

Certified for



Watson

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 Cloud.....	13
Section 4 - Developing Cognitive applications using Watson Developer Cloud Services.....	17
Section 5 - Administration & DevOps for applications using IBM Watson Developer Cloud Services.....	21
Next Steps.....	23

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 Middleware Certification item writers use these objectives to develop the questions that they write 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	<u>Define the main characteristics of a cognitive system.</u>
1.2	<u>Explain neural nets.</u>
1.3	<u>Explain machine learning technologies (supervised, unsupervised, reinforcement learning approaches).</u>
1.4	<u>Define a common set of use cases for cognitive systems.</u>
1.5	<u>Define Precision, Recall, and Accuracy.</u>
1.6	<u>Explain the importance of separating training, validation and test data.</u>
1.7	<u>Measure accuracy of service.</u>
1.8	<u>Perform Domain Adaption using Watson Knowledge Studio (WKS).</u>
1.9	<u>Define Intents and Classes.</u>
1.10	<u>Explain difference between ground truth and corpus.</u>
1.11	<u>Define the difference between the user question and the user intent.</u>
Section 2 - Use Cases of Cognitive Services	
2.1	<u>Select appropriate combination of cognitive technologies based on use-case and data format.</u>
2.2	<u>Explain the uses of the Watson services in the Application Starter Kits.</u>
2.3	<u>Describe the Watson Conversational Agent.</u>
2.4	<u>Explain use cases for integrating external systems (such as Twitter, Weather API).</u>
Section 3 – Fundamentals of IBM Watson Developer Cloud	
3.1	<u>Distinguish cognitive services on WDC for which training is required or not.</u>
3.2	<u>Provide examples of text classification using the NLC.</u>
3.3	<u>Explain the Watson SDKs available as part of the services on Watson Developer Cloud.</u>
3.4	<u>Explain the Watson REST APIs available as part of the services on Watson Developer Cloud.</u>
3.5	<u>Explain and configure Natural Language Classification.</u>
3.6	<u>Explain and configure Visual recognition.</u>
3.7	<u>Explain how Personality Insights service works.</u>
3.8	<u>Explain how Tone Analyzer service works.</u>
3.9	<u>Explain and execute Alchemy Language services.</u>
3.10	<u>Explain and configure Retrieve and Rank service.</u>

Section 4 - Developing Cognitive applications using Watson Developer Cloud Services	
4.1	<u>Call a Watson API to analyze content.</u>
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>Deploy a web application to IBM Bluemix.</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>Monitor resource utilization of applications using IBM Watson services.</u>
5.3	<u>Monitoring application performance on IBM Bluemix.</u>
5.4	<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
 - 1.1.3.1. Supervised learning versus unsupervised learning
- 1.1.4. Cognitive systems use natural language processing

1.2. Explain neural nets.

SUBTASK(S):

- 1.2.1. Neural Nets mimic how neurons in the brain communicate
 - 1.2.1.1. Explain the role of synapse and neuron
 - 1.2.1.2. Understand weights and bias
 - 1.2.1.3. List various approaches to neural nets
 - 1.2.1.4. Explain forward and backward propagation
 - 1.2.1.5. Explain gradient descent
 - 1.2.1.6. Explain the activation functions used in gradient descent: Sigmoid vs. hyperbolic tangent function

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

SUBTASK(S):

- 1.3.1. Explain the connection between Machine learning and Cognitive systems
Reference:
http://www.research.ibm.com/software/IBMResearch/multimedia/Computing_Cognition_WhitePaper.pdf
- 1.3.2. Describe some of the main machine learning concepts:
 - 1.3.2.1. Supervised learning:
 - 1.3.2.1.1. Classification
 - 1.3.2.1.2. Regression/Prediction
 - 1.3.2.1.3. Semi-supervised learning
 - 1.3.2.2. Unsupervised learning:
 - 1.3.2.2.1. Artificial neural network
 - 1.3.2.2.2. Association rule learning

