Image recognition system using IBM Cloud Visual Recognition

PROJECT AIM:

Creating an image recognition system using IBM Cloud Visual Recognition for the purpose of enhancing visual storytelling is a great idea. Here's a high-level overview of the steps involved in such a project:

Project Setup:

Create an IBM Cloud account if you don't already have one.

Access IBM Cloud Visual Recognition service and create an instance.

Data Collection:

Gather a diverse dataset of images for training. This dataset should cover a wide range of categories relevant to your application.

Data Preprocessing:

Clean and preprocess the image data. This may involve resizing, normalizing, and augmenting the images to improve model performance.

Training the Model:

Use the IBM Cloud Visual Recognition service to train a custom image classification model using your prepared dataset.

Fine-tune the model to improve accuracy.

Integration:

Develop a user-friendly web interface or mobile app where users can upload images.

Image Classification:

Implement the integration with the IBM Cloud Visual Recognition API to analyze and classify uploaded images.

Extract the top classes or labels associated with the image.

Generating Descriptions:

To enhance the visual storytelling aspect, you can integrate a natural language generation (NLG) model or service to generate AI-powered captions or descriptions for the images.

This might involve using models like GPT-3 to generate descriptive text that corresponds to the recognized objects in the image.

User Interaction:

Design the user interface to display both the image and the generated captions.

Allow users to edit or customize the generated captions if desired.

Feedback and Improvement:

Implement a feedback mechanism where users can provide feedback on the accuracy of the generated captions. This data can be used to continuously improve the model.

Scaling and Optimization:

As your platform gains users and more data, ensure that your system can scale accordingly.

Continuously monitor and optimize the performance of both the image recognition and caption generation components.

Security and Privacy:

Ensure that user data and uploaded images are handled securely and with respect to privacy regulations.

Testing and Evaluation:

Thoroughly test your system with different types of images to ensure accurate classification and meaningful captions.

Deployment:

Deploy your platform to a reliable hosting environment.

Documentation and Support:

Provide documentation and support for users to understand how to use your platform effectively.

Marketing and User Engagement:

Promote your platform to your target audience and engage with users to gather feedback and improve the platform continually.

Monitoring and Maintenance:

Implement monitoring tools to keep an eye on system performance and address any issues promptly.

Regularly update and retrain your image recognition model to adapt to new trends and categories.

Remember that creating a successful image recognition and captioning platform is an iterative process that involves continuous improvement and user feedback. Make sure to keep the user experience at the forefront of your development efforts to create a compelling and engaging platform for visual storytelling.

Phase 2: Innovation in Image Recognition

Introduction:

In the world of image recognition, innovation is the key to unlocking the full potential of visual content analysis. This phase explores groundbreaking features that will elevate your system, setting the stage for a future where images are understood in new and profound ways.

1. Multi-Modal Recognition: Combine image recognition with other modalities like audio or text for a richer understanding of content. This could enable your system to analyze images in the context of spoken or written descriptions.

2. Real-time Recognition: Implement real-time image recognition for applications like augmented reality, where users can point their device at objects, and your system can provide instant information or descriptions.

3. Personalization: Develop a recommendation system that learns from user interactions and provides personalized image recommendations or captions based on individual preferences.

4. Visual Search: Implement a visual search feature, allowing users to search for images using images as queries. This can be particularly useful for e-commerce applications.

5. AR and VR Integration: Explore integration with augmented reality (AR) and virtual reality (VR) platforms to create immersive experiences where objects in the real world are recognized and augmented with digital content.

6. Semi-Supervised Learning: Investigate semi-supervised or self-supervised learning techniques to reduce the amount of labeled data required for training and make your system more adaptable.

7. Multilingual Support: Enhance the system's language capabilities to support multiple languages for image captions and descriptions.

8. **Emotion Recognition**: Incorporate emotion recognition in images, allowing your system to describe not only objects but also the emotions or sentiments conveyed in the visual content.

9. **Automated Content Tagging**: Develop an automated tagging system that can recognize and tag images with relevant keywords or labels, making them easier to search and organize.

10. **AI Ethics and Bias Mitigation**: Invest in research and technologies that address bias in AI systems, ensuring that your image recognition system is fair and ethical.

