

1 マイクロソフトコグニティブサービス  
2  
3 著者名 川上 玲  
4 ご所属 東京大学情報理工学系研究科  
5 英文表題 Microsoft Cognitive Services  
6 英文氏名・ご所属 Rei Kawakami, Graduate School of  
7 Information Science and Technology, the University of  
8 Tokyo  
9 キーワード (6個以内の和文キーワード)  
10 深層学習, API, Azure, 画像認識, 自然言語処理  
11 英文キーワード (2年後 J-STAGE 公開時に使用)  
12 Deep learning, API, Azure, Image recognition, Natural  
13 language processing  
14 内容概要 (170字程度 2年後 J-STAGE 公開時に使用)  
15 深層学習のネットワークは、特に訓練時に大量のデータや  
16 計算資源を必要とします。そのようなデータや計算資源を  
17 持たない開発者や個人でも利用できるように、API を通じ  
18 て公開するサービスが各社からリリースされています。本  
19 稿では、Microsoft Azure Cognitive Services を使い、文字列  
20 や顔の認識、画像識別、自然言語処理などの API の利用方  
21 法を紹介します。  
22  
23 以下 (本文 1行 26字 2段組)  
24 **1. まえがき**  
25 深層学習のネットワークは、特に訓練時に大量のデータ  
26 や計算資源を必要とします。そのようなデータや計算資源  
27 を持たない開発者や個人でも手軽に利用できるように、  
28 APIを通じて公開するサービスが2015年頃から現れ始めま  
29 した。その後、各社から競う様にリリースされ、現在では  
30 Microsoft Azure Cognitive Services, Google Cloud API, IBM  
31 Watson API, Amazon (Rekognition, Transcribe, など), 画像  
32 や動画に限れば Clarifai などが公開されています。光学文  
33 字認識や顔認識など API で公開されている機能で十分な場  
34 合や、自社でのデータ収集が難しい場合、また、計算資源  
35 が限られるモバイルアプリケーションを作成する場合には  
36 このような API を用いる方がアプリケーションの精度が良  
37 く開発も早いと思われま。す。  
38 そこで本稿では、Microsoft Azure Cognitive Services を例  
39 に、こういった API の利用方法をご紹介します。これらの  
40 API のサービスでは、クラウド上の計算サーバが https 経由  
41 でリクエストを受け付け、結果を返す仕様になっているこ  
42 とが多く見受けられます。このため、様々な種類の言語か  
43 ら利用できる一方で、インターネットに繋がっていないと  
44 利用できません。また https のリクエスト時に、サービスを  
45 利用する有効なアカウントからのリクエストであることを  
46 示す Key などが必要となります。以下では Key は取得済み  
47 である前提で Python を使った API のサンプルコードをご説  
48 明していきます。Key は API のリクエストが頻繁でない

