Project 1: Microsoft Azure Cognitive for language processing

Thi Khuyen LE

Mentor: ***Camara Moussa***

Avril 14, 2021

1. Context
2. Chatbot objectives
3. Chatbot functionalities
4. Review of Azure Cognitive services
5. Applications
   - Detection (without translation)
   - Detection (with translation)
   - Transliteration (without translation)
   - Transliteration (with translation)
   - Get length of text (without translation)
   - Get length of text (with translation)
6. Conclusions and discussions

Context:

We are in the era 4.0, the era of technology. All the industries as banking, education, research, medical, tourism, . . . are growing rapidly. Hence, the demand of communications, connectivities increases.

Limitations:

- The human resource is limited.
- Waste of time and money.
- Work pressure is too high.
- Languages, geographical location, time zone, . . .

$\Rightarrow$ Chatbot

1 Context

2 Chatbot objectives

3 Chatbot functionalities

4 Review of Azure Cognitive services

5 Applications
- Detection (without translation)
- Detection (with translation)
- Transliteration (without translation)
- Transliteration (with translation)
- Get length of text (without translation)
- Get length of text (with translation)

6 Conclusions and discussions

Chatbot objectives:

- Customer support:
  - ◇ Simplification the transaction and the connection of clients.
  - ◇ Records technical incidents encountered by clients and give them solutions.
- Advice and sale:
  - ◇ Send good offers to clients.
  - ◇ Recommendation and marketing.

1 Context

2 Chatbot objectives

3 Chatbot functionalities

4 Review of Azure Cognitive services

5 Applications
- Detection (without translation)
- Detection (with translation)
- Transliteration (without translation)
- Transliteration (with translation)
- Get length of text (without translation)
- Get length of text (with translation)

6 Conclusions and discussions

Functionalities of chatbot:

- Polyglot: Answer clients in their languages.
- Reliable: Give suitable answers to client questions.
- Relevant: Orient the client to the right interlocutor when the chatbot reaches his limits.

Purpose of project: Discover the polyglot function of chatbot.

Methodology: Use the pre-trained model on Microsoft Azure for text recognition.

1 Context

2 Chatbot objectives

3 Chatbot functionalities

4 Review of Azure Cognitive services

5 Applications
- Detection (without translation)
- Detection (with translation)
- Transliteration (without translation)
- Transliteration (with translation)
- Get length of text (without translation)
- Get length of text (with translation)

6 Conclusions and discussions

Azure cognitive services:

- Language
  - ◇ Translator: Detect and translate more than 90 support languages.
  - ◇ Text analytics: Detect sentiment, key phrases and named entities.
  - ◇ Q&A maker: create conservational questions and answers over the given data.
  - ◇ Immersive reader: help readers understand text using audio and visual cues.
  - ◇ Language understanding: build natural language understanding into apps, bots or IoT devices.
- Vision
  - ◇ Computer vision: Analyse content in image and video.
  - ◇ Custom vision: customize image recognition to fit yout business needs.
  - ◇ Face: Detect and identity people.
  - ◇ Form recognizer: Extract the content from the document.
  - ◇ Video indexer: analyze the visual and audio channels of a video, and index its content.
- Decision (Content moderator): Detect potentially offensive or unwanted content.

1. Context
2. Chatbot objectives
3. Chatbot functionalities
4. Review of Azure Cognitive services
5. Applications
   - Detection (without translation)
   - Detection (with translation)
   - Transliteration (without translation)
   - Transliteration (with translation)
   - Get length of text (without translation)
   - Get length of text (with translation)
6. Conclusions and discussions

How to use Azure cognitive services from local computer?

Requirements:

- Subscription key,
- Endpoint.

Command structure

```
import requests, uuid, json

# subscription_key and endpoint
subscription_key = "your_subscription_key"
endpoint = "https://api.cognitive.microsofttranslator.com/"

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

document = ['id': 1,

'text': '1988년 8월 어머니는 보호책임자 유기치사 혐의로 체포 및 기소되었고, 징역 3년에 집행유예 4년의 판결을 받았다. 장남 역시 삼녀의 사망에 관여하였다고 보고 상해치사 및 시신유기죄로 도쿄 가정법원에 송치되지만 상황을 고려해 고아원에 보내졌다. ']

Detection (without translation)

Commands

```
path = '/detect'
constructed_url = endpoint + path
params = {
    'api-version': '3.0'
}
request = requests.post(constructed_url, params = params, headers = headers,
                        json = document)
response = request.json()

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


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

Detection (with translation)

Commands

```
path = '/translate'
constructed_url = endpoint + path
params = {
    'api-version': '3.0',
    'to': ['en']
}
request = requests.post(constructed_url, params = params, headers = headers,
                        json = document)
reponse = request.json()

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


