Course Guide

Artificial Intelligence Analyst 2019

Course code SAAI ERC 1.0

IBM Training

January 2019 edition

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Contents

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

Cloudant®	dashDB®	Db2®
IBM Cloud™	IBM Watson®	Redbooks®
SPSS®	Watson™	

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

RStudio® is a registered trademark of RStudio, Inc.

Social® is a trademark or registered trademark of TWC Product and Technology, LLC, an IBM Company.

Other product and service names might be trademarks of IBM or other companies.

Course description

Artificial Intelligence Analyst 2019

Duration: 5 days

Purpose

The Artificial Intelligence Analyst career path prepares students to apply AI concepts to build real-life solutions. This career path introduces students to basic concepts of AI, machine learning algorithms, natural language processing, chatbots, and computer vision. Students apply the concepts they learn to practical examples by using IBM Watson services and tools on IBM Cloud.

Audience

Undergraduate senior students from IT related academic programs, for example, computer science, software engineering, information systems and others.

Prerequisites

Before attending Module III AI Analyst classroom, students must meet the following prerequisites:

- Successful completion of Module I *AI Overview* (self-study).
- Successful completion of Module II *Prerequisites* (self-study).
- Successful completion of Exercise 0, *Setting up your hands-on environment*.

Objectives

After completing this course, you should be able to:

- Explain what AI is.
- Describe the field of AI and its subfields: Machine learning, natural language processing (NLP), and computer vision.
- List applications of AI in the industry and government.
- Describe machine learning.
- Describe different type of machine learning algorithms.
- Apply machine learning algorithms to specific problems.
- Explain deep learning.
- Explain convolutional neural networks and neural networks.
- Describe examples of unsupervised and supervised learning.
- Describe IBM Watson.
- Explain how Watson technology is applied to solve real world problems.
- Explain the capabilities of each Watson service.
- Describe Watson Studio, its components, and key applications.
- Describe the CRISP-DM process model and explain where machine learning fits in the CRISP-DM process.
- Create machine learning models for different machine learning algorithms by using Watson Studio.
- Explain domain adaptation.
- Describe the purpose of training the various Watson services.
- Describe IBM Watson Knowledge Studio capabilities and use.
- Explain what NLP is.
- List tools and services for NLP.
- Identify NLP use cases.
- Explain main NLP concepts.
- Explain how to evaluate the quality of an NLP algorithm.
- Identify the Watson services based on NLP technology.
- Use IBM Watson Discovery to build a cognitive query application.
- Describe chatbot applications and chatbots design guidelines.
- Explain core concepts and artifacts needed to build a chatbot application.
- Build chatbot applications with Watson Assistant and Node-RED.
- Explain what computer vision is.
- Identify computer vision use cases.
- Explain how computer vision analyzes and processes images and describe commonly used computer vision techniques.
- Use the Watson Visual Recognition service to classify an image, detect faces, and recognize text in an image.
- Create custom models with Watson Visual Recognition.
- Train the Watson Visual Recognition service with Watson Studio.
- Integrate multiple Watson services to build a comprehensive intelligent solution.

Agenda

Note

The following unit and exercise durations are estimates, and might not reflect every class experience.

The exercise durations do not include the optional parts or sections.

Students in this course use an IBM Cloud Lite account to perform the exercises. This account will never expire, therefore students can continue working on the optional exercises after the class.

Day 1

(00:30) Welcome
(01:15) Unit 1 - Introduction to machine learning
(01:00) Exercise 1 - Applying machine learning algorithms
(01:00) Lunch break
(01:00) Unit 2 - Introduction to IBM Watson
(01:15) Exercise 2 - Exploring Watson services

Day 2

(00:30) Unit 3 - Introduction to IBM Watson Studio
(01:30) Exercise 3 - Getting started with Watson Studio
(00:30) Unit 4 - Introduction to IBM Watson Machine Learning
(01:00) Lunch break
(02:00) Exercise 4 - Getting started with Watson Machine Learning
(01:00) Exercise 5 - Exploring Deep Learning and Neural Network Modeler with Watson Studio

Day 3

(00:30) Unit 5 - Introduction to natural language processing (NLP)
(00:30) Unit 6 - NLP concepts and components
(00:30) Unit 7 - NLP evaluation metrics
(00:30) Unit 8 - NLP and IBM Watson
(01:00) Lunch break
(01:30) Exercise 6 - Ingest, Convert, Enrich and Query with Watson Discovery Service
(00:45) Unit 9 - Introduction to IBM Watson Knowledge Studio
(01:30) Exercise 7 - Creating a machine learning model with Watson Knowledge Studio

Day 4

(00:30) Unit 10 - Introduction to chatbots
(01:00) Unit 11 - Introduction to IBM Watson Assistant
(00:45) Exercise 8 - Getting started with Watson Assistant
(01:30) Exercise 9 - Help Desk chatbot
(01:00) Lunch break
(00:30) Unit 12 - Introduction to computer vision
(00:30) Unit 13 - Computer vision fundamentals
(00:45) Unit 14 - IBM Watson Visual Recognition
(02:00) Exercise 10 - Classifying images with Watson Visual Recognition

Day 5

(00:45) Unit 15 - Designing and building an intelligent solution
(01:30) Exercise 11 - Creating a cognitive banking FAQ chatbot
(01:00) Exercise 12 - Integrating Watson Knowledge Studio with Discovery for the procurement domain (optional)
(01:00) Lunch break
(01:30) Practice test

Unit 1. Introduction to machine learning

Estimated time

01:15

Overview

This unit recaps the main topics in Module I, AI overview and provides a deeper view into complex subjects, such as:

- Machine learning
- Machine learning algorithms
- Neural networks
- Deep learning

IBM Training

Unit objectives

- Explain what is machine learning.
- Describe what is meant by statistical model and algorithm.
- Describe data and data types.
- Describe machine learning types and approaches (Supervised, Unsupervised and Reinforcement).
- List different machine learning algorithms.
- Explain what neural networks and deep learning are, and why they are important in today's AI field.
- Describe machine learning components.
- List the steps in the process to build machine learning applications.
- Explain what domain adaptation is and its applications.

Figure 1-1. Unit objectives

1.1. What is machine learning?

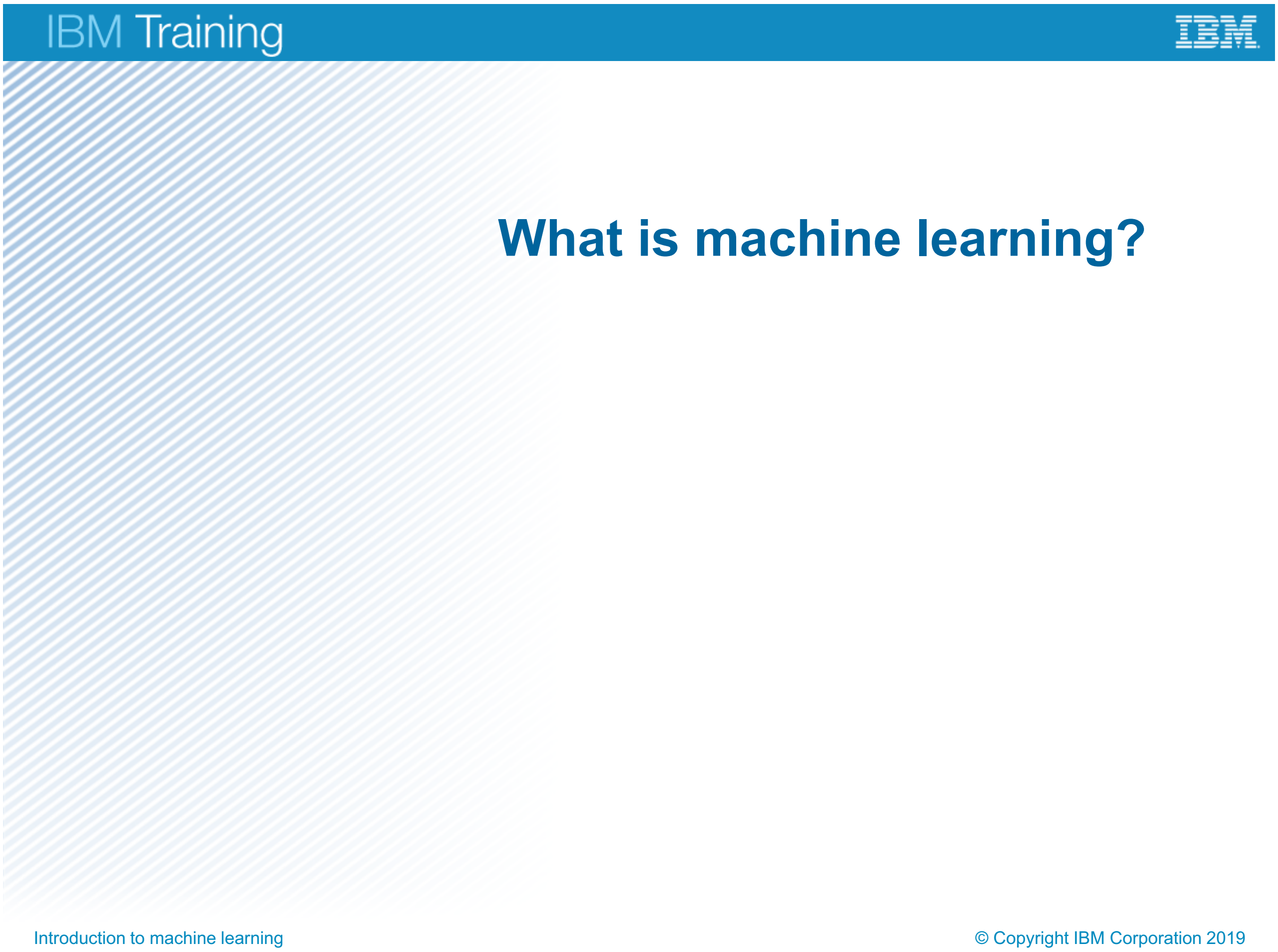


Figure 1-2. What is machine learning?

IBM Training

Topics

- What is machine learning?
- Machine learning algorithms
- What are neural networks?
- What is deep learning?
- How to evaluate a machine learning model?

Introduction to machine learning

Figure 1-3. Topics

IBM Training

Machine learning

- In 1959, the term “machine learning” was first introduced by Arthur Samuel. He defined it as the “*field of study that gives computers the ability to learn without being explicitly programmed*”.
- The learning process improves the machine ***model*** over time by using training data.
- The evolved model is used to make future predictions.

Introduction to machine learning

Figure 1-4. Machine learning

Arthur Samuel, former IBM engineer and a professor at Stanford, was one of the pioneers in the field of computer gaming and artificial intelligence. He was the first one to introduce the term “machine learning”. Machine learning is a field of artificial intelligence. It uses statistical methods to give computer the ability to "learn" from data, without being explicitly programmed.

If a computer program can improve how it performs certain tasks based on past experiences, then it has learned. This differs from performing the task always the same way because it has been programmed to do so.

The learning process improves the so-called “model” over time by using different data points (training data). The evolved model is used to make future predictions.

References:

https://link.springer.com/chapter/10.1007/978-1-4302-5990-9_1
https://link.springer.com/chapter/10.1007/978-94-009-0279-4_9

IBM Training

What is a statistical model

- A model in a computer is a mathematical function that represents a relationship or mapping between a set of inputs and a set of outputs.

f(x)=x^2

Violent crime incidents per day = Average Temperature x 2

- New data “X” can predict the output “Y”.

Y = b0 x X + b1

Introduction to machine learning

Figure 1-5. What is a statistical model

The representation of a model in the computer is in the form of a mathematical function. It is a relationship or mapping between a set of inputs and a set of outputs. For example, f(x)=x^2.

Assume that a system is fed with data indicating that the rates of violent crime are higher when the weather is warmer and more pleasant, even rising sharply during warmer-than-typical winter days. Then, this model can predict the crime rate for this year compared to last year’s rates based on the weather forecast.

Returning to the mathematical representation of the model that can predict crime rate based on temperature, we might propose the following mathematical model:

Violent crime incidents per day = Average Temperature × 2

This is an oversimplified example to explain that machine learning refers to a set of techniques for estimating functions (for example, predicting crime incidents) that is based on data sets (pairs of the day’s average temperature and the associated number of crime incidents). These models can be used for predictions of future data.

1.2. Machine learning algorithms

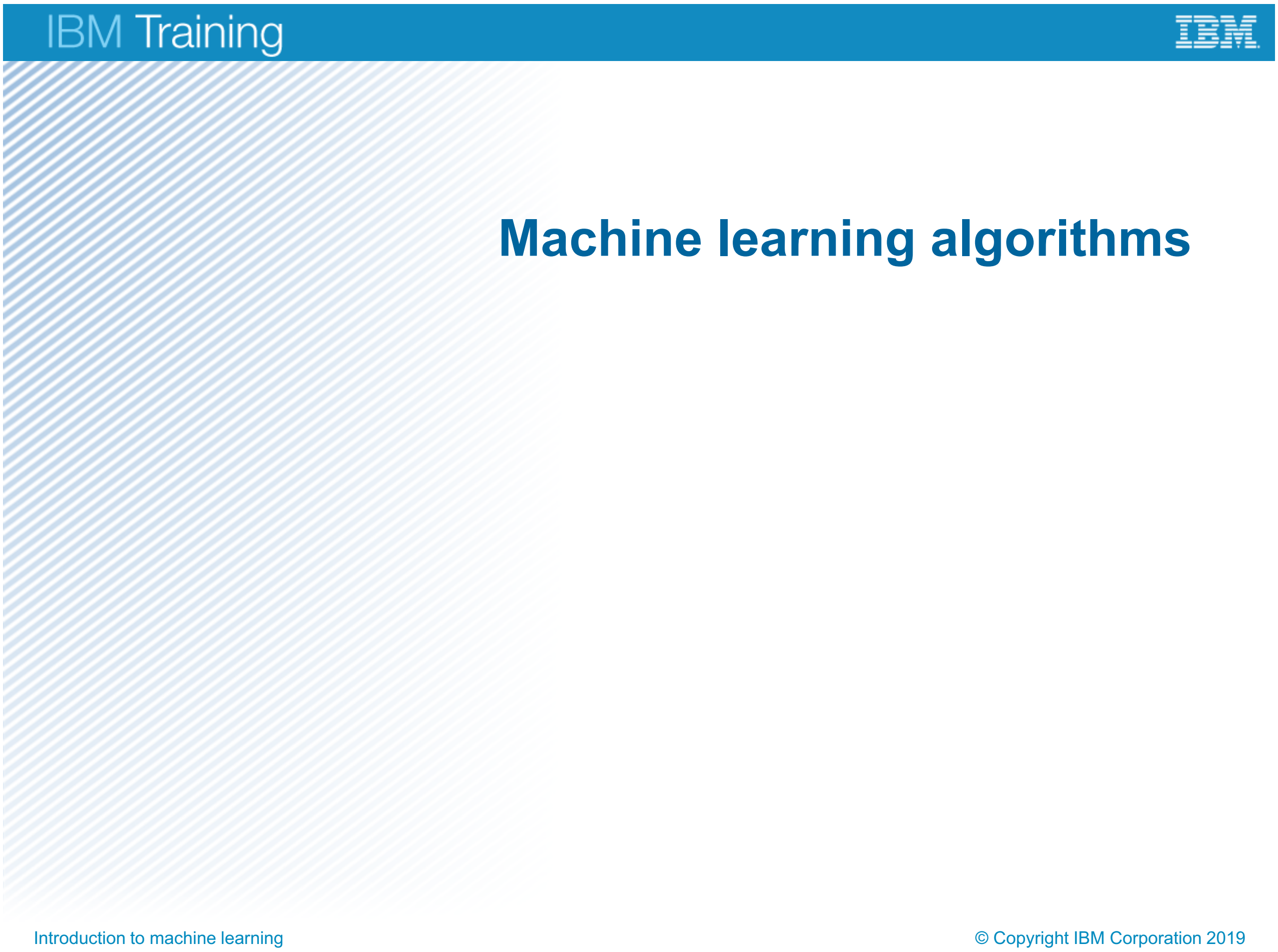


Figure 1-6. Machine learning algorithms

IBM Training

Topics

- What is machine learning?
- Machine learning algorithms
- What are neural networks?
- What is deep learning?
- How to evaluate a machine learning model?

Introduction to machine learning

Figure 1-7. Topics

IBM Training

Machine learning algorithms

- The machine learning algorithm is a technique through which the system extracts useful patterns from historical data. These patterns can be applied to new data.
- The objective is to have the system learn a specific input/output transformation.
- The data quality is critical to the accuracy of the machine learning results.

Introduction to machine learning

Figure 1-8. Machine learning algorithms

To estimate the function that represents the model, an appropriate learning algorithm must be used. In this context, the learning algorithm represents the technique through which the system extracts useful patterns from the input historical data. These patterns can be applied to new data in new situations. The objective is to have the system learn a specific input/output transformation and to make future predictions for a new data point. Finding the appropriate algorithms to solve complex problems in various domains and knowing how and when to apply them is an important skill that machine learning engineers should acquire. Because the machine learning algorithms depend on data, understanding and acquiring data with high quality is crucial for accurate results.

IBM Training

Machine learning approaches

1) Supervised learning: Train by using labeled data, and learn and predict new labels for unseen input data.

- Classification is the task of predicting a discrete class label, such as "black, white, or gray" and "tumor or not tumor".
- Regression is the task of predicting a continuous quantity, such as "weight", "probability", and "cost".

Introduction to machine learning

Figure 1-9. Machine learning approaches

Supervised learning is one of the main categories of machine learning. In supervised machine learning, input data (also known as training examples) comes with a label, and the goal of learning is to predict the label for new, unforeseen examples. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.

In practice, the problems that are solved by using supervised learning are grouped into either regression or classification problems.

Classification is the task of predicting a discrete class label, such as "black, white, or gray" and "tumor or not tumor".

Regression is the task of predicting a continuous quantity, such as "weight", "probability" and "cost".

IBM Training

Machine learning approaches (cont.)

2) **Unsupervised learning:** Detect patterns and relationships between data without using labeled data.

- **Clustering algorithms:** Discover how to split the data set into a number of groups such that the data points in the same groups are more similar to each other compared to data points in other groups.

Introduction to machine learning

Figure 1-10. Machine learning approaches (cont.)

Unsupervised learning is a machine learning type that learns from data that has not been labeled. The goal of unsupervised learning is to detect patterns in the data. One of the most popular types of unsupervised learning is clustering algorithms.

Clustering algorithms are algorithms that discover how to split the data set into a number of groups such that the data points in the same groups are more similar to each other compared to data points in other groups.

IBM Training

Machine learning approaches (cont.)

3) **Semi-supervised learning:**

- A machine learning technique that falls between supervised and unsupervised learning.
- It includes some labeled data with a large amount of unlabeled data.
- Here is an example that uses pseudo-labeling:
  a. Use labeled data to train a model.
  b. Use the model to predict labels for the unlabeled data.
  c. Use the labeled data and the newly generated labeled data to create a new model.

Introduction to machine learning

Figure 1-11. Machine learning approaches (cont.)

Many real practical problems fall into this category of machine learning where you have little labeled data and the rest of the data is unlabeled.

Labeling data is an expensive or time-consuming process. In addition, it mandates having domain experts to label data accurately. Think about labeling skin diseases images that must be labeled by a domain expert. Also, too much labeling data might introduce human biases into the model.

In semi-supervised learning, you try to get the best out of your unlabeled data. There are different techniques to achieve this task. For example, you can use pseudo-labeling, which aims to give approximate labels to unlabeled data. Pseudo-labeling works as follows:

1. Use labeled data to train a model.
2. Use the model to predict labels for the unlabeled data.
3. Use the labeled data and the newly generated labeled data to create a model.

References:

http://deeplearning.net/wp-content/uploads/2013/03/pseudo_label_final.pdf

https://www.analyticsvidhya.com/blog/2017/09/pseudo-labelling-semi-supervised-learning-technique/

IBM Training

Machine learning approaches (cont.)

4) Reinforcement learning

- Reinforcement learning uses trial and error (a rewarding approach).
- The algorithm discovers an association between the goal and the sequence of events that leads to a successful outcome.
- Example reinforcement learning applications:
  - Robotics: A robot that must find its way.
  - Self-driving cars.

Introduction to machine learning

Figure 1-12. Machine learning approaches (cont.)

Reinforcement learning is a goal-oriented learning that is based on interaction with the environment. As the system performs certain actions, it finds out more about the world. Reinforcement learns through trial and error (a rewarding approach).

The algorithm discovers an association between the goal and the sequence of events that leads to a successful outcome.

Example reinforcement learning problems:

- Robotics: A robot that must find its way.
- Self-driving cars.

IBM Training

Machine learning algorithms

Understanding your problem and the different types of ML algorithms helps in selecting the best algorithm.

Here are some machine learning algorithms:

- Naïve Bayes classification (supervised classification – probabilistic)
- Linear regression (supervised regression)
- Logistic regression (supervised classification)
- Support vector machine (SVM) (supervised linear or non-linear classification)
- Decision tree (supervised non-linear classification)
- K-means clustering (unsupervised learning)

Introduction to machine learning

Figure 1-13. Machine learning algorithms

In the following slides, we explore different machine learning algorithms. We describe the most prominent algorithms. Each algorithm belongs to a category of learning. We explore supervised and unsupervised algorithms, regression and classification algorithms, and linear and non-linear classification.

IBM Training

Naïve Bayes classification

- Naïve Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.
  - For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter.
  - Features: Color, roundness, and diameter.
  - Assumption: Each of these features contributes independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

Introduction to machine learning

Figure 1-14. Naïve Bayes classification

Naïve Bayes classifiers is a powerful and simple supervised machine learning algorithm. It assumes that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter.

Features: Color, roundness, and diameter.

A Naïve Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

IBM Training

Naïve Bayes classification (cont.)

Example: Use Naïve Bayes to predict whether the Red, Round shaped, 10 cm diameter label is an apple or not.

Sample No	Color	Shape	Diameter	Is Apple?
1	Red	Round	>= 10 CM	Yes
2	Red	Round	>= 10 CM	No
3	Red	Round	>= 10 CM	Yes
4	Yellow	Round	>= 10 CM	No
5	Yellow	Round	< 10 CM	Yes
6	Yellow	Cylinder	< 10 CM	No
7	Yellow	Cylinder	< 10 CM	Yes
8	Yellow	Cylinder	>= 10 CM	No
9	Red	Cylinder	< 10 CM	No
10	Red	Round	< 10 CM	Yes

Figure 1-15. Naïve Bayes classification (cont.)

Imagine that you have the data set that is shown in the table in this slide. The column with title "Is Apple?" represents the label of the data. Our objective is to make a new prediction for an unknown object. The unknown object has the following features:

- Color: Red
- Shape: Round
- Diameter: 10 cm

Note 1: Sometimes the terminology "parameters" or "variables" is used to describe the "features".

Note 2: "Annotated data" or "labeled data" refer to the same terminology.

IBM Training

Naïve Bayes classification (cont.)

To do a classification, you must perform the following steps:

1. Define two classes (C**Y** and C**N**) that correspond to Apple = Yes and Apple = No.
2. Compute the probability for C**Y** as **x:** *p*(C**Y** | **x**):

   p(Apple = Yes | Colour = Red, Shape = round, Diameter => 10 cm)
3. Compute the probability for C**N** as **x:** *p*(C**N** | **x**):

   p(Apple = No | Colour = Red, Shape = round, Diameter => 10 cm)
4. Discover which conditional probability is larger:

   If *p*(C**Y** |**x**) > *p(*C**N** |**x**), then it is an apple.

Introduction to machine learning

Figure 1-16. Naïve Bayes classification (cont.)

Your algorithm basically depends on calculating two probability values:

- **Class probabilities**: The probabilities of having each class in the training data set.
- **Conditional probabilities**: The probabilities of each input feature giving a specific class value.

The process for solving this problem is as follows:

1. Define two classes C**Y** and C**N** that correspond to Apple = Yes and Apple = No.
2. Compute the probability for C**Y** as **x:** *p*(C**Y** | **x**): *p(Apple = Yes* **|** *Colour = Red, Shape = round, Diameter => 10 cm)*
3. Compute the probability for C**N** as **x:** *p*(C**N** | **x**): *p(Apple = No* **|** *Colour = Red, Shape = round, Diameter => 10 cm)*
4. Discover which conditional probability is larger: If *p*(C**Y** |**x**) > *p(*C**N** |**x**), then it is an apple.

IBM Training

Naïve Bayes classification (cont.)

Naïve Bayes model: $p(C_{\mathbf{k}}|\mathbf{x}) = \dfrac{p(\mathbf{x}|C_{\mathbf{k}})p(C_{\mathbf{k}})}{p(\mathbf{x})}$

5. Compute *p*(**x**|C**Y**) = *p*(Colour = Red, Shape = round, Diameter =>10 cm | Apple = Yes).

   Naïve Bayes assumes that the features of the input data (the apple parameters) are independent.

$$p(\mathbf{x}|C_{\mathbf{k}}) = \prod_{i=1}^{D} p(x_i|C_{\mathbf{k}})$$

Introduction to machine learning

Figure 1-17. Naïve Bayes classification (cont.)

The Naïve Bayes formula is given by this model. Our target is to compute the formula to reach *p*(C**K** |**x**), where K is any class (C**Y** or C**N**).

5. Compute the conditional probability of having each feature given that the class is C**Y**: *p*(**x**|C**Y**) = *p*(Colour = Red, Shape = round, Diameter =>10 cm | Apple = Yes).

Because Naïve Bayes assumes that the features of the input data (the object features) are independent, to get the *p*(**x**|C**Y**) value, we calculate the conditional probability of each feature at a time with the class CY, and then multiply all the values.

IBM Training

Naïve Bayes classification (cont.)

Thus, we can rewrite *p*(**x**| C**Y**) as:

= *p*(Colour = Red | Apple = Yes) X *p*(Shape = round | Apple = Yes) X

p(Diameter => 10 cm | Apple = Yes)

Same for *p*(**x**| C**N**):

= *p*(Color = Red | Apple = No) X *p*(Shape = round | Apple = No) X

p(Diameter => 10 cm | Apple = No)

Introduction to machine learning

Figure 1-18. Naïve Bayes classification (cont.)

Thus, we can rewrite *p*(**x**| C**Y**) as:

= *p*(Colour = Red | Apple = Yes) X *p*(Shape = round | Apple = Yes) X *p*(Diameter => 10 cm | Apple = Yes)

We apply the same rule for *p*(**x**| C**N**) by multiplying the conditional probabilities of reach input feature given C**N**:

= *p*(Color = Red | Apple = No) X *p*(Shape = round | Apple = No) X *p*(Diameter => 10 cm | Apple = No)

IBM Training

Naïve Bayes classification (cont.)

6. Calculate each conditional probability:

p(Colour = Red | Apple = Yes) = 3/5 (Out of five apples, three of them were red.)
p(Colour = Red | Apple = No) = 2/5
p(Shape = Round | Apple = Yes) = 4/5
*p(*Shape = Round | *Apple = No) = 2/5*
p(Diameter = > 10 cm | Apple = Yes) = 2/5
p(Diameter = > 10 cm | Apple = No) = 3/5

Color	Shape	Diameter	Is Apple?
Red	Round	>= 10 CM	Yes
Red	Round	>= 10 CM	No
Red	Round	>= 10 CM	Yes
Yellow	Round	>= 10 CM	No
Yellow	Round	< 10 CM	Yes
Yellow	Cylinder	< 10 CM	No
Yellow	Cylinder	< 10 CM	Yes
Yellow	Cylinder	>= 10 CM	No
Red	Cylinder	< 10 CM	No
Red	Round	< 10 CM	Yes

Introduction to machine learning

Figure 1-19. Naïve Bayes classification (cont.)

Let us see how to calculate these conditional probabilities. For example, to calculate *p*(Colour = Red | Apple = Yes), you are asking, “What is the probability for having a red color object given that we know that it is an apple”.

You browse the table to see how many “is Apple?” has a “yes” label. You see that the occurrence is five times.

Now, from the table, how many of these five occurrences are when you have a color = red? You find that there are three occurrences for red color. Therefore, *p*(Colour = Red | Apple = Yes) = 3/5.

Repeat these steps for the rest of the features.

IBM Training

Naïve Bayes classification (cont.)

- *p*(Color = Red | Apple = Yes) X *p*(Shape = round | Apple = Yes) X

  p(Diameter = > 10 cm | Apple = Yes)

  = (3/5) x (4/5) x (2/5) = 0.192
- *p*(Color = Red | Apple = No) X *p*(Shape = round | Apple = No) X

  p(Diameter = > 10 cm | Apple = No)

  = (2/5) x (2/5) x (3/5) = 0.096
- *p*(Apple = Yes) = 5/10
- *p*(Apple = No) = 5/10

Introduction to machine learning

Figure 1-20. Naïve Bayes classification (cont.)

Now, we have all the values that we need. As mentioned in step 5, we multiply the conditional probabilities as follows:

p(Color = Red | Apple = Yes) X *p*(Shape = round | Apple = Yes) X *p*(Diameter = > 10 cm | Apple = Yes)

= (3/5) x (4/5) x (2/5) = 0.192

p(Color = Red | Apple = No)*p*(Shape = round | Apple = No)*p*(Diameter = > 10 cm | Apple = No)

= (2/5) x (2/5) x (3/5) = 0.096

p(Apple = Yes) = 5/10

p(Apple = No) = 5/10

IBM Training

Naïve Bayes classification (cont.)

Compare p(C**Y** | **x**) to p(CN | **x**):

$$\text{If } \frac{p(C_Y|\mathbf{x})}{p(C_N|\mathbf{x})} > 1 \therefore \mathbf{x} \in C_Y \text{ , else } \mathbf{x} \in C_N$$

$$\frac{p(C_Y|\mathbf{x})}{p(C_N|\mathbf{x})} = \frac{p(\mathbf{x}|C_Y)p(C_Y)}{p(\mathbf{x}|C_N)p(C_N)} = \frac{0.192 \times 0.5}{0.096 \times 0.5} = 2$$

Therefore, the verdict is that it is an apple.

Introduction to machine learning

Figure 1-21. Naïve Bayes classification (cont.)

Finally, we compare the values of *p*(C**Y** |**x**) versus *p(*C**N** |**x**). By substituting the values that were calculated in the previous steps, we discover that *p*(C**Y** |**x**) > *p(*C**N** |**x**), which means that the object is an apple.

IBM Training

Linear regression

- Linear regression is a linear equation that combines a specific set of input values (X) and an outcome (Y) that is the predicted output for that set of input values. As such, both the input and output values are numeric.
- The target variable is a continuous value.

Examples for applications:

- Analyze the marketing effectiveness, pricing, and promotions on the sales of a product.
- Forecast sales by analyzing the monthly company's sales for the past few years.
- Predict house prices with an increase in the sizes of houses.
- Calculate causal relationships between parameters in biological systems.

Introduction to machine learning

Figure 1-22. Linear regression

Regression algorithms are one of the key algorithms that are used in machine learning. Regression algorithms help analysts to model relationships between input variables X and the output label Y for the training data points. This algorithm targets supervised regression problems, that is, the target variable is a continuous value.

In simple linear regression, we establish a relationship between the target variable and input variables by fitting a line that is known as the regression line.

There are different applications that benefit from linear regression:

- Analyze the marketing effectiveness, pricing, and promotions on the sales of a product.
- Forecast sales by analyzing the monthly company's sales for the past few years.
- Predict house prices with an increase in the sizes of houses.
- Calculate causal relationships between parameters in biological systems.

IBM Training

Linear regression (cont.)

- Example: Assume that we are studying the real state market.
- Objective: Predict the price of a house given its size by using previous data.

Size	Price
30	30,000
70	40,000
90	55,000
110	60,000
130	80,000
150	90,000
180	95,000
190	110,000

Introduction to machine learning

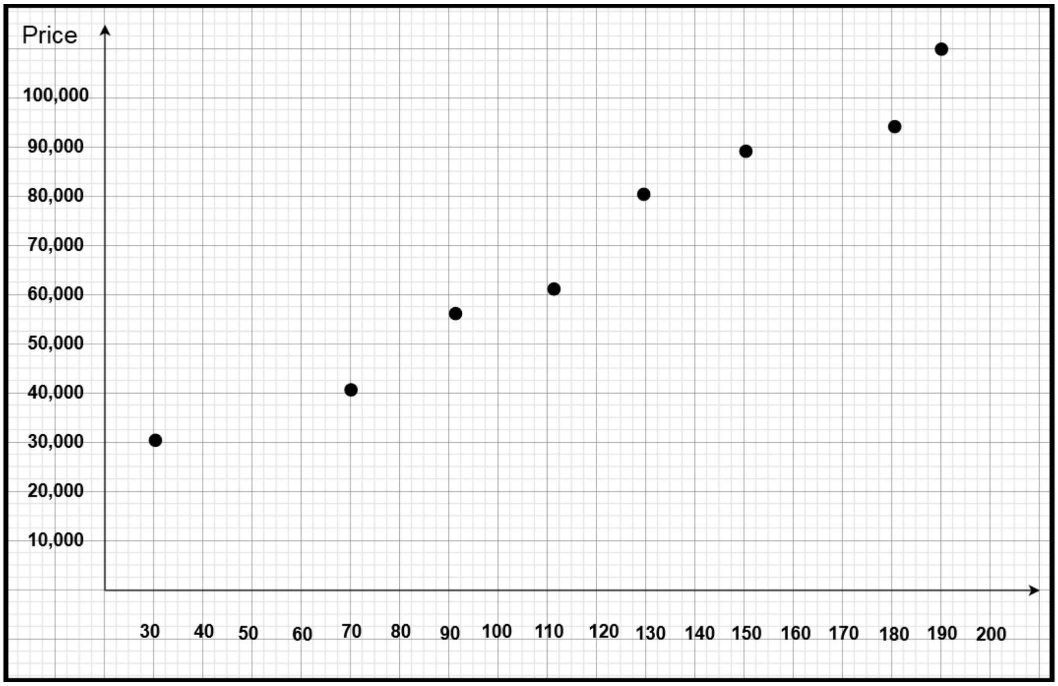
Figure 1-23. Linear regression (cont.)

Assume that we are studying the real state market and our objective is to predict the price of a house given its size by using previous data. The label in this case is the price column.

IBM Training

Linear regression (cont.)

Plot this data as a graph



Introduction to machine learning

Figure 1-24. Linear regression (cont.)

After plotting the points on the graph, they seem to be forming a line.

IBM Training

Linear regression (cont.)

- Can you guess what is the best estimate for a price of a 140-meter square house?
- Which one is correct?

A. $60,000

B. $95,000

C. $85,000

Size	Price
30	30,000
70	40,000
90	55,000
110	60,000
130	80,000
150	90,000
180	95,000
190	110,000

Introduction to machine learning

Figure 1-25. Linear regression (cont.)

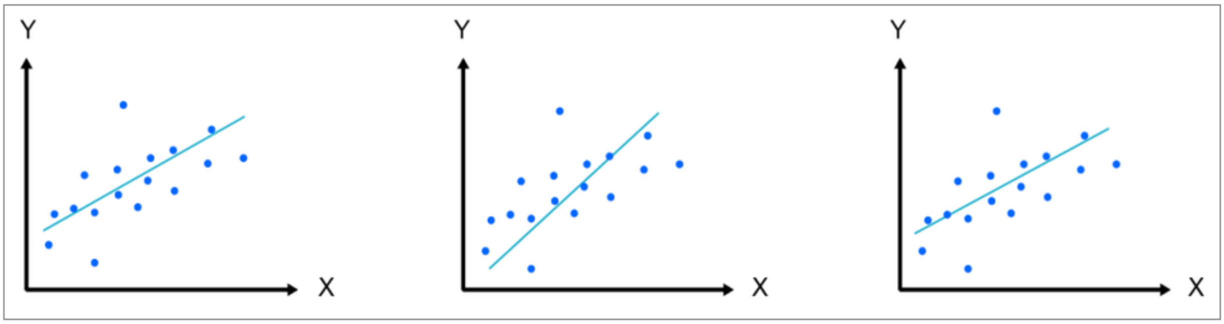
You want to find the price value of a 140-meter square house. Which of the following choices is correct?

1. $60,000
2. $95,000
3. $85,000

IBM Training

Linear regression (cont.)

- **Target**: A line that is within a “proper” distance from all points.
- **Error**: The aggregated distance between data points and the assumed line.
- **Solution**: Calculate the error iteratively until you reach the most accurate line with a minimum error value (that is, the minimum distance between the line and all points).



Introduction to machine learning

Figure 1-26. Linear regression (cont.)

To answer the question “What is the price for a 140-meter square house?”, we need to draw the line that best fits most of the data points.

How we can find the line that best fits all the data points? We can draw many lines, so which one is the best line?

The best line should have the minimal error value. The error refers to the aggregated distance between data points and the assumed line. Calculate the error iteratively until you reach the most accurate line with a minimum error value.

IBM Training

Linear regression (cont.)

- After the learning process, you get the most accurate line, the bias, and the slope to draw your line.
- Here is our linear regression model representation for this problem:

h(p) = p0 + p1 * X1

or

Price = 30,000 + 392* Size

Price = 30,000 + 392* 140

= 85,000

Introduction to machine learning

Figure 1-27. Linear regression (cont.)

After the learning process, you get the most accurate line, the bias, and the slope to draw your line.

p_0 is the bias. It is also called the intercept because it determines where the line intercepts the y axis.

p_1 is the slope because it defines the slope of the line or how x correlates with a y value before adding the bias.

If you have the optimum value of p_0 and p_1, you can draw the line that best represents the data.

IBM Training

Linear regression (cont.)

- Squared error function → $J(\mathsf{P}) = \frac{1}{2m}\sum_{i=1}^{m}(h_{\mathsf{P}}(x^{(i)}) - y^{(i)})^2$
  - m is the number of samples.
  - $h_{\mathsf{P}}(x^{(i)})$ is the predicted value for data point i.
  - $y^{(i)}$ is the actual value for data point i.

Target: Choose P values to minimize errors.

- Stochastic Gradient descent algorithm:

$$\mathbf{P}_j := \mathbf{P}_j - \alpha\left(h_{\mathbf{P}}(x^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

j is the feature number.

α is the learning rate.

Introduction to machine learning

Figure 1-28. Linear regression (cont.)

The squared error function J is represented by the difference between the predicted point and the actual points. It is calculated as follows:

$J(P) = (1/(2*m))\ \Sigma\ (h_p(x^i) - yi)^2$

Where:

- i is the number of a sample or data point within the data set samples.
- $h_p(x^i)$ is the predicted value for data point i.
- y^i is the actual value for data point i.
- m is the count of data set samples or data points.

We can use an optimization technique that is called **stochastic gradient descent**. The algorithm evaluates and updates the weights on every iteration to minimize the model error. The technique works iteratively. In each iteration, the training instance is exposed to the model once. The model makes a prediction and the corresponding error is calculated. The model is updated to reduce the error for the next prediction. The process continues to adjust the model weights to reach the smallest error.

Here we use the gradient descent algorithm to iteratively get the values of p0 and p1 (the intercept and slope of the line are also called weights) by the following algorithm:

$Pj := Pj – \alpha (h_p(x^i) - yi) x_j^i$

Where:

j is the feature number.

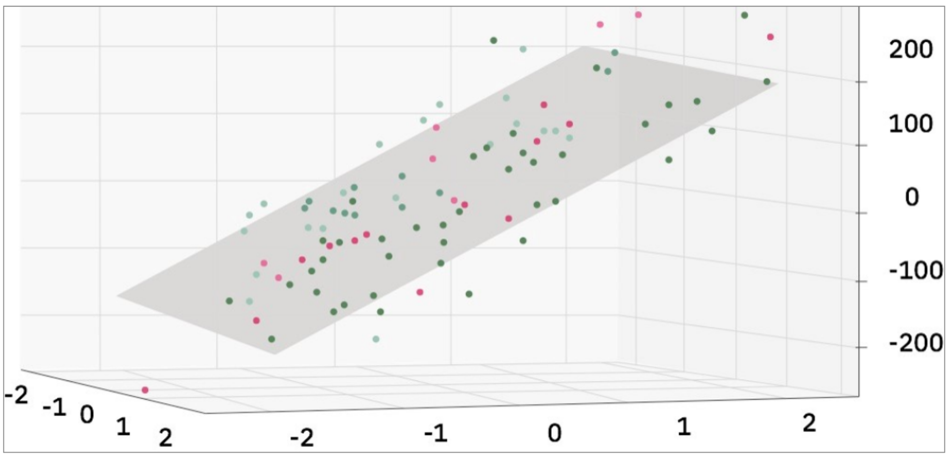
α is the learning rate.

IBM Training

Linear regression (cont.)

- In higher dimensions where we have more than one input (X), the line is called a plane or a hyper-plane.
- The equation can be generalized from simple linear regression to multiple linear regression as follows:

  Y(X)=p0+p1*X1+p2*X2+...+pn*Xn



Introduction to machine learning

Figure 1-29. Linear regression (cont.)

With more features, you do not have a line; instead, you have a plane. In higher dimensions where we have more than one input (X), the line is called a plane or a hyper-plane.

The equation can be generalized from simple linear regression to multiple linear regression as follows:

Y(X)=p0+p1*X1+p2*X2+...+pn*Xn

IBM Training

Logistic regression

- Supervised classification algorithm.
- **Target**: A dependent variable (Y) is a discrete category or a class (not a continuous variable as in linear regression).

  Example: Class1 = Cancer, Class2 = No Cancer

Introduction to machine learning

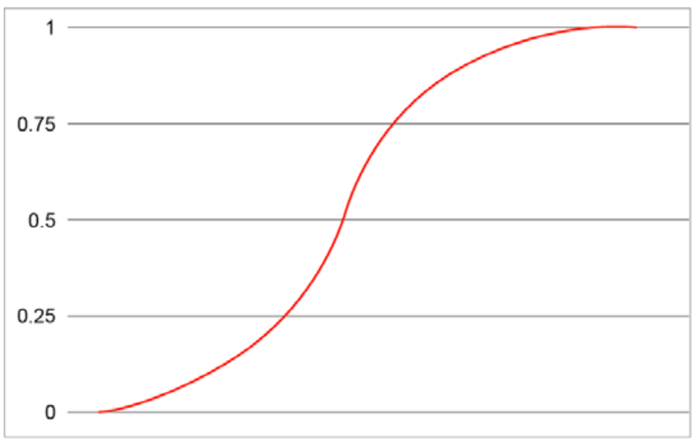
Figure 1-30. Logistic regression

Logistic regression is a supervised classification algorithm. It is different from linear regression where the dependent or output variable is a category or class. The target is a discrete category or a class (not a continuous variable as in linear regression), for example, Class1 = cancer, Class2 = No Cancer.

IBM Training

Logistic regression (cont.)

- Logistic regression is named for the function that is used at the core of the algorithm.
- The logistic function (sigmoid function) is an S-shaped curve for data discrimination across multiple classes. It can take any real value 0 – 1.



Logistic function

Introduction to machine learning

Figure 1-31. Logistic regression (cont.)

Logistic regression is named for the function that is used at the core of the algorithm, which is the logistic function. The logistic function is also known as the sigmoid function. It is an S-shaped curve (as shown in the figure) for data segregation across multiple classes that can take any real value 0 - 1.

IBM Training

Logistic regression (cont.)

- The sigmoid function squeezes the input value between [0,1].
- Logistic regression equation:

Y = exp(p0+p1X)/(1+exp(p0+p1X))

$$\mathbf{h}(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

Introduction to machine learning

Figure 1-32. Logistic regression (cont.)

During the learning process, the system tries to generate a model (estimate a set of parameters p0, p1, …) that can best predict the probability that Y will fall in class A or B given the input X. The sigmoid function squeezes the input value between [0,1], so if the output is 0.77 it is closer to 1, and the predicted class is 1.

IBM Training

Logistic regression (cont.)

- Example: Assume that the estimated values of p's for a certain model that predicts the gender from a person's height are p0= -120 and p1 = 0.5.
- Class 0 represents female and class 1 represents male.
- To compute the prediction, use:
  Y = exp(-120+0.5X)/(1+exp(-120+0.5X))
  Y = 0.00004539

  P(male|height=150) is 0 in this case.

Figure 1-33. Logistic regression (cont.)

Example: Assume that the estimated values of p's for a certain model that predicts the gender from a person's height are p0= -120 and p1=0.5.

Assume that you have two classes where class 0 represents female and class 1 represents male.

Y = exp(-120+0.5X)/(1+exp(-120+0.5X))

Y = 0.00004539

P(male|height=150) is 0 in this case.

IBM Training

Support vector machine

- The goal is to find a separating hyperplane between positive and negative examples of input data.
- SVM is also called a “large Margin Classifier”.
- The SVM algorithm seeks the hyperplane with the largest margin, that is, the largest distance to the nearest sample points.

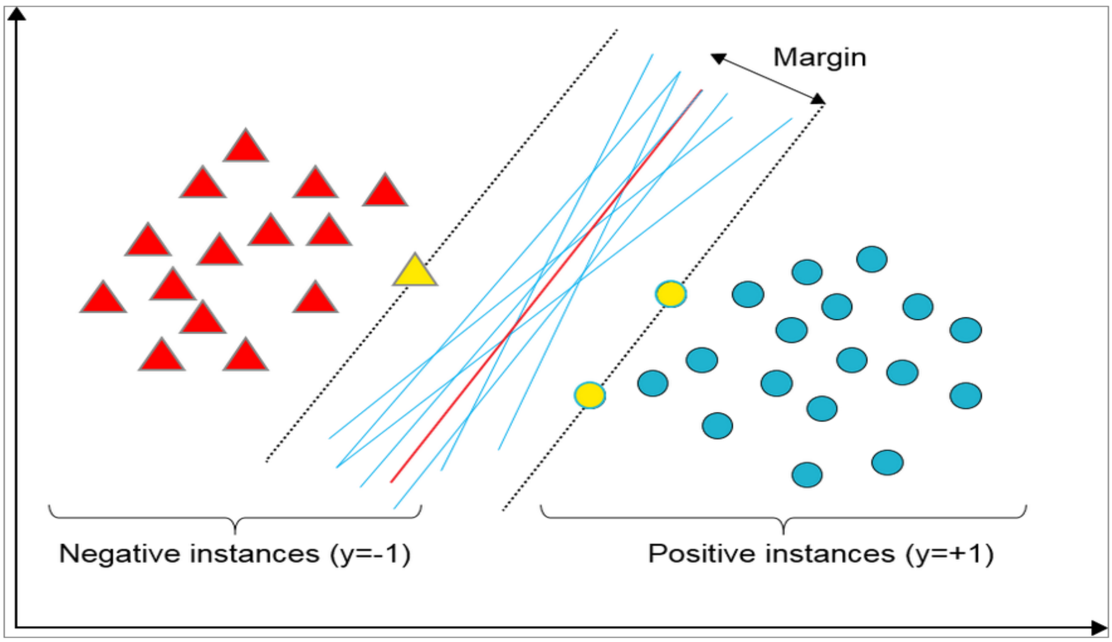
Introduction to machine learning

Figure 1-34. Support vector machine

SVM is a supervised learning model that can be a linear or non-linear classifier. SVM is also called a “large Margin Classifier” because the algorithm seeks the hyperplane with the largest margin, that is, the largest distance to the nearest sample points.

IBM Training

Support vector machine (cont.)



Highlighting the hyperplane with maximum margin with respect to the support vectors

Introduction to machine learning

Figure 1-35. Support vector machine (cont.)

Assume that a data set lies in a two-dimensional space and that the hyperplane will be a one-dimensional line.

Although many lines (in light blue) do separate all instances correctly, there is only one optimal hyperplane (red line) that maximizes the distance to the closest points (in yellow).

IBM Training

Decision tree

- A supervised learning algorithm that uses a tree structure to model decisions.
- It resembles a flow-chart or if-else cases.
- An example for applications is general business decision-making like predicting customers' willingness to purchase a given product in a given setting, for example, online versus a physical store.

Introduction to machine learning

Figure 1-36. Decision tree

A decision tree is a popular supervised learning algorithm that can be used for classification and regression problems. Decision trees are a popular prediction method. Decision trees can explain why a specific prediction was made by traversing the tree.

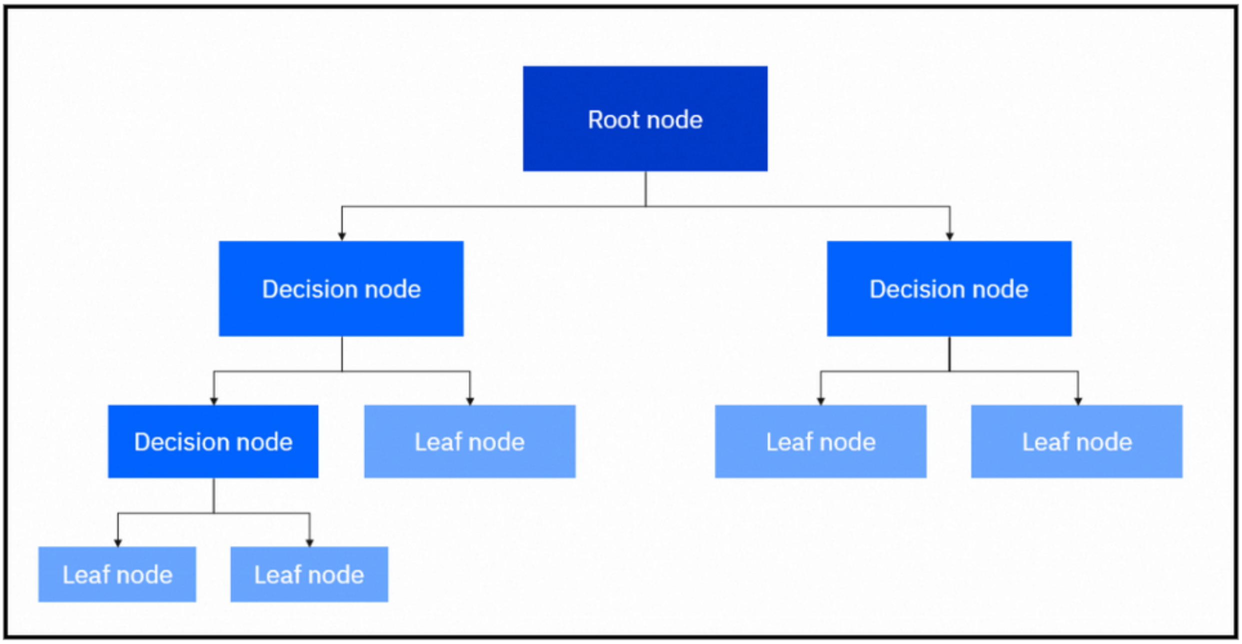
There are different examples for applications that can use decision tree in business. For example, predicting customers' willingness to purchase a given product in a given setting, for example, online versus a physical store.

Note

In our scope, we focus on a classification tree.

IBM Training

Decision tree (cont.)



Graphical representation of decision tree machine learning algorithm

Introduction to machine learning

Figure 1-37. Decision tree (cont.)

A decision tree includes three main entities: root node, decision nodes, and leaves. The figure shows the graphical representation of these entities.

A decision tree builds the classification or regression model in the form of a tree structure. It resembles a flowchart, and is easy to interpret because it breaks down a data set into smaller and smaller subsets while building the associated decision tree.

IBM Training

Decision tree (cont.)

Play Tennis Example: Data

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Introduction to machine learning

Figure 1-38. Decision tree (cont.)

The “Play Tennis” example is one of the most popular examples to explain decision trees.

In the data set, the label is represented by “PlayTennis”. The features are the rest of the columns: “Outlook”, “Temperature”, “Humidity”, and “Wind”. Our goal here is to predict, based on some weather conditions, whether a player can play tennis or not.

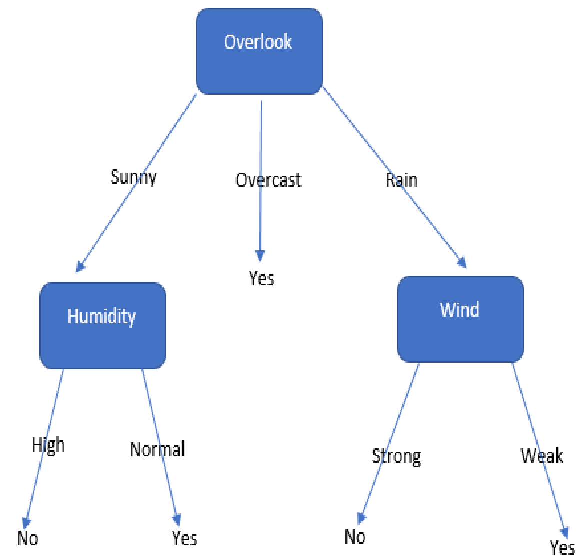
Reference:

http://jmvidal.cse.sc.edu/talks/decisiontrees/choosingbest.html?style=White

IBM Training

Decision tree (cont.)

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No



Introduction to machine learning

Figure 1-39. Decision tree (cont.)

Back to the example, the decision tree representation on the right side of the figure shows the following information:

- Each internal node tests an attribute.
- Each branch corresponds to an attribute value.
- Each leaf node assigns a classification.

Eventually, we want to make a classification of “if Play Tennis = {Yes, No}”.

Reference:

http://jmvidal.cse.sc.edu/talks/decisiontrees/choosingbest.html?style=White

IBM Training

Decision tree (cont.)

A decision tree is built by making decisions regarding the following items:

- Which feature to choose as the root node
- What conditions to use for splitting
- When to stop splitting

Introduction to machine learning

Figure 1-40. Decision tree (cont.)

The algorithm works by recursively splitting the data based on the value of a feature. After each split, the portion of the data becomes more homogeneous.

Now, the algorithm needs to decide:

1. Which feature to choose as the root node.
2. What conditions to use for splitting.
3. When to stop splitting.

IBM Training

Decision tree (cont.)

- Using entropy and information gain to construct a decision tree.
- **Entropy**: It is the measure of the amount of uncertainty and randomness in a set of data for the classification task.
- **Information gain:** It is used for ranking the attributes or features to split at given node in the tree.

  Information gain = (Entropy of distribution before the split)–(entropy of distribution after it)

Introduction to machine learning

Figure 1-41. Decision tree (cont.)

The Iterative Dichotomiser3 (ID3) algorithm works by using entropy and information gain to construct a decision tree. Entropy is the measure of the amount of uncertainty and randomness in a set of data for the classification task. Entropy is maximized when all points have equal probabilities. If entropy is minimal, it means that the attribute or feature appears close to one class and has a good discriminatory power for classification.

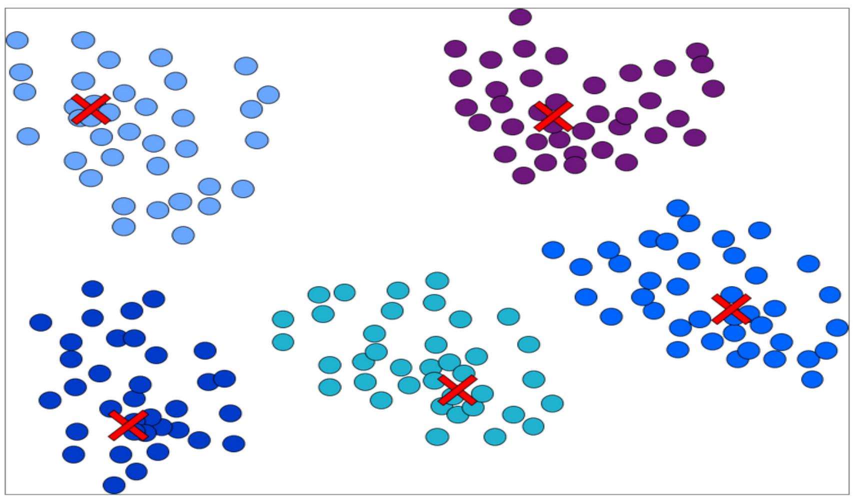
Entropy zero means that there is no randomness for this attribute.

Information gain is a metric that is used for ranking the attributes or features to split at given node in the tree. It defines how much information a feature provides about a class. The feature with the highest information gain is used for the first split.

IBM Training

K-mean clustering

- Unsupervised machine learning algorithm.
- It groups a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than those in other groups (other clusters).



Data partitioned into five clusters - Cluster centroids shown as crosses

Introduction to machine learning

Figure 1-42. K-mean clustering

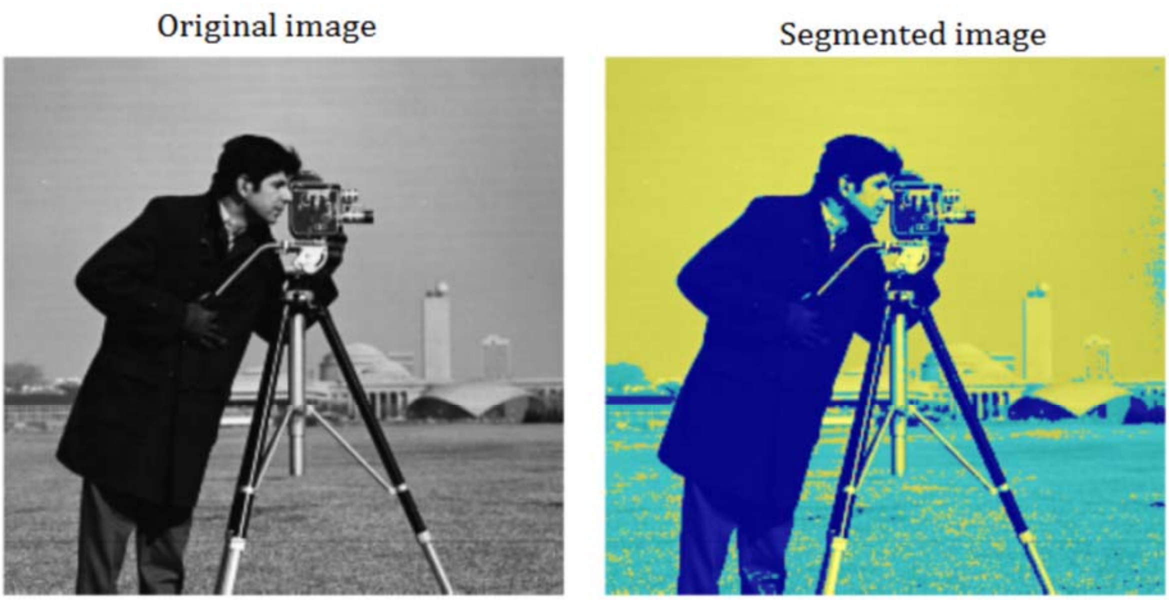
K-means clustering is an unsupervised machine learning technique. The main goal of the algorithm is to group the data observations into k clusters, where each observation belongs to the cluster with the nearest mean.

A cluster’s center is the centroid. The figure shown plots of the partition of a data set into five clusters, with the cluster centroids shown as crosses.

IBM Training

K-means clustering (cont.)

Examples for applications include customer segmentation, image segmentation, and recommendation systems.



Introduction to machine learning

Figure 1-43. K-means clustering (cont.)

Examples of applications include:

- Customer segmentation: Imagine that you are the owner of electronics store. You want to understand preferences of your clients to expand your business. It is not possible to look at each client's purchase details to find a good marketing strategy. But, you can group the details into, for example, five groups based on their purchasing habits. Then, you start building your marketing strategy for each group.
- Image segmentation and compression: The process of partitioning a digital image into multiple segments (sets of pixels) to simplify and change the representation of an image into something that is more meaningful and easier to analyze. To achieve this task, we need a process that assigns a label to every pixel in an image such that pixels with the same label share certain features. The image in this slide is segmented and compressed into three regions by using k-means clustering. With smaller number of clusters, it provides more image compression but at the expense of less image quality.
- Recommendation systems: These systems help you find users with the same preferences to build better recommendation systems.

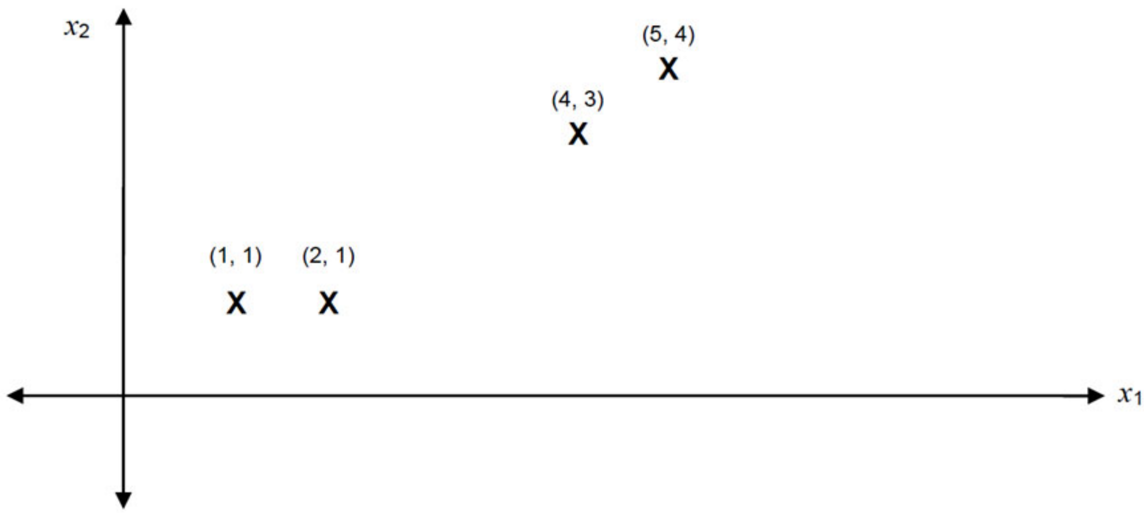
References:

https://www.mathworks.com/help/examples/images/win64/SegmentGrayscaleImageUsingKMeansClusteringExample_02.png
https://www.mathworks.com/help/examples/images/win64/SegmentGrayscaleImageUsingKMeansClusteringExample_01.png

IBM Training

K-means clustering (cont.)

- Example: Given the following data points, use K-means clustering to partition data into two clusters.



Introduction to machine learning

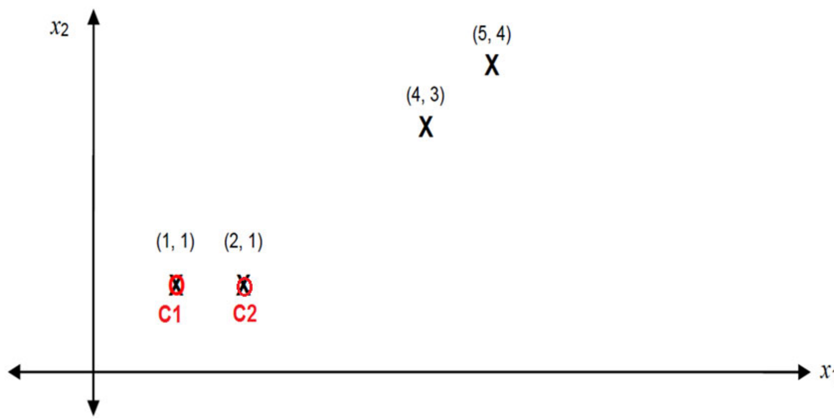
Figure 1-44. K-means clustering (cont.)

Assume that you have the data points that are show in the figure. Your goal is to cluster each data point into one of two groups. Thus, the cluster size is 2. C1 and C2 represent these two clusters.

IBM Training

K-means clustering (cont.)

- Set initial centroids are C1 (1,1) and C2 (2,1)



Introduction to machine learning

Figure 1-45. K-means clustering (cont.)

Assume initial centroids are C1, point (1,1) and C2, point (2,1)

IBM Training

K-means clustering (cont.)

Find a new centroid by using→ $C_{new} = \frac{1}{m} \times \sum_{i=1}^{m} (x^{i})$

Iteration 1:

- Now, we calculate for each point to which center it belongs. The result depends on the distance between the center and the point (by using Euclidian distance):
  Point 1: (1, 1) → d11 = Yes d12 = No

  This means point1(2,2) belongs to C1 and not C2 because it is closer to C1.

  - Point 2: (2, 1) → d21 = No, d22 = Yes
  - Point 3: (4, 3) → d31 = No, d32 = Yes
  - Point 4: (5, 4) → d41 = No, d42 = Yes

- Now, we calculate the new centroid as follows:
  - C1 = (1, 1)
  - C2 = 1/3 ((2, 1) + (4, 3) + (5, 4)) = **(3.67, 2.67)**

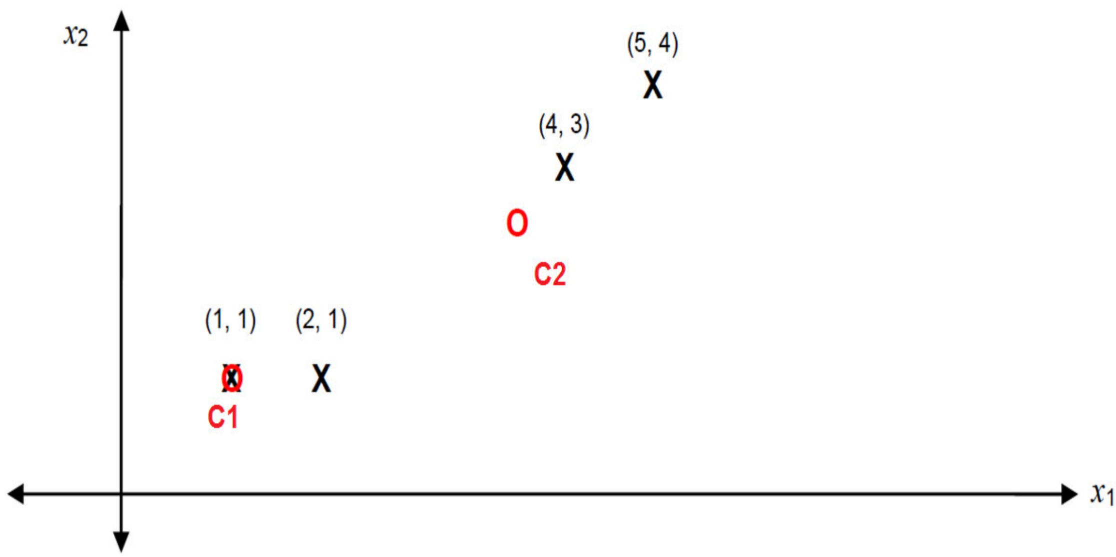
Introduction to machine learning

Figure 1-46. K-means clustering (cont.)

To compute the centroid of a cluster, use an iterative process where each point is examined and you determine whether it belongs to a specific cluster. Then, you compute the new centroid by using the mean of all points.

IBM Training

K-means clustering (cont.)



Introduction to machine learning

Figure 1-47. K-means clustering (cont.)

As you see, the new points in red are the new centroids. We apply another iteration to find a better centroid that represents each cluster.

IBM Training

K-means clustering (cont.)

Iteration 2:

- Point 1: (1, 1) → d11 = Yes, d12 = No
- Point 2: (2, 1) → d21 = Yes, d22 = No
- Point 3: (4, 3) → d31 = No, d32 = Yes
- Point 4: (5, 4) → d41 = No, d42 = Yes

Now, we calculate the new centroid as follows:

- C1 = 1/2 ((1, 1)+(2,1)) = (1.5,1)
- C2 = 1/2 ((4, 3) + (5, 4)) = **(4,5, 3.5)**

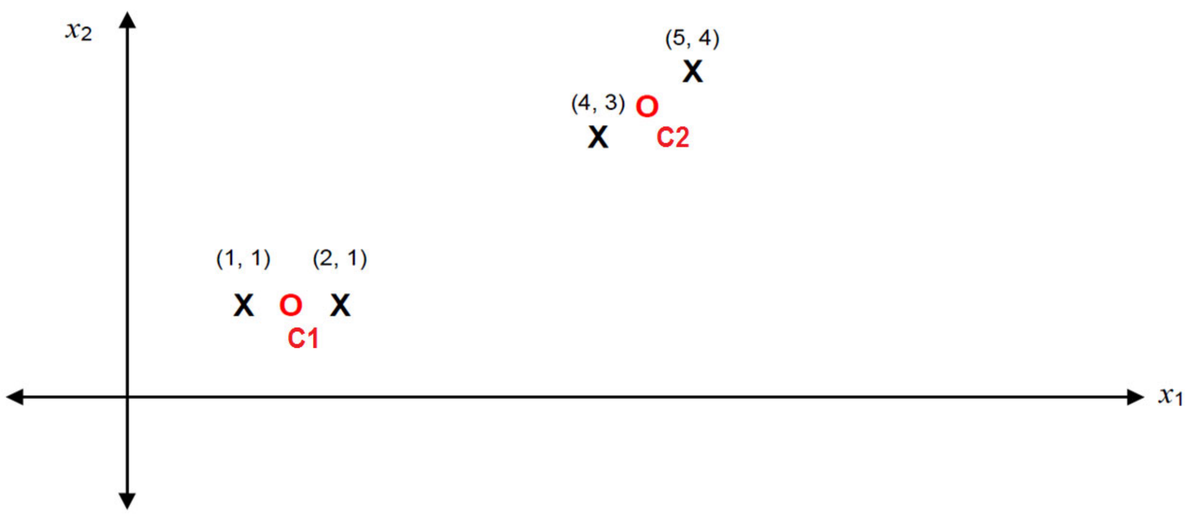
Introduction to machine learning

Figure 1-48. K-means clustering (cont.)

Now, we examine each point again against the centroid by using Euclidian distance and calculate the new centroids (C1 and C2).

IBM Training

K-means clustering (cont.)



Introduction to machine learning

Figure 1-49. K-means clustering (cont.)

As you see, the new red centroids represent the centers of the two clusters. The algorithm stops when the centroids do not change or change slightly, or if a maximum number of iterations are defined.

1.3. What are neural networks?

IBM Training

What are neural networks?

Introduction to machine learning

Figure 1-50. What are neural networks?

IBM Training

Topics

- What is machine learning?
- Machine learning algorithms
- What are neural networks?
- What is deep learning?
- How to evaluate a machine learning model?

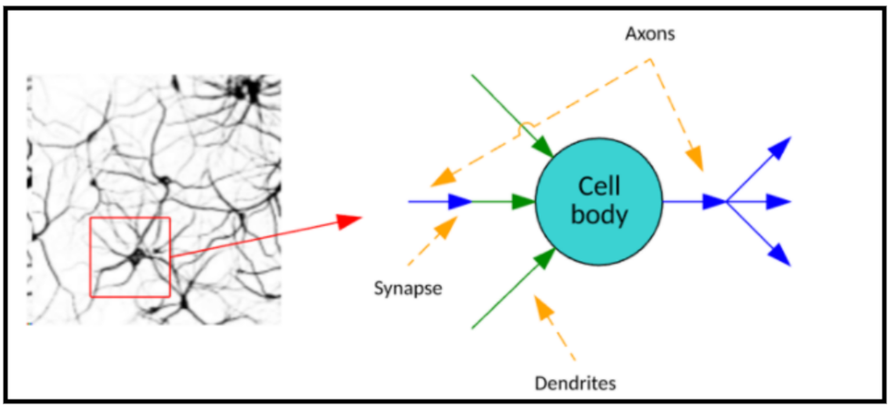
Introduction to machine learning

Figure 1-51. Topics

IBM Training

Neural networks

- Machine learning models that are inspired by the structure of the human brain.
- The human brain is estimated to have 100 billion neurons, and each neuron is connected to up to 10,000 other neurons.



Biological neural network

Introduction to machine learning

Figure 1-52. Neural networks

Neural networks represent an information-processing paradigm that is inspired by the human brain. In the brain, neurons are highly connected and communicate chemical signals through the synapses (a junction between two nerve cells) between the axons and dendrites. The human brain is estimated to have 100 billion neurons, with each neuron connected to up to 10,000 other neurons.

The figure shows a representation of a network of neurons in the brain.

IBM Training

Neural networks (cont.)

- Artificial neural networks are collections of interconnected “neurons” (called nodes) that work together to transform input data to output data.
- Each node applies a mathematical transformation to the data it receives; it then passes its result to the other nodes in its path.
- Examples for applications:
  - Object detection, tracking, and image and video analysis
  - Natural language processing (for example, machine translation)
  - Autonomous cars and robots

Introduction to machine learning

Figure 1-53. Neural networks (cont.)

Artificial neural networks communicate signals (numbers) through weights and activation functions that activate neurons. Using a training algorithm, these networks adjust those weights to solve a problem.

Each node applies a mathematical transformation to the data it receives; it then passes its result to the other nodes in its path. Each connection between nodes represents a different parameter to the model.

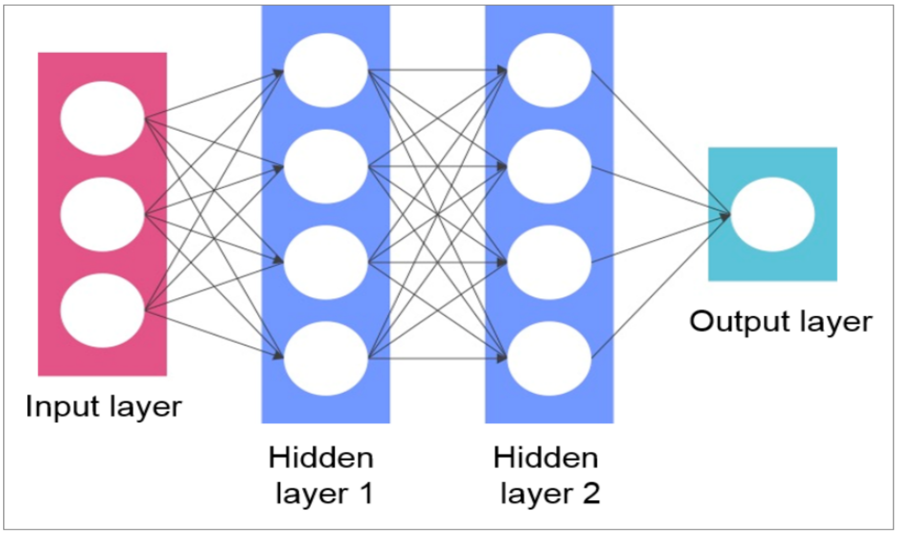
A neural network is useful for machine learning tasks that have too many features (millions). For example:

- Object detection, tracking, and image and video analysis by using a Convolutional Neural Network (CNN)
- Natural language processing tasks like speech recognition and machine translation by using a recurrent neural network (RNN)
- Autonomous cars and robots (more complex neural networks)

IBM Training

Neural networks (cont.)

- Three or more layers (an input layer, one or many hidden layers, and an output layer).
- Neural network models can adjust and learn as data changes.



Multilayer neural network

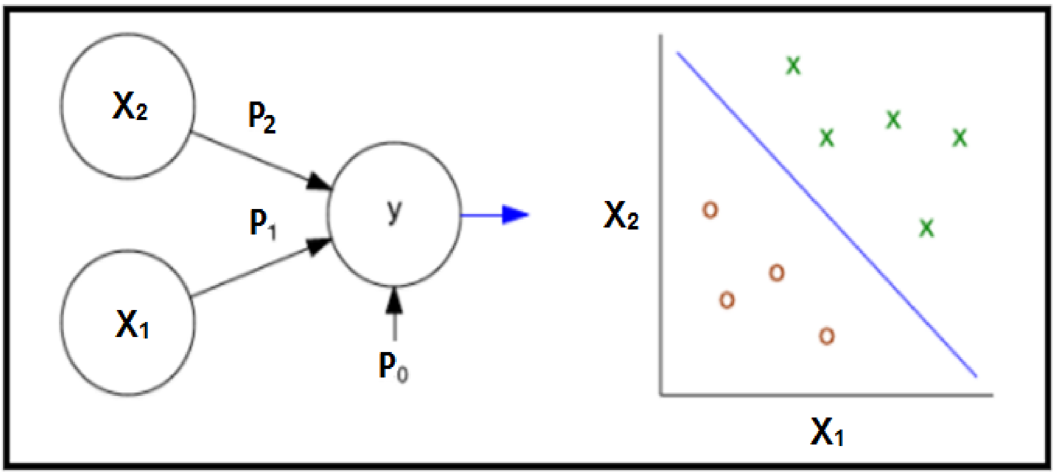
Figure 1-54. Neural networks (cont.)

A neural network is composed of three or more layers: an input layer, one or many hidden layers, and an output layer. Data is imported through the input layer. Then, the data is modified in the hidden and output layers based on the weights that are applied to their nodes. The typical neural network can consist of thousands or even millions of simple processing nodes that are densely interconnected.

IBM Training

Perceptron

- A single neuron model and originator for the neural network.
- Similar to linear classification, where each input has weight.
- One bias.



Perceptron and linear classification

Introduction to machine learning

Figure 1-55. Perceptron

A perceptron is a single neuron model that was an originator for neural networks. It is similar to linear regression. Each neuron has its own bias and slope (weights). For example, assume that a neuron have two inputs (X_1 and X_2), so it requires three weights (P1, P_2 and P0). The figure in this slide shows a weight for each input and one for the bias.

IBM Training

Neural networks: Backpropagation

Backpropagation is an algorithm for training neural networks that has many layers. It works in two phases:

- **First phase:** The propagation of inputs through a neural network to the final layer (called feedforward).
- **Second phase:** The algorithm computes an error. An error value is then calculated by using the wanted output and the actual output for each output neuron in the network. The error value is propagated backward through the weights of the network (adjusting the weights) beginning with the output neurons through the hidden layer and to the input layer (as a function of the contribution of the error).

Introduction to machine learning

Figure 1-56. Neural networks: Backpropagation

Backpropagation is an algorithm for training neural networks that have many layers. It works in two phases:

- Propagation of inputs through a neural network to the final layer (called feedforward).
- The algorithm computes an error. An error value is then calculated by using the wanted output and the actual output for each output neuron in the network. The error value is propagated backward through the weights of the network (adjusting the weights) beginning with the output neurons through the hidden layer and to the input layer (as a function of the contribution of the error).

Backpropagation continues to be an important aspect of neural network learning. With faster and cheaper computing resources, it continues to be applied to larger and denser networks.

1.4. What is deep learning?

IBM Training

What is deep learning?

Introduction to machine learning

Figure 1-57. What is deep learning?

IBM Training

Topics

- What is machine learning?
- Machine learning algorithms
- What are neural networks?
- What is deep learning?
- How to evaluate a machine learning model?

Introduction to machine learning

Figure 1-58. Topics

IBM Training

Deep learning

- Similar to a traditional neural network, but it has many more hidden layers.
- Deep learning has emerged now because of the following reasons:
  - Emergence of big data, which requires data processing scaling.
  - Improvement in processing power and the usage of GPUs to train neural networks.
  - Advancement in algorithms like the rectified linear unit (ReLU).

Introduction to machine learning

Figure 1-59. Deep learning

Deep learning is a machine learning technique that uses neural networks to learn. Although deep learning is similar to a traditional neural network, it has many more hidden layers. The more complex the problem, the more hidden layers there are in the model.

Deep learning has emerged now because of the following reasons:

- The continuous increase in big data requires data processing scaling to analyze and use this data correctly.
- Improvement in processing power and the usage of GPUs to train neural networks.
- Advancement in algorithms like the rectified linear unit (ReLU) instead of the Sigmoid algorithm helps make gradient descent converge faster.

IBM Training

Deep learning (cont.)

Applications:

- Multilayer perceptron (MLP): Classification and regression, for example, a house price prediction.
- Convolutional neural network (CNN): For image processing like facial recognition.
- Recurrent neural network (RNN): For one-dimensional sequence input data. Like audio and languages.
- Hybrid neural network: Covering more complex neural networks, for example, autonomous cars.

Introduction to machine learning

Figure 1-60. Deep learning (cont.)

There are various types of neural networks. Each network is more suitable for a type of machine learning problem. Here is an overview for these networks and their applications:

- Multilayer perceptron (MLP): A class of feed-forward artificial neural networks (ANNs). It is useful in classification problems where inputs are assigned a class. It also works in regression problems for a real-valued quantity like a house price prediction.
- Convolutional neural network (CNN): Takes an input as an image. It is useful for image recognition problems like facial recognition.
- Recurrent neural network (RNN): Has a temporal nature where the input may be a function in time, such as audio files. It is also used for one-dimensional sequence data. It is suitable for inputs like audio and languages. It can be used in applications like speech recognition and machine translation.
- Hybrid neural network: Covers more complex neural networks, for example, autonomous cars that require processing images and work by using radar.

Reference:

https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/

1.5. How to evaluate a machine learning model?

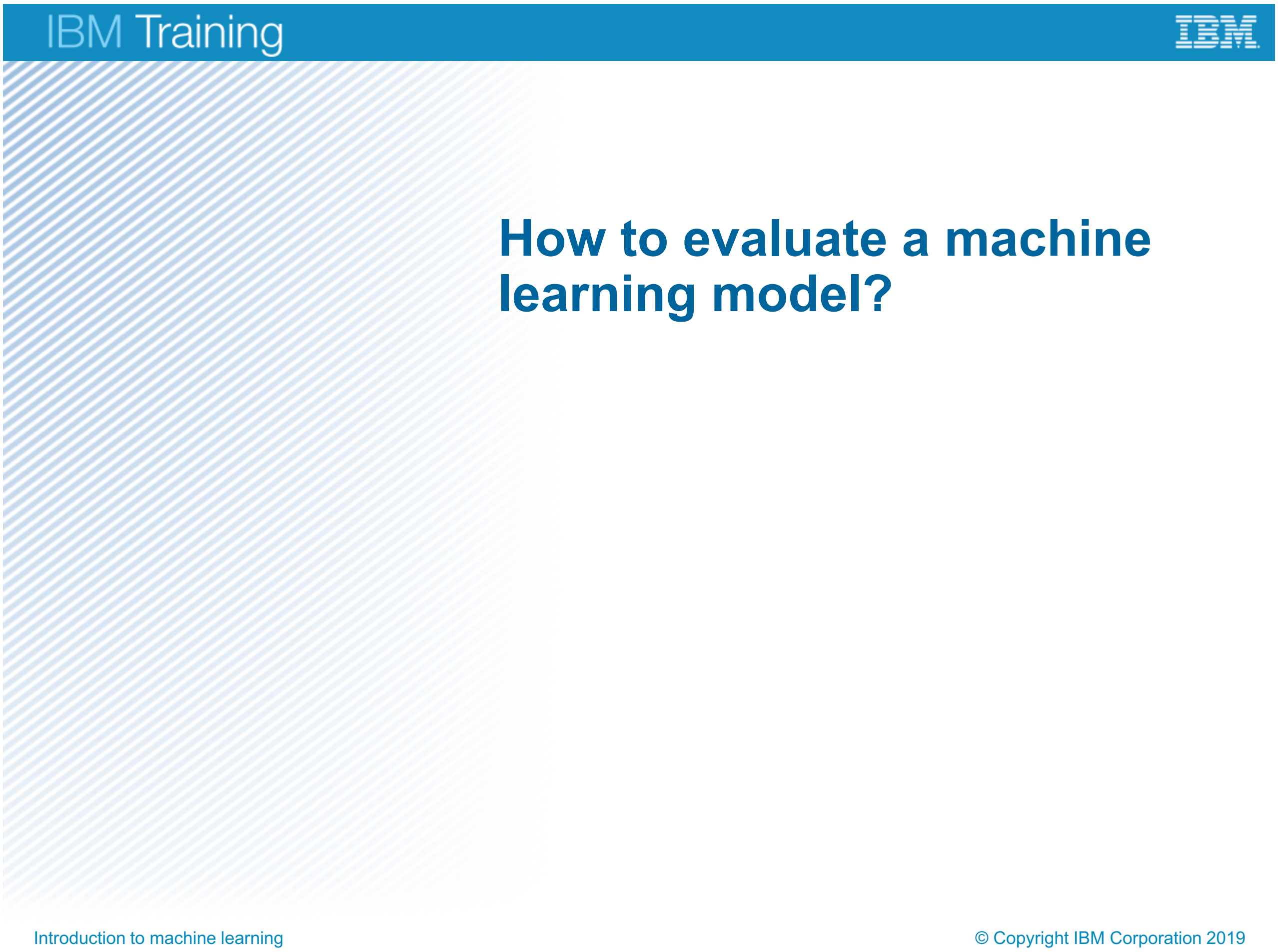


Figure 1-61. How to evaluate a machine learning model?

IBM Training

Topics

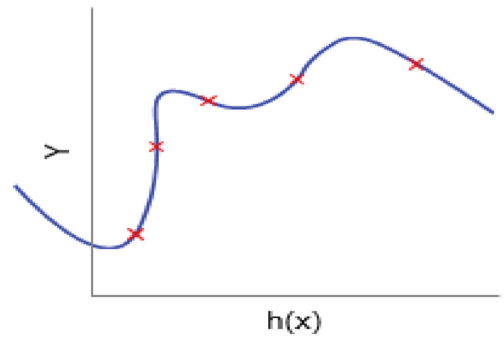
- What is machine learning?
- Machine learning algorithms
- What are neural networks?
- What is deep learning?
- How to evaluate a machine learning model?