49 場合は、無償で作成することができます。作成方法はウェ  
50 ブ版をご覧ください。  
51 **2. Computer Vision API**  
52 Computer Vision API では、画像の中に写る一般物体の検  
53 出や識別、文字の読み取り、著名人やランドマークの認識、  
54 などが利用できます。この API の機能は、Azure Cognitive  
55 Services の Computer Vision のウェブページ [1] で、ページ  
56 の下部に表示されている画像例をいくつかクリックすると、  
57 応答の違いを確認することができます。  
58 筆者らが用意した API への入力画像とその応答の例を図  
59 1 に三つ示します。応答の中には、その画像を記述する 20  
60 個程度のキーワード (タグ) や、画像を記述した文 (キャ  
61 プション) とその信頼度、画像のサイズや形式といったメ  
62 タデータがあります。まず、図 1 の一例目の応答で、特に  
63 文の部分でこの API が人工知能の父の一人である Marvin  
64 Minsky を認識していることが分かります。この API は 20  
65 万人の著名人を記憶しており、筆者らが試したところでは、  
66 安倍首相、麻生財務大臣、タレントの明石家さんま、ニュ  
67 ートンやガリレオの肖像画、「中国人の部屋」で人工知能  
68 を批判した哲学者 John Searle を認識していました。ダーウ  
69 インやパーセプトロンの発明者 Frank Rosenblatt は知らな  
70 かったようです。  
71 図 1 の二例目では、この API は「小さな青い鳥が枝に止  
72 まっている」という文を 0.945 という高い信頼度で出力し  
73 ています。屋外、小さい、動物などのタグも見られます。  
74 図 1 の三例目を見ると、人、ケーキ、誕生日、人のグル  
75 ープ、屋内、テーブル、子供、パーティ、などのタグがあり、  
76 「人のグループがテーブルに座り誕生日ケーキがある」と  
77 いう文を生成しています。「座っている」部分以外は的確  
78 に表現されています。生成文の信頼度は 0.878 と二例目よ  
79 りも低く、どこか間違えていることも理解しているよう  
80 です。  
81 Python3 から Computer Vision API を呼び出す基本的なコ  
82 ードを下記に解説します。コードは Microsoft の公式サイト  
83 からの引用であり、オリジナルのコードは [2] から参照で  
84 きます。  
85  
86 ##### Python 3.6 #####  
87 subscription\_key = '1234567890abcdefghijklmnopqrstuvwxyz'  
88 assert subscription\_key  
89 vision\_base\_url =  
90 "https://eastasia.api.cognitive.microsoft.com/vision/v1.0/"  
91 vision\_analyze\_url = vision\_base\_url + "analyze"  
92 image\_url =  
93 "http://www.nae-lab.org/~rei/share/cat.png"  
94  
95 上のコードは Key、計算サーバのある場所、や入力画像  
96 の URL などを指定しています。Subscription\_key には、

1 Microsoft アカウントから取得した 32 桁の Key を入力しま  
 2 す。Vision\_base\_url には Key を取得時に指定したサーバの  
 3 場所 (Endpoint) の下に/vision/v1.0/を足したものを、  
 4 vision\_analyze\_url はリクエスト送信先の URL です。上の  
 5 Key はダタラメなのでご注意ください。

```
6
7 import requests
8 headers = {'Ocp-Apim-Subscription-Key': subscription_key}
9 params = {'visualFeatures': 'Categories,Description,Color',
10          'language': 'en'}
11 data = {'url': image_url}
12 response = requests.post(vision_analyze_url, headers=headers,
13                          params=params, json=data)
14 response.raise_for_status()
15 analysis = response.json()
```

17 上のコードは、API へ送信するヘッダ、パラメータ、デ  
 18 ータを指定し、リクエストを実行します。requests は通信に  
 19 必要なライブラリで、API を呼び出し JSON オブジェクト  
 20 を返します。(JSON は JavaScript Object Notation の略で、  
 21 XML の様なテキストベースの形式です。) params は API  
 22 のどんな機能や情報を利用するかを指定します。ここでは  
 23 visualFeatures のカテゴリ、記述、色、を指定しています。  
 24 language は英語か中国語が選択できます。requests.post()が  
 25 https による通信の確立やデータの送信を行い、応答が  
 26 response に返ります。response.raise\_for\_status()は例外処理で  
 27 す。response.json()がデータをデコードし analysis に返しま  
 28 す。params で指定できるパラメータの詳細は、Computer  
 29 vision API のメインページ [1] で「API」をクリックすると  
 30 開くテストコンソールのページで見ることができます。ま  
 31 た、テストコンソールページでは、「場所」を指定すれば  
 32 ブラウザからテストコンソールを動かすことができます。  
 33 (ただし動作させるには Key の入力が必要になります。)

```
35 cpt = analysis["description"]["captions"][0]["text"].capitalize() 49
36 cnf = analysis["description"]["captions"][0]["confidence"] 50
37 key_words = analysis["description"]["tags"] 51
38 print(key_words) 52
39 print(image_caption) 53
40 print(confidence) 54
```

42 上のコードは、analysis の中身を見るためのものです。cpt  
 43 は生成文、cnf はその信頼度、key\_words が画像のタグの中  
 44 身です。print(analysis)などでも構造が分かると思います。 58

### 45 3. Custom Vision API 59

46 この API では、自分でタグ付けした訓練画像を数枚 (最  
 47 低 5 枚) アップロードし API 側で分類器を訓練することで、  
 48 カスタマイズした分類器を作成することができます。Azure 62

