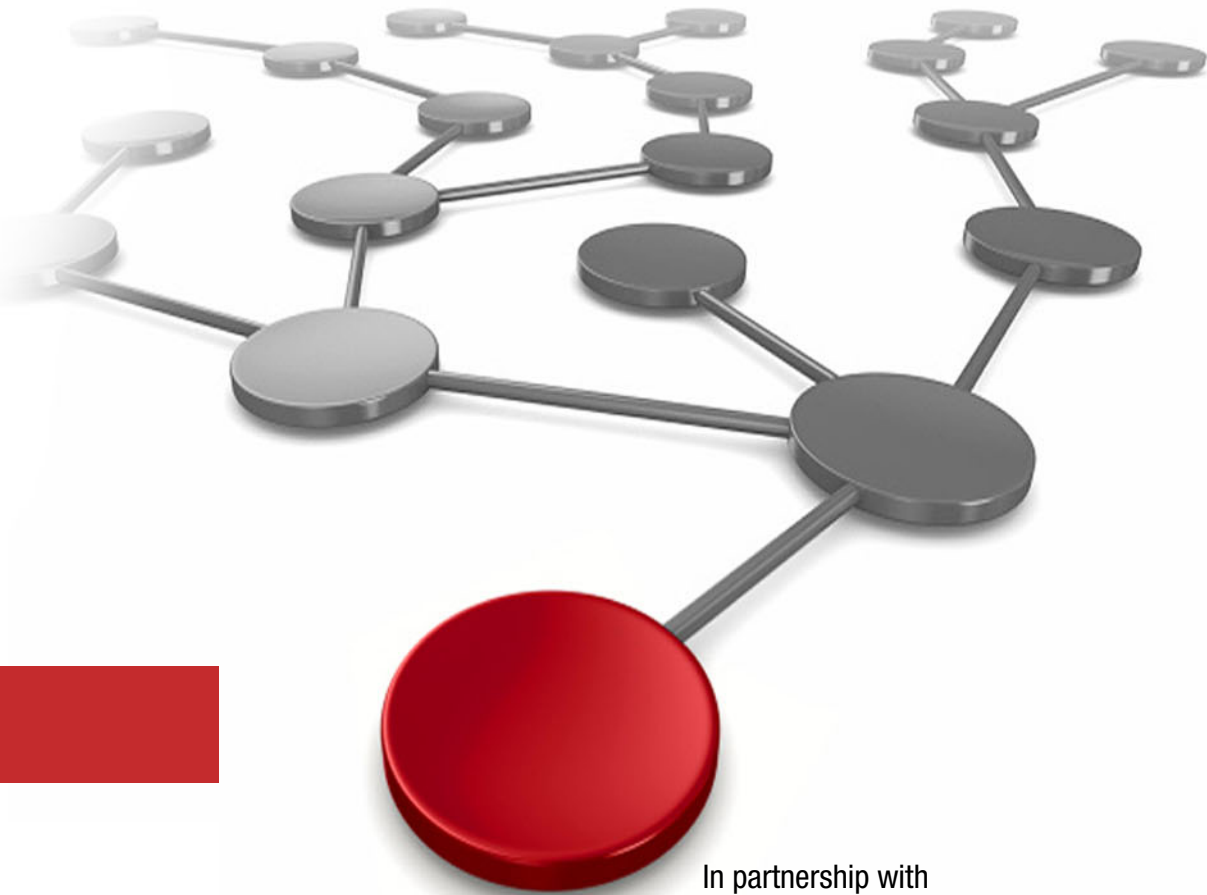


# Building Cognitive Applications with IBM Watson Services: Volume 1 Getting Started

Dr. Alfio Gliozzo  
Chris Ackerson  
Rajib Bhattacharya  
Addison Goering  
Albert Jumba  
Seung Yeon Kim  
Laksh Krishnamurthy  
Thanh Lam  
Angelo Littera  
Iain McIntosh  
Srini Murthy  
Marcel Ribas



 **Cloud**

In partnership with  
**IBM Skills Academy Program**





International Technical Support Organization

**Building Cognitive Applications with IBM Watson  
Services: Volume 1 Getting Started**

June 2017

**Note:** Before using this information and the product it supports, read the information in “Notices” on page v.

**First Edition (June 2017)**

This edition applies to IBM Watson services in IBM Bluemix.

© Copyright International Business Machines Corporation 2017. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.



