



IBM Watson Professional Certification Program

Study Guide Series

Exam C7020-230 - IBM Watson V3
Application Development

| | |
|--|-----------|
| Purpose of Exam Objectives | 3 |
| High-level Exam Objectives..... | 4 |
| Detailed Exam Objectives | 6 |
| Section 1 - Fundamentals of Cognitive Computing | 6 |
| Section 2 - Use Cases of Cognitive Services | 11 |
| Section 3 - Fundamentals of IBM Watson Developer | 13 |
| Section 4 - Developing Cognitive applications using Watson Developer Cloud Services | 18 |
| Section 5 - Administration& DevOps for applications using IBM Watson Developer Cloud Services | 23 |
| Next Steps..... | 24 |

Purpose of Exam Objectives

When an exam is being developed, the Subject Matter Experts work together to define the role the certified individual will fill. They define all of the tasks and knowledge that an individual would need to have in order to successfully implement the product. This creates the foundation for the objectives and measurement criteria, which are the basis for the certification exam.

The Watson Developer Certification item writers used these objectives to develop the questions that they wrote and which will appear on the exam.

It is recommended that you review these objectives. Do you know how to complete the task in the objective? Do you know why that task needs to be done? Do you know what will happen if you do it incorrectly? If you are not familiar with a task, then go through the objective and perform that task in your own environment. Read more information on the task. If there is an objective on a task there is about a 95% chance that you WILL see a question about it on the actual exam.

After you have reviewed the objectives and completed your own research, then take the assessment exam. While the assessment exam will not tell you which question you answered incorrectly, it will tell you how you did by section. This will give you a good indication as to whether you are ready to take the actual exam or if you need to further review the materials.

Note: This is the high-level list of objectives. As you review these objectives, click for a more detailed level of how to perform the task.

High-level Exam Objectives

| Section 1 - Fundamentals of Cognitive Computing | |
|---|--|
| 1.1 | Define the main characteristics of a cognitive system. |
| 1.2 | Explain neural nets. |
| 1.3 | Explain machine learning technologies (supervised, unsupervised, reinforcement learning approaches). |
| 1.4 | Define a common set of use cases for cognitive systems. |
| 1.5 | Define Precision, Recall, and Accuracy |
| 1.6 | Explain the importance of separating training, validation and test data. |
| 1.7 | Measure accuracy of service. |
| 1.8 | Perform Domain Adaption using Watson Knowledge Studio (WKS). |
| 1.9 | Define Intents and Classes. |
| 1.10 | Define the difference between the user question and the user intent. |
| Section 2 - Use Cases of Cognitive Services | |
| 2.1 | Select appropriate combination of cognitive technologies based on use-case and data format. |
| 2.2 | Explain the uses of the Watson services in the Application Starter Kits. |
| 2.3 | Describe the Watson Conversational Agent. |
| 2.4 | Explain use cases for integrating external systems (such as Twitter, Weather API). |
| 2.5 | Describe the IBM Watson Discovery Service |
| Section 3 - Fundamentals of IBM Watson Developer | |
| 3.1 | Distinguish cognitive services on WDC for which training is required or not. |
| 3.2 | Provide examples of text classification using the NLC. |
| 3.3 | Explain the Watson SDKs available as part of the services on Watson Developer Cloud. |
| 3.4 | Explain the Watson REST APIs available as part of the services on Watson Developer Cloud. |
| 3.5 | Explain and configure Natural Language Classification. |
| 3.6 | Explain and configure Visual recognition. |
| 3.7 | Explain how Personality Insights service works. |
| 3.8 | Explain how Tone Analyzer service works. |
| 3.9 | Explain and execute IBM Watson Natural Language Understanding services. |
| 3.10 | Explain, setup, configure and query the IBM Watson Discovery service. |
| 3.11 | Explain and configure the IBM Watson Conversation service |
| Section 4 - Developing Cognitive applications using Watson Developer Cloud Services | |
| 4.1 | Call a Watson API to analyze content. |

| | |
|--|--|
| 4.2 | <u>Describe the tasks required to implement the Conversational Agent / Digital Bot.</u> |
| 4.3 | <u>Transform service outputs for consumption by other services.</u> |
| 4.4 | <u>Define common design patterns for composing multiple Watson services together (across APIs).</u> |
| 4.5 | <u>Design and execute a use case driven service choreography (within an API).</u> |
| 4.6 | <u>Explain the process to provision and use an instance of an IBM Watson Bluemix service instance.</u> |
| 4.7 | <u>Explain the advantages of using IBM Bluemix as the cloud platform for Cognitive application development and deployment.</u> |
| Section 5 - Administration& DevOps for applications using IBM Watson Developer Cloud Services | |
| 5.1 | <u>Describe the process of obtaining credentials for Watson services.</u> |
| 5.2 | <u>Examine application logs provided on IBM Bluemix.</u> |

