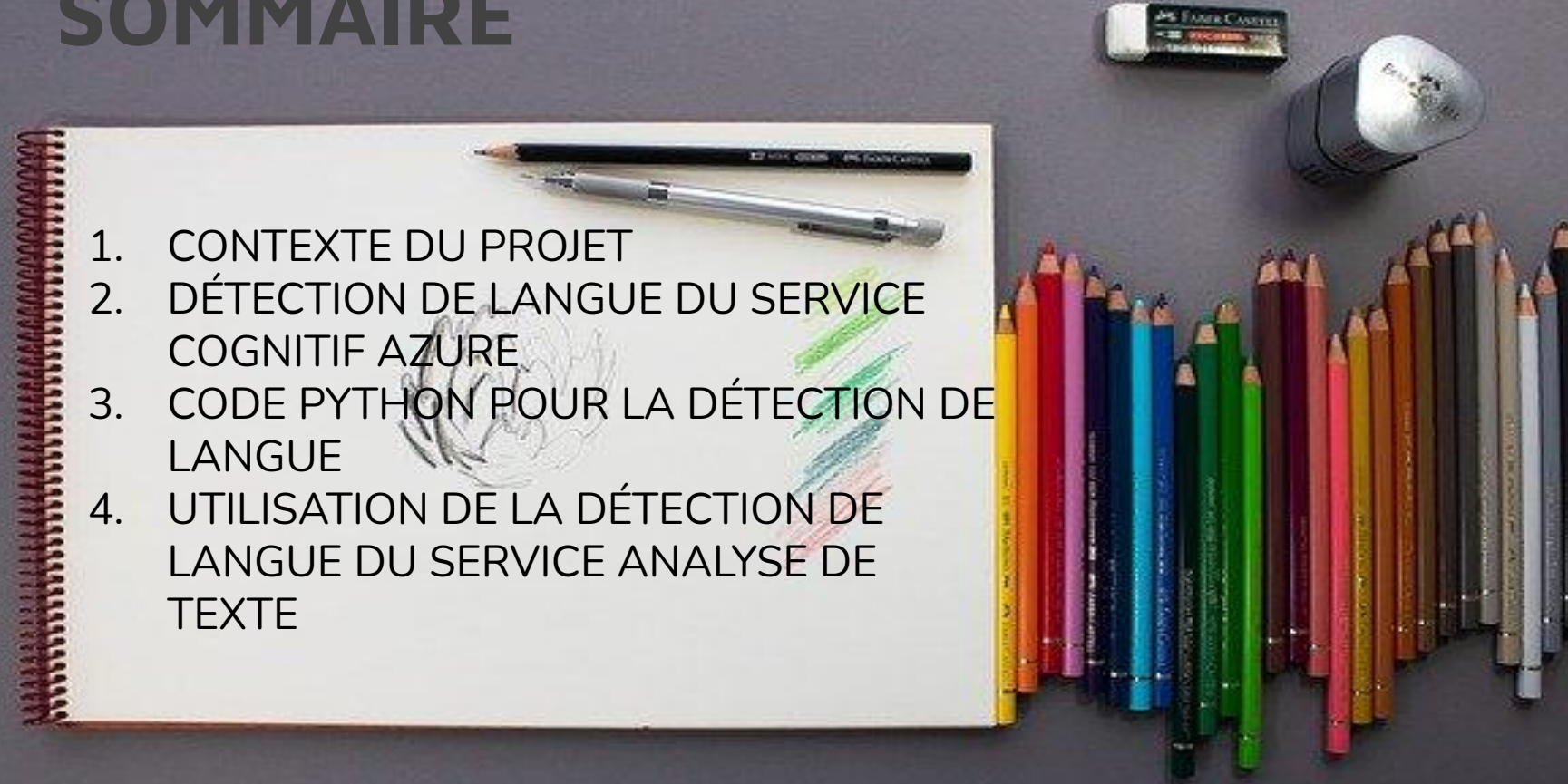


Réinventons  
l'expérience  
client avec le  
chatbot

BANQUE CREDIT



# SOMMAIRE

- 
1. CONTEXTE DU PROJET
  2. DÉTECTION DE LANGUE DU SERVICE COGNITIF AZURE
  3. CODE PYTHON POUR LA DÉTECTION DE LANGUE
  4. UTILISATION DE LA DÉTECTION DE LANGUE DU SERVICE ANALYSE DE TEXTE