#### 画像例



Marvin Minsky



青い鳥



誕生日会

#### 応答結果

```
'description': {'tags': ['person',
'indoor', 'man', 'cabinet', 'kitchen',
'standing', 'room', 'older', 'glasses',
'people', 'wearing', 'food', 'holding',
'table', 'old', 'white', 'group', 'shirt'],
'captions': [{'text': 'Marvin Minsky
wearing glasses', 'confidence':
0.7024199544329018}],
'metadata': {'height': 200, 'width':
200, 'format': 'Jpeg}}
```

```
'description': {'tags': ['bird',
'outdoor', 'animal', 'small', 'sitting',
'standing', 'perched', 'blue',
'branch', 'top', 'grass', 'black', 'little',
'wooden', 'colorful', 'brown',
'water', 'white', 'green', 'red'],
'captions': [{'text': 'a small blue bird
perched on a branch', 'confidence':
0.9450701151922265}]}
```

```
description': {'tags': ['person',
'cake', 'birthday', 'group', 'indoor',
'table', 'child', 'party', 'adult',
'gathered', 'photo', 'sitting', 'food',
'girl', 'posing', 'large', 'standing',
'front', 'small', 'young', 'people',
'family', 'bunch', 'long', 'little',
'holding', 'cutting', 'man',
'restaurant', 'pizza', 'cut', 'room',
'crowd'], 'captions': [{'text': 'a
group of people sitting at a table
with a birthday cake', 'confidence':
0.8776280326294233}]},
```

図 1 Computer Vision API の応答例

ではまだ Custom vision API はプレビューですが、  
 https://www.customvision.ai/ にアクセスすると Microsoft ア  
 カウントで無料で利用することができます。訓練画像が少  
 ない場合には試してみる価値があると思います。

著者らは、シイタケとカキシメジの画像をそれぞれ 5 枚  
 ずつ入力し分類器を作成してみました。カキシメジはシイ  
 タケと似ていますが毒があり、シイタケよりも少し柿色を  
 しています。図 2 の上段に訓練画像を示します。左の上下  
 がカキシメジで右がシイタケです。白黒の紙面だと分から  
 ないかもしれませんので、ウェブ版をご覧ください。図 2  
 の上段は訓練に使ったものの一部、下段はテスト画像です。  
 API は左のカキシメジを 100%の確信度でカキシメジだと



図2 左:カキシメジ 右:シイタケ

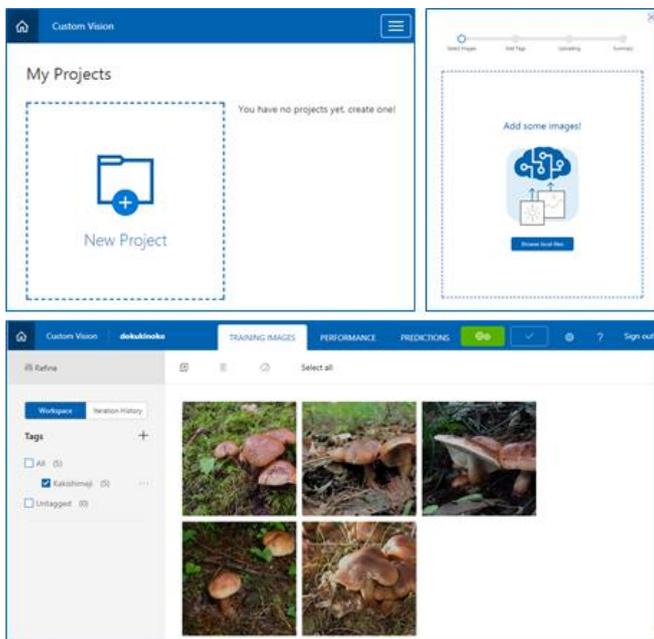


図3 Custom vision API に画像を追加している様子

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

答えました。これは正しく答えられた一方、図2の下段右のシイタケを 99.9%の確信度でシイタケだと答えました。カキシメジとシイタケの画像3枚ずつでテストしたところ全てのカキシメジは正しく答えた一方、シイタケの正解は1枚でした。したがって精度はおおよそ 71.4%となりました。下記に使用する際の手順を紹介します。ログインした画面で、図3のように新規プロジェクトを作成し、適当なプロジェクト名をつけ、分類したいカテゴリを一般、食べ物、ランドマークなどから選択し、Create project をクリックします。Add images を選択すると、画像を入力するダイアログになるので、Browse local files から画像を選択し、タグを入力して+ボタンを押すとタグが追加されます。(複数のタグ付けが可能です。)最後に upload \* files をクリックすると、タグ付きの画像を追加できます。分類するクラスの数だけ、Add images を繰り返します。画像の入力が終わったら、Train ボタンを押します。訓練が終わると、この分類器の訓練データにおける Precision と