11. **Collaborations and Partnerships**: Explore partnerships with other AI and tech companies for cutting-edge innovations or access to new datasets.

12. **AI-Enhanced Image Editing**: Create a feature for users to edit images directly in your platform using AI-powered tools, such as background removal or style transfer.

13. **Predictive Analytics**: Implement predictive analytics to forecast trends in visual content or recommend future image needs for businesses and content creators.

14. **Energy Efficiency**: Investigate energy-efficient AI models and deployment methods to reduce the carbon footprint of your system.

15. **Blockchain for Image Ownership**: Consider using blockchain technology to establish and verify the ownership of images, particularly useful for photographers and artists.

Examples:

1. **Data Collection and Preprocessing (Python with OpenCV):**
   - Program to scrape and collect images from the web based on user-defined categories.
   - Script to resize, normalize, and augment the images in your dataset for training.

```python
import cv2
import os

# Data collection
```

```
# Implement web scraping tools like BeautifulSoup or Scrapy to collect images.

# Data preprocessing
def preprocess_image(image_path, output_path, target_size=(224, 224)):
    image = cv2.imread(image_path)
    image = cv2.resize(image, target_size)
    image = (image- 128) / 128  # Normalize
    cv2.imwrite(output_path, image)

# Augmentation can be done using libraries like Augmentor.
```

2. **Training the Model (IBM Cloud Visual Recognition API):**
  - Use the IBM Cloud Visual Recognition API to train a custom model.

python

```
# Example using Python SDK for IBM Visual Recognition
from ibm_watson import VisualRecognitionV3
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

# Set up the API client
authenticator = IAMAuthenticator('your_api_key')
visual_recognition = VisualRecognitionV3(
    version='2018-03-19',
    authenticator=authenticator
)

# Train the custom model
with open('your_training_data.zip', 'rb') as training_data:
    model = visual_recognition.create_classifier(
        name='custom_model',
        positive_examples=training_data
    ).get_result()
```

3. **Innovative Features (Real-time Recognition in Python):**
  - Implement real-time image recognition using OpenCV.

python

```
import cv2
import requests
import json

# Real-time recognition
def real_time_recognition():
    cap = cv2.VideoCapture(0)  # Open the camera
    while True:
        ret, frame = cap.read()

        # Preprocess the frame if needed
        # ...

        # Send the frame to IBM Visual Recognition for classification
        # (Make API requests with the frame)
        response = requests.post(
            'https://api.us-south.visual-recognition.watson.cloud.ibm.com/.../classify',
            files={'image': ('frame.jpg', frame)},
            params={'version': '2018-03-19'},
            auth=('apikey', 'your_api_key')
        )

        # Process the response for real-time display
        result = json.loads(response.text)
        # ...

        # Display the result on the frame
        # ...

        cv2.imshow('Real-time Recognition', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Conclusion:

Innovating in image recognition has the power to redefine the way we interact with visual content. By implementing these advanced features and technologies, your image recognition system is poised to lead in accuracy, user experience, and ethical AI practices. Remember, innovation is an ongoing journey – keep learning, adapting, and engaging with users to stay at the forefront of this dynamic field. The future of visual content analysis is in your hands.

Cloud Application Development

Image Recognition with IBM Cloud Visual Recognition

Phase-3

Start building the image recognition system using IBM Cloud Visual Recognition. Create an IBM Cloud account, set up the Visual Recognition service, and obtain API keys.

Design a simple web interface where users can upload images and view the AI-generated captions.

Create an IBM Cloud account, set up the Visual Recognition service, and obtain API keys.

IBM Cloud Account:

After creating the IBM Cloud Account, login with the IBMid.

Enter your IBMid then click continue.

Enter your password, then click login.

The homepage of the IBM account as follows:

Set up the Visual Recognition service, and obtain API keys.

Set up the Visual Recognition service:

Open the homepage of your IBM account.

Click on the manage.

Click on the Access(IAM).

Then click on the Service IDs.

Click on Create button, to create the service.

Enter the name and the description for the service, then click on Create button to create the service.

Visual Recognition service has been created.

Then click on the API keys, to obtain the API key for the Visual Recognition Service.

Then enter the name and the description to create the API key and click on the Create button.

The API key has been successfully created for the Visual Recognition Service.

The API key has obtained.

Visual Recognition Service document.

Open the home page of your IBM account, click on the help.

Then click on the Docs to open the document of the Visual Recognition

Click on the search button at the drop-down menu, near the user account.

Then search for the Visual Recognition.

Open the DocumentAnalytics.VisualRecognitionClassifier actions

The main page of the document as follows.

VisualRecogClassify:

IBM Documentation — Search in IBM Datacap 9.1.5

IBM Datacap

Change version: 9.1.5

Show full table of contents

Filter on titles

DocumentAnalytics.VisualRecognitionClas actions
- VisualRecogClassify
- VisualRecogSetCredentials
- VisualRecogSetMinConfidence
- VisualRecogSetURL
- VisualRecogTrain

DocumentAnalytics.NaturalLanguageClassifi actions

Email actions

Equalize actions

Ewsmail actions

All products / IBM Datacap / 9.1.5 /

Was this topic helpful?

VisualRecogClassify

Last Updated: 2021-03-04

Classifies image by using IBM Watson Visual Recognition API.

Syntax