1.3.2.2.3. Hierarchical clustering

1.3.2.2.4. Cluster analysis

1.3.2.2.5. Outlier Detection

1.3.2.3. Reinforcement learning

Reference:

<http://stats.stackexchange.com/questions/144154/supervised-learning-unsupervised-learning-and-reinforcement-learning-workflow>

Reference:

https://en.wikipedia.org/wiki/List_of_machine_learning_concepts
<http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>

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

SUBTASK(S):

1.4.1. Customer Call Centers

1.4.1.1. Agent Assist: Q&A

1.4.1.1.1. Problem Solved: Provides a natural language help system so call agents can rapidly retrieve answers to customer questions

1.4.1.1.2. Services used: Conversation, natural language answer retrieval, keyword extraction, and entity extraction

1.4.1.2. Automation: Customer/Technical Support Tickets Routing

1.4.1.3. Customer: Go Moment

1.4.1.3.1. Problems Solved:

- a) Detect the topic of a ticket and route to the appropriate department to handle it
- b) room service, maintenance, housekeeping
- c) Escalate support tickets based on customer sentiment
- d) Route support requests to agents that already solved similar problems by detecting natural language similarities between new customer tickets and resolved ones.

1.4.1.3.2. Services used: natural language (text) classification, keyword extraction, entity extraction, and sentiment/tone analysis

1.4.1. Physicians

1.4.1.4. Expert Advisor:

1.4.1.4.1. Example: Watson Discovery Advisor

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

1.4.1.4.3. Services used: Conversation + natural language answer retrieval, entity extraction

1.4.2. Social Media

1.4.2.1. Data Insights:

1.4.2.1.1. Partner: Ground Signal

1.4.2.1.2. 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.

1.4.2.1.3. Services used: keyword, entity, and sentiment/tone analysis

1.5. Define Precision, Recall, and Accuracy.

SUBTASK(S):

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

1.5.1.1. Definition: Precision is the percentage of documents labelled as positive that are actually positive.

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

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

1.5.2.1. Recall is the percent of documents labelled as positive were successfully retrieved.

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

1.5.3. Accuracy:

1.5.3.1. Accuracy is the fraction of documents relevant to a query that were successfully retrieved.

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

1.5.4. Diagrams like this are often useful in capturing the True/False Positive/Negatives described above:

<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. Etc

1.6.1.4. 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

1.6.3.1.1. Data = Inputs + Outputs

1.6.3.1.2. Input + Output Sets → a set of functions that map input to output

1.6.3.2. We train these functions using the training data

- 1.6.3.3. We select which function gives less errors or better classification or prediction by feeding them each the validation data / blind (cross-validate)
- 1.6.3.4. We select the best outcome
- 1.6.3.5. We test the best outcome (function / neural net (weights,etc)) with the test data

1.7. Measure accuracy of service.

SUBTASK(S):

- 1.7.1. Define accuracy of a system
 - 1.7.1.1. $(\text{True positive} + \text{true negative}) / (\text{TP} + \text{TN} + \text{False P} + \text{False N})$
- 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

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)

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

1.10. Explain difference between ground truth and corpus.

SUBTASK(S):

- 1.10.1. Ground truth is used in both supervised and unsupervised machine learning approaches, yet portray different values and formats. For example, in a typical supervised learning system, ground truth consisted of inputs (questions) and approved outputs (answers). With the aid of logistical regression and iterative training the system improves in accuracy.
- 1.10.2. In unsupervised approach, such as NLC, the ground truth consists of a comma-separated csv or a JSON file that lists hundreds of sample utterances and a dozen or so intents (or classes) classifying those utterances.

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

SUBTASK(S):

- 1.11.1. 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)
- 1.11.2. The user question is the verbatim question
- 1.11.3. The user intent maps the user question to a known classification
- 1.11.4. This is a form of classifying question based on search goals
- 1.11.5. 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.
 - 1.11.5.1. Common examples of user intents:
 - 1.11.5.2. Automation: "Schedule a meeting with Sue at 5pm next Tuesday."
 - 1.11.5.3. Declarative: "I need to change my password."
 - 1.11.5.4. Imperative: "Show me the directions to my the nearest gas station."