# Contents

<b>Notices</b> .....	v
Trademarks .....	vi
<b>Preface</b> .....	vii
Authors .....	vii
Now you can become a published author, too! .....	x
Comments welcome .....	x
Stay connected to IBM Redbooks .....	xi
<b>Chapter 1. Introduction to cognitive computing</b> .....	1
1.1 Brief history of cognitive computing .....	2
1.1.1 The eras of computing .....	2
1.1.2 The future of computing is cognitive .....	4
1.1.3 Impact of cognitive computing to our lives .....	4
1.2 Basic concepts .....	5
1.3 Characteristics of cognitive systems .....	8
1.3.1 Solving real life problems with cognitive systems .....	9
1.4 References .....	10
<b>Chapter 2. Cognitive business and IBM Watson</b> .....	11
2.1 Landscape of cognitive computing in the industry .....	12
2.1.1 Consumer market: Cognitive computing offerings .....	13
2.1.2 Enterprise market: Cognitive computing offerings .....	14
2.1.3 Delivering cognitive services: Cloud and open source projects .....	15
2.1.4 Cognitive computing and the future of jobs .....	16
2.2 Introducing IBM Watson .....	16
2.2.1 Watson APIs: Build with Watson .....	17
2.2.2 IBM Watson applied to industries, businesses, and science .....	18
2.2.3 Watson use cases .....	23
2.2.4 Watson demonstrations .....	26
2.3 References .....	28
<b>Chapter 3. Introduction to question-answering systems</b> .....	29
3.1 The Jeopardy! challenge .....	30
3.2 DeepQA system architecture .....	31
3.3 Exploring the DeepQA pipeline through an example .....	34
3.3.1 Question analysis .....	34
3.3.2 Primary search .....	35
3.3.3 Hypothesis generation .....	36
3.3.4 Hypothesis and evidence scoring .....	36
3.3.5 Final merging and ranking .....	38
3.4 References .....	40
<b>Chapter 4. Evolution from DeepQA to Watson Developer Cloud</b> .....	41
4.1 Why commercialize Watson .....	42
4.2 Refresher of DeepQA architecture .....	44
4.3 Evolution to Watson Developer Cloud .....	46
4.3.1 Evolution of question analysis .....	48
4.3.2 Microservices and robust tooling evolved from DeepQA .....	54

4.4	Watson Conversation service . . . . .	56
4.5	Watson Discovery service . . . . .	58
4.6	Evolution summary . . . . .	60
4.7	References . . . . .	60
<b>Chapter 5. Domain adaptation . . . . .</b>		<b>61</b>
5.1	Introduction to domain adaptation . . . . .	62
5.2	IBM Watson Developer Cloud and domain adaptation . . . . .	63
5.2.1	Watson Conversation . . . . .	64
5.2.2	Watson Language Translator . . . . .	67
5.2.3	Watson Natural Language Classifier . . . . .	69
5.2.4	Watson Retrieve and Rank . . . . .	71
5.2.5	Watson Visual Recognition . . . . .	73
5.2.6	Watson Speech to Text . . . . .	75
5.2.7	Watson Text to Speech . . . . .	77
5.2.8	Watson Natural Language Understanding . . . . .	79
5.2.9	Watson Discovery . . . . .	80
5.3	Watson Knowledge Studio . . . . .	81
5.3.1	Watson Knowledge Studio domain adaptation overview . . . . .	82
5.3.2	Example: Creating a machine learning model . . . . .	84
5.3.3	Deploying a machine-learning annotator to Watson Natural Language Understanding . . . . .	104
5.3.4	Deploying a machine-learning annotator to Watson Discovery . . . . .	107
<b>Related publications . . . . .</b>		<b>109</b>
IBM Redbooks . . . . .		109
Online resources . . . . .		109
Help from IBM . . . . .		112

# Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, US*

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.


## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

# Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation, 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 <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

Bluemix®	Power Systems™	Watson™
Cognos®	PureApplication®	Watson Discovery Advisor™
Discovery Advisor®	Redbooks®	Watson Health™
Global Business Services®	Redbooks (logo)  ®	Watson IoT™
IBM®	Redpapers™	WebSphere®
IBM Watson®	Smarter Cities®	
IBM Watson IoT™	THINK®	

The following terms are trademarks of other companies:

Evolution, and Workforcebydesign are trademarks or registered trademarks of Kenexa, an IBM Company.

Microsoft, and the Windows logo 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.

Other company, product, or service names may be trademarks or service marks of others.

# Preface

The *Building Cognitive Applications with IBM Watson Services* series is a seven-volume collection that introduces IBM® Watson™ cognitive computing services. The series includes an overview of specific IBM Watson® services with their associated architectures and simple code examples. Each volume describes how you can use and implement these services in your applications through practical use cases.

The series includes the following volumes:

- ▶ *Volume 1 Getting Started*, SG24-8387
- ▶ *Volume 2 Conversation*, SG24-8394
- ▶ *Volume 3 Visual Recognition*, SG24-8393
- ▶ *Volume 4 Natural Language Classifier*, SG24-8391
- ▶ *Volume 5 Language Translator*, SG24-8392
- ▶ *Volume 6 Speech to Text and Text to Speech*, SG24-8388
- ▶ *Volume 7 Natural Language Understanding*, SG24-8398

Whether you are a beginner or an experienced developer, this collection provides the information you need to start your research on Watson services. If your goal is to become more familiar with Watson in relation to your current environment, or if you are evaluating cognitive computing, this collection can serve as a powerful learning tool.

This IBM Redbooks® publication, Volume 1, introduces cognitive computing and its motivating factors, history, and basic concepts. It describes the industry landscape for cognitive computing and introduces Watson, the cognitive computing offering from IBM. It also describes the nature of the question-answering (QA) challenge that is represented by the Jeopardy! quiz game and it provides a high-level overview of the QA system architecture (DeepQA), developed for Watson to play the game. This volume charts the evolution of the Watson Developer Cloud, from the initial DeepQA implementation. This volume also introduces the concept of domain adaptation and the processes that must be followed to adapt the various Watson services to specific domains.