23 Recall が出力されます。満足するまで画像を追加すれば精度を上げることができます。Quick Test をクリックし、画像の URL やローカルの画像をアップロードすることで、訓練した分類器の性能をチェックすることができます。プロジェクト作成時に General (compact) などを選ぶと、Android Tensorflow や IOS ヘモデルをエクスポートすることができます。

#### 4. LUIS API

31 LUIS (Language Understanding Intelligent Service) API は、入力された文章からユーザーの目的 (インテント) を解釈したり、文章からエンティティ (たとえば、人や場所、商品、など) を抽出したりすることができます。LUIS を使うことで、人間にとっては自然言語処理的な柔軟さで、機械とインタラクションが行えるようになります。この API の機能を概観するには、[3] (Azure LUIS など検索すると出てきます) をご覧頂くのが良いです。

39 このページの最初にもありますが、たとえば図4で示したように、LUIS を使えば卓上ライトやフロアランプのオンオフの切り替えを、自然言語の柔軟さで制御することができます。このような自然言語による制御を行うためには、事前にエンティティやインテントを定義しておく必要があります。そのようなモデルは、Microsoft アカウントがあれば <http://www.luis.ai> へアクセスして無料で作成することができます。以下では、このサンプルの作り方を示します。

47 Luis.ai へアクセスしたら、Create new app を選択し、アプリケーション名を入力し、言語を選択します (日本語もあります)。作成された app を選択すると、インテントが作成できます。Create new intent か Add prebuilt domain intent を選択しますが、既に Pre-built intent で近いものがあれば、これを選択した方が高い性能を得られます。無ければ、自分で作成します。図4のようなライトの制御であれば、HomeAutomation.TurnOn と、HomeAutomation.TurnOff を選択します。これらをクリックすると、使用例 (Utterance) のリストが見られます。

次に、左側のペインでエンティティを選択すると、エンティティの設定ができます。こちらでも Add prebuilt entities や Create new entity でエンティティを追加します。ライトの制御であれば、HomeAutomation.Room, HomeAutomation.Operation, HomeAutomation.Device を選択します。それぞれをクリックしてみると、使用例のリストが出てきます。エンティティを追加したことで、インテントも Turn [HomeAutomation.Device] on や set [HomeAutomation.Device] [HomeAutomation.Operation] などのような形に変化しています。これで右上の Train をクリックすると、訓練ができます。作成した Application を外部から使用するには、Publish する必要があります。右上の Publish ボタンをクリックし、Publish to production slot をク

1 リックします。Publish が終わると、下部に表示された 49  
 2 Endpoint や Key で利用できるようになります。 50  
 3 インテントやエンティティを自分で定義したい場合の例 51  
 4 を示します。Microsoft のチュートリアルを参考にしていま 52  
 5 す[4]。LUIS で、例えば MyCommunicator を作成してみま 53  
 6 す。SendMessage というインテントを作成します。インテ 54  
 7 ントを作成したら、使用例 (Utterance) を入力する必要が 55  
 8 あります。Utterance の例は下記の通りです。 56  
 9 Reply with I got your message, I will have the answer tomorrow 57  
 10 Send message of When will you be home? 58  
 11 Text that I am busy 59  
 12 Tell them that it needs to be done today 60  
 13 IM that I am driving and will respond later 61  
 14 Compose message to David that says When was that? 62

15 say greg hello  
 16 次に、None というインテントを作成します。App が何も  
 17 返答する必要がない場合やユーザが会話を終わらせたい場  
 18 合を意味します。Utterance は下記のようなものになります。  
 19 Cancel!  
 20 Good bye  
 21 What is going on?