```
bool VisualRecogClassify (string ClassifierName)
```

Parameters

string **ClassifierName** - Name of the Classifier to be used

Returns

True, if the action succeeds. Otherwise, **False**

Level

Page level.

IBM Documentation — Search in IBM Datacap 9.1.5

IBM Datacap

Change version: 9.1.5

Show full table of contents

Filter on titles

DocumentAnalytics.VisualRecognitionClas actions
- VisualRecogClassify
- VisualRecogSetCredentials
- VisualRecogSetMinConfidence
- VisualRecogSetURL
- VisualRecogTrain

DocumentAnalytics.NaturalLanguageClassifi actions

Email actions

Equalize actions

Ewsmail actions

Page level.

Details

This action identifies a page by using the Watson™ Recognition technology. This technology analyzes the image-based full page and attempts to find match within the classes that have been defined for the selected classifier. If a match is found, the Page type is populated with the ID of the category that was matched.

If a match is not found, the page type is set to "Other".

When classification is complete, a list of matches and their confidence values are stored in the "MatchingCategoryX" and "MatchingCategoryConfX" variables. The number of matches is stored in the variable "MatchingCategoriesCount".

To run classification without updating the page type, set the variable "UpdateDCOType" to "0" before calling this action. In this case, classification does not update the page type, but the variables that are mentioned above is populated.

Because this action supports image files (.jpg, .jpeg or .png) only. ConvertToJPEG must be called before using the Classify action for other types of files.

The ClassifierName can be a smart parameter.

Example

```
VisualRecogSetURL("@APPVAR(values/gen/url)")
VisualRecogSetCredentials("@APPVAR(values/adv/VRAPIKey)")
VisualRecogClassify("@APPVAR(values/gen/VRClassifierName)")
```

Note: VisualRecogSetURL action is not included if your IBM Visual Recognition token is created before May 2018.

VisualRecogSetCredentials:

IBM Documentation — Search in IBM Datacap 9.1.5

IBM Datacap

Change version: 9.1.5

Show full table of contents

Filter on titles

DocumentAnalytics.VisualRecognitionClas actions
- VisualRecogClassify
- VisualRecogSetCredentials
- VisualRecogSetMinConfidence
- VisualRecogSetURL
- VisualRecogTrain

DocumentAnalytics.NaturalLanguageClassifi actions

Email actions

Equalize actions

Ewsmail actions

All products / IBM Datacap / 9.1.5 /

Was this topic helpful?

VisualRecogSetCredentials

Last Updated: 2021-03-04

Sets the credentials to be used to do the classification.

Syntax

```
bool VisualRecogSetCredentials (string APIKey)
```

Parameters

string **APIKey** can be Smart Parameters.

Returns

True, if the action succeeds. Otherwise, **False**.

Level

Batch level.

IBM Documentation — Search in IBM Datacap 9.1.5

IBM Datacap

Change version: 9.1.5

Show full table of contents

Filter on titles

DocumentAnalytics.VisualRecognitionClas actions
- VisualRecogClassify
- VisualRecogSetCredentials
- VisualRecogSetMinConfidence
- VisualRecogSetURL
- VisualRecogTrain

DocumentAnalytics.NaturalLanguageClassifi actions

Email actions

Equalize actions

Ewsmail actions

True, if the action succeeds. Otherwise, **False**.

Level

Batch level.

Details

This action sets the credentials to be used to do the classification. The APIKey can be Smart Parameters.

Example