## Authors

This book was produced by a team of specialists from around the world working in collaboration with the IBM International Technical Support Organization.

**Dr. Alfio Gliozzo** is Research Leader at IBM T.J. Watson and Adjunct Professor at Columbia University, where he teaches cognitive computing. He was part of the research team that built Watson, the question-answering system that defeated the Jeopardy! grand champions of all time. Alfio's research focuses on knowledge induction from text, using deep learning, natural language processing, semantic web, and reasoning.

**Chris Ackerson** is a Senior Offering Manager with IBM Watson, leading development of enterprise artificial intelligence products. Prior to joining Offering Management, Chris was a Solutions Architect with the Watson Ecosystem, where he worked with dozens of IBM Business Partners to build commercial applications on top of the Watson Developer Cloud. Chris studied Electrical and Computer Engineering at Carnegie Mellon University, where his passion for machine learning and robotics began.

**Rajib Bhattacharya** is a Solution Architect in the Watson Customer Engagement Services Team at the IBM India Software Labs. Rajib has more than 12 years of experience in IT Consulting. He is a seasoned analytics architect focusing on business analytics, data warehousing, big data, and cognitive computing. He is a frequent speaker at international IBM conferences. Rajib holds multiple patents in analytics. He is passionate in helping IBM clients to consolidate their enterprise data, which enables them to make valuable business decisions using analytics tools. Rajib holds a Master of Computer Applications (MCA) degree from West Bengal University of Technology and is certified in multiple technologies including IBM Cognos®, Big Data, R Programming, and Agile methodologies. Over the years, he has developed and delivered various training programs on analytics and cognitive computing for IBM Software Services and IBM University Relations.

**Addison Goering** is a Certified IT Specialist in the IBM Cloud Learning and Skills Development team. His main specialty is the design, development, and delivery of courses in IBM Cloud. He has developed and delivered courses ranging from webinars to week-long workshops on products such as IBM WebSphere® Enterprise Service Bus, WebSphere Application Server, and IBM PureApplication® System. He was Lead Developer with the team that developed education curriculum for IBM PureApplication System. Addison holds a B.S. degree in elementary education from Keene State College in New Hampshire, US, mainframe certification from DePaul University in Chicago, and several certifications from IBM.

**Albert Jumba** is a Software Engineer in the IBM Hybrid Cloud organization in IBM Kenya. Albert has extensive experience working with clients in Africa and the Middle East on projects that include a broad range of technologies, such as IBM Bluemix®, security, cloud, collaboration, service management, IBM Smarter Cities®, and storage. Albert is a frequent guest instructor in universities across Africa as part of the IBM Skills Academy Program.

**Seung Yeon Kim** is a Client Technical Leader for Electronics, Telco, Media and Entertainment, and Healthcare Industries in IBM Korea. As a senior IT Architect, Seung Yeon led data monetization initiatives for communications service providers, Connected Living for Electronics, and Cognitive Knowledge Center for cross-industry enterprise. Seung Yeon has helped IBM clients to apply IBM cognitive computing technologies, such as Watson APIs, IBM Watson Health™, IoT, and data science to business solutions in several industries. She had over 15 years of experience in research and development in IBM and in the telecommunications industry before joining IBM. Her research projects include Device Symbiosis with the IBM T.J. Watson Research Center, SNF (social Network Platform) with IBM Haifa Research Center, and Internet-based virtual reality with the IBM Center for the Business of Government. Her academic background includes 3D computer graphics and robotics of computer sciences and her thesis is about autonomous motion planning of a humanoid robot's arm.

**Laksh Krishnamurthy** is a Senior Technical Staff Member with the IBM Watson group. He joined IBM Watson in 2012 and has led several research, customer, and internal projects. Laksh is passionate about machine learning and deep learning, especially in the area of domain adaptation. His current work includes enabling several internal IBM products and offerings to embed Watson services. Laksh is also an IBM Master Inventor and he helps with mentoring many IBMers in the area of patenting in IBM.

**Thanh Lam** is an Instructor and Course Developer with IBM Technical Training and Lab Services. Thanh creates courses for cloud computing and IBM Power Systems™. These courses provide technical training for clients around the world through IBM Global Training Partners. Thanh also designs and creates hands-on labs for these courses to help clients work with IBM products in the cloud computing areas. Thanh teaches these courses in virtual classrooms and at conferences. He is the co-author of several Redbooks publications. Thanh holds a degree in Doctor of Professional Studies in Computing from Pace University, New York.

**Angelo Littera** is a Certified Senior Technology Architect with IBM Italy. He joined IBM in 1996 and, with over nineteen years of practical experience, his involvement covered all aspects of the project life-cycle, from the engagement phase to the delivery phase. Angelo is responsible for designing technical solutions in the areas of cloud and cognitive computing and blockchain. During his career, Angelo worked on several complex projects to coordinate working groups of professionals from IBM and other companies, demonstrating horizontal competence and the ability to quickly acquire vertical skills, when needed. Angelo is the author of several IBM Redbooks publications and contributes to various IBM blog sites.