Detailed Exam Objectives

Section 1 - Fundamentals of Cognitive Computing

1.1. Define the main characteristics of a cognitive system.

SUBTASK(S):

- 1.1.1. Cognitive systems understand, reason and learn
 - 1.1.1.1. Must understand structured and unstructured data
 - 1.1.1.2. Must reason by prioritizing recommendations and ability to form hypothesis
 - 1.1.1.3. Learns iteratively by repeated training as it build smarter patterns
- 1.1.2. Cognitive systems are here to augment human knowledge not replace it
- 1.1.3. Cognitive systems employ machine learning technologies
- 1.1.4. Cognitive systems use natural language processing

Reference: <http://www.redbooks.ibm.com/redpieces/abstracts/sq248387.html?Open>

1.2. Explain neural nets.

SUBTASK(S):

Neural Nets mimic how neurons in the brain communicate

- 1.2.1. Explain the role of synapse and neuron
- 1.2.2. Understand weights and bias
- 1.2.3. List various approaches to neural nets
- 1.2.4. Explain forward and backward propagation
- 1.2.5. Explain gradient descent

Reference:

[Types of artificial neural networks](#)
https://en.wikipedia.org/wiki/Types_of_artificial_neural_networks
[Neural Networks Demystified - Part 1: Data and Architecture](#)
[Types of artificial neural networks](#)
[Neural Networks Demystified Part 2: Forward Propagation](#)
[Neural Networks Demystified - Part 3: Gradient Descent](#)
[Neural Networks Demystified - Part 4: Backpropagation](#)

1.3. Explain machine learning technologies (supervised, unsupervised, reinforcement learning approaches).

SUBTASK(S):

Describe some of the main machine learning concepts:

- 1.3.1. Supervised learning:
 - 1.3.1.1. Classification
 - 1.3.1.2. Regression/Prediction

- 1.3.1.3. Semi-supervised learning
- 1.3.2. Unsupervised learning:
 - 1.3.2.1. Artificial neural network
 - 1.3.2.2. Association rule learning
 - 1.3.2.3. Hierarchical clustering
 - 1.3.2.4. Cluster analysis
 - 1.3.2.5. Outlier Detection
 - 1.3.2.6. Reinforcement learning

Reference:

<http://stats.stackexchange.com/questions/144154/supervised-learning-unsupervised-learning-and-reinforcement-learning-workflow>
https://en.wikipedia.org/wiki/List_of_machine_learning_concepts
<http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>
<http://www.nnwj.de/supervised-unsupervised.html>

1.4. Define a common set of use cases for cognitive systems.

SUBTASK(S):

Customer Call Centers

1.4.1. **Agent Assist: email based Q&A**

- 1.4.1.1. **Problem Solved:** Provides a natural language help system for call agents to rapidly retrieve answers to customer questions
- 1.4.1.2. **Capabilities needed for this use case:** Conversation, natural language answer retrieval, keyword extraction, and entity extraction
- 1.4.1.3. **Services used:** Natural Language Classifier, Watson Discovery
- 1.4.1.4. **Benefits:**
 - 1.4.1.4.1. Detect the topic of a ticket and route to the appropriate department to handle it (Example: room service, maintenance, housekeeping)
 - 1.4.1.4.2. Escalate support tickets based on customer sentiment
 - 1.4.1.4.3. Route support requests to agents that already solved similar problems by detecting natural language similarities between new customer tickets and resolved ones.
 - 1.4.1.4.4. Automation: Customer/Technical Support Tickets Routing
- 1.4.1.5. **Watson Services used:** natural language (text) classification, keyword extraction, entity extraction, and sentiment/tone analysis

1.4.2. **Physicians**

- 1.4.2.1. Expert Advisor:
 - 1.4.2.1.1. **Example:** Watson Discovery Advisor

1.4.2.1.2. **Problem Solved:** Provides relevant medical suggestions and insights in natural language so physicians can more accurately diagnose patients.

1.4.2.1.3. **Services used:** Conversation, Natural Language Classifier, Watson Discovery

1.4.3. **Social MediaData Insights:**

1.4.3.1. **Partner:** Ground Signal

1.4.3.2. **Problem Solved:** Extract useful insights from social media such as Instagram and Twitter by determining the content of photos and topics/sentiment of user posts.

1.4.3.3. **Services used:** Natural Language Classifier, Watson Discovery, Visual Recognition

1.5. Define Precision, Recall, and Accuracy

SUBTASK(S):

1.5.1. **Precision:** Precision is the percentage of documents labelled as positive that are actually positive.