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. Services: Q&A, Text classification, entity extraction and, keyword extraction
 - 2.1.1.3. Watson-specific: NLC, R&R, Alchemy Language
- 2.1.2. Agent-assist for phone-based customer call center
 - 2.1.2.1. Data: customer voice recordings
 - 2.1.2.2. Services: Q&A, Speech recognition, text-to-speech, text classification, entity extraction, keyword extraction
 - 2.1.2.3. Watson-specific: NLC, R&R, Alchemy Language
- 2.1.3. Expert advisor use case for physicians
 - 2.1.3.1. Data: natural language intents
 - 2.1.3.2. Services: Q&A, Text classification, entity extraction and keyword extraction
 - 2.1.3.3. Watson-specific: NLC, R&R, Alchemy Language
- 2.1.4. Data insights for Instagram images
 - 2.1.4.1. Data: images
 - 2.1.4.2. Services: Image classification and natural OCR
 - 2.1.4.3. Watson-specific: Visual Insights
- 2.1.5. Data insights for Twitter
 - 2.1.5.1. Data: tweets
 - 2.1.5.2. Services: Text classification, entity extraction, keyword extraction
 - 2.1.5.3. Watson-specific: NLC and Alchemy Language

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

SUBTASK(S):

- 2.2.1. AlchemyLanguage
 - 2.2.1.1. Describe the inputs and outputs of the AlchemyLanguage service
 - 2.2.1.2. Identify common use cases for the AlchemyLanguage service
- 2.2.2. AlchemyData News
 - 2.2.2.1. Describe the inputs and outputs of the AlchemyData News service
 - 2.2.2.2. Identify common use cases for the AlchemyData News service
- 2.2.3. Tone Analyzer
 - 2.2.3.1. Describe the inputs and outputs of the Tone Analyzer service
 - 2.2.3.2. Identify common use cases for the Tone Analyzer service
- 2.2.4. Retr Classifier
 - 2.2.4.1. Describe the inputs and outputs of the Natural Language Classifier service

- 2.2.4.2. Identify common use cases for the Natural Language Classifier service
- 2.2.5. Dialog
 - 2.2.5.1. Describe the inputs and outputs of the Dialog service
 - 2.2.5.2. Identify common use cases for the Dialog service
- 2.2.6. Speech to Text
 - 2.2.6.1. Describe the inputs and outputs of the Speech to Text service
 - 2.2.6.2. Identify common use cases for the Speech to Text service
- 2.2.7. Add trade off analytics

2.3. Describe the Watson Conversational Agent.

SUBTASK(S):

- 2.3.1. Conversational agents in IBM use Dialog and Natural Language Classifier
- 2.3.2. Connie is the concierge at Hilton
- 2.3.3. 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):

- 2.4.1. When systems communicate with each other, this is considered Internet of Things
 - 2.4.1.1. Explain the components of systems communicating with one another
 - 2.4.1.2. Use case of Tweeter and sentiment analysis
 - 2.4.1.3. Use case of the Weather APIs and mission critical decision that are impacted by weather

Section 3 - Fundamentals of IBM Watson Developer Cloud

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.

3.3.1.1. Pre-trained Watson services:

3.3.1.1.1. Watson Text-to-Speech

3.3.1.1.2. Watson Speech-to-text

3.3.1.1.3. Language Translation (conversational, news, and patent domains)

3.3.1.1.4. Alchemy Language (open-domain)

3.3.1.1.5. Watson Visual Insights

3.3.1.1.6. Tone Analyzer

3.3.1.1.7. Personality Insights (social media domain)

3.3.1.2. Services requiring training:

3.3.1.2.1. Natural Language Classifier

3.3.1.2.2. Rank part of Retrieve and Rank

3.3.1.2.3. Visual recognition (custom models)

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

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

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

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.