Figure 1-62. Topics

IBM Training

Model evaluation

- **Overfitting** occurs when a machine learning model can fit the training set perfectly and fails with unseen future data.
  - **Reason:** Too many features are used or you are reusing training samples in testing.
  - **Solution**:
    - Fewer features
    - More data
    - Cross-validation



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

High variance
(overfit)

Introduction to machine learning

Figure 1-63. Model evaluation

After you have successfully trained your model, you need a methodology to follow to evaluate your machine learning model performance. A classic mistake is to use the same sample data that is used in training to test a model, which produces a false perfect score. This is called “overfitting” (also referred as “high variance”). The problem with overfitting is that your model fails at predicting future unseen data.

Another case that can cause overfitting is where you have unbalanced data. For example, assume that you are working on a data set for churn analysis. The customers who churned are actually 2% of your data set. Using this data set “as is” causes overfitting.

The objective of a good machine learning model is to generalize for any future data points. Overfitting also can occur if you are using too many features. Relatively, if the number of features is the same as or greater than the number of training samples, that can cause overfitting. One of the solutions to overcome overfitting is to increase the number of data set samples that is used for training compared to features. Another solution is to manually decrease the number of features, but that might result in removing useful information. Another solution is to perform model selection by using cross-validation.

References:

https://www.coursera.org/lecture/machine-learning/the-problem-of-overfitting-ACpTQ

https://en.oxforddictionaries.com/definition/overfitting

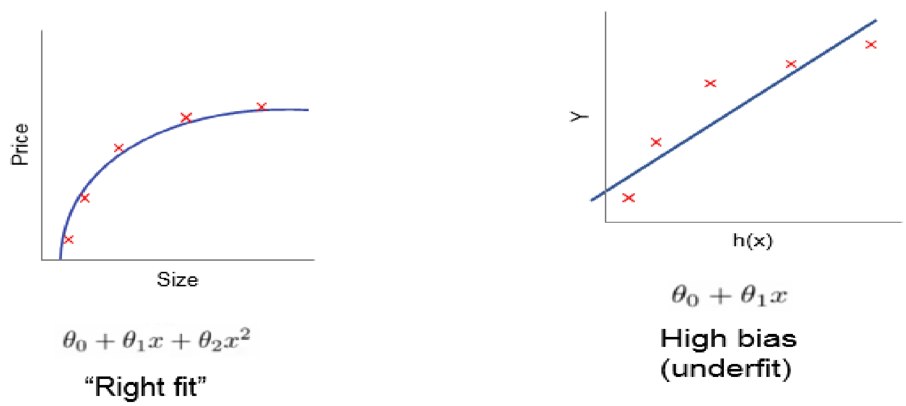
https://ai.stanford.edu/~ang/papers/cv-final.pdf

https://www.youtube.com/watch?v=OSd30QGMl88

IBM Training

Model evaluation (cont.)

- **Underfitting** occurs when a machine learning model cannot fit the training data or generalize to new data.
  - **Reason:** The model is using a simple estimator.
  - **Solution**: Add More features or use different estimator



Introduction to machine learning

Figure 1-64. Model evaluation (cont.)

Underfitting (also referred to as “high bias”) occurs when a machine learning model cannot fit the training data or generalize to new data.

A possible reason might be that the model is using a simple estimator. For example, you might be using a linear estimator, but what you actually need is a quadratic or higher degree polynomial estimator to develop your model like in “Right fit” graph.

Another reason might be that you are not using enough features, so your estimator fails to capture the structure of the data. A possible solution would be to add more features and try a different estimator.

There are other methods that are used to help resolve the overfitting and underfitting of your model such as regularization, but these methods are beyond the scope of this course.

References:

https://en.oxforddictionaries.com/definition/overfitting
https://www.youtube.com/watch?v=OSd30QGMl88

IBM Training

Model evaluation (cont.)

- **Cross-validation (CV)** is a process to evaluate a model by dividing the data set once or several times in training and testing.
- **Hold-out method:** Randomly splits the data set into a training set and test set.
- **K-fold cross validation:** Splits data into K subsamples where each subsample gets a chance to be the validation set, and K-1 is the training set.
- **Leave one out cross validation (LOO-CV):** Similar to K-fold except that one subsample that contains one data point is held out, and the rest of data is used for training.

Introduction to machine learning

Figure 1-65. Model evaluation (cont.)

It is common practice when applying a (supervised) machine learning task is to hold out part of the available data as a test set. There are different methods to achieve that task:

- Cross-validation (CV) is a process to evaluate a machine learning model by splitting a data set once or several times to train and test the model. The data set can be split into a training set to train the model and a validation set to pre-test the model. Select the model that has least error. Finally, there is a test set to evaluate the model. Thus, the data set can be split as 60% - 20% - 20% for training, validation, and testing sets.

  One criticism of this process is that splitting the data set into three parts reduces the number of samples that can be used for training the model.
- The hold-out method partitions the data set into a majority set for training and minority set for testing. The split of the training set to test set is 80% - 20% or 70% - 30%, with no fixed rule.

- K-fold cross validation randomly partitions data into K equal sized subsamples. For each iteration, one subsample is kept as validation set and the rest of the subsamples (K-1) are the training set. The iterations are repeated K times, where each subsample has one chance to be the validation set. The K results can then be averaged to produce a single model. The biggest advantage of K-fold is that all data is changed to be used for both training and validation. There is no strict rule for the number K, but it is commonly K=5 or K=10, which are 5-fold cross-validation or 10-fold cross-validation. For each subsample, you maintain approximately the same percentage of data of each target class as in the complete set, which is known as the Stratified K-fold method.
- Leave one out CV (LOO-CV) is similar to K-fold, but in this case each one sample data point is held out as a validation set, and the rest of data set is the training set. Comparing LOO-CV and K-fold, K-fold is faster and requires less computation, but in terms of accuracy, LOO-CV often has a high variance as an estimator.

References:

https://projecteuclid.org/download/pdfview_1/euclid.ssu/1268143839

http://scikit-learn.org/stable/modules/cross_validation.html

https://www.cs.cmu.edu/~schneide/tut5/node42.html

IBM Training

Unit summary

- Explain what is machine learning.
- Describe what is meant by statistical model and algorithm.
- Describe data and data types.
- Describe machine learning types and approaches (Supervised, Unsupervised and Reinforcement).
- List different machine learning algorithms.
- Explain what neural networks and deep learning are, and why they are important in today's AI field.
- Describe machine learning components.
- List the steps in the process to build machine learning applications.
- Explain what domain adaptation is and its applications.

Introduction to machine learning

Figure 1-66. Unit summary

IBM Training

Review questions

1. True or False: In logistic regression, the target output is a continuous value.
2. True or False: Autonomous cars are one of many applications of deep learning.
3. What type of learning is K-means clustering?
   A. Semi-supervised learning
   B. Supervised learning
   C. Unsupervised learning
   D. None of the above

Introduction to machine learning

Figure 1-67. Review questions

1. False
2. True
3. C. Unsupervised learning

IBM Training

Review questions (cont.)

4. Assume that you want to develop a learning algorithm to address the following problem: You want to examine individual customer accounts and determine whether they have been hacked or compromised. This should be treated as a supervised ___________.
   A. Classification
   B. Regression

Introduction to machine learning

Figure 1-68. Review questions (cont.)

A. Classification

IBM Training

Review answers

1. True or False: In a logistic regression, the target output is a continuous value.
   The answer is False.
2. True or False: Autonomous cars are one of many applications of deep learning.
   The answer is True.
3. What type of learning is K-means clustering?
   A. Semi-supervised learning
   B. Supervised learning
   C. Unsupervised learning
   D. None of the above
   The answer is C.

Introduction to machine learning

Figure 1-69. Review answers

IBM Training

Review answers (cont.)

4. Assume that you want to develop a learning algorithm to address the following problem: You want to examine individual customer accounts and determine whether they have been hacked or compromised. This should be treated as a supervised ___________.
   A. Classification
   B. Regression
   The answer is A.

Introduction to machine learning

Figure 1-70. Review answers (cont.)

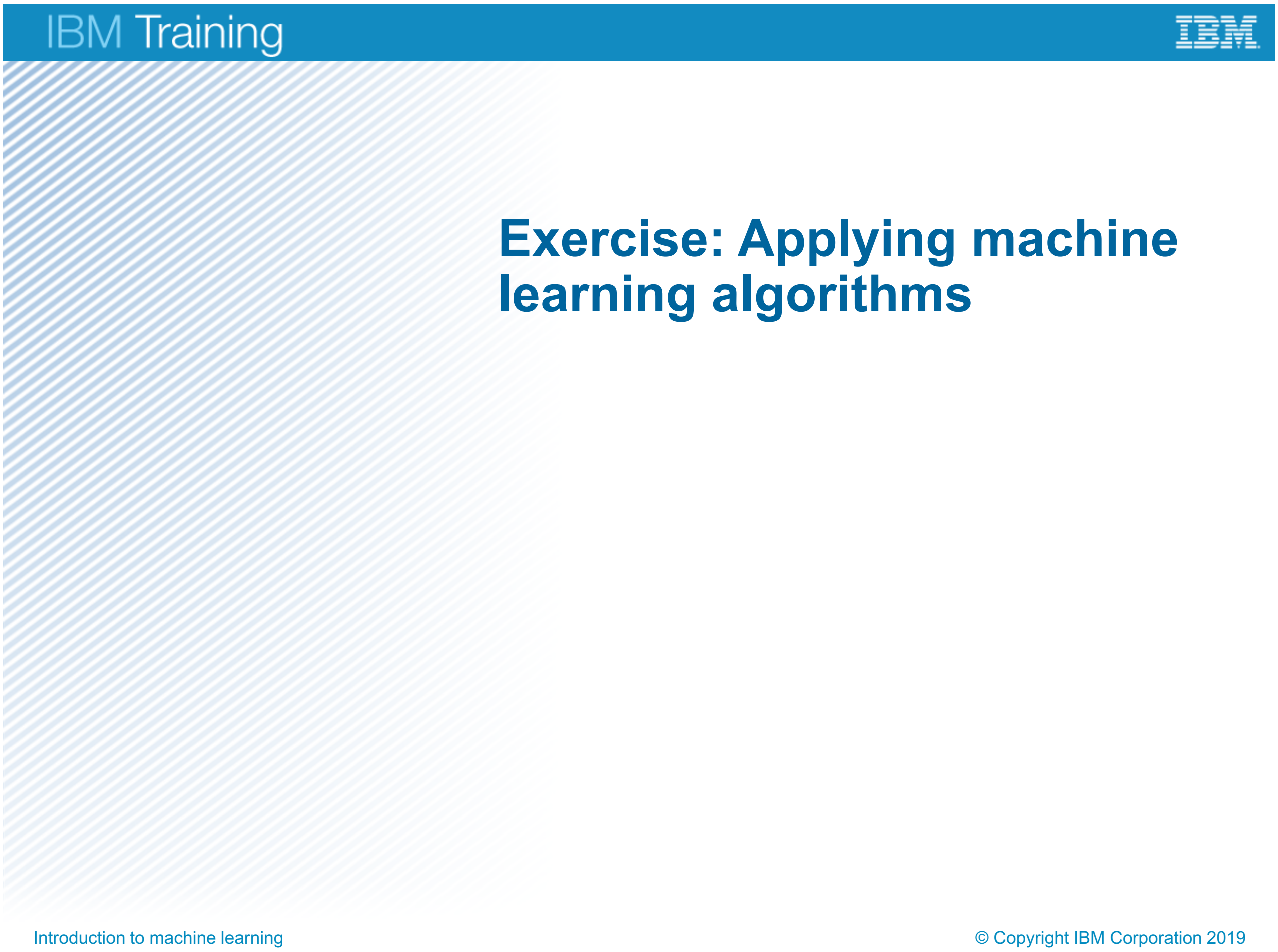


Figure 1-71. Exercise: Applying machine learning algorithms

IBM Training

Exercise objectives

- This exercise provides the foundational concepts that you need to understand to apply four popular machine learning algorithms:
  - K-means clustering
  - Naïve Bayes classifier
  - Linear regression
  - Decision tree
- After completing this exercise, you should be able to:
  - Determine the centroids of a data set with the K-means clustering algorithm.
  - Predict the class of an object with the Naïve Bayes classifier.
  - Apply the linear regression algorithm to solve supervised learning problems.
  - Construct a decision tree to predict outcomes.

Introduction to machine learning

Figure 1-72. Exercise objectives

After completing this exercise, you should be able to:

- Determine the centroids of a data set with the K-means clustering algorithm.
- Predict the class of an object with the Naïve Bayes classifier.
- Apply the linear regression algorithm to solve supervised learning problems.
- Construct a decision tree to predict outcomes.

Unit 2. Introduction to IBM Watson

Estimated time

01:00

Overview

This unit provides an overview of key IBM Watson services, their purpose, how they work, and it helps you get started with Watson services on IBM Cloud.

IBM Training

Unit objectives

- Explain what IBM Watson is.
- List the Watson service.
- Explain the capabilities of each Watson service.
- Describe the purpose of training the various Watson services to adapt them to a closed-domain.
- List the Watson services that can be trained.
- List the Watson services that cannot be trained.
- Describe what Watson Knowledge Studio is.
- List the Watson services that can be trained with Watson Knowledge Studio.
- Create a Watson service instance.

Introduction to IBM Watson

Figure 2-1. Unit objectives

2.1. IBM Watson overview

IBM Training

IBM Watson overview

Introduction to IBM Watson

© Copyright IBM Corporation 2019

Figure 2-2. IBM Watson overview

This section provides a high-level overview of IBM Watson

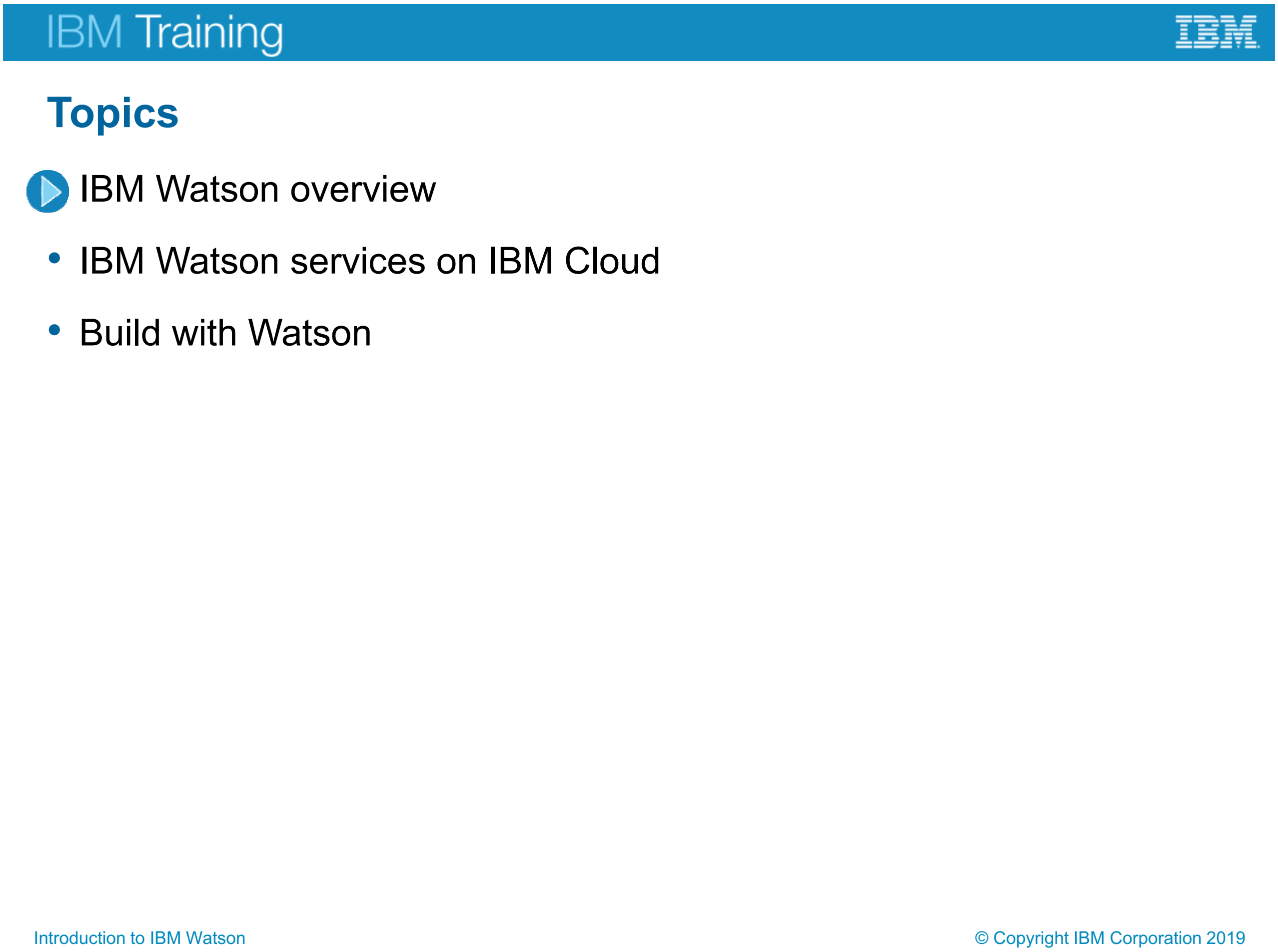


Figure 2-3. Topics

IBM Training

IBM Watson

- IBM Watson is the artificial intelligence (AI) offering from IBM.
- IBM Watson has the following capabilities:
  - Understands all forms of data.
  - Interacts naturally with people.
  - Learns and reasons at scale.
- You do not need to know the details of every associated AI subfield.
- You must have a high-level understanding of each subfield.
- You must know how to apply the correct AI technology to the problem by using AI application programming interfaces (APIs) or a ready-to-use AI framework.

Introduction to IBM Watson

Figure 2-4. IBM Watson

IBM Watson is an AI system that enables a new partnership between people and computers. It is the AI offering from IBM. Watson combines five core capabilities:

- Interacts with people more naturally based on the person's preference.
- Quickly imports key industry materials by partnering with experts to scale and elevate expertise.
- Enables new products and services to sense, reason, and learn about users and the world around them.
- Uses data to improve business processes and forecasting, which increases operational effectiveness.
- Enhances exploration and discovery, which uncovers unique patterns, opportunities, and actionable hypotheses.

IBM Watson includes AI technologies. Those technologies are complex, and to understand them fully requires many years of study. However, to add AI capabilities to your applications, you do not need to know the details that are associated with each AI subfield. You must have a high-level understanding of each subfield, for example, natural language processing (NLP), computer vision, and machine learning. Then, you must know how to apply the correct AI technology to a problem by using AI APIs or a ready-to-use AI framework.

IBM Training

IBM Watson history

- Named after the first CEO of IBM, Thomas J. Watson.
- Originally designed as a QA system architecture named DeepQA.
- In 2011, the Watson computer system competed on Jeopardy! against the two most successful champions: Brad Rutter and Ken Jennings. Watson won the first place prize of $1 million.

Introduction to IBM Watson

Figure 2-5. IBM Watson history

Watson was named after the first CEO of IBM, Mr Thomas J. Watson.

In 2011, the Watson computer system competed on Jeopardy! a US nationally televised quiz show, against former winners Brad Rutter and Ken Jennings. Watson won the first place prize of $1 million. Watson had access to 200 million pages of structured and unstructured content that filled 4 TB of disk storage, including the full text of Wikipedia.

The original Watson system that won Jeopardy! was based on a QuestionAnswering (QA) system architecture, which was developed for Watson to play the game and which is known as the *DeepQA* architecture. This system was specifically developer to answer trivia question.

DeepQA was a software architecture for deep content analysis and evidence-based reasoning. It uses advanced natural language processing (NLP), information retrieval, reasoning, and machine learning. In summary, DeepQA generates and scores many hypotheses by using an extensible collection of natural language processing, machine learning, and reasoning algorithms, which gather and weigh evidence over both unstructured and structured content to determine the answer with the best confidence. The answer with the strongest confidence is Watson’s final answer. DeepQA was trained in the open domain Wikipedia which is very appropriate to answer trivia questions.

The original DeepQA architecture, designed to play the Jeopardy! game, evolved trough several iterations to the Watson services and offerings available today. And it will continue to evolve through the years with the major advancements in AI.

IBM Training

IBM Watson relies on data, information, and expertise

Available as a set of open application programming interfaces (APIs), software as a service (SaaS) tools and industry solutions.

Data, information, and expertise create the foundation.

Examples include:	Newspapers
Analyst reports	Blogs
Tweets	Wiki
Wire tap transcripts	Court rulings
Battlefield docs	International crime database
Emails	Stolen vehicle data
Texts	Missing persons data
Forensic reports	

Introduction to IBM Watson

Figure 2-6. IBM Watson relies on data, information, and expertise

Data, information, and expertise create the foundation for working with Watson. The figure shows examples of data and information that Watson can analyze and learn from, and derive new insights that were never discovered before.

IBM Training

IBM Watson services offerings

- IBM Watson is available as the following offerings:
  - A set of services on IBM Cloud
  - Software as a Service (SaaS)
  - Set of industry solutions
  - Other solutions
- The focus of this unit is the IBM Watson services on IBM Cloud.

Introduction to IBM Watson

Figure 2-7. IBM Watson services offerings

IBM Watson is available as a set of services on IBM Cloud, SaaS cloud offerings, and a set of industry solutions. Besides the core offerings, a large set of resources and documentation are available for application developers.

The focus of this unit (and this course) is on IBM Watson services on IBM Cloud only.

2.2. IBM Watson services on IBM Cloud

IBM Training

IBM Watson services on IBM Cloud

Introduction to IBM Watson

Figure 2-8. IBM Watson services on IBM Cloud

This section provides an overview of the Watson services and tools available on IBM Cloud.

IBM Training

Topics

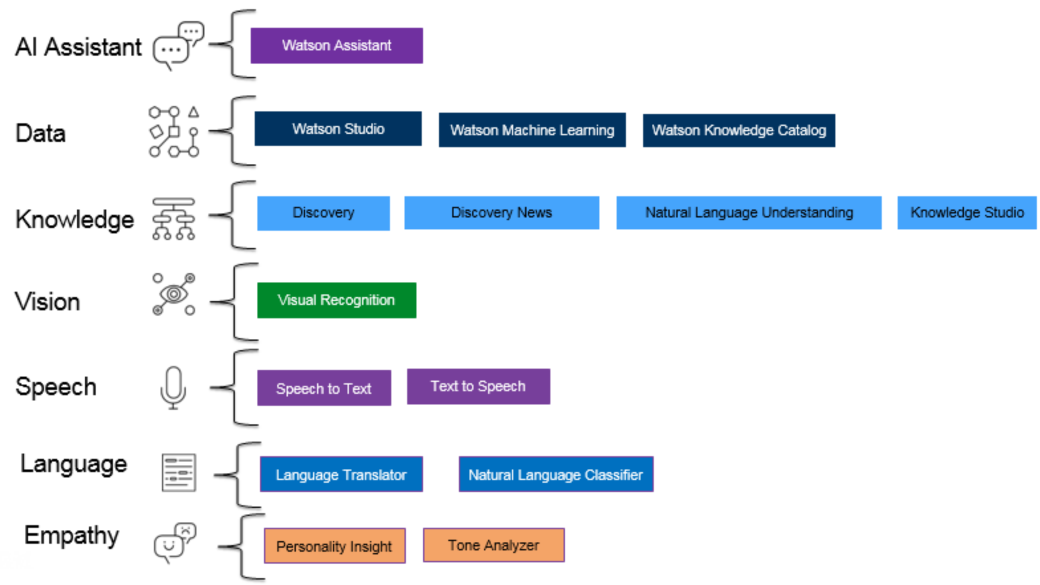
- IBM Watson overview
- IBM Watson services on IBM Cloud
- Build with Watson

Introduction to IBM Watson

Figure 2-9. Topics

IBM Training

Watson services on IBM Cloud - Summary



Introduction to IBM Watson

Figure 2-10. Watson services on IBM Cloud - Summary

The figure shows the Watson services on IBM Cloud circa 2018. These services will continue to be enhanced and new services will be introduced in the future.

The following services are available as REST APIs and SaaS tools that developers can use to build AI solutions or add AI features to their applications. See the website https://www.ibm.com/watson/products-services/.

AI Assistant (Chatbot): Integrate diverse conversation technologies into your application:

- Watson Assistant: Quickly build a chat bot by using tools and dialog trees.

Data: Collect, organize and analyze your data, then achieve trust, scale, and automation across your full AI lifecycle:

- Watson Studio: collaborative environment with AI tools that a team can use to collect and prepare training data, and to design, train, and deploy machine learning models.
- Watson Machine Learning: Enables users to perform two fundamental operations of machine learning: training and scoring.
- Watson Knowledge Catalog: Machine learning data catalog (MLDC) that enables you to access, curate, categorize and share data, knowledge assets and their relationships, wherever they reside.

Knowledge: Get insights through accelerated data optimization capabilities:

- Discovery: Unlock hidden value in data to find answers, monitor trends, and surface patterns.
- Discovery News: Explore news and blogs with smarter news from Watson that includes concepts, sentiment, relationships and categories.
- Natural Language Understanding: NLP for advanced text analysis.
- Knowledge Studio: Teach Watson to discover meaningful insights in unstructured text.

Vision: Identify and tag content, and then analyze and extract detailed information that is found in an image:

- Visual Recognition: Tag and classify visual content by using machine learning.

Speech: Converts text and speech with the ability to customize models:

- Speech to Text: Easily converts audio and voice into written text.
- Text to Speech: Converts written text into natural-sounding audio.

Language: Analyzes text and extracts metadata from unstructured content:

- Language Translator: Translates text from one language to another.
- Natural Language Classifier: Interprets and classifies natural language. Applies natural language processing and machine learning techniques to return the best matching classes for a sentence or phrase.

Empathy: Understands tone, personality, and emotional state:

- Personality Insights: Predicts personality characteristics through text.
- Tone Analyzer: Understands emotions and communication style in text.

IBM Training

Watson Assistant (formerly Conversation)

- Adds a natural language interface to your application to automate interactions with your users.
- Build the conversation flow and train the service by using an easy-to-use web interface.
- Integrate Watson Assistant with other Watson services to enrich the chatbot interaction experience.
- Example applications: Virtual agents and chat bots.

Introduction to IBM Watson

Figure 2-11. Watson Assistant (formerly Conversation)

Watson Assistant

IBM Watson Assistant is a cognitive bot that you can customize for your business needs, and deploy across multiple channels to bring help to your customers where and when they need it.

You can add a natural language interface to your application to automate interactions with your users.

Example applications include virtual agents and chat bots that can integrate and communicate on any channel or device, including mobile devices, messaging platforms, and robots.

Train the Watson Assistant service by using an easy-to-use web interface so that you can quickly build natural conversation flows between your apps and users, and deploy scalable and cost-effective solutions.

IBM Training

Watson Discovery

- Adds cognitive search and content analytics to applications to identify patterns, trends, and insights.
- Unlocks actionable insights into unstructured data.
- Unifies structured and unstructured data.
- Uses simple query language to eliminate the need for manual filtering of results.
- Includes the Discovery API and Discovery tooling.
- Example application: Find answers to FAQs

Introduction to IBM Watson

Figure 2-12. Watson Discovery

Watson Discovery

Add a cognitive search and content analytics engine to applications to identify patterns, trends, and actionable insights that drive better decision-making. Rapidly build cognitive, cloud-based exploration applications that unlock actionable insights that are hidden in unstructured data. Securely unify structured and unstructured data with pre-enriched content, and use a simplified query language to eliminate the need for manual filtering of results.

With Discovery, you can prepare your unstructured data, create a query that will pinpoint the information you need, and then integrate those insights into your new application or existing solution.

The Discovery services includes:

- Discovery API: The Discovery service supports a number of SDKs to simplify the development of applications. The SDKs are available for many popular programming languages and platforms, including Node.js, Java, and Python.

- Discovery tooling: The Discovery service includes a complete set of online tools - the Discovery tooling - to help you quickly setup an instance of the service and populate it with data. The Discovery service tooling has been designed to save time by eliminating the need to use APIs to configure and populate your service.

With Discovery you can build applications that extract the correct answers to FAQs by enriching and searching data collections.

IBM Training

IBM

Natural Language Understanding

- Analyze semantic features of text input, including the following items:
  - Concepts, Entities, Keywords, Categories, Sentiment, Emotion, Relations, Semantic roles
- Categorize content.
- Develop custom annotation models to identify domain-specific entities and relations in unstructured text by using Knowledge Studio.
- Example applications: Categorize news articles and blog posts and sort them based on general concepts, keywords, and entities.

Introduction to IBM Watson

Figure 2-13. Natural Language Understanding

Natural Language Understanding

Analyze text to extract metadata from content such as concepts, entities, keywords, categories, sentiment, emotion, relations, semantic roles.

Categorize your content using a five-level classification hierarchy. View the complete list of categories at https://console.bluemix.net/docs/services/natural-language-understanding/categories.html#categories-hierarchy.

Custom annotation models are developed by using Watson Knowledge Studio to identify industry- and domain-specific entities and relations in unstructured text.

Example applications: Categorize news articles and blog posts and sort them based on general concepts, keywords, and entities.

IBM Training

Natural Language Classifier

- Applies AI techniques to return the best matching classes for a sentence or phrase.
- You create a classifier instance by providing a set of representative strings and a set of one or more correct classes for each training.
- After training, the new classifier can accept new questions or phrases and return the top matches with a probability value for each match.
- Example applications:
  - Banking and finance: Classify investments, risks, and transactions.
  - Services: Categorize service queries, messages, and responses to help address problems and deploy solutions quicker.
  - Email: Classify mail as spam or non-spam.

Introduction to IBM Watson

Figure 2-14. Natural Language Classifier

Natural Language Classifier

The Natural Language Classifier service applies AI techniques to return the best matching classes for a short text (sentence or phrase). For example, you submit a question and the service returns keys to the best matching answers or next actions for your application.

You create a classifier instance by providing a set of representative strings and a set of one or more correct classes for each training. After training, the new classifier can accept new questions or phrases and return the top matches with a probability value for each match.

Example applications:

- Banking and finance: Classify investments, risks, and transactions.
- Services: Categorize service queries, messages, and responses to help address problems and deploy solutions quicker.
- Email: Classify mail as spam or non-spam.

IBM Training

Visual Recognition

- Uses deep learning algorithms to analyze images.
- Analyzes the images for scenes, objects, faces, colors, food, text, explicit content, and other subjects that provide insights into the visual content.
- A set of built-in models provides accurate results without training:
  - General model: Default classification from thousands of classes.
  - Face model: Facial analysis with age and gender.
  - Explicit model: Whether an image is inappropriate for general use.
  - Food model: Specifically for images of food items.
  - Text model (Private beta): Text extraction from natural scene images.
- Train custom models to create specialized classes.

Introduction to IBM Watson

Figure 2-15. Visual Recognition

Visual Recognition

Visual Recognition understands the content of images. This service uses deep learning algorithms to analyze images for scenes, objects, faces, colors, food, text, explicit content, and other subjects that provide insights into the visual content. The response includes keywords that provide information about the content.

A set of built-in models provides highly accurate results without training:

- General model: Default classification from thousands of classes.
- Face model: Facial analysis with age and gender.
- Explicit model: Whether an image is inappropriate for general use.
- Food model: Specifically for images of food items.
- Text model (Private beta): Text extraction from natural scene images

The Visual Recognition service can be used for diverse applications and industries, such as these:

- Manufacturing: Use images from a manufacturing setting to make sure products are being positioned correctly on an assembly line.
- Visual Auditing: Look for visual compliance or deterioration in a fleet of trucks, planes, or windmills in the field, train custom models to understand what defects look like.
- Insurance: Rapidly process claims by using images to classify claims into different categories.
- Social listening: Use images from your product line or your logo to track buzz about your company on social media.
- Social commerce: Use an image of a plated dish to find out which restaurant serves it and find reviews, use a travel photo to find vacation suggestions based on similar experiences, use a house image to find similar homes that are for sale.
- Retail: Take a photo of a favorite outfit to find stores with those clothes in stock or on sale, use a travel image to find retail suggestions in that area, use the photo of an item to find out its price in different stores.
- Education: Create image-based applications to educate about taxonomies, use pictures to find educational material on similar subjects.
- Public safety: Automated, real-time video stream analysis to include targeted observations such as facial recognition and automated license-plate reading, identify a suspect's car with unknown whereabouts to locate instances of that model, parked or in motion, in any surveilled part of the country.

IBM Training

Speech to Text

- Converts human voice into the corresponding text.
- Uses machine intelligence to combine information about grammar and language structure with knowledge of the composition of the audio signal to generate an accurate transcription.
- Provides APIs that you can use to add speech transcription capabilities to your applications.
- The languages that are currently available are English (US and UK), Japanese, Arabic, Mandarin Chinese, Portuguese (Brazil), Spanish, French, German, and Korean.
- Example applications:
  - Voice control of applications, embedded devices, and vehicle accessories
  - Transcribing meetings and conference calls
  - Dictating email messages and notes

Introduction to IBM Watson

Figure 2-16. Speech to Text

Speech to Text

The Speech to Text service converts the human voice into the corresponding text. Use this service to convert audio and voice into text for quick understanding of content. It can be used anywhere there is a need to bridge the gap between the spoken word and their written form, including voice control of embedded systems, transcription of meetings and conference calls, and dictation of email and notes. This easy-to-use service uses machine intelligence to combine information about grammar and language structure with knowledge of the composition of the audio signal to generate an accurate transcription.

The following languages are currently available: English (US), English (UK), Japanese, Arabic (Modern Standard Arabic (MSA), Broadband model only), Mandarin Chinese, Portuguese (Brazil), Spanish, French (Broadband model only), German (Broadband model only), and Korean.

The Speech to Text service can take an audio voice file as input and convert the audio into written text.

This service can be used in practical applications such as these:

- Transcribe calls in a contact center to identify what is being discussed, when to escalate calls, and to understand content from multiple speakers.
- Transcribe a technical explanation provided by a subject matter expert to help to create documentation.
- Transcribe speech from a movie and create subtitles or captions.
- Transcribe voice mail messages to obtain a written log of messages left, for example, by clients or patients.
- Create voice control applications, for example an automated home control system that is based on your speech. Functions can include switching the light on and off, controlling electronic devices, or even opening the front door remotely.

IBM Training

Text to Speech

- Converts written text into natural sounding audio in various languages and voices.
- The currently available languages are Portuguese (Brazil), English (UK and US), French, German, Italian, Japanese, and Spanish.
- Example applications: Voice-driven and screenless applications, where audio is the preferred method of output:
  - Interfaces for the disabled, such as assistance tools for the vision-impaired
  - Reading text and email messages aloud to drivers
  - Video-script narration and video voice over
  - Reading-based educational tools
  - Home-automation solutions

Introduction to IBM Watson

Figure 2-17. Text to Speech

Text to Speech

Use the Watson Text to Speech API to convert written text into natural sounding audio in various languages and voices. The Text to Speech service processes text and natural language to generate synthesized audio output with the appropriate cadence and intonation.

The service supports voices in the following languages: Brazilian Portuguese, English (UK and US dialects), French, German, Italian, Japanese, and Spanish (Castilian, Latin American, and North American dialects). The service offers at least one male or female voice, sometimes both, for each language.

The service is appropriate for voice-driven and screenless applications, where audio is the preferred method of output:

- Interfaces for the disabled, such as assistance tools for the vision-impaired
- Reading text and email messages aloud to drivers
- Video-script narration and video voice over
- Reading-based educational tools
- Home-automation solutions

This service can be used in practical applications such as these:

- Create audio narration from a written script for a variety of applications, such as online tutorials for an e-learning course, audio books, and so on.
- Provide callers with information, such as company location, store hours, and account information that is extracted from a database or organization’s documentation and converted to audible speech.
- Develop interactive educational material for children.
- Communicate directions, hands-free.

IBM Training

IBM

Language Translator

- Identifies the language of text and translates it into different languages programmatically.
- High-quality, domain-specific text translation.
- The following domains and languages are supported:
  - The News domain to translate English to and from French, Spanish, Portuguese, or Arabic.
  - The Conversational domain to translate English to and from Spanish or Portuguese.
  - The Patent domain to translate Spanish, Portuguese, Japanese, or Korean to English.
- Extend provided models to learn custom terms and phrases.
- Example application: Enable a help desk representative to assist international customers through chat.

Introduction to IBM Watson

Figure 2-18. Language Translator

Language Translator

This service provides high-quality and domain-specific text translation from one language to another. It is one of the few offerings that focuses on domain-specific Statistical Machine Translation.

The following domains and languages are supported:

- The News domain to translate English to and from French, Spanish, Portuguese, or Arabic.
- The Conversational domain to translate English to and from Spanish or Portuguese.
- The Patent domain to translate Spanish, Portuguese, Japanese, or Korean to English

You can create your own custom translation model. Most of the provided translation models in Language Translator can be extended to learn custom terms and phrases or a general style that is derived from your translation data.

Example application: Enable a help desk representative to assist international customers through chat.

IBM Training

Personality Insights

- Infers personality characteristics that can be extracted from text based on three primary models:
  - **Big Five** personality characteristics: Agreeableness, Conscientiousness, Extraversion, Emotional range, and Openness.
  - **Needs** describe which aspects of a product are likely to resonate with a person: Excitement, Harmony, Curiosity, Ideal, Closeness, Self-expression, Liberty, Love, Practicality, Stability, Challenge, and Structure.
  - **Values** describe motivating factors that influence a person's decision making: Self-transcendence / Helping others, Conservation / Tradition, Hedonism / Taking pleasure in life, Self-enhancement / Achieving success, and Open to change / Excitement.
- Example applications:
  - Brand analytics and customer care
  - Market segmentation and campaigns

Introduction to IBM Watson

Figure 2-19. Personality Insights

Personality Insights

The Personality Insights service infers personality characteristics based on three primary models:

- **Big Five** personality characteristics represent the most widely used model for generally describing how a person engages with the world. The model includes five primary dimensions: Agreeableness, Conscientiousness, Extraversion, Emotional range, and Openness. Each dimension has six facets that further characterize an individual according to the dimension.
- **Needs** describe which aspects of a product are likely to resonate with a person. The model includes twelve characteristic needs: Excitement, Harmony, Curiosity, Ideal, Closeness, Self-expression, Liberty, Love, Practicality, Stability, Challenge, and Structure.
- **Values** describe motivating factors that influence a person's decision making. The model includes five values: Self-transcendence / Helping others, Conservation / Tradition, Hedonism / Taking pleasure in life, Self-enhancement / Achieving success, and Open to change / Excitement.

Watson Personality Insights derives insights from transactional and social media data to identify psychological traits that determine purchase decisions, intent, and behavioral traits.

Use the Watson Personality Insights API in your applications to predict personality characteristics, needs, and values that are extracted from written text.

- Enables deeper understanding of people's personality characteristics, needs, and values to help engage users on their own terms.
- Extracts a set of personality and social traits based on the way a person communicates: Agreeableness, openness, conscientiousness, introversion and extraversion, and emotional range.

Examples applications:

• Brand analytics and customer care

• Market segmentation and campaigns

IBM Training

Tone Analyzer

- Tone impacts the effectiveness of communication in different contexts.
- Watson Tone Analyzer uses linguistic analysis to identify various tones at both the sentence and document level.
- It detects three types of tones from text:
  - Emotion (anger, disgust, fear, joy, and sadness)
  - Social tendencies (openness, conscientiousness, extroversion and introversion, agreeableness, and emotional range)
  - Language styles (analytical, confident, and tentative)
- Example applications:
  - Understand how written communications are perceived and then improve the tone of the communications.
  - Businesses can learn the tone of their customers' communications and then respond appropriately to each customer.

Introduction to IBM Watson

Figure 2-20. Tone Analyzer

Tone Analyzer

People show various tones, such as joy, sadness, anger, and agreeableness, in daily communications. Such tones can impact the effectiveness of communication in different contexts.

Watson Tone Analyzer uses linguistic analysis to identify various tones at both the sentence and document level. This insight can then be used to refine and improve communications.

It detects three types of tones:

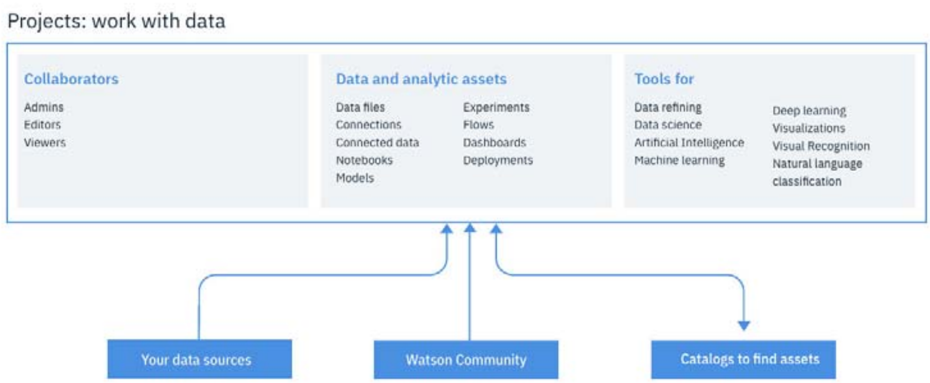
- Emotion (anger, disgust, fear, joy, and sadness)
- Social propensities (openness, conscientiousness, extroversion and introversion, agreeableness, and emotional range)
- Language styles (analytical, confident, and tentative) from text

Use the Watson Tone Analyzer API in your applications to understand emotions, social tendencies, and perceived writing style.

IBM Training

Watson Studio

- Collaborative environment with AI tools to collect and prepare training data, and to design, train, and deploy machine learning models.
- It is a SaaS solution that is delivered on IBM Cloud.
- Watson Studio AI tools support popular frameworks, including: TensorFlow, Caffe, PyTorch, and Keras.
- The architecture of Watson Studio is centered around the project.



Introduction to IBM Watson

Figure 2-21. Watson Studio

Watson Studio

IBM Watson Studio is a collaborative environment with AI tools that a team can use to collect and prepare training data, and to design, train, and deploy machine learning models.

It is a SaaS solution that is delivered on IBM Cloud.

Watson Studio provides a suite of tools for data scientists, application developers, and subject matter experts (SMEs) to work collaboratively and easily with data. They can then use that data to build, train, and deploy models at scale. These tools are preconfigured so that builders do not have to spend time installing, setting up, and maintaining them. The built-in catalog function enables knowledge sharing and retention. Watson Studio can infuse AI into your business.

It enables you to analyze data by using RStudio, Jupyter, and Python in a configured and collaborative environment that includes added value, such as managed Spark and IBM Watson Machine Learning.

The architecture of Watson Studio is centered around the project. A project is where you organize your resources and work with data.

You can think of Watson Studio AI tools in four categories:

- Visual recognition
- Natural language classification
- Machine learning
- Deep learning

For more information see
https://dataplatform.cloud.ibm.com/docs/content/getting-started/overview-ws.html?context=analytics

IBM Training

Watson Machine Learning

- Machine Learning is a service on IBM Cloud with features for training and deploying machine learning models and neural networks. It provides:
  - **Interfaces** for building, training, and deploying models: Python client library external link, Command line interface, REST API external link.
  - **Deployment infrastructure** for hosting your trained models.
  - **Hyperparameter optimization** for training complex neural networks.
  - **Distributed deep learning** for distributing training runs across multiple servers.
  - **GPUs** for faster training.

Introduction to IBM Watson

Figure 2-22. Watson Machine Learning

Watson Machine Learning

IBM Watson Machine Learning is a full-service IBM Cloud offering that makes it easy for data scientists and developers to work together to integrate predictive analytics with their applications.

The Watson Machine Learning service enables your organization to use the models in your end-to-end solutions without the impact of licensing, installation, and configuration that is required by the same products when they are installed on-premises. You can use machine learning and deep learning models in production. Use an automated and collaborative workflow to grow intelligent business applications easily and with more confidence.

Using Watson Machine Learning, you can build sophisticated analytical models, trained with your own data, that you can deploy for use in applications.

Machine Learning is a service on IBM Cloud with features for training and deploying machine learning models and neural networks:

- **Interfaces** for building, training, and deploying models: Python client library external link, Command line interface, REST API external link
- **Deployment infrastructure** for hosting your trained models

  After you create, train, and evaluate a model, you can deploy it. When you deploy a model you save it to the model repository that is associated with your Watson Machine Learning service. Then, you can use your deployed model to score data and build an application.
- Hyperparameter optimization for training complex neural networks

  You can run your experiments with HPO to easily find the best quality model. Hyperparameter Optimization (HPO) is a mechanism for automatically exploring a search space of potential Hyperparameters, building a series of models and comparing the models using metrics of interest. To use HPO you must specify ranges of values to explore for each Hyperparameter.
- **Distributed deep learning** for distributing training runs across multiple servers

  Deep learning models training can be significantly accelerated with distributed computing on GPUs.
- **GPUs** for faster training

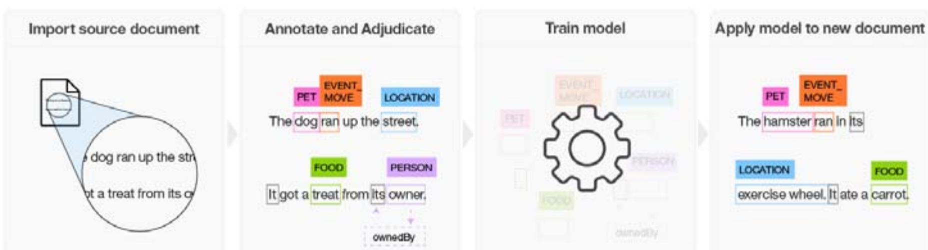
  IBM Watson Machine Learning deep learning simplifies the process to train models in parallel with an on-demand graphics processing units (GPU) compute cluster that you can scale to your specific needs.

For an overview of Watson Machine Learning, see: https://youtu.be/5kMDIBpxi_k

IBM Training

IBM Watson Knowledge Studio

- Enables developers and domain experts to collaborate and create a machine learning model that understands the linguistic nuances, meaning, and relationships specific to an industry or domain.
- Provides easy-to-use tools for annotating unstructured domain literature.
- The annotations are used to create a custom machine learning model that understands the language of the domain.
- The model can be deployed directly to IBM Watson Explorer, Watson Natural Language Understanding, and Watson Discovery.



Introduction to IBM Watson

Figure 2-23. IBM Watson Knowledge Studio

IBM Watson Knowledge Studio

IBM Watson Knowledge Studio enables developers and domain experts to collaborate and create a machine learning model that understands the linguistic nuances, meaning, and relationships specific to an industry or domain.

Provides easy-to-use tools for annotating unstructured domain literature, and uses those annotations to create a custom machine-learning model that understands the language of the domain.

These annotators can identify mentions and relationships in unstructured data and be easily administered throughout their lifecycle by using one common tool.

Annotator components can be deployed directly to IBM Watson Explorer, Watson Natural Language Understanding, and Watson Discovery.

The diagram illustrates how it works:

1. Based on a set of domain-specific source documents, the team creates a type system that defines entity types and relation types for the information of interest to the application that will use the model.
2. A group of two or more human annotators annotates a small set of source documents to label words that represent entity types, to identify relation types where the text identifies relationships between entity mentions, and to define coreferences, which identify different mentions that refer to the same thing, that is, the same entity. Any inconsistencies in annotation are resolved, and one set of optimally annotated documents is built, which forms the ground truth.
3. Watson Knowledge Studio uses the ground truth to train a model.

2.3. Build with Watson

IBM Training

Build with Watson

Introduction to IBM Watson

Figure 2-24. Build with Watson

This section provides the initial steps to get started using the Watson APIs. It also lists the key resources available to developers.

IBM Training IBM

Topics

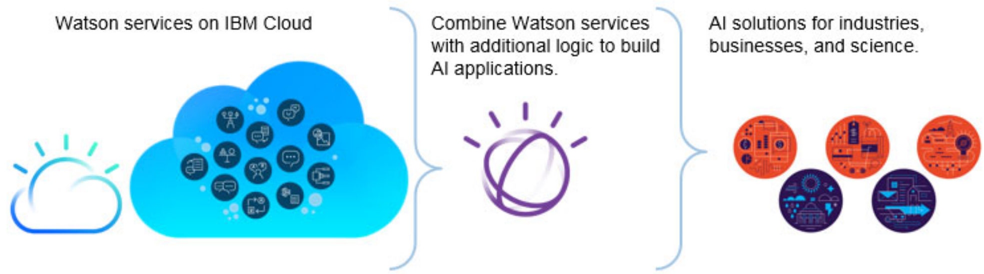
- IBM Watson overview
- IBM Watson services on IBM Cloud
- Build with Watson

Introduction to IBM Watson

Figure 2-25. Topics

IBM Training

Building AI solutions with IBM Watson services on IBM Cloud



Introduction to IBM Watson

Figure 2-26. Building AI solutions with IBM Watson services on IBM Cloud

Watson services on IBM Cloud provide a cloud-hosted marketplace where application providers of all sizes and industries can tap into resources for developing applications that are powered by Watson services. Developers can combine the Watson services (and other services that are available in IBM Cloud) with additional logic to build applications with AI capabilities.

The goal of Watson services on the IBM Cloud is to provide flexible platform for building AI applications in industry domains. The microservices architecture enables developers to envision a broad range of potential applications by mixing and matching services.

The available resources include developer toolkits, educational materials, and access to Watson APIs and SaaS tools. This approach makes IBM Watson technology available as a development platform in the cloud to enable a worldwide community of software application providers to build a new generation of applications that is infused with Watson AI capabilities.

IBM Training

Watson services and domain adaptation

- AI systems must be trained to understand new domains and perform new tasks.
- Built-in models included in Watson services are generic or geard towards popular applications of the service.
  - The goal is to enable the initial use of the service with no training but the accuracy for a particular application might not be satisfactory.
- Services must be trained to improve their performance for specific application.
- Some Watson services cannot be trained by users.
  - IBM is in charge of training these services.

Introduction to IBM Watson

Figure 2-27. Watson services and domain adaptation

One key element of AI systems is the capability to learn and adapt overtime. Rather than being explicitly programmed, AI systems learn from their interactions with their users and from their experiences with their environment. Machine learning gives computers the ability to learn and act without being explicitly programmed. This means that the model gets better over time by learning from its mistakes and new experiences (being exposed to new data). When developing machine learning models, the models are built from a fixed source, for example, open domain Wikipedia, and they are deployed to similar or different domains, for example the Travel domain. To improve accuracy, the new domain data must be used to train the model. This task is known as *domain adaptation*.

Like humans, AI systems must be trained to understand new domains and perform new tasks. For example, understanding medical records to identify medical conditions and associated prescriptions requires deep knowledge of drugs and diseases. In order to be able to perform these tasks, humans go to college, get a medical degree and, after many years of training and study, become doctors. Likewise, AI systems must be trained to

become experts in specific domains. Training is performed by subject matter experts (SMEs) providing human supervision and domain-specific knowledge bases representing entities and relations of interest for the new domain. A similar process must be followed to apply Watson technology to specific domains.

Domain adaptation consists of the necessary activities to adapt an open-domain system to a specific domain (a closed-domain).

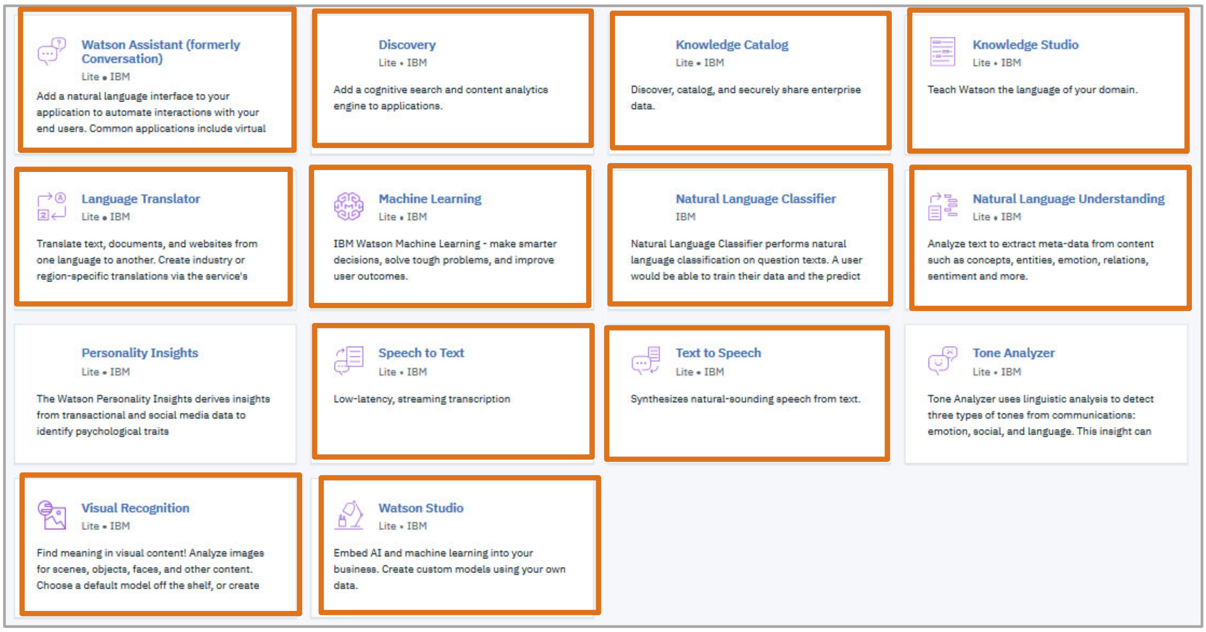
Information extraction analytics that are provided by Watson APIs are open domain, which means that they can recognize named entities that belong to basic types, such as company, person, and location, but they cannot recognize more specific distinctions, such as names of banks, insurance companies, and their products. Likewise, the built-in models in the Visual Recognition service can classify thousands of classes for popular images, faces, food but the models must be trained to improve performance for a specific images classification.

To become an SME in a specific industry or domain, some Watson services must be trained.

IBM Training

Watson services and domain adaptation (cont.)

- Services that **can** be trained by the user.



Introduction to IBM Watson

Figure 2-28. Watson services and domain adaptation (cont.)

This figure shows the Watson services that you can train to adapt them to a closed domain:

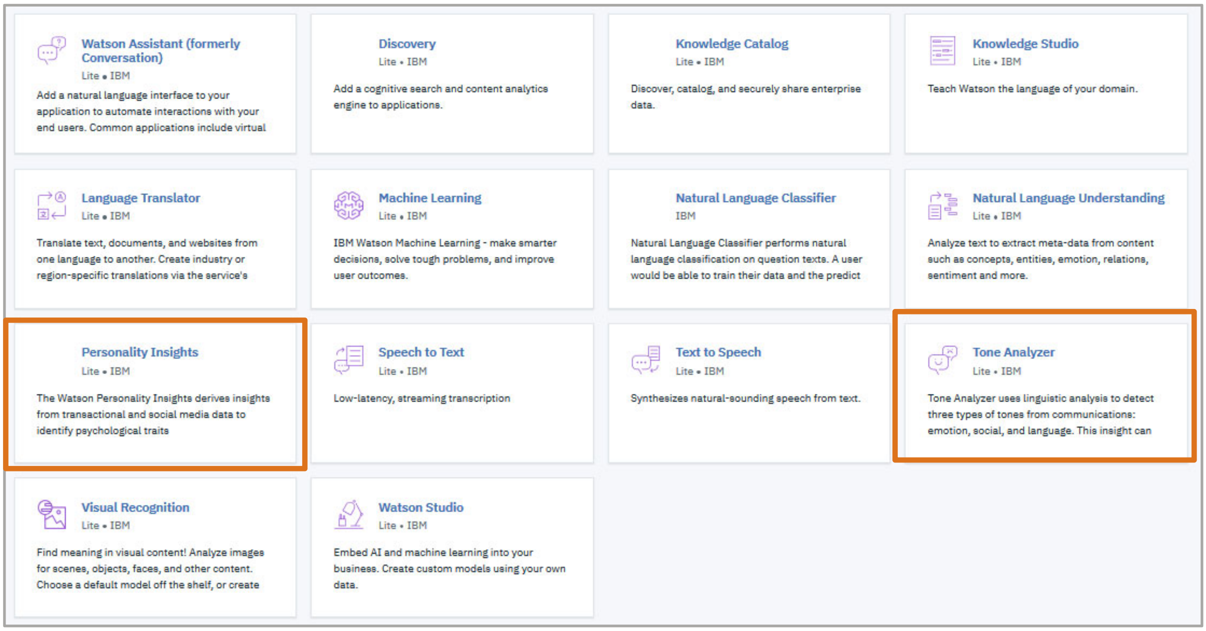
- Watson Assistant: This service must be trained to recognize what the user is saying and respond accordingly.
- Discovery: The relevance of natural language query results can be improved in the Discovery service with training. Relevancy training teaches the service what results should be surfaced higher than others so that your users can get the correct answer to their question faster. You also can use IBM Watson Knowledge Studio to train a custom model and deploy it to the Discovery service.
- Language Translator service: This service can be trained over time to provide better accuracy when running translations. To train, the service must learn from previous translations. For example, Language Translator can be trained to reduce the error in translating financial news across languages. Another example is that you are creating a translator for customer support and you have company-specific terms that you want handled in a certain way in conversations. A third example is that you are creating a way for your engineers in one country to look up patent data in another language, and you often file patents on a specific technology. You can use your own data to create a custom dictionary and a custom translation model in the Language Translator service.

- Natural Language Classifier: This service can be trained to identify new classes in unstructured text, such as financial news from blog posts.
- Natural Language Understanding: This service is trained on an open domain by default. With custom annotation models that are developed by using IBM Watson Knowledge Studio, you can further customize the service to identify domain-specific entities and relations in your content.
- Speech to Text: This service can be customized for a particular domain by creating a language model to provide the nuances in that domain in terms of vocabularies and word pronunciations. By training the service, you can improve the accuracy of speech recognition for domains such as medicine, law, information technology, and others.
- Text to Speech: This service provides a customization interface so that you can create a dictionary of words and their translations for a specific language. This dictionary of words and their translations is referred to as a custom voice model.
- Visual Recognition: This service includes a set of built-in classes that provides highly accurate results without training. You can also train and create a custom models (or custom classifiers). With a custom model, you can train the Visual Recognition service to classify images to suit your business needs. By creating a custom model, you can use the Visual Recognition service to recognize images that are not available with pre-trained classification, for example, to recognize company logos in pictures.
- Knowledge Studio: This service is used to train Watson with the language of your domain by using custom machine learning models that identify entities and relationships that are unique to your industry in unstructured text.

IBM Training

Watson services and domain adaptation (cont.)

- Services that **cannot** be trained by the user.



Introduction to IBM Watson

Figure 2-29. Watson services and domain adaptation (cont.)

The figure illustrates the Watson services that cannot be trained by the user. For these services, IBM is in charge of the required training.

IBM Training

Getting started with Watson services on IBM Cloud

You must complete these steps to use the Watson services on IBM Cloud:

1. Set up your IBM Cloud account.
2. Find and select the Watson service in the IBM Cloud catalog.
3. Create the service instance.
4. Get the service credentials to authenticate to your service from your application.
5. Start coding your application and calling the Watson APIs to infuse AI capabilities into your app.

Introduction to IBM Watson

Figure 2-30. Getting started with Watson services on IBM Cloud

To get started with Watson services on IBM Cloud, you must complete these steps before you can use the service in your applications:

1. Set up your IBM Cloud account.
2. Find and select the Watson service that you need from the IBM Cloud catalog.
3. Create a service instance. IBM Cloud provides resources to your applications through a service instance.
4. Get service credentials. You need the service credentials to authenticate to your service instance from your application.
5. Start coding your application and calling the Watson APIs to infuse AI capabilities into your app.

For more information about these steps, see Exercise 1.

IBM Training

Watson SDKs

- Enable developers to use Watson REST APIs in their favorite programming language.
- Developers should consider using the SDKs instead of calling the REST APIs directly.
- Watson SDKs supported by IBM:
  - Android SDK, Java SDK, Node.js SDK, Python SDK, Ruby SDK, Salesforce SDK, Swift SDK, .NET SDK, OpenWhisk SDK, and Unity SDK
- Watson SDKs from the Watson community of developers:
  - Go SDK, PHP SDK, Ruby wrappers, Scala SDK

Introduction to IBM Watson

Figure 2-31. Watson SDKs

IBM Training

Additional resources for developers

For more information access these resources:

- Explore the complete list of Watson APIs: Watson Products and Services
- Get started on IBM Cloud : IBM Cloud essentials
- Access developer's resources:
  - Build with Watson
  - Documentation and API Reference
  - Watson SDKs
  - Building Cognitive Applications with IBM Watson Services
  - AI articles and tutorials
  - Watson webinars
  - Building with Watson: Application Starter Kits for developers
  - Watson Starter Kits

Introduction to IBM Watson

Figure 2-32. Additional resources for developers

- Watson products and services: https://www.ibm.com/watson/products-services/
- IBM Cloud essentials: https://developer.ibm.com/courses/all/bluemix-essentials/
- Build with Watson: https://console.bluemix.net/developer/watson/dashboard
- Documentation and API Reference: https://console.bluemix.net/developer/watson/documentation
- Watson SDKs: https://console.bluemix.net/docs/services/watson/getting-started-sdks.html#sdks
- Building Cognitive Applications with IBM Watson Services: http://www.redbooks.ibm.com/redbooks.nsf/pages/cognitiveapps?Open
- AI articles and tutorials: https://developer.ibm.com/technologies/artificial-intelligence/
- Watson webinars: https://www.ibm.com/watson/webinars/
- Building with Watson: Application Starter Kits for developers: https://developer.ibm.com/tv/building-with-watson-application-starter-kits-for-developers/
- Watson Starter Kits: https://cloud.ibm.com/developer/watson/starter-kits

IBM Training

Unit summary

- Explain what IBM Watson is.
- List the Watson service.
- Explain the capabilities of each Watson service.
- Describe the purpose of training the various Watson services to adapt them to a closed-domain.
- List the Watson services that can be trained.
- List the Watson services that cannot be trained.
- Describe what Watson Knowledge Studio is.
- List the Watson services that can be trained with Watson Knowledge Studio.
- Create a Watson service instance.

Introduction to IBM Watson

Figure 2-33. Unit summary

IBM Training

Review questions

1. Enables applications to use natural language to automatically respond to user questions:
   A. Natural Language Understanding
   B. Watson Assistant
   C. Natural Language Classifier
   D. Discovery
   E. Speech to Text.

2. Provides tools that SMEs can use to annotate unstructured domain literature, and uses those annotations to create a custom machine-learning model that understands the language of the domain. :
   A. Watson Studio
   B. Watson Machine Learning
   C. Watson Knowledge Studio
   D. Watson Health

Introduction to IBM Watson

Figure 2-34. Review questions

IBM Training

Review questions (cont.)

3. True or False: The Personality Insights service analyzes text to extract metadata from content, such as concepts, entities, keywords, categories, sentiment, emotion, relations, and semantic roles.

4. The following Watson services cannot be trained by users:
   A. Watson Assistant.
   B. Personality Insights.
   C. Visual Recognition.
   D. Tone Analyzer.
   E. A and D.
   F. B and D.
   G. C and D.

Introduction to IBM Watson

Figure 2-35. Review questions (cont.)

IBM Training

Review questions (cont.)

5. This Watson service requires Watson Knowledge Studio to be customized for a specific domain:
   A. Visual Recognition.
   B. Natural Language Understanding.
   C. Watson Assistant.

Introduction to IBM Watson

Figure 2-36. Review questions (cont.)

IBM Training

Review answers

1. Enables applications to use natural language to automatically respond to user questions:
    A. Natural Language Understanding
    B. Watson Assistant
    C. Natural Language Classifier
    D. Discovery
    E. Speech to Text
2. Provides tools that SMEs can use to annotate unstructured domain literature, and uses those annotations to create a custom machine-learning model that understands the language of the domain. :
    A. Watson Studio
    B. Watson Machine Learning
    C. Watson Knowledge Studio
    D. Watson Health

Introduction to IBM Watson

Figure 2-37. Review answers

IBM Training IBM

Review answers (cont.)

3. True or False: The Personality Insights service analyzes text to extract metadata from content, such as concepts, entities, keywords, categories, sentiment, emotion, relations, and semantic roles.
   The answer is: False
   The Natural Language Understanding service performs this task.

4. The following Watson services cannot be trained by users:
   A. Watson Assistant.
   B. Personality Insights.
   C. Visual Recognition.
   D. Tone Analyzer.
   E. A and D.
   F. B and D.
   G. C and D

Introduction to IBM Watson © Copyright IBM Corporation 2019

Figure 2-38. Review answers (cont.)

IBM Training

Review answers (cont.)

5. This Watson service requires Watson Knowledge Studio to be customized for a specific domain:
   A. Visual Recognition.
   B. Natural Language Understanding.
   C. Watson Assistant.

Introduction to IBM Watson

Figure 2-39. Review answers (cont.)

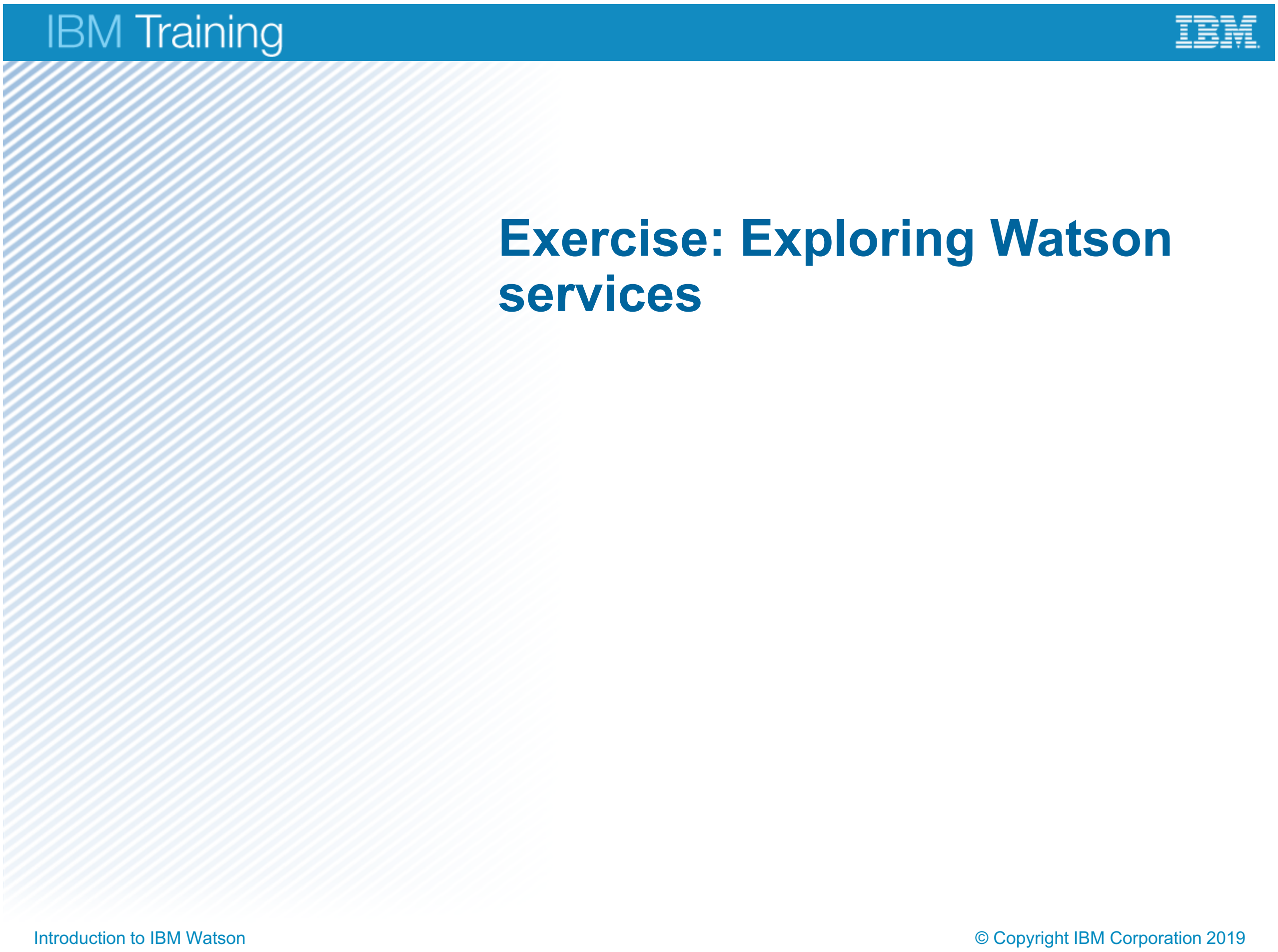


Figure 2-40. Exercise: Exploring Watson services

IBM Training

Exercise objectives

- This exercise introduces you to Watson REST APIs. You will use cURL to submit requests to and receive responses from the following Watson services:
  - Natural Language Understanding
  - Visual Recognition
  - Tone Analyzer
  - Language Translator
- After completing this exercise you should be able to:
  - Create Watson service instances.
  - Copy credentials from a service instance.
  - Submit API calls with the appropriate parameters.
  - Analyze the response that is returned from the Watson service.
  - Use Watson API reference documentation.

Introduction to IBM Watson

Figure 2-41. Exercise objectives

After completing this exercise, you should be able to accomplish the following tasks:

- Create Watson service instances.
- Copy credentials from a service instance.
- Submit API calls with the appropriate parameters.
- Analyze the response that is returned from the Watson service.
- Use Watson API reference documentation.

Unit 3. Introduction to IBM Watson Studio

Estimated time

00:30

Overview

This unit provides a high level overview of IBM Watson Studio, its components, key applications and the value added by the IBM offering.

IBM Training

Unit objectives

- Describe Watson Studio.
- Identify industry use cases.
- List Watson Studio offerings.
- Create Watson Studio projects.
- Describe Watson Studio and Spark.
- Describe Watson Studio and Object Storage.
- Explain Watson Studio high availability considerations.
- Prepare and analyze data.
- Use Jupyter notebooks.

Introduction to IBM Watson Studio

Figure 3-1. Unit objectives

3.1. Watson Studio overview

IBM Training

Watson Studio overview

Introduction to IBM Watson Studio

Figure 3-2. Watson Studio overview

IBM Training

Topics

- Watson Studio overview
- Watson Studio components
- Jupyter notebook

Introduction to IBM Watson Studio

Figure 3-3. Topics

IBM Training

What is Watson Studio

Watson Studio is a **collaborative platform** for data scientists, built on **open source components** and **IBM added value**, and is available in the **cloud** and **on-premises**.

- Collaborative platform:
  - Community environment for sharing resources, tutorials, and data sets
  - Simplified communication between different users and job roles
- Open-source components: Python, Scala, R, SQL, Apache Spark, and notebooks (Jupyter and Zeppelin)
- IBM added value: Watson Machine Learning, Flow editor, Decision Optimization, SPSS predictive analytics algorithms, analytics dashboard, and more
- Watson Studio Cloud, Watson Studio Local, and Watson Studio Desktop

Introduction to IBM Watson Studio

Figure 3-4. What is Watson Studio

Watson Studio is a **collaborative platform** for data scientists that is built on **open source components** and **IBM added value**, and is available in the **cloud** or **on-premises**. The collaborative platform enables the users, whether they are data scientists, data engineers, or application developers to share resources and work together seamlessly within the platform.

Watson Studio is built upon open source components such as Python, Scala, R, SQL, Apache Spark, and notebooks.

If the open source tools are not enough for your needs, IBM added value components such as Watson Machine Learning, Flow editor, Decision Optimization, SPSS predictive analytics algorithms, analytics dashboard, and more.

Watson Studio is available as three different offerings:

- Watson Studio Cloud, which is what you use in this course.
- Watson Studio Local, which is the on-premises version.
- Watson Studio Desktop is a light-weight version that you can install on your notebook.

This course uses Watson Studio Cloud.

For more information, see:

- Watson Studio Overview
https://dataplatform.cloud.ibm.com/docs/content/getting-started/overview-ws.html
- Decision Optimization https://dataplatform.cloud.ibm.com/docs/content/DO/DOinWStudio.html
- SPSS predictive analytics algorithms
https://dataplatform.cloud.ibm.com/docs/content/analyze-data/spss-algorithms.html?audience=wdp
- Analytics dashboard
https://dataplatform.cloud.ibm.com/docs/content/analyze-data/analytics-dashboard.html?audience=wdp

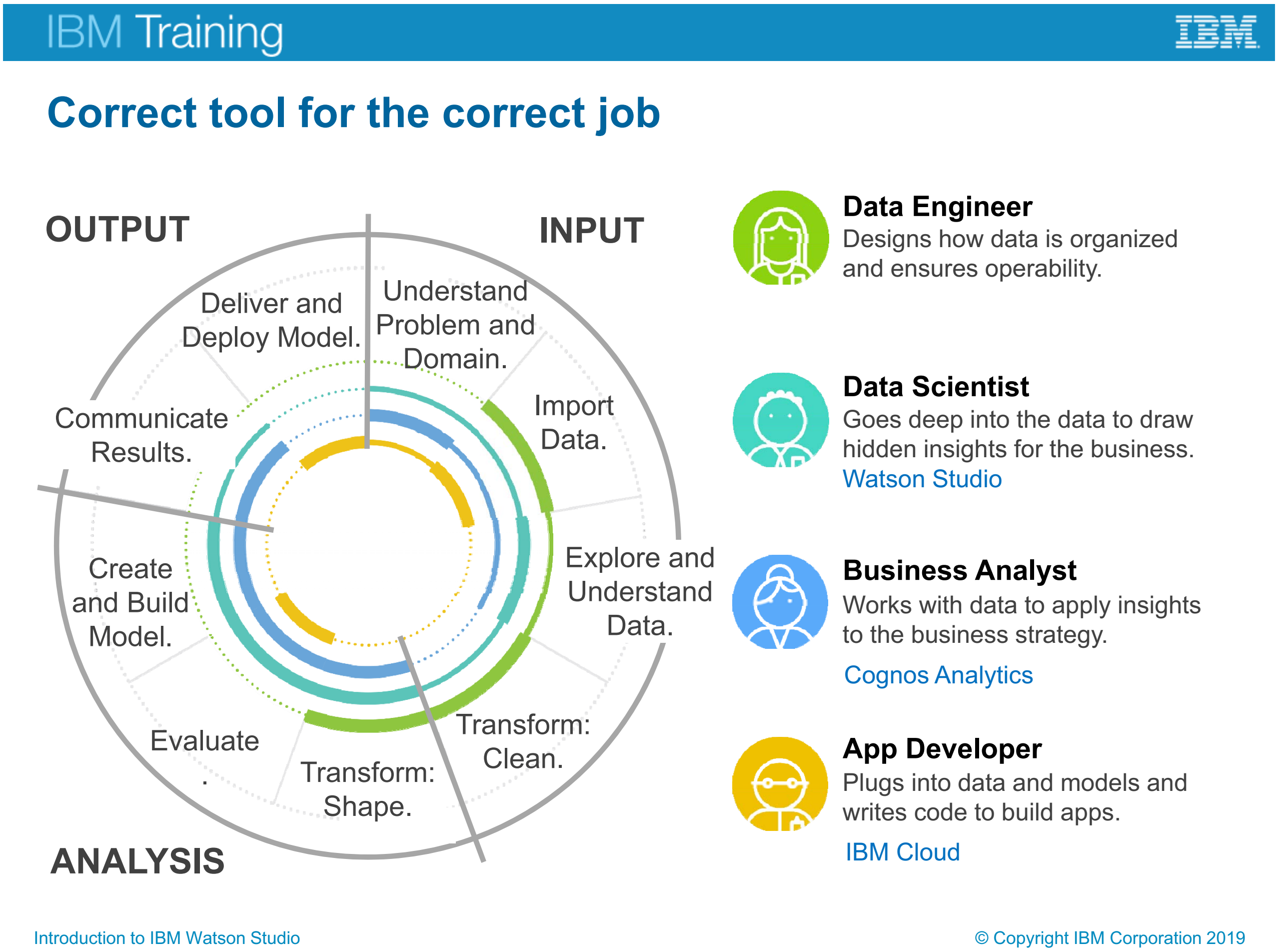


Figure 3-5. Correct tool for the correct job

Open-source tools give you a wide range of choices. The downside of having too many choices is knowing the correct one to pick. Essentially, you want to pick the correct tool for the correct job. Watson Studio is no exception.

Watson Studio is designed for a specific persona, but other personas can use it as it relates to their jobs.

Look at the diagram on the slide. Starting from the top and going clockwise, you have the input, analysis, and output phases. Within each phase are the objectives of that phase. Each objective can overlap between various user personas.

Look at the list of personas on the right side: the data engineer, the data scientist, the business analyst, and the app developer. Each persona has primary tools that helps them do their job. For example, the data scientist's main tool is Watson Studio.

Also, there might be a team of different personas. Whatever the case is, you must decide what tool is correct for the job, regardless of the personas. The definitions of personas can vary between different companies and evolve over time.

In addition to the tailored interface of Watson Studio, collaboration ties the team and the organization together by enabling them to share projects, code, and ideas. Imagine a data engineer builds a new data source and shares that asset with the data scientist and the business analyst. The business analyst immediately builds the reports and dashboards they need. The data scientist experiments with the data and builds a model that passes all the tests and can be used with new applications. The team can immediately share that model with the application developer, who deploys a new application by using the model. As this project progresses, the team members update and share their statuses with the team, asking questions and sharing ideas or requirements.

In this scenario, data and analytics development become a team effort and is no longer done in silos. Additionally, because these assets can be published, other departments can reuse these assets, making the entire organization more agile.

IBM Training

Industry use cases

Retail

Shopping experience
Loss & fraud prevention
Task & workforce optimization
Pricing & assortment optimization

Transportation

End-to-end customer experience
Operations planning & optimization
Predictive maintenance
Route and asset optimization

Banking

Optimize offers & cross-Sell
Risk management
Fraud and crime management
Financial performance

Media and Entertainment

Audience & fan insight
Social media insight
Content demand forecasting
Marketing & advertising optimization

Manufacturing

Inventory optimization
Predictive maintenance
Health, safety, & environment
Production planning and scheduling

Telco

Subscriber analytics
IOT analytics
Proactive marketing
Network design

Introduction to IBM Watson Studio

Figure 3-6. Industry use cases

Data science spans multiple industries, but you can see that data analysis applied to some of these use cases is not entirely new. In fact, organizations have been doing these types of activities for many years. The advantage that you have with Watson Studio is that you can easily collaborate with other data scientists by using well-known tools that are widely used in the industry.

Figure 3-7. IBM Watson Studio

Watson Studio is built as a collaborative platform. Watson Studio provides an easy way for you to learn how to start using the platform. You can create state-of-the-art products that are based on the data that you derive by using open source and IBM added value tools. As you innovate, you can collaborate with your team and the community to share and gain insights.

IBM Training

Watson Studio offerings

- Watson Studio Cloud (public cloud):
  - IBM Cloud subscription (https://datascience.ibm.com).
  - https://github.com/IBMDataScience.
- Watson Studio Local (private cloud and on-premises):
  - Run on your own clusters.
  - Same tools and features.
- Watson Studio Desktop (local workstation):
  - Download and try on your desktop.

Introduction to IBM Watson Studio

Figure 3-8. Watson Studio offerings

Watson Studio is available in the following offerings:

- Watson Studio Cloud is available through IBM Cloud as the public cloud option. To use it, you must have an IBM Cloud account. If you do not already have one, you will be directed to register with IBM Cloud the first time you attempt to access the Watson Studio environment.
- IBM Watson Studio Local is a ready-to-use enterprise solution for data scientists and data engineers. It offers a suite of data science tools, such as RStudio, Spark, Jupyter, and Zeppelin notebooks, that are integrated with proprietary IBM technologies. This offering can be installed on a private cloud or on-premises on the organization’s servers. It includes the same tools and features that are available in Watson Studio Cloud.
- Watson Studio Desktop is the local version of the tool that you install on your local machine. It enables you to experience the Watson Studio tooling before you decide on one of the other two options.

For more information, see the following resources:

Watson Studio Cloud:
https://dataplatform.cloud.ibm.com/docs/content/getting-started/overview-ws.html

Watson Studio Local:
https://content-dsxlocal.mybluemix.net/docs/content/SSAS34_current/local/overview.html

Watson Studio Desktop:
https://www.ibm.com/support/knowledgecenter/SSBFT6_1.1.0/wsd/overview.html

IBM Training

Watson Studio and other IBM Cloud services

- Watson Studio is a part of Watson Data Platform:
  - All Watson Data Platform services are seamlessly integrated and loosely coupled through IBM Cloud.
- Examples:
  - Watson Studio is aware of the Watson Machine Learning deployment service.
  - Watson Studio is aware of Db2 Warehouse on Cloud (formerly dashDB), Analytics Engine, and other data sources.
  - Watson Studio is aware of created Apache Spark services.
- The services do not depend on each other and some can be used as stand-alone (loosely coupled).

Introduction to IBM Watson Studio

Figure 3-9. Watson Studio and other IBM Cloud services

Watson Studio is part of the Watson Data Platform and integrates seamlessly, but coupled loosely, with other tools through IBM Cloud. Here are some examples about how Watson Studio can work with other tools:

- Watson Studio is aware of the Watson Machine Learning deployment service.
- Watson Studio is aware of Db2 Warehouse on Cloud (formerly dashDB), IBM Analytics Engine, and other data sources.
- Watson Studio is aware of created Apache Spark services.

Each service is independent of the other ones, so you use only what you need.

IBM Training

Watson Studio high availability (public cloud)

- Designed for 24x7 availability: Continuous availability and continuous delivery
- Backup and recovery:
  - Watson Studio is disaster-resistant.
  - Notebooks in Watson Studio are stored in a three-way Cloudant cluster in multiple geographic zones.
  - Watson Studio provides integration with GitHub and an interface for downloading notebooks if the customer wants to use their own backup.

Introduction to IBM Watson Studio

Figure 3-10. Watson Studio high availability (public cloud)

Watson Studio is designed for 24x7 availability, that is, continuous delivery and availability. Features and updates are rolled out without downtime. Notebooks in Watson Studio are stored in a three-way Cloudant cluster in multiple geographic zones. Watson Studio also provides integration with GitHub so that you can manually download the notebooks if you want to use your own backups.

3.2. Watson Studio components

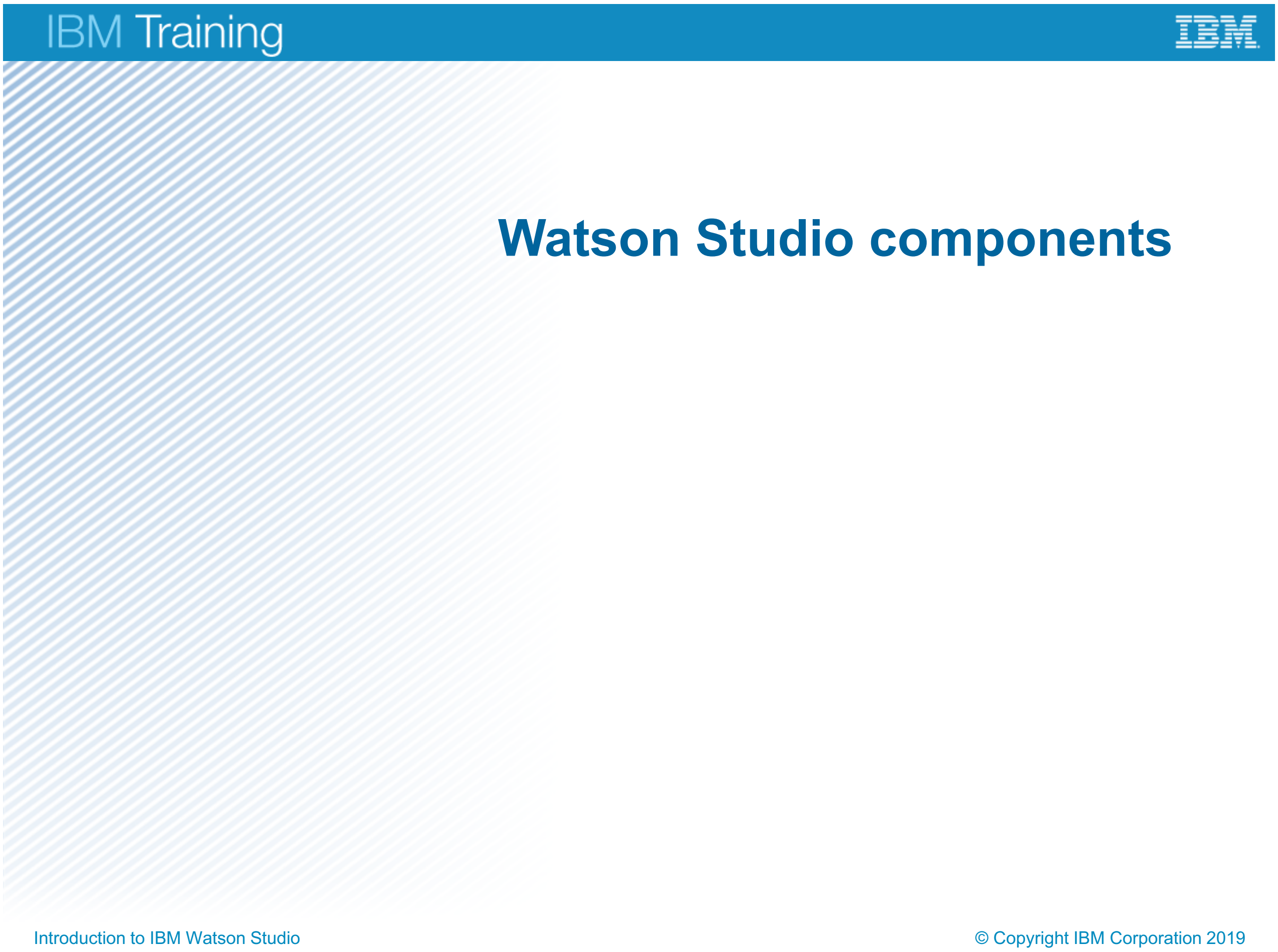


Figure 3-11. Watson Studio components

IBM Training

Topics

- Watson Studio overview
- Watson Studio components
- Jupyter notebook

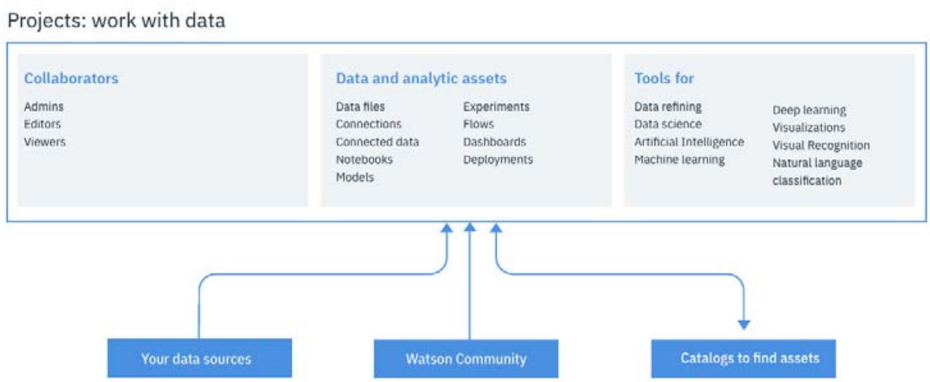
Introduction to IBM Watson Studio

Figure 3-12. Topics

IBM Training

Projects

- The architecture of Watson Studio is centered around the project.
- Projects are a way to organize resources for a specific data science task or goal.
- Integrate collaborators, data and analytic assets, tools, community resources, your own data and so on, to support finding insights for a well defined or narrow goal.