```
VisualRecogSetURL("@APPVAR(values/gen/url)")
VisualRecogSetCredentials("@APPVAR(values/adv/VRAPIKey)")
VisualRecogClassify("@APPVAR(values/gen/VRClassifierName)")
```

Note: VisualRecogSetURL action is not included if your IBM Visual Recognition token is created before May 2018.

Parent topic:

→ DocumentAnalytics.VisualRecognitionClassifier actions

While IBM values the use of inclusive language, terms that are outside of IBM's direct influence, for the sake of maintaining user understanding, are sometimes required. As other industry leaders join IBM in embracing the use of inclusive language, IBM will

© Copyright IBM Corporation

VisualRecogSetMinConfidence:

IBM Documentation — Search in IBM Datacap 9.1.5

IBM Datacap

Change version: 9.1.5

Show full table of contents

Filter on titles

DocumentAnalytics.VisualRecognitionClassifier actions
- VisualRecogClassify
- VisualRecogSetCredentials
- VisualRecogSetMinConfidence
- VisualRecogSetURL
- VisualRecogTrain

DocumentAnalytics.NaturalLanguageClassifier actions

Email actions

Equalize actions

Ewsmail actions

All products / IBM Datacap / 9.1.5 /

Was this topic helpful?

VisualRecogSetMinConfidence

Last Updated: 2021-03-04

Sets the minimum confidence score for classification matching.

Syntax

```
bool VisualRecogSetMinConfidence (string MinScore)
```

Parameters

string **MinScore** - Minimum score for classification matching. Valid values are fractional values between zero and one (for example: 0.0 and 1.0).

Returns

True, if the parameter value is between the valid range of zero to one (0.0 and 1.0) Otherwise, **False**.

Level

IBM Documentation — Search in IBM Datacap 9.1.5

IBM Datacap

Change version: 9.1.5

Show full table of contents

Filter on titles

DocumentAnalytics.VisualRecognitionClassifier actions
- VisualRecogClassify
- VisualRecogSetCredentials
- VisualRecogSetMinConfidence
- VisualRecogSetURL
- VisualRecogTrain

DocumentAnalytics.NaturalLanguageClassifier actions

Email actions

Equalize actions

Ewsmail actions

All level.

Details

When Classify searches for a classification match, a score between zero (no match) and one (a positive match) is calculated. This action sets the minimum score that a match must be considered a match. Any matches with a score less than the value specified is rejected. With this action, you can control the tolerance for documents matching an existing example.

When setting up the parameter in your application, use the decimal character from the system locale that is defined for the application in the Taskmaster Application Manager. For example, when the decimal character is a period, use a value from 0.0 to 1.0. When the decimal character is a comma, use a value in the range 0,0 - 1,0.

The MinScore can be a Smart Parameter. This action must be called before the Classify action.

Example

```
VisualRecogSetCredentials("@APPVAR(values/adv/VRAPIKey)")
VisualRecogSetMinConfidence(0.9)
VisualRecogClassify("@APPVAR(values/gen/VRClassifierName")
```

Parent topic:

→ DocumentAnalytics.VisualRecognitionClassifier actions

VisualRecogSetURL:

IBM Documentation — Search in IBM Datacap 9.1.5

IBM Datacap — Change version: 9.1.5

Show full table of contents

Filter on titles

DocumentAnalytics.VisualRecognitionClassifier actions
- VisualRecogClassify
- VisualRecogSetCredentials
- VisualRecogSetMinConfidence
- VisualRecogSetURL
- VisualRecogTrain

DocumentAnalytics.NaturalLanguageClassifier actions
- Email actions
- Equalize actions
- Ewsmail actions

All products / IBM Datacap / 9.1.5 /

Was this topic helpful?

VisualRecogSetURL

Last Updated: 2021-03-04

Updates the new URL to be used to do the classification.

Syntax