```
"detectedLanguage": { "language": "ko", "score": 1.0},
"translations": [{
"text": "In August 1988, her mother was arrested and indicted on charges of
 organicchising the protective officer, and was sentenced to three years in prison
 and four years of probation. The eldest son was also involved in the death of his
 three daughters and was sent to an orphanage in consideration of the situation,
 although he was sent to the Tokyo Family Court for injury and body injury. ",
"to": "en" }]
```

Transliteration (without detection)

Commands

```
path = '/transliterate'
constructed_url = endpoint + path
params = {
    'api—version': '3.0',
    'language': 'ko',
    'fromScript': 'Kore',
    'toScript': 'latn'
}
request = requests.post(constructed_url, params = params, headers = headers,
                        json = document)
response = request.json()

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


```
"script":"latn",
"text":"1988nam 8woll eomeonineun bohochaegimja yugichisa hyeomuiro chepo mukgye
gisodoeossgo, jinggeok 3nyeone jiphaeng 4nine pangyeoreul badassda. jangnam yeoksi
samnyeoui samange gwanyeohayeossdago bogo sanghaechisa mukgye sisinyugijoro tokio
gajeongbeopwone songchidoejiman sanghwangeul goryeohae goawone bonaejyeossda."
```

Transliteration (with detection)

Commands

```
path = '/transliterate' #transliterate endpoint
constructed_url = endpoint + path
params = {
    'api-version': '3.0',
    'language': 'ko',
    'toScript': 'latn'
}
request = requests.post(constructed_url, params = params, headers = headers,
                                    json = document)
response = request.json()

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

```
  "detectedLanguage":{"language": "ko"; "score": 1.0 }
```

"translations":["text":"1988년 8월 어머니는 보호책임자 유기치사 혐의로 체포 및 기소되었고, 징역 3년에 집행유예 4년의 판결을 받았다. 장남 역시 삼녀의 사망에 관여하였다고 보고 상해치사 및 시신유기죄로 도쿄 가정법원에 송치되지만 상황을 고려해 고아원에 보내졌다. "; "to":"ko"; "transliteration": "script":"Latn"; "text":"1988nam 8woll eomeonineun bohochaegimja yugichisa hyeomuiro chepo mukgye gisodoeossgo, jinggeok 3nyeone jiphaeng 4nine pangyeoreul badassda. jangnam yeoksi samnyeoui samange gwanyeohayeossdago bogo sanghaechisa mukgye sisinyugijoro tokio gajeongbeopwone songchidoejiman sanghwangeul goryeohae goawone bonaejyeossda. "]

Get length of text (without translation)

Commands

```
path = '/breaksentence'

constructed_url = endpoint + path

params = {
    'api-version': '3.0',
}

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

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

```
 "detectedLanguage":{"language": "ko", "score": 1.0},
 "sentLen":[66, 72]
```

Get length of text (with translation)

Commands

```
path = '/translate'
constructed_url = endpoint + path
params = {
    'api-version': '3.0',
    'to': 'en',
    'includeSentenceLength': True
}
request = requests.post(constructed_url, params = params, headers = headers,
                        json = document)
response = request.json()

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


```
"detectedLanguage": {"language": "ko","score": 1.0}
"translations":[{
     "sentLen":{
          "srcSentLen": [66, 72];
          "transSentLen": [178, 208]}
      "text":"In August 1988, her mother was arrested and indicted on charges of
      organicchising the protective officer, and was sentenced to three years in
      prison and four years of probation. The eldest son was also involved in the
      death of his three daughters and was sent to an orphanage in consideration
      of the situation, although he was sent to the Tokyo Family Court for injury
      and body injury. ",
      "to":"en"}]
```

1. Context
2. Chatbot objectives
3. Chatbot functionalities
4. Review of Azure Cognitive services
5. Applications
   - Detection (without translation)
   - Detection (with translation)
   - Transliteration (without translation)
   - Transliteration (with translation)
   - Get length of text (without translation)
   - Get length of text (with translation)
6. Conclusions and discussions

Conclusions and discussions

Conclusions

- Microsoft Azure is a useful platform for natural language processing. It provides multiples tools for handling different task as detecting, translating, transliterating text in different languages.
- The pre-trained model in this platform is a premise for creating the polyglot functionality of chatbots.

Discussions:

- The number of processed languages in the Microsoft Azure platform is still limit. There are only 90 languages, while there are in total 7,139 languages in the global.
- The performance of the pre-trained model is not really high, it gives incorrect result when processing mixed languages. Hence this model should de improved.

THANK YOU!