Introduction to IBM Watson Studio

Figure 3-13. Projects

The architecture of Watson Studio is centered on projects where everything is seamlessly integrated. You create and organize projects to suit your business needs. Projects consist of data assets, collaborators, analytic assets, and community resources that are combined with many open source and added value tools.

Data assets are the files in your object store or connections, such as a database, data services, streaming data, and other external files.

Collaborators can be assigned to your projects as admins, editors, or viewers.

Analytic assets are the notebooks and the models that you develop.

Watson Studio has a suite of tools that is available to help you with your job in the open source space. It also has a suite of added value tools such as Decision Optimization Watson Machine Learning, and Streaming Analytics.

You can think of Watson Studio AI tools in four categories:

- Visual recognition
- Natural language classification
- Machine learning
- Deep learning

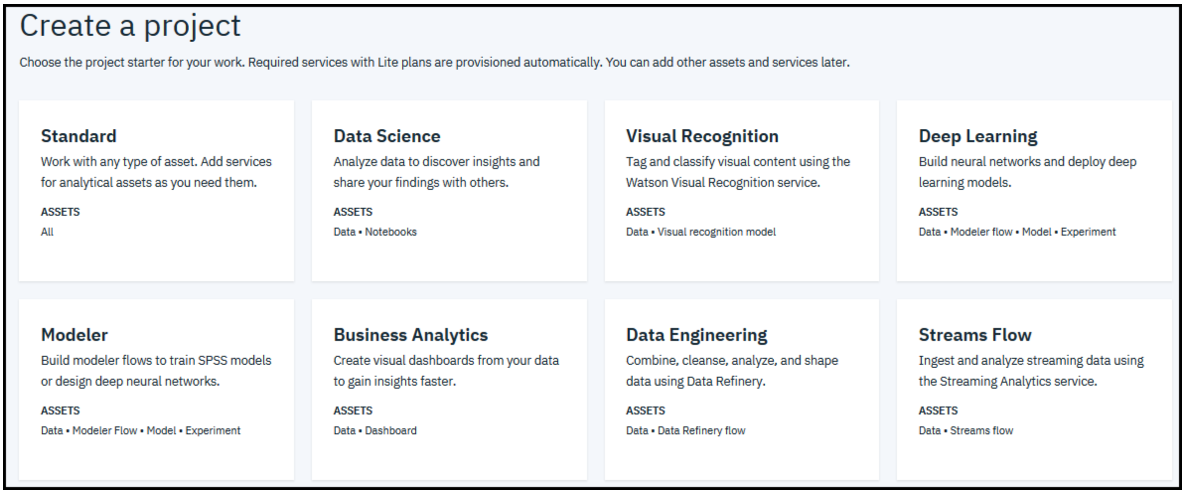
The runtime environment of Watson Studio is Apache Spark.

For more information, see https://dataplatform.cloud.ibm.com/docs/content/getting-started/overview-ws.html?context=analytics.

IBM Training

Creating a project

When you create a project in Watson Studio, you choose from these project starters.



Introduction to IBM Watson Studio

Figure 3-14. Creating a project

When you create a project in Watson Studio, you choose from these project starters:

Standard: Add collaborators and data assets. Add services as you need them.

Business Analytics: Visualize and share data in dashboards without coding.

Data Engineering: Cleanse and shape data with a Data Refinery flow.

Data Science: Analyze data with Jupyter notebooks or RStudio.

Deep Learning: Develop neural networks and test them in deep learning experiments.

Modeler: Build, train, test, and deploy machine learning models.

Streams Flow: Ingest streaming data with a streams flow.

Visual Recognition: Analyze and classify images in models.

All project starters include data assets and the IBM Cloud Object Storage service to store project files. After you create a project, you can add other analytical assets and associate other services.

IBM Training

Watson Studio and Cloud Object Storage

- Projects require an object storage to store non-database data sources.
- An IBM Cloud Object Storage instance can be created when the project is created.
  - A project can be associated with an existing Cloud Object Storage instance.
- Information stored in IBM Cloud Object Storage is encrypted and resilient.
- Each project has its own dedicated bucket.
  - Buckets can be managed from the Watson Studio project interface.
- Cloud Object Storage supports two APIs:
  - Swift API: Available through Watson Studio
  - Amazon Simple Storage Service (S3) API

Introduction to IBM Watson Studio

Figure 3-15. Watson Studio and Cloud Object Storage

Object storage provides the space where unstructured data for your project is stored.

An IBM Cloud Object Storage instance can be created at project creation time or a new project can be associated with an existing Cloud Object Storage instance.

Object Storage supports two APIs:

- The Swift API, which is available through Watson Studio.
- The S3 API, where you provide external credentials.

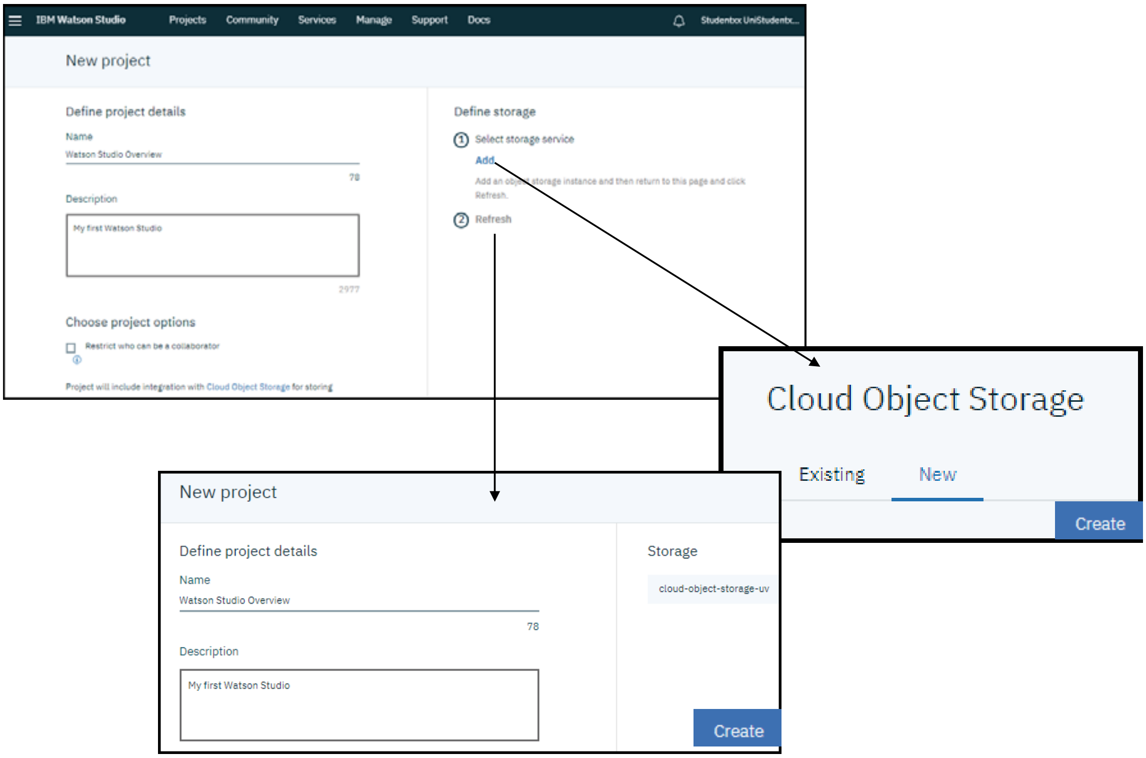
Information stored with IBM Cloud Object Storage is encrypted and resilient.

Cloud Object Storage uses buckets to organize the data. Each project has its own dedicated bucket.

The Cloud Object Storage can be managed from the Watson Studio project interface.

IBM Training

Creating a project and defining its storage



Introduction to IBM Watson Studio

Figure 3-16. Creating a project and defining its storage

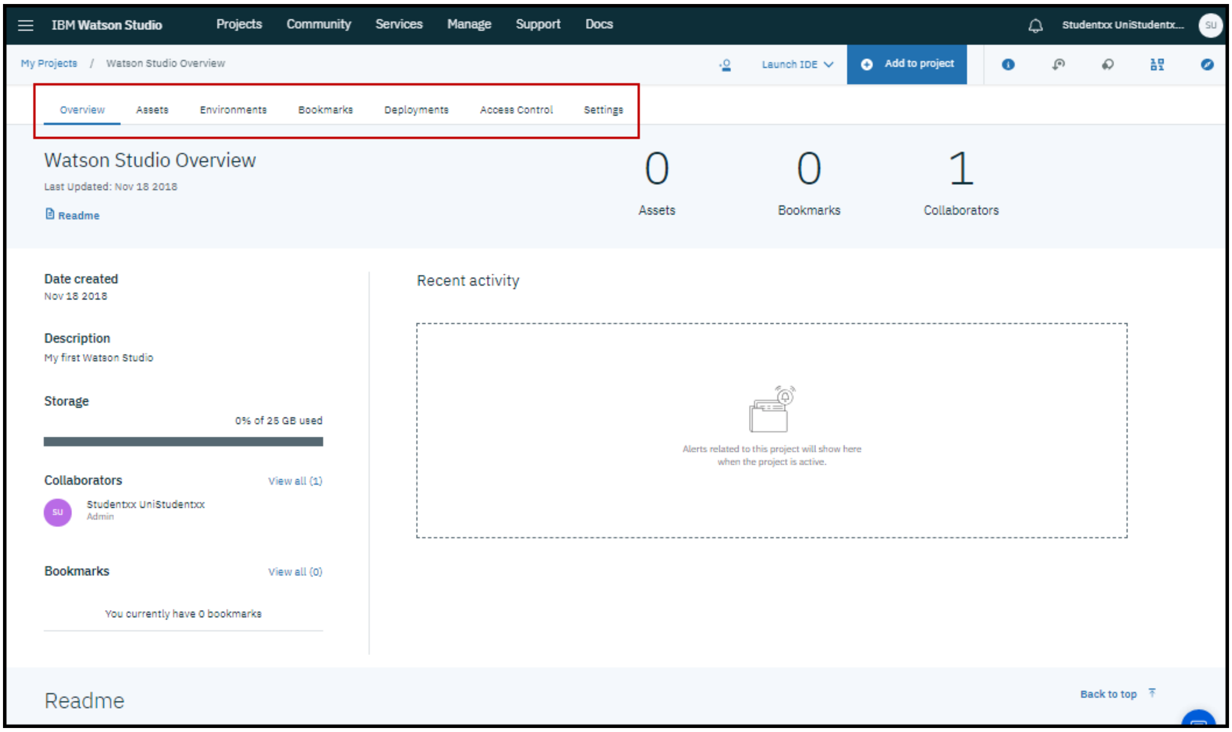
Projects include integration with IBM Cloud Object Storage for storing project assets. When a new project is created, a new Cloud Object Storage service is created to provide an unstructured cloud data store. It is also possible to associate a project with an existing Cloud Object Storage.

To create a new project:

1. Click **+ New**.
2. Enter a name for the project.
3. In the Define storage pane, click **Add**.
4. Click **Refresh**.
5. The Cloud Object Storage service page with the New tab selected is displayed. Click **Create**.
6. Click **Refresh** in the project page.
7. The newly created Cloud Object Storage instance is associated with the project.
8. Click **Create** to create the project.

IBM Training

Watson Studio project tabs



Introduction to IBM Watson Studio © Copyright IBM Corporation 2019

Figure 3-17. Watson Studio project tabs

The Watson Studio project page includes tabs with specific information about the project. The following tabs are included:

Overview. Provides basic information about the project such as:

- Description
- Storage usage
- Collaborators
- Bookmarks
- Recent activity

Assets. Lists all the different types of assets that you can add to you project such as:

- Data assets
- Models
- Notebooks
- Dashboards

- Experiments
- Modeler flows

The project starter you choose affects the type of analytic assets you see on the **Assets** page in the project. All project starters include *data assets* and the IBM Cloud Object Storage service to store project files. After you create a project, you can add other analytical assets and associate other services.

Environments. In this tab you can define the hardware size and software configuration for the runtime associated with Watson Studio tools. An environment definition defines the hardware and software configuration that you can use to run tools like notebooks, model builder, or the flow editor in Watson Studio. With environments, you have dedicated resources and flexible options for the compute resources used by the tools.

Bookmarks. Lists the community content that you have bookmarked; you can easily access that content from this project.

Deployments. Contains any models that you created and deployed, for example machine learning models.

Access Control. Here you can add an manage additional collaborators for your project.

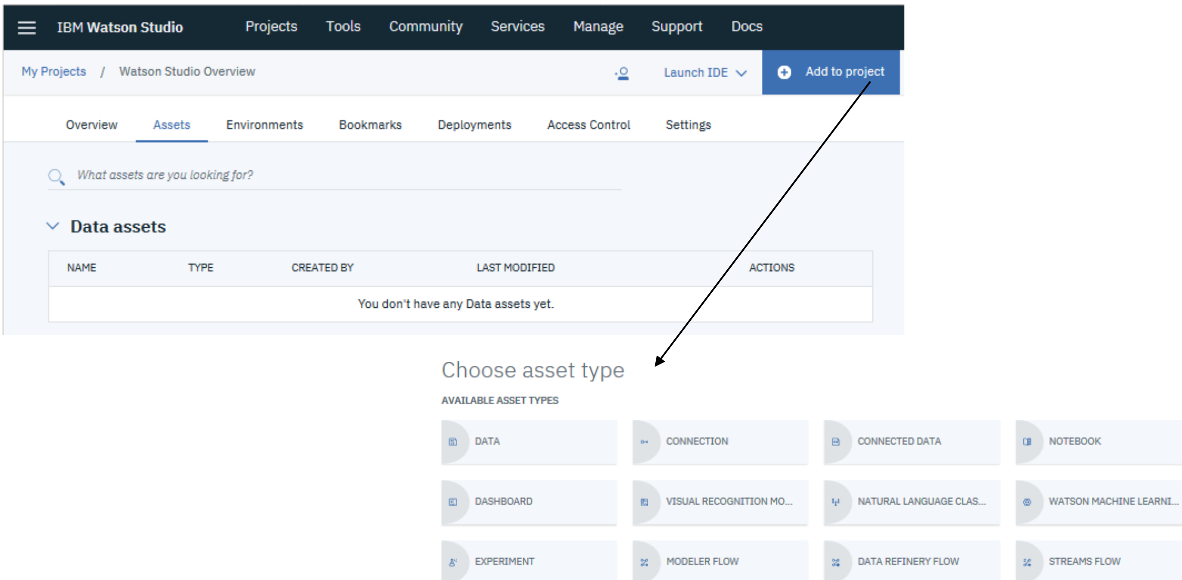
Settings. Project settings is the area to define key aspects of your project like its name and description. Use Settings to add tools and key services, check storage, and define access tokens.

For more information, see
https://dataplatform.cloud.ibm.com/docs/content/manage-data/project-tools.html?context=analytics
.

IBM Training

Assets

- An *asset* is an artifact in a project that contains information, or metadata, about data or data analysis.



Introduction to IBM Watson Studio

Figure 3-18. Assets

An *asset* is an artifact in a project that contains information, or metadata, about data or data analysis.

Data assets are the types of assets that point to data, for example, a file or a data set that is accessed through a connection to an external data source.

Analytical assets are the types of assets that run code to analyze data.

An asset might have associated files or require specific IBM Cloud services. For example, many types of analytical assets require a service.

The information that you can view about an asset and the actions that you can perform on an asset vary by asset type. A few actions apply to all asset types, for example, you can create, edit the properties of, and remove or delete all types of assets. You can edit the contents of most types of assets in projects. For data assets that contain relational data, you can shape and cleanse the data with the Data Refinery tool. For analytical assets, you can edit the contents in the appropriate editor tool. In catalogs, you can edit only the metadata for the asset.

A new project has no assets. By default, the Data assets type is added with all project starters. To add other asset types to your project click **Add to project**.

For more information, see *About assets* at
https://dataplatform.cloud.ibm.com/docs/content/getting-started/assets.html?audience=wdp&context=analytics

IBM Training

Communities

- Articles: Curated articles of interests to data scientists
- Data sets: Multiple open source data sets, including local, state, and government data
- Notebooks and tutorials:
  - Example notebooks
  - Notebook tutorials about how to perform specific use cases and access different data sources

Introduction to IBM Watson Studio

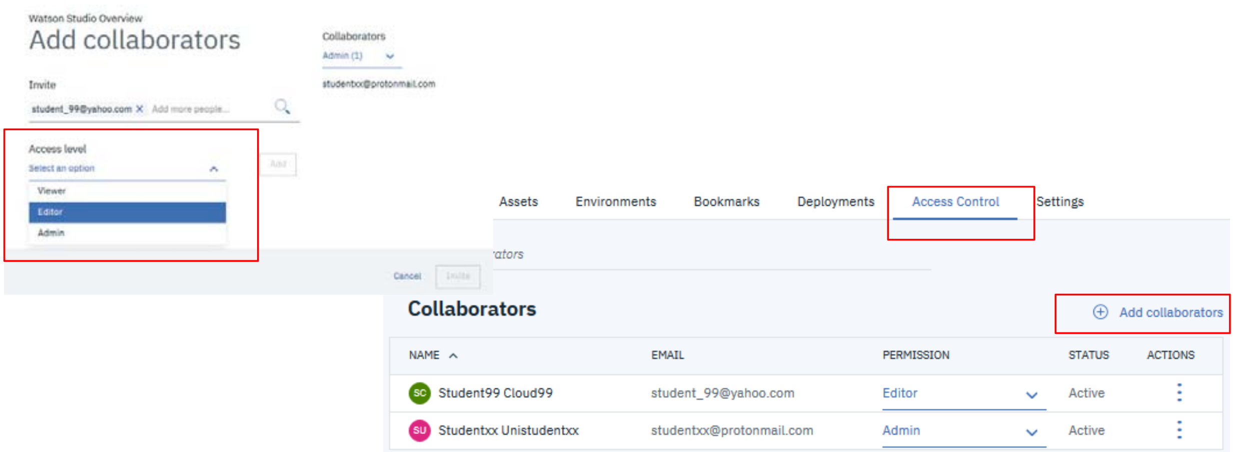
Figure 3-19. Communities

Watson Studio communities are a place where you can find articles of interests for data scientist. These are external articles that you can peruse. Within the Watson Studio communities, you can also find and use open data sets that are ready to use, including local, state, and government data. You download the data set and load it into your project. If you are working within a notebook, you can add the data set to the project. Communities are a great place to get started if you are exploring the data science space. There are sample notebooks and tutorials to get you started or to learn about new libraries and use cases.

IBM Training

Collaborating in a project

- Project collaborators share knowledge and resources.
- Admin role is required to manage collaborators (add, remove, and edit).
- Collaborators are added at the project level.
- Only collaborators in a project can access the project’s data and notebooks assets.



Introduction to IBM Watson Studio

Figure 3-20. Collaborating in a project

After you create a project, add collaborators to share knowledge and resources. Project collaborators share knowledge and resources and help one another complete jobs. You must have the Admin role in the project to manage collaborators.

Only the collaborators in your project can access your data, notebooks and other assets. Collaborators can be removed from a project or have their permissions changes.

To add collaborators to your project:

1. From your project, click the **Access Control** tab, and then click **Add collaborators**.
2. Add the collaborators who you want to have the same access level:
   - Type email addresses into the **Invite** field.
   - Copy multiple email addresses, separated by commas, and paste them into the **Invite** field.
3. Choose the access level for the collaborators and click **Add**:
   - **Viewer**: View the project.
   - **Editor**: Control project assets.
   - **Admin**: Control project assets, collaborators, and settings.
4. Add more collaborators with the same or different access levels.

5. Click **Invite**.

If the invited users have existing IBM Cloud accounts with Watson Studio activated, they are added to your project immediately.

If an invited user does not have a IBM Cloud account, the user receives an email invitation to create a IBM Cloud account and activate Watson Studio. When the user activates Watson Studio, the user can see your project and the user's status on your collaborators list changes from **Invited** to **Active**. If necessary, you can resend or cancel an invitation.

For more information, see *Project collaborators* at https://dataplatform.cloud.ibm.com/docs/content/getting-started/collaborate.html?context=analytics

IBM Training

Watson Studio and Spark

- Apache Spark is an open-source distributed general-purpose cluster-computing framework:
  - Provides a core data processing engine.
  - Libraries for SQL, machine learning, graph computation, and stream processing run on Spark and can be used together in an application.
- You need a Spark service or Spark environment if:
  - Your notebook includes Spark APIs.
  - You create machine learning models or model flows with Spark runtimes.
- Spark options in Watson Studio:
  - Spark environments offered under Watson Studio.
  - Spark services offered through IBM Cloud.

Introduction to IBM Watson Studio

Figure 3-21. Watson Studio and Spark

Apache Spark is an open source distributed cluster computing framework optimized for extremely fast and large scale data processing. Spark provides a core data processing engine. Several libraries for SQL, machine learning, graph computation, and stream processing run on Spark and can be used together in an application. To learn more about Apache Spark, see documentation and examples at spark.apache.org.

If your notebook includes Spark APIs, or you create machine learning models or model flows with Spark runtimes, you need to associate the tool with a Spark service or environment. With Spark environments, you can configure the size of the Spark driver and the size and number of the executors. You can use Spark environments for notebooks, the model builder and Spark modeler flows.

Spark options

In Watson Studio, you can use:

- Spark environments offered under Watson Studio

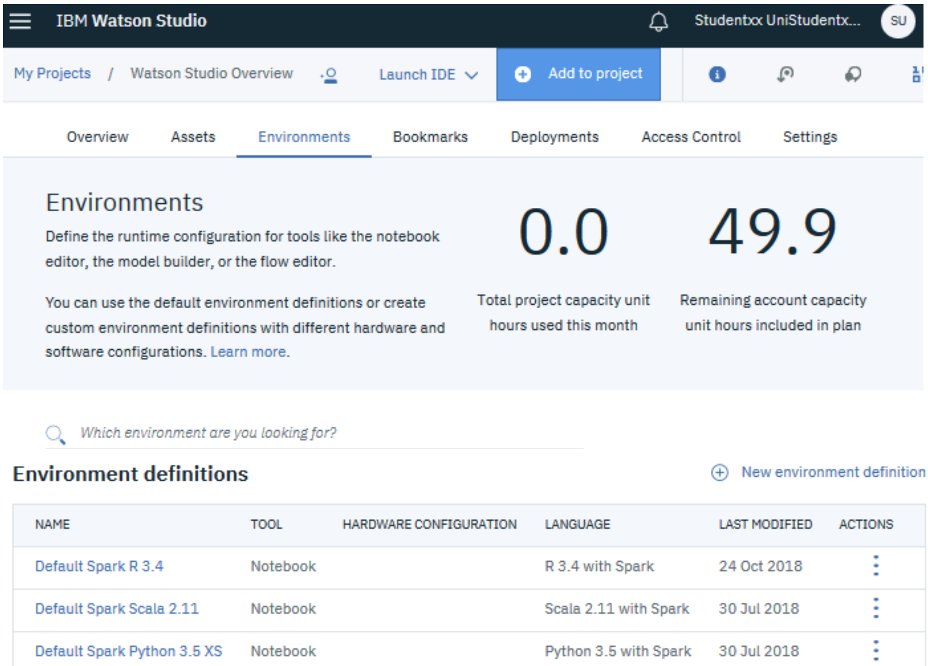
  All Watson Studio users can create Spark environments with varying hardware and software configurations. Spark environments offer Spark kernels as a service (SparkR, PySpark and Scala). Each kernel gets a dedicated Spark cluster and Spark executors. Spark environments are offered under Watson Studio and, like default environments, consume capacity unit hours (CUHs) that are tracked.

- Spark services offered through IBM Cloud

  You can add the Spark service to your project. Spark is a fast, in-memory engine for very large data sets. In Watson Studio, you can use the Apache Spark service for your Python, Scala, or R notebooks.

IBM Training

Spark default environments



Introduction to IBM Watson Studio

Figure 3-22. Spark default environments

Spark environments offered under Watson Studio

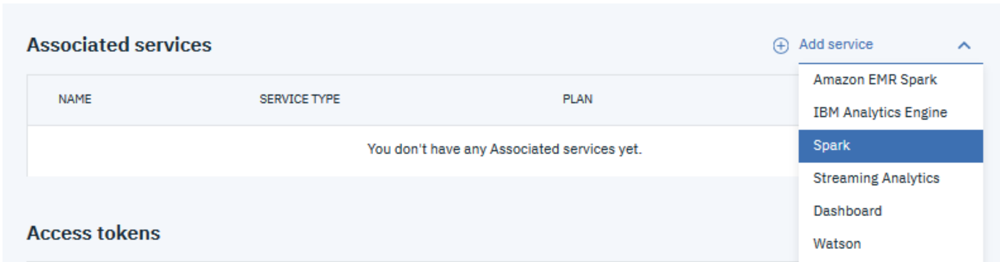
All Watson Studio users can create Spark environments with varying hardware and software configurations. Spark environments offer Spark kernels as a service (SparkR, PySpark and Scala). Each kernel gets a dedicated Spark cluster and Spark executors. Spark environments are offered under Watson Studio and, like default environments, consume capacity unit hours (CUHs) that are tracked.

The slide shows the default Spark environments definition under the project's Environment tab.

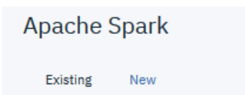
IBM Training

Associating a Spark service

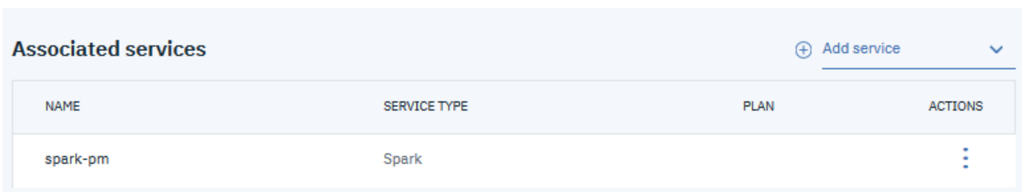
- Click Studio project > Settings > Associated services > Add service.



- Create Spark service instance



- Spark service added to Watson Studio project



Introduction to IBM Watson Studio

Figure 3-23. Associating a Spark service

Spark services offered through IBM Cloud

You can add the Spark service to your project:

1. Select the **Settings** tab.
2. Scroll-down to **Associated services**.
3. Click **Add service**.
4. Select **Spark**.
5. Create the Spark service instance.
6. The service is listed in the Associated services section.

The slide shows the main screen captures in the steps listed above.

3.3. Jupyter notebook

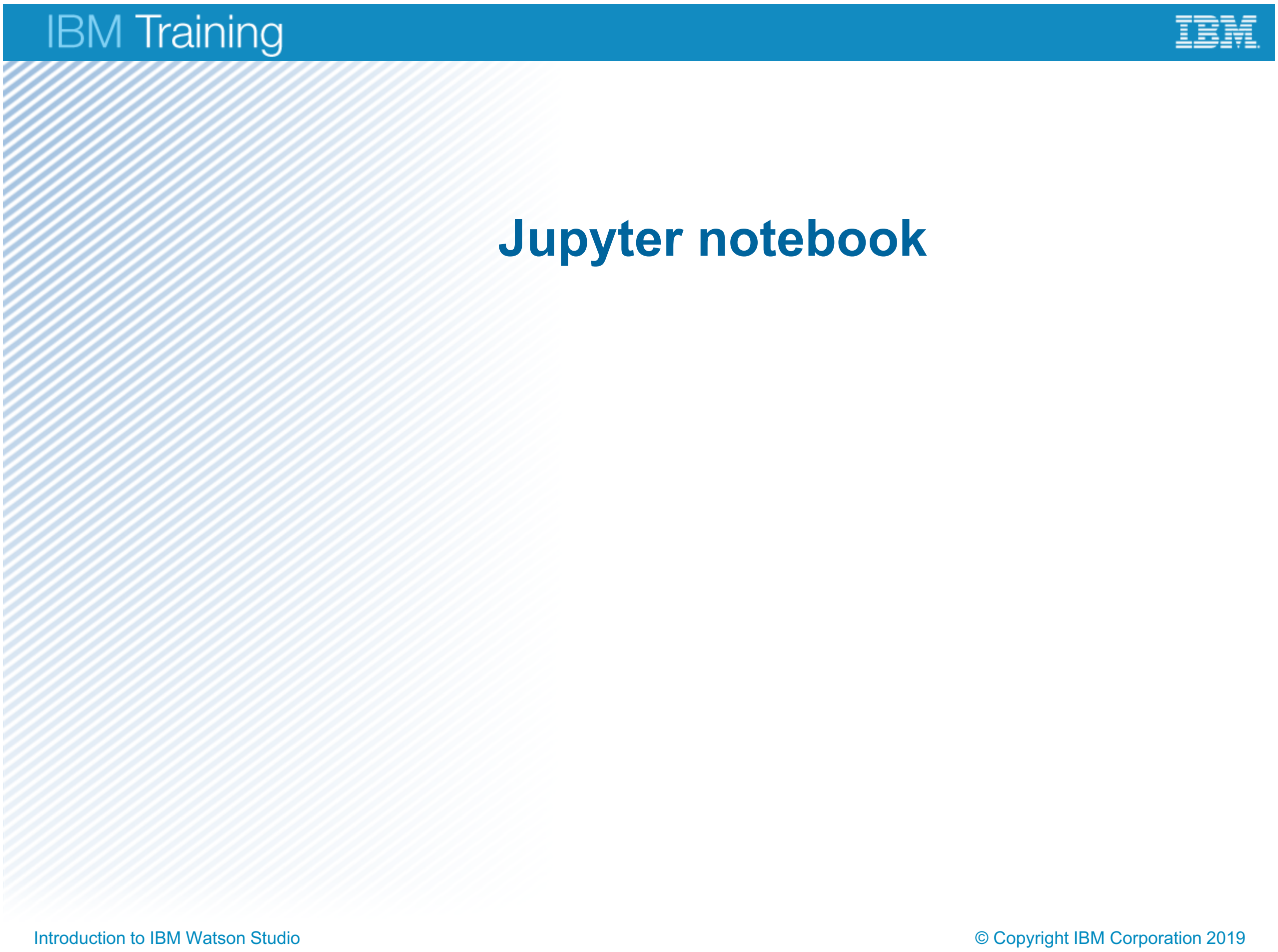


Figure 3-24. Jupyter notebook

IBM Training

Topics

- Watson Studio overview
- Watson Studio components
- Jupyter notebook

Introduction to IBM Watson Studio

Figure 3-25. Topics

IBM Training

Jupyter notebooks overview

- A web-based environment where you can run small pieces of code that process your data and immediately visualize the results.
- Include all the building blocks to work with data:
  - The data
  - The code computations that process the data
  - Visualizations of the results
  - Text and rich media to enhance understanding
- Other options:
  - Zeppelin notebooks (Watson Studio Local)
  - Rstudio for statistical computing

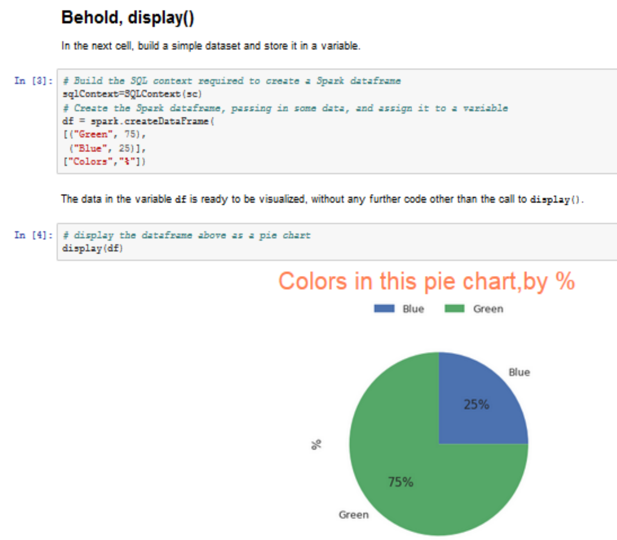


Figure 3-26. Jupyter notebooks overview

A Jupyter notebook is a web-based environment for interactive computing. You can run small pieces of code that process your data, and you can immediately view the results of your computation. Notebooks include all of the building blocks you need to work with data:

- The data
- The code computations that process the data
- Visualizations of the results
- Text and rich media to enhance understanding

From markdown and live code to visualizations, notebooks include all of the building blocks you need to work with data and enhance your understanding.

You can uses notebooks to add formatted text around executable code to describe what is being done and show the result, including graphics, concurrently.

There are a few popular notebooks, including the Jupyter Notebook, which is popular with Python programmers because Python is popular with data scientists.

Other notebooks can be added over time. For example, Watson Studio Local added another popular notebook that is called Zeppelin.

Another popular environment for data science, statistical analysis, and graphic representation is provided by the R language. Watson Studio includes RStudio for people that prefer the R development environment.

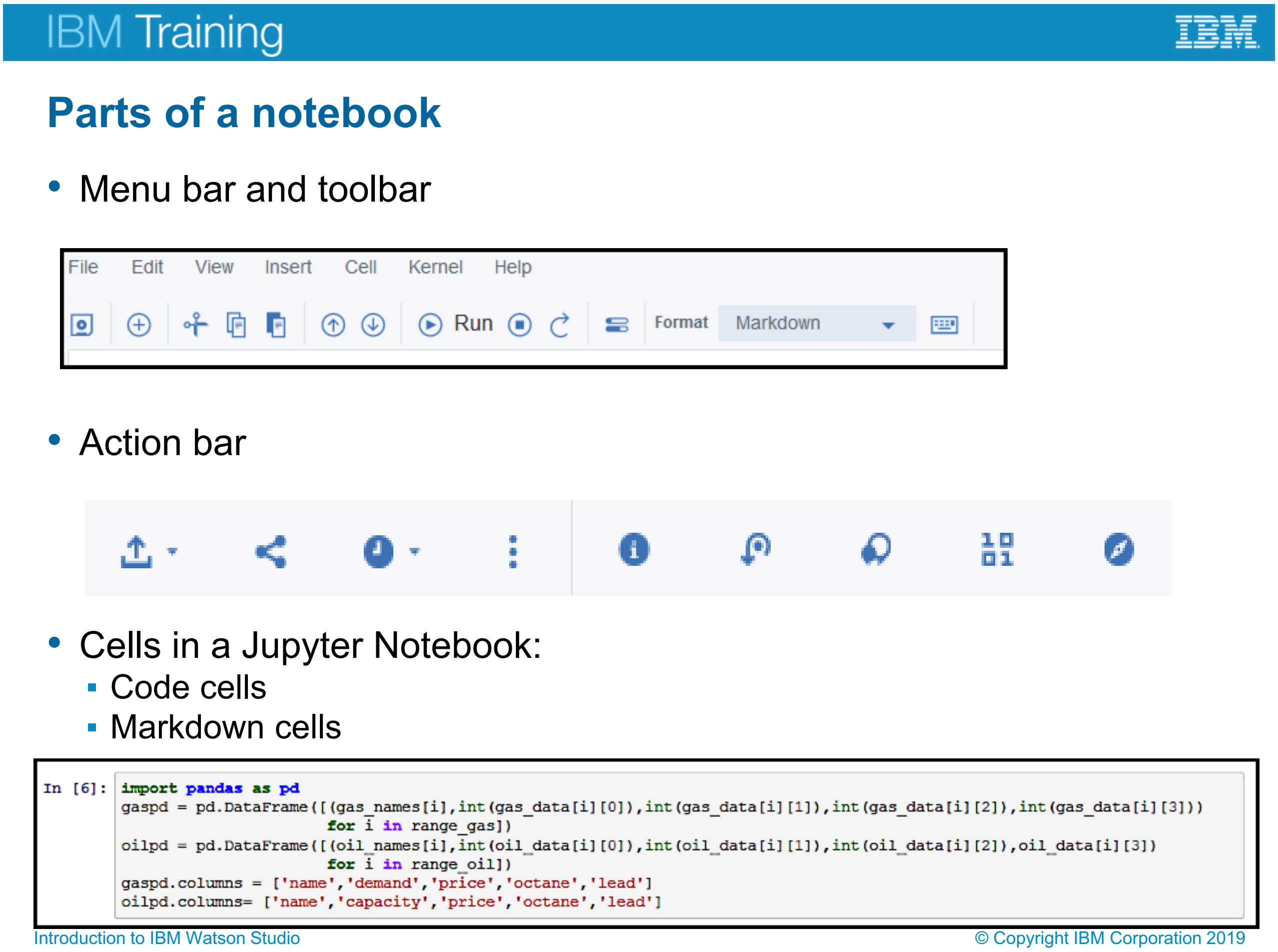


Figure 3-27. Parts of a notebook

The **menu bar** controls most of the operations and settings for the notebook. Simple and common functions are supported through the menu bar.

From the **action bar** you can select features that enhance notebook collaboration. From the action bar, you can:

- Publish your notebook as a gist or to GitHub.
- Create a permanent URL so that anyone with the link can view your notebook.
- Schedule to run your notebook at a given time.
- Add a project token so that code can access the project resources.
- View your notebook information. Change the name of your notebook by editing it in the **Name** field. If your notebook runs in an environment, you can view the environment details and runtime status.
- Save versions of your notebook.
- View and add data sources.
- Post comments to project collaborators.

- Integrate data into your notebook.
- Find resources in the community, for example, useful data sets.

Cells in a Jupyter notebook

A Jupyter notebook consists of a sequence of cells. The flow of a notebook is sequential. You enter code into an input cell, and when you run the cell, the notebook executes the code and prints the output of the computation to an output cell.

You can change the code in an input cell and re-run the cell as often as you like. In this way, the notebook follows a read-evaluate-print loop paradigm. You can choose to use tags to describe cells in a notebook.

The behavior of a cell is determined by a cell's type. The types of cells include:

- Code cells: They are the cells where you edit and run the code. The output of the cell appears beneath the cell. Tags can be used to describe the cell for code readability and maintainability. Cells can be rerun as often as you like.
- Markdown cells are used to document and comment the process. You can use this to structure your notebook by using the markdown language. Images and file attachments can be added to the notebook by using the markdown cells.

IBM Training

Creating notebooks

- Must have a project and a Spark instance.
  - Associate with existing Spark instance or create a new instance.
- Options to create a Jupyter notebook:
  - Blank, From a file, From a URL
  - Copy sample notebook from the community
- Specify a name for the notebook.
- Specify the language:
  - Python 2
  - R
  - Scala
  - Python 3.5 (experimental)
- Specify the Spark version:
  - 2.1, 2.0, 1.6

Introduction to IBM Watson Studio

Figure 3-28. Creating notebooks

Before you can create a notebook, you must have a project and associate a Spark instance with it. If you already have an existing Spark instance, you may use it. Otherwise, you must create a Spark instance.

When the project is set up, you can create the notebook by using one of these ways:

- Blank
- From a file
- From a URL

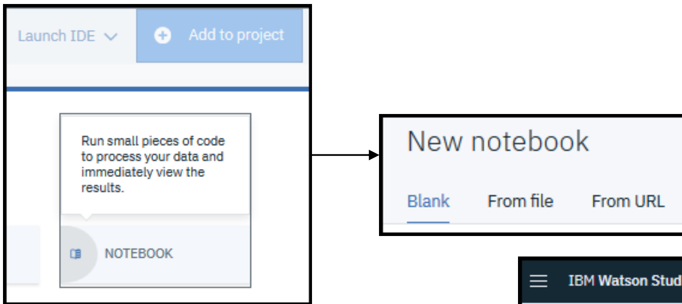
You must specify a name for your notebook. If you have a blank notebook, you must specify the language and the Spark version that you want to use with the notebook. You can create a notebook by importing a Python, Scala, or R notebook file (.ipynb) from your local workstation. You can also create a notebook from a URL by providing the URL.

Alternatively, you can copy a sample notebook from the Community to help you get started.

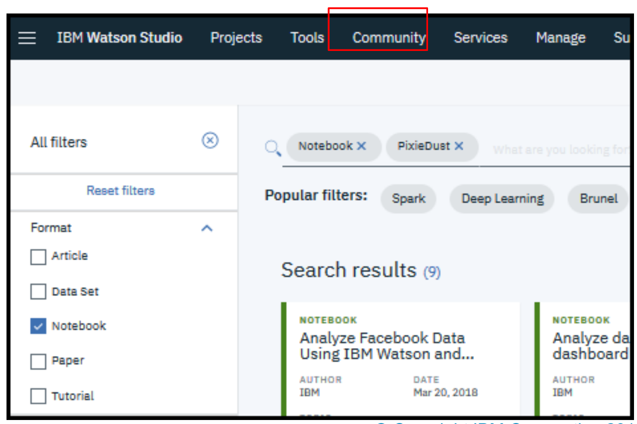
IBM Training

Creating notebooks (cont.)

- From the project, click Add to project > Notebook > New notebook.
  - Select an option: Blank, From file, or From URL.



- Copy sample from Community.



Introduction to IBM Watson Studio

Figure 3-29. Creating notebooks (cont.)

Create a notebook file

To create a notebook file in the project:

1. From your project, click **Add to Project > Notebook**.
2. On the **New Notebook** page, specify the method to use to create your notebook. You can create a blank notebook, upload a notebook file from your file system, or upload a notebook file from a URL. The notebook you create or select must be a *.ipynb* file.
3. Specify the language: Python, R, or Scala.
4. Specify the runtime environment. After you have created an environment definition on the **Environments** page of your project, it will appear in the list of runtimes you can select from at the time you create the notebook.
5. Click **Create Notebook**. The notebook opens in edit mode.

Copy a sample notebook from the Community

Sample notebooks from the Community are based on real-world scenarios and contain many useful examples of computations and visualizations that you can adapt to your analysis needs.

To copy a sample notebook:

1. Click **Community** from the Watson Studio toolbar, then filter for **Notebooks** to show only notebook cards.
2. Find the card for the sample notebook you want and click the card. You can view the notebook contents to browse the steps and the code that it contains.
3. To work with a copy of the sample notebook, click the **Copy** icon. The **New Notebook** screen opens.
4. Choose the project for the notebook.
5. Specify the runtime environment. After you have created an environment definition on the **Environments** page of your project, it will appear in the list of runtimes you can select from at the time you create the notebook.
6. Click **Create Notebook**. The notebook opens in edit mode.

IBM Training

Coding and running notebooks

- Import libraries:
  - Preinstalled from Python or R.
  - Install custom or third-party libraries for any language. For Scala, no libraries are preinstalled in the Spark service.
- Load and access data:
  - Add a file from the local system to your object store.
  - Use a free data set from the Watson Studio home page.
  - Load data from a data source connection.
- Prepare and analyze the data.
- Collaborate with other project members or share outside Watson Studio.
- Visualize the results.

Introduction to IBM Watson Studio

Figure 3-30. Coding and running notebooks

To code and run notebooks, complete the following steps:

1. Import the libraries that you need for the notebook. You can import preinstalled libraries if you are using Python or R. If you are using Scala, you must install the libraries manually, and they are available for the duration of that notebook.
2. Load and access the data. To load data into your own notebooks, you can choose one of these options:
   - Add a file from your local system to your object storage.
   - Use a free data set from the community.
   - Load data from a data source connection, such as an external data source or another IBM Cloud service.
3. Prepare and analyze data

   With the libraries and data loaded, you can start data analysis by preparing the data, analyzing the data, and making predictions. You can write your own code with Python functions to work with data and IBM Cloud Object Storage in notebooks.

4. Collaborate with other project members and share your notebook with people outside Watson Studio.

   You can create a URL to share the last saved version of a notebook on social media or with people outside of Watson Studio. The URL shows a read-only view of the notebook. Anyone who has the URL can view or download the notebook. Whenever you save a new version of the notebook with the File > Save Version command, the URL shows the new version. The shared notebook shows the author of the shared version and when the notebook version was last updated.

5. Visualize the data. Use visualizations in your notebooks to present data visually to help identify patterns, gain insights, and make decisions. Many of your favorite open source visualization libraries, such as matplotlib, are pre-installed on Watson Studio. All you have to do is import them. In the upcoming exercise, you use visualization libraries such as PixieDust to visualize your data.

IBM Training

Unit summary

- Describe Watson Studio.
- Identify industry use cases.
- List Watson Studio offerings.
- Create Watson Studio projects.
- Describe Watson Studio and Spark.
- Describe Watson Studio and Object Storage.
- Explain Watson Studio high availability considerations.
- Prepare and analyze data.
- Use Jupyter notebooks.

Introduction to IBM Watson Studio

Figure 3-31. Unit summary

IBM Training

Review questions

1. True or false: Watson Studio is designed only for the data scientist persona.

2. True or false: Object storage provides the space where unstructured data for your project is stored.

3. True or false: Collaborators can be given different access levels.

4. True or false: Collaboration within Watson Studio is an optional add-on component that must be purchased.

5. True or false: Watson Studio communities provide access to articles, tutorials, and even data sets that you can use.

Introduction to IBM Watson Studio

Figure 3-32. Review questions

IBM Training

Review answers

1. True or false: Watson Studio is designed only for the data scientist persona.

2. True or false: Object storage provides the space where unstructured data for your project is stored.

3. True or false: Collaborators can be given different access levels.

4. True or false: Collaboration within Watson Studio is an optional add-on component that must be purchased.

5. True or false: Watson Studio communities provide access to articles, tutorials, and even data sets that you can use.

Introduction to IBM Watson Studio

Figure 3-33. Review answers

1. True or false: Watson Studio is designed only for the data scientist persona.
   False: “The data engineer, the data scientist, the business analyst, and the app developer. Each persona has primary tools that help them do their job. For example, the data scientist's main tool is Watson Studio but other personas can use it as it relates to their jobs. . Also, there might be a team of different personas. Whatever the case is, you must decide what tool is correct for the job, regardless of the personas. The definitions of personas can vary between different companies and evolve over time.”
2. True or false: Object storage provides the space where unstructured data for your project is stored.

   True: “An IBM Cloud Object Storage instance can be created at project creation time or a new project can be associated with an existing Cloud Object Storage instance.”
3. **True** or false: Collaborators can be given certain access levels.

   True: “Add collaborators to your project by using their email addresses. If they have an existing account on IBM Cloud, they are added immediately. Otherwise, they receive an invite to create a Watson Studio account. Choose the permissions for each collaborator. The Admin role can control project assets, collaborators, and settings. The Editor role can control project assets. The Viewer role can view the project. Collaborators can be removed from a project or have their permissions updated.”

4. True or false: Collaboration within Watson Studio is an optional add-on component that must be purchased.

   False: “Collaboration is a standard feature in Watson Studio.”

5. Watson Studio communities provide access to articles, tutorials, and even data sets that you can use.

   True: “Watson Studio communities are a place where you can find articles of interests for data scientist. These are external articles that you can peruse. Within the Watson Studio communities, you can also find and use open data sets that are ready to use, including local, state, and government data. You download the data set and load it into your project. If you are working within a notebook, you can add the data set to the project. Communities are a great place to get started if you are exploring the data science space. There are sample notebooks and tutorials to get you started or to learn about new libraries and use cases.”

IBM Training

Exercise: Getting started with Watson Studio

Introduction to IBM Watson Studio

Figure 3-34. Exercise: Getting started with Watson Studio

IBM Training

Exercise objectives

- This exercise introduces you to the basic tasks that you will perform when using Watson Studio.
- After completing this exercise, you should be able to:
  - Create a Watson Studio project.
  - Manage the project.
  - Assign collaborators.
  - Load a data set into the project’s object store.
  - Manage Cloud Object Storage.
  - Analyze data by using Watson Studio.
  - Use PixieDust for data visualization.

Introduction to IBM Watson Studio

Figure 3-35. Exercise objectives

Unit 4. Introduction to IBM Watson Machine Learning

Estimated time

00:30

Overview

This unit describes the CRoss Industry Standard Process for Data Mining (CRISP-DM) and explains the process of preparing data for a machine learning algorithm. This unit provides an overview of the IBM Watson Machine Learning service available on IBM Cloud.

IBM Training

Unit objectives

- Describe the Cross-Industry Standard Process for Data Mining (CRISP-DM) process model.
- Explain where machine learning fits in the CRISP-DM process.
- Describe data preparation before feeding into machine learning algorithms.
- Describe Watson Machine Learning features and capabilities.

Introduction to IBM Watson Machine Learning

Figure 4-1. Unit objectives

4.1. Cross-Industry Standard Process for Data Mining

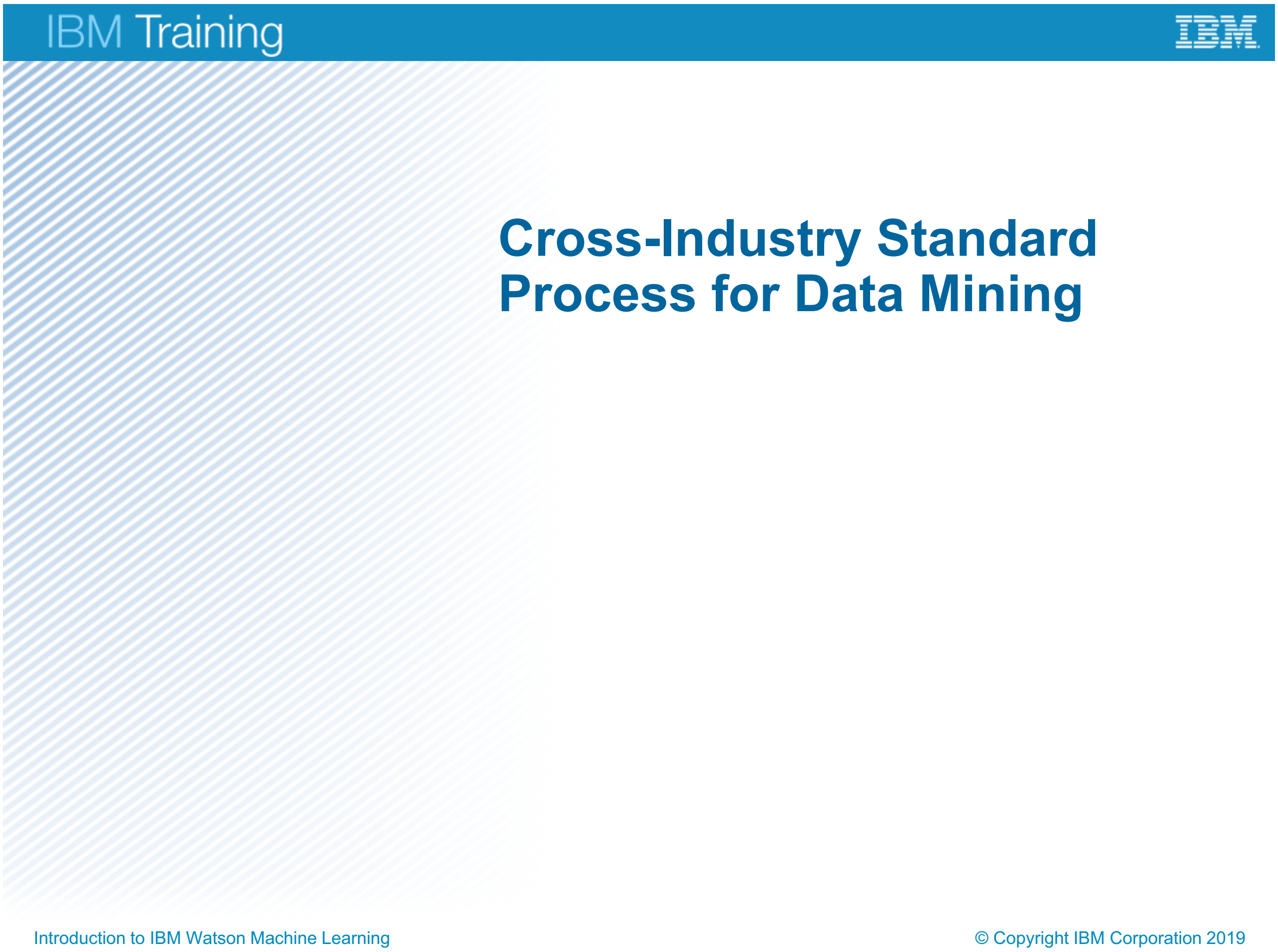


Figure 4-2. Cross-Industry Standard Process for Data Mining

IBM Training

Topics

- Cross-Industry Standard Process for Data Mining
- Data preparation
- Watson Machine Learning

Introduction to IBM Watson Machine Learning

Figure 4-3. Topics

IBM Training

Cross-Industry Standard Process for Data Mining overview

- CRISP-DM stands for Cross-Industry Standard Process for Data Mining.
- It is an open standard guide that describes common approaches that are used by data mining experts.
- As a methodology, CRISP-DM includes descriptions of the typical phases of a project, including tasks details.
- As a process model, CRISP-DM provides an overview of the data mining lifecycle.

Introduction to IBM Watson Machine Learning

Figure 4-4. Cross-Industry Standard Process for Data Mining overview

CRISP-DM is an industry-proven way to guide your data mining efforts. It is the most widely used analytics model. It describes common approaches that are used by data mining experts.

As a methodology, it includes descriptions of the typical phases of a project, the tasks that are involved with each phase, and an explanation of the relationships between these tasks.

As a process model, CRISP-DM provides an overview of the data mining lifecycle. In a nutshell, it consolidates preferred practices.

Reference:
https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.crispdm.help/crisp_overview.htm

IBM Training

CRISP-DM lifecycle

Introduction to IBM Watson Machine Learning

Figure 4-5. CRISP-DM lifecycle

The lifecycle model consists of six phases with arrows indicating the most important and frequent dependencies between phases. The sequence of the phases is not strict. In fact, most projects move back and forth between phases as necessary.

It starts with business understanding, and then moves to data understanding, data preparation, modelling, evaluation, and deployment.

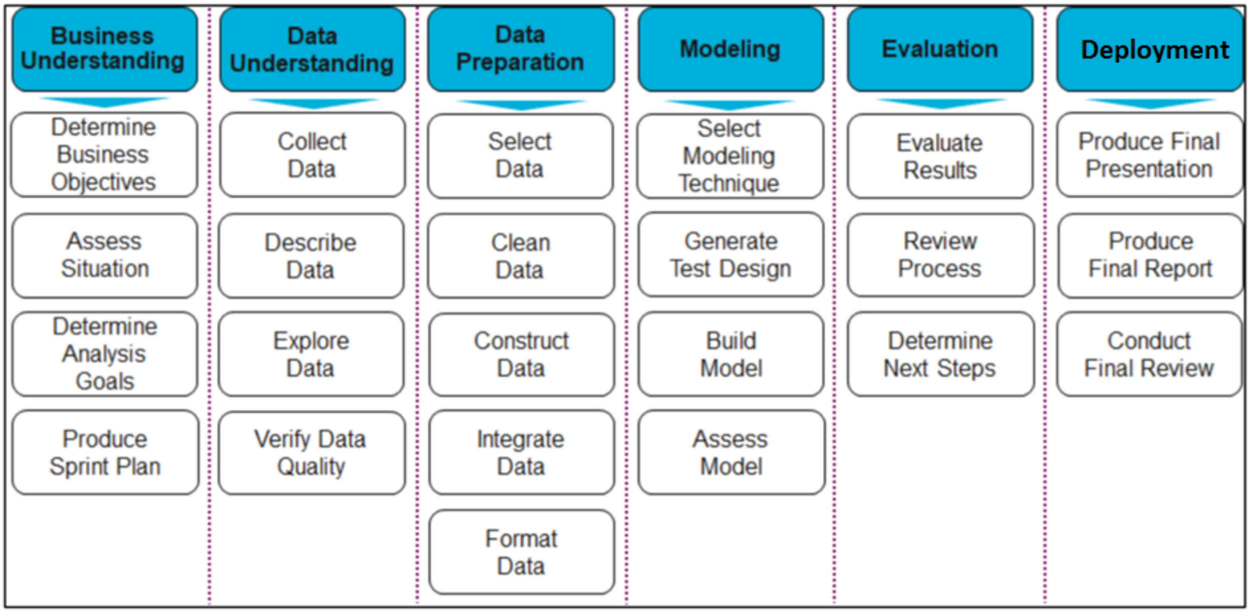
The CRISP-DM model is flexible and can be customized easily. For example, if your organization aims to detect money laundering, it is likely that you examine large amounts of data without a specific modelling goal. Instead of modelling, your work focuses on data exploration and visualization to uncover suspicious patterns in financial data.

In such a situation, the modelling, evaluation, and deployment phases might be less relevant than the data understanding and preparation phases. However, it is still important to consider some of the questions that are raised during these later phases for long-term planning and future data mining goals.

Reference:
https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.crispdm.help/crisp_overview.htm

IBM Training

Phases in CRISP-DM



Introduction to IBM Watson Machine Learning

Figure 4-6. Phases in CRISP-DM

This slide summarizes the tasks and outcomes of each phase.

4.2. Data preparation

IBM Training

Data preparation

Introduction to IBM Watson Machine Learning

Figure 4-7. Data preparation

IBM Training

Topics

- Cross-Industry Standard Process for Data Mining
- Data preparation
- Watson Machine Learning

Introduction to IBM Watson Machine Learning

Figure 4-8. Topics

IBM Training

Data preparation

The process of preparing data for a machine learning algorithm has the following phases:

- Data selection
- Data preprocessing
- Data transformation

Introduction to IBM Watson Machine Learning

Figure 4-9. Data preparation

Machine learning algorithms depend highly on the quality and quantity of data. It is important that you provide these algorithms the correct data. Data preparation is a large subject that can involve many iterations, exploration, and analysis. Becoming proficient at data preparation will make you a master at machine learning.

In this section, we cover basics tasks of the data preparation process: data selection, data preprocessing, and data transformation.

References:

https://www.simplilearn.com/data-preprocessing-tutorial

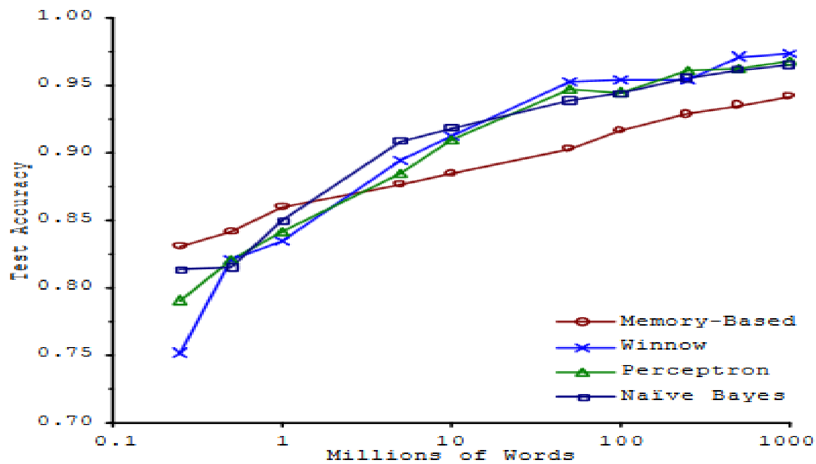
https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/

IBM Training

Data selection

Think about:

- What data is available, what data is missing, and what data can be removed?
- Is the selected sample an accurate representation of the entire population?
- Is more data better?



Banko and Brill [2001] http://www.aclweb.org/anthology/P01-1005

Introduction to IBM Watson Machine Learning

Figure 4-10. Data selection

This step is concerned with selecting a subset of all the available data with which you are working. Consider what data is available, what data is missing, and what data can be removed.

Make some assumptions about the data that you require and record those assumptions. Ask yourself:

- What data is available? For example, through which media, such as database tables or other systems? Is it structured or unstructured? Make sure that you are aware of everything that you can use.
- What data is not available and you want to get? For example, data that is not or cannot be recorded. Can you develop or simulate this data?
- What data can be removed (because it does not address the problem)? Document which data you excluded and why.

It is common to think that the more data we have the better result we get, but this is not necessarily true. According to the paper "Scaling to Very Very Large Corpora for Natural Language Disambiguation" by Microsoft Researchers Banko and Brill [2001] [1]. They proved that for a given problem, different algorithms perform virtually the same and the accuracy increases when adding more data (in this case, words). But, for large amounts of data, the improvements start to become negligible.

References:

[1]http://www.aclweb.org/anthology/P01-1005

https://www.dummies.com/programming/big-data/phase-3-of-the-crisp-dm-process-model-data-preparation/

https://www.simplilearn.com/data-preprocessing-tutorial

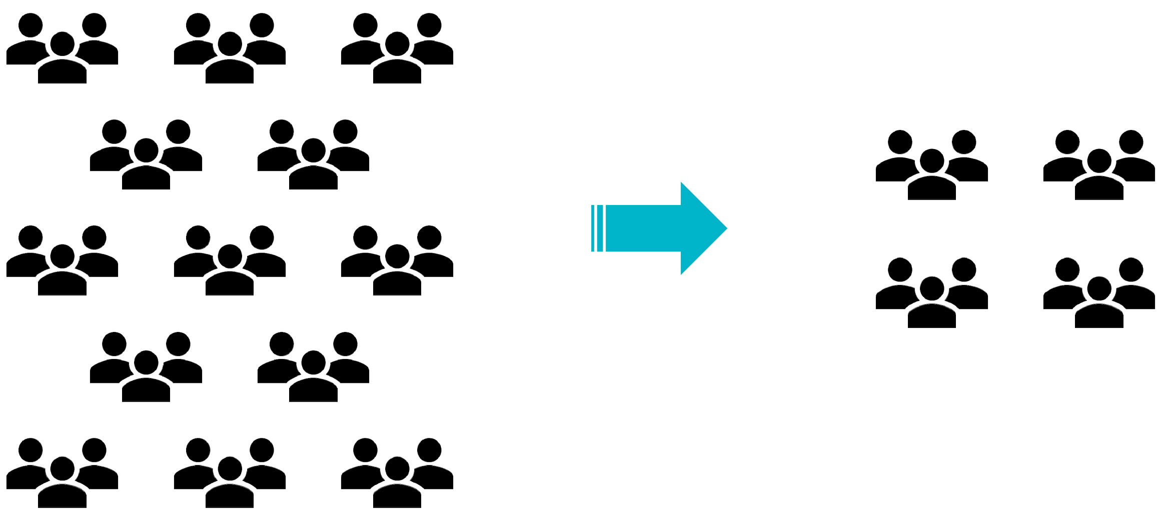
https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/

IBM Training

Data selection: Samples selection

Selecting the right size of the sample is a key step in data preparation. Samples that are too large or too small might give skewed results.

- Sampling Noise
- Sampling Bias



Introduction to IBM Watson Machine Learning

Figure 4-11. Data selection: Samples selection

Selecting the correct size of the sample is a key step in data preparation. Samples that are too large or too small might give skewed results.

Sampling Noise: Smaller samples cause sampling noise because they are trained on non-representative data.

Sampling Bias: A sample is biased if certain samples are underrepresented or overrepresented relative to others in the population. Larger samples work well if there is no sampling bias, that is, when the correct data is picked.

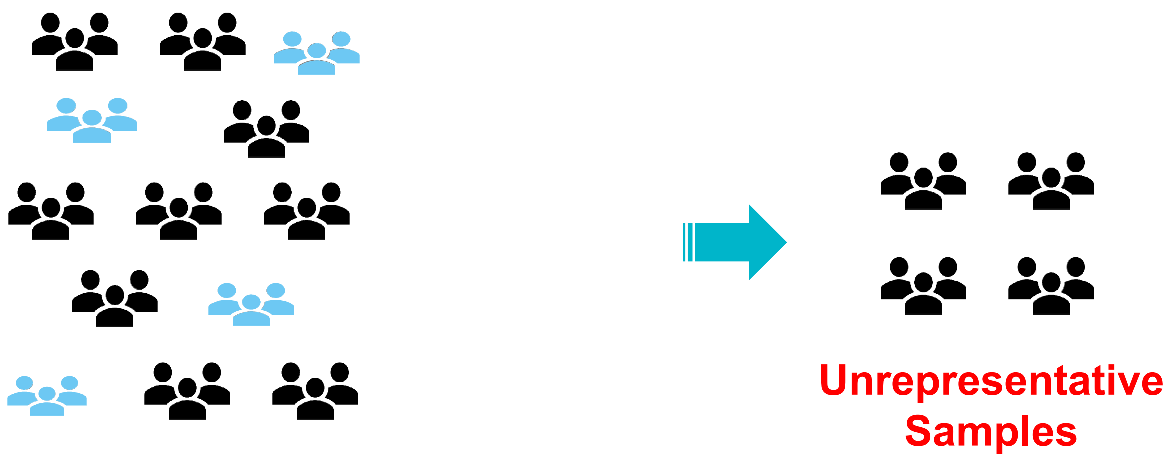
Reference:

https://www.simplilearn.com/data-preprocessing-tutorial

IBM Training

Data selection: Survey sampling

- Undercoverage
- Non-response bias
- Voluntary response bias



Introduction to IBM Watson Machine Learning

Figure 4-12. Data selection: Survey sampling

A good sample is representative, meaning that each sample point represents the attributes of a known number of population elements. The bias that results from an unrepresentative sample is called selection bias. There are various types of bias in survey like Undercoverage, Non-response bias and voluntary response bias. To explain these types assume that you would like to conduct a survey about customers and their purchase preferences.

- **Undercoverage**: Undercoverage occurs when some members of the population are inadequately represented in the sample. For example, when a minority is underrepresented in a sample, this situation can affect the validity of the sample. In this example, assume that you underrepresented the customers who have a low income.
- **Non-response bias**: Sometimes, members that are chosen for the sample are unwilling or unable to participate in the survey. Non-response bias is the bias that results when respondents differ in meaningful ways from non-respondents. This bias can affect the validity of the sample. Non-response bias is a common problem with mail surveys because the response rate is often low, making mail surveys vulnerable to non-response bias.
- **Voluntary response bias**: This bias occurs when sample members are self-selected volunteers, as in voluntary samples. When this happens, the resulting sample tends to over represent individuals who have strong opinions.

References:

https://en.wikipedia.org/wiki/Survey_sampling

https://sites.google.com/site/apstatisticssurveys/bias-in-surveys/bias-due-to-unrepresentative-samples

IBM Training

Data preprocessing: Overview

- Data challenges:
  - Noise and outliers
  - Missing values
  - Inconsistent values
  - Duplicate data
- Data preprocessing is a data mining technique that involves transforming raw data into an understandable format.

Introduction to IBM Watson Machine Learning

Figure 4-13. Data preprocessing: Overview

Why we use data preprocessing

In actual scenarios, data is generally incomplete:

- Data might lack attribute values, lacking certain attributes of interest.
- Data is noisy that contains errors or outliers. For example, human height as a negative value.
- Data might be inconsistent, containing discrepancies in codes or names. For example, a record for two employees having the same ID.

We usually come across much raw data that is not fit to be readily processed by machine learning algorithms. We must preprocess the raw data before it is fed into various machine learning algorithms. Organize your selected data by formatting, cleaning, and sampling from the data. Poor data quality negatively affects many data processing efforts.

References:

https://hackernoon.com/what-steps-should-one-take-while-doing-data-preprocessing-502c993e1caa

https://medium.com/datadriveninvestor/machine-learning-ml-data-preprocessing-5b346766fc48

https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/

IBM Training

Data preprocessing: Steps overview

1. **Data cleaning:** Complete missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
2. **Data integration:** Using multiple databases, other data sources, or files.
3. **Data sampling:** Faster for exploring and prototyping.
4. **Data dimensionality reduction:** Reducing the volume but producing the same or similar analytical results.
5. **Data formatting:** The data that you selected might not be in a format that is suitable for you to use.

Introduction to IBM Watson Machine Learning

Figure 4-14. Data preprocessing: Steps overview

The following list summarizes the steps that are used in preprocessing data:

1. **Data cleaning:** Complete missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
2. **Data integration:** Using multiple databases, other data sources, or files.
3. **Data sampling:** You can take a smaller representative sample of the selected data that might be much faster for exploring and prototyping solutions before considering the whole data set.
4. **Data dimensionality reduction:** Reducing the dimensions of the data and producing the same or similar analytical results. This is a kind of data compression that uses less memory or disk space and speeds the learning algorithm. For example, assume that you have a feature for size in centimeters and another feature for size in inches. By removing one of those redundant features, you reduce the dimensions of your data from 3D to 2D.
5. **Data formatting:** The data that you selected might not be in a format that is suitable for you to use. For example: the data might be in a relational database and you want it in a comma-separated file.

References:

https://medium.com/datadriveninvestor/machine-learning-ml-data-preprocessing-5b346766fc48

https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/

IBM Training

Data preprocessing: Steps overview (cont.)

- **Data transformation:**
  - Scaling
  - Aggregation
  - Decomposition

Introduction to IBM Watson Machine Learning

Figure 4-15. Data preprocessing: Steps overview (cont.)

Data transformation (also called feature engineering) is a set of actions that covers transformation of the processed data. Engineering features from your data can use some time, but they can enhance the machine learning performance. There are three common data transformations:

- **Scaling**: The pre-processed data may contain attributes with a mixture of scales for various quantities, such as meters, grams, and dollars. The features should have the same scale, for example, 0 (smallest value) to 1 (largest value).
- **Aggregation**: There may be features that can be aggregated into a single feature, which is more meaningful to the problem that you are trying to solve.
- **Decomposition**: There may be complex features where it is more useful to split into parts. For example, a feature representing a date and time stamp in a long format can be split out further into only the hour of the day. Think about what your problem really needs.

References:

https://medium.com/datadriveninvestor/machine-learning-ml-data-preprocessing-5b346766fc48

https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/

4.3. Watson Machine Learning

IBM Training

Watson Machine Learning

Introduction to IBM Watson Machine Learning

Figure 4-16. Watson Machine Learning

IBM Training

Topics

- Cross-Industry Standard Process for Data Mining
- Data preparation
- Watson Machine Learning

Introduction to IBM Watson Machine Learning

Figure 4-17. Topics

IBM Training

Watson Machine Learning

- Watson Machine Learning is a service on IBM Cloud with features for training and deploying machine learning models and neural networks.
- Watson Machine Learning is integrated with IBM Watson Studio.
- Enables users to perform two fundamental operations of machine learning: training and scoring.
  - **Training** is the process of refining an algorithm so that it can learn from a data set.
  - **Scoring** is the operation of predicting an outcome by using a trained model.

Figure 4-18. Watson Machine Learning

Machine Learning is a service on IBM Cloud with features for training and deploying machine learning models and neural networks. To design, train, and deploy machine learning models in IBM Watson Studio, you must associate an IBM Watson Machine Learning service instance and supporting services (such as IBM Cloud Object Storage) with a project.

Machine Learning enables users to perform two fundamental operations of machine learning: training and scoring.

- **Training** is the process of refining an algorithm so that it can learn from a data set. The output of this operation is called a model. A model encompasses the learned coefficients of mathematical expressions.
- **Scoring** is the operation of predicting an outcome by using a trained model. The output of the scoring operation is another data set containing predicted values.

References:

https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-overview.html

https://console.bluemix.net/docs/services/PredictiveModeling/index.html#WMLgettingstarted

IBM Training

Watson Machine Learning features

- Interfaces for building, training, and deploying models:
  - Python client library external link
  - Command-line interface (CLI)
  - REST API external link
- Deployment infrastructure
- Distributed deep learning
- GPUs for faster training

Introduction to IBM Watson Machine Learning

Figure 4-19. Watson Machine Learning features

The Machine Learning service provides the following features:

- Interfaces for building, training, and deploying models: a Python client library external link, CLI, and a REST API external link.
- Deployment infrastructure for hosting your trained models. Although training is a critical step in the machine learning process, Machine Learning enables you to streamline the functioning of your models by deploying them and getting business value from them over time and through all of their iterations.
- Distributed deep learning for distributing training runs across multiple servers.
- GPUs for faster training.

References:

https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-overview.html

https://console.bluemix.net/docs/services/PredictiveModeling/index.html#WMLgettingstarted

IBM Training

Watson Machine Learning personas

- Data scientists:
  - Use data transformations and machine learning algorithms.
  - Use notebooks or external tools.
  - Often collaborate with data engineers to explore and understand the data.
- Developers build intelligent applications that use the predictions output of machine learning models.

Introduction to IBM Watson Machine Learning

Figure 4-20. Watson Machine Learning personas

Data scientists create machine learning pipelines that use data transformations and machine learning algorithms. They typically use notebooks or external tools to train and evaluate their models. Data scientists often collaborate with data engineers to explore and understand the data.

Developers build intelligent applications that use the predictions output by machine learning models.

References:

https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-overview.html

https://console.bluemix.net/docs/services/PredictiveModeling/index.html#WMLgettingstarted

IBM Training

Watson Machine Learning: Building the model

You can build your model by using one of the following tools (shown on this and the next slide):

1. Model builder:
   - Automatic data preparation
   - Technique recommendation
   - Manual mode to select the estimator

Introduction to IBM Watson Machine Learning

Figure 4-21. Watson Machine Learning: Building the model

There are different tools that you can use to create a machine learning model:

- Model builder.
- Flow editor.
- A notebook to prepare data, train the model, and deploy the model.

You build your model by selecting the data, manipulating the data, and selecting estimators or algorithms to use for classification. You can transform the data by appending new columns or mapping existing data to a new column. An estimator trains the data and produces a model, which you can deploy and use for prediction.

Before you can create a model, you must have an Apache Spark service and a machine learning service. If you do not already have a machine learning service, you are given a link and directed to create one during the initial model creation step.

The model builder is step-by-step process that uses a model by using popular machine learning algorithms. All that you have to do is simply upload your training data, and then let the model builder automatically prepare your data and recommend techniques that suit your data.

For automatic data preparation, the model builder automatically prepares your training data (in both automatic mode and manual mode). For example, it extracts the first 1000 records as a data sample to determine whether string categories exceed the maximum that are allowed. It also handles missing string values, and defines missing values as a separate category. If the label column contains strings, the model encodes the label column to a column of indexes and converts features that are categorical to category indexes.

Technique recommendation

For example, it recommends using a binary classification, multi-classification, or regression.

Manual mode to select an estimator

By using *manual mode* in the model builder, you can choose one or more specific estimators, such as linear regression, logistic regression, or a decision tree.

References:

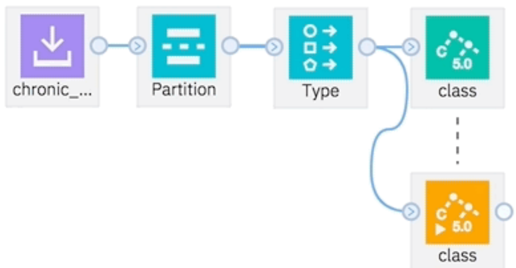
https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-pipelines.html

https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-model-builder.html#data

IBM Training

Watson Machine Learning: Building the model (cont.)

2. Flow Editor (modeler flow):
   - Use the Flow Editor to create a machine learning flow.
   - Use the Flow Editor to create a deep learning flow.



3. Use a notebook.

Introduction to IBM Watson Machine Learning

Figure 4-22. Watson Machine Learning: Building the model (cont.)

You can create a machine learning flow, which is a graphical representation of a data model, or a deep learning flow, which is a graphical representation of a neural network design, by using the **Flow Editor**. Use it to prepare or shape data, train or deploy a model, or transform data and export it back to a database table or file in IBM Cloud Object Storage.

The third tool that you can use to create or use a machine learning model is to use a notebook to write the code and implement the machine learning API. After a model is created, trained, and deployed, you can run the deployed model in a notebook.

References:

https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-canvas.html

https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-notebook.html

IBM Training

Watson Machine Learning: Deploying the model

- A deployed model is saved to the model repository.
- Deployments methods:
  - Deploy a model from a notebook.
  - Deploy a model from the Flow Editor.
  - Deploy a model from the model builder.
  - Deploy a batch or steaming model.
- Continuous learning and automated monitoring of model performance.

Introduction to IBM Watson Machine Learning

Figure 4-23. Watson Machine Learning: Deploying the model

After you create, train, and evaluate a model, you can deploy it. Although it is possible to score a model without deploying it, a model must be deployed before it can be scored from the Watson Machine Learning APIs. Also, a model can have only a single deployment. For a trial account, a user can have only one deployed model at a time.

When you deploy a model, you save it to the model repository that is associated with your Watson Machine Learning service. Then, you can use your deployed model to score data and build an application.

There are different ways to deploy your model:

- Deploy a model from a notebook.
- Deploy a model from the Flow Editor.
- Deploy a model from the model builder.
- Deploy a batch or steaming model.

Because model deployment is not a one-time event, you can use the Machine Learning continuous learning system, which provides automated monitoring of model performance, retraining, and redeployment to ensure prediction quality.

Reference:

https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-deploy.html

IBM Training

Unit summary

- Describe the Cross-Industry Standard Process for Data Mining (CRISP-DM) process model.
- Explain where machine learning fits in the CRISP-DM process.
- Describe data preparation before feeding into machine learning algorithms.
- Describe Watson Machine Learning features and capabilities.

Introduction to IBM Watson Machine Learning

Figure 4-24. Unit summary

IBM Training IBM

Review questions

1. In CRISP-DM, which phase focuses on machine learning?
   - A. Business understanding
   - B. Data preparation
   - C. Modeling
   - D. Evaluation
   - E. Deployment

2. True or False: Watson Machine Learning integrates with Watson Studio.

Introduction to IBM Watson Machine Learning

Figure 4-25. Review questions

1. C: Modeling.
2. True

IBM Training

Review questions (cont.)

3. The most time-consuming phase in CRISP-DM is:
   A. Business understanding
   B. Data preparation
   C. Evaluation preparation
   D. Deployment preparation
4. Which of the following is **not** part of data processing?
   A. Data Cleaning
   B. Data prediction
   C. Data Integration
   D. Data Sampling
   E. Data formatting

Introduction to IBM Watson Machine Learning

Figure 4-26. Review questions (cont.)

3. B. Data preparation

4. B. Data prediction

IBM Training

Review questions (cont.)

5. Which of the following are Watson Machine Learning personas?
   A. Data scientist
   B. Business analyst
   C. Developers
   D. Project manager
   E. Tester
   F. All of the above
   G. A & C
   H. C & E

Introduction to IBM Watson Machine Learning

Figure 4-27. Review questions (cont.)

3. G. A (Data Scientist) and C (Developer)

IBM Training

Review answers

1. In CRISP-DM, which phase focuses on machine learning?
   A. Business understanding
   B. Data preparation
   C. **Modeling**
   D. Evaluation
   E. Deployment

2. **True** or False: Watson Machine Learning integrates with Watson Studio.

Figure 4-28. Review answers

1. C: Modeling.
2. True

IBM Training

Review answers (cont.)

3. The most time-consuming phase in CRISP-DM is:
   A. Business understanding
   B. Data preparation
   C. Evaluation preparation
   D. Deployment preparation
4. Which of the following is **not** part of data processing?
   A. Data Cleaning
   B. Data prediction
   C. Data Integration
   D. Data Sampling
   E. Data formatting

Introduction to IBM Watson Machine Learning

Figure 4-29. Review answers (cont.)

3. B. Data preparation

4. B. Data prediction

IBM Training

Review answers (cont.)

5. Which of the following are Watson Machine Learning personas?
   A. Data scientist
   B. Business analyst
   C. Developers
   D. Project manager
   E. Tester
   F. All of the above
   G. A & C
   H. C & E

Introduction to IBM Watson Machine Learning

Figure 4-30. Review answers (cont.)

3. G. A (Data Scientist) and C (Developer)

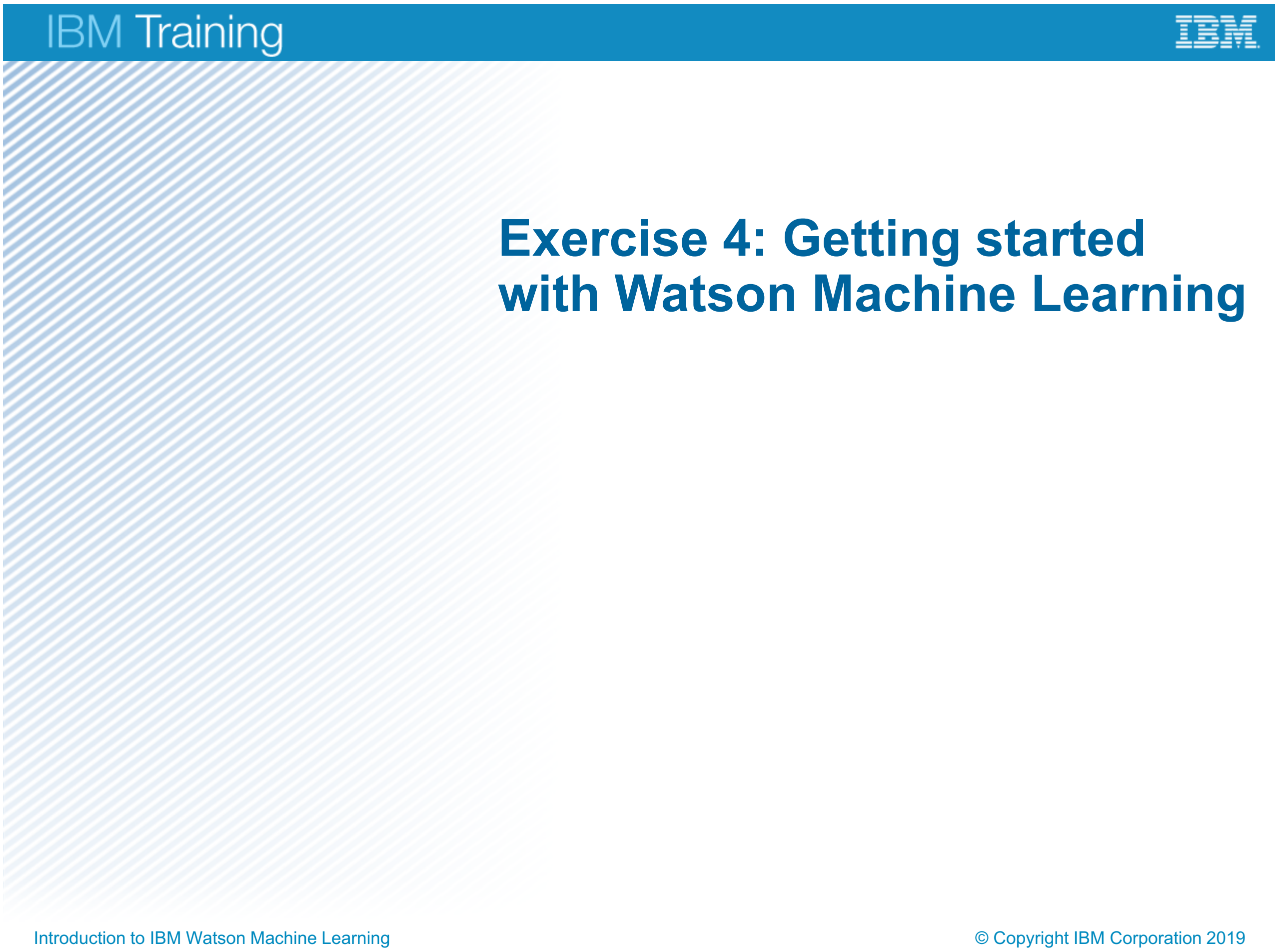


Figure 4-31. Exercise 4: Getting started with Watson Machine Learning

IBM Training

Exercise objectives

- This exercise introduces you to the basic tasks that you have to perform while building machine learning models for different algorithms using Watson Machine Learning service and Watson Studio.
- After completing this exercise, you should be able to:
  - Create a machine learning model by using Watson Studio and Watson Machine Learning.
  - Use data sets to train the model.
  - Use different estimators to train the machine learning mode, which represent different machine learning algorithms.
  - Deploy machine learning models.
  - Evaluate the deployed models.
  - Call the deployed models from your applications.
  - Test the model with your data.