## 1. CONTEXTE DU PROJET

## 1. CONTEXTE DU PROJET

Le chatbot apporte un support clients mais aussi des informations sur les services bancaires





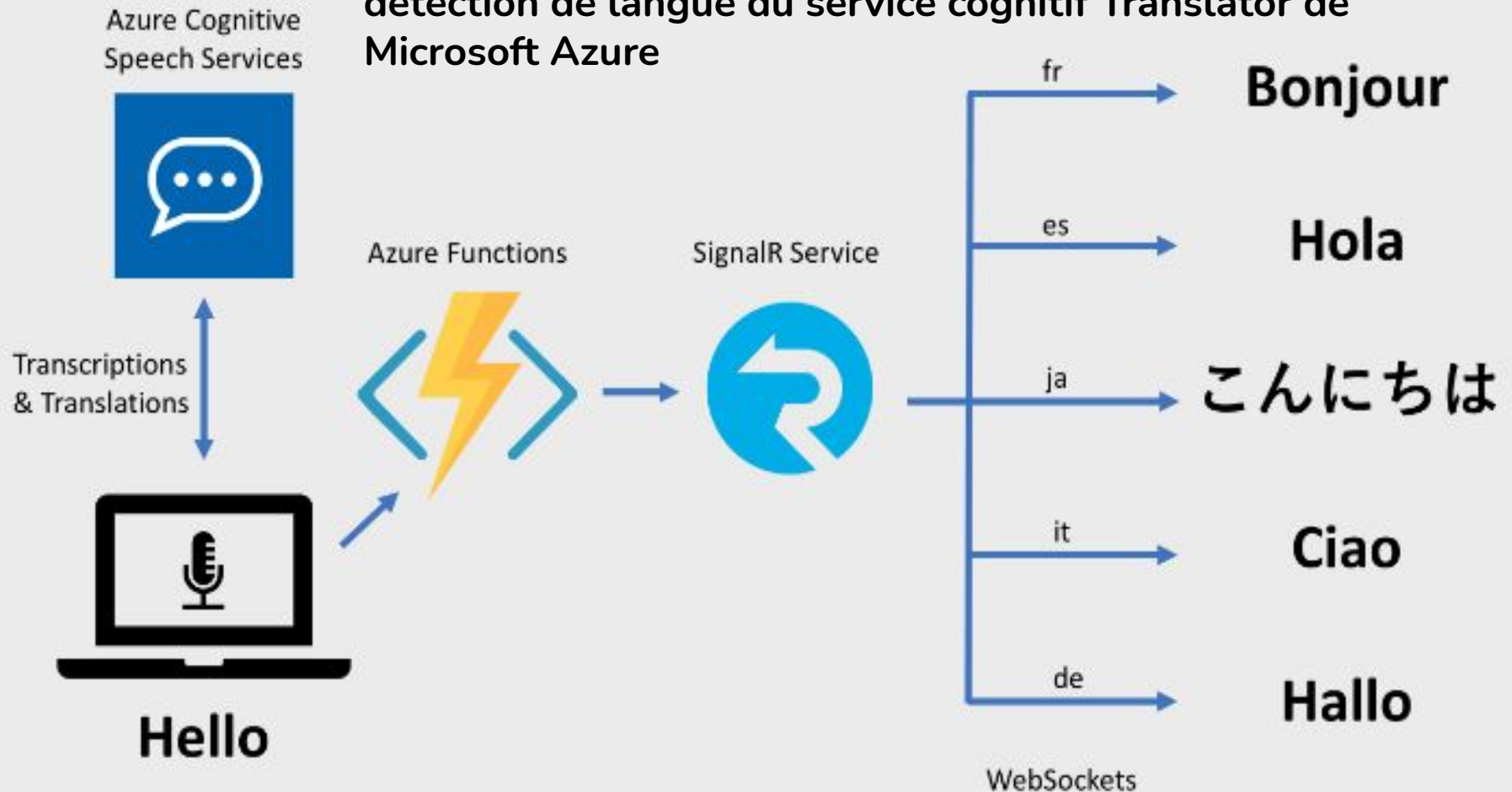
Le chatbot apporte un support clients mais aussi  
des informations sur les services bancaires



