


Cognitive Services Topic Deep Dive with Azure ML Anomaly Detection

with Demos/HOLs

Presenter: Micheleen Harris

 Microsoft

What you'll know at the end of this session

1. How to use the API Reference for Cognitive Services Entity Linking Intelligent Service (ELIS)
2. What Cognitive Services Text Analytics API detectors entail
3. How to interface programmatically with Azure ML Anomaly Detection
4. How to create a model with Cognitive Services Language Understanding and Intelligent Service (LUIS)



Cognitive Services Entity Linking Intelligent Service (ELIS)

Entity Linking Intelligent Service (ELIS)

For months, the four scientific instruments at the heart of the James Webb Space Telescope have been sealed in what looks like a huge pressure cooker. It's a test chamber that simulates the grueling operating conditions they will face after Webb is launched into orbit in 2018. But in fact, "pressure cooker" is an apt metaphor for the whole project. The infrared Webb observatory is the biggest, most complex, and most expensive science mission that NASA has ever attempted. Like that of its predecessor, the Hubble Space Telescope, Webb's construction has been plagued by redesigns, schedule slips, and cost overruns that have strained relationships with contractors, international partners, and supporters in the U.S. Congress. Lately the project has largely stuck to its schedule and its \$8 billion budget. But plenty could still go wrong, and the stakes are high: Both the future of space-based astronomy and NASA's ability to build complex science missions depend on its success.

GO ➔

Highlight Content

JSON

For months, the four scientific instruments at the heart of the James Webb Space Telescope have been sealed in what looks like a huge pressure cooker. It's a test chamber that simulates the grueling operating conditions they will face after Webb is launched into orbit in 2018. But in fact, "pressure cooker" is an apt metaphor for the whole project. The infrared Webb observatory is the biggest, most complex, and most expensive science mission that NASA has ever attempted. Like that of its predecessor, the Hubble Space Telescope, Webb's construction has been plagued by redesigns, schedule slips, and cost overruns that have strained relationships with contractors, international partners, and supporters in the U.S. Congress. Lately the project has largely stuck to its schedule and its \$8 billion budget. But plenty could still go wrong, and the stakes are high: Both the future of space-based astronomy and NASA's ability to build complex science missions depend on its success.

Sample 1

Sample 2

Sample 3

Docs: <https://www.microsoft.com/cognitive-services/en-us/entitylinking-api/documentation/overview>

Getting started: <https://www.microsoft.com/cognitive-services/en-us/entitylinking-api/documentation/GettingStarted>

ELIS – Identifying from context



ELIS Response

Mars is the fourth planet from the Sun and the second smallest planet in the Solar System, after Mercury. Named after the Roman god of war, it is often referred to as the "Red Planet" because the iron oxide prevalent on its surface gives it a reddish appearance.

GO>

Highlight Content

JSON

```
},
  "name": "De Graft",
  "wikipediaId": "Mercury (planet)",
  "score": 0.999
},
{
  "matches": [{
    "text": "Roman god of war",
    "entries": [{
      "offset": 122
    }]
  }],
  "name": "Mars",
  "wikipediaId": "Mars (mythology)",
  "score": 0.007
}]
```

(entity linking)

ELIS Demo

Obtaining the subscription and using the API Reference (follow along if you like):

Go to: <https://www.microsoft.com/cognitive-services/en-us/documentation>

Go to: My account (upper right corner)

Request new trials

Select "Entity Linking", Agree, Subscribe

Once you have a subscription to a service, if you go back to Documentation page, click ELIS drop-down

Go to: API Reference

This is where you'll find technical info like the Request URL, params, and response, etc. Plus code samples in many programming languages.

Cognitive Services Language Text Analytics API

Text Analytics - Detectors

- **Sentiment** - Is text positive or negative?
- **Key phrases** - What are people discussing in a single article?
- **Topics** - What are people discussing across many articles?
- **Languages** - What language is text written in?

Notes:

Language – 120 languages available as labels

Key phrase – talking points, English only currently

Topic detection – identify main topics over several articles, will need at least 100 documents as input

Nice quick start guide: <https://azure.microsoft.com/en-us/documentation/articles/cognitive-services-text-analytics-quick-start/>

Tip: for sentiment analysis it's a good idea to split text into sentences to gain higher precision.

