Credit

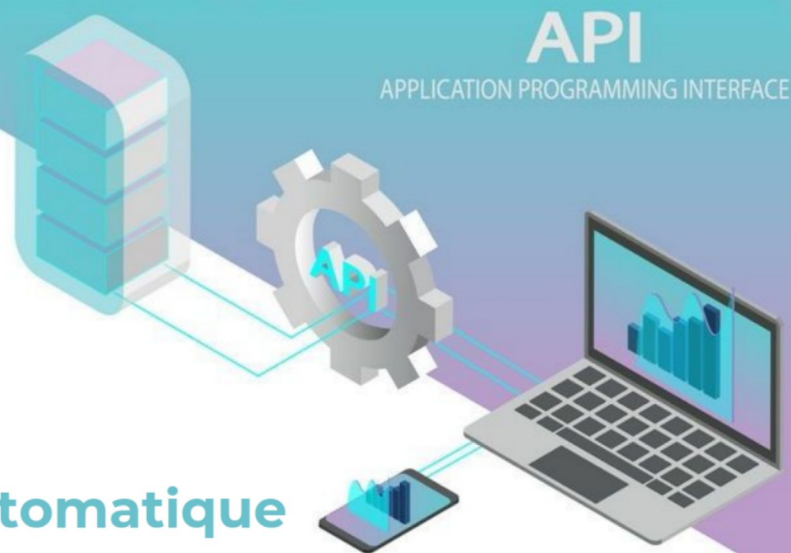
Azure Cognitive Service

Un ensemble de modèles pré-entraînés utilisables contre rémunération.

- vision par ordinateur
- analyse audio
- analyse de texte (dont translator)
- aide à la décision (données tabulaires)

Translator est un service de traduction automatique

- un service API REST (donc compatible avec n'importe quel système d'exploitation)
- traduction de texte en temps réel
- traduction de documents
- détection des langues
- plus de 100 langues supportées (103 en octobre 2021)



Credit

Azure Translator

Contraintes

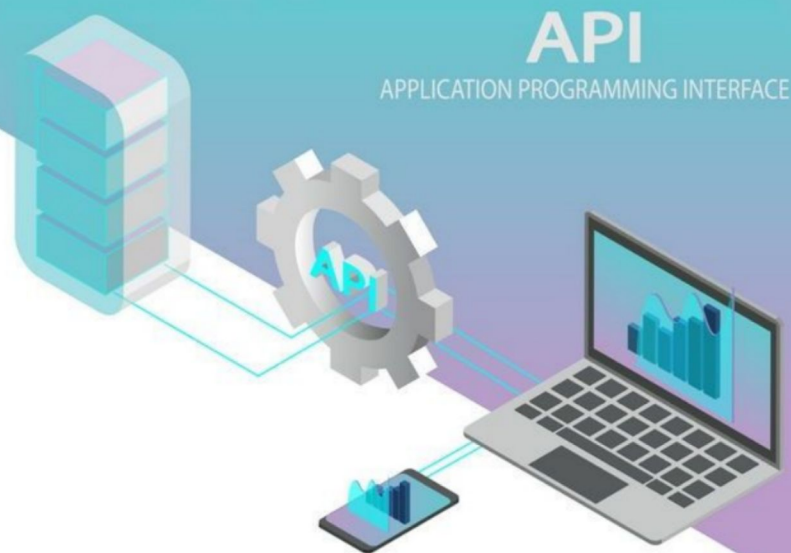
- au maximum 100 éléments par requête
- au maximum 50 000 caractères par requête
- le prix dépend du nombre de caractères envoyés

(8.998€ par million avec une instance S1)

Points intéressants

- les modèles sont **pré-entraînés** (no training required)
- les modèles **peut être personnalisés** (feature-extraction, fine-tuning)
- les modèles sont **mis-à-jour par Microsoft** (no datadrift problem)
- les modèles sont **hébergés dans le cloud Azure** (hosting / servers / access prices included)
- les ressources créées sont **facilement déployable dans le monde entier**
- les **données envoyées ne sont pas stockées de façon permanente**