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 Face Detection/Recognition
- 3.6.4. Describe the capabilities of Natural Scene OCR

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
References:
<https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/personality-insights.html>
- 3.7.3. Describe the personality models of the Personality Insights service
References:
<https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/personality-insights/basics.shtml>

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 Alchemy Language - Sentiment Analysis and Emotion Insights service

References:

<https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/tone-analyzer/index.shtml>

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

3.9. Explain and execute Alchemy Language services.

SUBTASK(S):

- 3.9.1. Identify the capabilities of Alchemy Language
- 3.9.2. Describe the text extraction features of Alchemy Language
- 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

References:

<http://www.alchemyapi.com/api/sentiment-analysis>

- 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

References:

<http://blog.alchemyapi.com/a-step-closer-to-building-empathetic-systems>

- 3.9.7.3. Describe the main types of positive/negative sentiment extracted from digital text

References:

<http://www.alchemyapi.com/api/sentiment/proc.html>

- 3.9.7.4. Describe the API types provided by the Sentiment Analysis service

References:

<http://www.alchemyapi.com/api/sentiment/urls.html>

- 3.9.7.5. Describe the differences between sentiment and emotion analyses

3.10. Explain and configure Retrieve and Rank service.

SUBTASK(S):

- 3.10.1. Explain the function of the Retrieve and Rank service
- 3.10.2. Configure the Retrieve and Rank service
 - 3.10.2.1. Create Solr cluster
 - 3.10.2.2. Create and upload Solr configuration
 - 3.10.2.3. Create Solr collection
 - 3.10.2.4. Upload and index documents
 - 3.10.2.5. Create / update ground truth
 - 3.10.2.6. Create and train Ranker
 - 3.10.2.7. Evaluate result / update ground truth

Section 4 - Developing Cognitive applications using Watson Developer Cloud Services

4.1. Call a Watson API to analyze content.

SUBTASK(S):

- 4.1.1. **Alchemy Language**
 - 4.1.1.1. Create an instance of the Alchemy Language service in Bluemix
 - 4.1.1.2. Select the correct API to call for text extraction, sentiment analysis, or any of the Alchemy Language services.
 - 4.1.1.3. Pass your content to your Alchemy services' endpoint through a RESTful API call
 - 4.1.1.4. Natural Language Classifier
 - 4.1.1.5. Gather sample text from real end users (fake initially if you have to...but not much)
 - 4.1.1.6. Determine the users intents that capture the actions/needs expressed in the text
 - 4.1.1.7. Classify your user text into these user intents
 - 4.1.1.8. Separate your user text into train/test datasets
 - 4.1.1.9. Create an instance of the Natural Language Classifier service in Bluemix
 - 4.1.1.10. Train an NLC classifier on your training dataset
 - 4.1.1.11. Pass your content to your NLC services' endpoint through a RESTful API call
 - 4.1.1.12. Determine the accuracy, precision, and recall of the NLC classifier using your test dataset
- 4.1.2. **Personality Insights**
 - 4.1.2.1. Create an instance of the Personality Insights service in Bluemix
 - 4.1.2.2. Gather text from users in their own voice
 - 4.1.2.3. Ensure you meet the minimum limits for word count (currently 5,000 words) to limit sampling error.
 - 4.1.2.4. 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. Document the primary conversation flow for your users.
- 4.2.2. Determine ways users could diverge from this flow and how to redirect them back.
- 4.2.3. Determine the primary user intents plus paraphrases at various nodes in your conversation
- 4.2.4. Define profile variables
- 4.2.5. Create instances of Dialog and NLC services on Bluemix
- 4.2.6. Upload these intents plus paraphrases into the NLC Service

- 4.2.7. Build out conversation flow with Dialog Service
- 4.2.8. Present your beta conversation agent to end users to capture real end user text
- 4.2.9. Identify areas where users strayed outside the domain of your conversation agent
- 4.2.10. Identify areas where your conversation agent misunderstood the user
- 4.2.11. Update your conversation agent with new intents plus real end user text