Formula: $\text{True Positives} / (\text{True Positives} + \text{False Positives})$

1.5.2. **Recall:** Recall is the percent of documents labelled as positive were successfully retrieved.

Formula: $\text{True Positives} / (\text{True Positives} + \text{False Negatives})$

1.5.3. **Accuracy:** Accuracy is the fraction of documents relevant to a query that were successfully retrieved.

Formula: $(\text{True Positives} + \text{True Negatives}) / \text{Total Document Count}$

References:

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

<https://www.quora.com/What-is-the-best-way-to-understand-the-terms-precision-and-recall>

1.6. Explain the importance of separating training, validation and test data.

SUBTASK(S):

1.6.1. One school of thought: partition the data into the above three

1.6.1.1. 50/25/25

1.6.1.2. 60/20/20

1.6.1.3. 50/50

1.6.2. Another: Using the Bootstrap method vs Cross Validation

- 1.6.2.1. the bootstrap method allows us to simulate the process of obtaining new data sets, so that we can estimate the error/variability of our estimate without generating additional samples.
- 1.6.3. Training Process:
 - 1.6.3.1. Data = Training Data + Cross-Validation Data + Test Data

<https://www.quora.com/What-is-the-difference-between-bootstrapping-and-cross-validation>

<https://developer.ibm.com/dwblog/2016/chatbot-cognitive-performance-metrics-accuracy-precision-recall-confusion-matrix/#respond>

<https://www.quora.com/What-is-a-training-data-set-test-data-set-in-machine-learning-What-are-the-rules-for-selecting-them>

1.7. Measure accuracy of service.

SUBTASK(S):

- 1.7.1. Define accuracy of a system (True positive + true negative)/(TP + TN + False P + False N)
- 1.7.2. Explain factors that affect accuracy in supervised learning
 - 1.7.2.1. Sample size of training set data, dangers of over fitting, curated content
- 1.7.3. Explain factors that affect accuracy of unsupervised learning
 - 1.7.3.1. Sample size, curse of dimensionality, over/under fitting
- 1.7.4. Running a Blind set test
- 1.7.5. Importance of iterative training using feedback that has diminished costs derivative

Reference:

<https://developer.ibm.com/dwblog/2016/chatbot-cognitive-performance-metrics-accuracy-precision-recall-confusion-matrix/#respond>

https://www.ibm.com/developerworks/community/blogs/jfp/entry/Overfitting_In_Machine_Learning?lang=en

1.8. Perform Domain Adaption using Watson Knowledge Studio (WKS).

SUBTASK(S):

- 1.8.1. Analyze the domain
- 1.8.2. Create a project
- 1.8.3. Develop the type system
- 1.8.4. Understand "mention", "entity", and "relation"
- 1.8.5. Define entity types, relationships and co-references
- 1.8.6. Define and load dictionaries for pre-annotation
- 1.8.7. Understand Human Annotation
- 1.8.8. Understand SIRE (machine learning information extraction)

Reference:

https://www.ibm.com/watson/developercloud/doc/wks/wks_projconfig.shtml

1.9. Define Intents and Classes.

SUBTASK(S):

- 1.9.1. The Natural Language Classifier service available via WDC, enables clustering or classification based on some measure of inherent similarity or distance given the input data. Such clustering is known as intents or classes.
- 1.9.2. Where classes may include images, intent is a similar clustering for written utterances in unstructured natural language format.

Reference:

<https://developer.ibm.com/watson/blog/2015/09/28/combining-natural-language-classifier-and-dialog-to-create-engaging-applications-2/?shared=email&msg=fail>
<https://developer.ibm.com/answers/topics/watson-conversation/>
https://en.wikipedia.org/wiki/Statistical_classification

1.10. Define the difference between the user question and the user intent.

SUBTASK(S):

- 1.10.1. To answer correctly, we need to understand the intent behind the question, in order to first classify it then take action on it (e.g., with a Dialog API)
- 1.10.2. The user question is the verbatim question
- 1.10.3. The user intent maps the user question to a known classification
- 1.10.4. This is a form of classifying question based on search goals
- 1.10.5. Intents are the superset of all actions your users may want your cognitive system to undertake. Put another way, questions are a subset of user intents. Questions usually end in "?", but sometimes we need to extract the user intent from the underlying context. Common examples of user intents:
 - Automation: "Schedule a meeting with Sue at 5pm next Tuesday."
 - Declarative: "I need to change my password."
 - Imperative: "Show me the directions to my the nearest gas station."