For a Bot example leveraging text analytics see: <http://docs.botframework.com/en-us/bot-intelligence/language/#example-emotional-bot>


key phrases e.g. 'different country' and 'common phrases' in the sentence 'When I travel to a different country, I always like to try to learn how to say common phrases like hello and thank you.'

topic e.g. id main issues in user feedback forms

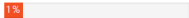
key phrase and sentiment currently support English, Spanish, German and Japanese

Text Analytics Examples

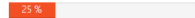
Example 1:

Language: ⓘ English (confidence: 100%)
Key phrases: ⓘ I had a wonderful experience! The rooms were wonderful and the staff were helpful.
Sentiment: ⓘ  98 %

Example 2:

Language: ⓘ English (confidence: 100%)
Key phrases: ⓘ I had a terrible time at the hotel. The staff were rude and the food was awful.
Sentiment: ⓘ  1 %

Example 3:

Language: ⓘ English (confidence: 100%)
Key phrases: ⓘ I had an okay stay at this hotel. The rooms were wonderful but the staff was rude.
Sentiment: ⓘ  25 %

Text Analytics Demo

<https://text-analytics-demo.azurewebsites.net/>

Another resource:

API Demo - <https://text-analytics-demo.azurewebsites.net/Home/SampleCode>

Also, check out the API Reference site for this API by clicking on this link on the Documentation page

Azure ML Anomaly Detection

General Azure ML docs: <https://azure.microsoft.com/en-us/documentation/services/machine-learning/>

Anomaly Detection with Azure Machine Learning APIs

When is it good to have an anomaly detection service?

Please, share your thoughts with the class.

When is it good to have an anomaly detection service?

Perhaps to watch out for:

- Too many login failures
- Spikes or dips in customer checkouts
- An increase in the dynamic range of file ingestion speeds
- in a cloud service
- An upward trend in system temperature

These are cases found from monitoring a system where a closer look may be called for. They are indicative of abnormal or anomalous behavior and could indicate a problem. The data could be streaming from a device or come from log files, but no matter the source an anomaly detection model could help predict when a system needs to be examined further.

The AML Anomaly Detection API Detection Categories

1. Positive and negative trends
2. Changes in the dynamic range of values
3. Spikes and Dips

These machine learning detectors track such changes in values over time and reports ongoing changes in their values as anomaly scores. They do not require adhoc threshold tuning and their scores can be used to control false positive rate.

Some more scenarios are:

service monitoring by tracking KPIs over time, usage monitoring through metrics such as number of searches, numbers of clicks, performance monitoring through counters like memory, CPU, file reads, etc. over time.

The AML Anomaly Detection API has Two Models

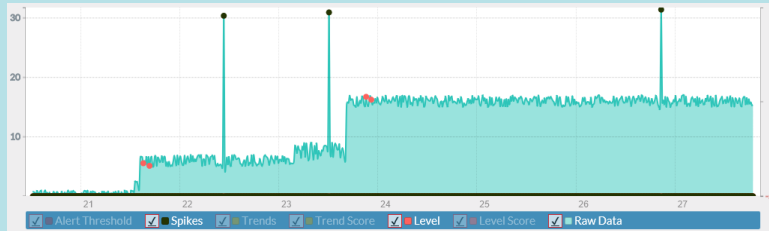
1. Score API
2. ScoreWithSeasonality API

- The Score API is used for running anomaly detection on non-seasonal time series data.
- The ScoreWithSeasonality API is used for running anomaly detection on time series that have seasonal patterns. This API is useful to detect deviations in seasonal patterns.

The APIs run a number of anomaly detectors on the data and returns their anomaly scores.

AML Anomaly Detection Score API

Figure representing common input data (time series, non-seasonal)



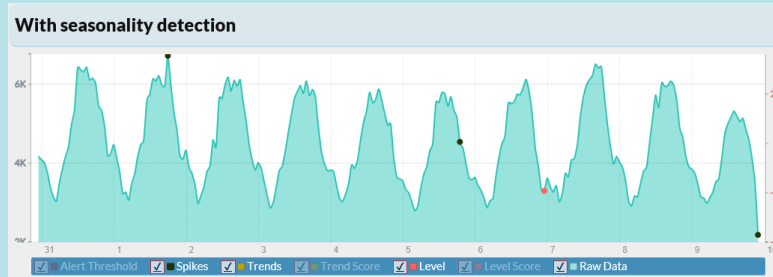
This time series has 2 distinct level changes, and 3 spikes. The red dots show the time at which the level change is detected, while the black dots show the detected spikes.

We'd use the

- Level change detectors
- Spike detectors

AML Anomaly Detection ScoreWithSeasonality API

Figure representing common input data (time series, seasonal)



The time series has one spike (the 1st black dot), two dips (the 2nd black dot and one at the end), and one level change (red dot).

We'd use the

- Level change detectors
- Spike detectors (it's actually for spikes *and* dips)

Note that both the dip in the middle of the time series and the level change are only discernable after seasonal components are removed from the series.