Introduction to IBM Watson Machine Learning

Figure 4-32. Exercise objectives

After completing this exercise, you should be able to:

- Create a machine learning model by using Watson Studio and Watson Machine Learning.
- Use data sets to train the model.
- Use different estimators to train the machine learning mode, which represent different machine learning algorithms.
- Deploy machine learning models.
- Evaluate the deployed models.
- Call the deployed models from your applications.
- Test the model with your data.

Unit 5. Introduction to natural language processing

Estimated time

00:30

Overview

This unit introduces Natural Language Processing. It covers key applications of NLP, basics concepts and terminology, tools and services and NLP challenges.

IBM Training

Unit objectives

- Explain what natural language processing is.
- Identify NLP use cases.
- Explain basic NLP concepts and terminology.
- List the tools and services for NLP.

Introduction to natural language processing

Figure 5-1. Unit objectives

5.1. Natural language processing overview

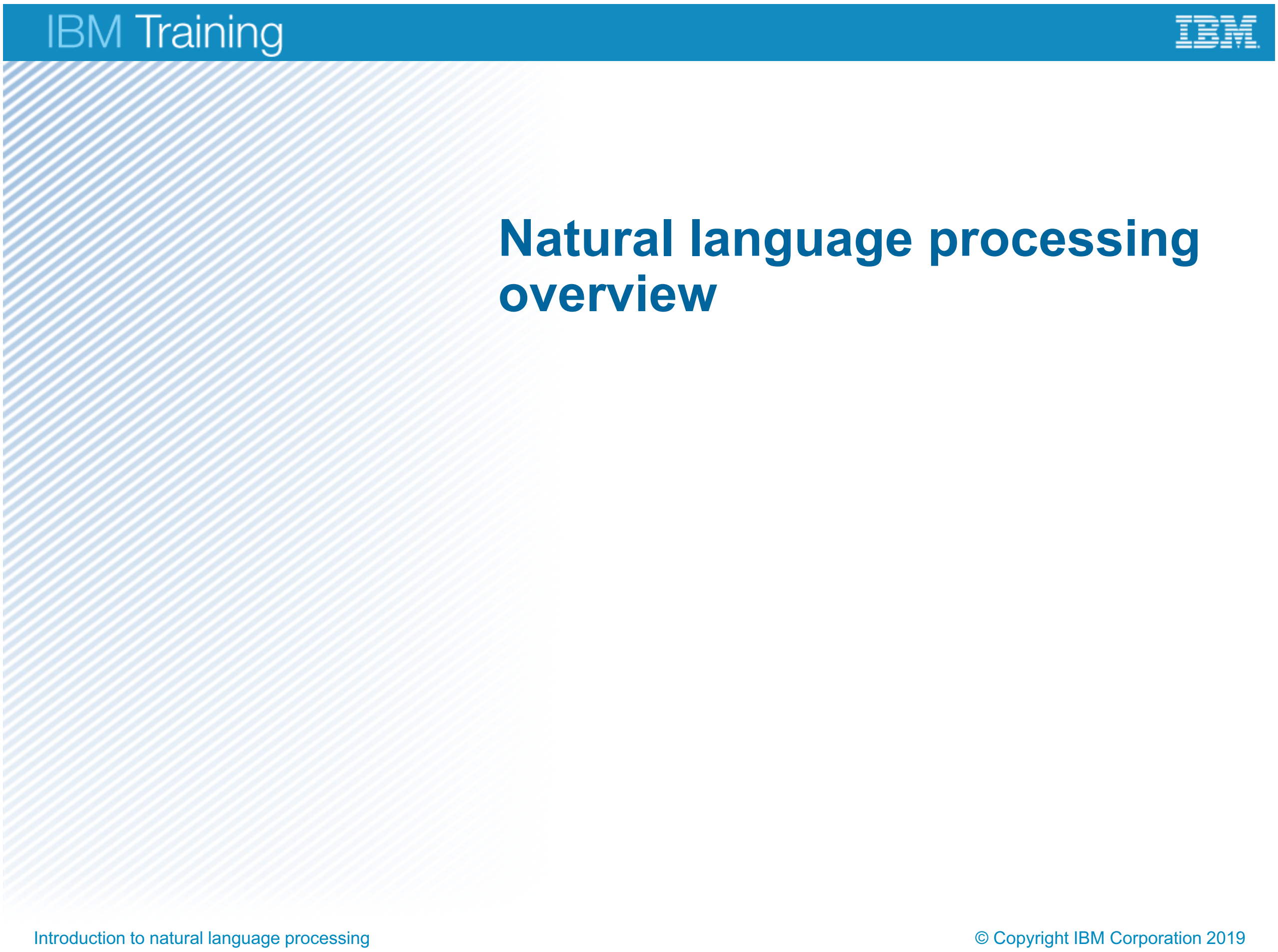


Figure 5-2. Natural language processing overview

IBM Training

Topics

- Natural language processing overview
- Natural language processing use cases
- Natural language processing basic concepts and terminology
- Natural language processing tools and services

Introduction to natural language processing

Figure 5-3. Topics

IBM Training

What is natural language processing

- NLP is the study of the computational treatment of natural (human) language.
- It enables machines to understand human communication to extract different information.
- Examples of NLP applications: Analysis of text in emails, human speech, social media, or optical character recognition (OCR) from documents (text that is scanned from actual documents).
- NLP has its origins in machine translation from the 1950s.
- NLP advanced over the years by combining the power of artificial intelligence (AI), computational linguistics, and computer science.

Introduction to natural language processing

Figure 5-4. What is natural language processing

NLP stands for natural language processing. It is a subfield of computer science and AI concerned with the processing human language by computers. It is one of the most important fields in computer science in both industry and academia.

NLP enables machines to understand human communication to extract different information.

Examples of NLP applications include analysis of text in emails, human speech, social media, or optical character recognition (OCR) from documents (text that is scanned from actual documents).

NLP has its origins in machine translation from the 1950s.

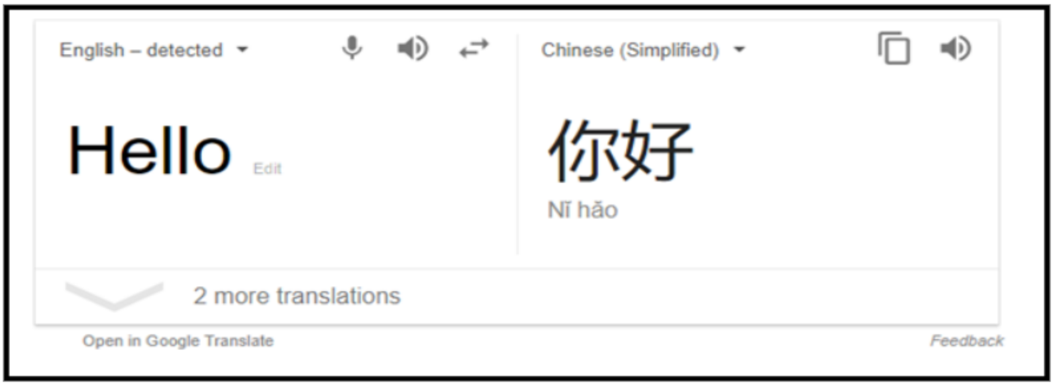
The first machine translation was from English to Russian and vice versa, but with poor and inaccurate results.

Machine translation and other NLP applications advanced over the years by combining the power of artificial intelligence (AI), computational linguistics, and computer science.

IBM Training

Natural language processing applications

- Machine translation
- Information retrieval: Search engines
- Spell checkers
- Natural language assistants



Google Translate

Figure 5-5. Natural language processing applications

Here are the most popular NLP tasks:

- Machine translation: Automatically translating one language to another
- Information retrieval: Search engines, such as Google and Bing
- Spell checkers
- Natural language assistants, such as Siri and Alexa

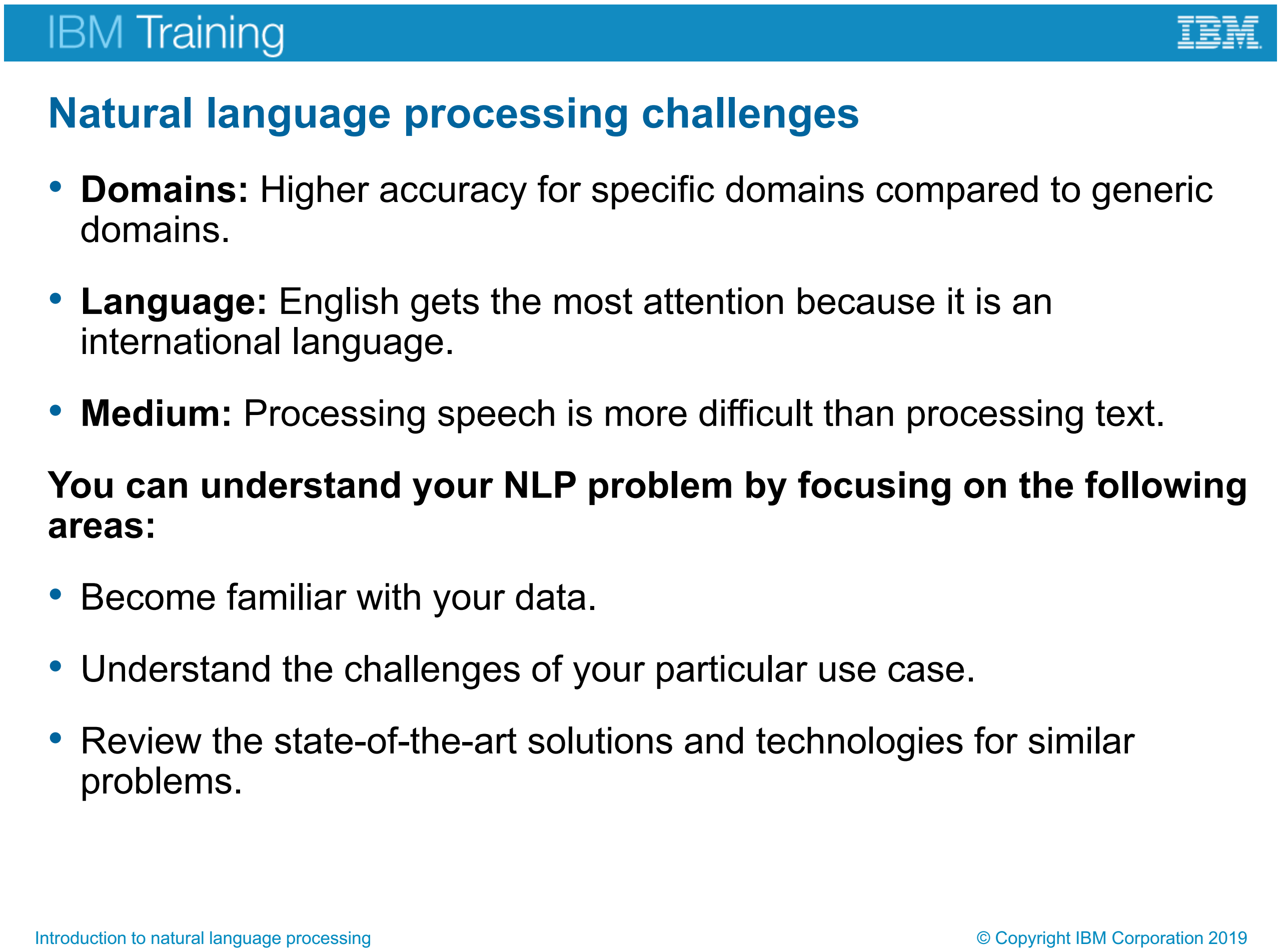


Figure 5-6. Natural language processing challenges

There are always some challenges that need to be tackled for any case. In NLP, here are the most popular challenges:

- **Domains**: Higher accuracy for specific domains compared to generic domains.
- **Language:** English gets the most attention because it is an international language.
- **Medium:** Processing speech is more difficult than processing text.

To resolve some of these challenges, you must become familiar with your data and understand the challenges of your particular use case. Think about how you will acquire the data and how to validate its quality. Think about the deployment of your solution and how you are planning to cover all these points.

Finally, review the state-of-the-art solutions and technologies for similar problems and how these issues were resolved.

5.2. Natural language processing use cases

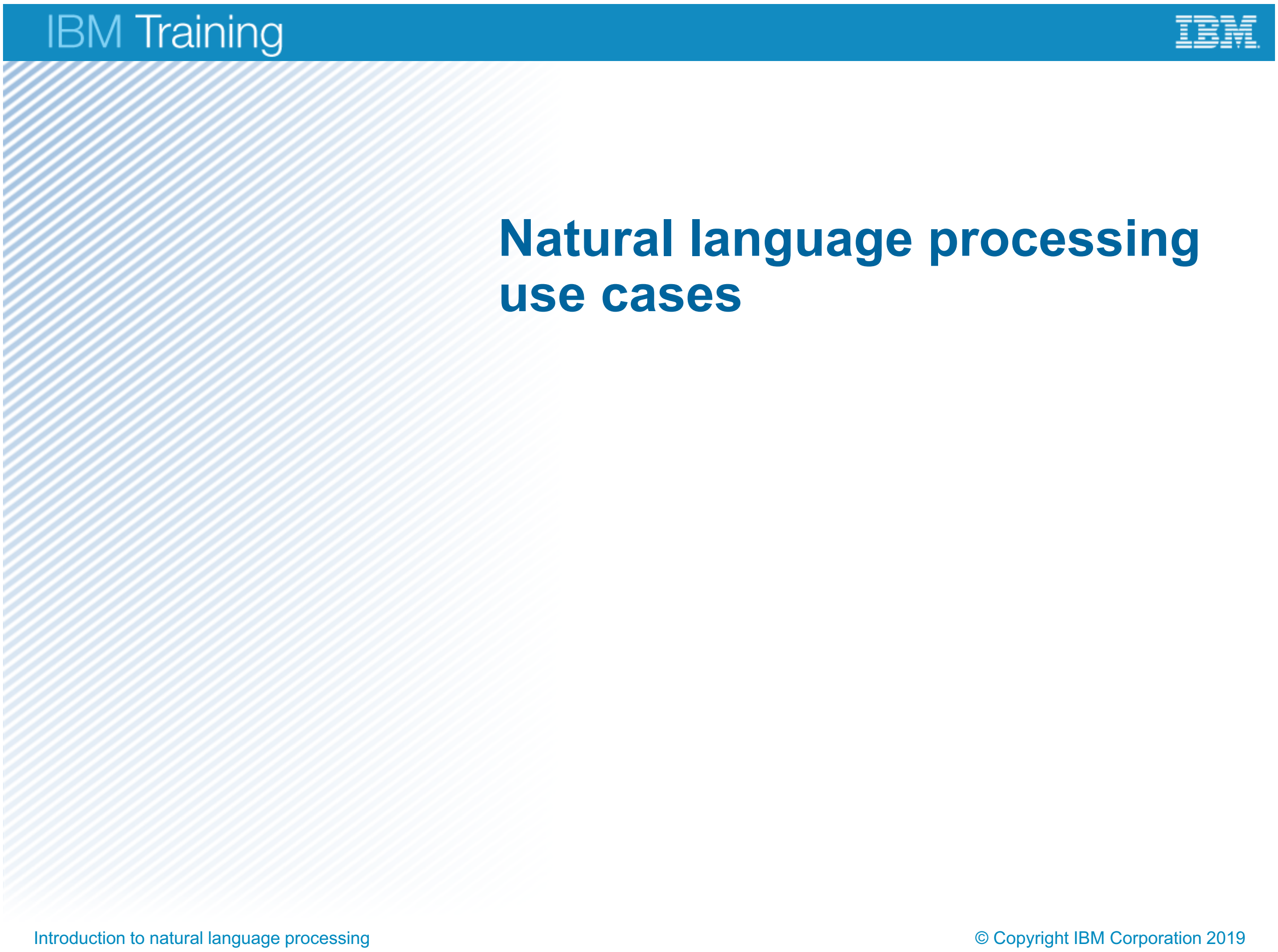


Figure 5-7. Natural language processing use cases

IBM Training

Topics

- Natural language processing overview
- Natural language processing use cases
- Natural language processing basic concepts and terminology
- Natural language processing tools and services

Introduction to natural language processing

Figure 5-8. Topics

IBM Training

Natural language processing use cases

Information extraction

- **Goal:** Parse the input text to extract valuable output.
  - Examples: Entity extraction, relation identification extraction, text summarization
- **Unstructured text**: Dynamic structure (for example emails, newspaper articles, and user reviews).
- **Structured text:** Defined structure (for example, a database table).

Introduction to natural language processing

Figure 5-9. Natural language processing use cases

NLP technology can be used to extracting information from unstructured text such as emails, newspaper articles, and user reviews into structured text.

Entity extraction refers to extracting *entities* from the text such as organizations, people, locations and so on. For example, the World Health Organization, IBM, Sara, John, Paris, US.

Relation extraction refers to identifying the relationship between entities, for example, “Abraham Lincoln was a US president”; “Ginni Rometty is the CEO of IBM”.

Text summarization refers to the technique of shortening long pieces of text. Automatic text summarization is a common use case in machine learning and natural language processing.

Structured text mostly takes the form of tables or values in a structured form.

The goal of information extraction is to parse the incoming text, identify important mentions and their relations, and extract valuable output into structured text. Doing so can be used to automate the process of reading articles and passages to convert this information into a structured format. Computer systems can then manage this information and take proper actions.

IBM Training

Natural language processing use cases (cont.)

Sentiment analysis

- The process of identifying emotions or opinions that are expressed in user input.
- This analysis is used in marketing, retention plans, and emotional intelligence for chatbots.

User Reviews

Her is Spike Jonze's finest film yet
29 December 2013 | by monkyman347 (California) – See all my reviews

I've been a fan of Spike Jonze's films since I first saw Being John Malkovich. Although the wonderful script deserves some of the credit for making that film so great, it was immediately clear to me that Spike Jonze was a director with a fresh and imaginative perspective. His next film, the 2002 meta-comedy Adaptation, confirmed this with its dry wit and multilayered narrative. Now, after a slightly less successful (but still enjoyable) adaptation of Where The Wild Things Are, Spike Jonze has written and directed his most complete and poignant film yet, Her.

The story, taking place in a near future when people spend more time talking to their computers than they do to each other, stars Joaquin Phoenix as Theodore Twombly, a lonely man whose job is to write heartfelt personal letters for people not willing to do it themselves. Theodore happens to see an ad for a new computer operating system that is programmed with a personality, and decides to give it a shot. His new operating system Samantha, voiced by Scarlett Johansson, is not only intelligent but also charming and understanding, and she and Theo quickly fall in love.

It's understandable if that premise sounds bizarre on paper, but in execution Her is far more sweet than creepy. The film radiates warmth and intelligence, and there is a fair amount of witty humor to ensure that it never becomes too self-serious. It has an engaging style similar to that of Sofia Coppola's Lost In Translation. Like in that film, there's a certain poetic yet whimsical quality to the dialogue in Her and both the main characters are plagued by feelings of loneliness.

Introduction to natural language processing

Figure 5-10. Natural language processing use cases (cont.)

Sentiment analysis is the process of identifying emotions or opinions that are expressed in user input.

Sentiment analysis answers various questions, such as how people feel about your product or whether your customers are satisfied with your customer service.

It is used in marketing and retention plans, and emotional intelligence for chatbots, that is, it enables chatbots to direct the conversation.

Machine learning algorithms brought many advances to this field and are still improving.

IBM Training

Natural language processing use cases (cont.)

- **Question and answering**: Building solutions that can answer questions that are asked by humans in natural language.
  - Examples: Natural language questions used to retrieve answers from forums; FAQ application; chatbots
- **Speech recognition:** Converts spoken language into text.
  - Example: Chatbot interactive talk

Introduction to natural language processing

Figure 5-11. Natural language processing use cases (cont.)

Question and answering

Building solutions that can answer questions that are asked by humans in natural language. A question and answering system can be used for the following tasks:

- Retrieving answers from forums.
- Building a Frequently Asked Questions (FAQs) system.
- Training chatbots

Speech recognition is another use case that helps advancing the capabilities of many different applications. It converts spoken language into text. It can be used in many applications in several domains, such as having an interactive talk with a chatbot. It can also be used in Internet of Things (IoT) applications.

5.3. Natural language processing basic concepts and terminology

Figure 5-12. Natural language processing basic concepts and terminology

This section introduces basic concepts and terminologies such as synonymy, polysemy, hyponymy, and hypernymy. The taxonomy for similar concepts has applications in the education and machine learning fields because they rely on word-sense disambiguation.

IBM Training

Topics

- Natural language processing overview
- Natural language processing use cases
- Natural language processing basic concepts and terminology
- Natural language processing tools and services

Introduction to natural language processing

Figure 5-13. Topics

IBM Training

Basic concepts and terminologies

- **Synonyms:** Words that are written differently but are similar in meaning.
  - Example: Clever and smart
- **Antonyms:** Words that have meanings that are opposite to each other.
  - Example: Clever and stupid
- **Usage example:** In information retrieval, you might want to expand the keywords search by retrieving the synonyms of the query words.

Introduction to natural language processing

Figure 5-14. Basic concepts and terminologies

Synonyms are words that are written differently but are similar in meaning. For example:

- Clever and smart
- Begin and start
- Beautiful and pretty
- Sad and unhappy

Antonyms are words that have meanings that are opposite to each other. For example:

- Clever and stupid
- Begin and end
- Beautiful and ugly
- Sad and happy

Usage example: In information retrieval, you might want to expand the keywords search by retrieving the synonyms of the query words.

IBM Training

Basic concepts and terminologies (cont.)

Homonyms: Words that have the same written form but have unrelated meanings. There are two types of homonyms:

- **Homographs:** Words that have the same written form. For example:
  - This answer is **right**.
  - The building is on the **right** side of the river.
  - You have the **right** to remain silent.
  - Come here **right** now.
- **Homophones:** Words that sound similar when spoken but have different meanings and spellings. For example:
  - "left" and "lift".
  - "right" and "write".

Introduction to natural language processing

Figure 5-15. Basic concepts and terminologies (cont.)

Homonyms are words that have the same written form but have unrelated meanings. There are two types of homonyms:

- Homographs
- Homophones

Homographs are words that have the same written form. For example:

- This answer is ***right***.
- The building is on the ***right*** side of the river.
- You have the ***right*** to remain silent.
- Come here ***right*** now.

Although the word ***right*** has the same written form in the examples, you notice the difference between the meanings in each sentence.

Homophones are words that sound similar when spoken but have different meanings and spellings. For example:

- "left" and "lift".
- "right" and "write".

IBM Training

Basic concepts and terminologies (cont.)

- Homonyms **challenges**:
  - How do you translate “right” so that it has the correct meaning?
  - How do you differentiate two words that sound similar when you convert speech to text?

Figure 5-16. Basic concepts and terminologies (cont.)

Homonyms introduce challenges into NLP operations such as machine translation and speech recognition. How do you translate *right* so that it has the correct meaning? How do you differentiate two words that sound similar when you convert speech to text?

IBM Training

Basic concepts and terminologies (cont.)

- **Polysemy:** Words that have the same written form and a related meaning. For example:
  - You must face your fear.
  - Her face is beautiful.
- **Hyponymy:** A word is a hyponym of another word if it represents a subclass of the other word. For example:
  - Orange is a hyponym of fruit.
  - Yellow is a hyponym of color.

Introduction to natural language processing

Figure 5-17. Basic concepts and terminologies (cont.)

Polysemy refers to words that have the same written form and related meaning. For example:

- You must *face* your fear.
- Her face is beautiful.

Hyponymy: A word is a hyponym of another word if it represents a subclass of the other word. For example:

- Orange is a hyponym of fruit.
- Yellow is a hyponym of color.

IBM Training

Basic concepts and terminologies (cont.)

- **Hypernymy:** One word is the hypernym of another word if it represents a superclass of the other word. For example:
  - Fruit is a hypernym of orange.
  - Color is a hypernym of yellow.
- **Usage example:** Comparing the semantic similarity.

Introduction to natural language processing

Figure 5-18. Basic concepts and terminologies (cont.)

Hypernymy: One word is the hypernym of another word if it represents a superclass of the other word. For example:

- *Fruit* is a hypernym of orange.
- *Color* is a hypernym of yellow.

Usage example: Comparing the semantic similarity.

5.4. Natural language processing tools and services

IBM Training

Natural language processing tools and services

Introduction to natural language processing

Figure 5-19. Natural language processing tools and services

IBM Training

Topics

- Natural language processing overview
- Natural language processing use cases
- Natural language processing basic concepts and terminology
- Natural language processing tools and services

Introduction to natural language processing

Figure 5-20. Topics

IBM Training

Natural language processing tools and services

Open-source NLP tools:

- **Apache OpenNLP:** Provides tokenizers, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, co-reference resolution, and more.
- **Stanford Core NLP:** A suite of NLP tools that provide part-of-speech tagging, a named entity recognizer, a co-reference resolution system, sentiment analysis, and more.
- **Natural Language Toolkit (NLTK):** A Python library that provides modules for processing text, classifying, tokenizing, stemming, tagging, parsing, and more.
- **WordNet:** One of the most popular lexical databases for the English language. Supported by various API and programming languages.

Introduction to natural language processing

Figure 5-21. Natural language processing tools and services

There are many open source tools that you can use for NLP. For example:

- Open NLP that is based on Java. It provides many functions for text processing, such as tokenizers, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, co-reference resolution, and more. For more information, see https://opennlp.apache.org/
- Stanford Core NLP, which is written in Java. It is a suite of NLP tools that provide part-of-speech tagging, a named entity recognizer, a co-reference resolution system, sentiment analysis, and more. It supports many languages, such as English, German, French, Arabic, Spanish, and Chinese. For more information, see https://stanfordnlp.github.io/CoreNLP/.
- NLTK provides the same processes as the other NLP suites, but in the Python language. For more information, see https://www.nltk.org/.
- WordNet is a popular lexical database that is used in research. There are many APIs and languages that you can use to access WordNet. For example, you can make a call to retrieve a synonym of a word. WordNet is available online and as an offline version that you can download. For more information, see https://wordnet.princeton.edu/.

There are other libraries, such as Unstructured Information Management Architecture (UIMA). IBM Watson uses UIMA to analyze unstructured data. The Apache Clinical Text Analysis and Knowledge Extraction System (Apache cTAKES) is a UIMA-based system that is used to extract information from medical records.

IBM Training

Natural language processing tools and services (cont.)

Services:

- Examples: IBM Cloud, Microsoft Cloud (Azure), and Google Cloud
- IBM offers its AI services through IBM Cloud. The NLP services that are provided include the following ones (among others):
  - Watson Natural Language Classifier for text classification
  - Watson Natural Language Understanding for entity identification and relation extraction

Figure 5-22. Natural language processing tools and services (cont.)

Instead of using low-level libraries, you can use many cloud services that accomplish high-level NLP tasks, for example, IBM Cloud, Microsoft Cloud (Azure), and Google Cloud.

IBM offers its AI services through IBM Cloud. The NLP services that are provided include the following ones (among others):

- Watson Natural Language Classifier for text classification
- Watson Natural Language Understanding for entity identification and relation extraction

You can rapidly build a complete NLP application by using these services.

IBM Training

Unit summary

- Explain what natural language processing is.
- Identify NLP use cases.
- Explain basic NLP concepts and terminology.
- List the tools and services for NLP.

Introduction to natural language processing

Figure 5-23. Unit summary

IBM Training

Review questions

1. Email spam detection is a use case of:
   A. Sentiment analysis
   B. Classification
   C. Chatbots
   D. Translation
2. “Apple" is the …… of “fruit"
   A. Synonym
   B. Antonym
   C. Hypernymy
   D. Hyponymy

Introduction to natural language processing

Figure 5-24. Review questions

1. B. Classification.
2. D. Hyponymy.

IBM Training

Review questions (cont.)

3. What are examples of information extraction?
   A. Entities identification
   B. Relation identification
   C. Automatic summarization
   D. All the above
4. “Ugly" is the …… of “pretty“.
   A. Synonym
   B. Antonym
   C. Hypernymy
   D. Hyponymy

Introduction to natural language processing

Figure 5-25. Review questions (cont.)

3. D. All the above.
4. B. Antonym.

IBM Training

Review answers

1. Email spam detection is a use case of:
   A. Sentiment analysis
   B. Classification
   C. Chatbots
   D. Translation
2. “Apple" is the …… of “fruit"
   A. Synonym
   B. Antonym
   C. Hypernymy
   D. Hyponymy

Introduction to natural language processing

Figure 5-26. Review answers

1. Classification.
2. Hyponymy.

IBM Training

Review answers (cont.)

3. What are examples of information extraction?
   A. Entities identification
   B. Relation identification
   C. Automatic summarization
   D. All of the above
4. “Ugly" is the …… of “Pretty"
   A. Synonym
   B. Antonym.
   C. Hypernymy
   D. Hyponymy.

Introduction to natural language processing

Figure 5-27. Review answers (cont.)

3. All the above.
4. Antonym.

Unit 6. Natural language processing concepts and components

Estimated time

00:30

Overview

This unit covers natural language processing components, the NLP pipeline, natural language understanding, natural language generation, information retrieval, and information extraction.

IBM Training

Unit objectives

- Define the NLP categories.
- Describe the NLP pipeline.
- Explain the challenges in natural language understanding (NLU).
- Explain the concepts of information retrieval and extraction.
- Describe sentiment analysis.

Natural language processing concepts and components

Figure 6-1. Unit objectives

6.1. Natural language processing categories

Figure 6-2. Natural language processing categories

IBM Training

Topics

- Natural language processing categories
- Language ambiguities
- Natural language processing pipeline
- Information retrieval
- Information extraction

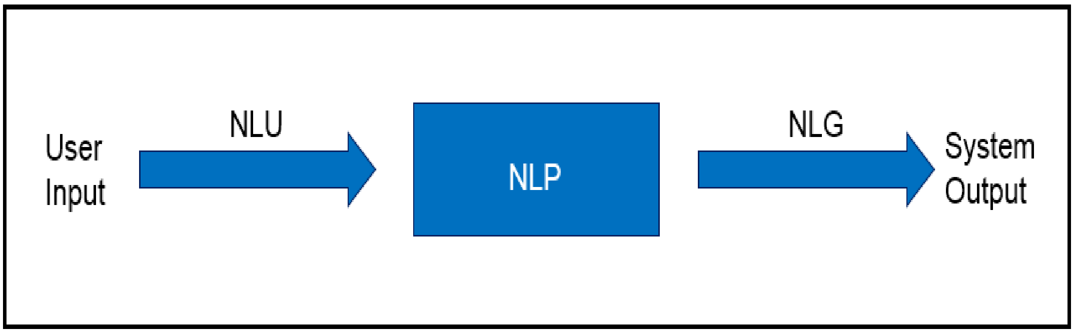
Natural language processing concepts and components

Figure 6-3. Topics

IBM Training

Natural language processing categories

- Natural language understanding (NLU)
- Natural language generation (NLG)



Natural language processing concepts and components

Figure 6-4. Natural language processing categories

There are two categories of NLP:

- Natural language understanding (NLU): It is the NLP task of extracting insights from natural language inputs.
- Natural language generation (NLG): it is the NLP task of building natural language outputs from non-linguistic inputs.

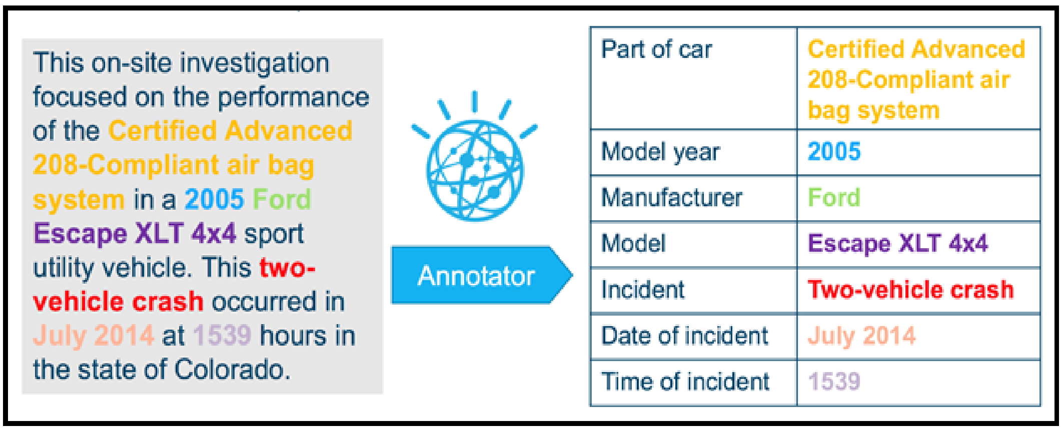
Natural language exists in various media, such as text in documents, reports, or messages, or speech.

The graphic in the slide shows that NLU is the NLP process that takes input text while NLG is the NLP process that generates text as output.

IBM Training

NLU applications

- Unstructured to structured
- Question and answer system
- Sentiment analysis



Natural language processing concepts and components

Figure 6-5. NLU applications

NLU analyzes language to gain insights into the text. There many examples of NLU applications:

- Mapping a user’s unstructured input to a computer representation (structured data) and relation extraction
- Question and answering system
- Sentiment analysis

The graphic in the slide shows an example of converting unstructured to structured data. The input to the NLU system is natural language (unstructured text) and through the NLP process it is converted to structured data (table).

References:

http://www.lacsc.org/papers/PaperA6.pdf

https://nlp.stanford.edu/~wcmac/papers/20140716-UNLU.pdf

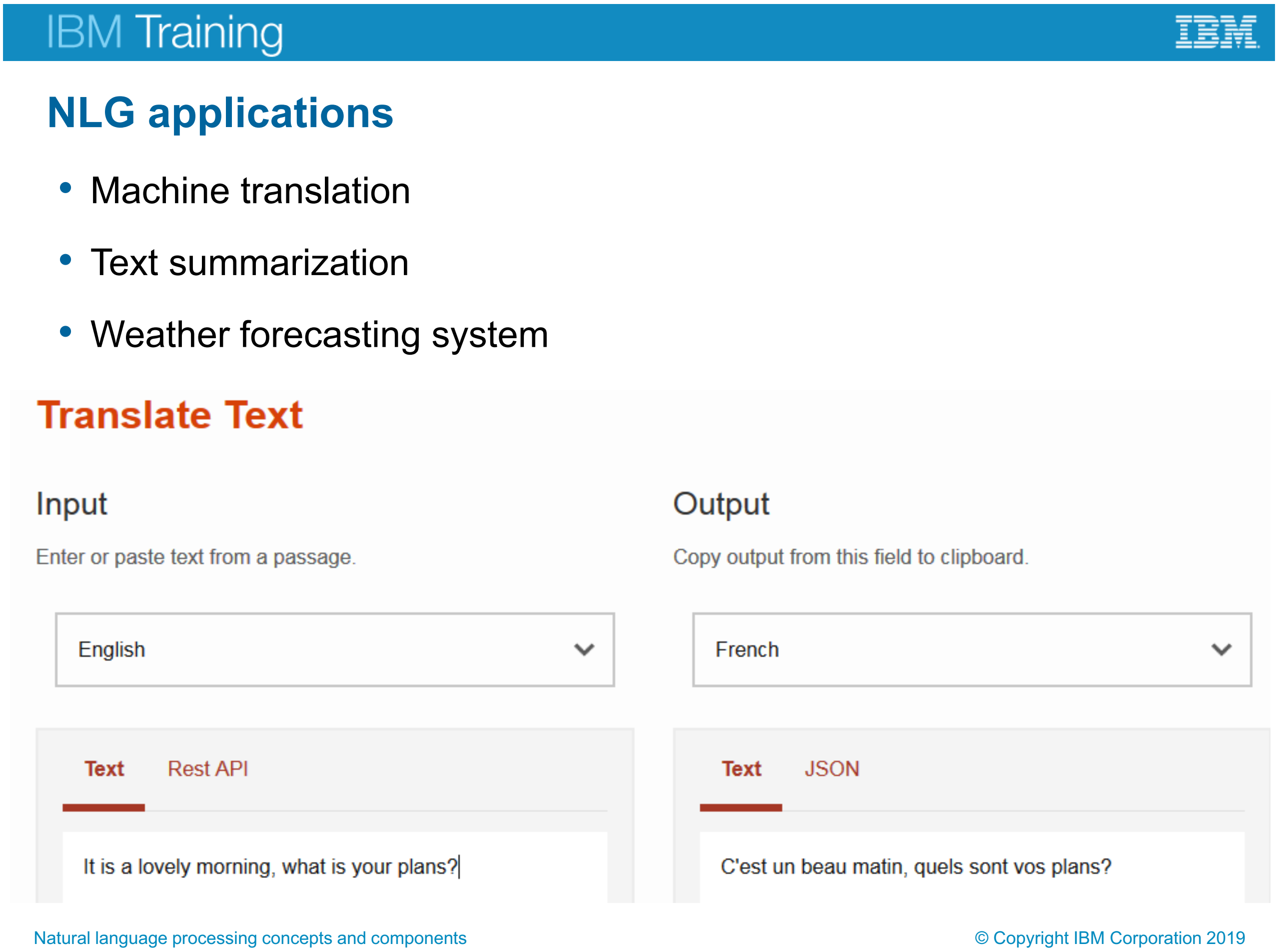


Figure 6-6. NLG applications

Here are some NLG application examples:

- Translator.
- Text summarization.
- Weather Forecasting Systems, which compile graphical weather maps representations and interpret the numbers that come from the prediction system. By using NLG, you can transform this information into natural language.

References:

https://www.ibm.com/watson/services/language-translator/

http://www.academia.edu/3879518/Natural_Language_Generation_Scope_Applications_and_Approaches

6.2. Language ambiguities

IBM Training

Language ambiguities

Natural language processing concepts and components

Figure 6-7. Language ambiguities

IBM Training

Topics

- Natural language processing categories
- Language ambiguities
- Natural language processing pipeline
- Information retrieval
- Information extraction

Natural language processing concepts and components

Figure 6-8. Topics

IBM Training

Language ambiguities

- **Lexical ambiguity**: A primitive level, such as at the word level
- Verb versus noun:
  - "We will ***dance*** all night."
  - "This is the salsa ***dance***."
  - "John ***will*** go to work."
  - "His uncle left him millions in his ***will***."

Natural language processing concepts and components

Figure 6-9. Language ambiguities

Building applications with NLP processes is not a trivial task. Natural language includes many ambiguities, such as a lexical ambiguity, which is a primitive level, such as at the word level.

Two examples:

- The word *dance* can be a verb ("We will *dance* all night.") or a noun ("This is the salsa *dance*.").
- The word *will* can be a helping verb to indicate an action in the future ("John *will* go to work") or a noun ("His uncle left him millions in his *will*.").

IBM Training

Language ambiguities (cont.)

- **Syntactic-level ambiguity:** A sentence that can be parsed in various ways.
  - Example: “She pointed at the guy with the umbrella.”
- **Anaphora ambiguity** occurs when it is not clear which one is the antecent noun that a pronoun replaces because more than one possible antecendent exists.
  - Example: “When Mary invited Ann to play she did not know that she would be late”.

Natural language processing concepts and components

Figure 6-10. Language ambiguities (cont.)

Here are two more types of language ambiguity:

- **Syntactic-level ambiguity:** A sentence that can be parsed in various ways. For example: “She pointed at the guy with the umbrella.” Did she use her umbrella to point at the guy or did she point at the guy who is carrying an umbrella?
- **Anaphora** is an expression for which the interpretation depends on another expression that was previously introduced. The referring term is called *anaphor* and it is usually a pronoun. The pronoun takes the place of a noun but, to avoid ambiguity, the pronoun must refer clearly to the noun that the pronoun replaces.
- An **anaphora ambiguity** occurs when more than one possible antecedent exists. For example: “When Mary invited Ann to play she did not know that she would be late.” Is the first “she” replacing Mary or Ann? Who did not know? Is the second “she” replacing Mary or Ann? Who would be late?

References:

https://en.wikipedia.org/wiki/Syntactic_ambiguity

https://en.wikipedia.org/wiki/Anaphora_(linguistics)

http://www.ijircce.com/upload/2014/sacaim/59_Paper%2027.pdf

6.3. Natural language processing pipeline

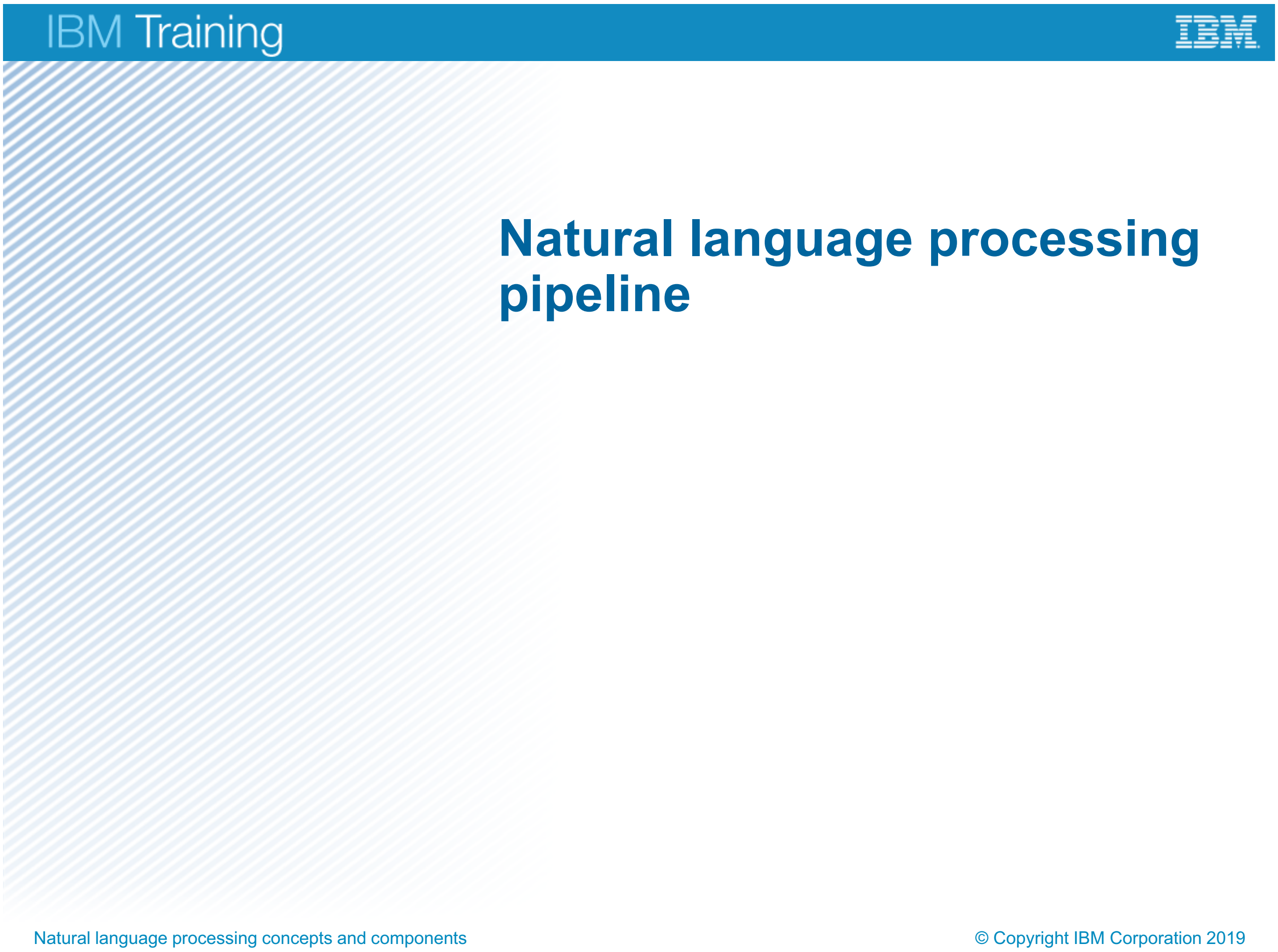


Figure 6-11. Natural language processing pipeline

IBM Training

Topics

- Natural language processing categories
- Language ambiguities
- Natural language processing pipeline
- Information retrieval
- Information extraction

Natural language processing concepts and components

Figure 6-12. Topics

IBM Training

Natural language processing pipeline

- A pipeline is a way to design a program in which the output of one module feeds into the input of the next module.
- Using the NLP pipeline divides the tasks of NLU, which makes NLU less complicated.
- For example, use NLP pipeline to understand the following sentence: "*Yes, I received your invitation, and I will happily attend your party*."

Natural language processing concepts and components

Figure 6-13. Natural language processing pipeline

When you work on an NLP task like machine translation, there are a set of processes and activities that you must perform. This set of processes is the NLP pipeline.

A pipeline is a process for designing a program in which the output of one module feeds into the input of the next module. You use the pipeline to break down the complexity of the NLP task into a smaller set of less complicated tasks. There are no strict rules for the activities that must be done in the pipeline.

In the following slides, we explore some of these activities by using the following example by assuming the following statement:

"Yes, I received your invitation, and I will happily attend your party."

This statement is understandable by a human but not a computer. We use the pipeline to walk though each stage to analyze the statement.

Reference:

https://stanfordnlp.github.io/CoreNLP/pipelines.html

IBM Training

Natural language processing pipeline (cont.)

1. Sentence segmentation:

- Detect sentence boundaries.
- There are two sentences in the example that are separated by the conjunction “and”:
  - “*Yes, I received your invitation, and I will happily attend your party*.”
- Apply this process to the example:
  - “Yes, I received your invitation.”
  - “I will happily attend your party.”

Natural language processing concepts and components © Copyright IBM Corporation 2019

Figure 6-14. Natural language processing pipeline (cont.)

Sentence segmentation focuses on finding the boundaries of sentences in text, that is, where the sentence starts and ends. This is not an easy task to accomplish due to the possible ambiguity that is caused by punctuation marks.

For example, a period might indicate the end of a sentence, but it might also indicate an abbreviation, a decimal point, or an email address.

Assume that the two sentences here are separated by the conjunction “and”, so the result of this stage is the following sentences:

- “Yes, I received your invitation.”
- “I will happily attend your party.”

Reference:

https://en.wikipedia.org/wiki/Sentence_boundary_disambiguation

IBM Training

Natural language processing pipeline (cont.)

2. Tokenization:

- Breaks a sentence into tokens.
- Tokenization uses delimiters, for example, a space ” “.
- Apply tokenization to the example:
  - First sentence: “Yes””,” “I” “received” “your” “invitation”
  - Second sentence: “I” “will” “happily” “attend” “your” “party”

Natural language processing concepts and components

Figure 6-15. Natural language processing pipeline (cont.)

Tokenization is a basic process that breaks a sentence into a group of words, punctuations, numbers, and alphanumerics that are called tokens. The tokenization can be done on multiple delimiters. Assume in the example that you use the white space as delimiter and apply tokenization to the example:
First sentence: “Yes””,” “I” “received” “your” “invitation”
Second sentence: “I” “will” “happily” “attend” “your” “party”

White space is a standard delimiter, but is not sufficiently useful by itself. For example, segmenting “New York” into “New” and “York” might introduce ambiguity because they both refer to a state, so you should preserve the space in-between.

To read more about “the art of tokenization”, see the following website:
https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization?lang=en

Note

Sentence segmentation, tokenization, and other activities in the pipeline might also differ depending on the language. For example, you cannot use the same English tokenizer with the Japanese language.

IBM Training

Natural language processing pipeline (cont.)

3. Parts of speech (POS) tagging:

- Tagging each token by what it corresponds to in a certain data set (English grammar).
- POS helps the computer to understand language and grammar and derive meaning from the input sentence.
- The most famous data sets is the **Penn Treebank project**.

Natural language processing concepts and components

Figure 6-16. Natural language processing pipeline (cont.)

POS tagging: The process of tagging each token with its grammatical representation, such as noun, verb, or adjective.

POS helps the computer to understand language and grammar and derive meaning from the input sentence.

In the example, the data set is English grammar, so according to the POS that are defined, we want to label each token by its proper grammar value, that is, the corresponding part of speech.

POS helps the computer to understand language and grammar and derive meaning from the input sentence. Many human-annotated data sets are publicly available. One of the most famous data sets is the one used by the Penn Treebank project.

References:

https://en.wikipedia.org/wiki/Part-of-speech_tagging

https://nlp.stanford.edu/software/tagger.shtml

IBM Training

Natural language processing pipeline (cont.)

Example of POS tagging

Tag	Description
JJ	Adjective
NN	Noun Single
NNS	Noun Plural
RB	Adverb
RP	Particle
VB	Verb, base form
VBD	Verb, past tense

Natural language processing concepts and components

Figure 6-17. Natural language processing pipeline (cont.)

Review the tags example to learn about tagging.

IBM Training

Natural language processing pipeline (cont.)

Applying NLP pipeline steps to the example:

- Example: “*Yes, I received your invitation, and I will happily attend your party*.”
- The first sentence is: “yes””,” “I” “got” “your” “invitation”.
- The second sentence is: “I” “will” “happily” “attend” “your” “party”.
- Apply POS tags to both sentences.
  - First sentence: “yes/UH” “,/,” ”I/PRP” “got/VBD” “your/PRP$” “invitation/NN” “,/,”
  - Second sentence: “I/PRP” “will/MD” “happily/RB” “attend/VB” “your/PRP$” “party/NN”

Natural language processing concepts and components

Figure 6-18. Natural language processing pipeline (cont.)

Example: “*Yes, I received your invitation, and I will happily attend your party*.”

1. Apply sentence segmentation:
   - Sentence 1: Yes, I received your invitation.
   - Sentence 2: I will happily attend your party.
2. Apply tokenization by using white space as delimiter:
   - The first sentence is: “yes””,” “I” “got” “your” “invitation”.
   - The second sentence is: “I” “will” “happily” “attend” “your” “party”.
3. Apply POS tagging:
   - The output for the first sentence is: “yes/UH” “,/,” ”I/PRP” “got/VBD” “your/PRP$” “invitation/NN” “,/,”
   - The output for the second sentence is: “I/PRP” “will/MD” “happily/RB” “attend/VB” “your/PRP$” “party/NN”

IBM Training

Natural language processing pipeline (cont.)

4. **Morphological processing:** Defines how morphemes are constructed.

- For example, “happily” is broken into happy –ly.
- Word meanings change when you add or remove any affixes. For example, certain and uncertain, or late and latest.
- Nouns can change to adjectives, for example, sadness to sad.
- Two words merge to form a new one, for example, “football”, divided into two tokens “foot” and “ball”.

Figure 6-19. Natural language processing pipeline (cont.)

Morphological parsing is the process of determining the morphemes of a word.

Morphemes represent the smallest grammatical units in a language and are not identical to a word. The main difference between a morpheme and word is that a morpheme might or not stand alone, and a word is freestanding.

Morphology focus on recognizing how base words were modified to form other words with similar meanings but often with different syntactic categories.

In this step, you break strings of language into sets of tokens that correspond to discrete words. For the example, a word token, such as “happily”, can be broken into two subword tokens as happy –ly. This stage does not need to be performed in our example.

Word meanings change when you add or remove any affixes. For example, certain and uncertain, or late and latest. Nouns can change to adjectives, for example, sadness to sad.

Morphology applies when two words merge to form a new one, for example, “football”, where it can be divided into two tokens “foot” and “ball” to help understand the word meaning.

POS structures the tokens, morphology adjusts the meaning of each token. By changing the language that is used, the rules also change.

References:

https://en.wikipedia.org/wiki/Morphological_parsing

https://en.wikipedia.org/wiki/Morpheme

IBM Training

Natural language processing pipeline (cont.)

5. Word-level (lexical) semantics:

- Deals with the meaning of words.
  - Example: “yes/UH” “,/,” ”I/PRP” “got/VBD” “your/PRP$” “invitation/NN” “,/,”.
- Can replace the “got” token with “received”.
  - Example: “yes/UH” “,/,” ”I/PRP” “received/VBD” “your/PRP$” “invitation/NN” “,/,”.

Natural language processing concepts and components

Figure 6-20. Natural language processing pipeline (cont.)

5) Word-level (lexical) semantic: Semantics deal with the meaning of words. In the example, we have the following input for this stage:

1+1=2 **Example**

“yes/UH” “,/,” ”I/PRP” “got/VBD” “your/PRP$” “invitation/NN” “,/,”

The “got” token in the first sentence means “received”, so this token may be replaced with a more appropriate token if necessary. So, the output would be:

“yes/UH” “,/,” ”I/PRP” “received/VBD” “your/PRP$” “invitation/NN” “,/,”

IBM Training

Natural language processing pipeline (cont.)

6. Parsing:

- Evaluate in terms of grammatical correctness.
- The input from the previous stage was:
  - First sentence: “yes/UH” “,/,” ”I/PRP” “received/VBD” “your/PRP$” “invitation/NN” “,/,”
  - Second sentence: “I/PRP” “will/MD” “happily/RB” “attend/VB” “your/PRP$” “party/NN”

Natural language processing concepts and components

Figure 6-21. Natural language processing pipeline (cont.)

Parsing or syntactic analysis is the process of evaluating text in terms of grammatical correctness (if required).

The input from the previous stage was:

- First sentence: “yes/UH” “,/,” ”I/PRP” “received/VBD” “your/PRP$” “invitation/NN” “,/,”
- Second sentence: “I/PRP” “will/MD” “happily/RB” “attend/VB” “your/PRP$” “party/NN”

Reference:

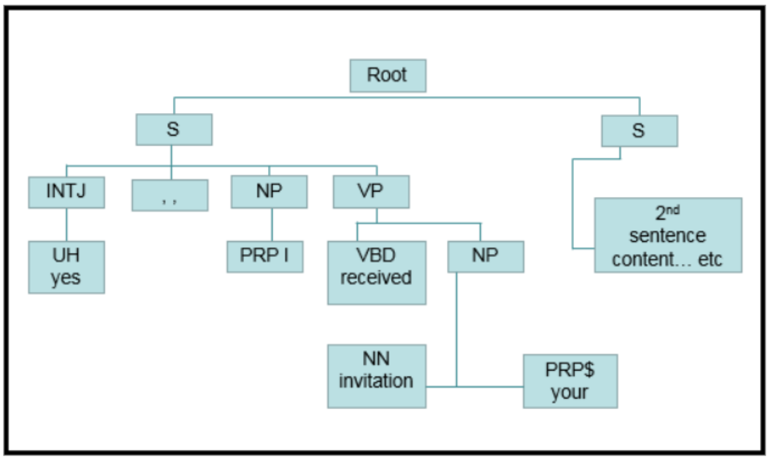
https://hpi.de/fileadmin/user_upload/fachgebiete/plattner/teaching/NaturalLanguageProcessing/NLP2016/NLP05_SyntacticParsing.pdf

IBM Training

Natural language processing pipeline (cont.)

Parsing (cont.)

(ROOT (S (S (INTJ (UH yes)) (, ,) (NP (PRP I)) (VP (VBD got) (NP (PRP$ your) (NN invitation)))) (, ,) (CC and) (S (NP (PRP I)) (VP (MD will) (ADVP (RB happily)) (VP (VB attend) (NP (PRP$ your) (NN party)))))))



Sentence graphical structure

Natural language processing concepts and components

Figure 6-22. Natural language processing pipeline (cont.)

One method for parsing is to group text into a tree structure. This technique is used in NLP tasks like machine translations, grammar checker, speech recognition, and relation extraction.

Reference:

http://disi.unitn.it/moschitti/Teaching-slides/NLP-IR/NLP-Parsing.pdf

6.4. Information retrieval

IBM Training

Information retrieval

Natural language processing concepts and components

Figure 6-23. Information retrieval

IBM Training

Topics

- Natural language processing categories
- Language ambiguities
- Natural language processing pipeline
- Information retrieval
- Information extraction

Natural language processing concepts and components

Figure 6-24. Topics

IBM Training

Information retrieval

- Retrieve relevant information from a collection of information resources.
- Information retrieval is the foundation of many search engines.
- Upcoming tasks:
  - Stemming
  - Normalization
  - Term Frequency – Inverse Document Frequency (TF-IDF)

Natural language processing concepts and components

Figure 6-25. Information retrieval

The objective is to obtain relevant information from a collection of information resources.

Information retrieval is the foundation of many search engines.

Next, we will discuss a few tasks that are used in information retrieval:

- Stemming
- Normalization
- TF-IDF

Reference: https://nlp.stanford.edu/IR-book/html/htmledition/irbook.html

IBM Training

Stemming

- Reduce a word to its word stem by removing its affixes.
- For example, the word “unbelievable” can be stemmed into “believ” by removing the prefix “un” and the suffix “able”.
- The stem does not have to match the morphological root of the word. For example, “Chance” stemmed to “Chanc”.
- The most popular English stemming algorithm is the Porter's algorithm (Porter, 1980).

Natural language processing concepts and components

Figure 6-26. Stemming

Stemming is a task where algorithms reduce a word to its word stem by removing its affixes. For example, the word “unbelievable” may be stemmed into “believ” by removing the prefix “un” and the suffix “able”.

The stem does not have to match the morphological root of the word. Therefore, the stem form can have no actual meaning, for example, the stem of the word "chance" is "chanc“, which is not an English word on its own. The most popular English stemming algorithm is the Porter's algorithm (Porter, 1980).

IBM Training

Porter stemmer

Step Number	Rule	Example
1	sses → ss ies → I ss → ss s → ø	gla**sses** → gla**ss** part**ies** → parti loss → loss hats → hat
2	ing → ø ed → ø	talking → talk discovered → discover
3	ational → ate izer → ize ator → ate	operational → operate recognizer → recognize collaborator → collaborate
4	al → ø able → ø ate → ø	electrical → electric doable → do investigate → investing

Natural language processing concepts and components

Figure 6-27. Porter stemmer

This table demonstrates some of the rules of the Porter stemmer.

IBM Training

Normalization

- Used in many NLP problems, such as sentiment analysis and information retrieval.
- Examples:
  - Case folding: Child → child
  - Duplication removal: Hiiiiii → Hi
  - Acronyms processing: WHO → World Health Organization
  - Format normalization: $100 → 100 dollars
  - Value normalization: 2 July 1980 → DATE

Natural language processing concepts and components

Figure 6-28. Normalization

Text normalization is the process of transforming text into a single form, which ensures consistency before operations are performed on it.

For example, imagine that you have a dictionary that contains a set of words, and you also have query text that includes one of the words in your dictionary. Assume that the word in the query is **C**hild with a capital letter C. But the equivalent word in the dictionary is *c*hild with a lowercase c letter. Thus, both words do not match.

The solution is to ensure that the query text is "normalized" into lowercase and the set of words in dictionary is also normalized into lowercase so that the query and dictionary have consistent text.

Here are some examples of normalization:

Examples:

- Case folding: Child → child
- Duplication removal: Hiiiiii → Hi
- Acronyms processing: WHO → World Health Organization
- Format normalization: $100 → 100 dollars
- Value normalization: 2 July 1980 → DATE

There are no strict rules for how normalization should be done. It depends on what type of text will be normalized and how it will be processed later.

IBM Training

Term Frequency – Inverse Document Frequency

- TF-IDF is a combination of two weighting methods for information retrieval to determine how important a term is.
- **TF**: Term Frequency measures how many times a term t occurs in a document d. This value is denoted by $\mathbf{tf}_{t,d}$.
- **IDF**: Inverse Document Frequency measures how rare a term is. You must decrease the frequent terms weight while increasing the weight of the exceptional ones.

Natural language processing concepts and components

Figure 6-29. Term Frequency – Inverse Document Frequency

The weighting methods in information retrieval determine how important a term is:

TF: Term Frequency measures how many times a term t occurs in a document **d**. This value is denoted by $\mathbf{tf}_{t,d}$.

IDF: Inverse Document Frequency measures how rare a term is.

Reference:

https://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html

IBM Training

Term Frequency – Inverse Document Frequency (cont.)

- **idf**= log (Total number of documents (**N)** / Number of documents with term ***t*** (**df$_t$**))

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}$$

- To generate a composite weight, term frequency and inverse document frequency are multiplied as follows:

$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$$

Natural language processing concepts and components

Figure 6-30. Term Frequency – Inverse Document Frequency (cont.)

While computing TF, all terms are considered equally important. However, certain terms, such as "is", "of", and "that", usually appear in any document thus these terms are not actually important. These non-important words are called *stop words*. Thus, we need to decrease the frequent terms weight while increasing the weight of the exceptional ones. This is done through the **idf**$_t$ equation shown in the slide.

6.5. Information extraction

IBM Training

Information extraction

Natural language processing concepts and components

Figure 6-31. Information extraction

IBM Training

Topics

- Natural language processing categories
- Language ambiguities
- Natural language processing pipeline
- Information retrieval
- Information extraction

Figure 6-32. Topics

IBM Training

Information extraction

- Refers to the automatic extraction of structured information from unstructured or semi-structured text.
- Examples of applications where information extraction is used:
  - Entities and relation identification
  - Sentiment analysis

Natural language processing concepts and components

Figure 6-33. Information extraction

Information extraction refers to the automatic extraction of structured information from unstructured or semi-structured text.

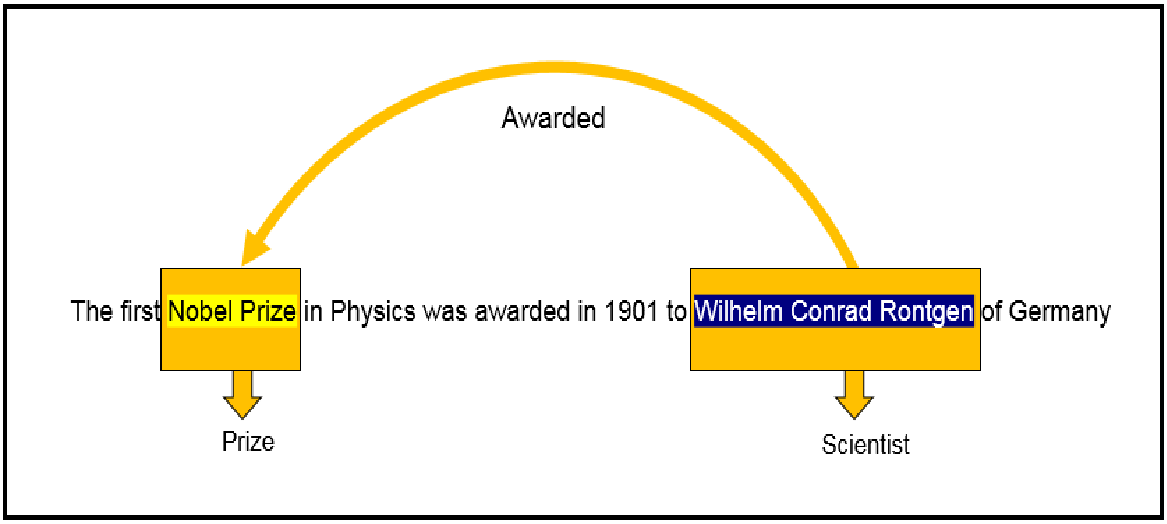
For example, information extraction is used in the following applications:

- Entities and relation identification
- Sentiment analysis

IBM Training

Information extraction (cont.)

Entities and relation identification



Natural language processing concepts and components

Figure 6-34. Information extraction (cont.)

As shown in this slide, “Nobel Prize” and “Wilhelm Conrad Rontgen” are identified as entities. The relationship is identified as awarded, which indicates the relationship between Wilhelm and the Nobel Prize.

IBM Training

Information extraction (cont.)

Sentiment analysis:

- Detects user’s opinions in text.
- Used in the field of chatbots, social media analysis, and others.
- Examples:
  - “Their service is amazing” is a positive sentiment.
  - “The quality of food in this restaurant is terrible” is a negative sentiment.
  - “I am going to school” is a neutral sentiment.

Natural language processing concepts and components

Figure 6-35. Information extraction (cont.)

Sentiment analysis is the process of identifying emotions or opinions that are expressed in user input. It is used heavily in the fields of chatbots and social media analysis. It is also used in marketing because it captures a user’s opinion about a particular product or service so that an organization can take corrective action to keep the user satisfied.

Examples:

- “Their service is amazing” is a positive sentiment.
- “The quality of food in this restaurant is terrible” is a negative sentiment.
- “I am going to school” is a neutral sentiment.

IBM Training

Unit summary

- Define the NLP categories.
- Describe the NLP pipeline.
- Explain the challenges in natural language understanding (NLU).
- Explain the concepts of information retrieval and extraction.
- Describe sentiment analysis.

Natural language processing concepts and components

Figure 6-36. Unit summary

IBM Training

Review questions

1. The technique that is used to unify all date formats in your data set into one common format is called:
   a. Stemming
   b. Tokenization
   c. Normalization

2. What is the process of labeling tokens with their grammatical label?
   a. Tokenization
   b. POS tagging
   c. Lexical analysis
   d. Parsing

Natural language processing concepts and components

Figure 6-37. Review questions

Normalization

1. POS tagging

IBM Training

Review questions (cont.)

3. True or False: For any NLP task, you must apply every step in the NLP pipeline.
4. True or False: Sentiment analysis is an NLU application.

Natural language processing concepts and components

Figure 6-38. Review questions (cont.)

2. False
3. True

IBM Training

Review answers

1. The technique that is used to unify all date formats in your data set into one common format is called:
   A. Stemming
   B. Tokenization
   C. Normalization
2. What is the process of labeling tokens with their grammatical label?
   A. Tokenization
   B. POS tagging
   C. Lexical analysis
   D. Parsing

Natural language processing concepts and components

Figure 6-39. Review answers

1. Normalization
2. POS tagging

IBM Training

Review answers (cont.)

3. True or False: For any NLP task, you must apply every step in the NLP pipeline.
4. True or False: Sentiment analysis is an NLU application.

Figure 6-40. Review answers (cont.)

3. False
4. True

Unit 7. Natural language processing evaluation metrics

Estimated time

00:30

Overview

This unit explains how to evaluate the quality of your natural language processing (NLP) algorithm.

IBM Training

Unit objectives

- Define various metrics to measure the quality of NLP algorithms.
- Understand the difference between these metrics.

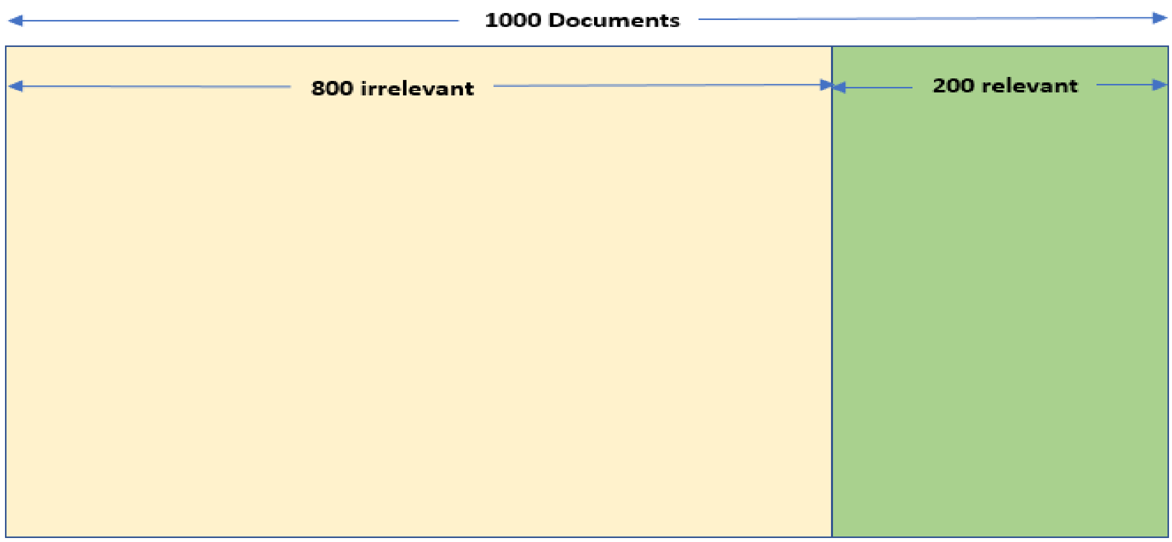
Natural language processing evaluation metrics

Figure 7-1. Unit objectives

IBM Training

System evaluation

- How can we measure the solution quality?
- Target: You developed a new search engine. You must define how well it works.



Natural language processing evaluation metrics

Figure 7-2. System evaluation

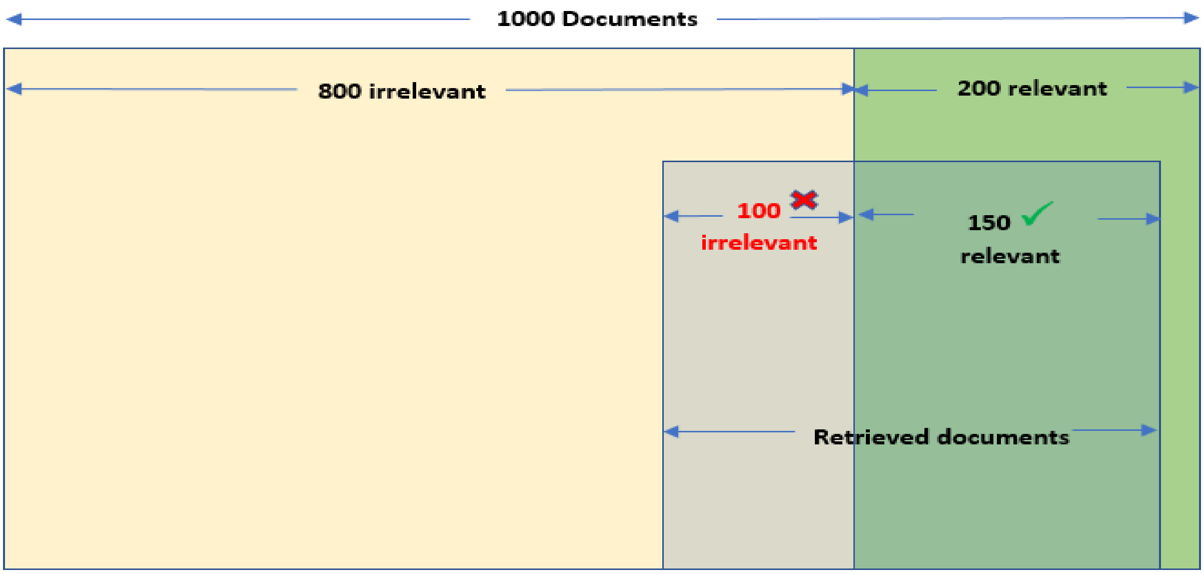
How can we measure the solution quality? In this presentation, we focus on a basic metric to evaluate system performance in information retrieval. Assume that you developed a search algorithm that helps you to retrieve related words from a corpus that contains 1000 documents.

From these 1000 documents, assume 200 are relevant to the word cat, and the other 800 documents are irrelevant.

IBM Training

System evaluation (cont.)

- You ran a search test for the word “cat”.
- After the test ran, the search engine retrieved the documents that are shown here.



Natural language processing evaluation metrics

Figure 7-3. System evaluation (cont.)

You test your solution by searching for the word “cat”. Your algorithm returns 250 documents, where 150 documents are relevant (which means your algorithm missed 50 relevant documents) and 100 documents are irrelevant (which means your algorithm correctly eliminated 700 of the irrelevant documents).

IBM Training

System evaluation (cont.)

Confusion matrix

		Relevant documents in results set	Irrelevant documents in results set
Algorithm results	Retrieved	True positive (Tp)	False positive (Fp)
	Not retrieved	False negative (Fn)	True negative (Tn)

- How many **relevant** documents were **retrieved** by the algorithm? 150 documents → True positive (Tp).
- How many **irrelevant** documents were **retrieved** by the algorithm? 100 documents → False positive (Fp) (total 250 documents retrieved – 150 relevant documents).
- How many **relevant** documents did the algorithm **not retrieve**? 50 documents → False negative (Fn).
- How many **irrelevant** documents did the algorithm **not retrieve**? 700 documents → True negative (Tn).

Natural language processing evaluation metrics

Figure 7-4. System evaluation (cont.)

A confusion matrix, also known as an error matrix, is a specific table layout that enables visualization of the performance of an algorithm.

Based on the results for this example, how many **relevant** documents were **retrieved** by the algorithm? The answer is 150 documents. This value is the True positive (Tp).

Based on the results for this example, how many **irrelevant** documents were **retrieved** by the algorithm? The answer is 100 documents (total 250 documents retrieved – 150 relevant documents). This value is the False positive (Fp).

Based on the results for this example, how many **relevant** documents did the algorithm **not retrieve**? The answer is 50 documents. This value is the False negative (Fn).

Based on the results for this example, how many **irrelevant** documents did the algorithm **not retrieve**? The answer is 700 documents. This value is the True negative (Tn).

The objective is to improve the algorithm to decrease the Fp and Fn values.

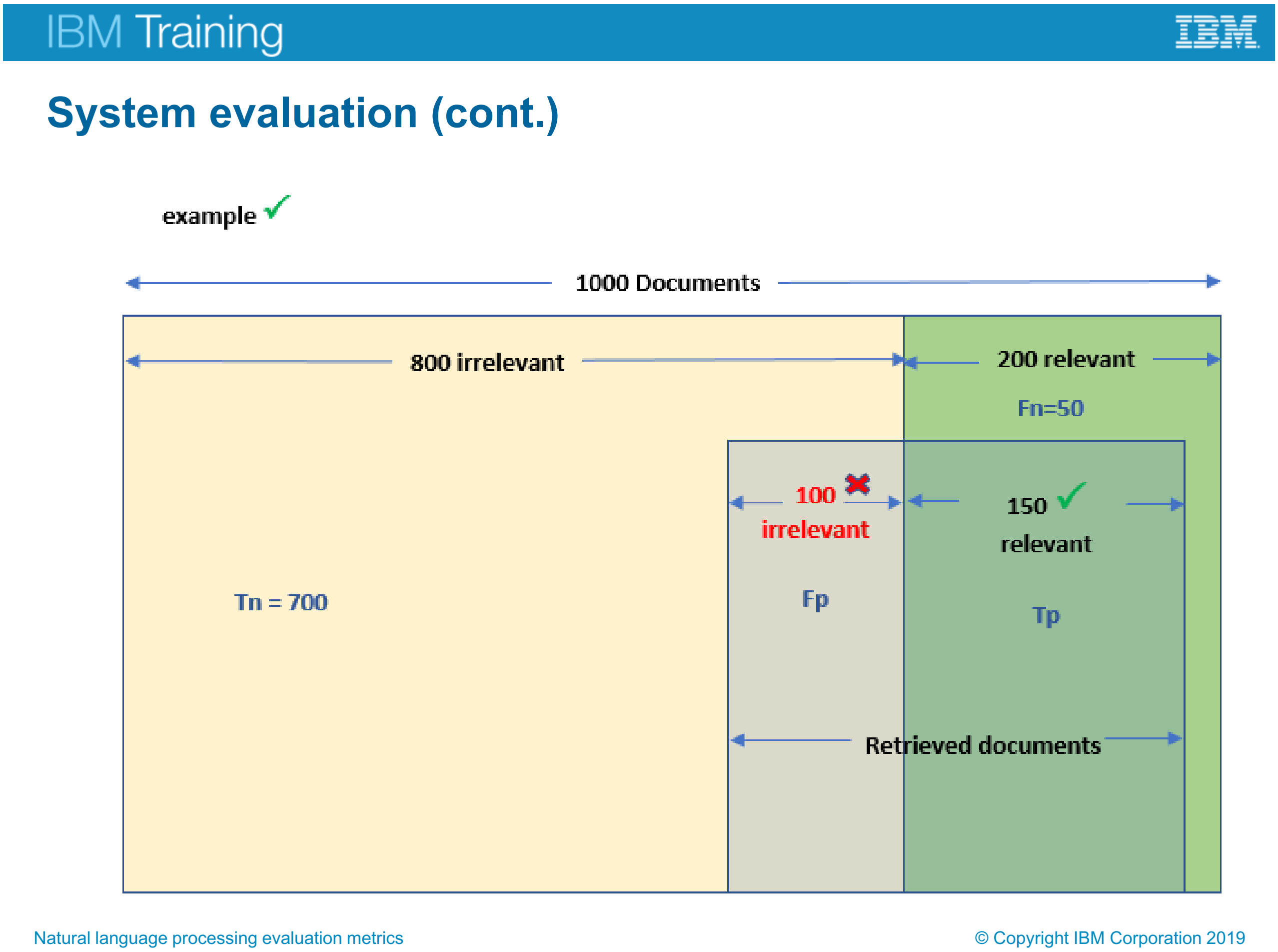


Figure 7-5. System evaluation (cont.)

We map the confusion matrix to the graph to produce the visuals for Tp, Fp, Tn, and Fn. We add Tp to the retrieved relevant documents area and Fp to the retrieved irrelevant area. We add Fn to the not retrieved relevant area and Tn to the not retrieved irrelevant area.

IBM Training

System evaluation (cont.)

Accuracy

- Calculates how many correct results your solution managed to identify.

```
Accuracy = (Tp+Tn)/(Tp+Tn+Fp+Fn)
```

- Apply the formula to the example.

```
Accuracy = (150+700)/(1000)
```

- Useful for symmetric data sets where the values of positive and negatives are almost the same.

Natural language processing evaluation metrics

Figure 7-6. System evaluation (cont.)

Accuracy is as a numeric measure of how good your algorithm is. It calculates how many correct results your solution managed to identify, which is the proportion of true results among the total number of cases that are examined.

Accuracy is defined by the following formula, which includes the Tp, Tn, Fp, and Fn metrics:

Accuracy = (Tp+Tn)/(Tp+Tn+Fp+Fn)

By applying the values from the example, accuracy can be calculated as follows:

Accuracy = (150+700)/1000 = 0.85

Accuracy is a good measure but only when you have symmetric data sets where the number of positive values and negatives values are almost the same. For example, if your data set is split as 90 positive samples and 10 negative samples, classifying all as positive gives a 0.90 accuracy score.

Therefore, we must look at other metrics such as precision and recall to evaluate the quality of the algorithm.

IBM Training

System evaluation (cont.)

Precision

- Represents the fraction of retrieved documents that are relevant.

```
Precision = Tp/(Tp+Fp)
```

- Apply the formula to the example.

```
Precision = 150/(150+100)
Precision = 150/250 = 0.60
```

Natural language processing evaluation metrics

Figure 7-7. System evaluation (cont.)

Precision is a numeric measure that represents the fraction of retrieved documents that are relevant. It is defined by the following formula:

Precision = Tp/(Tp+Fp)

Apply the formula to the example:

Precision = 150/(150+100)

150 retrieved relevant / 250 total retrieved = 0.60

IBM Training

System evaluation (cont.)

Recall

- Represents the fraction of relevant documents that were retrieved.

```
Recall = Tp/(Tp+Fn)
```

- Apply the formula to the following example.

```
Recall = (150)/(150+50)
Recall = 150/200 = 0.75
```

Natural language processing evaluation metrics

Figure 7-8. System evaluation (cont.)

Recall is a numeric measure that represents the fraction of relevant documents that were retrieved. It is defined by the following formula:

Recall = Tp/(Tp+Fn)

Apply the formula to the example:

Recall =150/(150+50)

150 retrieved relevant / 200 total relevant = 0.75

IBM Training

System evaluation (cont.)

F-Score (F-measure**)**

- Enables you to tradeoff precision against recall.
- The higher the F-score value is, the better the algorithm is.
- Here is the formula.

```
F = 2*Precision*Recall/(Precision+Recall)
```

- Apply the formula to the example.

```
F = (2*0.60*0.75)/(0.60+0.75)
F = 0.9/1.35= 0.6667
```

Natural language processing evaluation metrics

Figure 7-9. System evaluation (cont.)

The **F-score** (also called F-measure) is a measure that enables you to tradeoff precision against recall by approximately averaging the precision and recall values.

The formula for F-score is:

F = (2*Precision*Recall)/(Precision +Recall)

Apply the formula to the example:

F = (2*0.60*0.75)/(0.60+0.75)

= 0.9/1.35= 0.6667

IBM Training

Unit summary

- Define various metrics to measure the quality of NLP algorithms.
- Understand the difference between these metrics.

Natural language processing evaluation metrics

Figure 7-10. Unit summary

IBM Training

Review questions

1. True or False: To calculate the F-score, you must calculate accuracy first.
2. True or False: Accuracy is the harmonic mean of precision and recall.
3. Assume that you developed an algorithm and calculated the Precision as 0.5 and the Recall as 0.5, what is the value of F-score in this case?
   a. 0.25
   b. 1.00
   c. 0.50

Natural language processing evaluation metrics

Figure 7-11. Review questions

1. False
2. False
3. C: 0.50

IBM Training

Review questions (cont.)

4. The F-score is calculated as *(hint P=precision, R= Recall)*:
   a. 2*(P+R)/(P+R)
   b. 2*(P*R)/(P+R)
   c. (P*R)/2*(P+R)
   d. (P+R)/2*(P*R)

Natural language processing evaluation metrics

Figure 7-12. Review questions (cont.)

4. B: 2*(P*R)/(P+R)

IBM Training

Review answers

1. True or False: To calculate the F-score, you must calculate accuracy first.
2. True or False: Accuracy is the harmonic mean of precision and recall.
3. Assume that you developed an algorithm and calculated the Precision as 0.5 and the Recall as 0.5. What is the value of F-score in this case?
   A. 0.25
   B. 1.00
   C. 0.50

Natural language processing evaluation metrics

Figure 7-13. Review answers

1. False
2. False
3. C: 0.50

IBM Training

Review answers (cont.)

4. F-score is calculated as *(hint P=precision, R= Recall)*:
   - A. 2*(P+R)/(P+R)
   - B. 2*(P*R)/(P+R)
   - C. (P*R)/2*(P+R)
   - D. (P+R)/2*(P*R)

Natural language processing evaluation metrics

Figure 7-14. Review answers (cont.)

4. B: 2*(P*R)/(P+R)

Unit 8. Natural language processing and IBM Watson

Estimated time

00:30

Overview

This unit lists the Watson services and software that are based on natural language processing (NLP) and explains the main capabilities of IBM Watson Natural Language Classifier, IBM Watson Natural Language Understanding, and IBM Watson Discovery.

IBM Training

Unit objectives

- List the NLP Watson services.
- List the Watson services that perform information extraction.
- Describe the capabilities of IBM Watson Natural Language Classifier.
- Describe the capabilities of the IBM Watson Natural Language Understanding.
- Describe the capabilities of IBM Watson Discovery.

Natural language processing and IBM Watson

Figure 8-1. Unit objectives

8.1. Watson NLP services

IBM Training

Watson NLP services

Natural language processing and IBM Watson

Figure 8-2. Watson NLP services

IBM Training

Topics

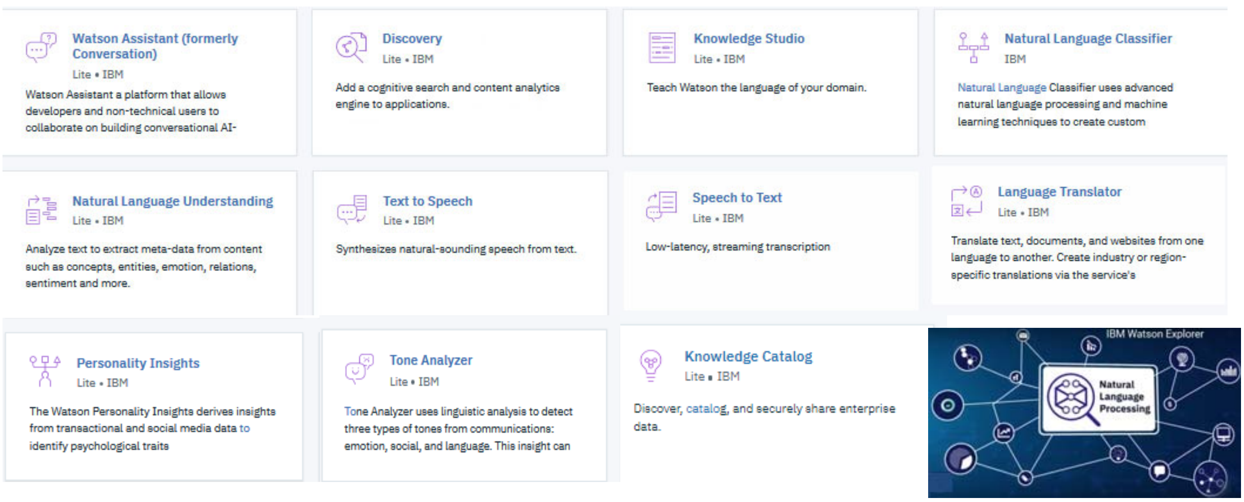
- Watson NLP services
- Watson Natural Language Classifier
- Watson Natural Language Understanding
- Watson Discovery

Natural language processing and IBM Watson

Figure 8-3. Topics

IBM Training

Watson NLP services



Natural language processing and IBM Watson

Figure 8-4. Watson NLP services

Many Watson services have NLP at their core. The slide shows all the Watson NLP services on IBM Cloud and IBM Watson Explorer which is an NLP application but it is not available on IBM Cloud. For more information about IBM Watson Explorer, see https://www.ibm.com/products/watson-explorer.