Reference:

<https://developer.ibm.com/answers/topics/watson-conversation/>
<https://www.ibm.com/watson/developercloud/doc/conversation/index.html>

Section 2 - Use Cases of Cognitive Services

2.1. Select appropriate combination of cognitive technologies based on use-case and data format.

SUBTASK(S):

- 2.1.1. Agent-assist for email-based customer call center
 - 2.1.1.1. **Data:** customer emails
 - 2.1.1.2. **Features needed:** Q&A, Text classification, entity extraction and, keyword extraction
 - 2.1.1.3. **Watson-specific Services addressing the requirements:** Natural Language Classifier, Watson Discovery, Natural Language Understanding
- 2.1.2. Agent-assist for phone-based customer call center
 - 2.1.2.1. **Data:** customer voice recordings
 - 2.1.2.2. **Features needed:** Q&A, Speech recognition, text-to-speech, text classification, entity extraction, keyword extraction
 - 2.1.2.3. **Watson-specific Services addressing the requirements:** Watson Conversation, Watson Discovery, Natural Language Understanding, Watson Text to Speech, Watson Speech to Text
- 2.1.3. Expert advisor use case for physicians
 - 2.1.3.1. **Data:** natural language intents
 - 2.1.3.2. **Features needed:** Q&A, Text classification, entity extraction and keyword extraction
 - 2.1.3.3. **Watson-specific Services addressing the requirements:** Natural Language Classifier, Watson Conversation, Watson Discovery, Natural Language Understanding
- 2.1.4. Data insights for Instagram images
 - 2.1.4.1. **Data:** images
 - 2.1.4.2. **Features needed:** Image classification and natural OCR
 - 2.1.4.3. **Watson-specific:** Visual Recognition
- 2.1.5. Data insights for Twitter
 - 2.1.5.1. **Data:** tweets
 - 2.1.5.2. **Features needed:** Text classification, entity extraction, keyword extraction, personality profile
 - 2.1.5.3. **Watson-specific Services addressing the requirements:** Natural Language Classifier, Natural Language Understanding, Personality Insights and Watson Discovery

2.2. Explain the uses of the Watson services in the Application Starter Kits.

SUBTASK(S):

Developers can use Starter Kits to get jump start building the cognitive application. Starter Kits includes code examples with combination of multiple services for common use cases. Students should understand the functionality of starter kits and underlying services used.

Reference:

<https://www.ibm.com/watson/developercloud/starter-kits.html>

2.3. Describe the Watson Conversational Agent.

SUBTASK(S):

2.3.1. Connie is the concierge at Hilton

2.3.2. Pepper is the helper Robot in Tokyo by Softbank

2.4. Explain use cases for integrating external systems (such as Twitter, Weather API).

SUBTASK(S):

[Explain the components of systems communicating with one another](#)

[Use case of Twitter and sentiment analysis](#)

[Use case of the Weather APIs and mission critical decision that are impacted by weather](#)

2.5. Describe the IBM Watson Discovery Service

SUBTASK(S):

2.5.1. List the functionalities that the Watson Discovery service provides

2.5.2. Explain the components of the Discovery service

2.5.3. Use the Discovery service via APIs or the Discovery Tooling

2.5.3.1. Explain how to setup the environment

2.5.3.2. Explain how to configure the Discovery service

2.5.3.3. Explain how to add content

2.5.3.4. Explain how to build queries

Reference:

<https://www.ibm.com/watson/developercloud/doc/discovery/index.html>

https://www.youtube.com/watch?v=FikHwoJ6_FE

<https://www.youtube.com/watch?v=fmIPeopG-ys&t=1s>

Section 3 - Fundamentals of IBM Watson Developer

3.1. Distinguish cognitive services on WDC for which training is required or not.

SUBTASK(S):

3.1.1. Some IBM Watson services work out-of-the-box as they were pre-trained in a specific domain (domain-adapted). Other Watson services require training. For pre-trained services, it's critical to know the adapted domains as they indicate the areas in which the service will perform best.

Pre-trained Watson services:

- 3.1.2.1. Watson Text-to-Speech
- 3.1.2.2. Watson Speech-to-text
- 3.1.2.3. Language Translator (conversational, news, and patent domains)
- 3.1.2.4. Natural Language Understanding
- 3.1.2.5. Tone Analyzer
- 3.1.2.6. Personality Insights (social media domain)
- 3.1.2.7. Watson Discovery News
- 3.1.2.8. Watson Discovery Service
- 3.1.3. Services requiring training:
 - 3.1.3.1. Natural Language Classifier
 - 3.1.3.2. Visual recognition (custom models)
 - 3.1.3.3. Watson Conversation

Reference:

<https://www.ibm.com/watson/products-services/>

3.2. Provide examples of text classification using the NLC.

SUBTASK(S):