**Iain McIntosh** is an IBM Certified Architect with the IBM Watson Cloud Complex Implementations team. Iain has been an Architect at IBM for 17 years, creating innovative and value-driven solutions in automation, systems management, service management, and Watson for IBM clients. Iain has led several solution designs and built projects for IBM Watson Discovery Advisor® and Watson Developer Cloud. Iain holds Advanced Diplomas in Electrical Engineering and Control System Engineering. He has co-authored two IBM publications, one of which was presented at an IEEE conference.

**Srini Murthy** is Senior Certified IT Architect with IBM India. He has over 23 years of experience in the IT industry. His areas of expertise include consulting, cloud, analytics, Watson, cross-IBM solutions and enterprise architecture. Srini holds an MBA degree from Edinburgh Business School. He has successfully developed solutions to realize business transformation in the Banking, Media and Entertainment, and Healthcare industries. In his current role, Srini is passionate about building cloud and cognitive computing solutions to achieve an organization's business goals. He is a technology evangelist for Watson, analytics, and commerce, and devotes time to deploy these technologies in a heterogeneous environment with open source and non IBM technologies. He has developed several artifacts in the areas of architecture, industry points of view (POVs), and emerging technologies.

**Marcel Ribas** is a Bluemix Technical Specialist with IBM. He is an Electrical Engineer and holds a Master's Degree in Computer Science, with emphasis in Cognitive Computing. His research included developing a cognitive system to facilitate online learning. He taught programming at the Universidade do Sul de Santa Catarina, in Brazil. A blend of coder and system administrator, Marcel has an extensive background in software development and IT infrastructure projects, and recently has been helping companies of all sizes in their cloud journey. His areas of interest include cloud-native applications, microservices, cognitive computing, open-source projects, and productivity. He currently resides in Austin, TX in the US and enjoys creating innovative applications with IBM Bluemix.

The project that produced this publication was managed by **Marcela Adan**, IBM Redbooks Project Leader, ITSO.

Thanks to the following people for their contributions to this project:

Garry D'Orazio

**IBM Global Markets, IBM Australia**

Patricia Amor Garcia De Jalon

**Learning Development Europe, IBM Global Business Services®**

Keving Gong

**IBM Watson Visual Recognition Offering Management**

Prabhat Manocha

**IBM Global Markets Enterprise, IBM India**

Juan Pablo Napoli

**Skills Academy Worldwide Leader, IBM Global University Programs**

Kyungsoon Kelly Um  
**IBM Global Markets Enterprise, IBM Korea**  
Elias Valenzuela  
**Learning Development Europe, IBM Global Business Services**

## Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- ▶ Send your comments in an email to:

[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400



## Stay connected to IBM Redbooks

- ▶ Find us on Facebook:  
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:  
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:  
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:  
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:  
<http://www.redbooks.ibm.com/rss.html>





# Introduction to cognitive computing

The explosion of data, mainly unstructured data, over the past few years led to the development of a new type of computer system known as a *cognitive system*. Unlike the programmable computers that preceded it, the focus of cognitive systems is not about doing fast calculations on large amounts of data through traditional computer programs. Cognitive systems are about exploring the data, finding new correlations, and new context in that data to provide new solutions. Cognitive systems aim at expanding the boundaries of human cognition rather than replacing or replicating the way the human brain works.

Cognitive computing is becoming a new industry. A new industrial revolution is coming, connected to job automation in transportation, customer care, and healthcare, to name a few. The livelihood of such a revolution will be a new generation of skilled developers who understand cognitive computing well enough to envision new business applications, and ultimately build the new cognitive web.

This chapter briefly summarizes the history of cognitive computing and the eras of computing because, to understand the future of cognitive computing, placing it in historical context is important.

This chapter also describes basic concepts that are relevant to any discussion about cognitive systems. It introduces the key characteristics and highlights some applications, widely used today, that are based on cognitive technologies.

The following topics are covered in this chapter:

- ▶ Brief history of cognitive computing
- ▶ Basic concepts
- ▶ Characteristics of cognitive systems
- ▶ References

## 1.1 Brief history of cognitive computing

The concept of intelligent machines has existed for a long time. Surprisingly, in the 19th century, mathematician George Boole's 1854 book, *The Laws of Thought*, showed that logical operators (and, or, not) provided the basis for the laws of thought. About that same time, Charles Babbage conceived of creating what he described as an *analytical engine*.

In 1950, Alan Turing, an English computer scientist and mathematician, addressed the problem of artificial intelligence and proposed an experiment that became known as the *Turing test*. It is a test of a machine's ability to exhibit intelligent behavior similar to a human. The test was an adaptation of a Victorian-style competition called the "imitation game." The Turing experiment was based on a human evaluator that judged natural language conversations between a human and a machine designed to generate human-like responses. The test studied whether the interrogator can determine which responses are given by a computer and which ones by a human. The idea was that if the questioner could not tell the difference between human and machine, the computer would be considered to be thinking.