22 最後に Message というエンティティを作成します。Intent  
 23 の SendMessage を選択し、Reply with I got your message, I  
 24 will have the answer tomorrow の Utterance で、I から tomorrow  
 25 までを選択し、図 5 のように、Create new entity を選択しま  
 26 す。Entity type は Simple とし、Entity の名前は Message と  
 27 します。同様に、全ての Utterance で Message の指定を繰り返  
 28 します。Text that I am busy の例では、I am busy が、say greg  
 29 hello では、hello が該当します。最後に Train ボタンを押し、  
 30 訓練を行うことができます。アプリケーションでは、text  
 31 I'm driving and will be 30 minutes late to the meeting というク  
 32 エリをテストしてみると、SendMessage というインテント  
 33 が抽出され、I'm driving and will be 30 minutes late to the  
 34 meeting が Message として認識されています。チュートリ 63  
 35 アルでは、Bot と接続する方法なども解説されており、簡 64  
 36 単に ChatBot を作成することができます。 65

## 5. むすび

38 深層学習によって、様々な分野で高い性能を持つ認識手  
 39 法が登場し、また Microsoft Cognitive Services のような API  
 40 の登場によりこれらが手軽に利用できるようになっていま  
 41 す。皆様のアプリケーション開発に役立てて頂ければと願  
 42 っています。

43  
 44 (文末) 66  
 45 67  
 46 文献 68  
 47 [1] <https://azure.microsoft.com/ja-jp/services/cognitive-services/> 69  
 48 computer-vision/ 70

[2] <https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/quickstarts/python>  
 [3] <https://azure.microsoft.com/ja-jp/services/cognitive-services/language-understanding-intelligent-service/>  
 [4] <https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-quickstart-primary-and-secondary-data>

## 著者紹介

川上玲 (かわかみれい)

2008 年東京大学大学院博士了 (情報理工学)。

東大生産研などを経て 2018 年より東京大学特任講師。

コンピュータビジョンに関する研究に従事。

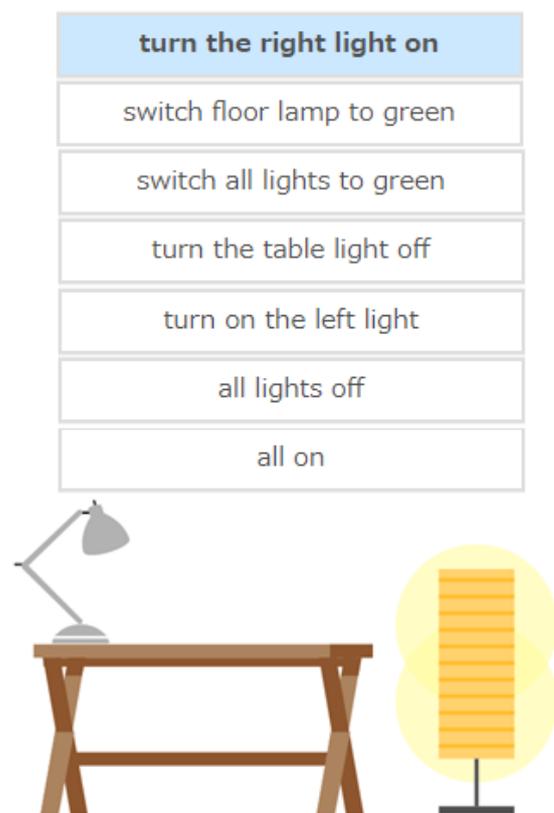


図 4 ライトの制御

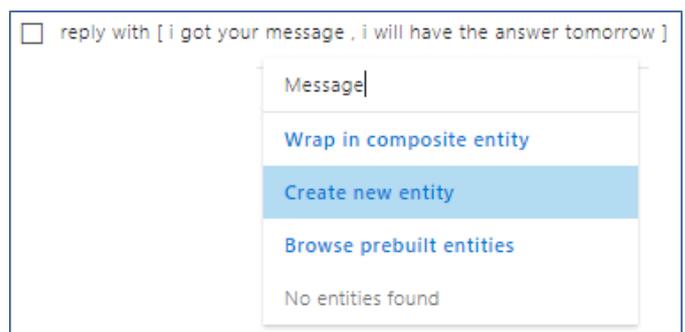


図 5 Utterance に Entity を入力