- 3.2.1. Sentiment analysis
- 3.2.2. Spam email detection
- 3.2.3. Customer message routing
- 3.2.4. Academic paper classification into technical fields of interest
- 3.2.5. Forum post classification to determine correct posting category
- 3.2.6. Patient reports for escalation and routing based on symptoms
- 3.2.7. News article analysis
- 3.2.8. Investment opportunity ranking
- 3.2.9. Web page topic analysis

Reference:

<https://console.bluemix.net/docs/services/natural-language-classifier/natural-language-classifier-overview.html#about>

3.3. Explain the Watson SDKs available as part of the services on Watson Developer Cloud.

SUBTASK(S):

- 3.3.1. Identify the programming languages with SDKs available
- 3.3.2. Describe the advantage and disadvantages of using an SDK
- 3.3.3. Find the Watson SDKs and other resources on the WDC GitHub

Reference:

https://github.com/watson-developer-cloud?cm_mc_uid=&cm_mc_sid_50200000=1498205956&cm_mc_sid_52640000=

3.4. Explain the Watson REST APIs available as part of the services on Watson Developer Cloud.

SUBTASK(S):

- 3.4.1. Identify the Language services on WDC
- 3.4.2. Identify the Vision services on WDC
- 3.4.3. Identify the Speech services on WDC
- 3.4.4. Identify the Data Insights services on WDC

Reference:

<https://www.ibm.com/watson/developercloud/>

3.5. Explain and configure Natural Language Classification.

SUBTASK(S):

- 3.5.1. The service enables developers without a background in machine learning or statistical algorithms to interpret the intent behind text.
- 3.5.2. Configure:
 - 3.5.2.1. Gather sample text from real end users (fake initially if you have to but not much)
 - 3.5.2.2. Determine the users intents that capture the actions/needs expressed in the text
 - 3.5.2.3. Classify your user text into these user intents
 - 3.5.2.4. Separate your user text into train/test datasets
 - 3.5.2.5. Train an NLC classifier on your training dataset
 - 3.5.2.6. Pass the user input to an NLC classifier
 - 3.5.2.7. Determine the accuracy, precision, and recall of the NLC classifier using your test dataset
 - 3.5.2.8. Improve the confidence level iteratively through back propagation or other means.

Reference:

<https://www.ibm.com/watson/developercloud/natural-language-classifier/api/v1/>

3.6. Explain and configure Visual recognition.

SUBTASK(S):

- 3.6.1. Describe the process for training a classifier
- 3.6.2. Explain how to identify images with a specified classifier
- 3.6.3. Describe the capabilities of facial, gender, and age recognition
- 3.6.4. Describe the capabilities of Natural Scene OCR
- 3.6.5. Explain how collections are built and the use of similarity search

Reference:

<https://www.ibm.com/watson/developercloud/doc/personality-insights/user-overview.html>

3.7. Explain how Personality Insights service works.

SUBTASK(S):

- 3.7.1. Describe the intended use of the Personality Insights service
- 3.7.2. Describe the inputs and outputs of the Personality Insights service
- 3.7.3. Describe the personality models of the Personality Insights service
- 3.7.4. References:

<https://www.ibm.com/watson/developercloud/doc/personality-insights/user-overview.html>

3.8. Explain how Tone Analyzer service works.

SUBTASK(S):

- 3.8.1. Describe the common use cases of the Tone Analyzer service
- 3.8.2. Describe the basic flow of the Tone Analyzer service
- 3.8.3. Explain the three categories of tone scores and their sub-tones: emotional tone, social tone, and language tone.
- 3.8.4. Explain how Tone Analyzer service is different from the Natural Language Understanding - Sentiment Analysis and Emotion Insights service

References:

<https://www.ibm.com/watson/developercloud/tone-analyzer/api/v3/>

<https://www.ibm.com/watson/developercloud/doc/tone-analyzer/index.html>

<https://www.ibm.com/blogs/watson/2016/02/293/>

3.9. Explain and execute IBM Watson Natural Language Understanding services.

SUBTASK(S):

- 3.9.1. Identify the capabilities of Natural Language Understanding
- 3.9.2. Describe the text extraction features of Natural Language Understanding
- 3.9.3. Distinguish between keywords, entities, and concepts
- 3.9.4. Distinguish between document-level and targeted sentiment
- 3.9.5. Explain the difference between the taxonomy call and the knowledge graph

- 3.9.6. Explain disambiguation as it relates to entities
- 3.9.7. Explain how Emotion Analysis service works
 - 3.9.7.1. What emotions does Emotion Analysis detect?
 - 3.9.7.2. Describe the main use cases for applying the Emotion Insights service
 - 3.9.7.3. Describe the main types of positive/negative sentiment extracted from digital text
 - 3.9.7.4. Describe the API types provided by the Sentiment Analysis service
 - 3.9.7.5. Describe the differences between sentiment and emotion analyses

Reference:

<https://www.ibm.com/watson/developercloud/natural-language-understanding/api/v1/>

3.10. Explain, setup, configure and query the IBM Watson Discovery service.

SUBTASK(S):

- 3.10.1. How to create the Data Collection repository
 - 3.10.1.1. Explain the significance of working with sample documents
 - 3.10.1.2. Explain the difference between Default and switching to a new custom collection
 - 3.10.1.3. Explain when might you need more than one collection
- 3.10.2. What are some of the steps required when you customize your configuration
 - 3.10.2.1. Identify sample documents
 - 3.10.2.2. Convert sample documents
 - 3.10.2.3. Add enrichments
 - 3.10.2.4. Normalize data
- 3.10.3. What are the four standard document formats and explain the conversion flow (MS Word, PDF, HTML and JSON)
- 3.10.4. Adding Enrichments, explain the following enrichments:
 - 3.10.4.1. Entity Extraction
 - 3.10.4.2. Keyword Extraction
 - 3.10.4.3. Taxonomy Classification
 - 3.10.4.4. Concept Tagging
 - 3.10.4.5. Relation Extraction
 - 3.10.4.6. Sentiment Analysis
 - 3.10.4.7. Emotion Analysis
- 3.10.5. Explain document size limitations
- 3.10.6. Explain the essence of the Normalization step, the last step in customizing your configuration

- 3.10.7. Explain some of the methods or ways of adding content after you are satisfied with the configuration work.
 - 3.10.7.1. Adding content through the API
 - 3.10.7.2. Adding content through the UI
 - 3.10.7.3. Adding content through the data crawler
- 3.10.8. Querying your data
 - 3.10.8.1. Explain the three search query parameters (filter, query, aggregation)
 - 3.10.8.2. Explain the three structure query parameters (count, offset, return)
 - 3.10.8.3. Explain Aggregations

References:

<https://www.ibm.com/watson/developercloud/discovery/api/v1/>
<https://www.ibm.com/watson/developercloud/doc/discovery/index.html><https://www.ibm.com/blogs/watson/2016/12/watson-discovery-service-understand-data-scale-less-effort/>
<https://www.youtube.com/watch?v=fmIPeopG-ys&t=1s>
<https://www.ibm.com/blogs/bluemix/2016/11/watson-discovery-service/>

3.11. Explain and configure the IBM Watson Conversation service

SUBTASK(S):

- 3.11.1. Creating a workspace
- 3.11.2. Define intents (user input)
- 3.11.3. Define entities (relevant term or object)
- 3.11.4. Build a dialog (branching conversation flow)
- 3.11.5. Test the conversation agent / bind to application
- 3.11.6. Know how other Watson services can add value to Conversation service
- 3.11.7. Describe the difference between short tail and long tail conversational exchanges.
- 3.11.8. Explain how Discovery can support long tail conversations in Watson Conversation.

Reference:

<https://www.ibm.com/watson/developercloud/conversation/api/v1/>

Section 4 - Developing Cognitive applications using Watson Developer Cloud Services

4.1. Call a Watson API to analyze content.

SUBTASK(S):

- 4.1.1. Natural Language Understanding - Create an instance of the Natural Language Understanding service in Bluemix
 - 4.1.1.1. Select the correct API to call for text extraction, sentiment analysis, or any of the Natural Language Understanding services.
 - 4.1.1.2. Pass your content to your Alchemy services' endpoint through a RESTful API call
 - 4.1.1.3. Natural Language Classifier
 - 4.1.1.4. Gather sample text from real end users (fake initially if you have to...but not much)
 - 4.1.1.5. Determine the users intents that capture the actions/needs expressed in the text
 - 4.1.1.6. Classify your user text into these user intents
 - 4.1.1.7. Separate your user text into train/test datasets
 - 4.1.1.8. Create an instance of the Natural Language Classifier service in Bluemix
 - 4.1.1.9. Train an NLC classifier on your training dataset
 - 4.1.1.10. Pass your content to your NLC services' endpoint through a RESTful API call
 - 4.1.1.11. Determine the accuracy, precision, and recall of the NLC classifier using your test dataset
- 4.1.2. Personality Insights - Create an instance of the Personality Insights service in Bluemix
 - 4.1.2.1. Gather text from users in their own voice
 - 4.1.2.2. Ensure you meet the minimum limits for word count to limit sampling error.
 - 4.1.2.3. Pass your content to your Personality Insight services' endpoint through a RESTful API call

4.2. Describe the tasks required to implement the Conversational Agent / Digital Bot.

SUBTASK(S):