The term *artificial intelligence* was first coined by Prof. John McCarthy for a conference on the subject, held at Dartmouth College in 1956. McCarthy defines the subject as the "science and engineering of making intelligent machines, especially intelligent computer programs."<sup>1</sup>

In 1960, computing pioneer J.C.R. Licklider wrote his seminal paper *Man-Computer Symbiosis*.<sup>2</sup> The paper describes Licklider's vision for a complementary or symbiotic relationship between humans and computers. The following quote is an example of Licklider's research and insights:

"Man-computer symbiosis is an expected development in cooperative interaction between men and electronic computers. It will involve very close coupling between the human and the electronic members of the partnership. The main aims are:

1. To let computers facilitate formulative thinking as they now facilitate the solution of formulated problems.
2. To enable men and computers to cooperate in making decisions and controlling complex situations without inflexible dependence on predetermined programs...

Preliminary analyses indicate that the symbiotic partnership will perform intellectual operations much more effectively than man alone can perform them."

### 1.1.1 The eras of computing

To understand the future of cognitive computing, placing it in historical context is important. To date, two distinct eras of computing have occurred: the *tabulating era* and the *programming era*. We are entering the third and most transformational era in computing's evolution, the *cognitive computing era* (cognitive era).

---

<sup>1</sup> What is Artificial Intelligence, The Society for the Study of Artificial Intelligence and Simulation of Behavior, <http://www.aisb.org.uk/public-engagement/what-is-ai>

<sup>2</sup> *Man-Computer Symbiosis*, J.C.R. Licklider, 1960, <https://groups.csail.mit.edu/medg/people/psz/Licklider.html>

Figure 1-1 shows the three eras of computing.

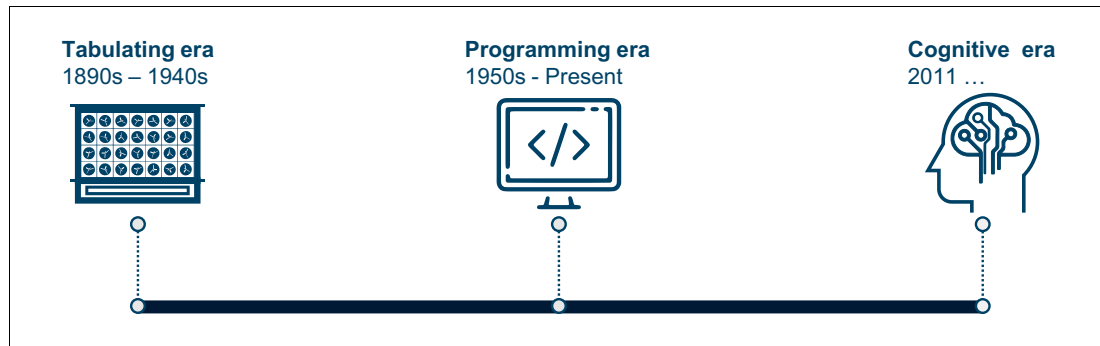


Figure 1-1 The three eras of computing

The eras can be described as follows:

► **Tabulating era (1890s - 1940s)**

The first era of computing consisted of single-purpose electromechanical systems that counted, using punched cards to input and store data, and to eventually instruct the machine what to do. These tabulation machines were essentially calculators designed to count and summarize information and they did it really well but were ultimately *limited to a single task*.

These machines supported the scaling of business and society and were used in government applications such as process population census data and business applications such as accounting and inventory control. Tabulating machines evolved to a class of machines, known as *unit record equipment*, and the data processing industry.

► **Programming era (1950s - present)**

This era started with the shift from mechanical tabulators to electronic systems and began during World War II, driven by military and scientific needs. Following the war, digital “computers” evolved rapidly and moved into businesses and governments. The programmable computing era begins.

The big change is the introduction of general purpose computing systems that are programmable: they can be reprogrammed to perform different tasks and solve multiple problems in business and society. But ultimately, they must be programmed and are still somewhat constrained in the interaction with human beings. Everything we now know as a computing device, from the mainframe to the personal computer, to the smartphone and tablet, is a programmable computer. Some experts believe that this era of computing will continue to exist indefinitely.

► **Cognitive era (2011 - future)**

As Licklider predicted, cognitive computing is a necessary and natural evolution of programmable computing. Cognitive computing systems are meant to *extend the boundaries of human cognition*. Cognitive computing technologies are not about replacing or necessarily even replicating the way that the human brain works; they are about extending the capabilities of the human brain. Humans excel at reasoning, deep thinking, and solving complex problems. But the human ability to read, analyze, and process huge volumes of data, both structured and unstructured, is quite poor. That, of course, is the strength of the computer system. The first role of a cognitive computing system is to combine strengths of human and machine into a collaborative situation.

Another key element of cognitive systems is a more *natural interaction* between human and machine, combined with the capability to *learn and adapt over time*.

## 1.1.2 The future of computing is cognitive

In his paper, *Computing, cognition and the future of knowing: How humans and machines are forging a new age of understanding*,<sup>3</sup> Dr. John E. Kelly III states this:

“Those of us engaged in serious information science and in its application in the real world of business and society understand the enormous potential of intelligent systems. The future of such technology — which we believe will be cognitive, not “artificial”— has very different characteristics from those generally attributed to AI, spawning different kinds of technological, scientific and societal challenges and opportunities, with different requirements for governance, policy and management.”