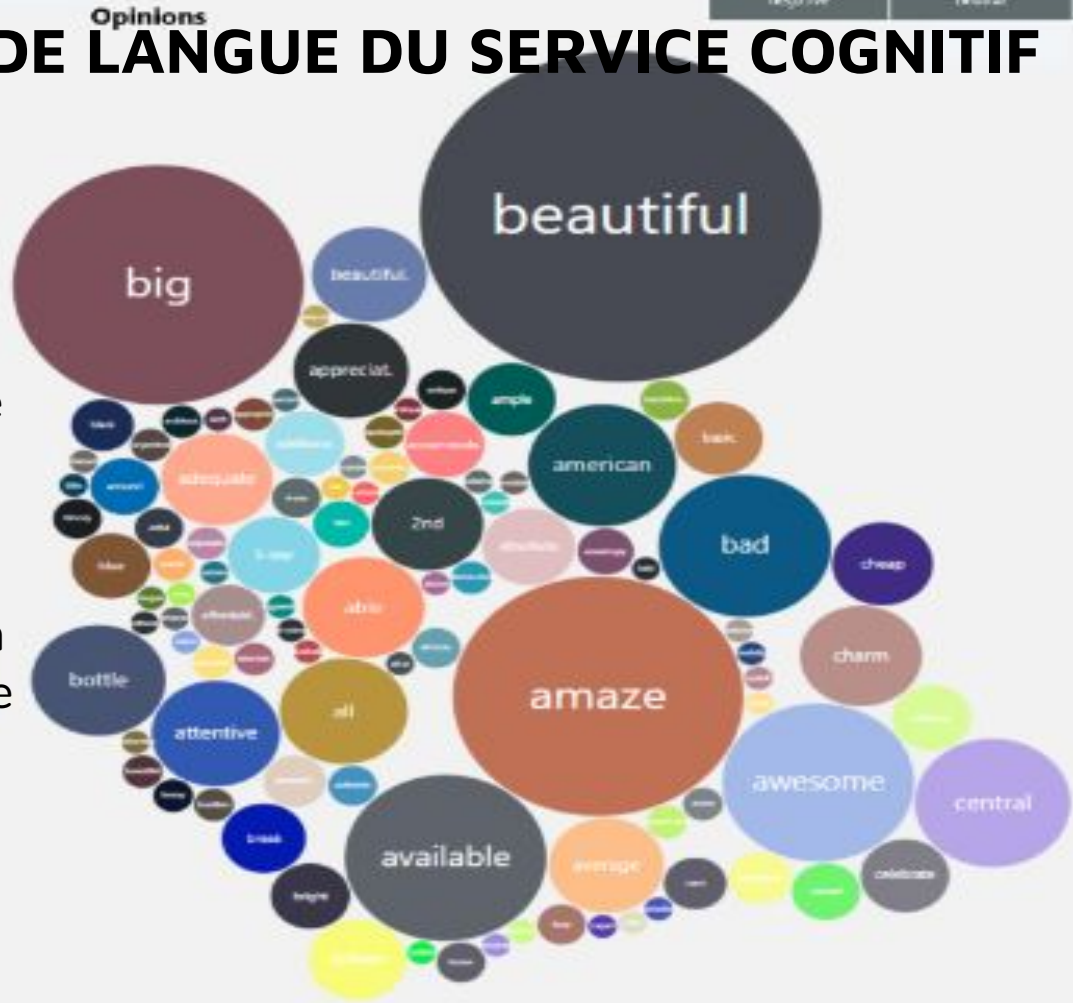


Le chatbot doit être fiable, pertinent et polyglotte

Ce projet traite de la fonctionnalité polyglotte en utilisant la détection de langue du service cognitif Translator de Microsoft Azure



Le service Analyse de texte de Microsoft Azure nous fournit 4 fonctionnalités principales : détection de la langue, l'analyse des sentiments, l'extraction des phrases clé, la reconnaissance des entités.

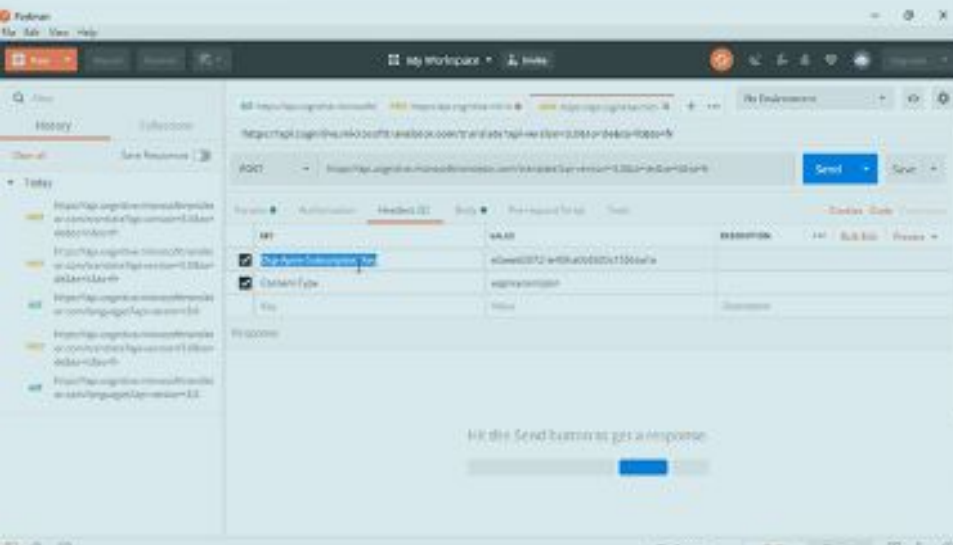




## Translator Text Is a REST API



Uses the REST verbs POST and GET  
Requires body to be in JSON format  
API key passed in the header to authorize



L'utilisation de la détection de langue passe par l'appel de l'API Translator via la clé d'authentification et le point de terminaison définis lors de la création d'une ressource Translator dans Azure.

Pour chaque texte envoyé à analyser, le service de détection de la langue nous renvoie sous format JSON : le code ISO 6391 de la langue (exemple 'en' pour l'anglais) et un score de confiance entre 0 et 1.

```
[  
  {  
    "isTranslationSupported": true,  
    "isTransliterationSupported": true,  
    "language": "en",  
    "score": 1.0  
  }  
]
```