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. AlchemyNews
 - 4.3.5.1. Use the top article returned by the search from AlchemyNews as input to AlchemyLanguage-Sentiment Analysis and Tone Analyzer
 - 4.3.5.2. Use the top article returned by the search from AlchemyNews 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. Services: Watson Dialog + Natural Language Classifier + entity extraction (Alchemy Language)
- 4.4.2. Q&A
 - 4.4.2.1. 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

- 4.4.2.2. Services: Watson Dialog + Natural Language Classifier + Retrieve and Rank
- 4.4.3. Agent Assist
 - 4.4.3.1. Goal: Provide natural language help systems so call agents can rapidly retrieve answers to customer questions
 - 4.4.3.2. Services: Watson Dialog + Natural Language Classifier + entity extraction (Alchemy Language)
- 4.4.4. Automated Customer Support Routing
 - 4.4.4.1. 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.
 - 4.4.4.2. Services: Keyword extraction and sentiment analysis (Alchemy Language)
- 4.4.5. Data Insights
 - 4.4.5.1. 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.
 - 4.4.5.2. Services used: Keyword extraction, entity extraction, and sentiment/tone analysis (Alchemy Language)

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. Dialog
 - 4.5.2.1. Upload a dialog
 - 4.5.2.2. Retrieve content
 - 4.5.2.3. Update content for specific nodes
 - 4.5.2.4. Start new conversation
 - 4.5.2.5. Set profile variables
- 4.5.3. Language Translation
 - 4.5.3.1. Upload glossary
 - 4.5.3.2. Return translation status
 - 4.5.3.3. Translate input
- 4.5.4. Alchemy Vision
 - 4.5.4.1. Recognize face
 - 4.5.4.2. Extract link
 - 4.5.4.3. Tag image
 - 4.5.4.4. Detect text

4.6. Deploy a web application to IBM Bluemix.

SUBTASK(S):

- 4.6.1. Configure application's manifest to request the correct memory and app instance allocations
 - 4.6.1.1. Configure application with service credentials extracted from VCAP services
 - 4.6.1.2. Create instances of your required services in IBM Bluemix
 - 4.6.1.3. Install Cloud Foundry command line tools
 - 4.6.1.4. Log-in to IBM Bluemix from the command line
 - 4.6.1.5. Push the application to IBM Bluemix using the Cloud Foundry command line tools

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

References:

https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/getting_started/gs-credentials.shtml

- 5.1.4. Manage organizations, spaces, and assigned users in IBM Bluemix

References:

<https://console.ng.bluemix.net/docs/admin/adminpublic.html#administer>

- 5.1.5. Using tokens with Watson services
 - 5.1.5.1. Obtain a token
 - 5.1.5.2. Use a token
 - 5.1.5.3. Get a token programmatically

References:

https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/getting_started/gs-tokens.shtml

5.2. Monitor resource utilization of applications using IBM Watson services.

SUBTASK(S):

- 5.2.1. Monitor applications running on Cloud Foundry

References:

https://console.ng.bluemix.net/docs/monitor_log/monitoringandlogging.html#monitoring_logging_bluemix_apps

- 5.2.2. Monitor applications by using IBM Monitoring and Analytics for Bluemix

References:

<https://console.ng.bluemix.net/docs/services/monana/index.html#gettingstartedtemplate>

5.3. Monitoring application performance on IBM Bluemix.

SUBTASK(S):

- 5.3.1. Configure performance monitoring
- 5.3.2. Monitor performance of applications

References:

https://console.ng.bluemix.net/docs/services/PerformanceInsights/index.html#perfinsght_prfmn_c

5.4. Examine application logs provided on IBM Bluemix.

SUBTASK(S):

- 5.4.1. Log for apps running on Cloud Foundry
- 5.4.2. View logs from the Bluemix dashboard
- 5.4.3. View logs from the command line interface
- 5.4.4. Filter logs
- 5.4.5. Configure external logs hosts
- 5.4.6. View logs from external logs hosts

References:

https://console.ng.bluemix.net/docs/monitor_log/monitoringandlogging.html#monitoring_logging_bluemix_apps

Additional links: <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.