In the same paper, Dr. Kelly defines cognitive computing:

“Cognitive computing refers to systems that learn at scale, reason with purpose and interact with humans naturally. Rather than being explicitly programmed, they learn and reason from their interactions with us and from their experiences with their environment.”

Current demands driven by big data and the need for more complex evidence-based decisions, are going beyond the previous rigid rule and logic-based approach to computing. Cognitive computing enables people to create a new kind of value by finding answers and insights locked away in volumes of data. Cognitive computing serves to enhance human expertise with systems that reason about problems like a human does.

When we as humans seek to understand something and to make a decision, we go through four key steps:

1. *Observe* visible phenomena and bodies of evidence.
2. *Interpret* what we see by drawing on what we know in order to generate hypotheses about it means.
3. *Evaluate* which hypotheses are right or wrong.
4. *Decide* (choose) the option that seems best and act accordingly.

Just as humans become experts by going through the process of observation, evaluation and decision-making, cognitive systems use similar processes to reason about the information they absorb.

## 1.1.3 Impact of cognitive computing to our lives

Whether you realize this or not, cognitive computing is already having an impact on our lives. Often when you talk to a call center, your interaction is likely with a computer. Articles that you read might have been written by a machine. In many cases, such as online shopping, cognitive computing understands your behavior and activities and makes recommendations based on that understanding. Chatbots that are powered by cognitive computing have been built to successfully support complaint-resolution services.

Many professions are being enhanced by cognitive computing. For example, a doctor diagnosing a patient with unusual symptoms would have to search through vast amounts of information to arrive at a proper diagnosis. Cognitive computing can assist that doctor by doing much of the research and preliminary analysis for the doctor and might also be able to recommend next steps.

---

<sup>3</sup> *Computing, cognition and the future of knowing: How humans and machines are forging a new age of understanding*, Dr. John E. Kelly III, October, 2015:  
[https://www.research.ibm.com/software/IBMResearch/multimedia/Computing\\_Cognition\\_WhitePaper.pdf](https://www.research.ibm.com/software/IBMResearch/multimedia/Computing_Cognition_WhitePaper.pdf)

Consider a wealth manager advising clients on their individual retirement portfolios. While basic facts and rules apply, individual needs, circumstances, and interests come into play. Sorting through all the related information and customizing the recommendations to a particular client can be an overwhelming task that is made easier by cognitive computing.

In essence, cognitive computing can put into context the information many professionals handle on a daily basis in order to drive real value from it.

## 1.2 Basic concepts

Consider these basic concepts:

► Cognition

Cognition, the “act of thinking,” is the mental process of acquiring understanding through thought and personal or shared experiences. Brain-based skills are part of every human action and are essential in carrying out any task, from the simplest to the most difficult. Tasks include human senses (hearing, touch, smell, sight, taste, and even extra-sensory perception), learning, remembering, motor skills, language, empathy, social skills, and problem solving capabilities.

As stated, cognition is the process of acquiring knowledge through thoughts, experiences, and senses. Cognitive processing helps us understand and interact with the world around us from the basic to the complex.

► Artificial Intelligence (AI)

The study and development of AI systems aim at building computer systems able to perform tasks that normally require human intelligence. AI-based machines are intended to perceive their environment and take actions that optimize their level of success. Today’s AI can be considered weak, in that it is designed to perform narrow and specific tasks. The goal of many researchers is to create strong AI that learns like a human and can solve human-type problems.

AI research uses techniques from many fields, such as computer science, philosophy, linguistics, economics, speech recognition, and psychology, which are manifested in applications, such as control systems, natural language processing, facial recognition, speech recognition, analytics, pattern matching, data mining, and logistics.

► Cognitive computing

Humans are inherently capable of a set of skills that help us learn, discover, and make decisions:

- Humans can apply common sense, morals, and reason through dilemmas.
- Humans can think of new ideas and make generalizations when essential clues and pieces of information are missing.
- But humans are restricted by the amount of time spent to learn, process, and absorb new information, and limited by the unconscious biases we all possess that influence the decisions we make.

Cognitive computing is among the subdisciplines that shape AI. It is about putting together a system that combines the best of human and machine capabilities (Figure 1-2 on page 6). Consider capabilities that humans naturally have, such as imagination and emotions, combined with capabilities that computers excel at, such as number crunching, identifying patterns, and processing huge amounts of information. Cognitive computing uses machine strengths to “simulate” the human thought processes in a computerized model.

Cognitive systems use techniques, such as machine learning, data mining, natural language processing, and pattern matching to mimic how a human brain works. Such systems are ideal to interact with an increasingly complex world.