The remainder of this presentation focuses on three Watson NLP services:

- Watson Natural Language Classifier
- Watson Natural Language Understanding
- Watson Discovery

Other presentations in this course cover Watson Assistant and Watson Knowledge Studio in detail.

IBM Training

Information extraction by using Watson services

- Watson NLP services that extract information from unstructured text:
  - Watson Natural Language Understanding
  - Watson Discovery
  - Watson Tone Analyzer
  - Watson Personality Insights

Natural language processing and IBM Watson

Figure 8-5. Information extraction by using Watson services

The Watson Natural Language Understanding, Watson Discovery, Watson Tone Analyzer, and Watson Personality Insights services are based on natural language processing (NLP) technology to extract information from unstructured text.

Unlike Watson Natural Language Classifier, these services can be used without training by the user because they include a built-in public model, which means they are pre-trained on a public training data set and ready to use immediately. However, to improve the accuracy of the service for a given domain, NLU and Discovery should also be trained.

8.2. Watson Natural Language Classifier

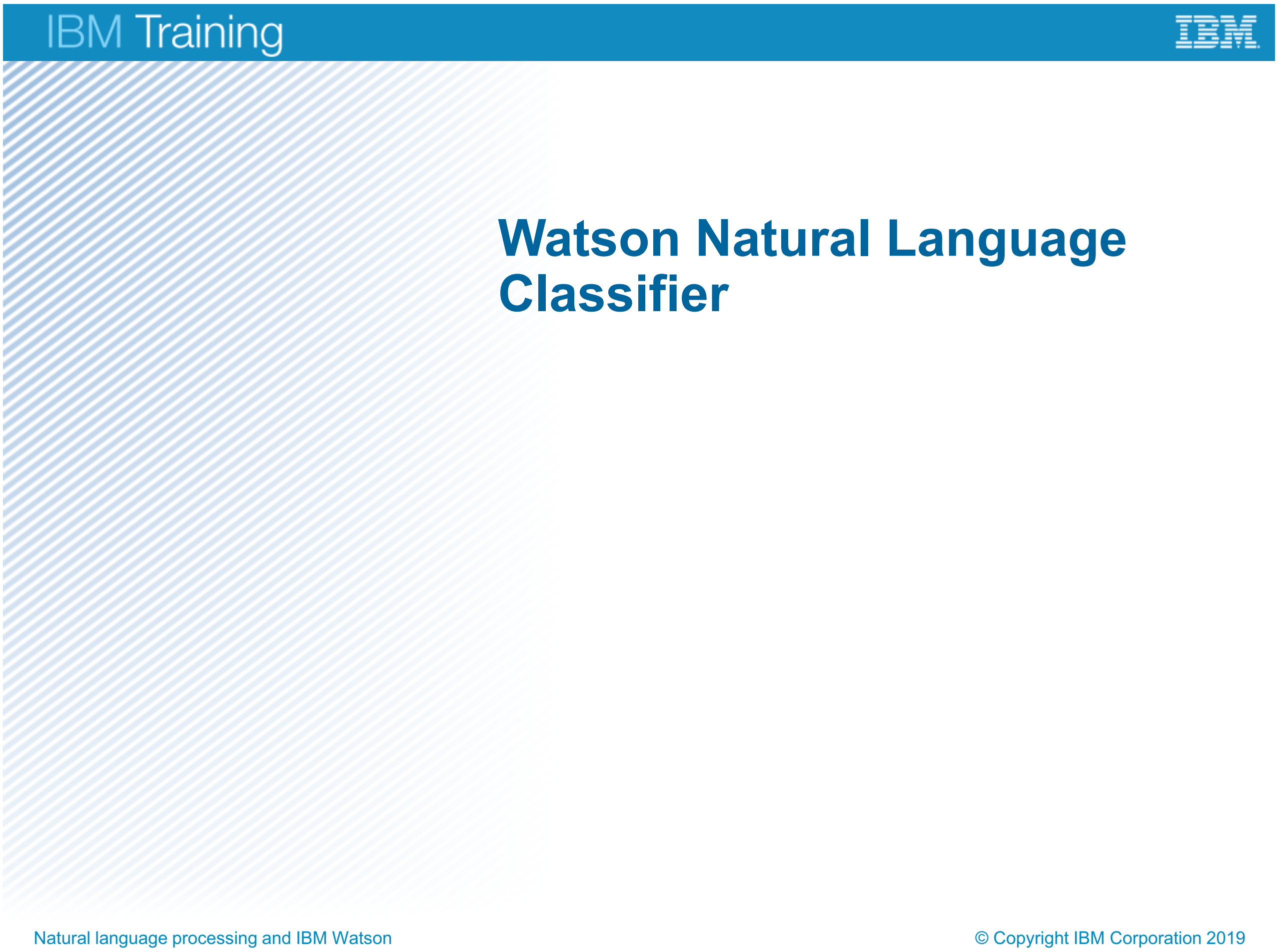


Figure 8-6. Watson Natural Language Classifier

IBM Training

Topics

- Watson NLP services
- Watson Natural Language Classifier
- Watson Natural Language Understanding
- Watson Discovery

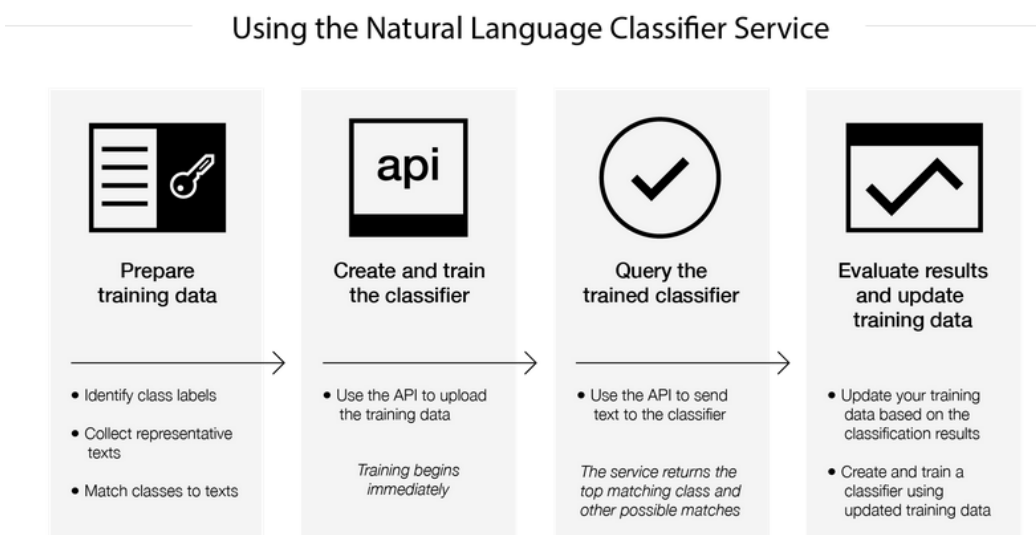
Natural language processing and IBM Watson

Figure 8-7. Topics

IBM Training

Watson Natural Language Classifier

- Classify natural language phrases into categories or classes.
- The following graphic shows the Watson NLC process.



Natural language processing and IBM Watson

Figure 8-8. Watson Natural Language Classifier

Watson Natural Language Classifier is a service on IBM Cloud. It mimics the human ability to classify naturally expressed phrases into categories or classes.

Watson NLC applies deep learning techniques to make predictions about the best predefined classes for short sentences or phrases. It helps your application understand the language of short texts and make predictions about how to handle them. It is useful for many use cases, for example, it can be used in banking and finance to classify investments, risks, and transactions.

The Watson NLC process has four main phases:

- **Training data preparation:** You identify classes labels and collect the representative text sample for each label.
- **Create and train the classifier:** Upload the training data to initiate training.
- **Query the trained classifier:** Test the classifier by sending text to classify.
- **Evaluate and update the training data:** Based on the test result, update the training data and train your classifier as needed.

A classifier learns from your example data and then can return information for texts on which it is not trained.

Reference:

https://console.bluemix.net/docs/services/natural-language-classifier/natural-language-classifier-overview.html#about

IBM Training

Watson Natural Language Classifier (cont.)

Training the comma-separated value data format

Will we have a cold day today?	temperature
When will the cold subside?	temperature
What highs are we expecting?	temperature
What lows are we expecting?	temperature
Is it warm?	temperature
Is it chilly?	temperature
What's the current temp in Celsius?	temperature
What is the temperature in Fahrenheit?	temperature
Is it windy?	conditions
Will it rain today?	conditions
What are the chances for rain?	conditions
Will we get snow?	conditions
Are we expecting sunny conditions?	conditions
Is it overcast?	conditions
Will it be cloudy?	conditions
How much rain will fall today?	conditions
How much snow are we expecting?	conditions
Is it windy outside?	conditions

Natural language processing and IBM Watson

Figure 8-9. Watson Natural Language Classifier (cont.)

The data that is used to train the Watson NLC is in comma-separated value (CSV) format. In the CSV format, a row in the file represents an example record. Each record has two or more columns:

- The first column is for sample input text that is used for training
- The second column is the category or label that matches the text.

As shown in the training set, multiple variations for each label were inserted. The more examples and variations that are added, the better the results that you can expect from the classifier.

Reference:

https://console.bluemix.net/docs/services/natural-language-classifier/using-your-data.html#data-preparation

IBM Training

Watson Natural Language Classifier (cont.)

- Example: *Is it hot outside?*
- The Watson NLC instance was trained on only two categories: *temperature* and *conditions*.

Natural language processing and IBM Watson

Figure 8-10. Watson Natural Language Classifier (cont.)

In the online Watson NLC demonstration, the Watson NLC service instance that was used was trained on only two categories: *temperature* and *conditions*. Therefore, based on the information that is entered by the user, the output is the category (temperature or conditions) and confidence score.

Notice in the output category *temperature* and confidence 100% were returned for our first input. ("Is it hot outside?"). Because this question was used for training, the classifier knows with certainty that it belongs to the temperature category.

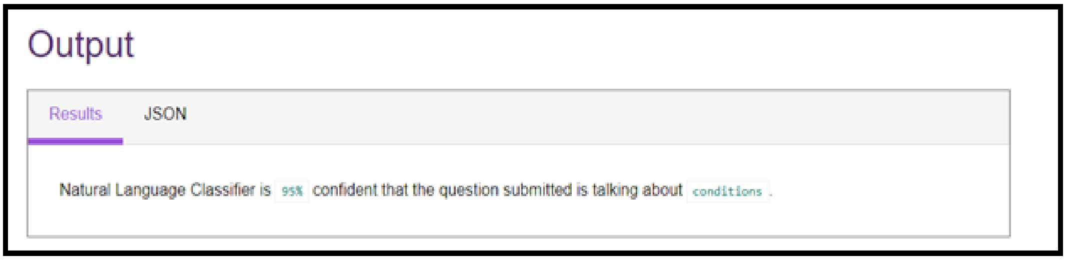
Reference:

https://natural-language-classifier-demo.ng.bluemix.net/

IBM Training

Watson Natural Language Classifier (cont.)

- The confidence level can change.
- Example: Will *it be foggy outside?* Watson NLC classification (conditions) and confidence score (95%).



Natural language processing and IBM Watson

Figure 8-11. Watson Natural Language Classifier (cont.)

When the sentence changes, the confidence varies depending on how similar the question is to the training data set.

8.3. Watson Natural Language Understanding

IBM Training

Watson Natural Language Understanding

Natural language processing and IBM Watson

Figure 8-12. Watson Natural Language Understanding

IBM Training

Topics

- Watson NLP services
- Watson Natural Language Classifier
- Watson Natural Language Understanding
- Watson Discovery

Figure 8-13. Topics

IBM Training

Watson Natural Language Understanding

- NLU extracts meaning from unstructured data.
- NLU processes input provided as text, HTML, or a public URL.
- NLU analyzes the input text and provides an output that includes:
  - Entities and relationships
  - Sentiment analysis
  - Keywords
- It can be trained by creating a custom model by using IBM Watson Knowledge Studio.

Natural language processing and IBM Watson

Figure 8-14. Watson Natural Language Understanding

Watson Natural Language Understanding (NLU) includes a set of text analytics features that you can use to extract meaning from unstructured data. NLU accepts user input and provides an output that includes entities and relationships that are found within the text and performs sentiment analysis of the input text.

Input can be in the form of text, raw HTML, or a public URL. The service cleans HTML content before analysis by default so the results can ignore most advertisements and other unwanted content.

You can create custom models by using Watson Knowledge Studio to detect custom entities and relations in NLU.

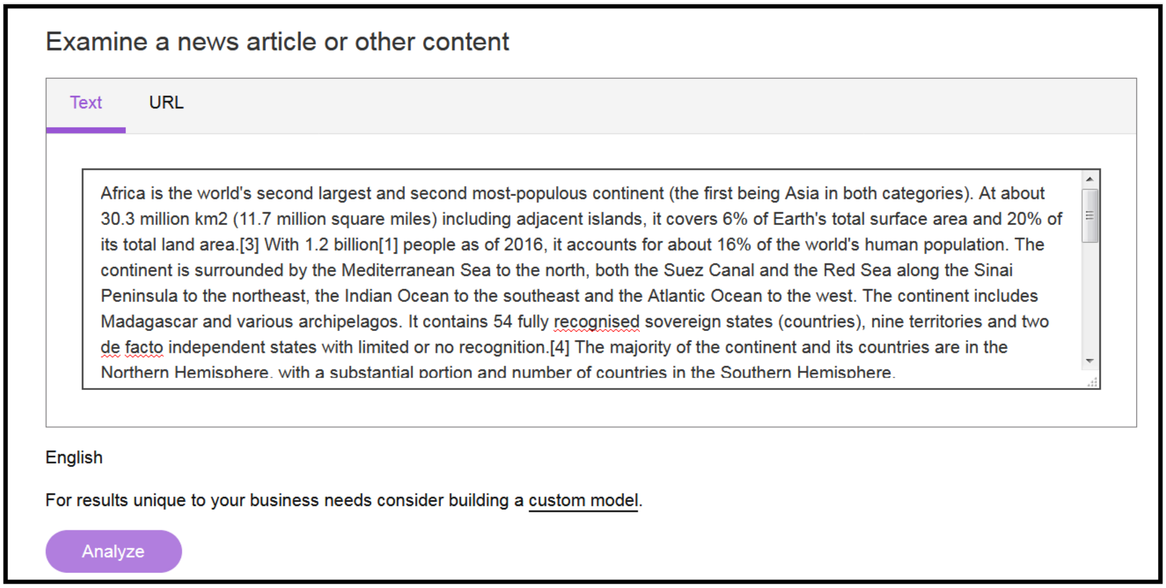
Reference:

https://console.bluemix.net/apidocs/natural-language-understanding#introduction

IBM Training

Watson Natural Language Understanding (cont.)

- Here is an example that uses the “Africa” article from Wikipedia.



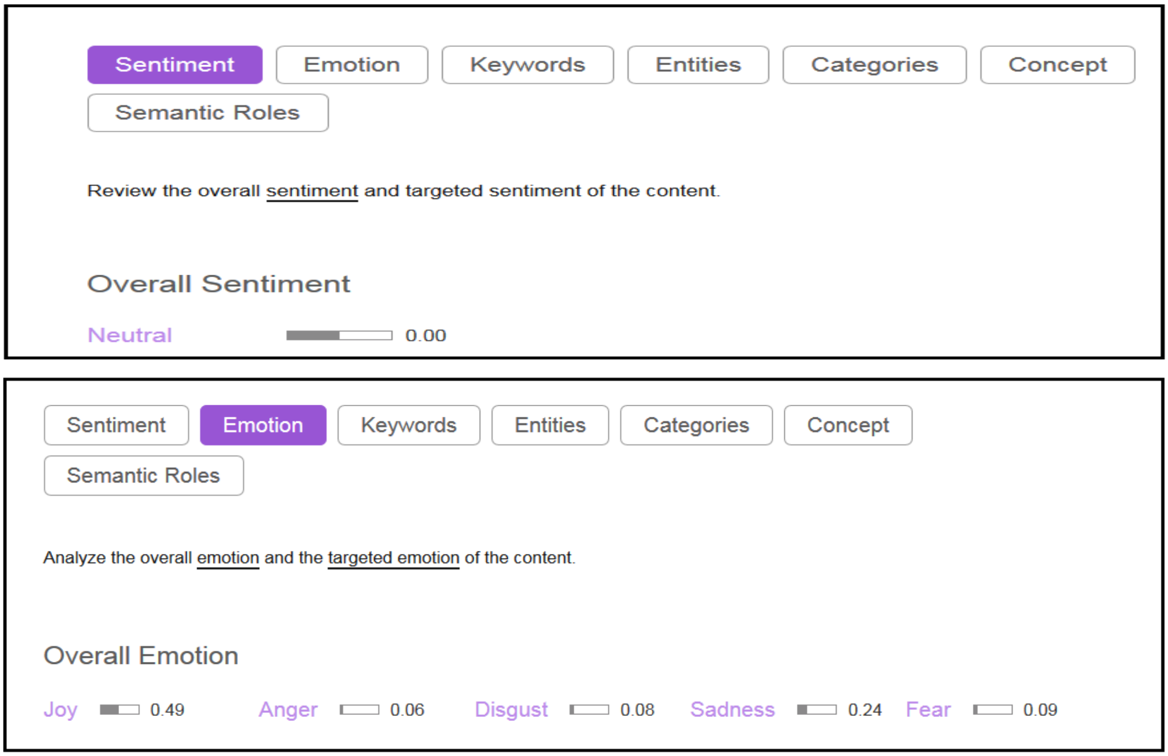
Natural language processing and IBM Watson

Figure 8-15. Watson Natural Language Understanding (cont.)

Using the Watson Natural Language Understanding service, we examine the “Africa” article from Wikipedia. We use the online demonstration to analyze and display the results.

IBM Training

Watson Natural Language Understanding (cont.)



Natural language processing and IBM Watson

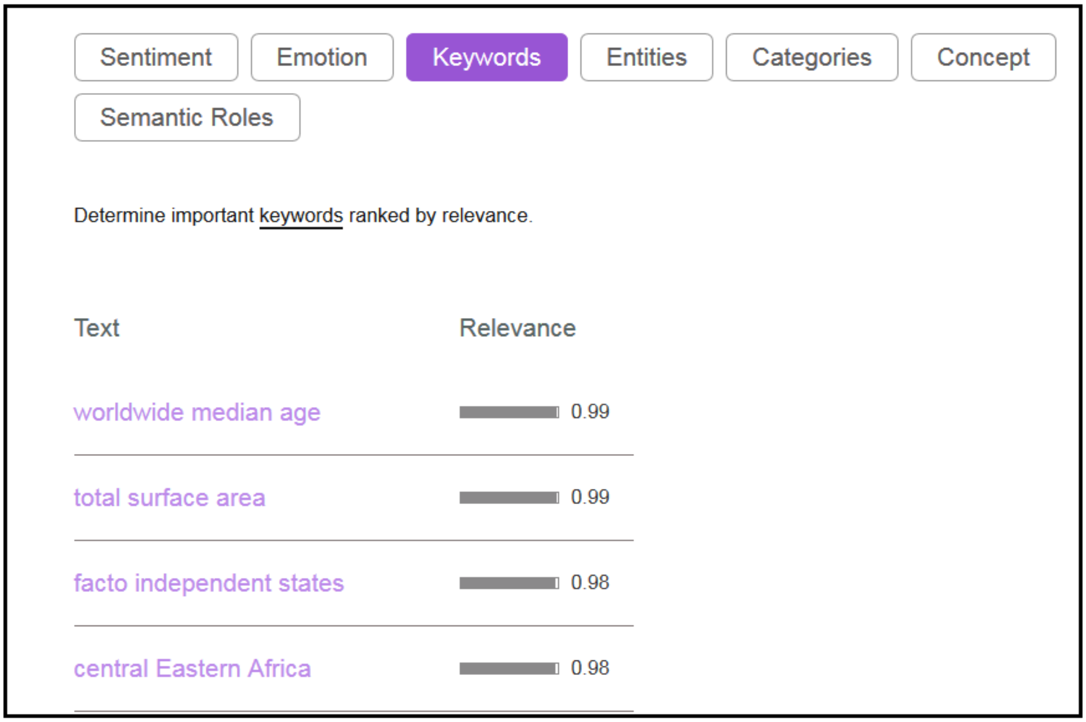
Figure 8-16. Watson Natural Language Understanding (cont.)

Choose different tabs to view the results:

- The Sentiment tab analyzes the general sentiment of the content or the sentiment toward specific target phrases.
- The Emotion tab detects anger, disgust, fear, joy, or sadness that is conveyed in the content or by the context around target phrases that are specified in the targets parameter.

IBM Training

Watson Natural Language Understanding (cont.)



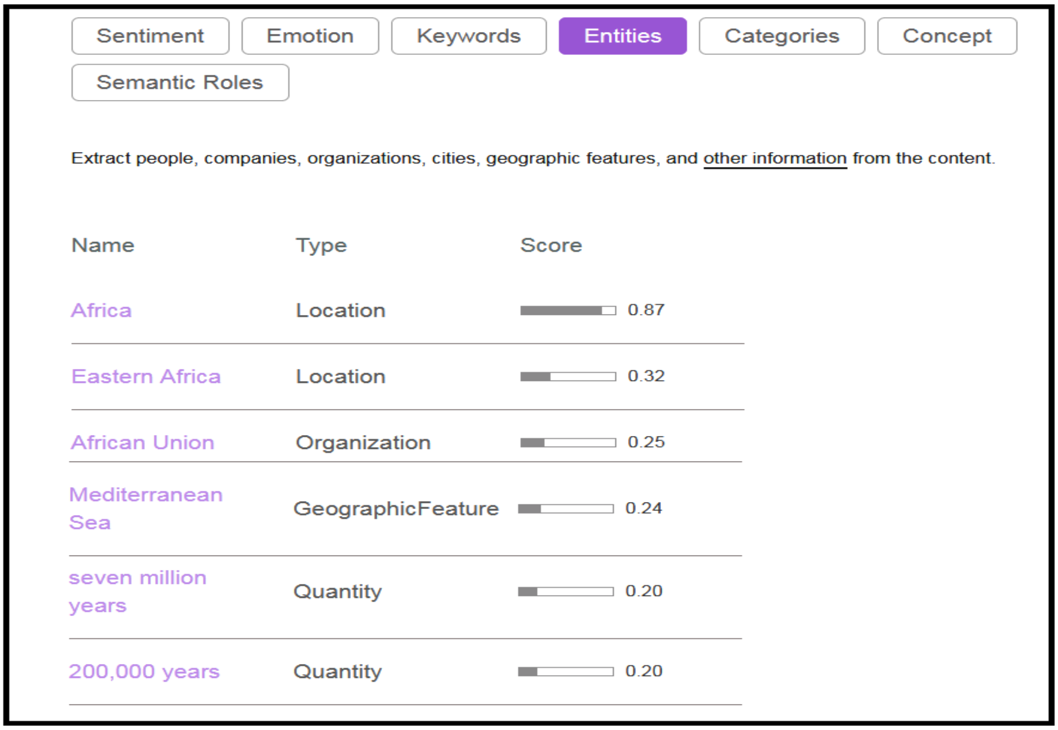
Natural language processing and IBM Watson

Figure 8-17. Watson Natural Language Understanding (cont.)

The Keywords tab returns important keywords in the content.

IBM Training

Watson Natural Language Understanding (cont.)



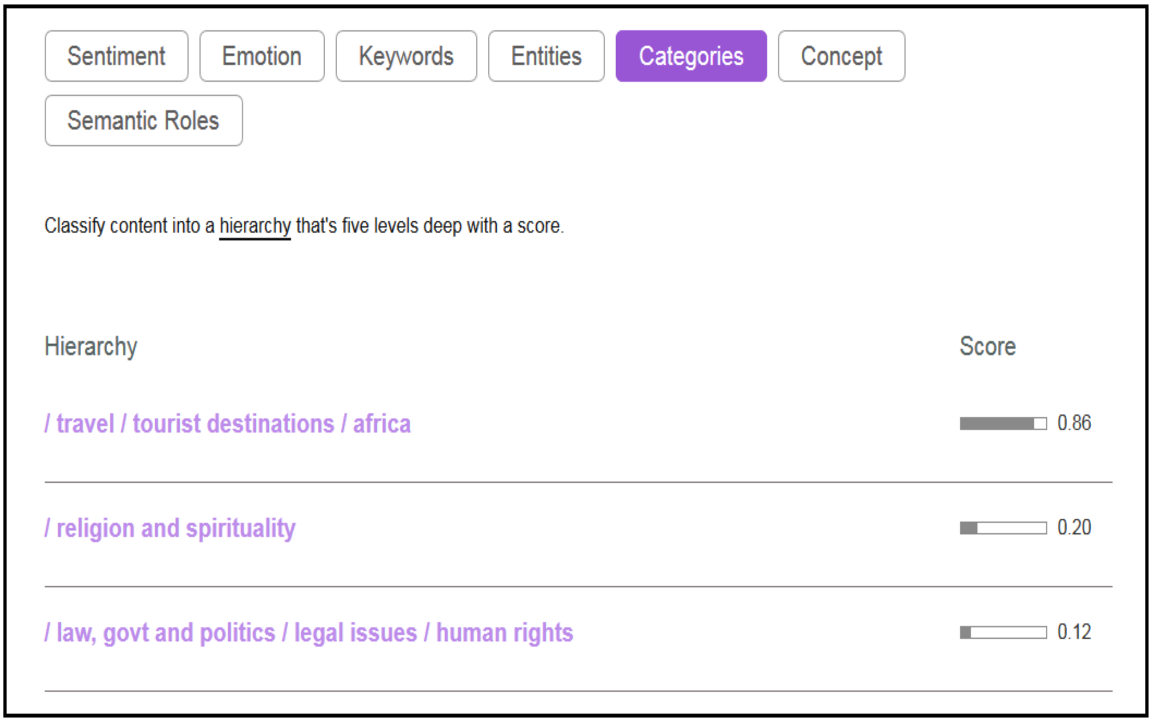
Natural language processing and IBM Watson

Figure 8-18. Watson Natural Language Understanding (cont.)

The Entities tab identifies people, locations, organizations, and other subjects.

IBM Training

Watson Natural Language Understanding (cont.)



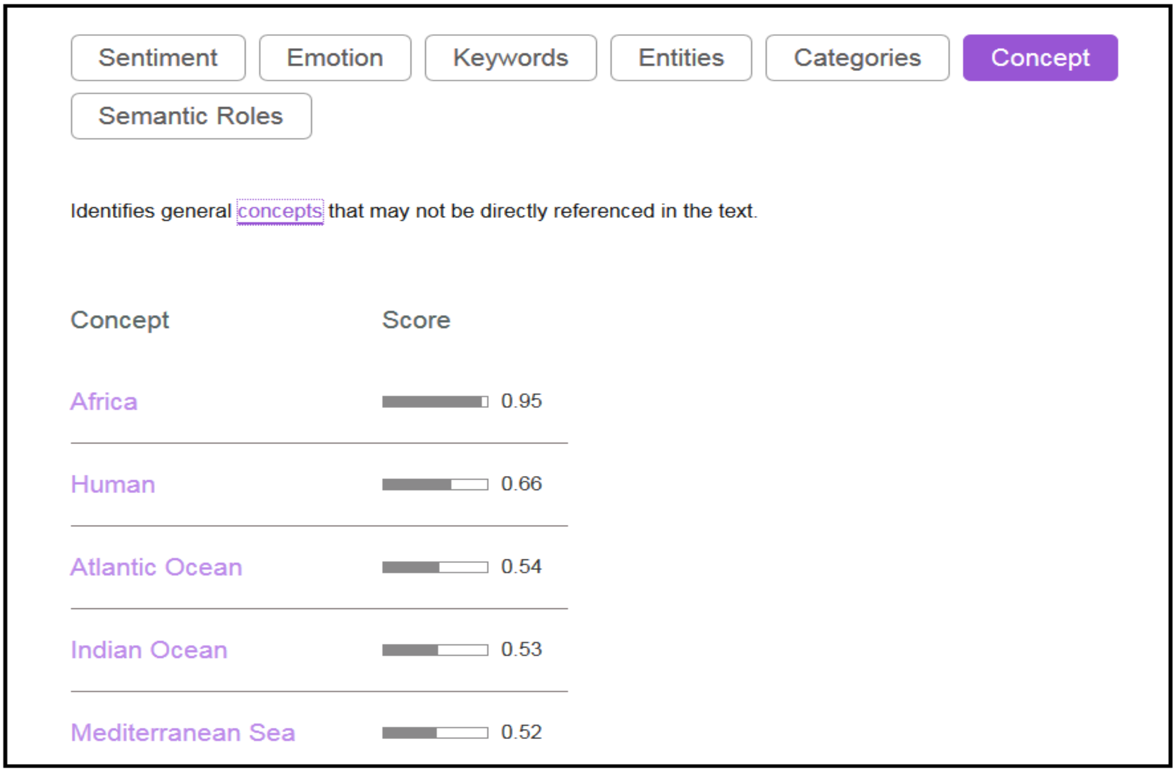
Natural language processing and IBM Watson

Figure 8-19. Watson Natural Language Understanding (cont.)

The Categories tab classifies and categorizes the content into a five-level taxonomy.

IBM Training

Watson Natural Language Understanding (cont.)



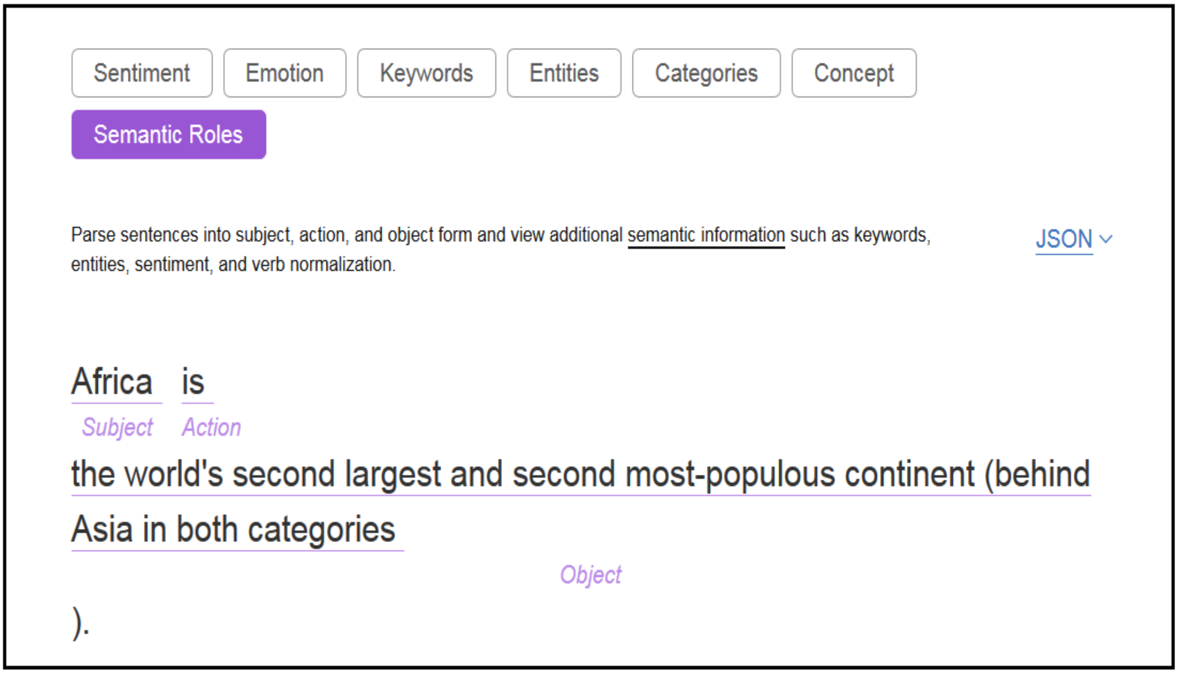
Natural language processing and IBM Watson

Figure 8-20. Watson Natural Language Understanding (cont.)

The Concepts tab recognizes high-level concepts that are related to the text. For example, the analysis of a research paper about deep learning would likely return the concept "Artificial Intelligence", even if that term is not explicitly mentioned in the paper.

IBM Training

Watson Natural Language Understanding (cont.)



Natural language processing and IBM Watson

Figure 8-21. Watson Natural Language Understanding (cont.)

The Semantic Roles tab parses sentences into subject, action, and object form.

8.4. Watson Discovery

IBM Training

Watson Discovery

Natural language processing and IBM Watson

Figure 8-22. Watson Discovery

IBM Training

Topics

- Watson NLP services
- Watson Natural Language Classifier
- Watson Natural Language Understanding
- Watson Discovery

Natural language processing and IBM Watson

Figure 8-23. Topics

IBM Training

Watson Discovery

- Adds cognitive search and content analytics to applications to identify patterns, trends, and insights.
- Unlocks actionable insights into unstructured data.
- Unifies structured and unstructured data.
- Uses simple query language to eliminate the need for manual filtering of results.
- Parses documents and enables the user to search through the documents by using a natural language query.
- Can be trained.
- Includes the Discovery API and Discovery tooling.
- Example application: Find answers to FAQs

Natural language processing and IBM Watson

Figure 8-24. Watson Discovery

Watson Discovery

Watson Discovery is a cognitive search and content analytics engine that enables you to add a cognitive search and content analytics engine to applications to identify patterns, trends, and actionable insights that drive better decision-making. You can rapidly build cognitive, cloud-based exploration applications that unlock actionable insights that are hidden in unstructured data.

You can securely unify structured and unstructured data with pre-enriched content, and use a simplified query language to eliminate the need for manual filtering of results.

With Discovery, you can prepare your unstructured data, create a query that will pinpoint the information you need, and then integrate those insights into your new application or existing solution.

The Discovery services includes:

- Discovery API: The Discovery service supports a number of SDKs to simplify the development of applications. The SDKs are available for many popular programming languages and platforms, including Node.js, Java, and Python.
- Discovery tooling: The Discovery service includes a complete set of online tools - the Discovery tooling - to help you quickly setup an instance of the service and populate it with data. The Discovery service tooling has been designed to save time by eliminating the need to use APIs to configure and populate your service.

With Discovery, you can build applications that extract the correct answers to FAQs by enriching and searching data collections.

IBM Training

Discovery flow

- Discovery provides an end-to-end solution that includes document conversion, data storage, and a simplified query language.
- Discovery uses enrichments provided by the Natural Language Understanding service

Natural language processing and IBM Watson

Figure 8-25. Discovery flow

The slide shows the architecture of a complete Discovery service solution:

Data

Accepts the following document formats:

- .pdf
- .docx
- HTML
- JSON

Ingestion

Documents are automatically converted and enriched by using the Watson APIs to add NLP metadata to the content making it easier to explore and discover insights.

Storage

Data is indexed into a collection as part of your environment in the cloud.

Query

Understand data faster, create better hypothesis and deliver better outcomes.

Output

Actionable insights.

Comparing Watson NLU to Watson Discovery, notice that Watson Discovery is an end-to-end solution that includes document conversion, data storage, and a simplified query language.

Watson Discovery use enrichments that are powered by the Watson Natural Language Understanding service, that is, NLU is built into Watson Discovery.

IBM Training

Watson Discovery News

- Data collection of recent news articles.
- It is pre-enriched with the following cognitive metadata:
  - Keyword Extraction, Entity Extraction, Semantic Role Extraction, Sentiment Analysis, Relation Extraction, and Category Classification
- Supports English, Spanish, German, Korean, and Japanese.
- Updated continuously with new articles.
- Use cases
  - News alerting
  - Event detection
  - Trending topics in news

Natural language processing and IBM Watson

Figure 8-26. Watson Discovery News

Discovery News is a public data set that has been pre-enriched with cognitive insights. It is also included with Discovery. You can use this public, unstructured data set to query for insights that you can integrate into your applications.

Watson Discovery News is an indexed dataset that is pre-enriched with the following cognitive insights:

- Keyword extraction
- Entity extraction
- Semantic role extraction
- Sentiment analysis
- Relation extraction
- Category classification

Watson Discovery News is updated continuously with new articles, and is available in English, Spanish, German, Korean, and Japanese.

- Discovery News English is updated with approximately 300,000 new articles daily.
- Discovery News Spanish is updated with approximately 60,000 new articles daily.
- Discovery News German is updated with approximately 40,000 new articles daily.
- Discovery News Korean is updated with 10,000 new articles daily.
- Discovery News Japanese is updated with approximately 17,000 new articles daily.

The news sources vary by language, so the query results for each collection will not be identical.

Use cases for Watson Discovery News:

- **News alerting** - Create news alerts by taking advantage of the support for entities, keywords, categories, and sentiment analysis to watch for both news and how it is perceived.
- **Event detection** - The subject/action/object semantic role extraction checks for terms/actions such as "acquisition", "election results", or "IPO".
- **Trending topics in the news** - Identify popular topics and monitor increases and decreases in how frequently they (or related topics) are mentioned.

Reference:

https://www.ibm.com/blogs/bluemix/2017/08/watson-discovery-service-updates-news-nlp/

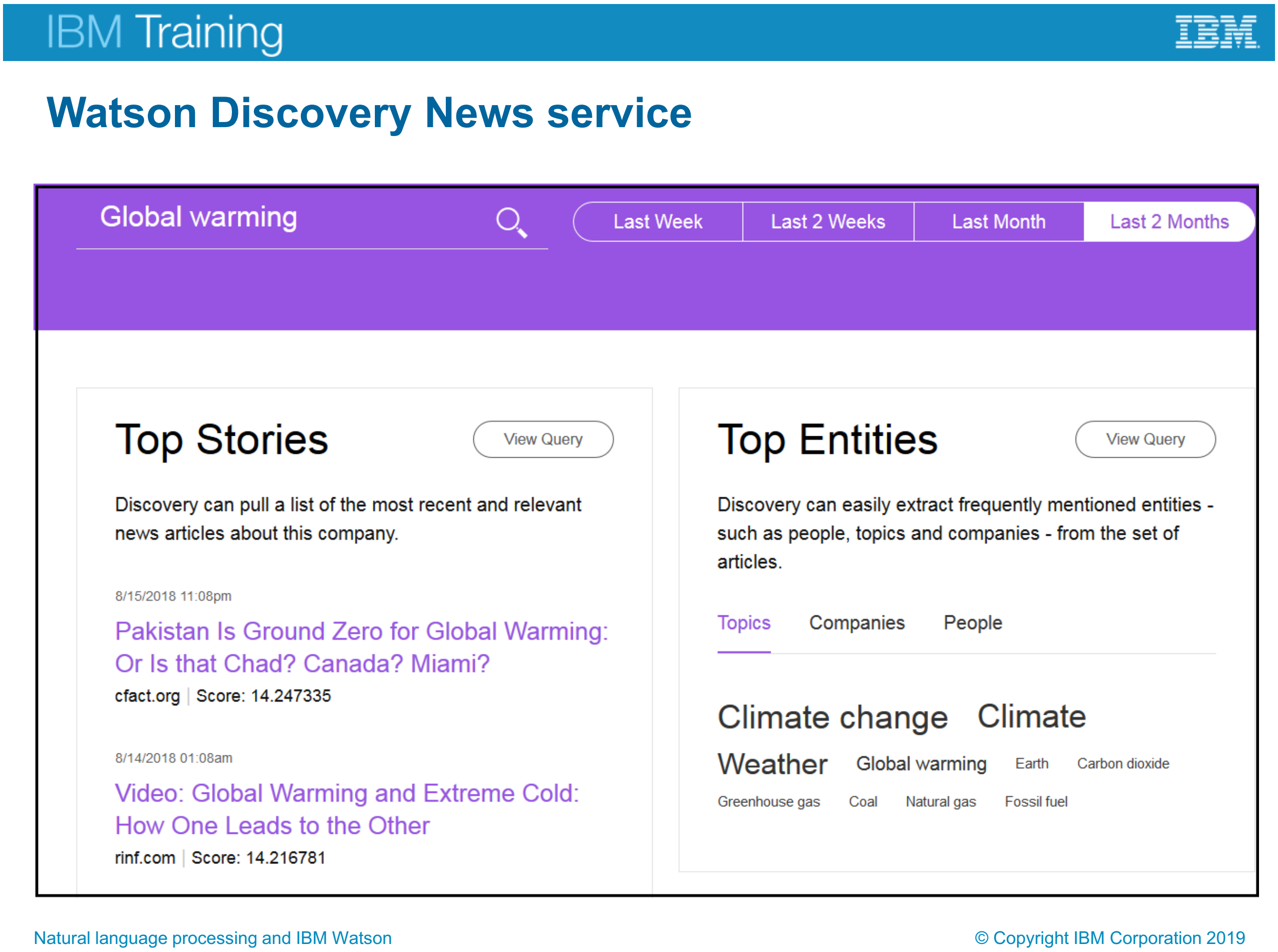


Figure 8-27. Watson Discovery News service

Access the Watson Discovery News demonstration to quickly find insights in the Watson Discovery News data collection of recent news articles. Easily explore a company's:

- Top stories over the last two months
- Top entities (people, topics, companies) mentioned in those articles
- Trend of public sentiment in news
- Anomalous periods of high press coverage
- Trend of most commonly paired entities (co-mentions)

You can access the demonstration at https://discovery-news-demo.ng.bluemix.net/.

IBM Training

Unit summary

- List the NLP Watson services.
- List the Watson services that perform information extraction.
- Describe the capabilities of IBM Watson Natural Language Classifier.
- Describe the capabilities of the IBM Watson Natural Language Understanding.
- Describe the capabilities of IBM Watson Discovery.

Natural language processing and IBM Watson

Figure 8-28. Unit summary

IBM Training

Review questions

1. Which of the following inputs can be analyzed by Watson Natural Language Understanding?
   A. Text.
   B. HTML.
   C. Public URL.
   D. All the above.
2. True or False: Watson Discovery News is updated continuously with new articles.

Natural language processing and IBM Watson

Figure 8-29. Review questions

1. D.
2. True.

IBM Training

Review questions (cont.)

3. Watson Discovery provides one of the following capabilities:
   A. Returns best matching class for a sentence or phrase.
   B. Identifies a variety of tones at both the sentence and document level.
   C. Derives insights from transactional and social media data to identify psychological traits.
   D. Adds cognitive search and content analytics to applications to identify patterns, trends, and insights.

4. True or False: Discovery provides an end-to-end solution that includes document conversion, data storage, and a simplified query language.

Natural language processing and IBM Watson

Figure 8-30. Review questions (cont.)

3. D (A is Natural Language Classifier, B is Tone Analyzer, and C is Personality Insights).
4. True.

IBM Training

Review answers

1. Which of the following inputs can be analyzed by Watson Natural Language Understandingt?
   A. Text.
   B. HTML.
   C. Public URL.
   D. All of the above.

2. True or False: Watson Discovery News is updated continuously with new articles.

Natural language processing and IBM Watson

Figure 8-31. Review answers

1. D
2. True.

IBM Training

Review answers (cont.)

3. Watson Discovery provides one of the following capabilities:
   - A. Returns best matching class for a sentence or phrase.
   - B. Identifies a variety of tones at both the sentence and document level.
   - C. Derives insights from transactional and social media data to identify psychological traits.
   - D. Adds cognitive search and content analytics to applications to identify patterns, trends, and insights.

4. True or False: Discovery provides an end-to-end solution that includes document conversion, data storage, and a simplified query language.

Natural language processing and IBM Watson

Figure 8-32. Review answers (cont.)

3. D (A is Natural Language Classifier, B is Tone Analyzer, and C is Personality Insights).
4. True.

IBM Training

Exercise: Ingest, convert, enrich, and query with Watson Discovery service

Natural language processing and IBM Watson

Figure 8-33. Exercise: Ingest, convert, enrich, and query with Watson Discovery service

IBM Training

Exercise objectives

- With Watson Discovery, it only takes a few steps to prepare your unstructured data, create a query that will pinpoint the information you need, and then integrate those insights into your new application or existing solution.
  - This exercise takes you through the process of preparing and querying a collection of documents.
- After completing this exercise, you should be able to:
  - Create a Watson Discovery service instance.
  - Create a collection.
  - Add content to a collection.
  - Create a custom configuration.
  - Build queries.
  - Use the Watson Discovery API.

Natural language processing and IBM Watson

Figure 8-34. Exercise objectives

After completing this exercise, you should be able to:

- Create a Watson Discovery service instance.
- Create a collection.
- Add content to a collection.
- Create a custom configuration.
- Build queries.
- Use the Watson Discovery API.

Unit 9. Introduction to IBM Watson Knowledge Studio

Estimated time

00:45

Overview

This unit introduces IBM Watson Knowledge Studio, its capabilities, and features. This unit explains the end-to-end domain adaptation process.

IBM Training

Unit objectives

- Describe IBM Watson Knowledge Studio.
- List the Watson services that are trained by Knowledge Studio.
- List the Knowledge Studio workspace resources.
- Explain the process to build Knowledge Studio models that can be deployed and used with other Watson services.

Introduction to IBM Watson Knowledge Studio

Figure 9-1. Unit objectives

9.1. IBM Watson Knowledge Studio overview

IBM Training

IBM Watson Knowledge Studio overview

Introduction to IBM Watson Knowledge Studio

Figure 9-2. IBM Watson Knowledge Studio overview

IBM Training

Topics

- IBM Watson Knowledge Studio overview
- Watson Knowledge Studio features
- Adapting the system to a new domain

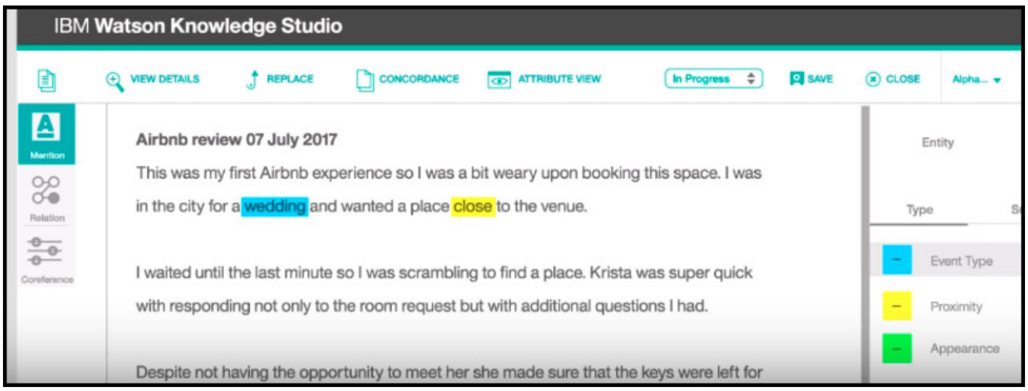
Introduction to IBM Watson Knowledge Studio

Figure 9-3. Topics

IBM Training

What is Watson Knowledge Studio

- Use Knowledge Studio to create:
  - A machine learning model that understands the linguistic nuances, meaning, and relationships that are specific to your industry.
  - A rule-based model that finds entities in documents.
- The model that you create with Knowledge Studio is a software component that can be plugged into a natural language processing (NLP) pipeline.



Introduction to IBM Watson Knowledge Studio

Figure 9-4. What is Watson Knowledge Studio

Knowledge Studio is a service on IBM Cloud that can help you create a machine learning model that understands the linguistic nuances, meaning, and relationships that are specific to your domain. It also can help you create a rule-based model that finds entities in documents that is based on rules that you define.

- **Machine learning model:** Uses a statistical approach to find entities and relationships in documents. This type of model can adapt as the amount of data grows.
- **Rule-based model:** Uses a declarative approach to finding entities in documents. This type of model is more predictable, and is easier to comprehend and maintain. However, it does not learn from new data. It can find only patterns that it has been trained to find.

The model that you create with Knowledge Studio is a software component that can be plugged into a natural language processing (NLP) pipeline.

References:

https://console.bluemix.net/docs/services/knowledge-studio/index.html#wks_overview_full

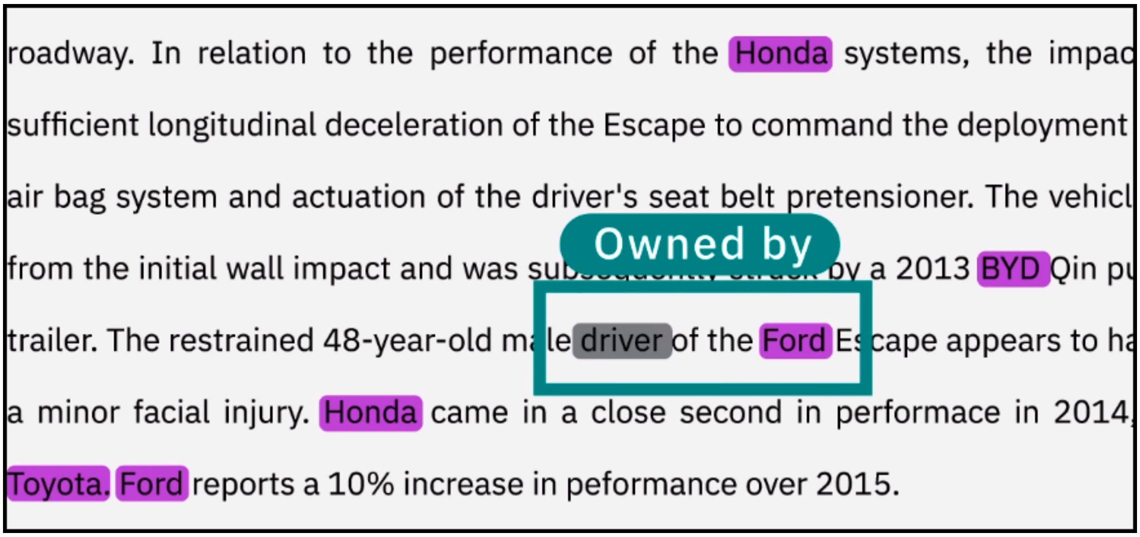
https://www.youtube.com/watch?v=r2xYHW0iyZM&index=3&list=PLZDyxLINKRY9yM_Deg6GhTU1WS0-JLwyX&

https://console.bluemix.net/docs/services/watson-knowledge-studio/annotate-documents.html#wks_lifecycle__wks_lifecycleS6

IBM Training

What is Watson Knowledge Studio (cont.)

- Easy-to-use tools for annotating unstructured domain literature.
- Uses those annotations to create a custom machine learning model.
- The accuracy of the model improves through iterative testing.
- Deploy the finished model to find and extract mentions of relationships and entities.



Introduction to IBM Watson Knowledge Studio

Figure 9-5. What is Watson Knowledge Studio (cont.)

Knowledge Studio provides easy-to-use tools for annotating unstructured domain literature, and uses those annotations to create a custom machine learning model that understands the language of the domain. The accuracy of the model improves through iterative testing, ultimately resulting in an algorithm that can learn from the patterns that it sees and recognizes those patterns in large collections of new documents.

You can deploy the finished machine learning model to other Watson cloud-based offerings and cognitive solutions to find and extract mentions of relationships and entities, including entity coreferences.

References:

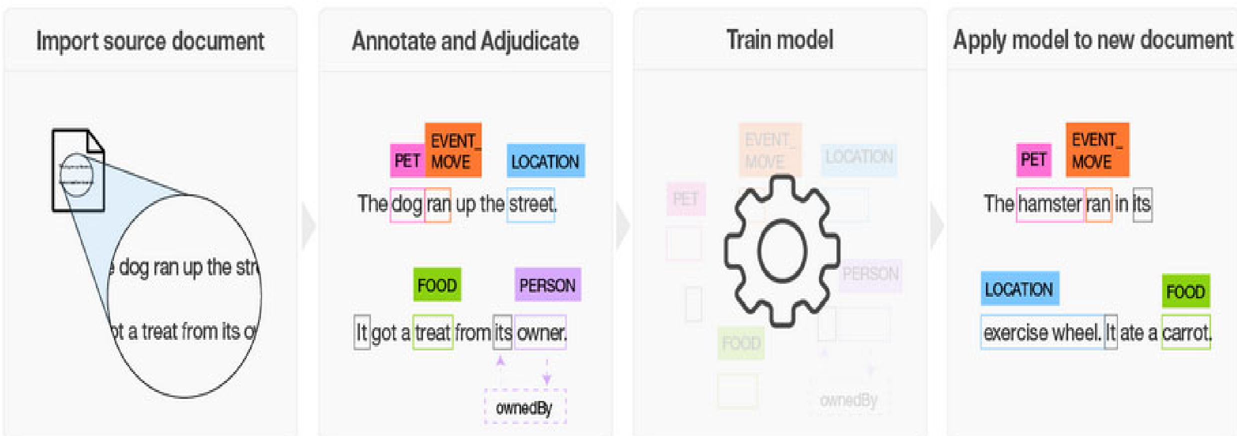
https://console.bluemix.net/docs/services/knowledge-studio/index.html#wks_overview_full

https://www.youtube.com/watch?v=r2xYHW0iyZM&index=3&list=PLZDyxLINKRY9yM_Deg6GhTU1WS0-JLwyX&

IBM Training

Building a machine learning model

1. Import domain-specific documents.
2. Human annotators annotate documents.
3. Knowledge Studio uses a ground truth to train a model.
4. A trained model is ready to find entities, relationships, and co-references in new documents.



Introduction to IBM Watson Knowledge Studio

Figure 9-6. Building a machine learning model

1. Based on a set of domain-specific source documents, the team creates a type system that defines entity types and relation types for the information of interest to the application that uses the model.
2. A group of two or more human annotators annotates a small set of source documents to label words that represent entity types, identify relationship types where the text identifies relationships between entity mentions, and define co-references, which identify different mentions that refer to the same thing, that is, the same entity. Any inconsistencies in the annotations are resolved, and one set of optimally annotated documents is built, which forms the *ground truth*.
3. Knowledge Studio uses the ground truth to train a model.
4. The trained model is used to find entities, relationships, and co-references in new documents.

References:

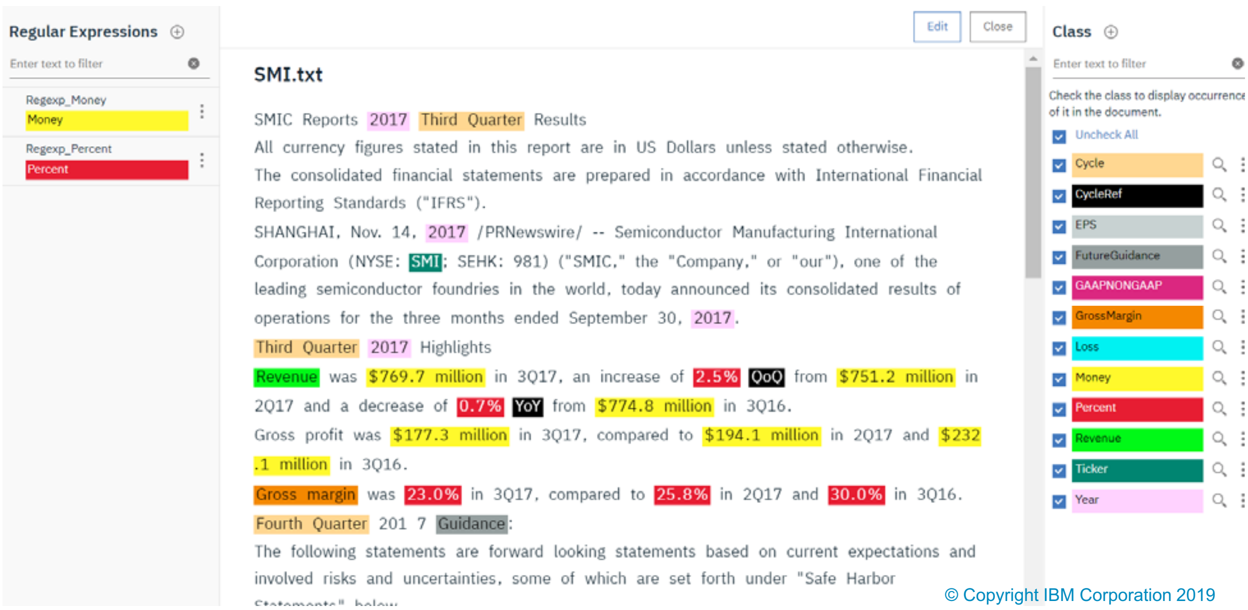
https://console.bluemix.net/docs/services/knowledge-studio/index.html#wks_overview_full

https://www.youtube.com/watch?v=r2xYHW0iyZM&index=3&list=PLZDyxLINKRY9yM_Deg6GhTU1WS0-JLwyX&

IBM Training

Building a rule-based model

- A rule-based model can recognize patterns in your documents.
- Create rules to capture patterns in documents and convey information about underlying entity types.



© Copyright IBM Corporation 2019

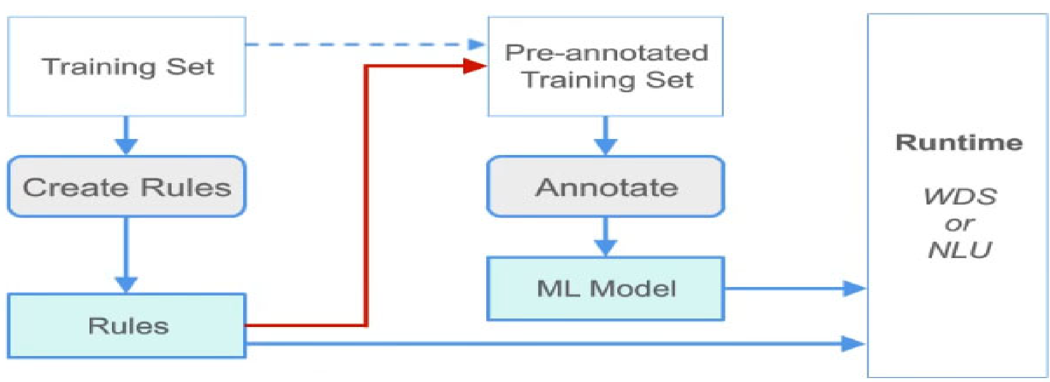
Figure 9-7. Building a rule-based model

Create a rule-based model that can recognize patterns in your documents. Use rules to capture patterns that occur in documents and convey information about underlying entity types. Rules can be defined to recognize person’s names, money, dates, percentage, and more.

IBM Training

Building a rule-based model (cont.)

- Tools to define rules:
  - Dictionary
  - Regular expression tool (Regex tool)
- Use the rule-based model to do useful things like pre-annotating documents to speed up the human annotation process for a machine learning model.



Introduction to IBM Watson Knowledge Studio

Figure 9-8. Building a rule-based model (cont.)

Knowledge Studio provides a **rule editor** that simplifies the process of finding and capturing common patterns in your documents as rules. It uses dictionaries to speed up the annotation process.

The rule editor provides some tools that help you define rules:

- **Dictionary:** Add a dictionary and assign it a class name. Any words that are found that match entries in the dictionary are automatically annotated with the dictionary class.
- **Regular expression:** A regular expression is a sequence of characters that define a search pattern. The Regex tool that is included in the rule editor recognizes expressions. For example, your documents might have several references like the following ones:
  - 35-year-old driver
  - 16-year-old learner

  The syntax *n-year-old x* is a pattern that typically represents a person. You can define a regular expression rule to find phrases that match the *n-year-old x* pattern, and annotate them as PERSON entity mentions.

References:

https://console.bluemix.net/docs/services/knowledge-studio/index.html#wks_overview_full

https://developer.ibm.com/tv/accelerate-watson-model-development-rule-based-approach/

IBM Training

Machine learning based model versus a rule-based model

Model type	Pros	Cons
Machine learning based	• Statistical (teaches by example). • Useful for complex text extraction that has many variations. • Scalable.	• Requires work to develop a supervised corpora (ground truth), and it requires certain amount of data. • Sometime trapped by corner cases.
Rule based	• Declarative (codify a set of rules). • Useful for basic text extraction, for example, extracting emails, URLs, and phone numbers. • Easy to comprehend and maintain.	• Requires work to develop, write, and define a set of rules. • As complexity grows, maintenance becomes an issue.

Introduction to IBM Watson Knowledge Studio

Figure 9-9. Machine learning based model versus a rule-based model

Machine learning models learn by examples that contain the information that we want to extract for the specific domain. A machine learning model is useful when the information that you are interested in does not have a certain form that can be codified into a rule, which is the case with unstructured natural language.

Rule-based models are most useful when they are used to extract data that is known to have a certain format, such as email addresses or phone numbers. If the information has many forms that it can come in, then you must define a rule or set of rules for each format, which is inefficient if the information that you must extract comes in many forms and has many variations.

The table summarizes both model types and their pros and cons.

Reference:

https://developer.ibm.com/tv/accelerate-watson-model-development-rule-based-approach/

IBM Training

Why do we need Watson Knowledge Studio

- With Knowledge Studio, you can create, evaluate, and improve models for new domains.
- A model adds annotations (metadata) to text that appears in natural language content.
- The annotations, identify mentions of entities of interest in your domain content, the relationships between them, and how the mentions co-reference the same entity.
- Applications can use the annotations to automatically analyze and process text.
- Application users benefit from this level of analysis by extracting meaning, discovering insights, and obtaining answers in a natural language context.

Introduction to IBM Watson Knowledge Studio

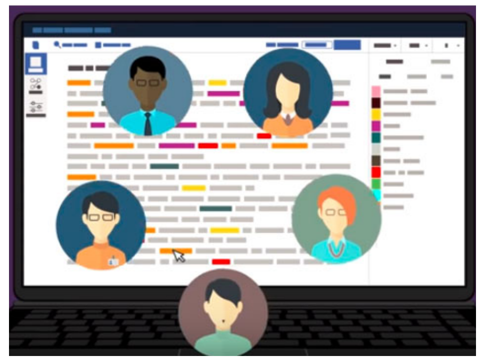
Figure 9-10. Why do we need Watson Knowledge Studio

With Knowledge Studio, you can create, evaluate, and improve models for new domains. A model adds annotations (metadata) to text that appears in natural language content. The annotations, which identify mentions of entities of interest in your domain content, the relationships between them, and how the mentions co-reference the same entity, can be used by applications to automatically analyze and process text. Application users benefit from this level of analysis by being able to extract meaning, discover insights, and obtain answers in a natural language context.

IBM Training

Why do we need Watson Knowledge Studio (cont.)

- **Teach by example:** Empower domain subject matter experts to teach Watson the language of your domain without requiring deep technical skills or coding.
- **Engage your experts:** Bring your experts together in a collaborative online environment to teach Watson.
- **Use everywhere:** Apply your models to Watson Discovery, Watson Natural Language Understanding, and Watson Explorer.
- **Knowledge Studio is end-to-end:** Combines annotation, training, and evaluation in a single collaboration-focused service.



Introduction to IBM Watson Knowledge Studio

Figure 9-11. Why do we need Watson Knowledge Studio (cont.)

To become a subject matter expert in an industry or domain, Watson services must be trained to understand specialized terminology and the relationships between entities. Knowledge Studio facilitates the training of some Watson offerings. Knowledge Studio simplifies the task of building machine learning models by helping subject matter experts to train Watson based on examples without code.

Knowledge Studio allows you to:

- **Teach by example:** Empower domain subject matter experts to teach Watson the language of your domain without requiring deep technical skills or coding.
- **Engage your experts:** Bring your experts together in a collaborative online environment to teach Watson how to understand the linguistic nuances of your domain.
- **Use everywhere:** Apply what you taught Watson in multiple applications by using models from Knowledge Studio in Discovery, Natural Language Understanding, and Explorer.
- **Knowledge Studio is end-to-end:** It is the only offering in the market that combines the powerful capabilities of annotation, training, and evaluation in a single collaboration-focused service.

9.2. Watson Knowledge Studio features

IBM Training

Watson Knowledge Studio features

Introduction to IBM Watson Knowledge Studio

Figure 9-12. Watson Knowledge Studio features

IBM Training

Topics

- IBM Watson Knowledge Studio overview
- Watson Knowledge Studio features
- Adapting the system to a new domain

Introduction to IBM Watson Knowledge Studio

Figure 9-13. Topics

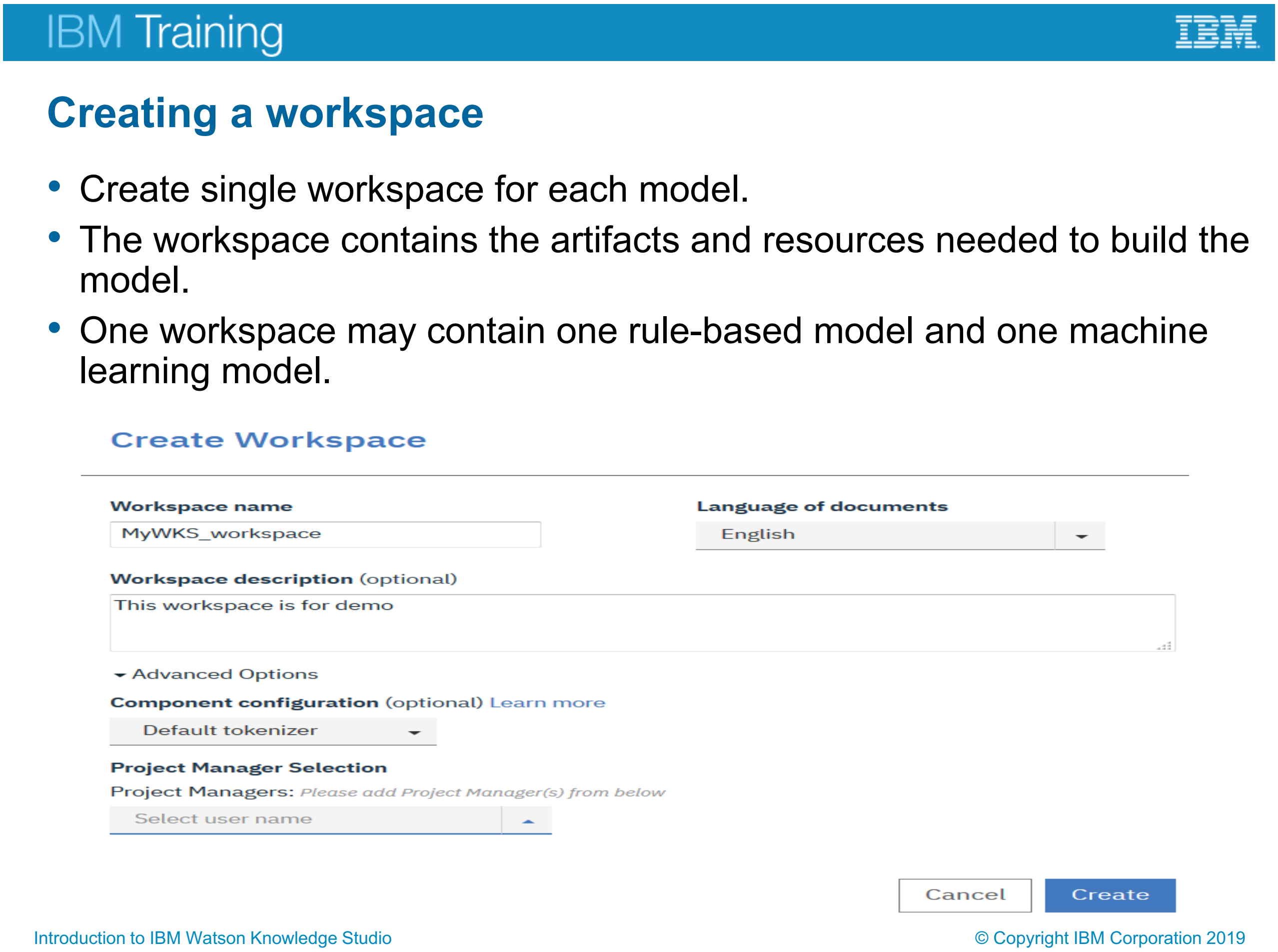


Figure 9-14. Creating a workspace

The first step in building a custom model is to create a workspace. For each model that you want to build and use, create a single workspace that contains the artifacts and resources that are needed to build the model. You can create one workspace that contains both one rule-based model and one machine learning model. Then, you train the model to produce a custom model that can be deployed to an external service for use. Follow these steps:

1. Give your workspace a name and provide a description about the domain to easily remember it.
2. Choose the language of your trained documents and dictionaries.
3. Optional: If you want to change the tokenizer that is used by the application from the default machine learning-based tokenizer, expand the **Advanced Options** section and select **Dictionary-based tokenizer**. For more information, see the References for this slide.
4. Optional: You can add project manager role, but because you have a Lite plan subscription, skip this step. You cannot add other users. As an administrator, you can perform all the tasks that a project manager typically performs.

References:

https://console.bluemix.net/docs/services/watson-knowledge-studio/create-project.html#create-project

https://www.youtube.com/watch?v=EQcEk2TX79c

IBM Training

Workspace resources

- You add the following types of resources to the workspace:
  - Type system
  - Dictionaries
  - Documents
- The resources are either created or uploaded for training the models.
- Each of these resources may be used differently depending on the model type.

Introduction to IBM Watson Knowledge Studio

Figure 9-15. Workspace resources

The Knowledge Studio workspace contains the following types of resources:

- Type System: Defines the entities and relationships between entities that matter to you.
- Dictionaries: Group words and phrases that should be treated equivalently by a model.
- Documents: Serve a different purpose depending on whether you are creating a machine learning model or a rule-based model.

The resources are either created or uploaded to train the models.

Each of these resources may be used differently depending on the model type.

Reference:

https://console.bluemix.net/docs/services/watson-knowledge-studio/artifacts.html#artifacts

IBM Training

Type system

- Defines entities and relationships between entities that are interesting for your domain:
  - Mentions. Example: “Watson”, “IBM”.
  - Entities types: Example: “PERSON”, “ORGANIZATION”.
  - Relation types: Example: founderOf, employedBy.
- Controls how content can be annotated.
- Can be created from scratch or uploaded from an existing type system.
- If you create a type system from scratch, define it directly in the Knowledge Studio editor.
- To avoid visual overload for human annotation, define no more than 50 entity types and 50 relation types.

Introduction to IBM Watson Knowledge Studio

Figure 9-16. Type system

A type system defines things that are interesting in your domain content that you want to label with an annotation:

- **Mentions:** A mention is any span of text that you consider relevant in your domain data. For example, “President Obama” or “IBM”.
- **Entities types:** Categorize a real-world thing. An entity mention is an example of a thing of that type. For example, the mention President Obama can be annotated as a PERSON entity type. The mention IBM can be annotated as an ORGANIZATION entity type.
- **Relation types:** Define a binary and ordered relationship between two entities. For a relation mention to exist, the text must explicitly define the relation and bind mentions of the two entities together, and must do so within a single sentence. For example, the sentence “Mary works for IBM” is textual evidence of the employedBy relation type.

The type system controls how content can be annotated by defining the types of entities that can be labeled and how relationships among different entities can be labeled. The model process manager typically works with subject matter experts for your domain to define the type system.

In Knowledge Studio, you can create a type system from scratch or upload one from an existing type system. To start a workspace, you might want to upload a type system that was created for a similar domain. You can then edit the type system to add or remove entity types or redefine the relationship types.

IBM Training

Knowledge from Language Understanding and Extraction type system

- The **K**nowledge from **L**anguage **U**nderstanding and **E**xtraction (KLUE) is a sample type system that will be used in exercises.
- It was derived by IBM Research based on the analysis of collections of news articles.
- Many industries publish dictionaries or ontologies of domain-specific terminology, such as metallurgy, geology, market intelligence, life science, electronic health records, and oncology.
- Consider referencing this type of resource to get an idea of the types of entities you might want to define in your own type system.

Introduction to IBM Watson Knowledge Studio

Figure 9-17. Knowledge from Language Understanding and Extraction type system

A sample type system based on the *KLUE* type system is provided for you to use with the Knowledge Studio exercises.

KLUE stands for Knowledge from Language Understanding and Extraction and was derived by IBM Research based on the analysis of collections of news articles.

Open the JSON KLUE type system from:
https://watson-developer-cloud.github.io/doc-tutorial-downloads/knowledge-studio/en-klue2-types.json

Show the structure of the KLUE type system to students mainly entityTypes and relationshipTypes

Show students “entityTypes” in JSON, Open number 24 and show students “PEOPLE” entity, and number 25 showing “EVENT_VIOLENCE” entity. Remind them that a type system defines the entity types that we are interested in detecting in our data.

Show students “relationshipTypes” in JSON, open number 2, it has a source entity type (represented by the id of that entity as was previously defined for entity types) and a target entity type and the label or name of the relationship that we are interested in which in this case is ownerOf.

Many industries, such in domains like metallurgy, geology, market intelligence, life science, electronic health records, and oncology publish dictionaries or ontologies of domain-specific terminology. Consider referencing this type of resource to get an idea of the types of entities you might want to define in your own type system.

IBM Training

Dictionary

- A dictionary groups words and phrases that should be treated equivalently by a model. Example:
  - September, Sept, Sep.
  - Monday, Tuesday, Wednesday (DAY_OF_WEEK).
- Dictionaries help the Knowledge Studio machine learning models to understand the language of the domain.
- You can create dictionaries in Knowledge Studio:
  - Manually by adding individual entries.
  - Upload several types of dictionary files.
- Several general-purpose dictionaries are built in to the application.

Introduction to IBM Watson Knowledge Studio

Figure 9-18. Dictionary

Dictionaries

In machine learning, a dictionary groups words and phrases that share something . An entry in the dictionary does not mean that all words in the entry mean the same thing, but that the words are to be treated equivalently by a model.

A dictionary is a list of words or phrases that are equivalent for information-extraction purposes, meaning that they are interchangeable for the purposes of identifying entity and relation mentions.

Consider this example: A dictionary entry contains the seven days of the week. To annotate a document, a human annotator assigns the entity type DAY_OF_WEEK to mentions of *Monday* and *Friday* in the text. Because the dictionary equates the seven days of the week, it helps ensure that a machine learning model correctly annotates occurrences of *Tuesday*, *Wednesday*, and the other days of the week in unseen documents at run time. In addition, equating these words also benefits information extraction in the surrounding text. What the machine learning model learns from training examples about the texts near *Monday* and *Friday* is applied to texts that the machine learning model sees near other days of the week because the dictionary states that these terms are equivalent for information-extraction purposes.

You do not need to create a dictionary that contains days of the week information. Several general-purpose dictionaries are built in to the application. Other built-in dictionaries include countries, place names, number words, animals, plants, diseases, measurement words (such as *ounce* and *meter*), and salutation title words (such as *Mr.* and *Mrs.*). You cannot disable or edit built-in dictionaries.

IBM Training

Dictionary usage

- Adding dictionaries is optional and may be used in two ways:
  - Machine learning usage: Used by the machine learning model to provide words or phrases that are equivalent for information-extraction purposes.
  - Pre-annotation usage: Used during pre-annotation to bootstrap the annotation effort.

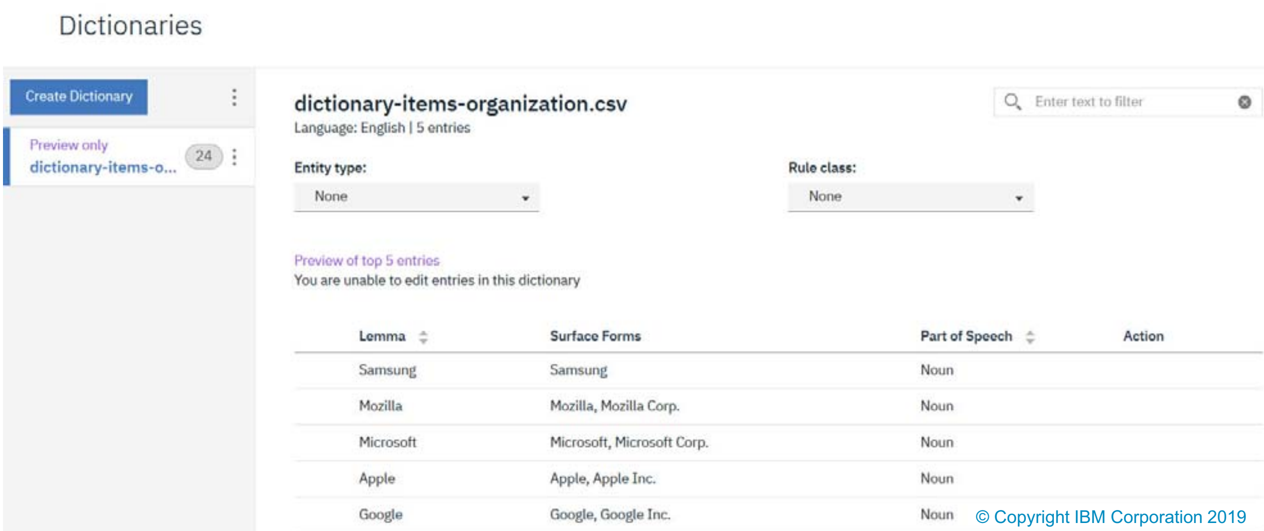


Figure 9-19. Dictionary usage

Adding dictionaries is optional and may be used in two ways:

- **Machine learning usage:** The machine learning model does not assume that a mention has a specific entity type just because the mention matches an entry in a dictionary that is associated with that entity type. It accounts for that information, but treats it as one piece of information among other pieces of information that it gathers through linguistic analysis. In fact, if none of the terms in a dictionary occur in the ground truth documents, then the dictionary is not used at all by the machine learning model.
- **Pre-annotation usage:** Dictionaries are important to the following pre-annotation processes:
  - Dictionary pre-annotator: You associate a dictionary with an entity type from the type system when you run the dictionary pre-annotator.
  - Rule-based model: You can optionally associate a dictionary with a rule class. Classes are then mapped to entity types from the type system when you run the rule-based model to pre-annotate documents. As a result, dictionary terms are, although circuitously, mapped to entity types for the rule-based model also.

In both cases, the dictionaries provide terms that the system can find and annotate as mentions. It assigns to each mention the entity type that is associated with the dictionary that contains the term. When a human annotator begins work on new documents that were pre-annotated, many mentions are already annotated based on the dictionary entries. The human annotator thus has more time to focus on assigning entity types to mentions that require deeper analysis.

Reference:

https://console.bluemix.net/docs/services/watson-knowledge-studio/dictionaries.html#dictionaries

IBM Training

CSV file dictionary

- The standard dictionary format.
- The maximum size of a CSV file that you can upload is 1 MB.
- The first row in the file must specify the following column headers: lemma,poscode,surface
- The remaining lines in the file specify the dictionary entries, where:
  - **Lemma**: Specifies the most representative word form for the entry.
  - **Poscode**: Specifies a code that identifies the part of speech.
  - **Surface**: Specifies equivalent terms, also called surface forms.
  - Example:

```
lemma,poscode,surface
IBM,3,IBM Corp.,IBM,International Business Machines
Department of Energy,3,DOE,Department of Energy
premium,4,premium,premium-grade
```

Introduction to IBM Watson Knowledge Studio

Figure 9-20. CSV file dictionary

Also referred to as the standard dictionary format, a dictionary in comma-separated value (CSV) format is a file that you can edit after you upload it. The maximum size of a CSV file that you can upload is 1 MB. If you have a larger dictionary file, then break the large file into multiple files and upload them one at a time into a single dictionary in your Knowledge Studio workspace.

To summarize the requirements, you must use a text editor to create the CSV file, not software like Microsoft Excel, and the file must use UTF-8 encoding that does not include the byte order mark (BOM) at the start of the text stream.

- **Lemma:** Specifies the most representative word form for the entry.
- **Poscode:** Specifies a code that identifies the part of speech. This part of speech information is used by the dictionary annotator to help with sentence tokenization.

  Note: This code supports the scenario where you want to upload a large machine-generated dictionary that does not include part of speech information in each entry. You can assign *unknown* to all entries by default. Avoid using this code, if possible.

  0 - Unknown

  1 - Pronoun

  2 - Verb

  3 - Noun

4 - Adjective

5 - Adverb

6 - Adposition

7 - Interjection

8 - Conjunction

9 - Determiner

10 - Quantifier

In English, noun (3), verb (2), and adjective (4) are the most common parts of speech that are used for dictionary entries.

- **Surface**

  Specifies equivalent terms, also called surface forms. Repeat the lemma as a surface form and use a comma to separate multiple surface forms. If a surface form includes a comma, enclose the surface form in quotation marks.

IBM Training

Documents

- Usage for a machine learning model:
  - Documents are used to train the machine learning model when they are annotated.
  - Examples: Journal articles or other industry-specific texts that are related to the domain for which the model is built.
- Usage for a rule-based model:
  - Add or upload documents from which you can draw patterns to define as rules.
  - Add documents with linguistic patterns that illustrate the types of rules you want to define.
- Documents that you add for defining rules are kept separate from documents that you add for annotation.

Introduction to IBM Watson Knowledge Studio

Figure 9-21. Documents

Documents are the main resources that are needed for custom model creation. Documents are used for different purposes based on the model being created.

- For a machine learning model:
  - Documents are used to train the machine learning model when they are annotated.
  - Examples: Journal articles or other industry-specific texts that are related to the domain for which the model is built.
- For a rule-based model:
  - Add or upload documents from which you can draw patterns to define as rules.
  - Add documents with linguistic patterns that illustrate the types of rules you want to define.

Documents that you add for defining rules are kept separate from documents that you add for annotation.

References:

https://console.bluemix.net/docs/services/knowledge-studio/documents-for-annotation.html#documents-for-annotation

https://console.bluemix.net/docs/services/knowledge-studio/rule-annotator-add-doc.html#wks_rule_anno_add

9.3. Adapting the system to a new domain

IBM Training

IBM

Adapting the system to a new domain

Introduction to IBM Watson Knowledge Studio

Figure 9-22. Adapting the system to a new domain

IBM Training

Topics

- IBM Watson Knowledge Studio overview
- Watson Knowledge Studio features
- Adapting the system to a new domain

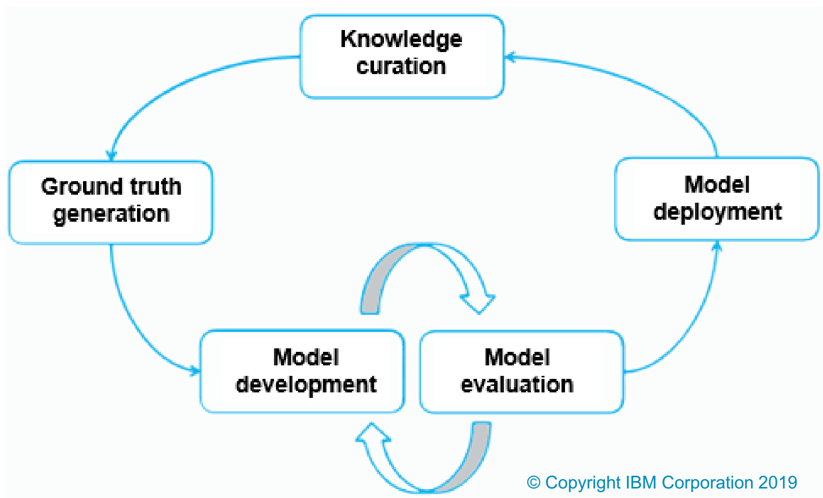
Introduction to IBM Watson Knowledge Studio

Figure 9-23. Topics

IBM Training

Model life cycle: End-to-end domain adaptation

- Adapting a system to a new domain, requires an iterative process with continuous improvements to increase the performance of the system.
- The creation of a model is an iterative multiple-step process that involves these stages:
  1. Knowledge curation
  2. Ground truth generation
  3. Model development
  4. Model evaluation
  5. Model deployment



Introduction to IBM Watson Knowledge Studio

Figure 9-24. Model life cycle: End-to-end domain adaptation

Adapting a cognitive system to a new domain, requires an iterative process with continuous improvements to increase the performance of the system. This iterative process aims to reach a level of incremental accuracy performing activities such as adding new functionalities, testing the system, identifying opportunities to improve the performance, doing headroom analysis, and finding possible solutions for the most frequent errors. The process requires the collaboration of domain experts, data scientists, natural language processing (NLP) experts, and machine learning developers.