<https://www.microsoft.com/en-us/translator/business/notrace/>



Credit

Démonstration



Jupyter notebook

```
>>> jupyter notebook notebook.ipynb
```



Script Python

```
>>> python Azure_Translator_Interface.py  
>>> python Azure_Translator_Interface.py -f data/export_hin.txt hin
```

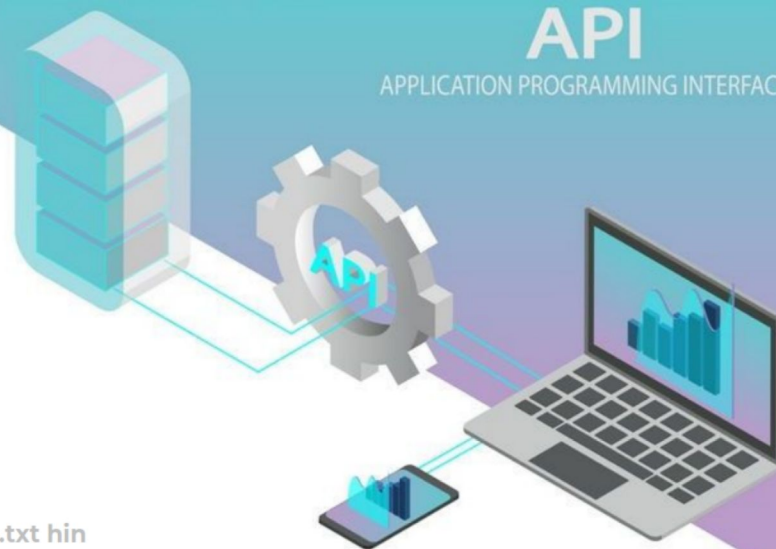


Dépôt Git

```
>>> https://github.com/Valkea/OC\_AI\_01
```

API

APPLICATION PROGRAMMING INTERFACE



Conclusions

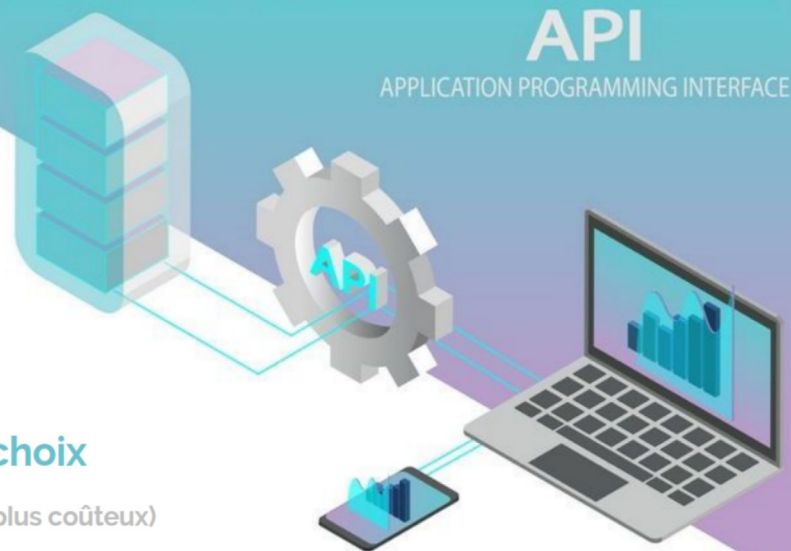


Le modèle pré-entraîné d'Azur translator

- semble être une bonne solution pour détecter le langage
(aucune faute de prédiction n'est apparue sur plus de 500 tests)
- a un temps de réponse intéressant
(moins de 300ms de latence en moyenne selon la page de monitoring de la ressource)
- peut détecter 67 langages différents
- propose des alternatives lorsqu'il n'est pas assez sûr de son choix
- peut être personnalisé pour des besoins spécifiques (mais c'est plus coûteux)

Alternatives

- SpaCy
- Amazon Comprehend - Detect the Dominant Language (104 langages)
- Google cloud - Detecting languages (100+ langages)



Merci

de m'avoir écouté, évalué et conseillé