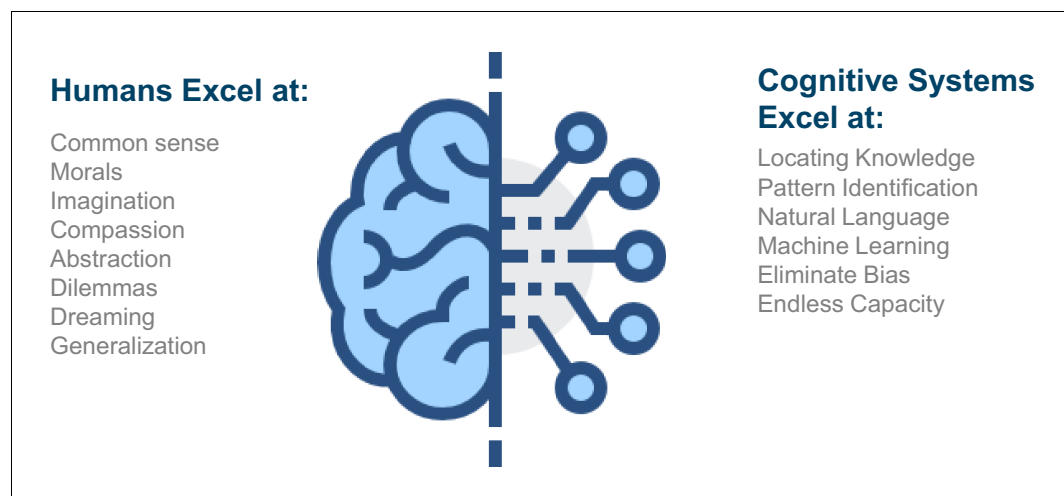


Figure 1-2 Humans and cognitive systems are complementary

► Big data

Often big data characteristics are defined by the five V's: variety, volume, velocity, veracity, and visibility. Big data requires innovative forms of information processing to draw insights, automate processes, and assist in decision making. Big data can be structured data that corresponds to a formal pattern, such as traditional data sets and databases. Also big data includes semi-structured and unstructured formats, such as word processing documents, videos, images, audio, presentations, social media interactions, streams, web pages, and many other kinds of content. Unstructured data is not contained in a regular database and is growing exponentially, making up the majority of all the data in the world.

► Question-answering (QA) technology

Cognitive systems can ingest millions of pages of text and apply question-answering technology to respond to questions posed by humans in natural language. This approach allows people to “ask” and get almost instantaneous answers to complex questions. Combined with other application programming interfaces (APIs) and advanced analytics, QA technology distinguishes itself from the conventional search (that is triggered by keywords) by providing a more conversational discussion.

► Machine learning (ML)

Machine learning is a type of AI that gives computers the ability to learn and act without being explicitly programmed. This means that the computer model gets better over time by learning from its mistakes and new experiences (being exposed to new data), increasing its intelligence. If a computer program can improve how it performs certain tasks that are 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.

► Natural language processing (NLP)

NLP is a field of AI and it refers to the processing by computers of natural language. Natural language is any human language, such as English, Spanish, Arabic, or Japanese, to be distinguished from computer languages, such as Java, Fortran, or C++.



NLP is the ability of computer software to understand human speech. By using NLP capabilities, computers can analyze text that is written in human language and identify concepts, entities, keywords, relations, emotions, sentiment, and other characteristics, allowing users to draw insights from content.

Any system that takes natural language as input and is capable of processing it is a natural language processing system (for example, spam-detection software). A spam classifier is a system that looks at the content of the email subject line to assess whether the received email is or is not spam.

- ▶ Cloud computing

Cloud computing is a general term that describes delivery of on-demand services, usually through the Internet, on a pay-per-use basis. Companies worldwide offer their services to customers. Services might be data analysis, social media, video storage, e-commerce, and cognitive computing in a way that is available through the Internet and supported by cloud computing.

- ▶ Application program interfaces (APIs)

In general, APIs expose capabilities and services. APIs enable software components to communicate with each other easily. The use of APIs as a method for integration injects a level of flexibility into the application lifecycle by making the task easier to connect and interface with other applications or services. APIs abstract the underlying workings of a service, application, or tool, and expose only what a developer needs, so programming becomes easier and faster.

Cognitive APIs are usually delivered on an open cloud-based platform, on which developers can infuse cognitive into digital applications, products, and operations by using one or more of the available APIs.

In the cognitive computing model, all these concepts are combined, eliminating the need for users to be experts in cognitive methods and to allow them to focus on creating better solutions (Figure 1-3).