The creation of a model is an iterative multiple-step process that involves several stages: knowledge curation, ground truth generation, model development, model evaluation, and model deployment.

The diagram summarizes the interactions between these five stages of model development and the typical activities that occur at each stage.

References:

http://www.redbooks.ibm.com/redbooks.nsf/redbookabstracts/sg248387.html?Open

https://console.bluemix.net/docs/services/watson-knowledge-studio/annotate-documents.html#wks_lifecycle

IBM Training

Knowledge curation

- This stage refers to the process of selecting, collecting, preserving, and maintaining content relevant to a specific domain.
- This stage is external to Knowledge Studio.
- Curation adds value to data: It transforms data into trusted information and knowledge.
- Knowledge curation involves the following activities:
  - Ensuring that the selected documents are representative of the domain content.
  - Ensuring that selected documents cover the entity types and relation types that are defined in your type system.
  - Providing more documents for a complex type system, and fewer for a simpler one.
  - Limiting each document to a page or two of content. This is a preferred practice, especially in the early stages of model development.
  - Ensuring that the documents include variations in vocabulary and phrasing.
  - In the case of skewed frequency-of-occurrence of entity types and relation types, trying to get at least 50 exemplars of each type.

Introduction to IBM Watson Knowledge Studio

Figure 9-25. Knowledge curation

Curation adds value to data by transforming data into trusted information and knowledge. This stage, which is external to Knowledge Studio, refers to the process of selecting, collecting, preserving, and maintaining content relevant to a specific domain.

This is an important stage because the curated content is used as input in the following stage.

You must ensure that the collected content is representative of your domain content and of high value to your application.

Ensure that the collected content covers the entity types and relation types that are defined in your type system. Sometimes, even a representative small sample of documents in the domain of interest is collected and the type system is modified and refined based on it.

Ensure that the data in the documents is distributed across all possible entity types, subtypes, and roles, and the relationships between them. A goal to aim for is to have at least 50 annotations for each entity type and 50 for each relation type in the document collection.

Strive to provide a set of documents that have a total size of about 300,000 words. Provide more words for a complex type system, and fewer for a simpler one.

Limit each document to a page or two of content (fewer than 2,000 words, and closer to 1,000 words per document is best). In the early stages of model development, keeping each document down to a few paragraphs is a preferred practice. A human annotator can mark mentions and relations in a long document, but attempts to mark co-references across multiple pages might prove unwieldy.

Documents should represent the breadth of the subject matter that the application covers, but in the case of skewed frequency-of-occurrence of entity types and relation types, try to get at least 50 exemplars of each type, and more for entity types that have mentions that tend to be phrases.

For more information about limitations, requirements, and supported input formats, see https://console.bluemix.net/docs/services/watson-knowledge-studio/create-project.html#machine-learning-model.

References:

https://github.com/IBM-Bluemix-Docs/watson-knowledge-studio/blob/master/documents-for-annotation.md

https://console.bluemix.net/docs/services/watson-knowledge-studio/annotate-documents.html#wks_lifecycle__wks_lifecycleS2

IBM Training

Ground truth generation

- This stage refers to the use of Knowledge Studio tools and best practices to produce a collection of vetted data called *ground truth* or *gold standard documents*.
  - The ground truth is used to adapt a Watson solution to a particular domain.
- Ground truth generation involves the following activities:
  1. SMEs creating or identifying the following resources for your domain:
     - > Annotation guidelines and example for human annotators.
     - > Type systems that define domain-specific objects and features.
     - > Dictionaries of terms that are to be treated as equivalent terms in your domain content.
  2. Creating a corpus of documents that are representative of your domain content.
  3. Pre-annotating documents based on the dictionaries that you add to a Knowledge Studio workspace.
  4. Annotating documents by human annotators. Documents are divided among human annotators who then use the Knowledge Studio ground truth editor to manually add annotations.
  5. Comparing the human annotation results and resolving conflicts.

Introduction to IBM Watson Knowledge Studio

Figure 9-26. Ground truth generation

This stage refers to the use of Knowledge Studio tools and best practices to produce a collection of vetted data that can be used to adapt a Watson solution to a particular domain. The accuracy of this vetted data, called *ground truth* or *gold standard documents*, is critical because inaccuracies in the ground truth will correlate to inaccuracies in the applications that rely on it.

An essential part of teaching Watson about a new domain involves providing it with knowledge about entities of interest in your domain content, the relationships between them, and how the entities co-reference each other. Collecting this knowledge includes the following activities:

1. Involving domain subject matter experts to create the following resources, or to identify existing resources that can be re-used or modified for your domain:
   - Annotation guidelines and examples to help human annotators learn how words and passages in your domain content are to be annotated.
   - Type systems that define the domain-specific types (objects) and features (data classifications) that can be discovered in your domain content through text analysis. The type system controls the types of annotations that a human annotator can add to documents.
   - Dictionaries of terms that are to be treated as equivalent terms in your domain content.
2. Creating a corpus of documents that are representative of your domain content.

3. Pre-annotating documents based on the dictionaries that you add to a Knowledge Studio workspace.

   This is an optional activity to make the process of human annotation easier. Instead of starting from scratch, it creates a base on which the human annotators can build.

   After you create a machine learning model, you can use the model to pre-annotate new documents that you add to the corpus. Pre-annotation is a process of machine-annotating a document to the extent possible before a machine learning model is available to do so. Pre-annotation can reduce human-annotation labor by replacing some human annotation creation with mere verification of the correctness of machine annotation.

4. Annotating documents by human annotators. Documents are divided among human annotators, who then use the IBM Watson Knowledge Studio ground truth editor tool to manually add annotations to small sets of documents. Humans annotate mentions, relations, and co-references. The annotated documents are used as training and testing data for the machine learning model.

5. Comparing the human annotation results and resolving conflicts. Adjudication in this phase is needed to ensure accurate and consistently annotated documents are promoted to ground truth, where they can be used to train and test a machine learning model.

References:

https://console.bluemix.net/docs/services/watson-knowledge-studio/annotate-documents.html#wks_lifecycle__wks_lifecycleS2

https://github.com/IBM-Bluemix-Docs/watson-knowledge-studio/blob/master/documents-for-annotation.md

https://console.bluemix.net/docs/services/watson-knowledge-studio/documents-for-annotation.html#documents-for-annotation

IBM Training

Model development

- This stage refers to the use of Knowledge Studio tools to create the model.
- After the ground truth is generated, the human annotation results are used to train an algorithm for automatically adding annotations to large collections of documents, such as collections that include millions of documents.
- Model development involves the following activities:
  1. Selecting the document sets that you want to use to train the model.
     Note: Only documents that became ground truth through approval or adjudication can be used to train the machine learning model.
  2. Specifying the percentage of documents that are to be used as training data, test data, and blind data.

Introduction to IBM Watson Knowledge Studio

Figure 9-27. Model development

This stage refers to the use of Knowledge Studio tools to create a model. After establishing ground truth, the human annotation results can be used to train an algorithm for automatically adding annotations to large collections of documents, such as collections that include millions of documents.

Model development involves the following activities:

1. Selecting the document sets that you want to use to train the model.

   Note: Only documents that became ground truth through approval or adjudication can be used to train the machine learning model.

2. Specifying the percentage of documents that are to be used as training data, test data, and blind data.

The documents that you add to the system must be allocated to the following system-level data sets when you create a machine learning model:

- **Training set:** A set of documents that was annotated through pre-annotation or by human annotators that is used to train the model. The goal of the training set is to teach the machine learning model about correct annotations, which includes teaching the model through text that was not annotated.
- **Test set:** A set of annotated documents that is used to test the trained model. After you run a test on the test set, perform a detailed diagnostic-purposed error analysis of the results. Close analysis helps you find weaknesses in the current model that can be addressed.
- **Blind set:** A set of annotated documents that is set aside and used to test the system periodically after several iterations of testing and improvement have occurred. To prevent accuracy from being tainted (for example, by making changes based only on annotations in known documents), blind data should be data that has not previously been viewed by users that were involved with creating the model. Reported results should come only from tests that are run on blind data. After you run a test on the blind set, look at only the most high-level scores, such as the overall mention and relation F1 scores. You do not want to learn too many details about the performance or it might influence the improvements that you choose to make to the model.

The goal of Knowledge Studio is to enable large teams to work together to build models. As such, it assumes that models are being produced by a team that includes a group of human annotators and a separate person or group of people that builds and tests the model, and makes improvements to it. Due to this assumption, the application is configured to push an equally proportioned grouping of documents from a single document set into the test, train, and blind sets. However, if your team is not segregated (for example, the people doing human annotation are also reviewing model test results in detail), then you might need to change the allocation of documents in these sets to more explicitly separate the documents that are being used in each one.

Why do I need a blind set?

Because you use test data to assess accuracy in detail, you get to know the documents and their features after a while. For example, you start to know which entity types, relation types, and text types in the documents are best understood by the machine learning model, and which are not. This information is important because it helps you focus on making the right improvements, that is, refining the type system, supplementing the training data to fill gaps, or adding dictionaries. As the test documents are used iteratively to improve the model, they can start to influence the model training indirectly. That is why the "blind" set of documents is so important.

You can review the annotations that are made on the test set. Infer which are undertrained entities and cycle through the steps again to provide better training material for these entities.

References:

https://console.bluemix.net/docs/services/watson-knowledge-studio/train-ml.html#train-ml

https://console.bluemix.net/docs/services/watson-knowledge-studio/annotate-documents.html#wks_lifecycle__wks_lifecycleS3

IBM Training

Model evaluation

- This stage refers to the use of Knowledge Studio tools to refine the model and improve performance.
- Model evaluation involves the following activities:
  1. Evaluating the results generated by the model against a test set of ground truth documents:
     - Accuracy analysis to identify the causes of annotation errors.
     - Headroom analysis to assess which errors require focus and where model refinements can yield the greatest impact.
  2. Reviewing the annotations that were added by the trained model to determine whether any adjustments must be made to the model to improve its ability to find valid entity mentions, relation mentions, and coreferences in the documents.
  3. Making adjustments repeatedly to improve performance until a satisfactory level of accuracy is achieved.

Introduction to IBM Watson Knowledge Studio

Figure 9-28. Model evaluation

This stage refers to the use of Knowledge Studio tools to refine the model and improve performance. The results generated by the model are evaluated against a test set of ground truth documents.

- *Accuracy analysis* identifies the causes of annotation errors.
- *Headroom analysis* helps you assess which errors require focus and where model refinements can yield the greatest impact.

After you determine areas in which the model is having trouble, take steps to improve its performance.

Adjustments can be made repeatedly to improve performance until a satisfactory level of accuracy is achieved.

References:

https://console.bluemix.net/docs/services/watson-knowledge-studio/evaluate-ml.html#evaluate-ml

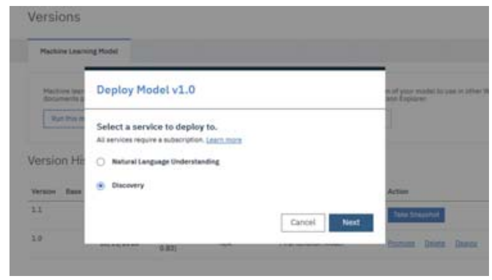
https://console.bluemix.net/docs/services/watson-knowledge-studio/improve-ml.html#improve-ml

https://console.bluemix.net/docs/services/watson-knowledge-studio/annotate-documents.html#wks_lifecycle__wks_lifecycleS4

IBM Training

Model deployment

- This stage refers to exporting components that enable the model to run in machine learning runtime environments and making the model accessible to other Watson applications.
- You can deploy the machine learning model for use by:
  - Watson Natural Language Understanding
  - Watson Discovery
  - Watson Explorer
- Model deployment involves the following activities:
  1. Creating a snapshot of the current model.
  2. Deploying the model to Watson Discovery or Watson Natural Language Understanding services
  3. Exporting the machine learning model so it can be used in Watson Explorer (optional).



Introduction to IBM Watson Knowledge Studio

Figure 9-29. Model deployment

This stage refers to exporting components that enable the model to run in machine learning runtime environments and making the model accessible to other Watson cognitive applications.

When you are satisfied with the performance of the model, you can leverage the machine learning model that you trained with Knowledge Studio by making it available for use by:

- IBM Watson Natural Language Understanding service
- IBM Watson Discovery service
- IBM Watson Explorer

You can deploy or export a machine learning model. A dictionary or Natural Language Understanding pre-annotator can only be used to pre-annotate documents within Knowledge Studio.

Model deployment involves the following activities:

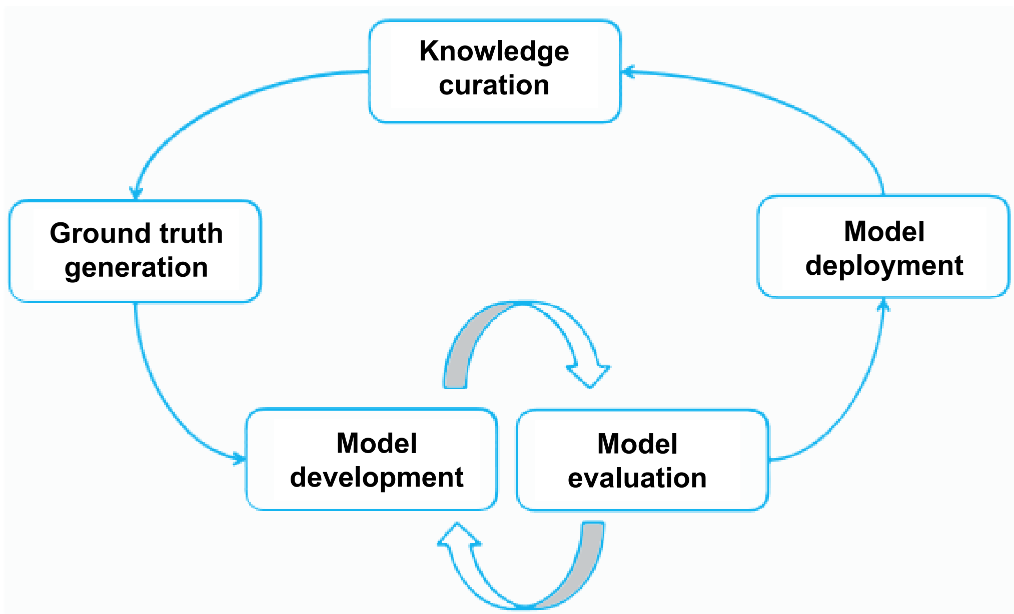
1. Creating a snapshot of the current model. This versions the model, which enables you to deploy one version, while you continue to improve the current version. The option to deploy does not appear until you create at least one version.
2. Deploying the model to Watson Discovery or Watson Natural Language Understanding services.
3. Exporting the machine learning model so it can be used in Watson Explorer. This step is not available for Lite IBM Cloud accounts.

Reference:

https://console.bluemix.net/docs/services/watson-knowledge-studio/annotate-documents.html#wks_lifecycle__wks_lifecycleS5

IBM Training

End-to-end domain adaptation wrap-up



Introduction to IBM Watson Knowledge Studio

Figure 9-30. End-to-end domain adaptation wrap-up

The following points summarize the end-to-end domain adaptation process with Watson Knowledge Studio:

- Adapting an AI system to a new domain, requires an iterative process with continuous improvements to increase the performance of the system. This iterative process aims to reach a level of incremental accuracy performing activities such as adding new functionalities, testing the system, identifying opportunities to improve the performance, doing headroom analysis, and finding possible solutions for the most frequent errors. The process requires the collaboration of domain experts, data scientists, natural language processing (NLP) experts, and machine learning developers.
- The creation of a model is an iterative multiple-step process that involves several stages: knowledge curation, ground truth generation, model development, model evaluation, and model deployment to runtime environments.
- Some tasks are performed outside Knowledge Studio and involve human experts to perform activities to ensure that the collected data that will be used to train the model is relevant to the domain, of high quality, and includes the right amount of representative documents.

- Knowledge Studio provides tools and artifacts to:
  - Guide human annotators while performing manual annotations.
  - Pre-annotate documents to bootstrap human annotations.
  - Evaluate annotations by multiple human annotators and generate the ground truth for your domain.
  - Create models.
  - Evaluate the performance of the trained model.
  - Deploy the model to Watson Discovery and Watson Natural Language Understanding or export the model for use in Watson Explorer.
- Domain adaptation is a continuous improvement process which requires an ongoing effort to improve the performance of the model. Domain adaptation is the primary purpose of Knowledge Studio.

IBM Training

Unit summary

- Describe IBM Watson Knowledge Studio.
- List the Watson services that are trained by Knowledge Studio.
- List the Knowledge Studio workspace resources.
- Explain the process to build Knowledge Studio models that can be deployed and used with other Watson services.

Introduction to IBM Watson Knowledge Studio

Figure 9-31. Unit summary

IBM Training

Review questions

1. True or False: Rule-based models are efficient for complex text extraction that has variations.
2. _______ defines entities and relationships between entities that are interesting for your domain (mentions, entity types, relation types).
   A. Dictionary
   B. Type system
   C. Pre-annotation
   D. Ground truth
3. The following Watson service **cannot** be trained by using Knowledge Studio.
   A. Watson Natural Language Classifier
   B. Watson Natural Language Understanding
   C. Watson Discovery
4. True or False: A ruled based model can be used to pre-annotate documents to speed up the human annotation process.

Introduction to IBM Watson Knowledge Studio

Figure 9-32. Review questions

1. False. Rule-based models are efficient for recognizing patterns in unstructured data. Machine learning models are efficient for complex text extraction.
2. B. Type system.
3. A. Natural Language Classifier.
4. True.

IBM Training

Review questions (cont.)

5. _____ groups words and phrases that should be treated equivalently by a model.
   A. Ruled-based model
   B. Ground truth
   C. Dictionary
   D. Machine learning model
   E. Type system

6. ____ is a collection of vetted data used to adapt a Watson solution to a particular domain.
   A. Ruled-based model
   B. Ground truth
   C. Dictionary
   D. Machine learning model
   E. Type system

Introduction to IBM Watson Knowledge Studio

Figure 9-33. Review questions (cont.)

5. C. Dictionary.
6. B. Ground truth.

IBM Training

Review answers

1. True or False: Rule-based models are efficient for complex text extraction that has variations.
2. _______ defines entities and relationships between entities that are interesting for your domain (mentions, entity types, relation types).
   A. Dictionary
   B. Type system
   C. Pre-annotation
   D. Ground truth
3. The following Watson service **cannot** be trained by using Knowledge Studio.
   A. Watson Natural Language Classifier
   B. Watson Natural Language Understanding
   C. Watson Discovery
4. True or False: A ruled based model can be used to pre-annotate documents to speed up the human annotation process.

Introduction to IBM Watson Knowledge Studio

Figure 9-34. Review answers

1. False. Rule-based models are efficient for recognizing patterns in unstructured data. Machine learning models are efficient for complex text extraction.
2. B. Type system.
3. A. Natural Language Classifier.
4. True.

IBM Training

Review answers (cont.)

5. _____ groups words and phrases that should be treated equivalently by a model.
   A. Ruled-based model
   B. Ground truth
   C. Dictionary
   D. Machine learning model
   E. Type system

6. ____ is a collection of vetted data used to adapt a Watson solution to a particular domain.
   A. Ruled-based model
   B. Ground truth
   C. Dictionary
   D. Machine learning model
   E. Type system

Introduction to IBM Watson Knowledge Studio

Figure 9-35. Review answers (cont.)

5. C. Dictionary.
6. B. Ground truth.

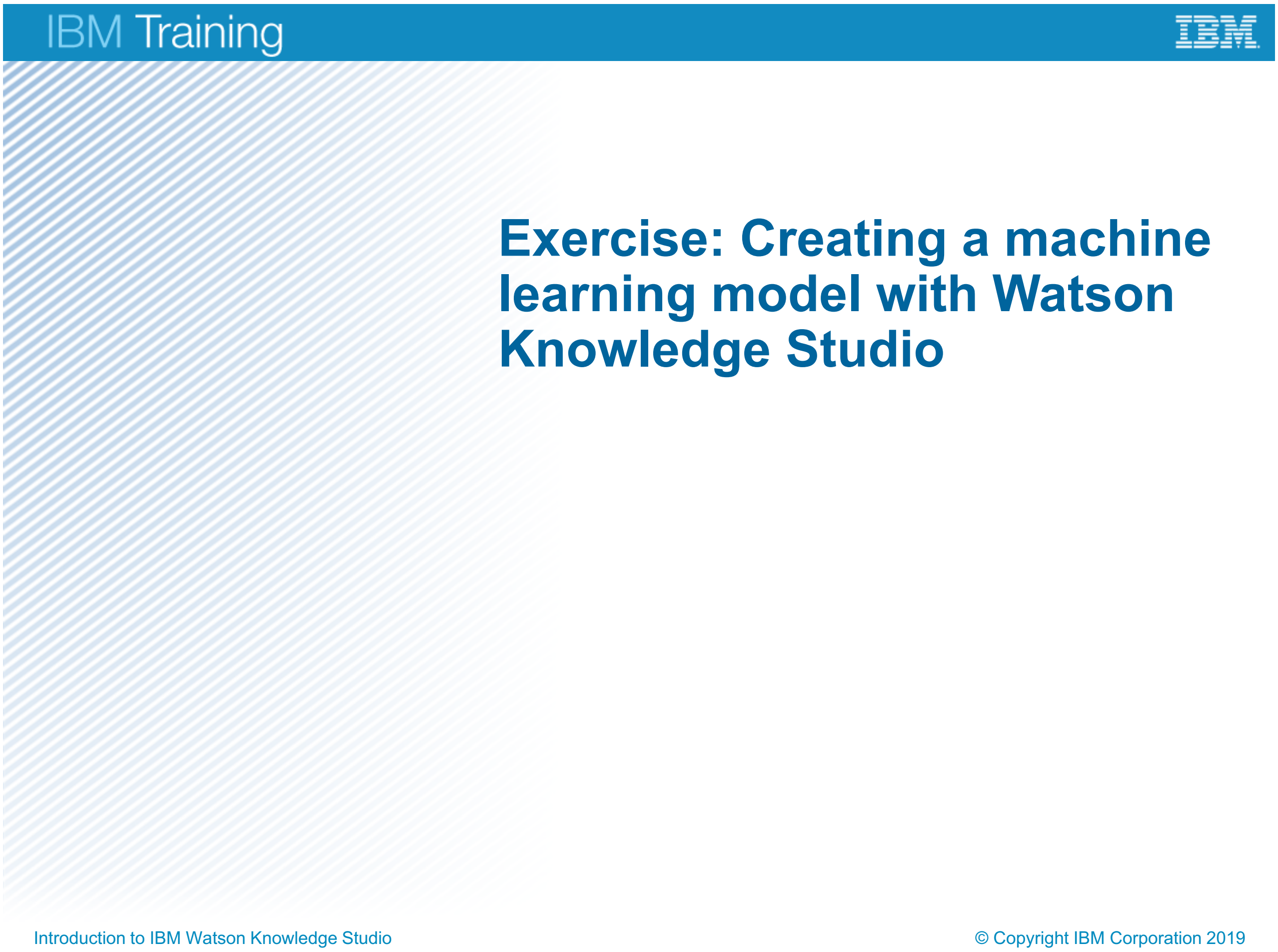


Figure 9-36. Exercise: Creating a machine learning model with Watson Knowledge Studio

IBM Training

Exercise objectives

- This exercise helps you understand the process for building a machine learning model that you can later deploy and use with other Watson services.
- After completing this exercise you should be able to:
  - Create a workspace for Knowledge Studio.
  - Configure the workspace resources.
  - Create document sets.
  - Pre-annotate documents.
  - Create tasks for human annotators.
  - Analyze the inter-annotator agreement and adjudicate conflicts.
  - Create machine learning models.
  - Deploy the model to Watson Natural Language Understanding or Watson Discovery.

Introduction to IBM Watson Knowledge Studio

Figure 9-37. Exercise objectives

After completing this exercise you should be able to:

- Create a workspace for Knowledge Studio.
- Configure the workspace resources.
- Create document sets.
- Pre-annotate documents.
- Create tasks for human annotators.
- Analyze the inter-annotator agreement and adjudicate conflicts.
- Create machine learning models.
- Deploy the model to Watson Natural Language Understanding or Watson Discovery.

Unit 10. Introduction to chatbots

Estimated time

00:30

Overview

This unit provides a high level introduction to chatbots, chatbot applications, and guidelines to consider when designing a chatbot.

IBM Training

Unit objectives

- Explain what a chatbot is.
- Describe the common applications of chatbots.
- Identify factors that drive the growing popularity of chatbots.
- Recognize the guidelines to consider when designing a chatbot.
- List examples of tools and services that you can use to create chatbots.

Introduction to chatbots

Figure 10-1. Unit objectives

10.1. Chatbots overview

IBM Training

Chatbots overview

Introduction to chatbots

Figure 10-2. Chatbots overview

IBM Training

Topics

- Chatbots overview
- Chatbots design
- Chatbots tools and services

Introduction to chatbots

Figure 10-3. Topics

IBM Training

What is a chatbot

- Software that interacts with the user in natural language.
- The application can have a chat interface.
- The conversation medium may be in text, speech, or both.
- Some example applications are customer support, sales activities, or entertainment.

Introduction to chatbots

Figure 10-4. What is a chatbot

A chatbot is software that interacts with the user through a chat interface by using natural language. The conversation might be in text, speech, or both. The chatbot chats with the user to assist with applications like customer support, sales activities, or entertainment.

IBM Training

Chatbots in the industry

Apple Siri, Microsoft Cortana, Google Assistant, and Amazon Alexa:

- Interact by using natural language and often in the form of speech.
- Support functions like setting alarms, running searches, and retrieving real-time information like news.

Introduction to chatbots

Figure 10-5. Chatbots in the industry

There are many examples of chatbots:

Siri: The intelligent assistant of Apple iPhone. Siri can help you with your schedule and find the nearest restaurant or movies that are available to watch.

Microsoft Cortana: Supported on various Microsoft Windows platforms. Also supported on Android and iOS. It can set reminders and answer questions by using information from the Bing search engine.

Google Assistant: Available on mobile and smart home devices. It provides the same functions that Google does: Searches the internet, schedules events and alarms, adjusts hardware settings on the user's device, and shows information from the user's Google account.

Amazon Alexa: Plays music and audiobooks, supports to-do lists, sets alarms, provides news and other real-time information like weather updates, traffic, and sports. Alexa can also control different smart devices that are part of a home automation system. You can give voice commands to control your home automation system like “Alexa, turn off the light in the living room.”.

References:

https://en.wikipedia.org/wiki/Google_Assistant

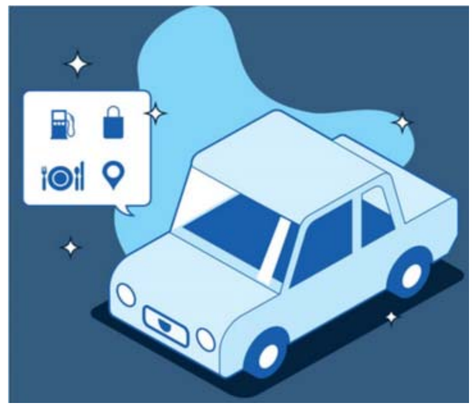
https://en.wikipedia.org/wiki/Cortana

https://en.wikipedia.org/wiki/Amazon_Alexa

IBM Training

Chatbots in the industry (cont.)

IBM Watson Assistant is tailored for domain-specific use cases.



Introduction to chatbots

Figure 10-6. Chatbots in the industry (cont.)

Another example of a chatbot is one that supports customers with their inquiries or even helps junior customer support agents with their job. IBM uses Watson technology (Watson Assistant) to create chatbots for those purposes.

References:

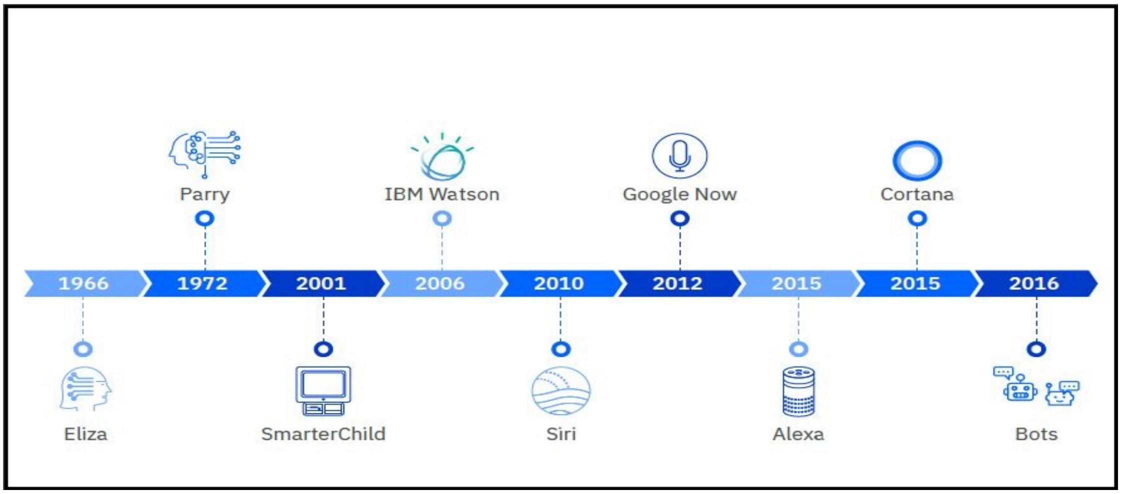
https://www.ibm.com/us-en/marketplace/watson-assistant-for-automotive

https://www.ibm.com/ro-en/marketplace/watson-assistant-for-industry

IBM Training

Chatbot history

- Bots passing the Turing test.
- ELIZA and PARRY did not use speech.
- SmarterChild was the predecessor to Siri, and was created by ActiveBuddy, Inc.



Introduction to chatbots

Figure 10-7. Chatbot history

1950s: The Turing test

The Turing test by Alan Turing was introduced as a standard of intelligence.

The standard is based on the capability of the computer program to impersonate or mimic a human in real time through a written conversation with a human judge. If the judge cannot distinguish between the program and a real human based on the conversation alone, the program passes the Turing test. Many programmers took the Turing test as a challenge.

1966: ELIZA

In 1966, Metzenbaum's program ELIZA tricked users into believing that they were having a conversation with a human. ELIZA passed the Turing test.

1972: PARRY

Another chatbot was invented around 1972 that was called PARRY. PARRY was more advanced than ELIZA. However, both ELIZA and PARRY could have a conversation only through text (not speech). Advances in natural language processing have led to the emergence of more capable chatbots since then.

2001: SmarterChild

In 2001, SmarterChild emerged. It was a chatbot that was used across SMS networks. It took advantage of the popularity of SMS and added the capability to understand natural language. It relied on database access for news, sports, entertainment, and other various topics. It is considered the precursor to Siri.

2006: IBM Watson

In 2006, IBM Watson started initial test runs. In 2011, Watson competed on the *Jeopardy!* quiz TV show. It understood and answered trivia questions, and won against human contestants (the Jeopardy! champions).

2010: Siri

Siri emerged in 2010 as a personal assistant feature of Apple iOS. Siri can search for answers on the web, create reminders, alarms, and events, call people, and much more.

2012: Google Now

In 2012, Google Now emerged, appearing as a feature in Google search. It delivered information to users based on predictions of what they might be interested in knowing.

2015: Amazon Alexa and Microsoft Cortana

In 2015, Amazon Alexa, the virtual assistant behind Amazon Echo, and Microsoft Cortana emerged.

2016: Chatbots

Facebook started a messaging platform enabling developers to create chatbots that can interact with Facebook users. Today, there are more than 33,000 chatbots on Facebook Messenger alone.

Figure 10-8. Why now

It seems that chatbots have become popular in recent years. Why now?

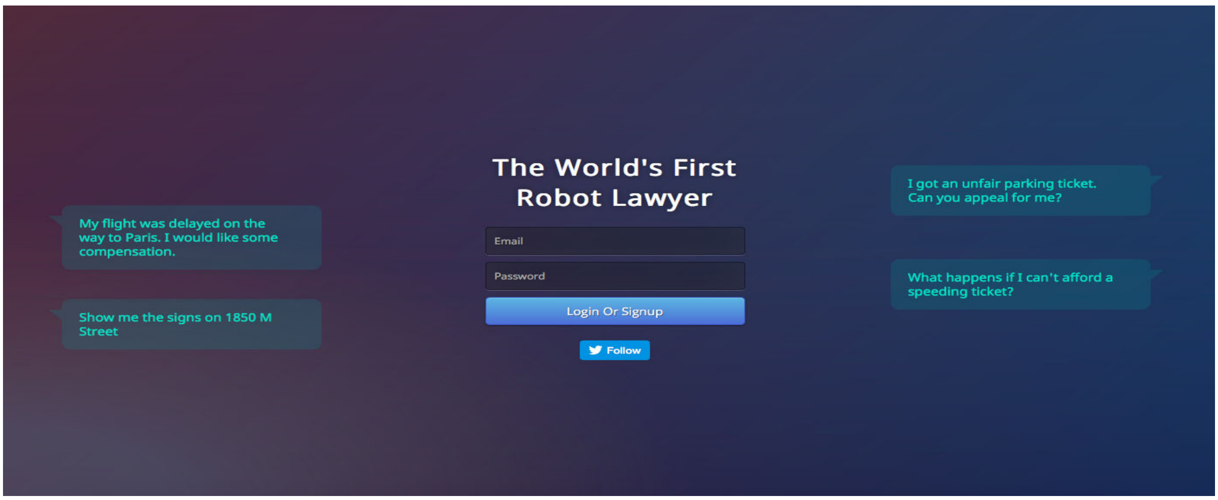
There are different reasons for the increase in the prevalence of chatbot:

- Rising popularity and high usage of messenger applications, such as WhatsApp, Facebook Messenger, and Slack.
- Recent breakthroughs in the AI field make it possible to develop intelligent chatbots that can communicate with users in human language and learn from their interaction with users. Chatbots can learn over time to provide more accurate and intelligent responses.
- Advancements in speech recognition make it possible to take human speech as input and enable users to have more natural interaction with computers.
- The growing popularity of cloud computing and APIs that provide integration capabilities make it possible for AI vendors to make AI services easily available to developers on the cloud. Developers can integrate chatbots into various applications and ensure reliable communication between the chatbot and the application by using APIs.

IBM Training

Popular chatbots

- DoNotPay chatbot for legal advice by Joshua Browder.
- Joshua Browder was named “Robin Hood of the internet” by the BBC.
- He used the IBM Watson Assistant service to improve the chatbot’s accuracy by 30%.



Introduction to chatbots

Figure 10-9. Popular chatbots

The DoNotPay chatbot is a popular chatbot. It was created by a 19-year old Stanford student who is named Joshua Browder, whom is called “Robin Hood of the internet” by the BBC.

Browder used a natural language interface to gather the data that is needed to complete the form. He also used the IBM Watson Assistant service to improve that chatbot’s accuracy by 30%.

IBM Training

Popular chatbots (cont.)

- Woebot in the medical domain.
- “Create the experience of a therapeutic conversation for all of the people that use the bot.”

Introduction to chatbots

Figure 10-10. Popular chatbots (cont.)

Woebot was created by a team of Stanford psychologists and AI experts.

Woebot uses short chat conversations, sentiment and tone analysis, and word games to help people who are looking for inexpensive therapy. The mission of the creators of Woebot is “to make mental health radically accessible to everyone”.

Woebot uses a combination of natural language processing, therapeutic expertise, personalized content, and a sense of humor to “create the experience of a therapeutic conversation for all of the people that use the bot.”

10.2. Chatbots design

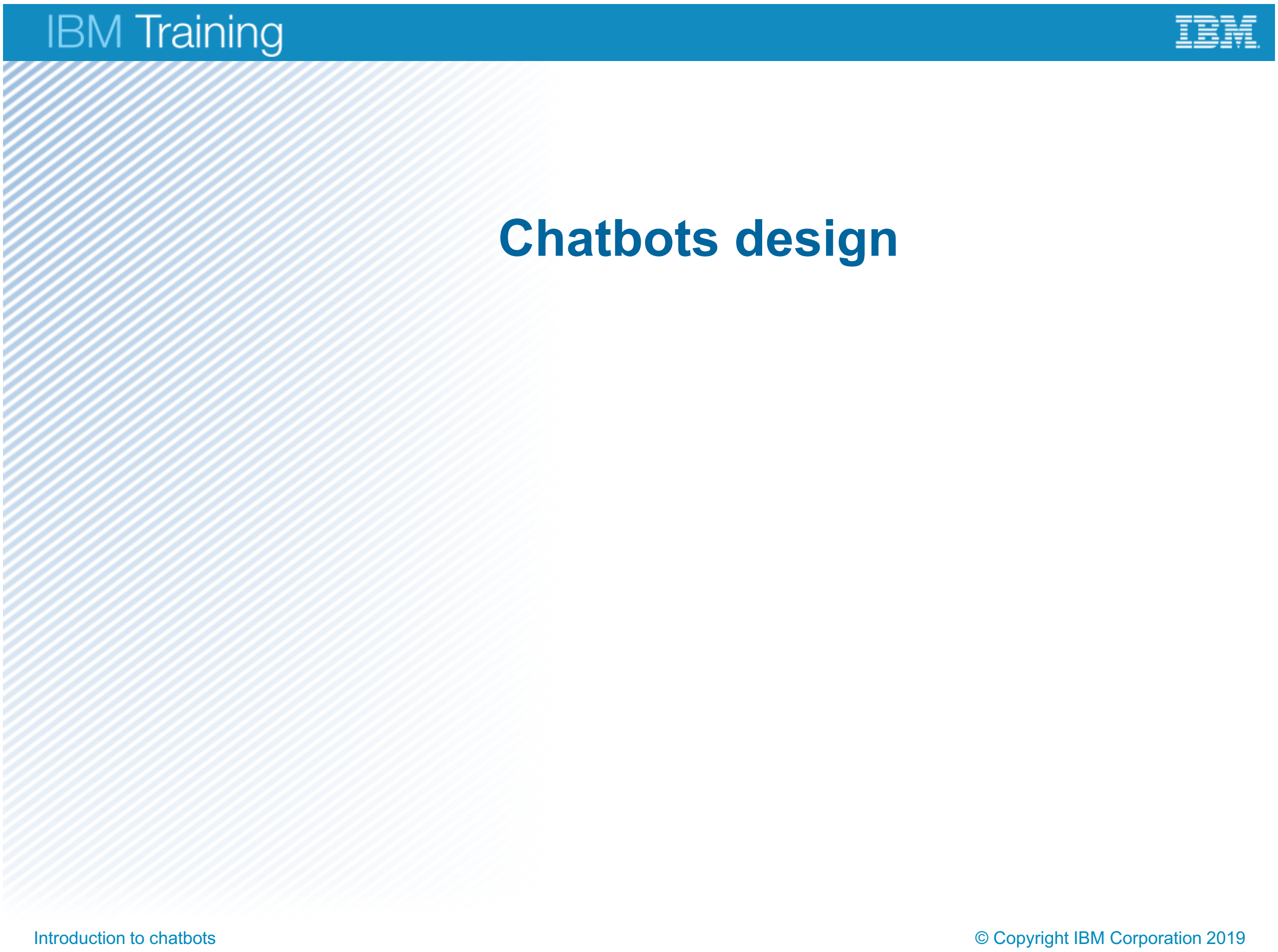


Figure 10-11. Chatbots design

IBM Training

Topics

- Chatbots overview
- Chatbots design
- Chatbots tools and services

Introduction to chatbots

Figure 10-12. Topics

IBM Training

Chatbot design

The basic goals of conversational interaction for chatbots are:

- Help users.
- Manage user expectations.
- Achieve the purpose of the solution.
- Keep the user engaged.
- Extract insights about the users.
- Reinforce the brand of the client.
- Increase sympathy and forgiveness in users.
- Make the solution look clever.

Introduction to chatbots

Figure 10-13. Chatbot design

Conversational interaction in the chat between the chatbot and the user takes place by exchanging natural language messages. Conversational interaction has an important role to play because it must help the customer understand how the chatbot can help them while managing expectations.

It is important to design the conversation correctly and appropriately for the type of chatbot solution and target audience. Not all solutions need the same amount of conversational interaction, but if you do not deliberately design the conversational interaction, you will not fully realize the potential of the solution to address the needs of your clients. If clients do not see enough value from the chatbot solution, they are unlikely to continue to use it.

The basic goals of conversational interaction are:

- How to help users and manage their expectations.
- Help achieve the purpose of the solution and drive value.
- Encourage the user to keep using the chatbot.
- Derive usable insights about the users.
- Reinforce the brand of the client.
- Increase sympathy and forgiveness in users.
- Make the solution look clever.

To achieve these goals, the dialog must be carefully designed to use the appropriate language and a suitable approach. Designing the chatbot is the part that is most removed from development. The result of this design is the overall impression of the chatbot on the user.

Here are key factors to consider in the conversational design:

- Positioning
- Tone and personality

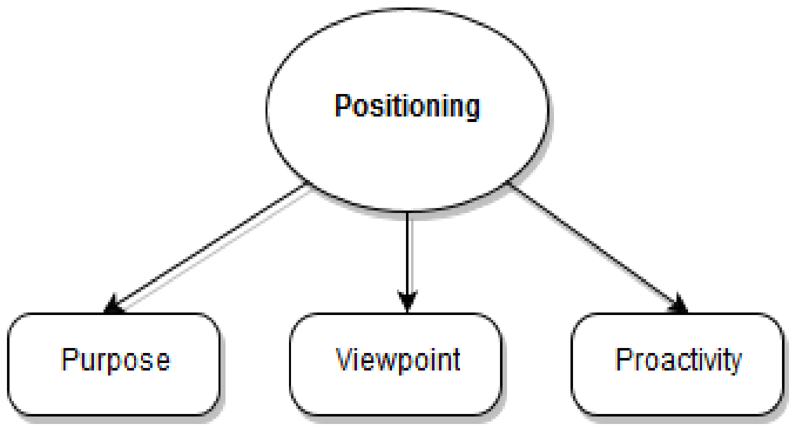
IBM Training

Chatbot design: Positioning

Positioning

The positioning of the chatbot is a collective term that covers these three aspects:

- Purpose
- Viewpoint
- Proactivity



Introduction to chatbots

Figure 10-14. Chatbot design: Positioning

To achieve your chatbot design goals, you must define different aspects about how to design your chatbot .

The first concept to consider is positioning. Positioning determines how the solution should relate to the user, and behave when interacting with the user. The positioning of the chatbot is a collective term that covers these three aspects:

- Purpose
- Viewpoint
- Proactivity

IBM Training

Chatbot design: Positioning (cont.)

Purpose: What is the purpose of the solution?

Examples:

- Clarify or answer the user's questions.
- Guide users to find the information they are looking for.
- Help users to complete a process.
- Collect the information that is needed to report a problem.

Introduction to chatbots

Figure 10-15. Chatbot design: Positioning (cont.)

Purpose: What is the purpose of the solution?

A conversational interaction should always have a clear purpose, that is, a well-defined job to do. For example:

- Clarify or answer the user's questions.
- Guide users to find the information they are looking for.
- Help users to complete a process.
- Collect the information that is needed to report a problem.

The purpose is the "job description" of the chatbot.

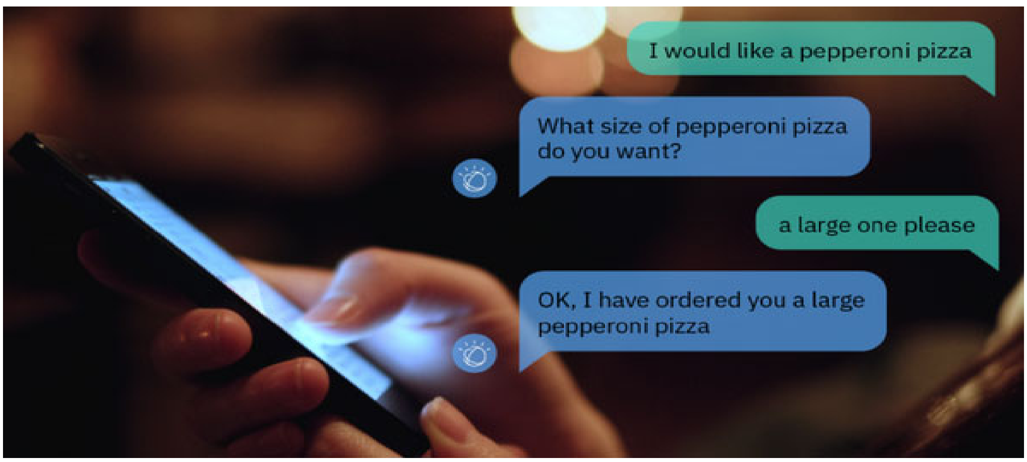
Chatbots, unlike standard application UI, tend to humanize interactions and they should be tailored to the needs of the users.

IBM Training

Chatbot design: Positioning (cont.)

Viewpoint: What role should the solution play in the relationship between the client and the user?

Examples: Assistant, friend, coach, motivator, salesperson, and others



Introduction to chatbots

Figure 10-16. Chatbot design: Positioning (cont.)

Viewpoint:

Viewpoint is the role that the chatbot solution should adopt in relation to the end user. As a designer, ask yourself:

- Who is the audience? (for example, students, prospect clients, users of your product, tourists, and so on)
- Who does the chatbot speak on behalf of? (for example, an employee who speaks on behalf of the company, as an advocate or friend to the user who speaks on behalf of the tourism organization, a tutor or teacher who speaks on behalf of a school or university, and so on.)

The relationship that the bot will have with the user should be decided upon in light of achieving the *purpose* previously determined: Does the chatbot represent a salesperson? an entertainer? a teacher? an enforcer or regulator? a companion or friend?

Viewpoint can be determined through these questions:

- What role should the solution play in the relationship between the client and the user?
- Should the chatbot present itself as an employee who speaks on behalf of the client and is aligned with the client interests?
- Should the chatbot present itself as an advocate or friend to the user who can access the client's information to help the user?

Some examples of roles or viewpoint are:

- Assistant
- Friend
- Coach
- Motivator
- Salesperson

Another question is, who is the audience? Are they students, prospect clients, users of a product, or others?

Reference:
https://www.ibm.com/blogs/watson/2017/12/3-types-of-business-chatbots-you-can-build/

IBM Training

Chatbot design: Positioning (cont.)

Proactivity:

- Level of proactiveness or reactiveness when engaging with user.
- A chatbot can be:
  - **Proactive** (lean forward): The chatbot reaches out to the user and asks questions.
  - **Reactive** (lean backward): The chatbot waits for the user to ask a question.
  - **Combination:** The chatbot uses both techniques.

Introduction to chatbots

Figure 10-17. Chatbot design: Positioning (cont.)

Proactivity is the degree to which the solution proactively engages and guides the user instead of reacting to the user, that is, waiting for the user to ask a question.

A chatbot can be:

- **Proactive:** The chatbot reaches out to the user by asking questions, offering information and choices, and guiding the user through processes.
- **Reactive:** Apart from a welcome message (and perhaps one or two other statements), your chatbot waits for the user to ask a question. It does not prompt, encourage, or suggest.
- **Combination:** In specific circumstances, the chatbot reaches out to the user, asking questions and guiding the user through processes. At other times, the chatbot waits for the user to ask a question.

IBM Training

Chatbot design: Positioning (cont.)

- **Example of a proactive chatbot:**
  - *Customer: When can I receive my new credit card?*
  - *Chatbot: An SMS will be sent to your phone after it is ready.*
  - *Chatbot: The bank is collaborating with XYZ to provide benefits to our credit card users. Would you be interested in knowing more?*

Introduction to chatbots

Figure 10-18. Chatbot design: Positioning (cont.)

Imagine a scenario where a chatbot is acting as a customer service employee on behalf of a bank.

The following conversation represents a proactive chatbot:

- Customer: When can I receive my new credit card?
- Chatbot: An SMS will be sent to your phone after it is ready.
- Chatbot: The bank is collaborating with XYZ to provide benefits to our credit card users. Would you be interested in knowing more?

The chatbot answers when asked, but also reaches out to the customer and offers suitable suggestions.

IBM Training

Chatbot design: Positioning (cont.)

- **Example of a reactive chatbot:**
  - *Customer: When can I receive my new credit card?*
  - *Chatbot: An SMS will be sent to your phone after it is ready.*

Introduction to chatbots

Figure 10-19. Chatbot design: Positioning (cont.)

The following conversation represents a *reactive* chatbot:

- Customer: When can I receive my new credit card?
- Chatbot: An SMS will be sent to your phone after it is ready.

The chatbot waits for the customer’s question, and answers only when asked.

IBM Training

Chatbot design: Tone and personality

- Represents the *voice* in which the chatbot solution speaks.
- **Examples:**
  - Informal and friendly tone: “Hi there, how can I help you?”
  - More formal, still friendly: “Good morning. How may I help you today?
  - Formal, not very friendly: “This is a service that is designed to answer your questions. How may I assist you with your inquiries?”

Introduction to chatbots

Figure 10-20. Chatbot design: Tone and personality

Tone and personality: Represents the voice in which the chatbot solution speaks. Do not confuse tone and personality with viewpoint, which is whom the chatbot speaks on behalf.

Examples:

- Informal and friendly tone: “Hi there, how can I help you?”
- More formal, still friendly: “Good morning. How may I help you today?”
- Formal, not very friendly: “This is a service that is designed to answer your questions. How may I assist you?”

IBM Training

Chatbot design: Tone and personality (cont.)

What happens if you do not define tone and personality?

The chatbot will still have them, but it likely will be incoherent, which leads to a random user experience.

Introduction to chatbots

Figure 10-21. Chatbot design: Tone and personality (cont.)

What happens if you do not define tone and personality?

All verbal expression has a tone, all conversation is held between two sides, with someone, or something with a personality in mind.

If you do not define tone and personality, the chatbot will still have them, but if they are not deliberately and explicitly determined the chatbot likely will be incoherent, which leads to a random user experience.

IBM Training

Chatbot design: Tone and personality (cont.)

- You can construct a complete character persona for your chatbot .
- A common feature that is often in chatbots is *humor*.
- Humor increases the user's understanding, forgiveness, and satisfaction.

Introduction to chatbots

Figure 10-22. Chatbot design: Tone and personality (cont.)

You should determine the kind of personality that your users like better or feel more comfortable interacting with when designing your chatbot.

Personality is affected by the choice of words for the chatbot’s responses.

You can construct a complete character persona for your chatbot.

A common feature that is often in chatbots is humor. Humor can increase the user's understanding and forgiveness when your chatbot cannot understand something. It might even delight the user and increase their satisfaction with the chatbot.

IBM Training

Chatbot design: Tone and personality for purpose

- **Example 1:** If the purpose is to engage users proactively and encourage them to act, a friendly, informal tone will be more successful than a formal or authoritative tone.
- **Example 2:** If the purpose is to provide information or guidance about a serious topic, a casual, chatty tone might undermine the credibility of the information that is provided.

Introduction to chatbots

Figure 10-23. Chatbot design: Tone and personality for purpose

Here are two examples of tone and personality for the purpose:

- If the purpose is to engage users proactively and encourage them to act, a friendly, informal tone will be more successful than a formal or authoritative tone.
- If the purpose is to provide information or guidance about a serious topic, a casual, chatty tone might undermine the credibility of the information that is provided.

IBM Training

Chatbot design: Tone and personality for viewpoint

- **Example 1:** If the viewpoint is that of an employee, the tone and personality should closely reflect the client's ideal staff member.
- **Example 2:** If the viewpoint is that of a partially or fully independent advocate or guide for the user, then the tone and personality should not be that of enforcer or authority figure. Instead, it should be more like an advisor, assistant, or coach.

Introduction to chatbots

Figure 10-24. Chatbot design: Tone and personality for viewpoint

Here are some examples for tone and personality for viewpoint:

- If the viewpoint is that of an employee, the tone and personality should closely reflect the client's ideal staff member.
- If the viewpoint is that of a partially or fully independent advocate or guide for the user, then the tone and personality should not be that of enforcer or authority figure. Instead, it should be more like an advisor, assistant, or coach.

IBM Training

Chatbot design: Tone and personality for proactivity

- If the solution is highly proactive, either an overly formal or overly casual and chatty tone may have an unpleasant effect because the former might feel artificial and the latter is likely to become annoying.
- Tone and personality for special cases: If a response uses offensive language:
  - The chatbot can shift to an “enforcer” viewpoint and adopt a more formal tone.
  - You may keep a count of the number of sensitive inputs and respond with an increasingly strong-worded response.
  - You may shut down the chat (in the case of threats and offensive language).

Introduction to chatbots

Figure 10-25. Chatbot design: Tone and personality for proactivity

If the solution is highly proactive, either an overly formal or overly casual and chatty tone may cause a negative reaction because the former might feel artificial and the latter is likely to become annoying.

Consider the appropriate tone and personality for special cases. There might be special cases where the tone response might differ from the rest of the solution. For example, if there are responses that use offensive language:

- The chatbot can shift to an “enforcer” viewpoint and adopt a more formal tone.
- You may keep a count of the number of sensitive inputs and respond with an increasingly strong-worded response.
- You may shut down the chat (in the case of threats and offensive language).

IBM Training

Chatbot design: Design issues

1. Understand the limitations of a chatbot.
2. Acknowledge limitations and do not be afraid to have the chatbot say, “I don’t know”, and give some suggestions of questions that they can ask.
3. Handle frustration. “I can see this isn’t going very well. Would you like to talk to a real person?”

Introduction to chatbots

Figure 10-26. Chatbot design: Design issues

Some of the issues that developers or designers face while designing chatbots include the following ones:

1. **Trying to do too much within a single chatbot, which can lead to a loss of value:** You must understand the limitations of a chatbot. It must have a narrow scope that is well-defined and well-bounded. It is not a chitchat.
2. **Users will be disappointed by incorrect or nonsensical responses:** To avoid this issue, acknowledge limitations and do not be afraid to have the chatbot say, “I don’t know”, and give some suggestions of questions that they can ask.
3. **Detect frustration, and handle it:** If the users are frustrated, this issue should be addressed. A good example response can be: “I can see that this isn’t going very well. Would you like to talk to a real person?” Some interactions are too complex for a software solution to handle. Detecting situations that are out of the scope of your chatbot and handing them off to a human quickly (before you waste the user’s time) is a better approach.

10.3. Chatbots tools and services

IBM Training

Chatbots tools and services

Introduction to chatbots

Figure 10-27. Chatbots tools and services

IBM Training

Topics

- Chatbots overview
- Chatbots design
- Chatbots tools and services

Introduction to chatbots

Figure 10-28. Topics

IBM Training

Chatbots tools and services

Here are examples of some available tools and services:

- IBM Watson Assistant service
- Dialogflow
- Microsoft Language Understanding Intelligent Service (LUIS)
- ChatScript
- Chatfuel

Introduction to chatbots

Figure 10-29. Chatbots tools and services

You can create chatbots by using cognitive services to help you design how your chatbot understands user's input and replies to the user. You can also use software to configure a chatbot that is already trained and programmed to answer specific questions from users in areas such as customer support.

Here are examples of some available tools and services:

- IBM Watson Assistant service.
- Dialogflow: Google-owned developer of human-computer interaction technologies that are based on natural language conversations.
- Microsoft Language Understanding Intelligent Service (LUIS).
- ChatScript: Written in C++.
- Chatfuel: A platform for creating AI chatbots for Facebook.

References:

https://chatfuel.com/

https://en.wikipedia.org/wiki/ChatScript

https://dialogflow.com/

https://en.wikipedia.org/wiki/Dialogflow

https://www.luis.ai/

IBM Training

Unit summary

- Explain what a chatbot is.
- Describe the common applications of chatbots.
- Identify factors that drive the growing popularity of chatbots.
- Recognize the guidelines to consider when designing a chatbot.
- List examples of tools and services that you can use to create chatbots.

Introduction to chatbots

Figure 10-30. Unit summary

IBM Training

Review questions

1. Chatbots can be used for:
   - A. Customer support
   - B. Entertainment
   - C. Sales assistant
   - D. All the above
2. True or False: Advancements in NLP and AI are key factors in the emergence of chatbots.
3. True or False: A chatbot that reaches out to the user by asking questions and offering information is considered reactive by design.
4. True or False: There are two types of viewpoint: lean forward and lean backward.

Figure 10-31. Review questions

1. D. All the above.
2. True.
3. False.
4. False.

IBM Training

Review questions (cont.)

5. ___ is the role that the chatbot solution should adopt in relation to the end user, for example customer support operator, coach, friend, or teacher.
   A. Viewpoint
   B. Purpose
   C. Positioning
   D. Proactivity
6. Represents the voice in which the chatbot solution speaks, for example, informal, friendly, or formal.
   A. Viewpoint
   B. Purpose
   C. Tone and personality
   D. Positioning
   E. Proactivity

Introduction to chatbots

Figure 10-32. Review questions (cont.)

5. A. Viewpoint.
6. C. Tone and personality.

IBM Training

Review answers

1. Chatbots can be used for:
   A. Customer support
   B. Entertainment
   C. Sales assistant
   D. All the above

   The answer is D.
2. True or False: Advancements in NLP and AI are key factors in the emergence of chatbots.
3. True or False: A chatbot that reaches out to the user by asking questions and offering information is considered reactive by design.
4. True or False: There are two types of viewpoint: lean forward and lean backward.
   False: There are two types of **proactivity:** lean forward and lean backward.

Introduction to chatbots

Figure 10-33. Review answers

1. D. All the above.
2. True.
3. False.
4. False. There are two types of *proactivity*: lean forward and lean backward.

IBM Training

Review answers (cont.)

5. ___ is the role that the chatbot solution should adopt in relation to the end user, for example customer support operator, coach, friend, or teacher.
   A. Viewpoint
   B. Purpose
   C. Positioning
   D. Proactivity
6. Represents the voice in which the chatbot solution speaks, for example, informal, friendly, or formal.
   A. Viewpoint
   B. Purpose
   C. Tone and personality
   D. Positioning
   E. Proactivity

Introduction to chatbots

Figure 10-34. Review answers (cont.)

5. A. Viewpoint.
6. C. Tone and personality.

Unit 11. Introduction to IBM Watson Assistant

Estimated time

01:00

Overview

This unit covers the core concepts that you need to understand to build a chatbot with IBM Watson Assistant.

IBM Training

Unit objectives

- Explain assistants and skills.
- Explain intents.
- Explain entities.
- Explain context variables.
- Describe how the nodes in a dialog are triggered.
- Describe how the dialog flow is processed.
- Describe the features that can be used to enrich the chatbot.

Introduction to IBM Watson Assistant

Figure 11-1. Unit objectives

11.1. Chatbot components

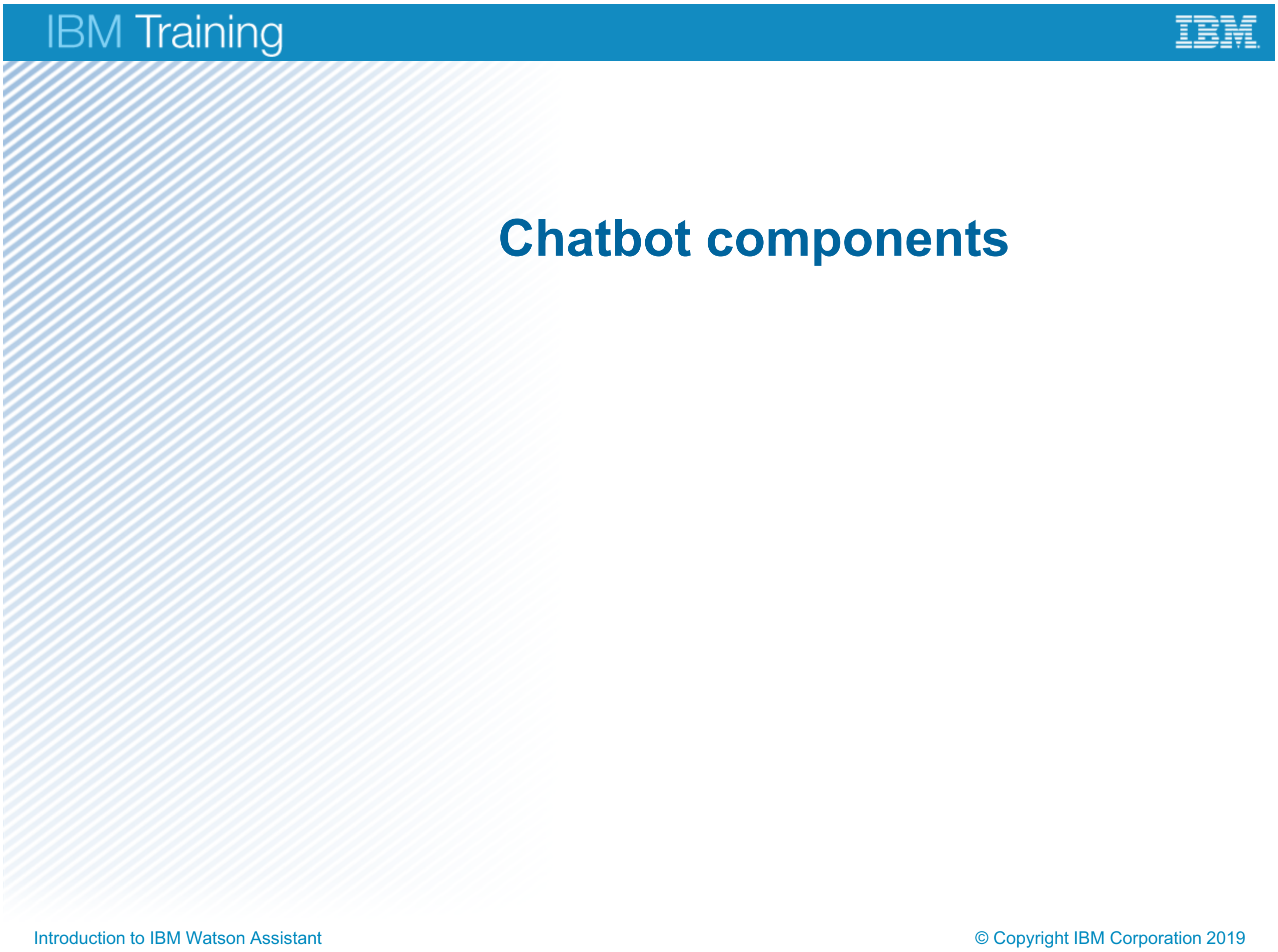


Figure 11-2. Chatbot components

IBM Training

Topics

- Chatbot components
- Watson Assistant components
- Enriching the chatbot

Introduction to IBM Watson Assistant

Figure 11-3. Topics

IBM Training

Chatbot components

- Most chatbot functions are divided into:
  - Intents
  - Entities
  - Dialog
- The dialog component is optional because some chatbots are developed to answer user questions in a question and answer manner that is similar to the approach that is used to answer frequently asked questions (FAQs).

Introduction to IBM Watson Assistant

Figure 11-4. Chatbot components

No matter what the specific purpose of your chatbot is, some core fundamentals are always involved. Most chatbot functions are divided into:

- Intents
- Entities
- Dialog

Each of these functions will be explained in detail and how they apply to the IBM Watson Assistant Service in the next section.

The dialog component is optional because some chatbots are developed to answer user questions in a question and answer manner that is similar to the approach that is used to answer frequently asked questions (FAQs).

These chatbots need to understand only what the user is asking and reply with the proper answer. Chatbots that are meant to answer FAQs do not have to engage in a conversation with the user and therefore they do not need a dialog component.

11.2. Watson Assistant components

IBM Training

Watson Assistant components

Introduction to IBM Watson Assistant

Figure 11-5. Watson Assistant components

IBM Training

Topics

- Chatbot components
- Watson Assistant components
- Enriching the chatbot

Figure 11-6. Topics

IBM Training

Watson Assistant components

Watson Assistant main components:

- Assistants
- Skills
- Intents
- Entities
- Dialog

Introduction to IBM Watson Assistant

Figure 11-7. Watson Assistant components

The Watson Assistant main components are:

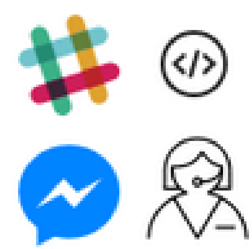
- Assistants
- Skills
- Intents
- Entities
- Dialog

IBM Training

Watson Assistant components (cont.)

Assistant:

- A cognitive bot that you can customize for your business needs.
- Customize the assistant by adding one or more skills to it to satisfy your customers' goals.
- The assistant can be deployed through multiple interfaces, including:
  - Messaging channels, like Slack and Facebook Messenger.
  - A simple chat widget on a website or on an existing company web page.
  - A custom application, such as a mobile app or a robot with a voice interface.



Introduction to IBM Watson Assistant

Figure 11-8. Watson Assistant components (cont.)

Assistants

An assistant is a cognitive bot that you can customize for your business needs and deploy across multiple channels to help your customers where and when they need it.

You customize the assistant by adding to it the skills that it needs to satisfy your customers' goals.

Add a dialog skill that can understand and address questions or requests with which your customers typically need help. You provide information about the subjects or tasks that your users ask about and how they ask about them, and the service dynamically builds a machine learning model that is tailored to understand the same and similar user requests.

You can deploy the assistant through multiple interfaces, including:

- Existing messaging channels, such as Slack and Facebook Messenger.
- You can give your users access to the assistant through a simple chat widget that you publish to a website or add to an existing company web page.
- You can design a custom application that incorporates it by making direct calls to the underlying APIs.

IBM Training

Watson Assistant components (cont.)

Skill (formerly Workspace):

- Sometimes referred to as *dialog skills*.
- Acts as a container for all of the artifacts and training data.
- A skill contains the following types of artifacts:
  - Intents
  - Entities
  - Dialog
- A skill uses the unique training data to build a machine learning model that can recognize these and similar user inputs. Each time that you add or change the training data, the training process is triggered to ensure that the underlying model stays up-to-date.

Introduction to IBM Watson Assistant

Figure 11-9. Watson Assistant components (cont.)

Skill

Formerly called Workspaces. It is sometimes referred to as *dialog skill*.

A skill acts as a container that contains the training data and logic that enables an assistant to help your customers.

A skill contains the following types of artifacts:

- Intents
- Entities
- Dialog

As you add information, the skill uses this unique data to build a machine learning model that can recognize these and similar user inputs. Each time that you add or change the training data, the training process is triggered to ensure that the underlying model stays up-to-date as your customer needs and the topics they want to discuss change.

IBM Training

Watson Assistant components (cont.)

Intent:

- Represents the purpose of or goal of a user's input.
- Tends to be general, such as a question about business location or a bill payment.
- For intent creation:
  - Plan the intents for application. Define an intent for each type of user request that you want your application to support.
  - Teach Watson about your intents. Supply many examples of user input and indicate the intents to which they map.
- In the tool, the name of an intent is always prefixed with '#'. For example, #payment or #inquiry.

Introduction to IBM Watson Assistant

Figure 11-10. Watson Assistant components (cont.)

Intent

An *intent* represents the purpose of a user's input, such as a question about business locations or a bill payment.

Plan the intents for an application: Consider what your customers might want to do and what you want your application to be able to handle on their behalf. For example, you might want your application to help your customers make a purchase. If so, you can add a #buy_something intent.

You define an intent for each type of user request that you want your application to support.

Teach Watson about your intents: After you decide the business requests that you want your application to handle for your customers, you must teach Watson about them. For each business goal (such as #buy_something), you must provide at least 10 examples of utterances that your customers typically use to indicate their goal. For example, "I want to make a purchase."

Ideally, find real-world user utterance examples that you can extract from existing business processes. The user examples should be tailored to your specific business. For example, if you are an insurance company, your user examples might look more like this, “I want to buy a new XYZ insurance plan.”

To train the dialog skill to recognize your intents, supply many examples of user input and indicate to which intents they map. The examples that you provide are used by the service to build a machine learning model that can recognize the same and similar types of utterances and map them to the appropriate intent.

In the tool, the name of an intent is always prefixed with the # character.

Start with a few intents, and test them as you iteratively expand the scope of the application.

IBM Training

Watson Assistant components (cont.)

Intent examples:

- “Good morning” -> #greeting.
- “Where can I find the nearest restaurant?” -> #location_info.
- “Where can I pay my electric bill?” -> #location_info.
- Remember: Intents represent what the user wants to achieve: a goal, an action, or verbs.

Introduction to IBM Watson Assistant

Figure 11-11. Watson Assistant components (cont.)

Intent examples:

- “Good morning” -> #greeting
- “Where can I find the nearest restaurant?” -> #location_info
- “Where can I pay my electric bill?” -> #location_info

Remember: Intents represent what the user wants to achieve: a goal, an action, or verbs.

IBM Training

Watson Assistant intents features

Content catalog:

- An easy way to add common intents to your Watson Assistant dialog skill.
- Contains prebuilt common intents that you can add to your application rather than building your own.
- For example, most applications require a greeting intent that starts a dialog with the user. You can add the **General** content catalog to add an intent that greets the user and does other useful things, like end the conversation.

Introduction to IBM Watson Assistant

Figure 11-12. Watson Assistant intents features

A *content catalog* is provided that contains prebuilt common intents that you can add to your application rather than building your own. For example, most applications require a greeting intent that starts a dialog with the user. You can add the **General** content catalog to add an intent that greets the user and does other useful things, like end the conversation.

Important: Content catalog intents are meant to provide a starting point, and not meant to be fully built-out for production use. Review and expand on these intents to make them better suited to how your application uses them.

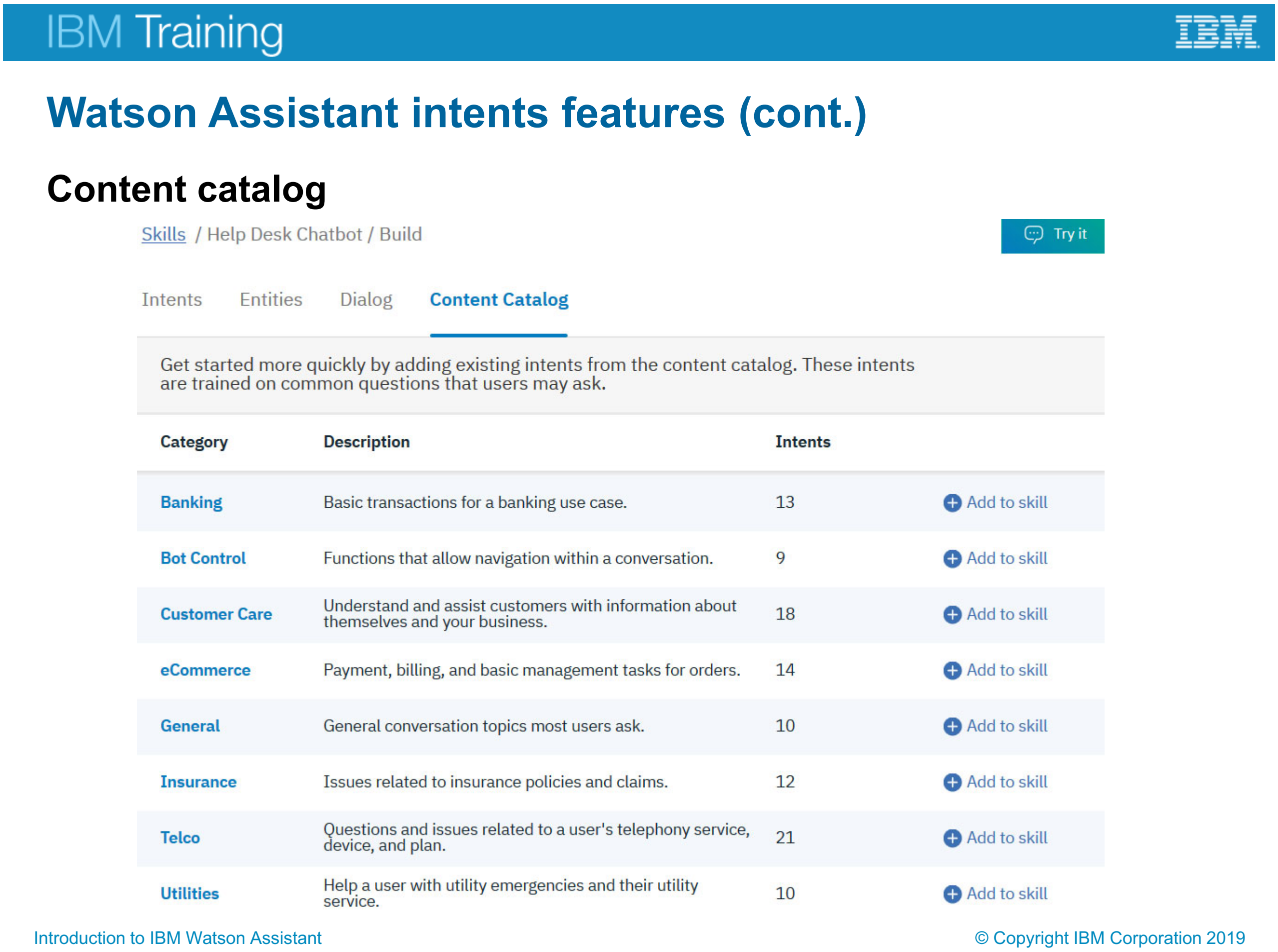


Figure 11-13. Watson Assistant intents features (cont.)

This slide shows the content catalog that is built into the Watson Assistant tool. The content catalog provides some general categories, such as General and Bot Control, which are good starters for anyone creating a chatbot. There also are starters for some use cases in certain industries, such as Banking, e-commerce, Telco, and Insurance, or general use cases that are found in most industries, such as Customer Care and Utilities. Within each category is a set of intents that is related to that category, which you can choose and add. After adding them, you can customize each intent by adding more examples to it or removing some examples.

IBM Training

Watson Assistant components (cont.)

Entity

- Represents a term or object that is relevant to the user's intent and that provides a specific context for an intent.
- If intents represent *verbs* (the action a user wants to do), entities represent *nouns* (the object of, or the context for, that action).
- By recognizing the entities that are mentioned in the user's input, the service can reply with a more targeted response or perform a required action.
- In the tool, the name of an entity is always prefixed with '@'. For example, @street or @contact_info.

Introduction to IBM Watson Assistant

Figure 11-14. Watson Assistant components (cont.)

Entity

Entities represent information in the user input that is relevant to the user's purpose.

If intents represent verbs (the action a user wants to do), entities represent nouns (the object of, or the context for, that action). For example, when the *intent* is to get a weather forecast, the relevant location and date *entities* are required before the application can return an accurate forecast.

Entities represent a class of object or a data type that is relevant to a user's purpose. By recognizing the entities that are mentioned in the user's input, the service can reply with a more targeted response or perform a required action.

IBM Training

Watson Assistant entity evaluation

- The Assistant service looks for terms in the user input by using exact matching of values, synonyms, or patterns that are defined for the entities.
- Synonym entity:
  - Specify a bunch of synonyms, that is, words that have exact or similar meanings.
  - For example, for the entity value **Blue**, its synonyms can be **Aqua**, **Navy**, and **Cyan.**
- Pattern entity:
  - You specify a regular expression that defines the pattern for mentions.
  - For example, for email patterns such as *name@ibm.com*, the pattern can be:

```
\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b
```

Introduction to IBM Watson Assistant

Figure 11-15. Watson Assistant entity evaluation

The Assistant service looks for terms in the user input that match the values, synonyms, or patterns that you define for the entity:

- **Synonym entity**: Synonyms are words or phrases that mean exactly or nearly the same as the corresponding entity. You define a category of terms as an entity (color), and then one or more values in that category (blue). For each value, you specify a bunch of synonyms (aqua, navy, and cyan). At run time, the service recognizes terms in the user input that exactly match the values or synonyms that you defined for the entity as mentions of that entity.
- **Pattern entity**: You define a category of terms as an entity (contact_info), and then define one or more values in that category (email). For each value, you specify a regular expression that defines the textual pattern of mentions of that value type. For an email entity value, you might want to specify a regular expression that defines a text@text.com pattern.
  This email regex example can capture all mentions of emails in their proper format.
  At run time, the service looks for patterns matching your regular expression in the user input, and identifies any matches as mentions of that entity.

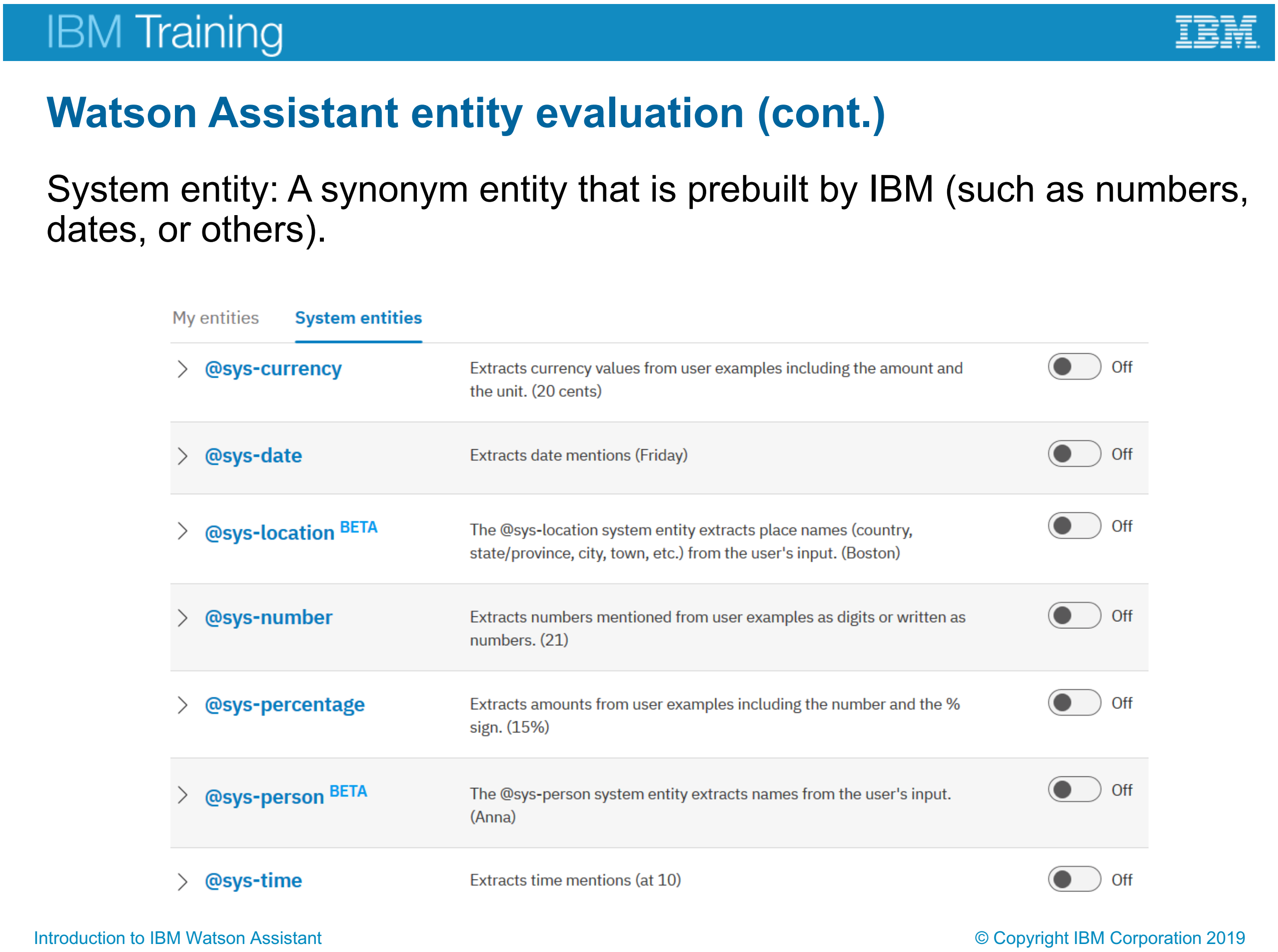


Figure 11-16. Watson Assistant entity evaluation (cont.)

System entity: A synonym entity that is prebuilt for you by IBM. They cover commonly used categories, such as numbers, dates, and times. You simply enable a system entity to start using it.

IBM Training

Watson Assistant entities features

Fuzzy matching:

- Can be enabled in entity definitions.
- By enabling fuzzy matching, the service can recognize terms similar to the entity value and synonyms without requiring an exact match.
- Detects entities in different grammatical forms. For example:
  - bananas > banana
  - running > run.
- Detects misspelled entities. For example:
  - girafe > giraffes.
- Detects partial matching.

Introduction to IBM Watson Assistant

Figure 11-17. Watson Assistant entities features

Fuzzy matching

If you want the service to recognize terms with a syntax that is similar to the entity value and synonyms that you specify but without requiring an exact match, enable the **fuzzy matching** feature.

This feature is available for all supported languages of Watson Assistant.

Detects entities in different grammatical forms: If your entity value is banana and the user inputs bananas, it can map them to each other.

Detects misspelled entities: If your entity value is giraffes and the user inputs girafe (misspelled), the service can map them to each other.

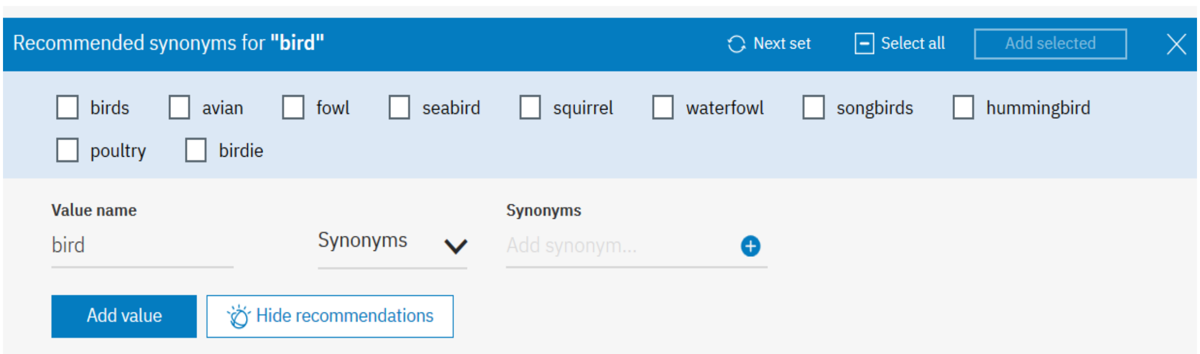
Detects partial matching.

IBM Training

Watson Assistant entities features (cont.)

Synonym recommendations

- The service can recommend synonyms for entity values.
- Used only for synonym entity values, not patterns.
- This feature finds related synonyms and uses natural language processing techniques to identify words that are similar to the existing synonyms in your entity value.



Introduction to IBM Watson Assistant

Figure 11-18. Watson Assistant entities features (cont.)

Entity synonym recommendations

The Watson Assistant service can recommend synonyms for your entity values. The recommender finds related synonyms based on contextual similarity that is extracted from a vast body of existing information, including large sources of written text, and uses natural language processing techniques to identify words that are similar to the existing synonyms in your entity value.

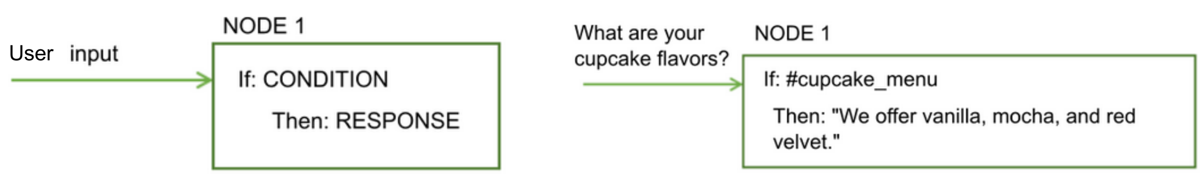
The figure shows some recommendations for the entity value “bird”.

IBM Training

Watson Assistant components (cont.)

Dialog

- A dialog is a branching conversation flow that defines how your application responds when it recognizes the defined intents and entities.
- The dialog is made up of nodes that are chained together in a tree structure (graphically).
- Each node includes conditions for the node to be active, and also an output object that defines the response that is provided.
- Think of the node as an if-then construction: If this condition is true, then return this response.



Introduction to IBM Watson Assistant

Figure 11-19. Watson Assistant components (cont.)

Dialog

A dialog is a branching conversation flow that defines how your application responds when it recognizes the defined intents and entities.

The dialog is made up of nodes that define steps in the conversation.

Dialog nodes are chained together in a tree structure (graphically).

Each node includes conditions for the node to be active, and also an output object that defines the response that is provided.

Think of the node as an if-then construction: If this condition is true, then return this response.

Condition: Specifies the information that must be present in the user input for this node in the dialog to be triggered.