```
VisualRecogSetURL (string url)
```

Parameters

The url can be Smart Parameters

Returns

True, if the action succeeds. Otherwise, **False**

Level

Batch level.

IBM Documentation — Search in IBM Datacap 9.1.5

IBM Datacap — Change version: 9.1.5

Show full table of contents

Filter on titles

DocumentAnalytics.VisualRecognitionClassifier actions
- VisualRecogClassify
- VisualRecogSetCredentials
- VisualRecogSetMinConfidence
- VisualRecogSetURL
- VisualRecogTrain

DocumentAnalytics.NaturalLanguageClassifier actions
- Email actions
- Equalize actions
- Ewsmail actions

Batch level.

Details

This action updates the new URL to be used to do the classification. IBM Visual Recognition token created after May 2018 must apply this action before VisualRecogSetCredentials().

Example

```
VisualRecogSetURL("@APPVAR(values/gen/url)")
        VisualRecogSetCredentials("@APPVAR(values/adv/VRAPIKey)")
        VisualRecogClassify("@APPVAR(values/gen/VRClassifierName)")
```

Parent topic:

→ DocumentAnalytics.VisualRecognitionClassifier actions

While IBM values the use of inclusive language, terms that are outside of IBM's direct influence, for the sake of maintaining user understanding, are sometimes required. As other industry leaders join IBM in embracing the use of inclusive language, IBM will continue to update the documentation to reflect those changes.

© Copyright IBM Corporation 2018

Contact IBM | Privacy | Terms of use | Accessibility | English

IBM Documentation Help

VisualRecogTrain:

IBM Documentation — Search in IBM Datacap 9.1.5

IBM Datacap — Change version: 9.1.5

All products / IBM Datacap / 9.1.5 /

VisualRecogTrain

Last Updated: 2021-03-04

Creates or replaces a Visual Recognition Classifier.

Syntax

```
bool VisualRecogTrain (string ZIPDirectory, string Name, string deleteExisting)
```

Parameters

string **ZIPDirectory** - Directory where this action stores the zip file with the training data. The zip file has the name of the classifier. This parameter supports Smart Parameters.

string **Name** - Name to give the classifier. This name is used when you later try to classify a page. This parameter supports Smart Parameters.

string **deleteExisting** - Delete the classifier of the same name if it exists. If this flag is set to true, it deletes the classifier of the same name, if it exists. If this flag is set to false and a classifier with the same name exists, this action adds the training data to the existing classifier. This parameter supports SmartParameters. The type of this parameter is string only to support SmartParameters. Internally, it is treated as a boolean. Specify '1' for true and '0' for false. Any other value is ignored and the action is default to false.

IBM Documentation — Search in IBM Datacap 9.1.5

IBM Datacap — Change version: 9.1.5

Level

Batch level.

Details

This action creates or replaces a Visual Recognition Classifier. Pages of the batch are used as the training data. It is expected that the batch is in a certain format. In particular, the batch needs to be divided into set documents, where the 'Type' associated with the document is class that is created in the Visual Recognition Service. All of pages in that document are treated as training data for that class. So, for example, if you wanted to train the classifier to recognize the main page and the trailing pages of a mortgage application, you would create a batch with two documents: Main_Page and Trailing_Page. Put all of the training page for the main page under the Main_Page document and the trailing pages under Trailing_Page document.

Example

```
VisualRecogSetURL("@APPVAR(values/gen/url)")
VisualRecogSetCredentials("@APPVAR(values/adv/VRAPIKey)")
VisualRecogTrain("@APPPATH(runtime)+\+..+\+ZIP","@APPVAR(values/gen/VRClassifierName)",1)
```

Parent topic:

→ DocumentAnalytics.VisualRecognitionClassifier actions

While IBM values the use of inclusive language, terms that are outside of IBM's direct influence, for the sake of maintaining user

© Copyright IBM Corporation

Cloud Application Development

Image Recognition with IBM Cloud Visual Recognition

Phase-4

Continue building the image recognition system by integrating IBM Cloud Visual Recognition and AI-generated captions.

Implement the image classification process using the IBM Cloud Visual Recognition API.

Use natural language generation to create captions for the recognized images.

Login to the IBM account with IBMid.

Enter password to login.

Home page of the IBM account.

Click on the help icon and then click on Docs in the drop-down menu.

Then click the search icon.

Then search for Visual Recignition.

Then open the DocumentAnalytics.VisualRecognitionClassifier actions.

DocumentAnalytics.VisualRecognitionClassifier actions.

DocumentAnalytics.NaturalLanguageClassifier actions

NLCClassify

Identifies a page by using the IBM Natural Language technology.

IBM Datacap
Change version
9.1.5
Show full table of contents
Filter on titles
VisualRecogSetCredentials
VisualRecogSetMinConfidence
VisualRecogSetURL
VisualRecogTrain
DocumentAnalytics.NaturalLangu actions
NLCClassify
NLCClassifyText
NLCSetCredentials
NLCSetLanguage
NLCSetMinConfidence
NLCTrain

NLCClassify

Last Updated: 2021-03-04

Identifies a page by using the IBM Natural Language technology.

Syntax