AML Anomaly Detection Lab

On Jupyter Notebooks

Instructions for getting to notebook:

1. This is where you will use the DSVM username and password on the slip of paper
2. Go <https://aka.ms/botnotebooks> You may encounter a certificate error in the browser you are in, but rest assured this is expected behavior right now.
 - On chrome: Click “Advanced” and then “proceed to....”
 - On IE: “Continue to this website”...
3. Use the credentials to log in to jupyter
4. Navigate to the “cogservices_samples” folder
5. Go to the AnomalyDetection.ipynb
 - **Save your work as you go** and it will persist and, note, you do not have to download anything until the VM is pulled down on Sept. 5 at EOD PST)
 - Submit bugs, errors, any problems to the email michhar@microsoft.com (Micheleen Harris)

Also, you can obtain this notebook on the bot-education github repository here:

<https://github.com/michhar/bot-education/tree/master/AzureMachineLearning/Samples>

Cognitive Services Language Understanding Intelligent Service (LUIS)

LUIS Concepts

Intent – aim or goal

Entities – a type or “notion” of person, place or thing

Utterances – the phrase we might use that is added training data

Utterances

If you have unlabeled utterances that your application should handle, they will be available when you edit the application under the "Search" and "Suggest" tabs.

Can link intents to actions and specify requirements for the action

Export LUIS app

Can download your work into a JSON file. This lets you share you application with other developers, or check your LUIS application into your version control.

Pre-Built:

LUIS also provides access to pre-built LUIS applications that use many of the same models found in Microsoft Cortana.

example:

intent – find news on topic and possibly share with another person

entities - We'd like to be able to say what kind of news we are interested in, and also, for sharing, to say who we'd like to share a story with. In order to capture the notion of a news topic, and a recipient for sharing, let's create two entity types: "Topic" and "Recipient".

Pre-built LUIS Cortana model

Just save id and key to use in app

×

Selected Cortana Models Prebuilt App

Help

The pre-built personal assistant V2 can interpret commands like "set an alarm for 6 pm" or "call mom". For a full list of functionality, see the [documentation page](#) for this pre-built application.

Query

URL

<https://api.projectoxford.ai/luis/v1/application?id=c413b2ef-382c-45bd-8ff0-7f6d60e2a821&subscription-key=a80761418f24f32ac9dfda2bba2958&q=find%20a%20good%20mexican%20restaurant>

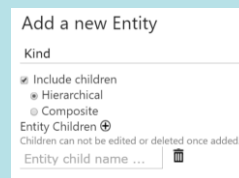
Advanced

▼

LUIS Concepts - Entities

Hierarchical entities – entity that inherits from other entities (start date and end date can inherit from the general date entity)

Composite entities - an entity that is form of a set of existing entities (e.g. a ticket entity as a composite of count and class)




Add a new Entity

Kind

- ☒ Include children
 - ☐ Hierarchical
 - ☐ Composite

Entity Children +

Children can not be edited or deleted once added.

Entity child name ... 

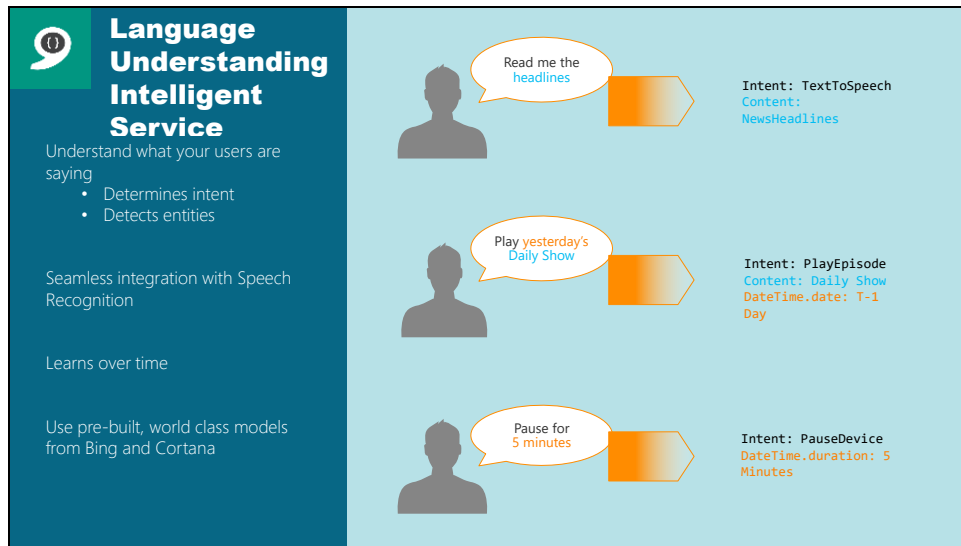
hierarchical entities - The generic entity acts as the parent and the children are the specific types, or sub-groups, under the parent, yet both share the same characteristics. e.g. Date (Start Date and End Date)

composite - In a flight booking app, a user may say “Book 2 adult tickets to Paris next Monday”. In this example, we may create a composite entity called “Tickets”, including the component entities “number” and “category” to capture the number and category of tickets to be booked. If not already existing, you need to define both entities before defining the composite.