### 3. CODE PYTHON POUR LA DÉTECTION DE LA LANGUE

```
176         default="r",
177     )
178     )
179     global min, max, t
180     min=0.01, max=1000.0,
181     t=1.0,
182
183
184
185 def execute(self, context):
186
187     # get the folder
188     folder_path = (os.path.dirname(self.filepath))
189
190     # get objects selected in the viewport
191     viewport_selection = bpy.context.selected_objects
192
193     # get export objects
194     obj_export_list = viewport_selection
195     if self.use_selection_setting == False:
196         obj_export_list = [i for i in bpy.context.scene.objects]
197
198     # deselect all objects
199     bpy.ops.object.select_all(action='DESELECT')
200
201     for item in obj_export_list:
202         item.select = True
203         if item.type == 'MESH':
204             file_path = os.path.join(folder_path, "{}.obj".format(item.name))
205             bpy.ops.export_scene.obj(filepath=file_path, use_selection=True,
206                                     axis_forward=self.axis_forward_setting,
207                                     axis_up=self.axis_up_setting,
208                                     use_animation=self.use_animation_setting,
209                                     use_mesh_modifiers=self.use_mesh_modifiers_setting,
210                                     use_edges=self.use_edges_setting,
211                                     use_smooth_groups=self.use_smooth_groups_setting,
212                                     use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,
213                                     use_normals=self.use_normals_setting,
214                                     use_uv=self.use_uv_setting,
215                                     use_materials=self.use_materials_setting,
```



```
# This simple app uses the '/detect' resource to identify  
# the language of the provided text or texts.
```

```
import os, requests, uuid, json
```

```
key1_var_name      = 'COGNITIVE_SERVICE_KEY'  
key2_var_name      = 'COGNITIVE_SERVICE_REGION'  
endpoint_var_name  = 'TRANSLATOR_TEXT_ENDPOINT'
```

```
if not key1_var_name in os.environ:  
    msg = 'Please set/export the environment variable 1: {}'  
    raise Exception(msg.format(key1_var_name))  
subscription_key = os.environ[key1_var_name]
```

```
if not key2_var_name in os.environ:  
    msg = 'Please set/export the environment variable 2: {}'  
    raise Exception(msg.format(key2_var_name))  
region_key = os.environ[key2_var_name]
```

```
if not endpoint_var_name in os.environ:  
    msg = 'Please set/export the environment variable 3: {}'  
    raise Exception(msg.format(endpoint_var_name))  
endpoint = os.environ[endpoint_var_name]
```

```
# If you encounter any issues with the base_url or path, make sure
# that you are using the latest endpoint:
# https://docs.microsoft.com/azure/cognitive-services/translator/

path = '/detect?api-version=3.0'
constructed_url = endpoint + path

headers = {
    'Ocp-Apim-Subscription-Key' : subscription_key,
    'Ocp-Apim-Subscription-Region' : region_key,
    'Content-type' : 'application/json',
    'X-ClientTraceId' : str(uuid.uuid4())
}

# Text to translate or to detect the language.
body = [{
    'text' : '未幾離異。四年，撰《金石錄後序》。晚年表上《金石錄》於朝。卒年七十餘。'
}]

request = requests.post(constructed_url, headers=headers, json=body)
response = request.json()

print(json.dumps(
    response, sort_keys=True, indent=4,
    ensure_ascii=False, separators=(',', ': ')))
```

Résultat de la requête :

```
[
  {
    "isTranslationSupported": true,
    "isTransliterationSupported": true,
    "language": "zh-Hant",
    "score": 1.0
  }
]
```



## 4. UTILISATION DE LA DÉTECTION DE LANGUE DU SERVICE ANALYSE DE TEXTE



Exécuter le script python dans Jupyter Notebook pour les 5 langues les plus parlées au monde à savoir le chinois, l'espagnole, l'anglais, l'hindi et l'arabe.