```
bool NLCClassify (string ClassifierName)
```

Parameters

string **ClassifierName** - Name of the Classifier to be used.

Returns

True, action is successful. Otherwise, **False**..

Level

Page level.

IBM Datacap
Change version
9.1.5
Show full table of contents
Filter on titles
VisualRecogSetCredentials
VisualRecogSetMinConfidence
VisualRecogSetURL
VisualRecogTrain
DocumentAnalytics.NaturalLangu actions
NLCClassify
NLCClassifyText
NLCSetCredentials
NLCSetLanguage
NLCSetMinConfidence
NLCTrain

True, action is successful. Otherwise, **False**..

Level

Page level.

Details

This action identifies a page by using the IBM Natural Language technology. This technology analyzes the full text of pages and attempts to find match within the classes that have been defined for the selected classifier. If a match is found, the Page type is populated with the ID of the category that was matched.

If a match is not found, the page type is set to "Other".

When classification is complete, a list of matches and their confidence values are stored in the "MatchingCategoryX" and "MatchingCategoryConfX" variables. The number of matches is stored in the variable "MatchingCategoriesCount" .

To run classification without updating the page type, set the variable "UpdateDCOType" to "0" before calling this action. In this case classification will not update the page type, but the variables that are mentioned above will still be populated.

Because the matching relies on a page's full text, a full page recognition action must be called before using the Classify action.

This action gets the text for the page in the following order:

- from the layout.xml that is file generated by the Recognize action
- from the .txt file that is generated by the RecognizeToFile action
- from the .cco file that is generated by the RecognizePage action

The **ClassifierName** can be a smart parameter.

Example

NLCClassifyText

Classifies the specified text by using the IBM Natural Language technology.

[NLCSetCredentials]

Sets the credentials to be used to do the classification.

[NLCSetLanguage]

Sets the language of the page to be classified.

IBM Datacap
Change version
9.1.5
Show full table of contents
Filter on titles
VisualRecogSetCredentials
VisualRecogSetMinConfidence
VisualRecogSetURL
VisualRecogTrain
DocumentAnalytics.NaturalLangu actions
NLCClassify
NLCClassifyText
NLCSetCredentials
NLCSetLanguage
NLCSetMinConfidence
NLCTrain

```
bool NLCSetLanguage (string LanguageCode)
```

Parameters

string **LanguageCode** - The two-letter code associate with the language of the page to be classified

Returns

True, if the action succeeds. Otherwise, **False**.

Level

All level.

Details

This action sets the language of the page to be classified. The LanguageCode can be a smart parameter. The possible values for the Language code are:

- 'en' : English
- 'ar' : Arabic
- 'fr' : French
- 'de' : German
- 'it' : Italian
- 'ja' : Japanese
- 'ko' : Korean
- 'pt' : Portuguese (Brazilian)

ibm cloud login - Search | IBM Cloud | NLCSetLanguage - IBM Docume | IBM Cloud Docs
https://www.ibm.com/docs/en/datacap/9.1.5?topic=actions-nlcsetlanguage
IBM Documentation Announcements IBM Redbooks
IBM Datacap
Change version
9.1.5
Show full table of contents
Filter on titles
VisualRecogSetCredentials
VisualRecogSetMinConfidence
VisualRecogSetURL
VisualRecogTrain
DocumentAnalytics.NaturalLangu actions
NLCClassify
NLCClassifyText
NLCSetCredentials
NLCSetLanguage
NLCSetMinConfidence
NLCTrain

- 'ja' : Japanese
- 'ko' : Korean
- 'pt' : Portuguese (Brazilian)
- 'es' :Spanish

This action must be called before the Classify action.

Example