LUIS - Training

When you "train" a model, LUIS generalizes from the examples you have labeled, and develops code to recognize the relevant intents and entities in the future

Internally, LUIS uses logistic regression classifiers to determine intents, and conditional random fields (CRFs) to determine the entities. The training process results in optimized classifiers and CRFs, referred to as models, that LUIS can use in the future.



Demo of creating LUIS model in Web UI

<https://www.luis.ai/> - create an app here (a model that's used for intents in natural language later on)

<https://www.luis.ai/Help#CreatingApplication> - for a complete guide on creating a LUIS model and app for consumption

Note that sentences longer than 500 characters will result in an error message.

The sentences that LUIS receives are automatically logged for future use.

LUIS returns a JSON object, with fields and scores for each of the entities and models you have created

- all on the help like above

Other Demos of Cognitive Services APIs

**Led by Anusua Trivedi
On Jupyter Notebooks**

See jupyter notebooks on repository: <https://github.com/michhar/bot-education/tree/master/AzureMachineLearning/Samples>
and <https://github.com/michhar/bot-education/tree/master/CognitiveServices/Samples>

A slide titled "Developer Resources" with a light blue background on the left and a dark blue background on the right. The title is in large, bold, dark blue font. The right side contains three sections: "Documentation" with a link to the Microsoft Cognitive Services documentation, "Client SDKs" with a link to the SDK samples, and "Example Code" with a link to a sample code repository and a note about API references.

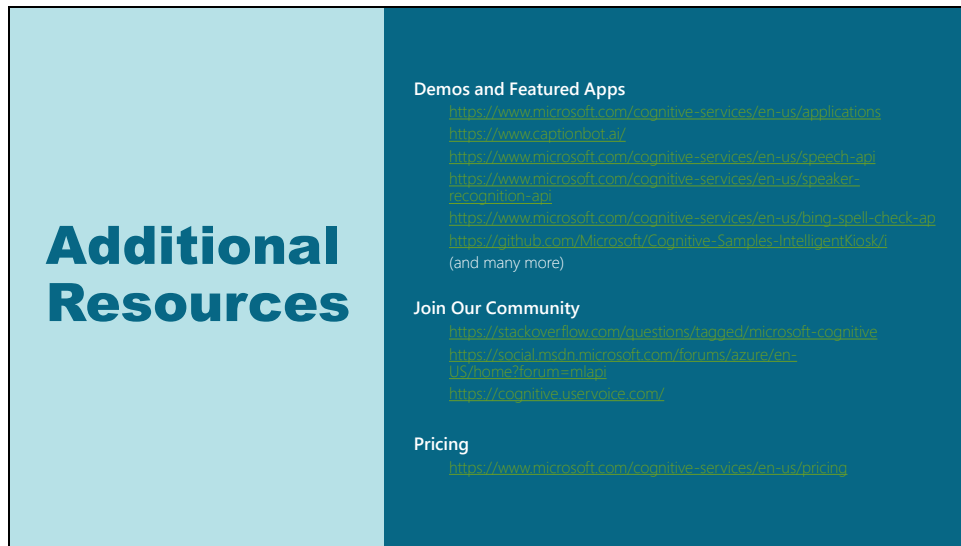
Developer Resources

Documentation
<https://www.microsoft.com/cognitive-services/en-us/documentation>

Client SDKs
<https://www.microsoft.com/cognitive-services/en-us/sdk-sample>

Example Code
<https://text-analytics-demo.azurewebsites.net/Home/SampleCode>
(On most API References as well)

Access to strong documentation, sample code and community resources is critical for developers to be able to understand and become users of Cognitive Services. Customize these links based on your own resources or use the ones listed here.



Additional Resources

Demos and Featured Apps

- <https://www.microsoft.com/cognitive-services/en-us/applications>
- <https://www.captionbot.ai/>
- <https://www.microsoft.com/cognitive-services/en-us/speech-api>
- <https://www.microsoft.com/cognitive-services/en-us/speaker-recognition-api>
- <https://www.microsoft.com/cognitive-services/en-us/bing-spell-check-api>
- <https://github.com/Microsoft/Cognitive-Samples-IntelligentKiosk/>

(and many more)

Join Our Community

- <https://stackoverflow.com/questions/tagged/microsoft-cognitive>
- <https://social.msdn.microsoft.com/forums/azure/en-US/home?forum=mlapi>
- <https://cognitive.uservoice.com/>

Pricing

- <https://www.microsoft.com/cognitive-services/en-us/pricing>

Access to strong documentation, sample code and community resources is critical for developers to be able to understand and become users of Cognitive Services. Customize these links based on your own resources or use the ones listed here.