# Interprétation des informations envoyées par l'API Translator sous format JSON



# Analyse de la fiabilité de la fonctionnalité Detect de l'API

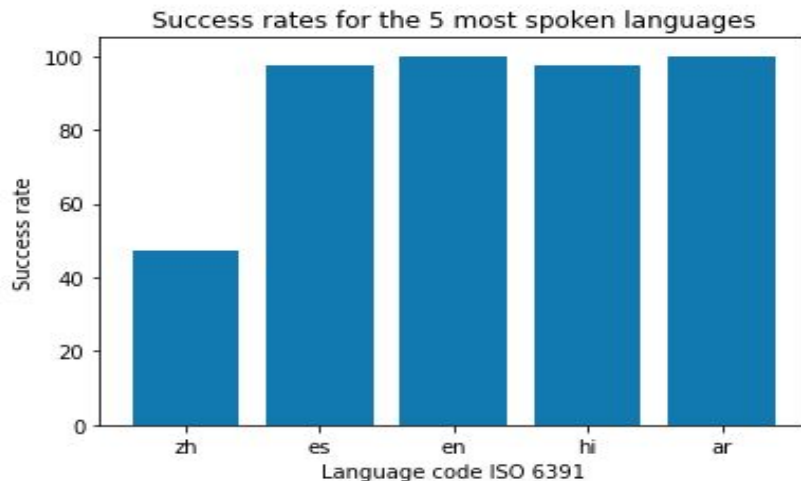




# TAUX DE RÉUSSITE POUR LES 5 LANGUES MAJEURES

- Création des fonctions `recupTexte(label, nb_para)` afin de récupérer les `nb_para=300` paragraphes du fichier `x_train.txt` selon le label de la langue à savoir `'zho','spa','eng','hin','ara'`.
- La fonction `detectLang(label,*param)` envoie les 300 textes des 5 langues vers l'API Detect Azure pour identification de la langue et calcule le taux de réussite.

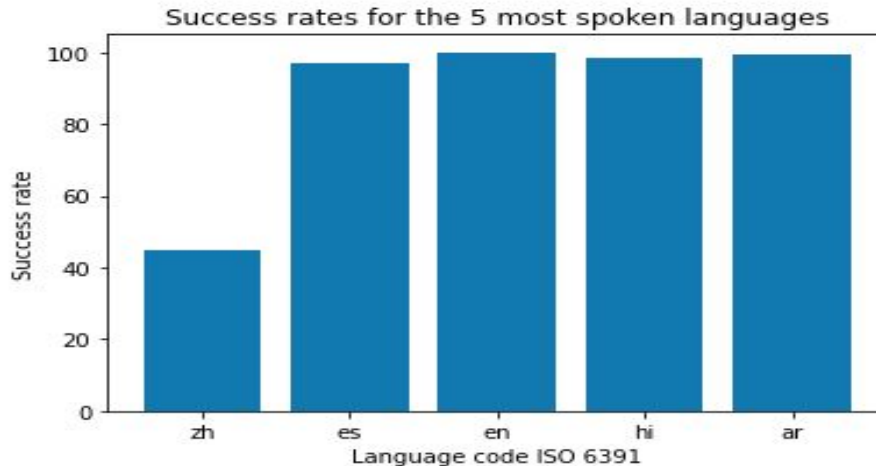
```
Détection du label : zho et chargement de 300 paragraphes
Détection du label : spa et chargement de 300 paragraphes
Détection du label : eng et chargement de 300 paragraphes
Détection du label : hin et chargement de 300 paragraphes
Détection du label : ara et chargement de 300 paragraphes
[47.33333333333333, 97.66666666666667, 100.0, 97.66666666666667, 99.66666666666667]
```