```
NLCSetLanguage("en")
NLCSetCredentials("@APPVAR(values/gen/NLCUserName)","@APPVAR(values/adv/NLCPassword)")
NLCSetMinConfidence(0.9)
Recognize()
NLCClassify("@APPVAR(values/gen/NLCClassifierName)")
```

Parent topic:

→ DocumentAnalytics.NaturalLanguageClassifier actions

While IBM values the use of inclusive language, terms that are outside of IBM's direct influence, for the sake of maintaining user understanding, are sometimes required. As other industry leaders join IBM in embracing the use of inclusive language, IBM will continue to update the documentation to reflect those changes.

© Copyright IBM Corporation 2018

NLCSetMinConfidence

Sets the minimum confidence score for classification matching.

[NLCTrain]

Creates or replaces an NLC Classifier.

IBM Datacap
Change version
9.1.5
Show full table of contents
Filter on titles
VisualRecogSetCredentials
VisualRecogSetMinConfidence
VisualRecogSetURL
VisualRecogTrain
DocumentAnalytics.NaturalLanguactions
NLCClassify
NLCClassifyText
NLCSetCredentials
NLCSetLanguage
NLCSetMinConfidence
NLCTrain

Syntax

```
bool NLCTrain (string CSVDirectory,string Name, string deleteExisting, string appendToCSV)
```

Parameters

string CSVDirectory - Directory where this action stores the csv file with the training data. CSV file has the name of the classifier. This parameter supports SmartParameters.

string Name - Name to give the classifier. This name is used when you later try to classify a page. This parameter supports SmartParameters.

string deleteExisting - Delete the classifier of the same name if it exists. If this flag is set to true, it deletes the classifier of the same name, if it exists. If this flag is set to false and a classifier with the same name exists, this action returns false. This parameter supports SmartParameters. The type of this parameter is string only to support SmartParameters. Internally, it is treated as a boolean. Specify '1' for true and '0' for false. Any other value is ignored and the action is default to false.

string appendToCSV - If an existing csv file found, this parameter indicates whether the training data from this batch should be appended to it or that it should be overwritten. If true, the classifier is trained with the data in the csv file and the new data from this batch. If false, the existing data in the csv file is overwritten and the classifier is trained with just the data in this batch. This parameter supports SmartParameters. The type of the parameter is string only to support SmartParameters. Internally, it is treated as a boolean. Specify '1' for true and '0' for false. Any other value is ignored and the action is default to true.

Returns

True, action is successful. Otherwise, False.

Level

ibm cloud login - Search | IBM Cloud | NLCTrain - IBM Documentation | IBM Cloud Docs
https://www.ibm.com/docs/en/datacap/9.1.5?topic=actions-nlctrain
IBM Documentation Announcements IBM Redbooks
IBM Datacap
Change version
9.1.5
Show full table of contents
Filter on titles
VisualRecogSetCredentials
VisualRecogSetMinConfidence
VisualRecogSetURL
VisualRecogTrain
DocumentAnalytics.NaturalLanguactions
NLCClassify
NLCClassifyText
NLCSetCredentials
NLCSetLanguage
NLCSetMinConfidence
NLCTrain

Returns

True, action is successful. Otherwise, False.

Level

Batch level.

Details

This action creates or replaces an NLC Classifier. Pages of the batch are used as the training data. It is expected that the batch is in a certain format. In particular, the batch needs to be divided into set documents where the 'Type' associated with the document is class that is created in NLC. All of pages in that document are treated as training data for that class. So, for example, if you wanted to train the classifier to recognize the main page and the trailing pages of a mortgage application, you would create a batch with two documents: Main_Page and Trailing_Page. Put all of the training page for the main page under the Main_Page document and the trailing pages under Trailing_Page document.

Example

```
NLCSetLanguage("en")
NLCSetCredentials("@APPVAR(values/gen/NLCUserName)","@APPVAR(values/adv/NLCPassword)")
NLCTrain("@APPPATH(runtime)+\+..+\+CSV","@APPVAR(values/gen/NLCClassifierName)",0,1)
```

Parent topic:

→ DocumentAnalytics.NaturalLanguageClassifier actions

V