- 4.2.1. Scope if there is any historical data to draw upon in the creation of the digital agent.
- 4.2.2. Set expectations of what your conversational agent will and will not do.
- 4.2.3. Determine the user intents (What the end-user says to the bot) and entities (what they are talking about) in your conversation. Factor in historical data if applicable.
- 4.2.4. Design and define ideal conversation flow by writing questions and responses
- 4.2.5. Create a service instance of Watson Conversation in Bluemix
- 4.2.6. Launch a workspace through the provisioned service instance

- 4.2.7. Configure service to recognize user intents
- 4.2.8. Configure service to recognize user entities
- 4.2.9. Identify the usage of system entities
- 4.2.10. Map user intents and entities and define responses within dialog nodes
- 4.2.11. Build out the dialog conversation flow for the agent.
- 4.2.12. Determine ways users could diverge from your conversation process flow and ways to redirect them back.
- 4.2.13. Think on measures of conversation control and shaping. Does your agent conversation scope out and address what the end-user needs from the agent?
- 4.2.14. Will the end-user be able to comprehend the vocabulary, terminology, and style of the digital agent?
- 4.2.15. Define a conversation node to have multiple responses.
- 4.2.16. Bind Conversation service instance to an application.
- 4.2.17. Present your beta conversation agent to end users to capture real end-user interaction.
- 4.2.18. Identify areas where your conversation agent misunderstood the user.
- 4.2.19. Identify areas where users strayed outside the domain of your conversation agent through the Improve section of your Conversation workspace.
- 4.2.20. Update your conversation agent with new intents or entities to strengthen interactions.
- 4.2.21. Understand improvements of accuracy by utilizing training sets against themselves (K-Fold Cross Validation) and blind tests for when changes are made with your intents, entities, and dialog.

4.3. Transform service outputs for consumption by other services.

SUBTASK(S):

- 4.3.1. Natural Language Classifier
 - 4.3.1.1. Using classifiers from NLC to drive dialog selections in Dialog
- 4.3.2. Personality Insights
 - 4.3.2.1. Use the service output from two different textual inputs and compare the personalities based on the results
- 4.3.3. Speech to text
 - 4.3.3.1. Use the transcribed output from speed to text as input to language translation
- 4.3.4. Language translation
 - 4.3.4.1. Use the translated text from language translation as input to text to speech
- 4.3.5. Watson Discovery News
 - 4.3.5.1. Use the top article returned by the search from Watson Discovery News as input to Watson Natural Language Understanding-Sentiment Analysis and Watson Tone Analyzer

- 4.3.5.2. Use the top article returned by the search from Watson Discovery News as input to relationship extraction to tell who is trending in the article

4.4. Define common design patterns for composing multiple Watson services together (across APIs).

SUBTASK(S):

Cognitive systems tend to gain more value as additional services are composed. With so many services, it's sometimes hard to tell which services work best together.

4.4.1. Conversation

- 4.4.1.1. Goal: Engage user in back-and-forth dialog while detecting and acting on user intent. The specifics of the actions taken are guided by the entities discovered.

- 4.4.1.2. [Watson Conversation Service](#)

4.4.2. Q&A

- 4.4.3.1. Goal: Answer a wide range of customer questions while offering precise answers for frequently asked facts and highly relevant passages for less frequent questions that may not have a single best answer

- 4.4.3.2. Services: [Watson Conversation Service + Watson Discovery Service](#)

4.4.4. Agent Assist

- 4.4.4.1. Goal: Provide natural language help systems so call agents can rapidly retrieve answers to customer questions

- 4.4.4.2. Services: [Watson Conversation Service + Watson Speech to Text Service + Watson Text to Speech Service + Watson Discovery Service](#)

4.4.5. Automated Customer Support Routing

- 4.4.5.1. Goal: Detect the topic of a ticket and route to the appropriate department to handle it. E.g. room service, maintenance, housekeeping in the case of hotel guest request routing.

- 4.4.5.2. Watson Natural Language Understanding (Services: Keyword extraction and sentiment analysis)

4.4.6. Social Monitoring

- 4.4.6.1. Goal: Monitor all posts with specific keywords (e.g. for a company's followers, sponsors, or critiques) to detect what's being discussed and the sentiment/tone associated to it.

- 4.4.6.2. Services used: Keyword extraction, entity extraction, and sentiment/tone analysis ([Natural Language Understanding](#))

4.4.7. Discovery Insights

- 4.4.7.1. Goal: 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.