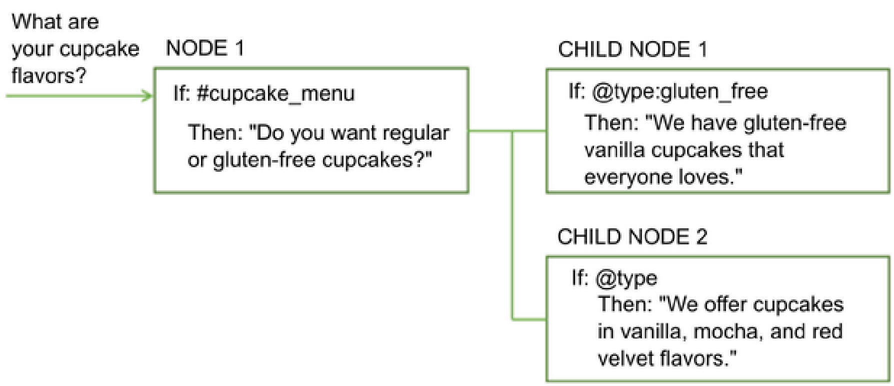
Response: The utterance that the service uses to respond to the user.

IBM Training

Watson Assistant components (cont.)

Dialog (cont.)

- A single node with one condition and response can handle simple user requests.
- To handle more complex tasks, you can add child nodes to ask the user for additional information.



Introduction to IBM Watson Assistant

Figure 11-20. Watson Assistant components (cont.)

Dialog

A single node with one condition and response can handle simple user requests.

To handle more complex tasks, you can add child nodes to ask the user for additional information.

A child node is processed after its parent node. NODE 1 is the parent node for CHILD NODE 1 and CHILD NODE 2.

Some useful definitions:

- A root node is a node that does not depend on other nodes. In our example, NODE 1 is a root node.
- A child node is a node that depends on another node, its parent node. The parent node is processed first, and based on that processing, the child node is either processed or not.

IBM Training

Watson Assistant components (cont.)

Context variables

- The dialog in a dialog skill is stateless. It does not retain information from one interaction with the user to the next.
- Context variables allow storing information that is passed back and forth across different dialog nodes, and between the dialog and application.
- A context variable is defined in a node. It can have a default value that can be changed.
- For example, if you identify the name of your user in the dialog flow, you can store the information in a context variable and retrieve it anytime that you want to call your user by name.

Introduction to IBM Watson Assistant

Figure 11-21. Watson Assistant components (cont.)

Context: As in a real-life conversation, context matters. The dialog context is the mechanism for passing information between the dialog and your application code. Context variables allow you to store information that is passed back and forth across different dialog nodes. For example, if you identify the names of your users in the conversation flow, you can store the information in the context and retrieve it anytime that you want to call your user by name.

IBM Training

Watson Assistant components (cont.)

Context variables (cont.)

- Context variables (*username* variable) are stored in context objects that are described as a JSON entry in the node.

```
"context": {
"username": "<? @sys-person.literal ?>"
}
```

- In the tool, the context variable is always prefixed with '*$*', for example, $username.
- The context object can be handled in the context editor interface for simplicity.

Introduction to IBM Watson Assistant

Figure 11-22. Watson Assistant components (cont.)

In this JSON object, there is a context object that contains the context variable **username**. If you identify the names of your users in the conversation flow, you can store the information in the context variable **username** as defined and retrieve it anytime that you want to call your user by name. This is a simple example. You can learn how to define different types of context variables at https://console.bluemix.net/docs/services/assistant/dialog-runtime.html#context-var-json.

IBM Training

Watson Assistant components (cont.)

Conditions

- Logical expressions that are evaluated to true or false.
- A node condition determines whether that node is used in the conversation.
- Conditions usually evaluate the *intents* and *entities* that are identified in the user responses, but they can also evaluate information that is stored in the *context*.

Introduction to IBM Watson Assistant

Figure 11-23. Watson Assistant components (cont.)

Conditions

Conditions are logical expressions that are evaluated to true or false. A node condition determines whether that node is used in the conversation or to choose among the possible responses to the user.

Conditions usually evaluate the intents and entities that are identified in the user responses, but they can also evaluate information that is stored in the context.

IBM Training

Watson Assistant components (cont.)

Responses

- The dialog response defines how to reply to the user.
- The dialog response consists of the messages that are based on the identified intents, entities, or context that are communicated to the user when the dialog node is activated.
- You can add variations to the response for a more natural experience.

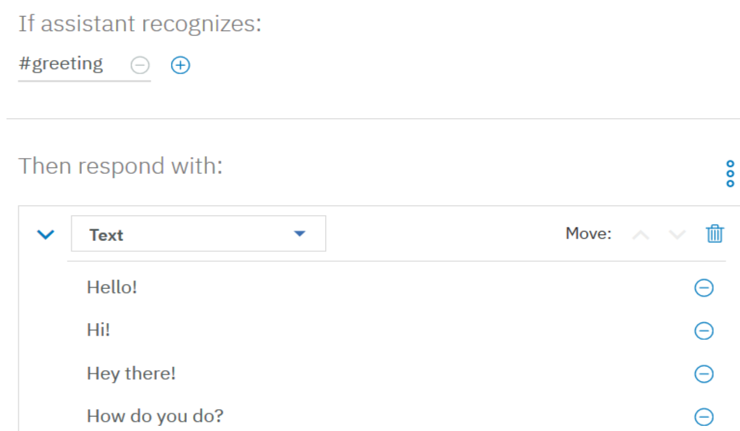


Figure 11-24. Watson Assistant components (cont.)

Responses

Responses are messages that are based on the identified intents and entities that are communicated to the user when the dialog node is activated. You can add variations of the response for a more natural experience, or add conditions to pick one response out of many in the same dialog node.

The figure shows an example of adding different variations for greetings if the node is triggered by a greeting intent from the user.

IBM Training

Watson Assistant dialog features

Rich responses

- In addition to the default response type of **text**, you can return other interactive response types to enhance the user experience.
- The following response types are supported:
  - Image: Embeds an image into the response.
  - Option: Adds a list of one or more options.
  - Pause: Forces the application to wait for a specified number of milliseconds before continuing with processing.

Introduction to IBM Watson Assistant

Figure 11-25. Watson Assistant dialog features

Rich responses

In addition to the default response type of *text*, you can return responses with multimedia or interactive elements, such as images or clickable buttons to simplify the interaction model of your application and enhance the user experience.

The following response types are supported:

- Image: Embeds an image into the response.
- Option: Adds a list of one or more options. When a user clicks one of the options, an associated user input value is sent to the service. How options are rendered can differ depending on where you deploy the dialog. For example, in one integration channel, the options might be displayed as clickable buttons, but in another they might be displayed as a drop-down list.
- Pause: Forces the application to wait for a specified number of milliseconds before continuing with processing. You can choose to show an indicator that the dialog is working on typing a response. Use this response type if you need to perform an action that might take some time.

IBM Training

Watson Assistant dialog features (cont.)

Multiple (conditional) responses

- In a single dialog node, you can provide different responses, each one triggered by a different condition. Use this approach to address multiple scenarios in a single node.
- **Example:** The service uses information that it collected earlier about the user's location to tailor its response, and provide information about the store nearest the user.

"Where are you located?"

Trigger: #store_locations

Response:
If $state == "MA"
"Our flagship store is located at 20 State St. in Boston."
If $state == "NH"
"You can find us in the Pheasant Lane Mall in Nashua."
If $state == "ME"
"We are located at 200 Commercial St. in Portland."

Else
"We have store locations throughout New England."

Introduction to IBM Watson Assistant

Figure 11-26. Watson Assistant dialog features (cont.)

Multiple (conditional) responses

In a single dialog node, you can provide different responses, each one triggered by a different condition. Use this approach to address multiple scenarios in a single node.

In this example, the service uses information that it collected earlier about the user's location to tailor its response, and provide information about the store nearest the user. The conditional responses are based on context values.

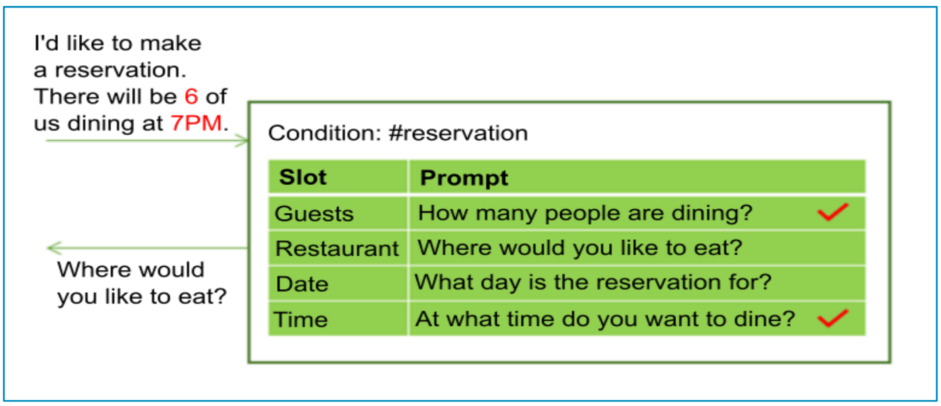
This single node now provides the equivalent function of four separate nodes.

IBM Training

Watson Assistant dialog features (cont.)

Slots

- Add slots to a dialog node to gather multiple pieces of information from a user within that node.
- Slots collect information at the user's pace. Details that the user provides are saved, and the service asks only for the details that the user did not provide.
- **Example:** User-provided values for multiple slots in one input.



Introduction to IBM Watson Assistant

Figure 11-27. Watson Assistant dialog features (cont.)

Slots

Add slots to a dialog node to gather multiple pieces of information from a user within that node. Slots collect information at the users' pace. Details that the user provides are saved, and the service asks only for the details the use did not provide.

Example:

The user wants to reserve a table in a restaurant. The needed information is the number of guests, the date, the time, and the name of the restaurant. When asked, the user might provide values for multiple slots at once. For example, the input might include the information, "There will be 6 of us dining at 7 PM". This one input contains two of the missing required values: the number of guests and time of the reservation. The service recognizes and stores both of them, each one in its corresponding slot. It then displays the prompt that is associated with the next empty slot, which in this case asks the user, "Where would you like to eat?" to determine the restaurant slot and stores the response. After the user replies, the service asks, "What day will this take place?" to determine the date slot, stores it, and gives a reply.

The following video provides a more detailed understanding of slots if you are interested:

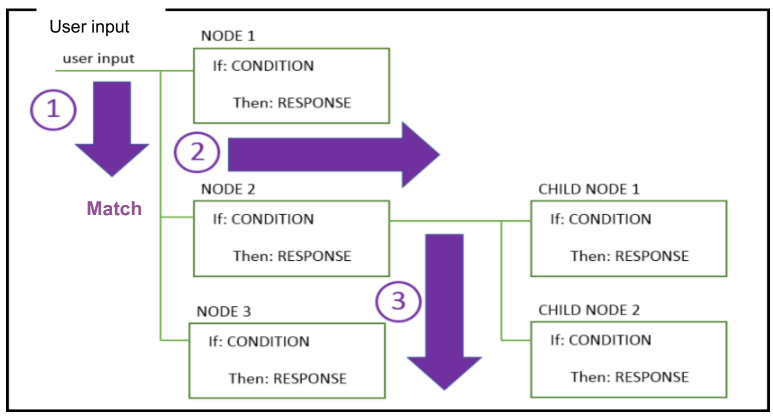
https://console.bluemix.net/docs/services/assistant/dialog-slots.html#dialog-slots

IBM Training

Watson Assistant components (cont.)

Dialog flow

- The dialog standard flow is processed from top to bottom and left to right.
- Triggered by a successful condition.
- It ends when it reaches the last node in the branch that it is following.



Introduction to IBM Watson Assistant

Figure 11-28. Watson Assistant components (cont.)

Dialog flow

The dialog is processed by the service from top to bottom.

As it travels down the tree, if the service finds a condition that is met, it triggers that node. It then moves from left to right on the triggered node to check the user input against any child node conditions. As it checks the children nodes, it moves again from top to bottom.

The service continues to work its way through the dialog tree until it reaches the last node in the branch that it is following.

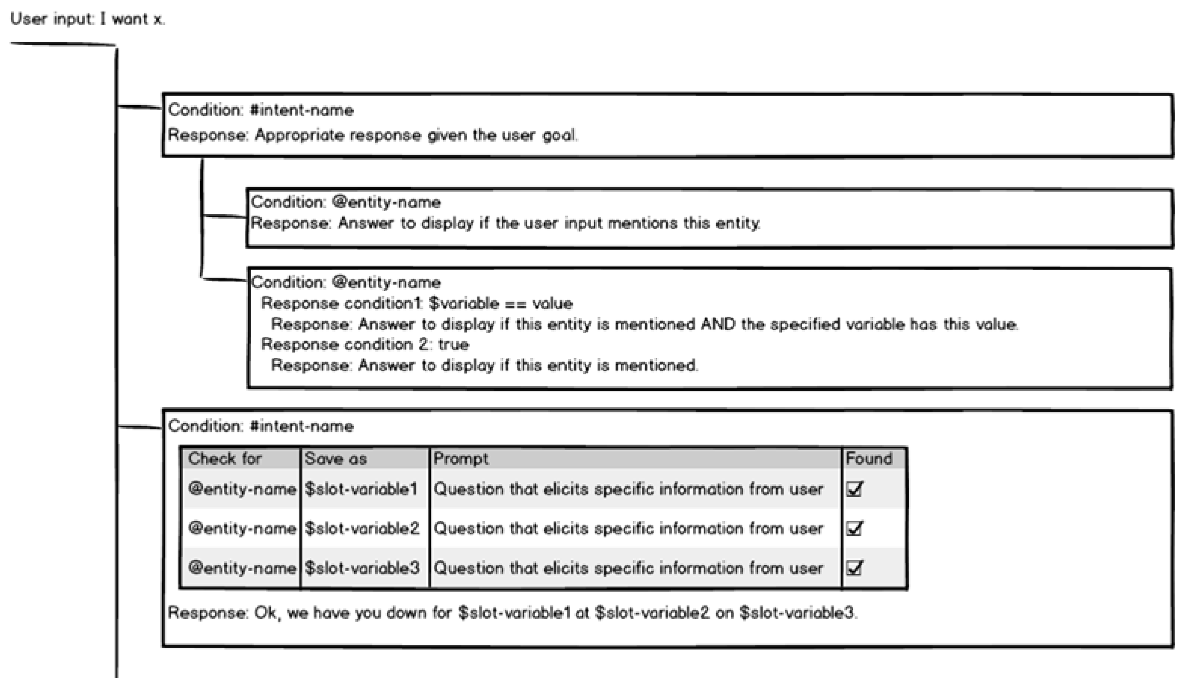
If none of the conditions evaluates to true, then the response from the last node in the tree, which typically has a special *anything_else* condition that always evaluates to true, is returned.

When you start to build the dialog, you must determine the branches to include and where to place them. The order of the branches is important because nodes are evaluated from first to last. The first node whose condition matches the input is used; any nodes that come later in the tree are not triggered.

IBM Training

Watson Assistant components (cont.)

Dialog depiction



Introduction to IBM Watson Assistant

Figure 11-29. Watson Assistant components (cont.)

Dialog depiction

This diagram shows a mockup of a dialog tree that is built with the GUI dialog builder tool. It contains two root dialog nodes. A typical dialog tree would likely have many more nodes, but this depiction provides a glimpse of what a subset of nodes might look like.

The first root node has conditions for an intent value. It has two child nodes and each have a condition on an entity value. The second child node defines two responses. The first response is returned to the user if the value of the context variable matches the value that is specified in the condition. Otherwise, the second response is returned.

This standard type of node is useful to capture questions about a certain topic and then in the root response ask a follow-up question that is addressed by the child nodes. For example, it might recognize a user question about discounts and ask a follow-up question about whether the user is a member of any associations with which the company has special discount arrangements. The child nodes provide different responses based on the user's answer to the question about association membership.

The second root node is a node with slots. It also has conditions on an intent value. It defines a set of slots, one for each piece of information that you want to collect from the user. Each slot asks a question to elicit the answer from the user. It looks for a specific entity value in the user's reply to the prompt, which it then saves in a slot context variable.

This type of node is useful for collecting details that you might need to perform a transaction on the user's behalf. For example, if the user's intent is to book a flight, the slots can collect the origin and destination location information, travel dates, and so on.

11.3. Enriching the chatbot

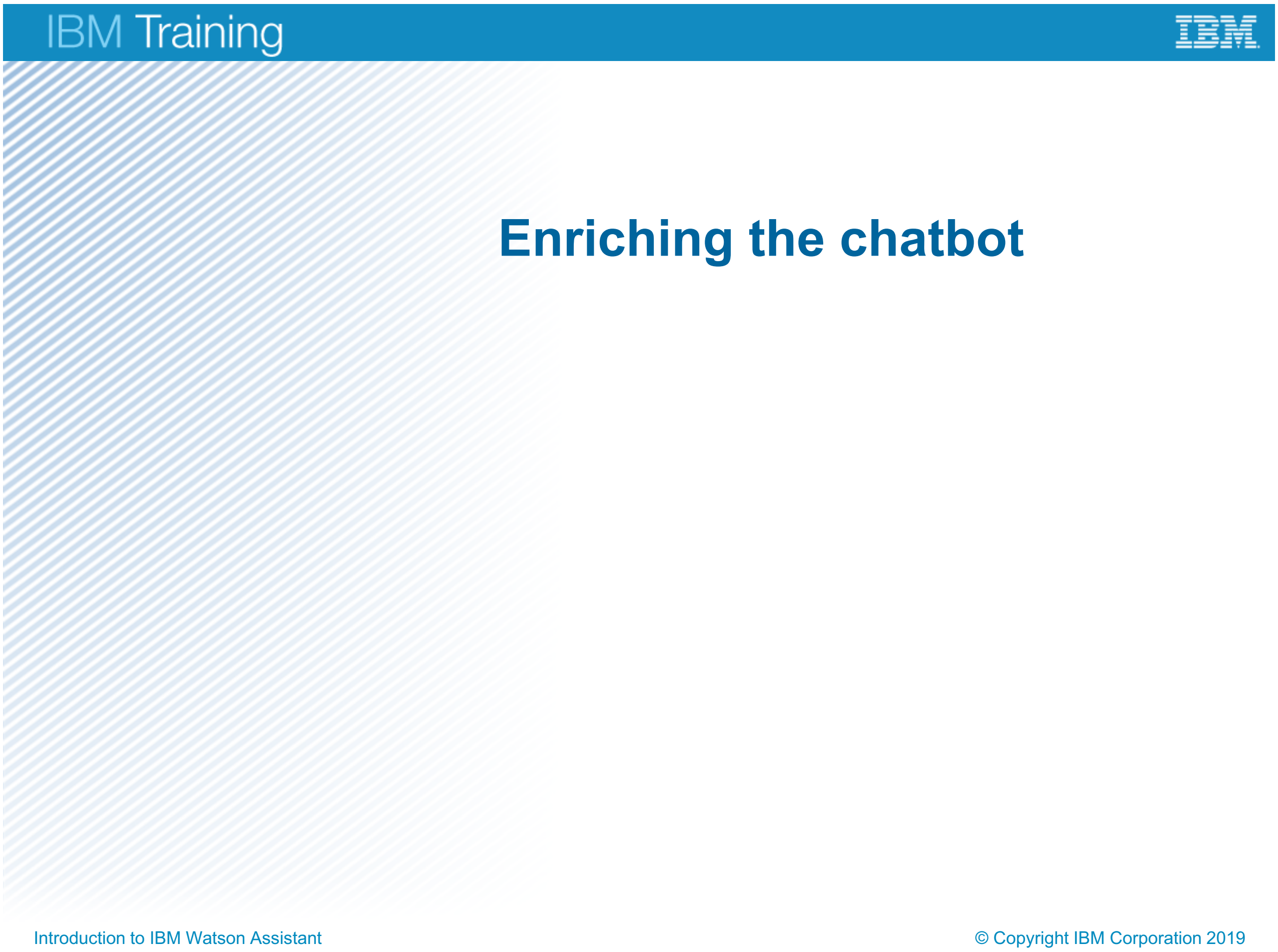


Figure 11-30. Enriching the chatbot

IBM Training

IBM

Topics

- Chatbot components
- Watson Assistant components
- Enriching the chatbot

Introduction to IBM Watson Assistant

Figure 11-31. Topics

IBM Training

Persona

- Creating a persona for your bot makes the bot friendlier and engages its users.
- You can create a bot persona by adding some distinguishing characteristics to your chatbot.
- ***Avatar:*** A face for your chatbot. A simple 2D graphic image, or even a 3D figure adds a nice touch, along with a name.
  - Adding a gender is not always a good idea. Generally, the chatbot should be gender-neutral, such as a friendly android (robot) or a friendly pet like a cat or a bird.
- **A voice:** Makes a chatbot more interactive and adds value to its personality.
  - Easy to implement because of the many available text to speech services.

Introduction to IBM Watson Assistant

Figure 11-32. Persona

Although it might be clear to the user that they are talking to a chatbot, creating a persona makes the bot friendlier and engages its users. You can create a bot persona by providing some distinguishing characteristics to your chatbot. For example, adding the following features can help to accomplish this goal:

- **Avatar**

  An avatar adds a face to your chatbot. A simple 2D graphic image adds a nice touch, along with a name. Adding a gender is not always a good idea. Generally, the chatbot should be gender-neutral, such as a friendly android (robot) or a friendly pet like a cat or a bird.

- **A voice**

  Adding a voice to your chatbot makes it more interactive and adds value to its personality. Doing so requires almost no effort because most text to speech services can be added to your existing output response text to transform it into audio.

IBM Training

Emotion and tone

- **Sentiment analysis** can map user emotion input to a positive, negative, or neutral status.
  - Your chatbot can then sense the user's positive comment and react to it, or sense a negative input and direct the conversation appropriately.
- **Tone analysis** can perform more advanced interpretation of the user's mood.
  - For example, it can detect whether the user is frustrated or thankful. The chatbot can then reply with a customized answer to each mood, giving your bot emotional AI that adds to its personality and makes it friendlier.

Introduction to IBM Watson Assistant

Figure 11-33. Emotion and tone

Emotion and tone

- **Sentiment analysis** can map user emotion input to a positive, negative, or neutral status. It adds emotion to your chatbot. Your chatbot can sense the user's positive comment and react to it, or sense a negative input and direct the conversation appropriately.
- **Tone analysis** can perform more advanced interpretation of the user's mood. For example, it can detect whether the user is frustrated or thankful. The chatbot can then reply with a customized answer to each mood, giving your bot emotional AI that adds to its personality and makes it friendlier.

IBM Training

Interfaces

- Using messaging platforms with your chatbot makes it more accessible and user friendly.
- Most services and tools provide plug-ins or easy-to-use recipes to plug your chatbot to these platforms.
- This approach simplifies the effort of designing a chat interface and eliminates the need to download another app or registering on a website to use the chatbot.

Introduction to IBM Watson Assistant

Figure 11-34. Interfaces

Interfaces

One of the key factors in the rise of chatbots is the growing popularity of messaging apps. Using messaging platforms with your chatbot makes it more accessible and friendly.

Most services and tools provide plug-ins or recipes to plug your chatbot to these platforms. This approach simplifies the effort of designing a chat interface and eliminates the need to download another app or registering on a website to use the chatbot.

IBM Training

Speech recognition

- You can use IBM, Google, and Microsoft speech to text services to enable your chatbot to understand user speech.
- This feature is heavily used in home automation chatbots and automation tasks in general.
  - For example, in Microsoft Windows you can give commands to Cortana by using your voice without touching a key or pressing a mouse button. You can do the same with Siri on the Apple iPhone.

Introduction to IBM Watson Assistant

Figure 11-35. Speech recognition

Speech Recognition

With advancements in speech to text technology, you can use IBM, Google, and Microsoft speech to text services to enable your chatbot to understand user speech.

This feature is heavily used in home automation chatbots and automation tasks in general. For example, in Microsoft Windows you can give commands to Cortana by using your voice without touching a key or pressing a mouse button. You can do the same with Siri on the Apple iPhone.

IBM Training

Unit summary

- Explain assistants and skills.
- Explain intents.
- Explain entities.
- Explain context variables.
- Describe how the nodes in a dialog are triggered.
- Describe how the dialog flow is processed.
- Describe the features that can be used to enrich the chatbot.

Introduction to IBM Watson Assistant

Figure 11-36. Unit summary

IBM Training

Review questions

1. The ____ name is always prefixed with '@'.
   A. Intent
   B. Entity
   C. Dialog
   D. Skill

2. True or False: The content catalog is used to add pre-built entities.

3. Which feature of a dialog can be used to ask only for input that the user has not already provided?
   A. Rich response
   B. Slot
   C. Conditional response
   D. Context variable

Introduction to IBM Watson Assistant

Figure 11-37. Review questions

1. B. Entity.
2. False. A content catalog that contains prebuilt common **intents** is provided with the Assistant tool.
3. B. Slot.

IBM Training

Review questions (cont.)

4. Which of the following features can be used to enrich your chatbot?
   - A. Avatar
   - B. Tone analysis
   - C. Speech recognition
   - D. Text to speech
   - E. All of the above
5. True or False: The dialog flow is from top to bottom and left to right.
6. Which of the following response types can be used in a dialog node?
   - A. Image
   - B. Text
   - C. Option
   - D. All the above

Introduction to IBM Watson Assistant

Figure 11-38. Review questions (cont.)

4. E. All of the above.
5. True.
6. D. All the above.

IBM Training

Review questions (cont.)

7. #pay_bill is an example of ____
   A. Intent
   B. Entity
   C. Skills
   D. Workspace

8. A branching conversation flow that defines how your chatbot responds when it recognizes the input from the user.
   A. Intent
   B. Dialog
   C. Condition
   D. Entity

9. True or False: @menu is an example of intent.

Introduction to IBM Watson Assistant

Figure 11-39. Review questions (cont.)

7. A. Intent.
8. B. Dialog.
9. False. @menu is an example of **entity**.

IBM Training

Review answers

1. The ____ name is always prefixed with '@'.
   A. Intent
   B. <u>Entity</u>
   C. Dialog
   D. Skill

2. True or **<u>False</u>**: The content catalog is used to add pre-built entities.

3. Which feature of a dialog can be used to ask only for input that the user has not already provided?
   A. Rich response
   B. <u>Slot</u>
   C. Conditional response
   D. Context variable

Introduction to IBM Watson Assistant

Figure 11-40. Review answers

1. B. Entity.
2. False. A content catalog that contains prebuilt common **intents** is provided with the Assistant tool.
3. B. Slot.

IBM Training

Review answers (cont.)

4. Which of the following features can be used to enrich your chatbot?
   - A. Avatar
   - B. Tone analysis
   - C. Speech recognition
   - D. Text to speech
   - **E. All of the above**
5. **True** or False: The dialog flow is from top to bottom and left to right.
6. Which of the following response types can be used in a dialog node?
   - A. Image
   - B. Text
   - C. Option
   - **D. All the above**

Introduction to IBM Watson Assistant

Figure 11-41. Review answers (cont.)

4. E. All of the above.
5. True.
6. D. All the above.

IBM Training

Review answers (cont.)

7. #pay_bill is an example of ____
   - **A. Intent**
   - B. Entity
   - C. Skills
   - D. Workspace
8. A branching conversation flow that defines how your chatbot responds when it recognizes the input from the user.
   - A. Intent
   - **B. Dialog**
   - C. Condition
   - D. Entity
9. True or **False**: @menu is an example of intent.

Figure 11-42. Review answers (cont.)

7. A. Intent.
8. B. Dialog.
9. False. @menu is an example of **entity**.

IBM Training

Exercise: Getting started with Watson Assistant.

Introduction to IBM Watson Assistant

Figure 11-43. Exercise: Getting started with Watson Assistant.

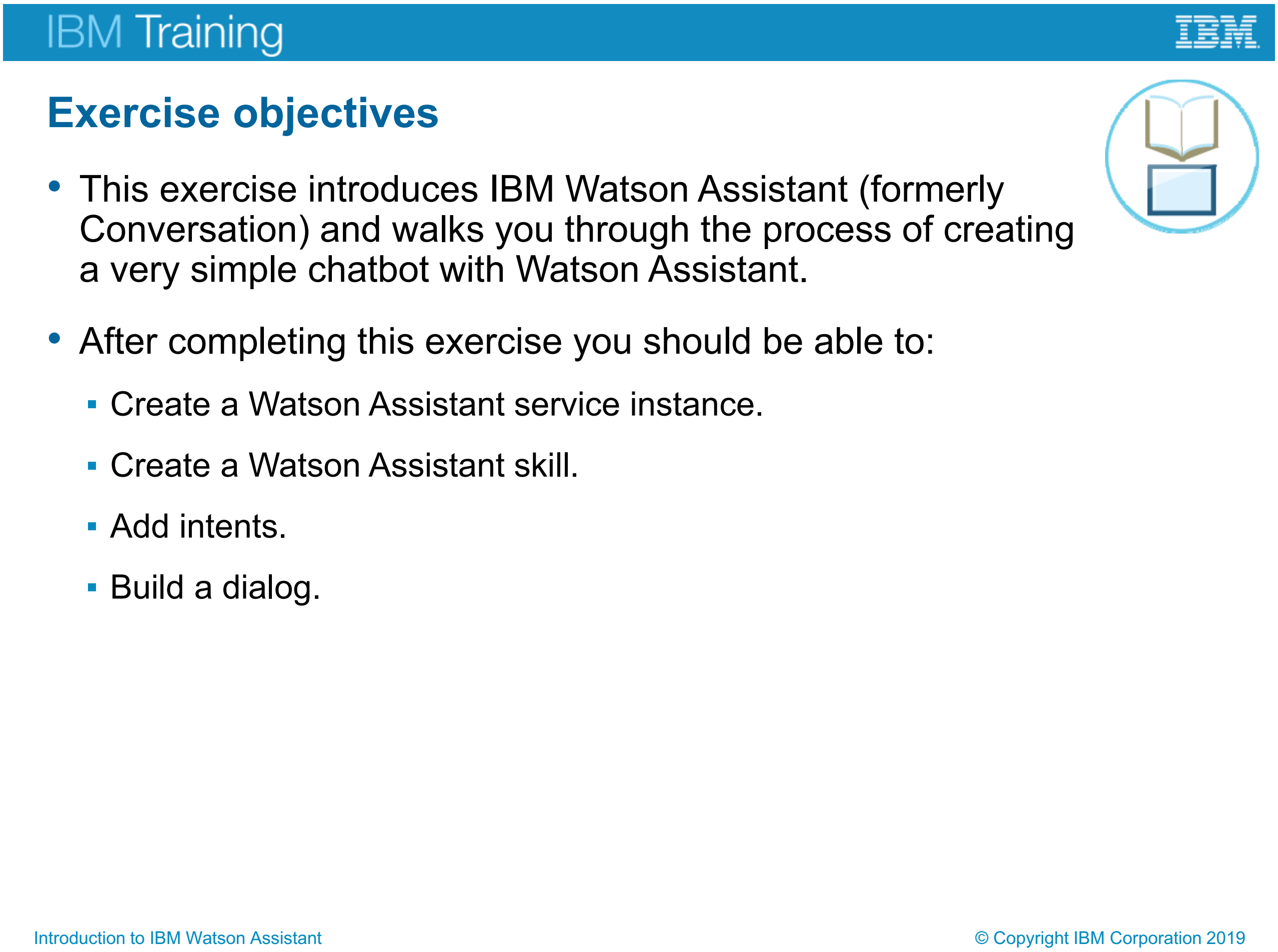


Figure 11-44. Exercise objectives

After completing this exercise you should be able to:

- Create a Watson Assistant service instance.
- Create a Watson Assistant skill.
- Add intents.
- Build a dialog.

IBM Training

Exercise: Help Desk Chatbot

Introduction to IBM Watson Assistant

Figure 11-45. Exercise: Help Desk Chatbot

IBM Training

Exercise objectives

- This exercise describes how to create a chatbot application quickly without coding and integrate it with the Watson Assistant service.
- After completing this exercise you should be able to:
  - Create a Watson Assistant skill.
  - Add intents and entities.
  - Build a dialog.
  - Create a Node-RED application that integrates with the Watson Assistant service.
  - Set up Slack as a front-end chat service for the Help Desk chatbot.

Introduction to IBM Watson Assistant

Figure 11-46. Exercise objectives

After completing this exercise you should be able to:

- Create a Watson Assistant skill.
- Add intents and entities.
- Build a dialog.
- Create a Node-RED application that integrates with the Watson Assistant service.
- Set up Slack as a front-end chat service for the Help Desk chatbot.

Unit 12. Introduction to computer vision

Estimated time

00:30

Overview

This unit provides a high-level introduction to computer vision (CV).

IBM Training

Unit objectives

- Define computer vision (CV).
- Know the history of CV and its advancement with artificial intelligence (AI).
- Identify CV applications and use cases.
- List tools and services for CV.

Introduction to computer vision

Figure 12-1. Unit objectives

IBM Training

What is computer vision

- Computer vision is a branch of science that is concerned with processing images to extract, analyze, and understand useful information from a single image or image sequence.
- It aims at simulating the human visual system.
- It uses various machine learning and deep learning algorithms to analyze images for scenes, objects, faces, and other content in videos, photos, and pictures in general.

Introduction to computer vision

Figure 12-2. What is computer vision

Computer vision is a branch of science that is concerned with processing images to extract, analyze, and understand useful information from a single image or image sequence. It aims to create an artificial (computer) system that can achieve the capabilities of a human visual system so that the machine can “see”.

CV uses various machine learning algorithms to analyze images for scenes, objects, faces, and other content in videos, photos, and pictures in general.

References:

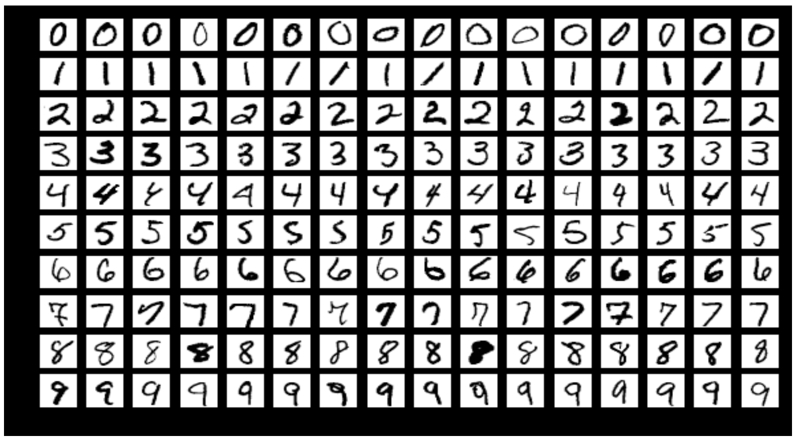
http://www.bmva.org/visionoverview

https://console.bluemix.net/docs/services/visual-recognition/index.html#about

IBM Training

Computer vision history

- Started with the emergence of AI In 1956.
- Convolutional neural networks were proposed in the well-known 1998 research paper by Yann LeCun and Léon Bottou.
  - 99.2% recognition accuracy on the MNIST data set.



https://commons.wikimedia.org/w/index.php?curid=64810040

Introduction to computer vision

Figure 12-3. Computer vision history

Started with the emergence of AI in 1956 and evolved with the advancements of AI.

Convolutional neural networks were proposed in the well-known 1998 research paper by Yann LeCun and Léon Bottou. In their detailed paper, their proposed neural network architecture “LeNet 5” reached an accuracy of 99.2% on the Modified National Institute of Standards and Technology (MINST) data set. The MINST data set is a large database of hand-written digits that is commonly used for training various image processing systems. The database is also widely used for training and testing in the field of machine learning.

Reference:

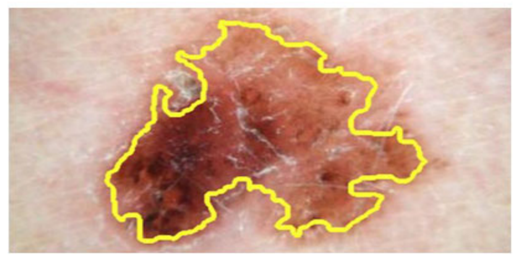
https://commons.wikimedia.org/w/index.php?curid=64810040

IBM Training

Computer vision applications

- **Manufacturing**: Ensure that products are being positioned correctly on an assembly line.
- **Visual auditing**: Look for visual compliance or deterioration in a fleet of trucks, planes, windmills, transmission or power towers , and so on.
- **Insurance**: Classify claims images into different categories.
- **Medical image processing**: Detect tumors.
- **Automotive industry:** Object detection for safety.

Identifying skin cancer by using CV



Introduction to computer vision

Figure 12-4. Computer vision applications

CV can be used in many applications and industries:

- **Manufacturing**: Use images from a manufacturing setting to make sure that products are being positioned correctly on an assembly line, or ensure product quality by identifying defective parts on the production line instead of by using manual validation.
- **Visual auditing**: Look for visual compliance or deterioration in a fleet of trucks, planes, or windmills, etc.. in the field. Train custom models to understand what defects look like.
- **Insurance**: Rapidly process claims by using images to classify claims into different categories.
- **Social listening:** Use images from your product line or your logos to track buzz about your company on social media.
- **Medical image processing:** Using images for diagnosing patients, such as the detection of tumors.
- **Automotive industry:** There are many applications for CV in cars. For example, while parking a car, a camera can detect objects and warn the driver when they get too close to them.

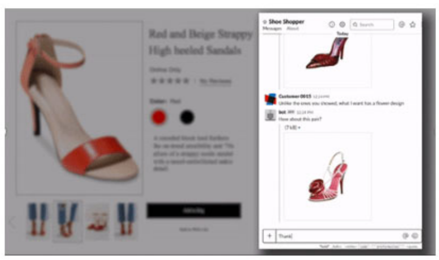
Reference:

https://www.research.ibm.com/artificial-intelligence/computer-vision/

IBM Training

Computer vision applications (cont.)

- **Social commerce:** Use an image of a house to find similar homes that are for sale.
- **Social listening**: Track the buzz about your company on social media by looking for product logos.
- **Retail:** Use the photo of an item to find its price at different stores.
- **Education:** Use pictures to find educational material on similar subjects.
- **Public safety:** Automated license-plate reading.



Interactive image search

Company's logo search

Introduction to computer vision

Figure 12-5. Computer vision applications (cont.)

Social commerce: Use an image of a house to find similar homes that are for sale.

Social listening: Use images from your product line or your logos to track buzz about your company on social media.

Retail: Use the photo of an item to find its price at different stores.

Education: Use pictures to find educational material on similar subjects.

Public safety: Automated license-plate reading.

Do you know of or heard of any other applications?

Reference:

https://www.research.ibm.com/artificial-intelligence/computer-vision/

IBM Training

Computer vision tasks

- **Object detection and recognition:** Detect certain patterns within the image.
- Examples:
  - Detecting red eyes when taking photos in certain conditions.
  - Face recognition.

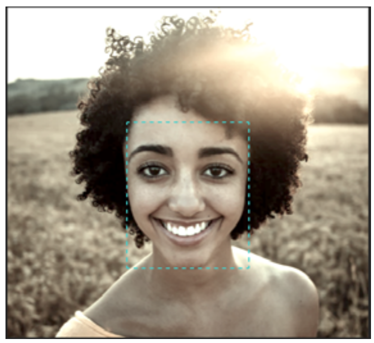


Figure 12-6. Computer vision tasks

A CV task represents a well-defined problem in CV that can be solved to a certain extent or accuracy by using one method or another.

Object detection and recognition deals with locating and detecting certain patterns within the image. For example, detecting red eyes when taking photos in certain conditions. Other applications include face detection and face recognition.

A CV task is also used in tracking objects, for example, tracking a ball during a football match, tracking the movement of a cricket bat, or tracking a person in a video.

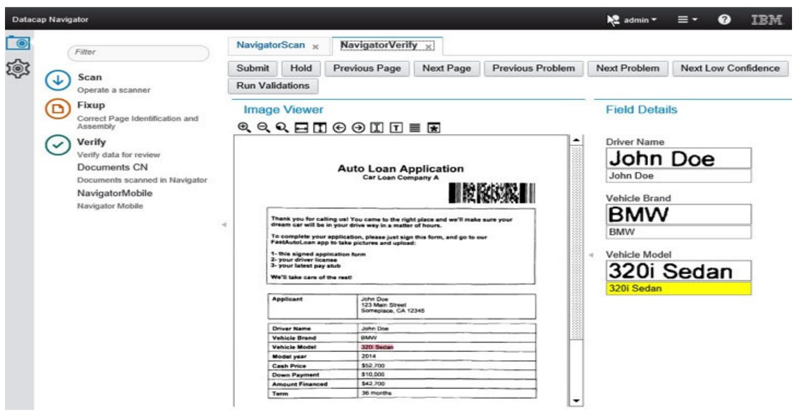
Reference:

https://www.ibm.com/watson/services/visual-recognition/demo/#demo

IBM Training

Computer vision tasks (cont.)

- **Content-based image retrieval:** Image retrieval from a database based on user's image query.
  - By using image actual feature contents such as colors, shapes, and textures
  - Not using image metadata (keywords, tags, or descriptions)
- **Optical character recognition (OCR):** Converting hand-written text to a digital format.



Introduction to computer vision

© Copyright IBM Corporation 2019

Figure 12-7. Computer vision tasks (cont.)

Content-based image retrieval or "query-by image content" (QBIC) is the retrieval of images from a database by using an image as a query. IBM was one of the pioneers in developing QBIC at the IBM Almaden Research Center. "Content-based" refers to actual feature contents of the image like colors, shapes, and textures. Other image retrieval systems rely on image metadata such as keywords, tags, or descriptions that are associated with the image instead of image content. Content-based image retrieval is preferred because searches that rely purely on metadata depend on the quality of annotation.

Optical character recognition (OCR): Scan papers and hand-written forms, identify the characters in them, and transform them into digital format (strings).

References:

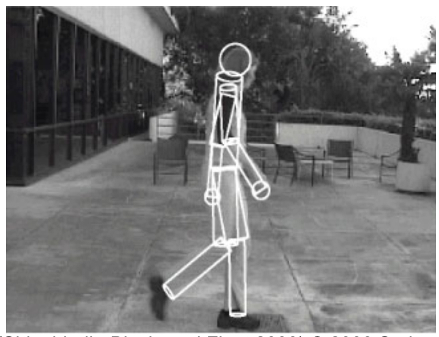
http://www.ugmode.com/prior_art/lew2006cbm.pdf

https://en.wikipedia.org/wiki/Content-based_image_retrieval#cite_note-Survey-1

IBM Training

Computer vision tasks (cont.)

- **Object tracking:** Following the position changes of a target object from one frame to another in an image sequence or video.
- The following photo shows an example of human tracking.



(Sidenbladh, Black, and Fleet 2000) © 2000 Springer.

Introduction to computer vision

Figure 12-8. Computer vision tasks (cont.)

Object tracking: Following the position changes of a target object from one frame to another within an image sequence or video.

References:

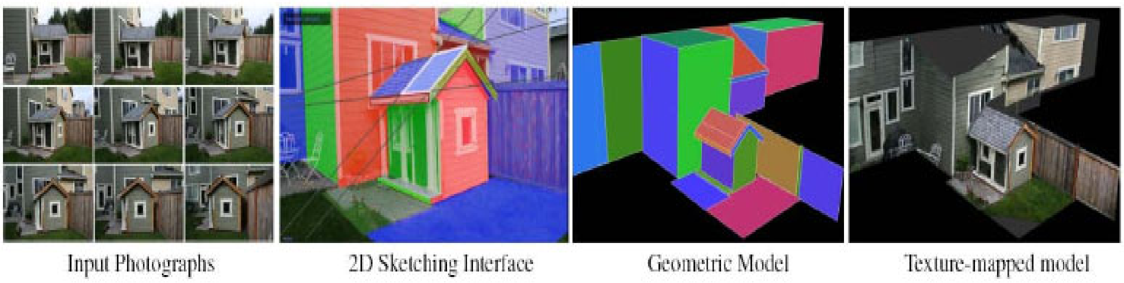
https://en.wikipedia.org/wiki/Computer_vision

APA citation for image: Sidenbladh, H., Black, M. J., & Fleet, D. J. (2000, June). Stochastic tracking of 3D human figures using 2D image motion. In *European conference on computer vision* (pp. 702-718). Springer, Berlin, Heidelberg.

IBM Training

Computer vision tasks (cont.)

- **Image restoration:** Fixing and restoring images that are corrupted by noise, such as motion blur, to their default state.
- **Scene reconstruction:** Creation of a 3D model by supplying the system with multiple 2D images from different views. The computer constructs a 3D model based on those images.



(Sinha, Steedly, Szeliskiet al. 2008) © 2008 ACM.

Introduction to computer vision

Figure 12-9. Computer vision tasks (cont.)

Image restoration: Fixing and restoring images that are corrupted by noise, such as motion blur, to their default state.

Scene reconstruction: Creation of 3D model by supplying the system with multiple 2D images from different views. The computer constructs a 3D model based on those images.

References:

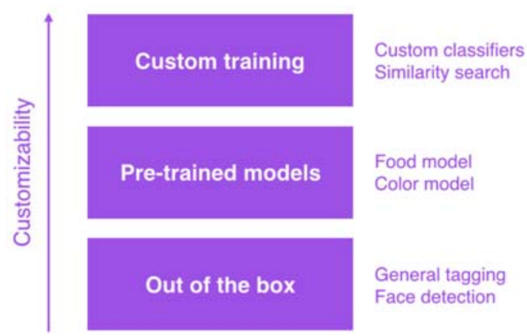
https://en.wikipedia.org/wiki/Computer_vision

APA image citation: Sinha, S. N., Steedly, D., Szeliski, R., Agrawala, M., & Pollefeys, M. (2008, December). Interactive 3D architectural modeling from unordered photo collections. In *ACM Transactions on Graphics (TOG)* (Vol. 27, No. 5, p. 159). ACM.

IBM Training

Computer vision tools

- **OpenCV:** CV open-source library
  - C++, Python, Java, and MATLAB interfaces
- **Microsoft Vision API:** Specialized in OCR and analyzing videos
- **IBM Watson Visual Recognition:** A service on IBM Cloud
  - A set of built-in classes provides highly accurate results without training.
  - A trained custom model enables the use of the Visual Recognition service to recognize images that are not available with pre-trained classification.



Introduction to computer vision

Figure 12-10. Computer vision tools

OpenCV: An open-source library that can be used to perform most CV tasks that are required on any language. It has C++, Python, Java, and MATLAB interfaces.

Microsoft Vision API: Specialized in OCR and analyzing videos.

IBM Watson Visual Recognition: This service is available on IBM Cloud. A set of built-in classes provides highly accurate results without training.

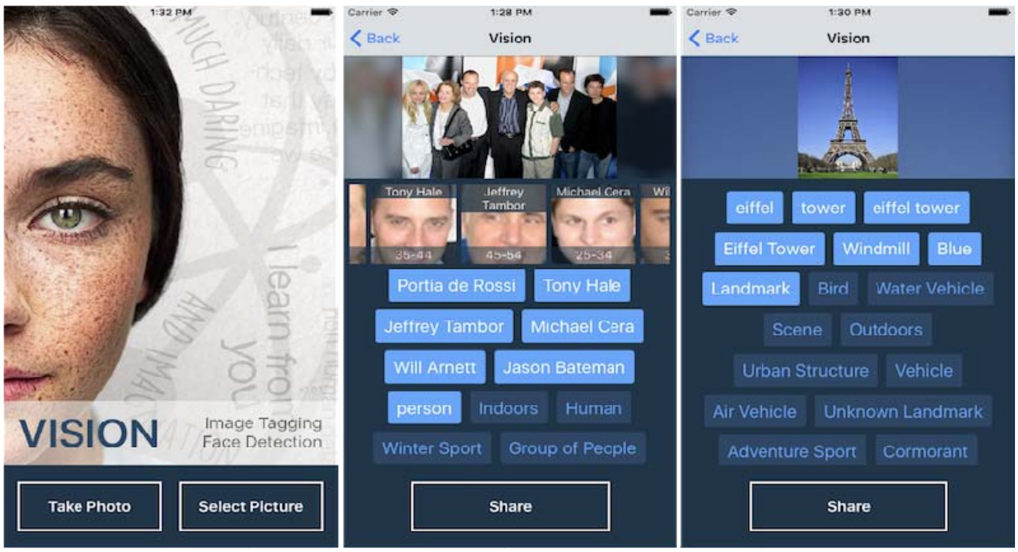
You can train custom models to create specialized classes. With a custom model, you can train the Visual Recognition service to classify images to suit your business needs. By creating a custom model, you can use the Visual Recognition service to recognize images that are not available with pre-trained classification.

Image reference:
https://www.ibm.com/blogs/bluemix/2017/05/watson-learns-see-food-introducing-watson-visual-recognition-food-model/

IBM Training

Computer vision use cases

- **Facial recognition**: We use facial recognition daily.
  - Taking a photo and applying effects by using your smartphone.
  - Tagging friends on social media.



Introduction to computer vision

Figure 12-11. Computer vision use cases

Facial recognition is one of the most-used features of CV. It is often used in smartphones when you take a photo or apply effects. Another widely used feature is tagging friends on social media. These features use face recognition and facial identification, in which a face is recognized from the image and identifies the person in the image. A facial recognition feature can recognize faces in an image, as shown in the figure.

Image: https://www.ibm.com/blogs/bluemix/2016/02/openwhisk-and-watson-image-tagging-app/

IBM Training

Computer vision use cases (cont.)

- **Augmented reality**: Manipulation and addition of a system-generated image (3D or 2D) as an overlay of a user's view.
- Examples: Google Glass, emoji filters in some mobile apps, and Pokémon Go.
- Mobile shopping applications will give in-store shoppers instant product details and promotions through their mobile devices.

Introduction to computer vision © Copyright IBM Corporation 2019

Figure 12-12. Computer vision use cases (cont.)

Augmented reality (AR) is the manipulation and addition of a system-generated image (3D or 2D) as an overlay of a user's view. Examples include Google Glass, emoji filters in some mobile apps, and Pokémon Go.

While in-store shopping accounts for 92 percent of retail volume, consumers are expecting the same levels of personalization and customization that they do when they shop online; 58 percent of consumers want in-store product information and 19 percent of consumers are already browsing their mobile devices while in-store. CV and AR technology can bring all the benefits of online shopping into traditional shops. By creating augmented reality mobile shopping applications (apps), as shown in figure, in-store shoppers immediately learn about product details and promotions through their mobile devices.

For more information on AR, watch the following video:

https://youtu.be/EAVtHjzQnqY

Reference:

https://www.ibm.com/blogs/bluemix/2017/03/augmented-reality-new-horizon-multi-sensory-learning/

IBM Training

Unit summary

- Define computer vision (CV).
- Know the history of CV and its advancement with artificial intelligence (AI).
- Identify CV applications and use cases.
- List tools and services for CV.

Introduction to computer vision

Figure 12-13. Unit summary

IBM Training

Review questions

1. One of the following items is NOT an application of CV:
   - A. Process claims by using images to classify claims into different categories.
   - B. Real-time video stream analysis that includes targeted observations, such as facial recognition and automated license-plate reading.
   - C. Use images from a manufacturing setting to make sure that products are being positioned correctly on an assembly line.
   - D. Sentiment analysis by analyzing the words that are posted in social media channels.
2. True or False: Watson Visual Recognition cannot be trained.

Introduction to computer vision

Figure 12-14. Review questions

1. D.
2. False. You can train custom models to create specialized classes.

IBM Training

Review questions (cont.)

3. Which of the following items are CV tools and services? (Select all that apply.)
   A. Chatfuel.
   B. API.ai.
   C. OpenCV.
   D. IBM Watson Visual Recognition.
   E. C and D
   F. All of the above
4. True or False: Convolutional neural networks were proposed in the well-known 1998 research paper by Yann LeCun and Léon Bottou.

Introduction to computer vision

Figure 12-15. Review questions (cont.)

3. E. C and D.
4. True.

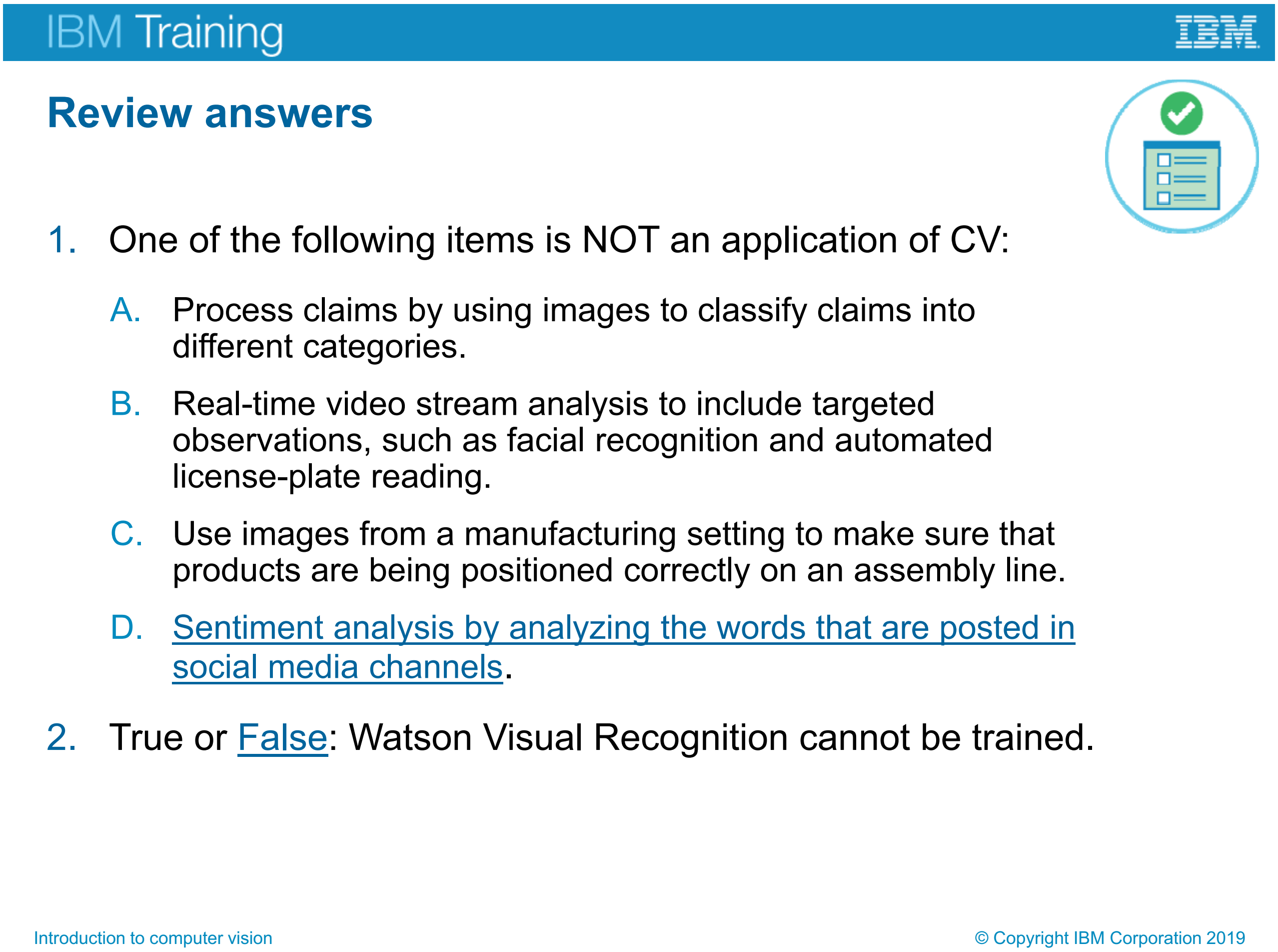


Figure 12-16. Review answers

1. D.
2. False. You can train custom models to create specialized classes.

IBM Training

Review answers (cont.)

3. Which of the following items are CV tools and services? (Select all that apply.)
   A. Chatfuel.
   B. API.ai.
   C. OpenCV.
   D. IBM Watson Visual Recognition.
   E. C and D
   F. All of the above
4. True or False: Convolutional neural networks were proposed in the well-known 1998 research paper by Yann LeCun and Léon Bottou.

Introduction to computer vision

Figure 12-17. Review answers (cont.)

3. E. C and D.
4. True.

Unit 13. Computer vision fundamentals

Estimated time

00:30

Overview

This unit explains the basic steps of a typical computer vision (CV) pipeline, how CV analyzes and processes images, and explores commonly used techniques in CV.

IBM Training

Unit objectives

- Describe image representation for computers.
- Describe the computer vision (CV) pipeline.
- Describe different preprocessing techniques.
- Explain image segmentation.
- Explain feature extraction and selection.
- Describe when object recognition takes place.

Computer vision fundamentals

Figure 13-1. Unit objectives

13.1. Image representation

IBM Training

Image representation

Computer vision fundamentals

Figure 13-2. Image representation

IBM Training

Topics

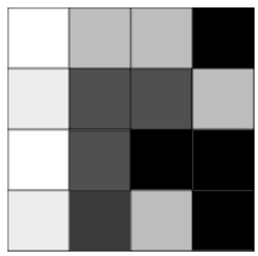
- Image representation
- Computer vision pipeline

Figure 13-3. Topics

IBM Training

Image representation

- Images are stored as a 2D array of pixels on computers.
- Each pixel has a certain value representing its intensity.
- Example of grayscale representation:
  - Image is black and white with shades of gray in between.
  - Pixel intensity is a number between 0 (black) and 255 (white).



```
int[ ][ ] array = {  {255, 170, 170,   0},
                     {220,  80, 80, 170},
                     {255,  80, 0,  0},
                     {175, 20, 170,  0}  };
```

Computer vision fundamentals

Figure 13-4. Image representation

It is important to understand how images are stored and represented on computer screens. Images are made of grids of pixels, that is, a two-dimensional (2D) array of pixels, where each pixel has a certain value representing its intensity. A pixel represents a square of color on the user's screen. There are many ways to represent color in images.

If the image is grayscale, that is, black and white with shades of gray in between, the intensity is a number in the range 0 - 255, where 0 represents black and 255 represents white. The numbers in between these two values are different intensities of gray.

For example, in the picture in the slide, assume that you selected a small square from the image (the part in the red square). The pixel representation for it is in black, white, and shades of gray. To the right of the picture is the numeric values of these pixels in a 2D array.

Note

The 256 scaling is a result of using 8-bit storage (2^8 =256).

References:

http://www.csfieldguide.org.nz/en/chapters/data-representation.html

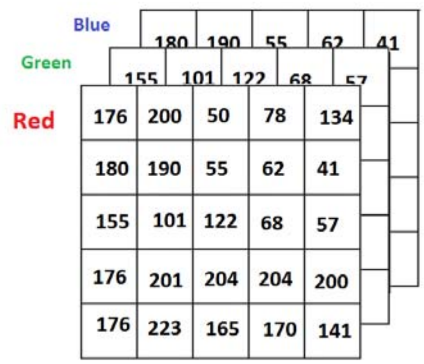
https://www.sqa.org.uk/e-learning/BitVect01CD/page_36.htm

https://www.mathworks.com/help/examples/images/win64/SegmentGrayscaleImageUsingKMeansClusteringExample_01.png

IBM Training

Image representation (cont.)

- Example of color representation:
  - The pixel color is represented as a mix of **R**ed, **G**reen, and **B**lue.
  - The pixel intensity becomes three 2D arrays or one 2D array, where each entry is an object containing the 3 color values of RGB



Computer vision fundamentals

Figure 13-5. Image representation (cont.)

If the image is colored, there are many color models that are available. The most well-known model is RGB, where the color of the pixels is represented as a mix of Red, Blue, and Green color channels. The value of the pixel intensity is represented as three 2D arrays that represent the three color channels, where each color value is in the range 0 – 255. The color value represents different intensities of the colors and all the colors that you see on the screen. The number of color variations that are available in RGB is 256 * 256 *256 = 16,777,216 possible colors.

Depending on the data structure used that image appears as

- Three 2D arrays, with each 2D array representing the values and intensities for one color. The image is a combination of all three arrays.
- One 2D array, where each entry in the array is an object containing the 3 color values of RGB

References:

http://www.csfieldguide.org.nz/en/chapters/data-representation.html

https://www.sqa.org.uk/e-learning/BitVect01CD/page_36.htm

https://www.mathworks.com/company/newsletters/articles/how-matlab-represents-pixel-colors.html

13.2. Computer vision pipeline

IBM Training

Computer vision pipeline

Computer vision fundamentals

Figure 13-6. Computer vision pipeline

IBM Training

Topics

- Image representation
- Computer vision pipeline

Computer vision fundamentals

Figure 13-7. Topics

IBM Training

Computer vision pipeline

- The steps and functions that are included are highly dependent on the application.
- Here is a conventional visual pattern recognition pipeline.

Computer vision fundamentals

Figure 13-8. Computer vision pipeline

The steps and functions that are included in a CV system algorithm pipeline are highly dependent on the application. The application might be implemented as a stand-alone part of a system and might apply to only a certain step of the pipeline, or it can be a larger application, that is, an end-to-end application that takes certain images and then applies all of the steps and functions in the pipeline.

Reference:

https://www.researchgate.net/publication/273596152_Hardware_Architecture_for_Real-Time_Computation_of_Image_Component_Feature_Descriptors_on_a_FPGA

IBM Training IBM

Computer vision pipeline (cont.)

1. Image acquisition:

- The process of acquiring images and saving them in a digital image format for processing.
- Images often use common formats, such as .jpeg, .png, and .bmp.
- Images are 2D images that are stored as arrays of pixels according to their color model.

Computer vision fundamentals © Copyright IBM Corporation 2019

Figure 13-9. Computer vision pipeline (cont.)

1. Image acquisition

Image acquisition is the process of acquiring images and saving them in a digital image format for processing in the pipeline.

Images are acquired from image sensors, such as commercial cameras. There are other types of light-sensitive devices that can capture different types of images. Other image sensors include radar, which uses radio waves for imaging.

Depending on the type of the device, the captured image can be 2D, 3D, or a sequence of images. These images often use common formats, such as .jpeg, .png, and .bmp.

Images are stored as arrays of pixels according to their color model.

Reference:

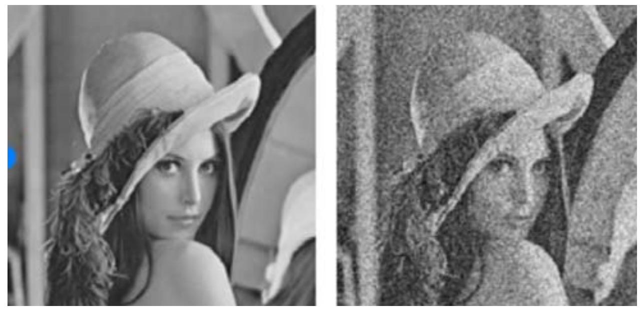
https://en.wikipedia.org/wiki/Computer_vision#System_methods

IBM Training

Computer vision pipeline (cont.)

2. Pre-processing:

- Preparing the image for the processing stage
- Examples:
  - Resizing images
  - Noise reduction
  - Contrast adjustment



Computer vision fundamentals

Figure 13-10. Computer vision pipeline (cont.)

2. Pre-processing:

This stage focuses on preparing the image for the processing stage.

Examples:

- Images are resized as a form of normalization so that all images are the same size.
- Noise reduction reduces noise that represents false information in the image, as shown in the slide.

  Image citation: A Spatial and Frequency Domain Analysis of the Effect of Removal Attacks on Digital Image Watermarks‖ - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/Effect-of-adding-salt-pepper-noise-to-an-image-a-Original-watermarked-image-b-Salt_fig3_229008840 [accessed 18 Nov, 2018]
- Contrast adjustment helps with the detection of image information. In the images of the little girl, each image represents a different contrast adjustment.

  Image source: https://www.mathworks.com/help/images/contrast-adjustment.html

References:

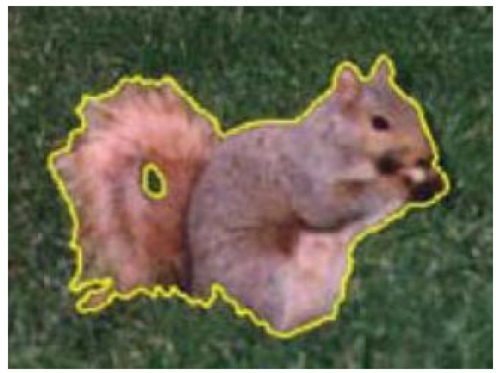
https://en.wikipedia.org/wiki/Computer_vision#System_methods

IBM Training

Computer vision pipeline (cont.)

3. Segmentation:

- Partitioning an image into regions of similarity.
- Grouping pixels and features with similar characteristics together.
- Helps with selecting regions of interest within the images. These regions can contain objects of interest that we want to capture.
- Segmenting an image into foreground and background.



(Cremers, Rousson, and Deriche 2007) © 2007 Springer

Computer vision fundamentals

Figure 13-11. Computer vision pipeline (cont.)

3. Segmentation

The position of the image segmentation phase is not fixed in the CV pipeline. It might be a part of the pre-processing phase or follow it (as in our pipeline) or be part of the feature extraction and selection phase or follow them.

Segmentation is one of the oldest problems in CV and has the following aspects:

- Partitioning an image into regions of similarity.
- Grouping pixels and features with similar characteristics together.
- Helps with selecting regions of interest within the images. These regions can contain objects of interest that we want to capture**.**
- Segmenting an image into foreground and background to apply further processing on the foreground.

Image citation: Cremers, D., Rousson, M., & Deriche, R. (2007). A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International journal of computer vision*, *72*(2), 195-215.

References*:*

https://en.wikipedia.org/wiki/Computer_vision

http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf

http://www.cse.iitm.ac.in/~vplab/courses/CV_DIP/PDF/lect-Segmen.pdf

IBM Training

Computer vision pipeline (cont.)

4. Feature extraction and selection:

- Image features are extracted.
- Image features examples:
  - Lines and edges
  - Corners

Computer vision fundamentals

Figure 13-12. Computer vision pipeline (cont.)

4. Feature extraction and selection

Image features are extracted from the image. Typical examples of such features are:

- Lines and edges: These features are where sharp changes in brightness occur. They represent the boundaries of objects.
- Corners: These features are points of interest in the image where intersections or changes in brightness happens.

These corners and edges represent points and regions of interest in the image.

Brightness in this context refers to changes the in pixel intensity value.

References*:*

https://en.wikipedia.org/wiki/Feature_detection_(computer_vision)

https://www.mathworks.com/discovery/edge-detection.html

https://en.wikipedia.org/wiki/Computer_vision

Image reference: https://www.mathworks.com/discovery/edge-detection.html

IBM Training

Computer vision pipeline (cont.)

- Some extracted features might be irrelevant or redundant.
- After feature extraction comes feature selection.
- Feature selection is choosing a feature subset that can reduce dimensionality with the least amount of information loss.

Computer vision fundamentals

Figure 13-13. Computer vision pipeline (cont.)

For feature extraction, all the features in the image that could be extracted are present. Now, we start selecting the features that are of interest. Some extracted features maybe irrelevant or redundant.

Here are some reasons to perform feature selection:

- Reduce the dimensions of the features that were extracted from the image.
- Avoid overfitting when training the model with the features.
- Create simpler models with less training time.

References:

https://en.wikipedia.org/wiki/Feature_selection

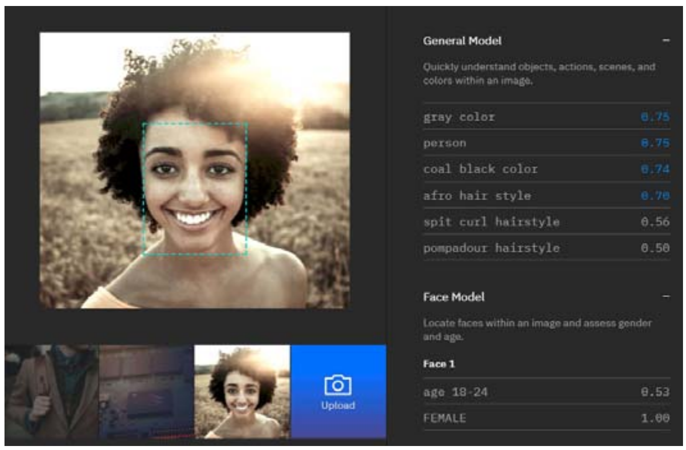
https://www.nature.com/articles/srep10312

IBM Training

Computer vision pipeline (cont.)

5. High-level processing:

- More processing is done on the segmented images to identify more features from the image.
- Example: After segmentation to partition a face region, identify features on the face, such as hair style, age, and gender.



Computer vision fundamentals

Figure 13-14. Computer vision pipeline (cont.)

5. High-level processing:

In high-level processing, further processing is done on the segmented images to identify more features from the image. For example, after the image is segmented to partition the face region, identify features on the face such as hair style, age, and gender.

In this step, object recognition or classification takes place.

IBM Training

Computer vision pipeline (cont.)

6. Decision making:

- Apply logic to make decisions based on detected images.
- Examples:
  - Security application: Grant access based on face recognition.
  - In a greenhouse environment, a notification is sent to the engineers to act if a plant is detected as suffering from decay.

Figure 13-15. Computer vision pipeline (cont.)

6. Decision making

In this stage, after getting the output from the CV pipeline, users can apply logic that is based on the findings to make decisions based on detected images.

For example, in security applications, the system can decide to grant access to a person if a face that is detected is identified as the owner of a facility.

Another example is in a greenhouse environment, where a notification is sent to the engineers to act if a plant is detected as suffering from decay.

IBM Training

Unit summary

- Describe image representation for computers.
- Describe the computer vision (CV) pipeline.
- Describe different preprocessing techniques.
- Explain image segmentation.
- Explain feature extraction and selection.
- Describe when object recognition takes place.

Computer vision fundamentals

Figure 13-16. Unit summary

IBM Training

Review questions

1. True or False: There is only one model to represent color in images, which is the RGB color model.
2. Which function is not part of the pre-processing stage?
   A. Resizing images
   B. Corner detection
   C. Contrast adjustment
   D. Noise reduction
3. True or False: In a CV pipeline, high-level processing deals with detecting different segments of the image that can be subject of analysis, like face detection.
4. Which of the following items is an example of feature extraction:
   A. Size
   B. Color
   C. Lines and edges
   D. Noise

Computer vision fundamentals

Figure 13-17. Review questions

1. False.
2. B.
3. False. In high-level processing, further processing is done on the segmented images to identify more features from the image. For example, after the area of a face is partitioned during image segmentation, features of the face such as the nose are identified by high level processing.
4. C. Lines and edges.

IBM Training

Review answers

1. True or False: There is only one model to represent color in images, which is the RGB color model.

2. Which function is not part of the pre-processing stage?
   A. Resizing Images
   B. Corner detection
   C. Contrast adjustment
   D. Noise reduction
3. True or False: In the CV pipeline, high-level processing deals with detecting different segments of the image that can be subject of analysis, like face detection.
   The answer is False.
4. Which of the following items is an example of feature extraction:
   A. Size
   B. Color
   C. Lines and edges
   D. Noise

Computer vision fundamentals

Figure 13-18. Review answers

1. False.
2. B. Corner detection.
3. False. In high-level processing, further processing is done on the segmented images to identify more features from the image. For example, after the area of a face is partitioned during image segmentation, features of the face such as the nose are identified by high level processing.
4. C. Lines and edges.

Unit 14. Introduction to IBM Watson Visual Recognition

Estimated time

00:45

Overview

This unit introduces the IBM Watson Visual Recognition service, describes its capabilities, and how to train the service.

IBM Training

Unit objectives

- List the features that are available with IBM Watson Visual Recognition.
- List the built-in models.
- Explain the capabilities of the built-in models.
- Explain the difference between the built-in and custom models.
- Describe how to train a custom model.
- Describe how to update a custom model.

Introduction to IBM Watson Visual Recognition

Figure 14-1. Unit objectives

14.1. Watson Visual Recognition overview

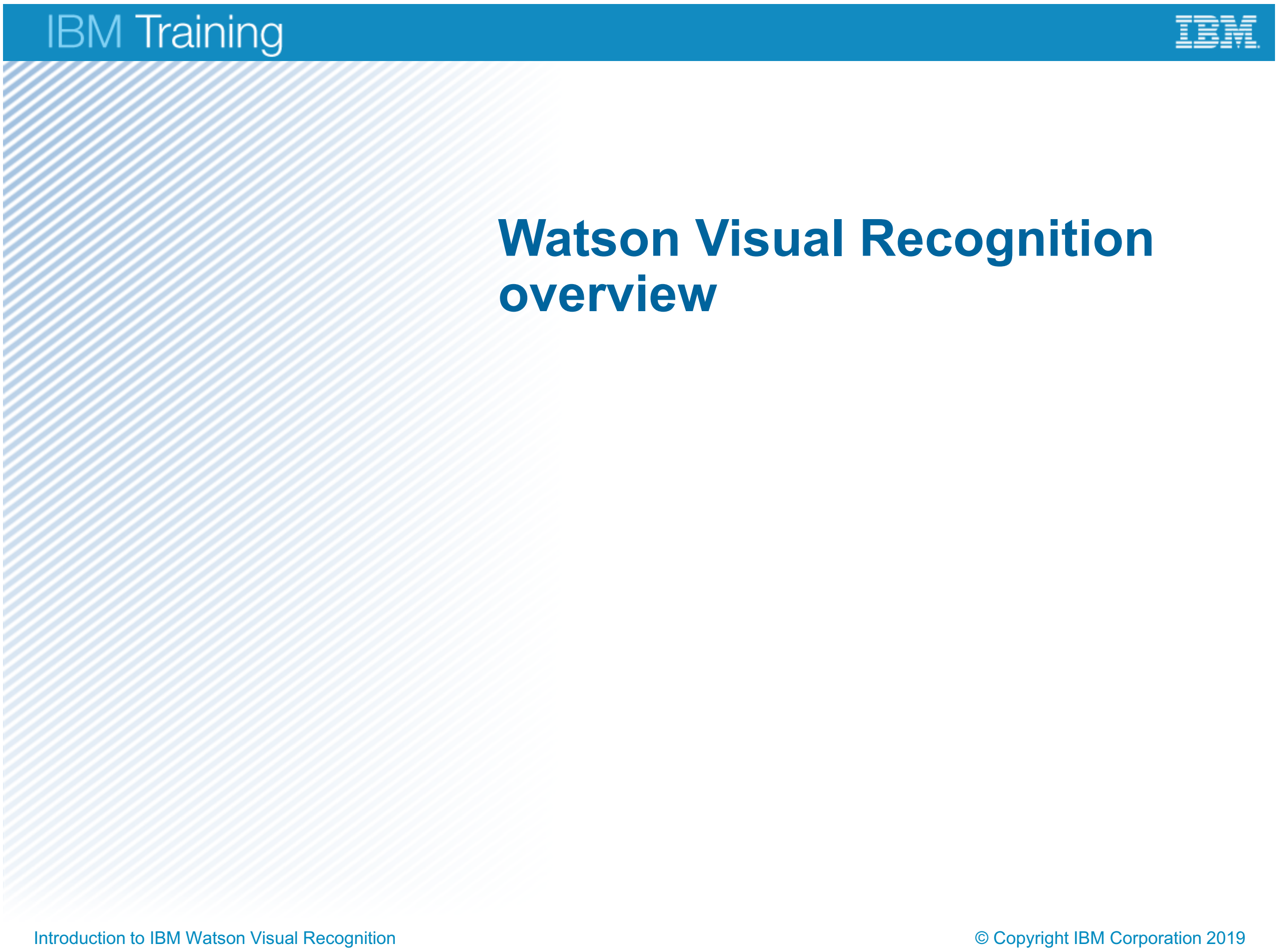


Figure 14-2. Watson Visual Recognition overview

IBM Training

Topics

- Watson Visual Recognition overview
- Visual Recognition built-in models
- Custom models

Introduction to IBM Watson Visual Recognition

Figure 14-3. Topics

IBM Training

Watson Visual Recognition service

- Uses deep learning algorithms to analyze images for scenes, objects, faces, food, and other content.
- The analysis response includes keywords that provide information about the content.
- A set of built-in classes provides highly accurate results without training.
- Can be trained to create custom models to classify images to suit specific business needs.

Introduction to IBM Watson Visual Recognition

Figure 14-4. Watson Visual Recognition service

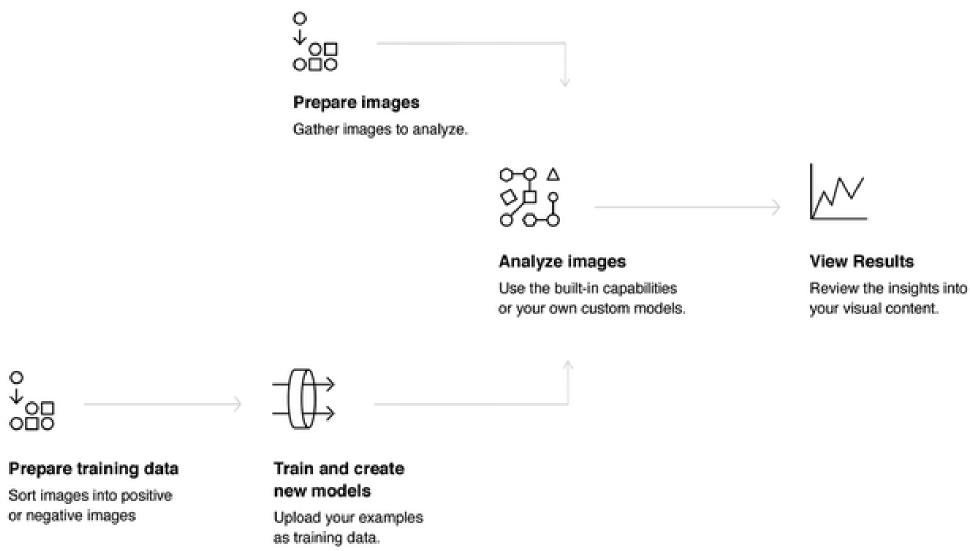
The Watson Visual Recognition service uses deep learning algorithms to analyze images for scenes, objects, faces, food, and other content. The services response includes keywords that provide information about the content. A set of built-in classes provides highly accurate results without training.

You can train custom models to create specialized classes. With a custom model, you can train the Visual Recognition service to classify images to suit your business needs. By creating a custom model, you can use the Visual Recognition service to recognize images that are not available with pre-trained classification.

IBM Training

Watson Visual Recognition service (cont.)

The following image shows the process of creating and using Watson Visual Recognition.



Introduction to IBM Watson Visual Recognition

Figure 14-5. Watson Visual Recognition service (cont.)

For built-in models, gather only the images that you want to analyze, send them to the built-in models, and then view the results. As shown in the graphic in this slide, the process starts from the top in “Prepare images.”, followed by “Analyze images.”, and then “View results.”

For custom models, first train your own custom model with your own training data. When the model is trained and ready, send the images that you want to analyze to the model, and then view the results. As shown in the graphic in this slide, the process starts from the bottom in “Prepare training data.”, followed by “Train and create new models.”, then “Analyze images.”, and then “View results.”

14.2. Visual Recognition built-in models

IBM Training

Visual Recognition built-in models

Introduction to IBM Watson Visual Recognition

Figure 14-6. Visual Recognition built-in models

IBM Training

Topics

- Watson Visual Recognition overview
- Visual Recognition built-in models
- Custom models

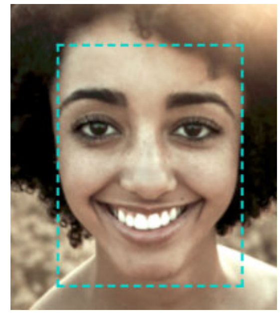
Introduction to IBM Watson Visual Recognition

Figure 14-7. Topics

IBM Training

Visual Recognition built-in models

- Built-in models that are available:
  - General model
  - Face model
  - Explicit model
  - Food model
  - Text model (private beta)



Introduction to IBM Watson Visual Recognition

Figure 14-8. Visual Recognition built-in models

A set of built-in models provides highly accurate results without training:

General model: Default classification from thousands of classes.

Face model: Facial analysis with age and gender.

Explicit model: Assess whether an image contains objectionable or adult content that might be unsuitable for general audiences.

Food model: Specifically for images of food items.

Text model (Private beta): Text extraction from natural scene images. The text model is not available for public use because it is still in beta.

We describe the General model, Face model, and Food model in the following slides.

Reference*:*

https://console.bluemix.net/docs/services/visual-recognition/index.html#models

IBM Training

Visual Recognition General model

- The General model returns classes from thousands of possible tags that are organized into categories and subcategories.
- Top-level categories:
  - Animals (including birds, reptiles, amphibians, and others)
  - Person and people-oriented information and activities
  - Food (including cooked food and beverages)
  - Plants (including trees, shrubs, aquatic plants, and vegetables)
  - Sports
  - Nature (including many types of natural formations and geological structures)
  - Transportation (land, water, and air)
  - Others, including furnishings, musical instruments, tools, colors, gadgets, devices, instruments, weapons, and buildings

Introduction to IBM Watson Visual Recognition

Figure 14-9. Visual Recognition General model

The General model returns classes from thousands of possible tags that are organized into categories and subcategories. The following list shows the top-level categories:

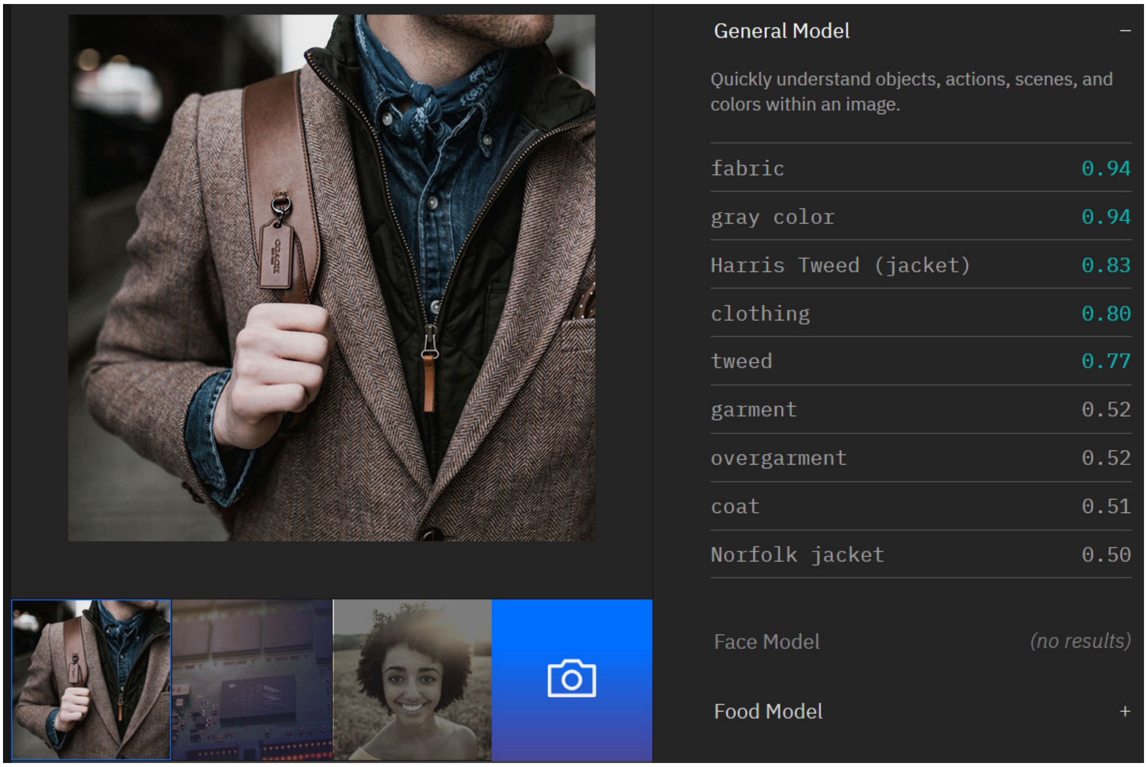
- Animals (including birds, reptiles, amphibians, and others)
- Person and people-oriented information and activities
- Food (including cooked food and beverages)
- Plants (including trees, shrubs, aquatic plants, and vegetables)
- Sports
- Nature (including many types of natural formations and geological structures)
- Transportation (land, water, and air)
- Others, including furnishings, fruits, musical instruments, tools, colors, gadgets, devices, instruments, weapons, buildings, structures and manufactured objects, clothing and garments, and flowers.

Reference*:*

https://console.bluemix.net/docs/services/visual-recognition/customizing.html#general-model

IBM Training

Visual Recognition General model (cont.)



Introduction to IBM Watson Visual Recognition

Figure 14-10. Visual Recognition General model (cont.)