# TAUX DE RÉUSSITE POUR 500 TEXTES

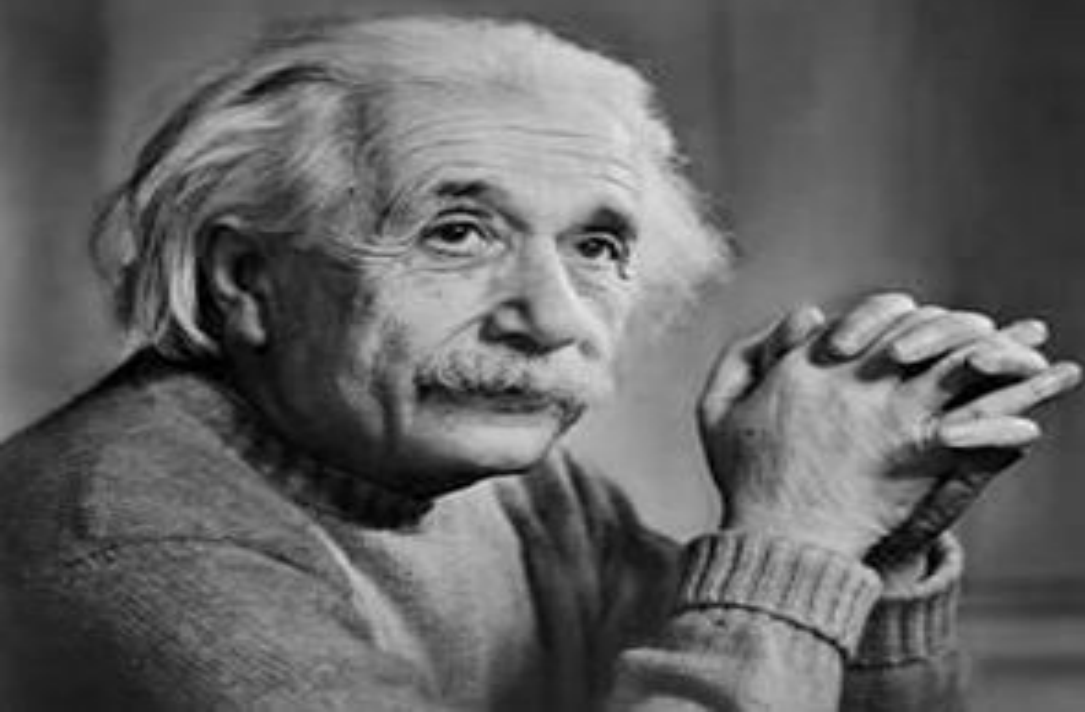
- Dans ce cas, les fonctions `recupTexte(label, nb_para)` et `detectLang(label,*param)` traitent 500 textes pour chacune des 5 langues du fichier `x_train` et les envoient vers l'API Detect Azure pour identification de la langue et calcule le taux de réussite.

```
Détection du label : zho et chargement de 500 paragraphes  
Détection du label : spa et chargement de 500 paragraphes  
Détection du label : eng et chargement de 500 paragraphes  
Détection du label : hin et chargement de 500 paragraphes  
Détection du label : ara et chargement de 500 paragraphes  
[44.6, 97.0, 100.0, 98.4, 99.6]
```



If you can't explain it **simply**, you  
don't understand it well enough.

— Albert Einstein





# EXPLICATION DES ÉCHECS D'IDENTIFICATION

- On constate que le taux de succès pour l'identification de la langue par l'API Detect Azure est moins important pour le chinois et très proches pour les 4 autres langues. Cela peut s'expliquer dans certains cas par la présence de mots en anglais par exemple dans des textes chinois, ou alors par le manque de contrôle des textes présents dans la base d'identification de la langue de Wikipedia [Wikipedia Language Identification Database](#) .

zh-Hans

最明顯的例子可舉德國為例，由於德國近20年來各方面鼓勵使用可再生能源，其成效極為卓越，所以在2010年減少了1億2000萬噸的二氧化碳，比2009年還更多1億1100萬噸的二氧化碳)，也因此讓德國提前達到減碳目標（京都議定書要求要減碳相對於1990年低於8%，而德國早於2007即達減少18.4%）。反觀台灣，2006年的二氧化碳排放量相對於1990卻增加了140%（進口能源值亦於同期增加126%）。

zh-Hant

1854年: 《蔓足亞綱》A Monograph of the Sub-class Cirripedia, with Figures of all the Species. The Balanidae (or Sessile Cirripedes); the Verrucidae, etc.

en