- 4.4.7.2. Services Used: ([Watson Discovery Service](#))

4.5. Design and execute a use case driven service choreography (within an API).

SUBTASK(S):

- 4.5.1. Natural Language Classifier
 - 4.5.1.1. Create a classifier
 - 4.5.1.2. Return label information
 - 4.5.1.3. List classifiers
- 4.5.2. Discovery Service
 - 4.5.2.1. Crawl, convert, enrich and normalize data.
 - 4.5.2.2. Securely explore your proprietary content as well as free and licensed public content.
 - 4.5.2.3. Apply additional enrichments such as concepts, relations, and sentiment through natural language processing.
 - 4.5.2.4. Query and analyze your results
- 4.5.3. Language Translator –Use a default language domain or customize a domain
 - 4.5.3.1. Choose the language to translate or allow Watson to auto-detect the language of your input
 - 4.5.3.2. Select the language for the translator to output
 - 4.5.3.3. Enter or paste text into the input to be translated
- 4.5.4. Visual Recognition
 - 4.5.4.1. Gather and prepare the training data for classifiers or collections
 - 4.5.4.2. Train and create new classifier or collection by uploading the training data to the API
 - 4.5.4.3. Classify images or search your collection by uploading image files to search against your collection
 - 4.5.4.4. View the search results of the identified objects, scenes, faces, and text that meet minimum threshold. For collections, view the search returns of similar images to the ones used to search.

4.6. Explain the process to provision and use an instance of an IBM Watson Bluemix service instance.

SUBTASK(S):

- 4.6.1. Specific steps:
 - 4.6.1.1. Setup an account on IBM Bluemix
 - 4.6.1.2. Create an instance of an IBM Watson service in the Bluemix account. An instance can be created using either the Bluemix GUI or the Bluemix command line interface.
 - 4.6.1.3. Obtain credentials from the IBM Watson service from the service instance.
 - 4.6.1.4. These credentials can be obtained programmatically if the application is also hosted on IBM Reference:

- 4.6.1.5. The credentials can also be viewed and recorded using the IBM Bluemix CLI or GUI interface.
 - 4.6.1.6. The application can then invoke the IBM Watson Service using the published REST API for that service. Many developers use the Watson API Explorer to create and view API invocations of the desired service using REST protocol.
- 4.6.2. Overall Getting Started reference:
- 4.6.2.1. “Getting started with Watson and Bluemix”
- Reference
- <https://watson-api-explorer.mybluemix.net/>
- <https://www.ibm.com/watson/developercloud/doc/common/getting-started-variables.html>
- <https://www.ibm.com/watson/developercloud/doc/common/index.html>

4.7. Explain the advantages of using IBM Bluemix as the cloud platform for Cognitive application development and deployment.

- 4.7.1. Bluemix provides services for securely integrating on-premises and off-premises applications
 - 4.7.2. The services in the Bluemix catalog can be used by applications to meet needs in many functional and non-functional areas
 - 4.7.3. Environmental, operational, and functional Security are built into the platform
 - 4.7.4. Scaling application capacity is simple for applications following modern application design (such as 12-factor methodology)
 - 4.7.5. Bluemix facilitates the integration and configuration of applications and services
 - 4.7.6. Bluemix is built on open source technologies with community governance
- Reference: “What is IBM Bluemix” at

<https://console.ng.bluemix.net/docs/overview/whatisbluemix.html#bluemixoverview>

Section 5 - Administration& DevOps for applications using IBM Watson Developer Cloud Services

5.1. Describe the process of obtaining credentials for Watson services.

SUBTASK(S):

- 5.1.1. Use the Bluemix web interface
- 5.1.2. Get service credentials in Bluemix
- 5.1.3. Get service credentials programmatically
- 5.1.4. Manage organizations, spaces, and assigned users in IBM Bluemix
- 5.1.5. Using tokens with Watson services
- 5.1.6. Obtain a token
 - 5.1.6.1. Use a token
 - 5.1.6.2. Get a token programmatically

References:

https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/getting_started/gs-credentials.shtml
<https://console.ng.bluemix.net/docs/admin/adminpublic.html#administer>
https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/getting_started/gs-tokens.shtml

5.2. Examine application logs provided on IBM Bluemix.

SUBTASK(S):

- 5.2.1. Log for apps running on Cloud Foundry
- 5.2.2. View logs from the Bluemix dashboard
- 5.2.3. View logs from the command line interface
- 5.2.4. Filter logs
- 5.2.5. Configure external logs hosts
- 5.2.6. View logs from external logs hosts

References:

https://console.ng.bluemix.net/docs/monitor_log/monitoringandlogging.html#monitoring_logging_bluemix_apps
<http://docs.cloudfoundry.org/devguide/deploy-apps/streaming-logs.html>

Next Steps

1. Take the [IBM Watson V3 Application](#) Development assessment test.
2. If you pass the assessment exam, visit pearsonvue.com/ibm to schedule your testing sessions.
3. If you failed the assessment exam, review how you did by section. Focus attention on the sections where you need improvement. Keep in mind that you can take the assessment exam as many times as you would like, however, you will still receive the same questions only in a different order.