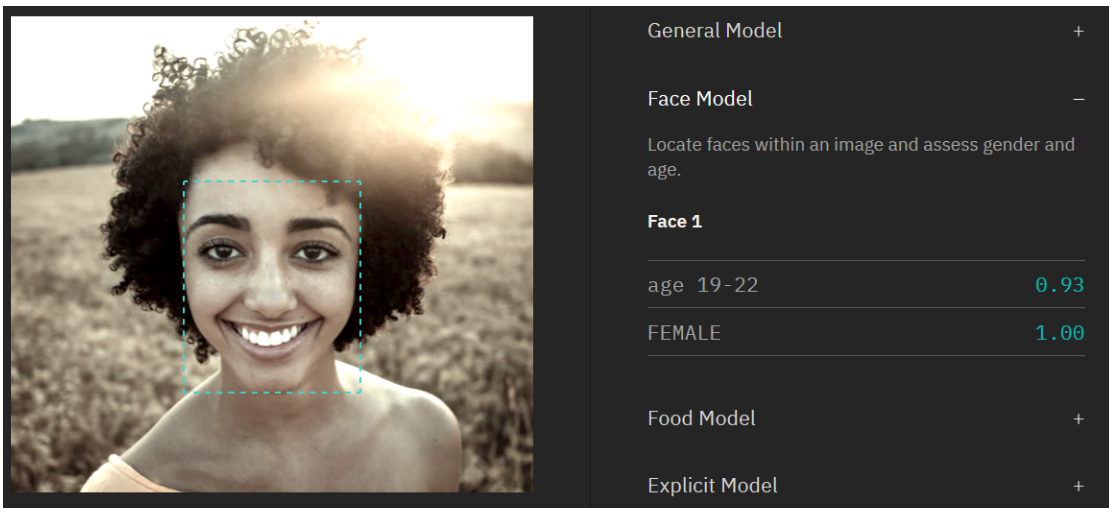
General model

The General model in the Watson Visual Recognition service analyses the image, classifies the image and the objects that it includes, and then returns the results with the associated confidence score.

IBM Training

Visual Recognition Face model

The Face model detects human faces in the image, and provides an estimated age range and gender.



Introduction to IBM Watson Visual Recognition

Figure 14-11. Visual Recognition Face model

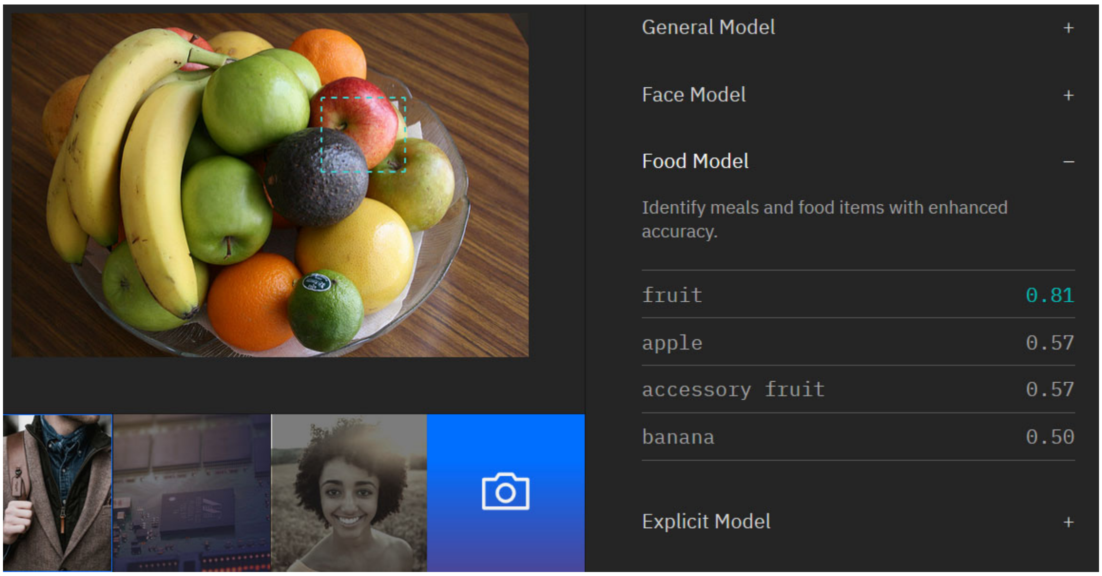
Face model

This model detects human faces in images. This slide shows an example for face detection that uses the Watson Visual Recognition Face model. The response that is returned from the model provides the location of the face in the image, which here is shown as a bounding box. This service also provides a general indication of the age range and gender of faces.

IBM Training

Visual Recognition Food model

The Food model uses a specialized vocabulary to identify food items with enhanced accuracy.



Introduction to IBM Watson Visual Recognition

Figure 14-12. Visual Recognition Food model

Food model

The Food model uses a specialized vocabulary of over 2,000 foods to identify meals, food items, and dishes with enhanced accuracy. The Food model recognizes more food and meals than the general (default) model.

14.3. Custom models

IBM Training

Custom models

Introduction to IBM Watson Visual Recognition

Figure 14-13. Custom models

IBM Training

Topics

- Watson Visual Recognition overview
- Visual Recognition built-in models
- Custom models

Introduction to IBM Watson Visual Recognition

Figure 14-14. Topics

IBM Training

Custom model training

- A custom model enables you to train the Visual Recognition service to classify images to suit your business needs.
- A custom model is a group of classes that are trained against each other.
- During training, classes are created when you upload separate compressed (.zip) files of positive examples for each class.
  - For example, to create a classifier that is called "fruit", you might upload a compressed file of images of pears, a compressed file of images of apples, and a compressed file of images of bananas in a single training call.
- You can also provide a compressed file of negative examples in the same training call to further improve your classifier.
  - For example, for the custom classifier “fruit”, you might provide a compressed file with images of various vegetables.
- You must provide at least two example .zip files: two positive examples files or one positive and one negative file.

Introduction to IBM Watson Visual Recognition

Figure 14-15. Custom model training

Although the General model is enhanced frequently, you might still need to create your own custom model for a specific case.

A custom model (also known as custom classifier) is a group of classes that are trained against each other to create a multi-faceted classifier that can identify highly specialized subjects and provide a score for individual classes.

During training, classes are created when you upload separate compressed (.zip) files of positive examples for each class. For example, to create a classifier that is called "fruit", you might upload a compressed file of images of pears, a compressed file of images of apples, and a compressed file of images of bananas in a single training call.

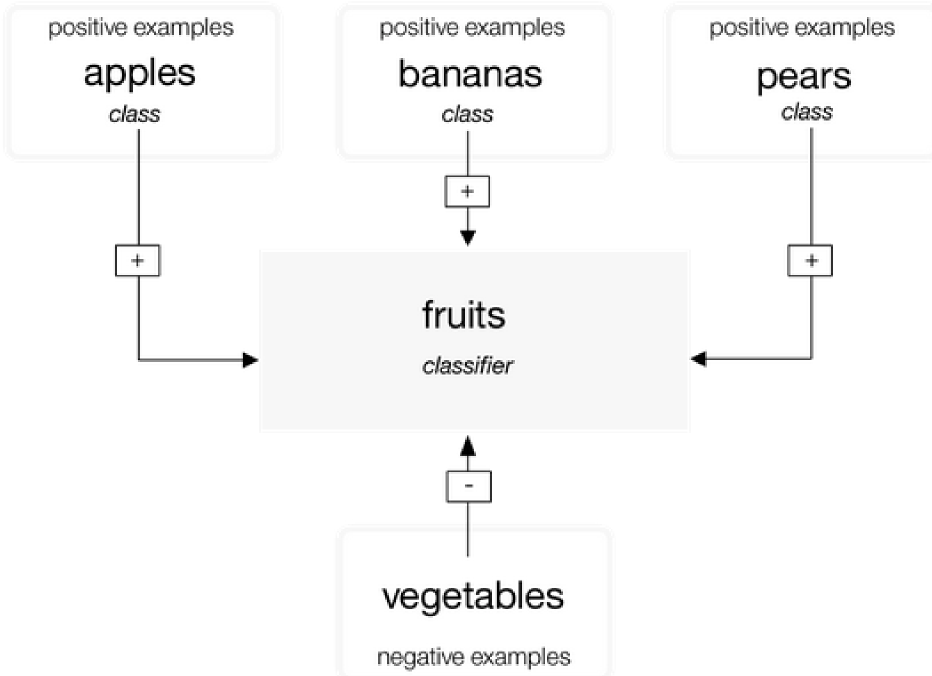
You can also provide a compressed file of negative examples in the same training call to further hone your classifier. Negative example files are not used to create a class. For example, for the custom classifier “fruit”, you might provide a compressed file with images of various vegetables.

Important

You must provide at least two example .zip files: two positive examples files or one positive and one negative file.

IBM Training

Custom model training (cont.)



Introduction to IBM Watson Visual Recognition

Figure 14-16. Custom model training (cont.)

Structure of the training data. Each class that represents a certain fruit has a separate compressed file, which contains the positive examples for the model training. A separate compressed file for negative examples can also be provided to further hone your classifier. Negative example files are not used to create a class. For the custom classifier "fruit", you might provide a .zip file with images of various vegetables.

IBM Training

Custom model training (cont.)

- The system trains three models for each set: Apples, Bananas, and Pears.
- For the Apples model, the group of pictures in “Apples” is trained as a positive example, and the other groups are trained as negative examples.
- If an image is duplicated in both the negative and positive sets, the rule is that a duplicate image is kept in the positive set.
- After training completes, the model returns the classifier “fruit” as an array that includes the classes “pears”, “apples”, and “bananas” with their respective confidence scores.

Introduction to IBM Watson Visual Recognition

Figure 14-17. Custom model training (cont.)

If you train a classifier with three sets of positive class pictures, such as Apples, Bananas, and Pears, the system trains three models internally.

For the Apples model, the group of pictures in “Apples” is trained as a positive example, and the other groups are trained as negative examples. The system then knows that bananas and pears are not apples. The other classes are used as negative examples for the Bananas and Pears models as well.

If an image is duplicated in both the negative and positive sets, the rule is that a duplicate image is kept in the positive set. After training completes, when the service identifies fruit in an image, it returns the classifier “fruit” as an array that includes the classes “pears”, “apples”, and “bananas” with their respective confidence scores.

IBM Training

Training images examples

Training images:

Testing image:

Introduction to IBM Watson Visual Recognition

Figure 14-18. Training images examples

Good example of training and testing images: The following images were used for training and testing by OmniEarth, an environmental technology company. These images are good for training because images in the training and testing sets should resemble each other with regard to angle, lighting, distance, size of subject, and other factors.

Reference*:*

https://www.ibm.com/blogs/bluemix/2016/10/watson-visual-recognition-training-best-practices/

IBM Training

Training images examples (cont.)

Training image:

Testing image:

Introduction to IBM Watson Visual Recognition

Figure 14-19. Training images examples (cont.)

Bad example of training and testing images: The following images demonstrate bad training because the training image shows a close-up shot of a single apple while the testing image shows a large group of apples taken from a distance with other visual items introduced (baskets, sign, and other items). It is possible that Watson might fail to classify the test image as ‘apples,’ especially if another class in the classifier contains training images of a large group of round objects (such as peaches, oranges, or other objects).

Try to have your training images from the setting in which your application runs and that are similar to the test images.

Reference*:*

https://www.ibm.com/blogs/bluemix/2016/10/watson-visual-recognition-training-best-practices/

IBM Training

Updating custom models

- You can update an existing custom model by adding new classes or by adding new images to existing classes.
- You must supply at least one compressed file with more positive or negative examples.
- Compressed files that contain positive examples are used to create and update “classes” to affect all of the classes in that classifier.
- The prefix that you specify for each positive example parameter is used as the class name within the new classifier. The “_positive_examples” suffix is required.

Introduction to IBM Watson Visual Recognition

Figure 14-20. Updating custom models

You can update a classifier by adding new classes or by adding new images to existing classes.

You must supply at least one compressed file with more positive or negative examples.

Compressed files that contain positive examples are used to create and update “classes” to affect all of the classes in that classifier.

The prefix that you specify for each positive example parameter is used as the class name within the new classifier. The “_positive_examples” suffix is required.

IBM Training

Updating custom models (cont.)

- Negative examples are not used to create a class within the created classifier, but they define what the updated classifier is not.
- Negative example files must contain images that do not represent the subject of any of the positive examples.
- The custom model uses binary “one versus the rest” models to train each class against the other classes.

Introduction to IBM Watson Visual Recognition

Figure 14-21. Updating custom models (cont.)

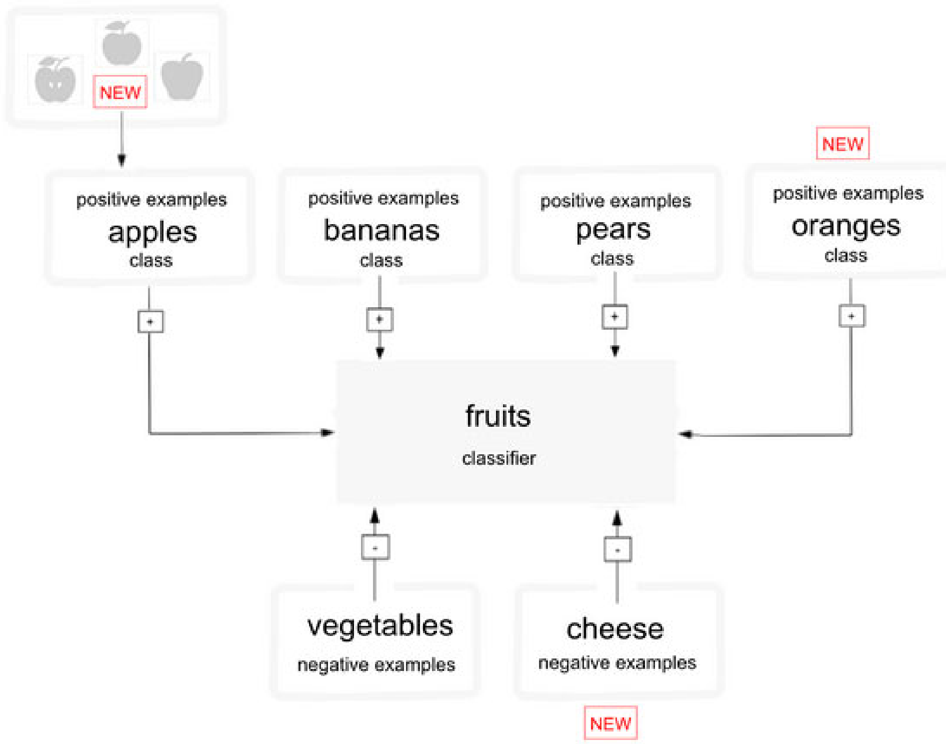
Negative examples are not used to create a class within the created classifier, but they define what the updated classifier is not.

Negative example files must contain images that do not represent the subject of any of the positive examples.

The custom model uses binary “one versus the rest” models to train each class against the other classes.

IBM Training

Updating custom models (cont.)



Introduction to IBM Watson Visual Recognition

Figure 14-22. Updating custom models (cont.)

The figure shows updating a classifier by adding new classes (oranges), by adding new images to existing classes (apples) and by adding new negative examples (cheese).

Let us assume you want to retrain your classifier with new positive classes: YellowPears and GreenPears. In order to do this, you'll need to manually look through your old pears.zip folder, and split the images out into two new folders: YellowPears.zip and GreenPears.zip.

For example, when creating YellowPears or GreenPears, every single yellow pear image from the original pears.zip training set should be exactly copied into the YellowPears.zip folder; otherwise, any image that is not copied exactly will be in the Pears training set, and used as a negative when YellowPears is trained.

Now, you simply retrain the system with YellowPears.zip and GreenPears.zip as positive examples. When you do this, the system recognizes the exact duplicate images in the YellowPears and GreenPears folders from the original pears.zip folder, and those images are retrained as positive examples for their new classes. The rule is that a duplicate image is kept in the positive set, if it is also found in both the negative and positive set for a class.

IBM Training

Custom model scores

- The classify method produces a score in the range of 0.0 - 1.0 for each image for each class.
- How do you use the scores?
  - Think about the possible actions to take in response to a classification.
  - Understand the final application to determine the “decision threshold”.

Introduction to IBM Watson Visual Recognition

Figure 14-23. Custom model scores

The classify method produces a score in the range of 0.0 - 1.0 for each image for each class.

The scores for custom classifiers are not comparable to the scores that are returned by the General built-in classifier.

How do you use the scores?

- Think about possible actions to take in response to a classification.
- This cost-benefit balance is crucial to deciding what to do with each class score, and only a user who understands the final application can determine it. The score value that is needed for the application to act is called the *decision threshold*. The service does not compute this value for you.

IBM Training

Custom model - Example

- Objective: Classify the status of your car parking spot.
- Possible statuses:
  - Your car is in the spot.
  - Some other car is in the spot.
  - The spot is empty.
  - The camera is blocked.

Introduction to IBM Watson Visual Recognition

Figure 14-24. Custom model - Example

Assume that you have a webcam that is monitoring a parking space. The objective is to classify the status of your car parking spot. You have the following possible statuses:

- Your car is in the spot.
- Some other car is in the spot.
- The spot is empty.
- The camera is blocked.

IBM Training

Custom model – Example (cont.)

- You collect training examples for each of these cases and train a custom model with four classes.
- Your application classifies images from the webcam to report the status of the spot, and the system notifies you with a message if the status is unexpected.
- When the service classifies the image from the camera, it produces four scores: myCar, unknownCar, emptySpot, and blockedCamera.
- The first action to consider is whether to send a notification.

Introduction to IBM Watson Visual Recognition

Figure 14-25. Custom model – Example (cont.)

You collect pictures as training examples for each of these cases and train a custom classifier with four classes.

Your application classifies images from the webcam to report the status of the spot, and the system notifies you with a message if the status is unexpected. When the service classifies the image from the camera, it produces four scores: myCar, unknownCar, emptySpot, and blockedCamera.

Now, based on this use case, the first action might be to send a notification about the parking status.

IBM Training

Custom model – Example (cont.)

- Suppose that you write your code to send a notification if the myCar score is less than 0.75, or if one of the other scores is greater than 0.6.
- During the day, there is one false alarm every three hours:
  - Reason: People walk by and obscure the car.
  - Accepting these alarms at night is annoying.
- The notification logic and threshold likely vary depending on:
  - The perceived risk of car theft.
  - The accuracy of your model.
  - The amount of annoyance that is caused by a false alarm.

Introduction to IBM Watson Visual Recognition

Figure 14-26. Custom model – Example (cont.)

Suppose you park in your spot and have the service start classifying the images. You see that the myCar score is computed as 0.8 on average over a few hours, the unknownCar score hovers around 0.3, emptySpot is around 0.15, and blockedCamera is around 0.1. Using this data, you write your code to notify you if the myCar score is less than 0.75, or if one of the other scores is greater than 0.6.

During the day, you receive approximately one false alarm every three hours when people walk by and obscure the car. The system sends you the photo along with the notice so that you can see that an issue does not exist. This approach works, but at night those false alarms every three hours become annoying. Your preferences for day versus night notification reflect the higher cost of a false alarm at night time for your application.

Therefore, the notification logic and threshold likely vary, depending on the perceived risk of car theft, the accuracy of your classifiers, and the amount of annoyance that is caused by a false alarm.

Similarly, as a person, you might face the same tradeoff. If the system notifies you that the camera was blocked, the accompanying image likely is all black or gray. Do you go to check on the car in person or ignore it? Your decision depends on your other priorities and the perceived risks.

IBM Training

Unit summary

- List the features that are available with IBM Watson Visual Recognition.
- List the built-in models.
- Explain the capabilities of the built-in models.
- Explain the difference between the built-in and custom models.
- Describe how to train a custom model.
- Describe how to update a custom model.

Introduction to IBM Watson Visual Recognition

Figure 14-27. Unit summary

IBM Training IBM

Review questions

1. True or False: By using the Watson Visual Recognition service, you can create your own custom model.

2. What is the highest confidence value that is presented by the Watson Visual Recognition model?
   A. 100.
   B. 0.9.
   C. 1.
   D. 10.

3. To use the Watson Visual Recognition service, the first thing that you do is:
   A. Create your own custom model.
   B. Create a Watson Visual Recognition service instance.
   C. Use the General model to detect faces.
   D. Use the custom model to detect faces.

Introduction to IBM Watson Visual Recognition

Figure 14-28. Review questions

1. True.
2. C. The answer is 1.
3. B.

IBM Training

Review questions (cont.)

4. True or False: The Watson Visual Recognition service must be trained before it can be used.

5. To train a custom model (classifier) you must provide at least:
   A. One positive example.
   B. One negative example.
   C. Two positive examples or one positive example and one negative example.
   D. Two positive examples and two negative examples.

Introduction to IBM Watson Visual Recognition

Figure 14-29. Review questions (cont.)

4. False. Visual Recognition includes a set of built-in models that provides highly accurate results without training.
5. C. At least two example .zip files must be provided: Two positive examples or one positive example and one negative example.

IBM Training

Review answers

1. True or False: By using the Watson Visual Recognition service, you can create your own custom classifier.
2. What is the highest confidence value that is presented by Watson Visual Recognition Classifier?
   A. 100.
   B. 0.9.
   C. 1.
   D. 10.
3. To use the Watson Visual Recognition service, the first thing that you must do is:
   A. Create your own custom classifier.
   B. Create a Watson Visual Recognition service instance.
   C. Use the default classifier to detect faces.
   D. Use the custom classifier to detect faces.

Introduction to IBM Watson Visual Recognition

Figure 14-30. Review answers

1. True.
2. C. The answer is 1.
3. B.

IBM Training

Review answers (cont.)

4. True or False: The Watson Visual Recognition service must be trained before it can be used.

5. To train a custom model (classifier) you must provide at least:
   A. One positive example.
   B. One negative example.
   C. Two positive examples or one positive example and one negative example.
   D. Two positive examples and two negative examples.

Introduction to IBM Watson Visual Recognition

Figure 14-31. Review answers (cont.)

Write your answers here:

4. False. Visual Recognition includes a set of built-in models that provides highly accurate results without training.

5. C. At least two example .zip files must be provided: Two positive examples or one positive example and one negative example.

IBM Training

Exercise: Classifying images with Watson Visual Recognition

Introduction to IBM Watson Visual Recognition

Figure 14-32. Exercise: Classifying images with Watson Visual Recognition

IBM Training

Exercise objectives

- This exercise guides you through using the default models in IBM Watson Visual Recognition to classify an image, detect faces, and recognize text in an image.
- After you learn how to classify an image by using the default General model, you will learn to create and train a custom model.
- After completing this exercise you should be able to:
  - Create a Watson Visual Recognition service and obtain the API key value.
  - Use Visual Recognition API methods to:
    - Classify images
    - Detect faces in an image
    - Recognize text in an image
  - Create and train a custom model.

Introduction to IBM Watson Visual Recognition

Figure 14-33. Exercise objectives

After completing this exercise you should be able to:

- Create a Watson Visual Recognition service and obtain the API key value.
- Use Visual Recognition API methods to:
  - Classify images.
  - Detect faces in an image.
  - Recognize text in an image.
- Create and train a custom model.

Unit 15. Designing and building an intelligent solution

Estimated time

00:45

Overview

This unit explains the benefits of integrating multiple Watson services to build a comprehensive intelligent solution. This unit presents two intelligent solutions use cases: A cognitive banking FAQ chatbot and an intelligent procurement system.

IBM Training

Unit objectives

- Explain the need to integrate multiple IBM Watson services to build an intelligent solution.
- Describe the general outline for the integration of IBM Watson Assistant with other services and applications.
- Explain the key concepts that enable Watson Assistant integration.
- Describe the integration flow between IBM Watson Assistant, IBM Watson Discovery, IBM Watson Natural Language Understanding, and IBM Watson Tone Analyzer to build the cognitive banking chatbot.
- Describe the integration flow between IBM Watson Knowledge Studio and Watson Discovery.

Designing and building an intelligent solution

Figure 15-1. Unit objectives

15.1. Integrating IBM Watson services to build a solution

Figure 15-2. Integrating IBM Watson services to build a solution

This section describes the benefits of integrating multiple AI services and introduces typical applications that you can create by combining multiple Watson services.

IBM Training

Topics

- Integrating IBM Watson services to build a solution
  - Watson Assistant integration with other services
  - Watson Assistant, Discovery, Natural Language Understanding, and Tone Analyzer integration flow
  - Knowledge Studio and Watson Discovery integration flow

Designing and building an intelligent solution

Figure 15-3. Topics

IBM Training

Benefits

- A single Watson service adds artificial intelligence (AI) capabilities to the application in one specific area.
- Integrating multiple Watson services adds more than one AI capability to the application, which enhances the application with richer cognitive capabilities.

Designing and building an intelligent solution

Figure 15-4. Benefits

Using one Watson service adds artificial intelligence (AI) capabilities to the application in one specific area.

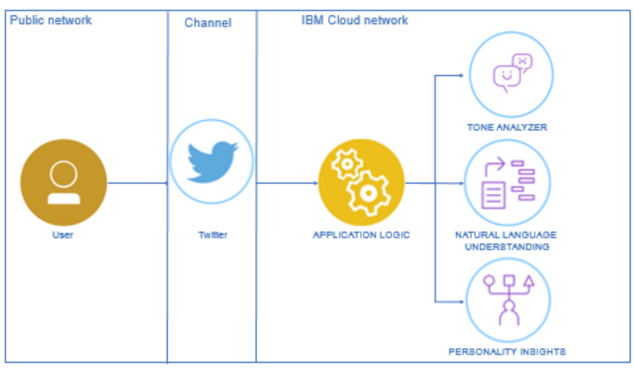
Integrating multiple Watson services adds more than one AI capability to the application, which enhances the application with richer cognitive capabilities.

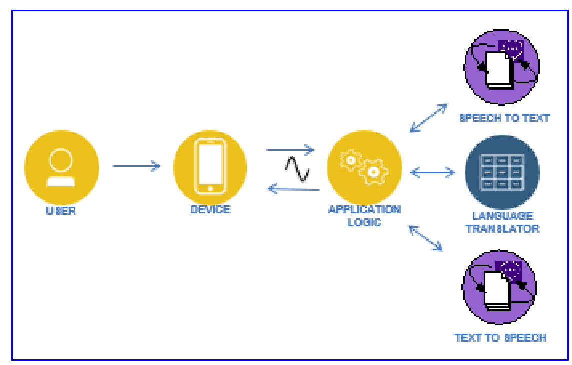
IBM Training

Building solutions with Watson services

Watson services on IBM Cloud are a set of REST APIs:

- Easy to integrate as solution components.
- Must be integrated with other parts of a solution, for example, back-end systems, social media channels, and more.





Designing and building an intelligent solution

Figure 15-5. Building solutions with Watson services

Watson services on IBM Cloud are a set of REST APIs. They are easy to use as a component of a solution within an application. They must be integrated with various other parts of the solution so that your users can interact with your instance of the Watson service.

The figure shows examples of applications that can be built by integrating multiple Watson services:

- Sentiment, personality, and tone analysis: This application integrates three Watson services to process tweets from users and analyze sentiment (Natural Language Understanding), tone (Tone Analyzer), and the personality of the user (Personality Insights).
- Real-time transcription: This Android application performs real-time transcription and translation by integrating IBM Watson Speech to Text, IBM Watson Text to Speech, and IBM Watson Language Translator services.

An application orchestrates the services integration and provides the logic of the solution.

IBM Training IBM

Resources for the integration of Watson services

Assets that are available to assist with the integration:

- Watson SDKs
- API Reference and documentation
- Watson Starter Kits

Designing and building an intelligent solution

Figure 15-6. Resources for the integration of Watson services

Although IBM provides several assets to help the integration effort, such as Watson SDKs, API references, and sample applications, other users have contributed GitHub repositories too. There are endless possibilities for the integration of Watson services with other channels or specific external systems.

It is not possible to cover the broad range of options and possibilities for integrating Watson services. This presentation provides an introduction to Watson services integration and an example that is implemented in the exercise that follows the presentation.

References:

Watson SDKs: https://console.bluemix.net/docs/services/watson/getting-started-sdks.html#sdks

API reference and documentation: https://console.bluemix.net/developer/watson/documentation

Watson Starter Kits: https://console.bluemix.net/developer/watson/starter-kits

15.2. Watson Assistant integration with other services

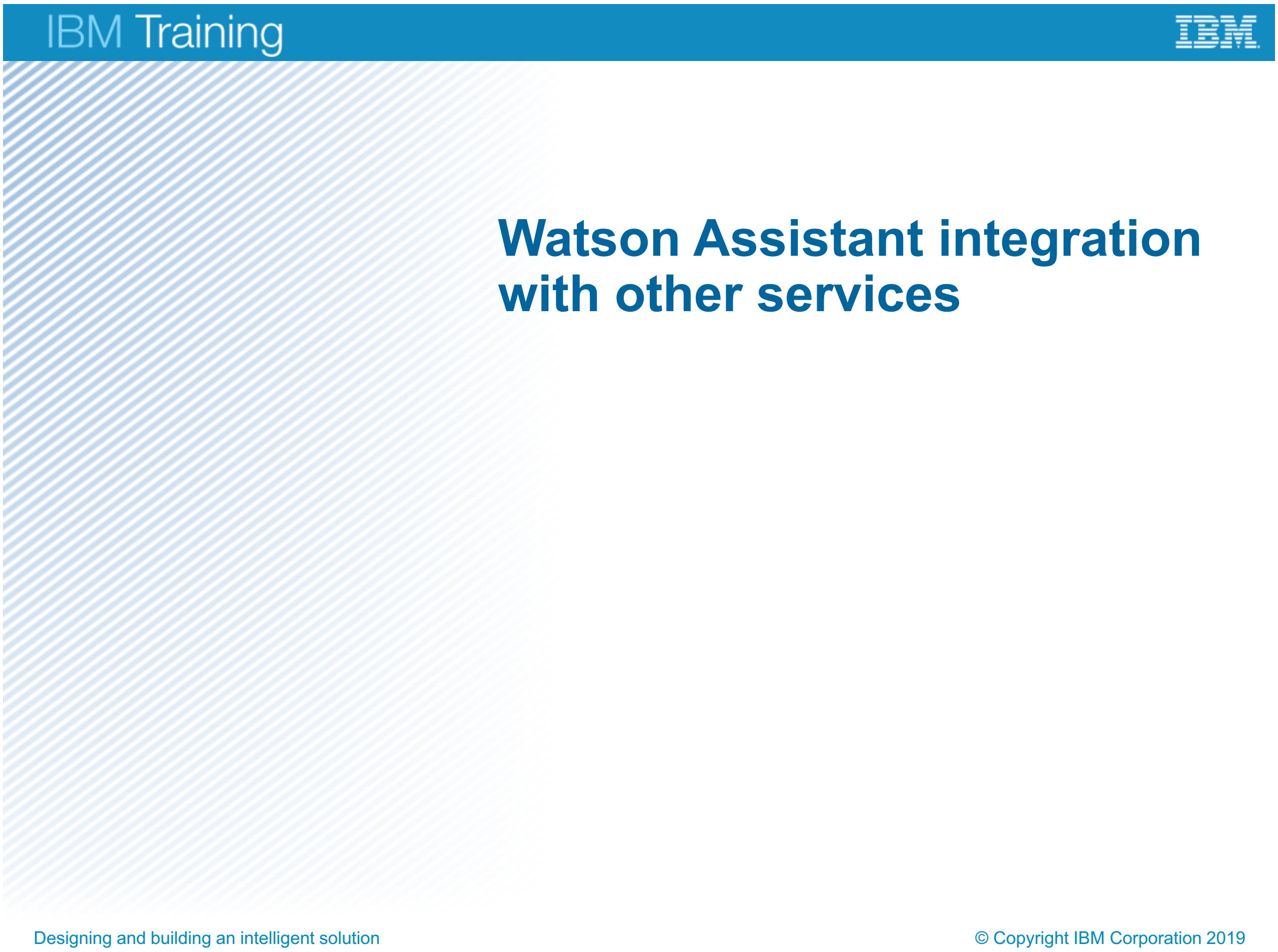


Figure 15-7. Watson Assistant integration with other services

This section introduces the integration of Watson Assistant with other Watson services and channels.

IBM Training

Topics

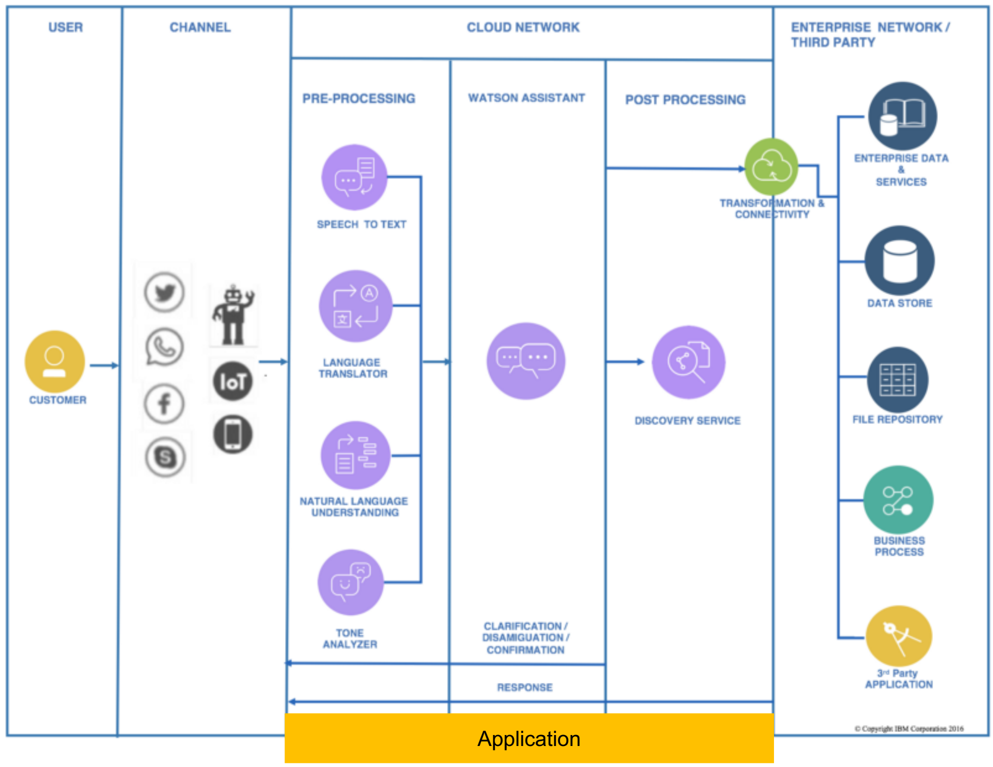
- Integrating IBM Watson services to build a solution
- Watson Assistant integration with other services
- Watson Assistant, Discovery, Natural Language Understanding, and Tone Analyzer integration flow
- Knowledge Studio and Watson Discovery integration flow

Designing and building an intelligent solution

Figure 15-8. Topics

IBM Training

Watson Assistant integration: General outline



Designing and building an intelligent solution

Figure 15-9. Watson Assistant integration: General outline

Although it is impossible to describe how to integrate with every possible integration point, there is a general outline that most integrations follow. The following major components are parts of a solution that integrates chatbots with other services and applications:

1. **Front end or channel**: This component can be a web page or an application window where the users type their questions, the responses are shown, images are displayed, and so on. It might be a messaging channel, an embedded chat widget, a mobile app, or even SMS messaging.
2. **Watson Assistant:** The “intelligence” behind the interaction is the Watson Assistant service. Watson Assistant takes the inputs, understands them, and drives what takes place next. Some examples of the output are displaying a response, detecting the intent of what is being asked, using a multi-modal interaction like showing a map, playing a video, or something more complex, such as reading from and writing to a database or even calling an enterprise service. Watson Assistant can be trained on your content, but the interaction follows some structure. Depending upon the content source or type, you might have to use some data transformation or connectivity patterns.
3. **Application layer**: This is the middle layer, which varies the most across different solutions. There is one job that this layer must accomplish: Passing information from the front end to the chatbot (Watson Assistant), including system **context**, and passing it back to carry the conversation. It is a translation layer that moves data from one side to the other and back.

4. **Pre-processing**: The flow gets more complex if you have additional interactions that you want to integrate. For example, suppose that you want to add Tone Analyzer to build an empathetic chatbot. This action is typically called *pre-processing* because it happens before calling Watson Assistant. The application layer takes the user input, runs it through this pre-processor layer, in this case to get the tone of the user statement, attaches the response, in this case tone analysis results as context for Watson Assistant, and then pass it on to Watson Assistant.
5. **Post-processing**: This is the step where the logic that is necessary to respond to the user query is. It happens after calling Watson Assistant but before returning a response to the front-end layer. The Watson Discovery service is an example of a post-processing service. Another use case for the post-processing layer is writing information to a database. For example, a user orders a *large pepperoni pizza*. Your application might have to make two callouts. The first call places the order in your point of sale (POS) system to order the pizza, and the second call might write the order to a database. This way, the next time the user logs in, they can say "*order my usual"* or something similar. Watson Assistant typically returns some *action* for your application to perform and it also returns some text. Your application can do the activities that are defined, and then show a message such as "*I'll remember that it is your favorite, and it's on the way. Thank you for your order*." Another approach is to directly call external services from the Watson Assistant dialog to enrich your conversation or allow users to complete activities by using post-processing.

Reference: *Integrate Watson Assistant With Just About Anything*, found at:

https://medium.com/ibm-watson/integrate-watson-assistant-with-just-about-anything-695bc1d29875

IBM Training

Key concepts to enable Watson Assistant integration

Communicating with the Watson services:

- Create a wrapper for the Watson service.
- Specify the service credentials.
- Specify the service version.
- Watson SDKs provide mechanisms for instantiating a service wrapper for several programming languages.
- Example for Watson Assistant in Node.js:

```
// Sets up Assistant service wrapper

var AssistantV2 = require('watson-developer-cloud/assistant/v2');

// Set up Assistant service wrapper.
var service = new AssistantV2({
  iam_apikey: '{apikey}', // replace with API key
  version: '2018-09-20'
});

var assistantId = '{assistant_id}'; // replace with assistant ID
var sessionId;
```

Designing and building an intelligent solution

Figure 15-10. Key concepts to enable Watson Assistant integration

The first step is to create a wrapper for the Watson Assistant service.

The wrapper is an object to which you send input and receive output from the service. When you create the service wrapper, specify the authentication credentials from the service key and the version of the Watson Assistant API (and the other services) that you use.

In this Node.js example, the wrapper is an instance of AssistantV2, which is stored in the variable `service`.

The Watson SDKs for other languages provide equivalent mechanisms for instantiating a service wrapper.

IBM Training

Key concepts to enable Watson Assistant integration (cont.)

Context:
- It maintains state information.
- It is an object that is passed back and forth between your application and the Watson Assistant service.
- Your application must maintain the context from one turn of the conversation to the next.
- Also used to store other data that you want to pass between your application and the Watson Assistant service. Examples:
  - Persistent data that you want to maintain throughout the conversation, such as a customer's name or account number.
  - Any data that you want to track, such as the status of the option settings.
- In each round of the conversation, send back the `response.context` object that was received in the previous round.

```
service.message({
      input: { text: newMessageFromUser },
      context : response.context,
    }, processResponse)
```

Designing and building an intelligent solution

Figure 15-11. Key concepts to enable Watson Assistant integration (cont.)

State information for your conversation is maintained by using the *context*. The context is an object that is passed between your application and the Watson Assistant service. It is the responsibility of your application to maintain the context from one turn of the conversation to the next. The context includes a unique identifier for each conversation with a user, and a counter that is incremented with each turn of the conversation.

In addition to maintaining your place in the conversation, the context can also be used to store any other data that you want to pass between your application and the Watson Assistant service. This information can include persisting the data that you want to maintain throughout the conversation (such as a customer's name or account number), or any other data that you want to track (such as the status of option settings).

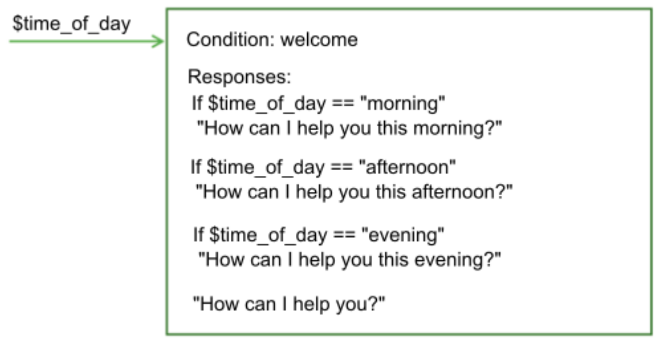
The context contains variables that are meant to be persistent. To pass information from one call to the next, the application developer must pass the previous API call's response context with each subsequent API call. For example, the dialog can collect the user's name and then refer to the user by name in subsequent nodes.

IBM Training

Key concepts to enable Watson Assistant integration (cont.)

Context variable:

- A context variable is a variable that you define in a dialog node or from an application.
- Optionally, you can specify a default value for a context variable.
- Nodes and application logic can change its value.
- Example: Your application sets a `$time_of_day` context variable, and passes it to the dialog node, which can use the information to tailor the greeting it displays to the user.



Designing and building an intelligent solution

Figure 15-12. Key concepts to enable Watson Assistant integration (cont.)

A context variable is a variable that you define in a node or in the application, and optionally specify a default value for it.

You can **pass information from the application to the dialog** by setting a context variable and passing the context variable to the dialog.

For example, your application can set a **$time_of_day** context variable, and pass it to the dialog, which can use the information to tailor the greeting it shows to the user.

In this example, the dialog knows that the application sets the variable to one of these values: *morning*, *afternoon*, or *evening*. It can check for each value, and depending on which value is present, return the appropriate greeting. If the variable is not passed or has a value that does not match one of the expected values, then a more generic greeting is displayed to the user.

The dialog can also add context variables to **pass information from one node to another** or to update the values of context variables. As the dialog asks for and gets information from the user, it can track the information and reference it later in the conversation. For example, in one node you might ask users for their name, and in a later node address them by name.

References:

https://www.ibm.com/cloud/garage/architectures/cognitiveConversationDomain

https://console.bluemix.net/docs/services/conversation/dialog-actions.html#dialog-actions

https://www.ibm.com/blogs/bluemix/2017/11/enhance-chatbot-conversation-context-variables-system-entities/

IBM Training

Key concepts to enable Watson Assistant integration (cont.)

Using intents, entities, and context for application and Watson Assistant integration:

- Watson Assistant identifies intents and entities in the user's input.
- The Watson Assistant dialog passes intent, entities, and context to the application.
- The application performs specific tasks that are based on intent, entity, and context values.

Designing and building an intelligent solution

Figure 15-13. Key concepts to enable Watson Assistant integration (cont.)

In general, any integration between the application and Watson Assistant is done by using intents, entities, and context objects:

- Watson Assistant dialog detects the intents and entities in the user's response.
- Watson Assistant dialog passes the intents, entities, and context objects to the application in JSON format.
- The application performs specific tasks according to the intents, entities, and context values that are sent to the application.

IBM Training

Key concepts to enable Watson Assistant integration (cont.)

- Callback function:
  - A function that is passed into another function as an argument.
  - A callback function is called at the completion of a task.
- Using the callback function for application and Watson Assistant integration:
  - Application calls the `assistant.message` API by passing the `payload` object and callback function in its arguments.
  - Watson Assistant processes the task, sets the output, intents, entities, and context in the response, and then returns to the callback function.
  - The response is processed by the application in the callback function.

Designing and building an intelligent solution

Figure 15-14. Key concepts to enable Watson Assistant integration (cont.)

Callback is an asynchronous equivalent for a function. A callback function is called at the completion of a task.

The callback function is used in the application and Watson Assistant integration:

1. The application calls the **`assistant.message`** API by passing the payload object and the callback function in its arguments.
2. Watson Assistant processes the task and sets the *output*, which is the chatbot response, intents, entities, and context in the response to the API, and then returns back to the callback function.
3. In the callback function, the response containing the output, intents, entities, and context is processed.

IBM Training

Making programmatic calls from a dialog node

- Calling external applications or services. Use cases:
  - Include the weather forecast in the chatbot.
  - Validate information that was collected from the user.
  - Perform calculations.
  - Check a flight status that is provided by an air traffic service.
  - Interact with an external application, for example, restaurant reservations.
- Approaches for integration:
  - Create an orchestration application that acts as the *middle layer* between Watson Assistant and the external service.
  - Call the external service directly from Watson Assistant dialog node.
- IBM Cloud Functions:
  - A Function-as-a-Service (FaaS) platform based on Apache OpenWhisk that runs functions in response to incoming events or direct invocations.
  - Provide a set of small, distinct, and independent actions.
  - Developers focus on the application logic and creating *actions* that are run on demand.
  - Cloud Functions actions contain your function code and are started by events or REST API calls.
  - Cloud Functions actions can be called from a dialog node.

Designing and building an intelligent solution

Figure 15-15. Making programmatic calls from a dialog node

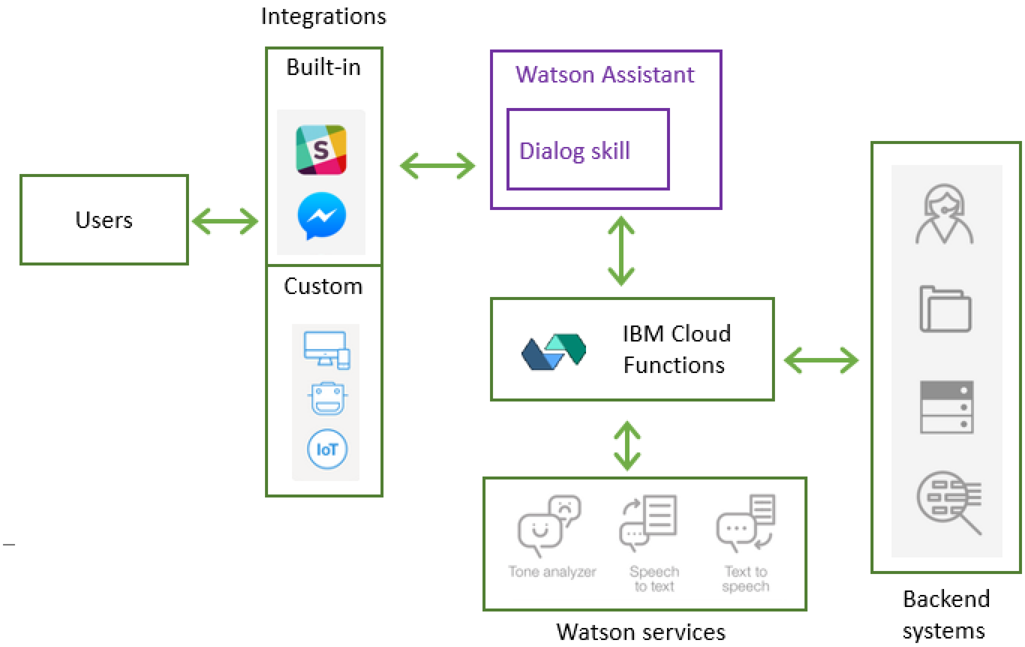
There are many reasons to call external applications or services to have richer interactions in the chatbot's dialogs. The typical way of achieving such integration is to create an orchestration application. This orchestration application is the *glue logic* between Watson Assistant and an external service. This approach is described in this presentation and the exercise that follows.

However, another approach is for a Watson Assistant dialog node to call the external service directly. This approach enables your client application to call the external service or use IBM Cloud Functions.

IBM Cloud Functions is based on Apache OpenWhisk. It is a Function-as-a-Service (FaaS) platform that runs functions in response to incoming events or direct invocations.

IBM Training

Watson Assistant integration architecture with Cloud Functions



Designing and building an intelligent solution

Figure 15-16. Watson Assistant integration architecture with Cloud Functions

The figure show the general outline for integration introduced earlier in this presentation but in this case, the architecture includes IBM Cloud Functions as the orchestration component. In this case, Watson Assistant calls the external service directly by using Cloud Functions.

IBM Training

Making programmatic calls from a dialog node (cont.)

Defining a programmatic call in a dialog node

- Add an *actions* array to the node by using the JSON editor.
- The `actions` array specifies the programmatic calls to make from the dialog.
- Store the response that is received from the service in a context variable and use it to condition the node's response.

```
{
 "context": {
   "variable_name" : "variable_value"
  },
  "actions": [
    {
      "name":"<actionName>",
      "type":"client | cloud_function | server | web_action",
      "parameters": {
        "<parameter_name>":"<parameter_value>",
        "<parameter_name>":"<parameter_value>"
      },
      "result_variable": "<result_variable_name>",
      "credentials": "<reference_to_credentials>"
    }
  ],
  "output": {
    "text": "response text"
  }
}
```

Designing and building an intelligent solution

Figure 15-17. Making programmatic calls from a dialog node (cont.)

To make programmatic calls directly from a dialog node, add an *actions* array to the node by using the JSON editor.

The *actions* array specifies the programmatic calls to make from the dialog. It can define up to five separate programmatic calls. The *actions* array includes action name, parameters list, and the response from the external function or service.

The figure shows the syntax to define the programmatic call:

- **<actionName>**: The name of the action or service to call.
- **<type>**: Indicates the type of call to make. Choose from the following types:
  - **client**: Sends a message response with programmatic call information in a standardized format that your external client application can use to perform the call or function, and get a result on behalf of the dialog. The JSON object in the response body specifies the service or function to call, any associated parameters to pass with the call, and how the result should be sent back.
  - **cloud_function**: Calls an IBM Cloud Functions action (one or more) directly. You must define the action itself separately by using IBM Cloud Functions.
  - **web_action**: Calls an IBM Cloud Functions web action (one or more) directly. You must define the web action itself separately by using IBM Cloud Functions.

- **<action_parameters>:** Any parameters that are expected by the external program, which is specified as a JSON object.
- **<result_variable_name>**: The name to use to reference the JSON object that is returned by the external service or program. The result is stored as a context variable so it can be displayed in the node response or accessed by dialog nodes that are triggered later.

Reference:

https://console.bluemix.net/docs/services/conversation/dialog-actions.html#dialog-actions

15.3. Watson Assistant, Discovery, Natural Language Understanding, and Tone Analyzer integration flow

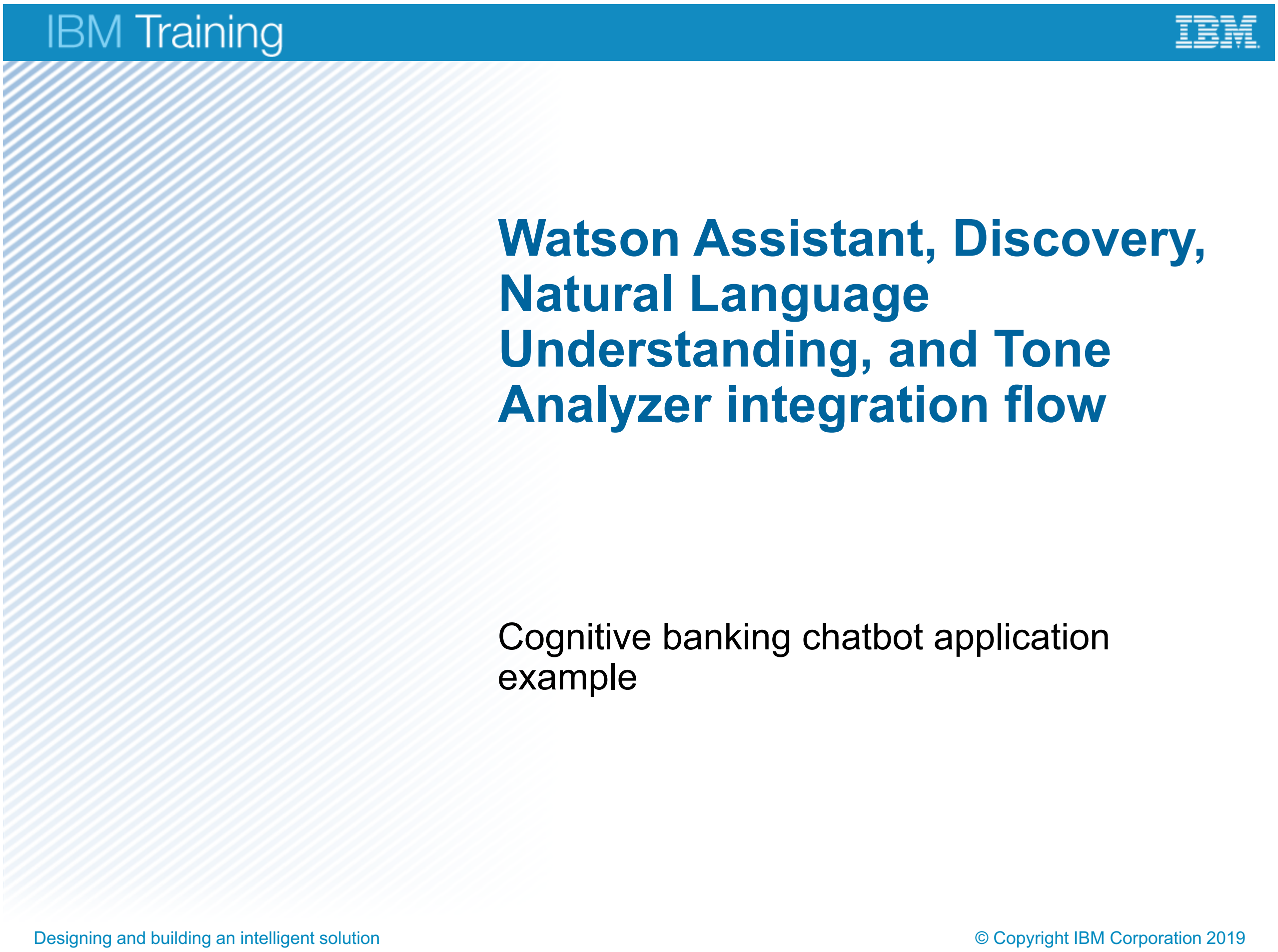


Figure 15-18. Watson Assistant, Discovery, Natural Language Understanding, and Tone Analyzer integration flow

This section describes the integration flow that is implemented by the cognitive banking chatbot application. This flow is a practical example of integrating multiple Watson services, an application that orchestrates the services integration, and back-end banking systems.

IBM Training

Topics

- Integrating IBM Watson services to build a solution
- Watson Assistant integration with other services
- Watson Assistant, Discovery, Natural Language Understanding, and Tone Analyzer integration flow
- Knowledge Studio and Watson Discovery integration flow

Designing and building an intelligent solution

Figure 15-19. Topics

IBM Training

Integration example: Cognitive banking chatbot

- The cognitive banking chatbot demonstrates how the Watson Assistant dialog is enhanced by integrating with the following items:
  - Natural Language Understanding to identify entities.
  - Tone Analyzer to detect the user's emotions.
  - Discovery to retrieve answers from a collection of FAQ documents.
- This unit explains the integration flow of the solution.
  - Exercise 11 describes the step-by-step implementation.

Designing and building an intelligent solution

Figure 15-20. Integration example: Cognitive banking chatbot

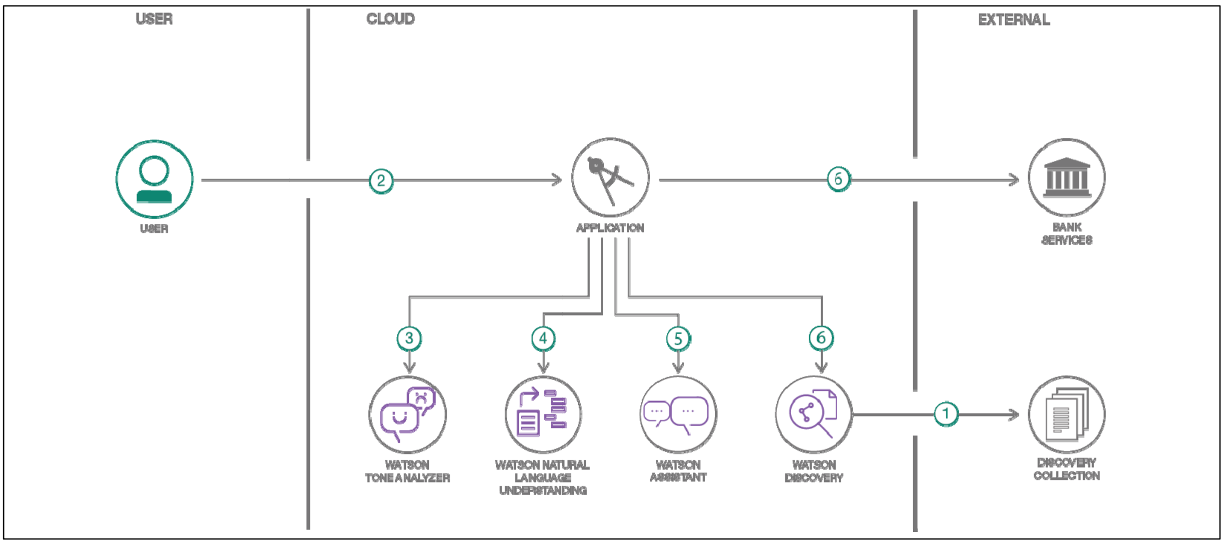
The cognitive banking chatbot provides an example of integration of multiple Watson services. The Watson Assistant dialog is enhanced by adding capabilities that are provided by other Watson APIs:

- Natural Language Understanding to identify entities
- Tone Analyzer to detect the user's emotions
- Discovery to retrieve answers from a collection of FAQ documents

Exercise 11 shows the full implementation steps for the "Cognitive banking chatbot" solution.

IBM Training

Cognitive banking chatbot integration flow



Designing and building an intelligent solution

Figure 15-21. Cognitive banking chatbot integration flow

Cognitive banking chatbot integration flow:

1. The FAQ documents are added to the Discovery collection.
2. The user interacts with a chatbot through the application UI.
3. The user input is processed with Tone Analyzer to detect anger. An anger score is added to the context.
4. The user input is processed with Natural Language Understanding. The context is enriched with Natural Language Understanding-detected entities and keywords, for example, a location.
5. The input and enriched context is sent to Assistant. Assistant recognizes intent, entities, and dialog paths. It responds with a reply or action.
6. Optionally, a requested action is performed by the application, which might include one of the following items:
    a. Look up additional information from bank services to append to the reply.
    b. Use Discovery to reply with an answer from the FAQ documents.

IBM Training

Building the cognitive banking chatbot application

1. Create the Watson services on IBM Cloud:
   - Watson Assistant
   - Natural Language Understanding
   - Tone Analyzer
   - Discovery
2. Import the skill (formerly workspace) to Watson Assistant.
3. Load the Discovery documents.
4. Associate the application with each Watson service instance:
   - Get the services credentials from IBM Cloud.
   - Set the service credentials in the .env file of the Node.js application.

Designing and building an intelligent solution

Figure 15-22. Building the cognitive banking chatbot application

This section describes, at a high level, how the cognitive banking chatbot solution is built by using a Node.js application as the orchestration application:

1. Create the Watson services (Watson Assistant, Natural Language Understanding, Tone Analyzer, and Discovery) on IBM Cloud.
2. Import the pre-developed workspace (*skill* in Watson Assistant V2) to the Watson Assistant service.
3. Load the FAQ documents in to the Discovery service.
4. Get the service credentials from the services on IBM Cloud, then set them in the .env file of the Node.js application.

IBM Training

Building the cognitive banking chatbot application (cont.)

5. Set up the communication with Watson services from the Node.js app:
   - Import each Watson service SDK.
   - Initialize each service with the service credentials and version.
6. Send input from the application to the Watson Assistant service:
   - Use the service's `message` API.
   - Set the workspace ID (Skill ID) in the payload.
   - Set the user's input in the payload.
   - Set the context object in the payload to maintain the state of the conversation.

Designing and building an intelligent solution

Figure 15-23. Building the cognitive banking chatbot application (cont.)

5. To set up the communication with the Watson services from Node.js application, complete the following steps:
   - Import the Watson services SDKs to the Node.js application.
   - Initialize each service with the credentials and the required version of the API.
6. Send the user's input to the Watson Assistant service:
   - Use the message API of the Watson Assistant service.
   - Set the Workspace ID (**Skill ID)** in the payload of the API.
   - Set the user's input in the payload of the API.
   - Set the context in the payload of the API to maintain the state of the conversation.
   - Set the callback function in the API arguments to process the response that is returned from the API.

IBM Training

Building the cognitive banking chatbot application (cont.)

7. Receive and process the response from Watson Assistant:
   - After the `message` API finishes, it returns the response to the callback function.
   - The response data that is returned contains:
     - Intents
     - Entities
     - Output
     - Context
   - *Intents, entities, output*, and *context* are set from the Watson Assistant dialog nodes.
   - The *output* contains the Watson Assistant dialog response to the user.
   - The *context* checks whether it contains an `action` object:
     - If there is an action object: Process the action to either get the response from the back-end service or from a Discovery query to the FAQ documents collection.
     - If there is no action object: The *output* is returned to the user “as is”.

Designing and building an intelligent solution

Figure 15-24. Building the cognitive banking chatbot application (cont.)

7. To receive and process the response from the Watson Assistant service:
   - After the message API finishes, it returns the response to the callback function to be processed.
   - The response that is returned contains the intents, entities, output, and the context that were set from the Watson Assistant dialog nodes.
   - The output contains the response of the Watson Assistant dialog.
   - Context is checked from the Node.js application to see whether it contains the “action” object:
     - If there is an action object: Process the action to either get the response from the back-end service or from a Discovery query to the FAQ documents collection.
     - If no action is found: The output of Watson Assistant dialog is returned to the user as is.

IBM Training

Build cognitive banking chatbot application (cont.)

8. Check whether the user is angry by using Tone Analyzer:
   - Use the service's `tone` API.
   - Set the user's input in the payload body.
   - The response data that is returned contains the score of the user's tone. If an angry tone is detected, the angry score is added to the Watson Assistant context.
9. Detect location in the user's questions by using Natural Language Understanding:
   - Use the service's `analyze` API.
   - Set the user's input in the payload body.
   - Select the entities and keywords as the features in the payload body.
   - The response data that is returned contains the entity type `Location` if location is present in the user's input. If the Location entity is detected, the Location is added to the Watson Assistant context.

Designing and building an intelligent solution

Figure 15-25. Build cognitive banking chatbot application (cont.)

8. The Node.js application checks for the user's tone by using the Tone Analyzer service as follows:
   - It uses the *tone* API of the service.
   - The user's input is set in the payload of the API, and the callback function is passed in its arguments.
   - The API returns the response data in the callback function, which contains the user's tone scores. If the user is angry, the API returns an angry tone type and score. The application sets the angry score in the context to be sent to the Watson Assistant service afterward.

9. The Node.js application detects the location in the user's question by using the Natural Language Understanding service as follows:
   - It uses the *analyze* API of the service.
   - The user's input is set in the payload of the API.
   - The *features* object, which contains the entities and keywords, is set in the payload.
   - The callback function is also passed in its arguments to process the response of the API.
   - The API returns the response data in the callback function, which contains the entities and keywords that were detected. If a location is detected in the user's input, the application sets the Location in the context to be sent to the Watson Assistant service afterward.

IBM Training

Building the cognitive banking chatbot application (Cont.)

10. Generate the chatbot response by using Discovery:
    - If the Watson Assistant context contains `action` to get a response from Discovery:
      - Use the service's `query` API.
      - Set the user's input in the payload.
      - The response data that is returned contains the response to be returned to the user.

Designing and building an intelligent solution

Figure 15-26. Building the cognitive banking chatbot application (Cont.)

10. The Node.js application sends a response to the user from the Discovery service FAQ documents as follows:
    - If the Watson Assistant dialog sets an "action" context with the value that indicates that the application must generate the response from Discovery:
      - The application uses the query API of the Discovery service.
      - The user's input is set in the payload of the API, and the callback function is passed in its arguments to process the returned response.
      - The response data that is returned contains the response that must be returned to the user.

15.4. Knowledge Studio and Watson Discovery integration flow

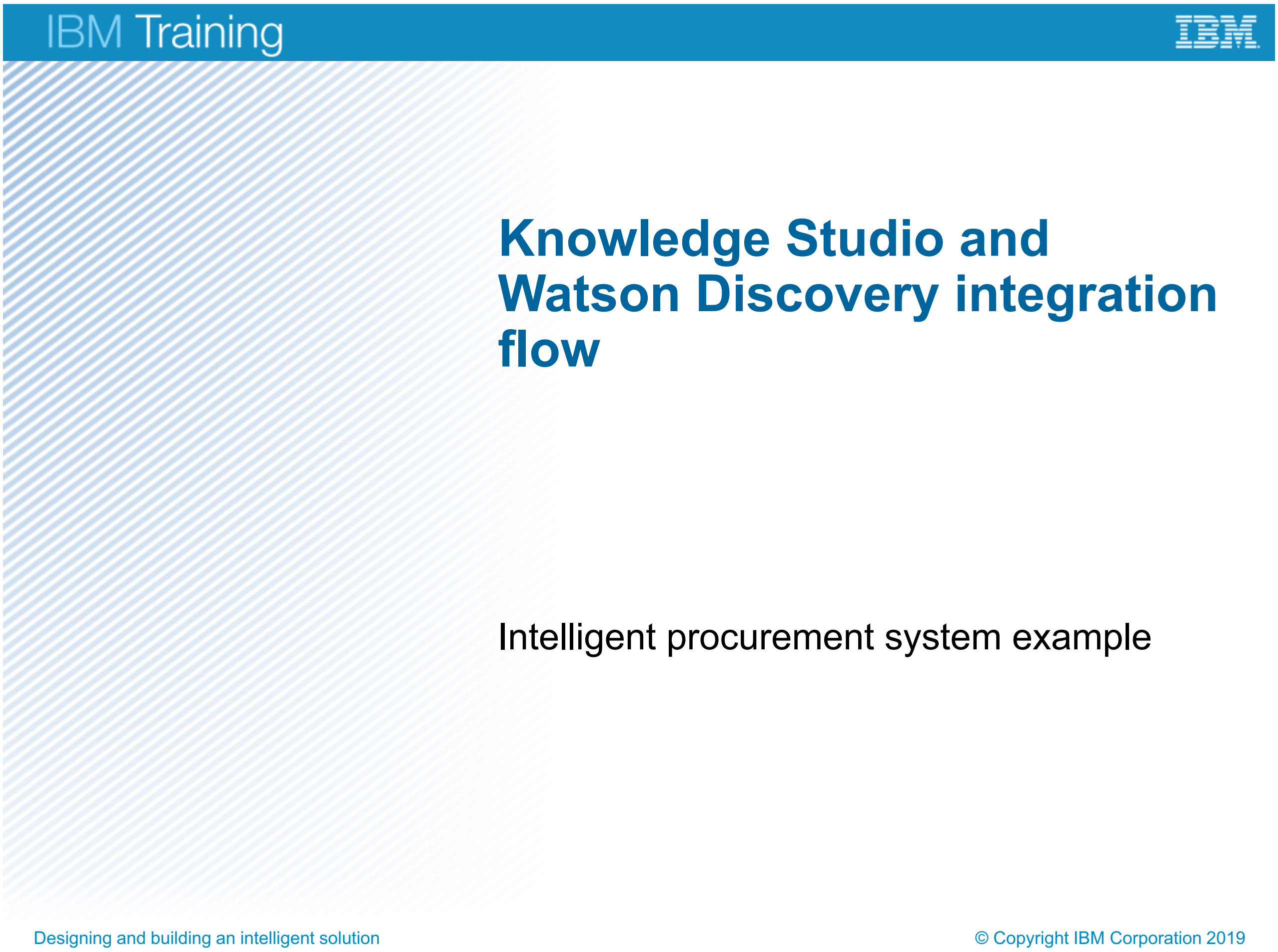


Figure 15-27. Knowledge Studio and Watson Discovery integration flow

IBM Training

Topics

- Integrating IBM Watson services to build a solution
- Watson Assistant integration with other services
- Watson Assistant, Discovery, Natural Language Understanding, and Tone Analyzer integration flow
- Knowledge Studio and Watson Discovery integration flow

Designing and building an intelligent solution

Figure 15-28. Topics

IBM Training

Intelligent procurement system

- The intelligent procurement system demonstrates how to use Watson Knowledge Studio and Watson Discovery to enhance the analysis of shipping and procurement information.
- Instead of hiring experts to analyze the reports that are captured from data sources, this solution provides expert analysis that is based on Watson Knowledge Studio and Discovery.
- This unit describes the integration flow of the solution.
  - Exercise 12 shows the step-by-step implementation.

Designing and building an intelligent solution

Figure 15-29. Intelligent procurement system

Currently, customers perform analyses of various market reports on their own or hire experts to make procurement decisions. These experts analyze reports that are captured from data sources, a process that can be time-consuming and prone to human error, which potentially might cause a chain effect of issues that can impact production. This code pattern explains how to create a complete end-to-end solution for a procurement use case. With this intelligent procurement system, which is based on Watson Knowledge Studio (WKS) and Watson Discovery, a customer can receive expert analysis more quickly and accurately.

Exercise 12 shows the full implementation steps for the “Intelligent procurement system” solution.

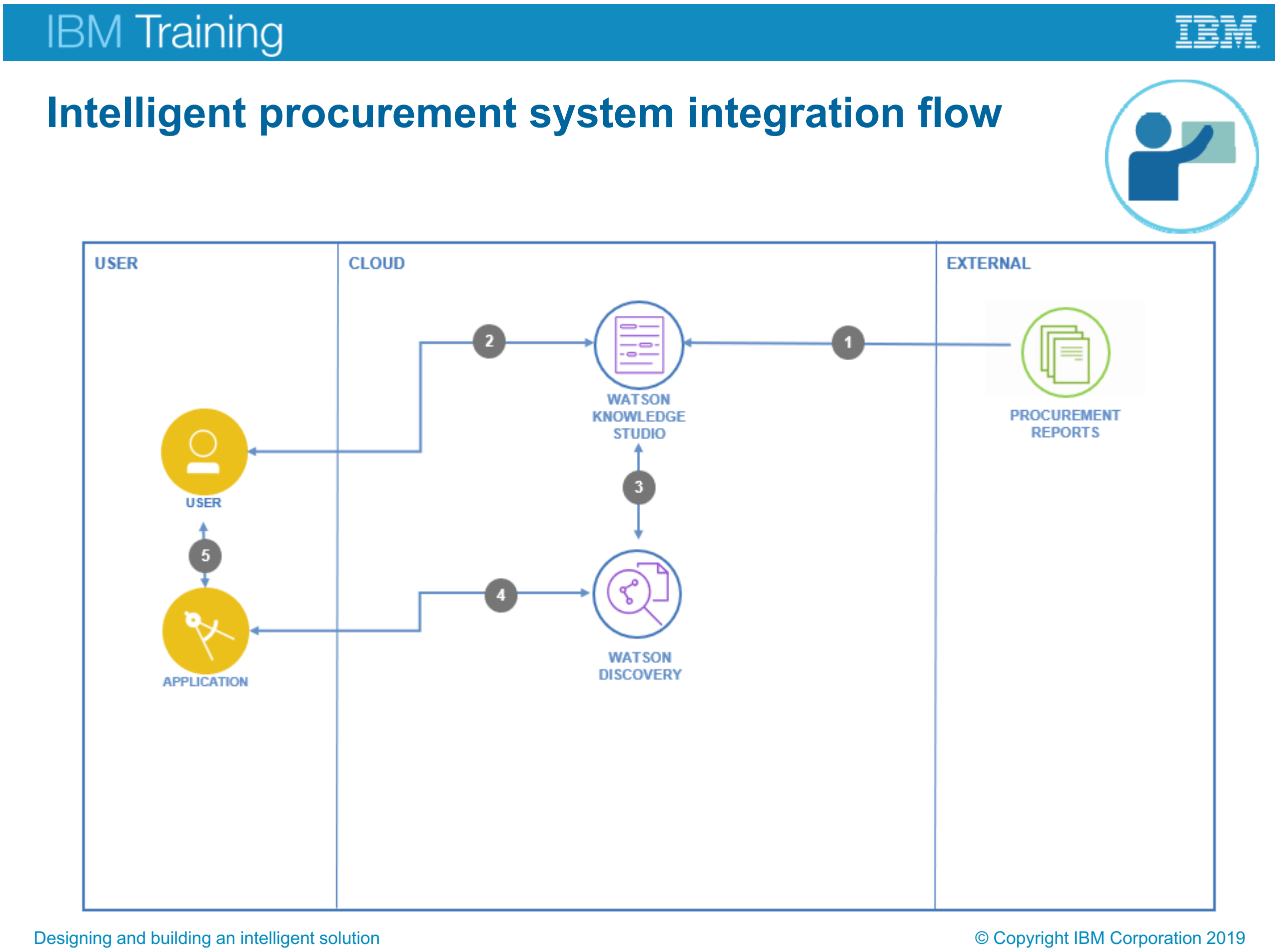


Figure 15-30. Intelligent procurement system integration flow

Intelligent procurement system integration flow:

1. Load the type system and corpus files into Watson Knowledge Studio.
2. A user generates a model by training and evaluating data.
3. The Knowledge Studio model is deployed to Watson Discovery.
4. An application queries Watson Discovery for procurement data.
5. A user uses the app to select a query to perform, which retrieves the data from Watson Discovery.

IBM Training

Unit summary

- Explain the need to integrate multiple IBM Watson services to build an intelligent solution.
- Describe the general outline for the integration of IBM Watson Assistant with other services and applications.
- Explain the key concepts that enable Watson Assistant integration.
- Describe the integration flow between IBM Watson Assistant, IBM Watson Discovery, IBM Watson Natural Language Understanding, and IBM Watson Tone Analyzer to build the cognitive banking chatbot.
- Describe the integration flow between IBM Watson Knowledge Studio and Watson Discovery.

Designing and building an intelligent solution

Figure 15-31. Unit summary

IBM Training

Review questions

1. True or False:
   It is better to integrate just one Watson service to add cognition to the application.

2. Which service can be integrated with the Watson Assistant to detect the user’s anger?
   A. Natural Language Classifier
   B. Tone Analyzer
   C. Watson Discovery
   D. Anger Detection

3. True or False:
   Integrating a custom model created for a particular domain by using Knowledge Studio with Discovery service enables organizations to get expert analysis of their document collections.

Designing and building an intelligent solution

Figure 15-32. Review questions

IBM Training

Review questions (cont.)

4. Which Watson service is used to analyze and search data in human-readable documents?
   A. Watson Knowledge Studio
   B. Natural Language Understanding
   C. Watson Discovery
   D. Language Translator

5. Which item maintains the conversation state information?
   A. Callback function
   B. Entity
   C. Context
   D. REST API

Designing and building an intelligent solution

Figure 15-33. Review questions (cont.)

IBM Training

Review answers

1. True or <u>False</u>: It is better to integrate just one Watson service to add cognition to the application.

2. Which service can be integrated with the Watson Assistant to detect the user's anger?
   A. Natural Language Classifier
   B. <u>Tone Analyzer</u>
   C. Watson Discovery
   D. Anger Detection

3. <u>True</u> or False: Integrating a custom model created for a particular domain by using Knowledge Studio with Discovery service enables organizations to get expert analysis of their document collections.

Designing and building an intelligent solution

Figure 15-34. Review answers

IBM Training

Review answers (cont.)

4. Which Watson service is used to analyze and search data inside human-readable documents?
   A. Watson Knowledge Studio
   B. Natural Language Understanding
   C. Watson Discovery
   D. Language Translator
   The answer is C.

5. Which item maintains the conversation state information?
   A. Callback function
   B. Entity
   C. Context
   D. REST API

Designing and building an intelligent solution

Figure 15-35. Review answers (cont.)

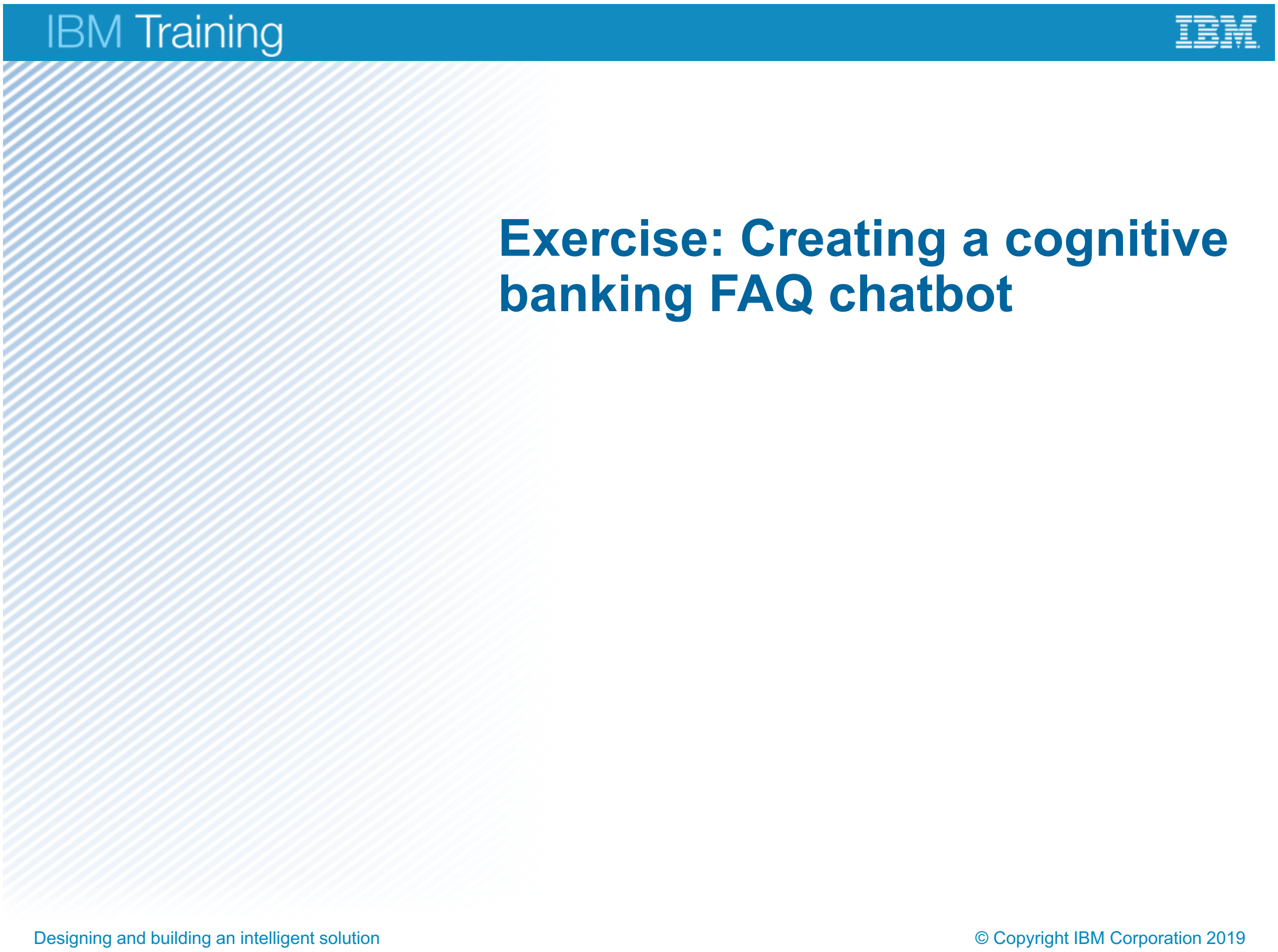


Figure 15-36. Exercise: Creating a cognitive banking FAQ chatbot

IBM Training

Exercise objectives

- This exercise introduces you to the cognitive banking FAQ chatbot application.
- In this application the Watson Assistant flow is enhanced by adding capabilities provided by other Watson APIs such as Watson Natural Language Understanding, Tone Analyzer, and Discovery.
- The objective is to design and implement a solution that integrates several Watson services by using the IBM Watson Node.js SDK.
- After completing this exercise, you should be able to:
  - Create a chatbot by using Watson Assistant and Node.js.
  - Use Watson Discovery with passage retrieval to find answers in FAQ documents.
  - Use Watson Tone Analyzer to detect emotion in a conversation.
  - Identify entities by using Watson Natural Language Understanding.

Designing and building an intelligent solution

Figure 15-37. Exercise objectives

After completing this exercise, you should be able to:

- Create a chatbot by using Watson Assistant and Node.js.
- Use Watson Discovery with passage retrieval to find answers in FAQ documents.
- Use Watson Tone Analyzer to detect emotion in a conversation.
- Identify entities by using Watson Natural Language Understanding.

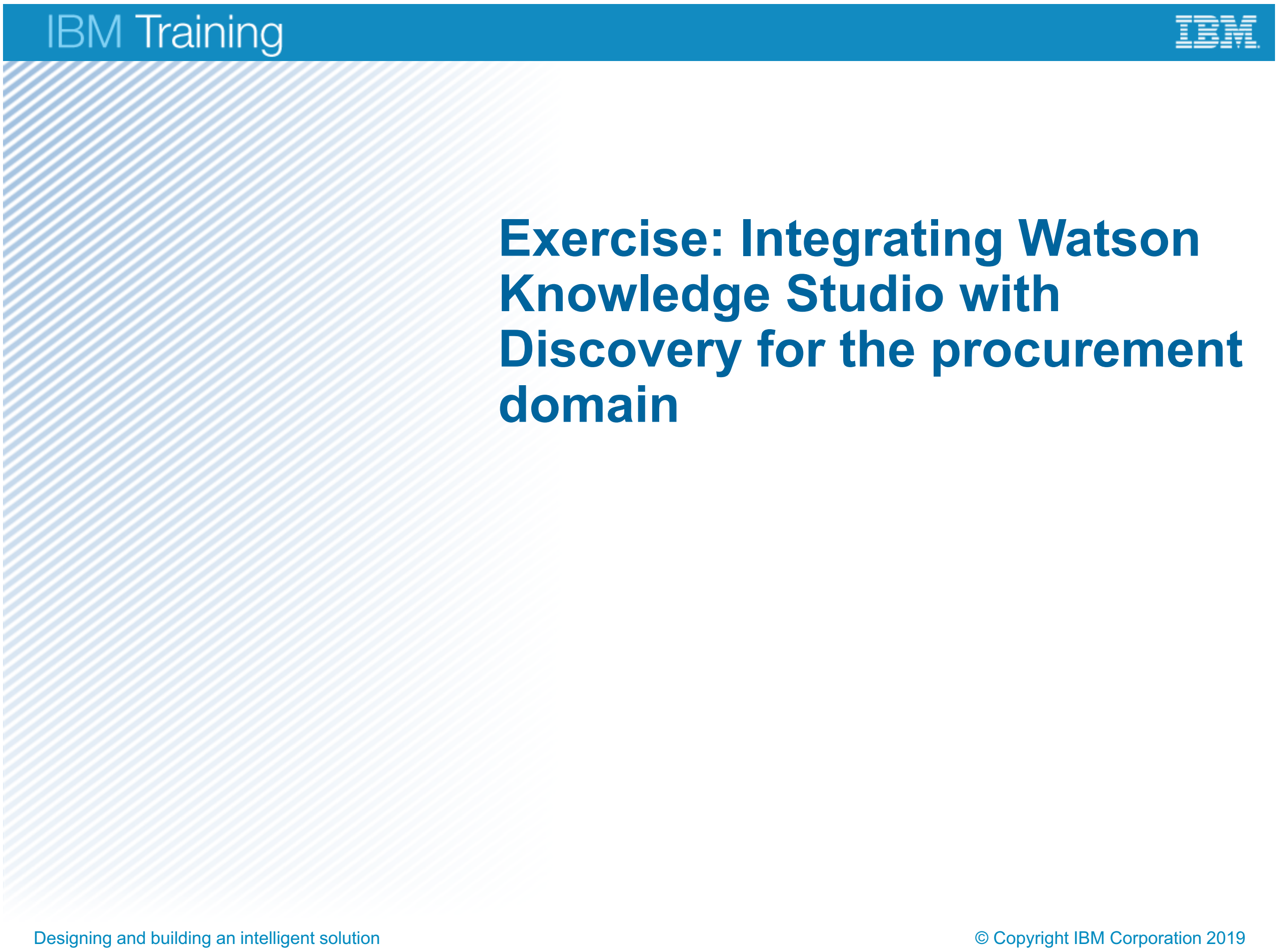


Figure 15-38. Exercise: Integrating Watson Knowledge Studio with Discovery for the procurement domain

IBM Training IBM

Exercise objectives

- This exercise shows you how to integrate a machine learning model from IBM Watson Knowledge Studio with the Discovery service to provide custom entity and relations enrichments for the procurement domain.
- After completing this exercise, you should be able to:
  - Create a machine learning model in Watson Knowledge Studio and deploy it to Watson Discovery.
  - Create a Watson Discovery custom configuration and leverage a Watson Knowledge Studio model to enrich entities and relations.
  - Integrate a custom model from Watson Knowledge Studio with the Discovery service to provide custom entity and relations enrichments customized for a specific procurement domain.

Designing and building an intelligent solution © Copyright IBM Corporation 2019

Figure 15-39. Exercise objectives

After completing this exercise, you should be able to:

- Create a machine learning model in Watson Knowledge Studio and deploy it to Watson Discovery.
- Create a Watson Discovery custom configuration and leverage a Watson Knowledge Studio model to enrich entities and relations.
- Integrate a custom model from Watson Knowledge Studio with the Discovery service to provide custom entity and relations enrichments customized for a specific procurement domain.

IBM Training