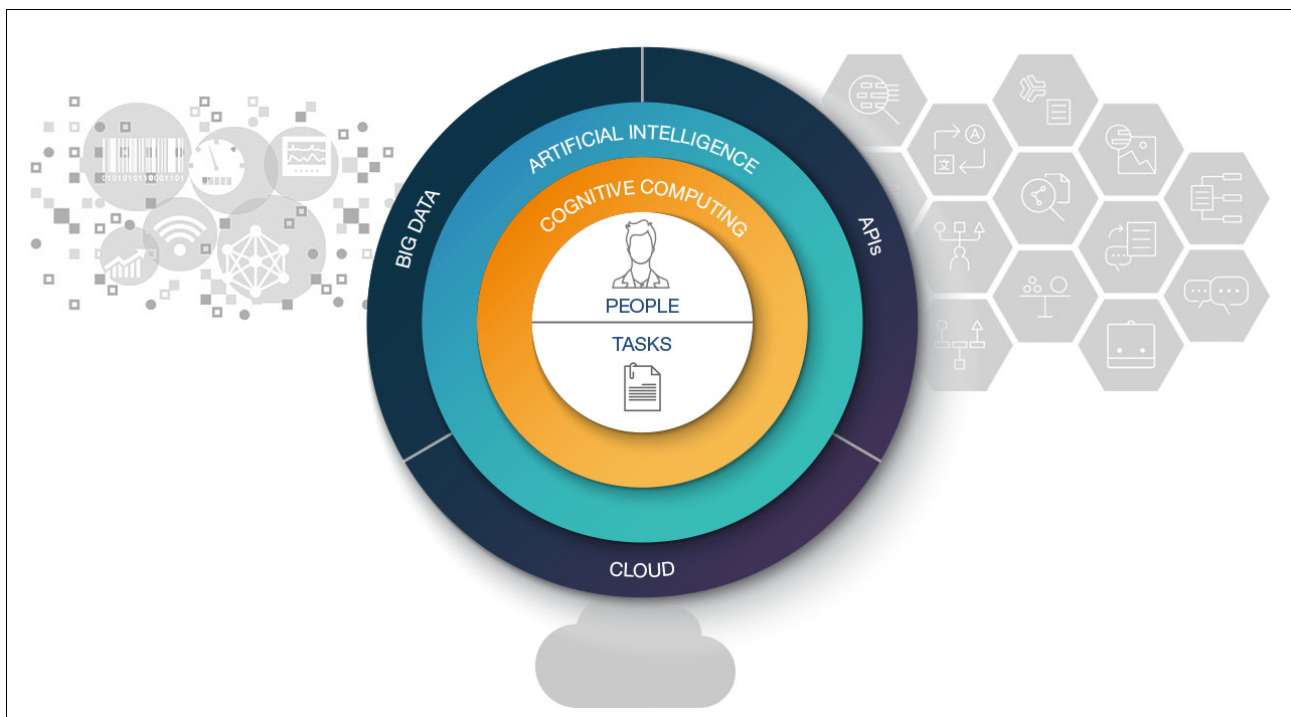


Figure 1-3 Cognitive systems free users to focus on building better solutions for day-to-day and big problems

How do these concepts and technologies relate to each other?

The cognitive computing model intends to have high value in various domains. By applying this model, users do not need to spend time learning intricate details about tools in order to use the tools effectively or spend time interpreting vast amounts of information to draw conclusions. Instead, users spend their time identifying useful patterns, making decisions, and taking action to improve business and operational processes.

Because cognitive computing mimics thinking, the quality of the output is only as good as the algorithms and models used at the start. These models are improved with machine learning.

While a human expert might spend weeks analyzing volumes of data, the computer model can do that in seconds. For example, a team of medical practitioners can carry out a study that monitors hundreds of children for many months to predict factors that cause diabetes in toddlers. In the near future, the same study can be accurately predicted by a computer model that takes seconds to analyze volumes of data, at a much lower cost. For even more added value, other sources of data can be included to improve prediction results. Examples of data to include are family history, life style, cultural norms, and family activities. These are the types of data that make the equivalent human-led research take several years to complete.

One big problem with most analytical tools is that they require a subject matter expert (for example, pilot, doctor, lawyer) to become a computer expert. One goal of cognitive computing is to demand only conversational skills from the subject matter expert to enable that person to draw valuable insights. Data mining and analysis now means “simply ask.” Over time, NLP and QA technologies have become better at identifying speech patterns and truly understanding what the user says in the context of the information available.

With much data to analyze, will you need a super computer so that you can gain insights? Here is where the power of the cloud computing can help. Various vendors have established cloud computing environments and offer access to the cloud over the Internet. Users request the services they need and provide access to their data. Vendors offer a pay-per-use model and provide customization of the environment to fit the particular needs of users. The cloud computing model greatly lowers barriers to access, and with global availability anyone in the world with Internet connectivity has access to these services.

Various APIs that provide access to various services enable quick, easy, and intuitive access to computing systems. Most of the APIs are independent of the programming language, which means your developers can work in any programming language. Using APIs for sharing data, services, and business functions between endpoints (such as applications, devices, and websites) creates the opportunity to lower the cost and time for integration.

## 1.3 Characteristics of cognitive systems

Many people believe that the only way to handle the onslaught of data today and the future is through the use of cognitive systems. Cognitive systems have several key characteristics:

- ▶ An important concept to understand is that the first key element of cognitive systems is to *expand the boundaries of human cognition* rather than replace or replicate the way the human brain works. Humans excel at thinking deeply and solving complex problems, however our ability to read, analyze, and process huge volumes of data is poor. Reading, analyzing, and leveraging huge volumes of data is the strength of computer systems. A key element of a cognitive system is to *combine those two strengths (human and computer) into a collaborative solution*. More than searching through huge amounts of data, the cognitive system must combine different pieces of information together, and possibly do some reasoning to make connections and relationships.

The system needs to do enough analysis to pull out key elements, understand the problem the human is trying to solve, and based on that context bring information to bare on the problem. The goal is for a human to easily leverage the information provided by the cognitive system and enable the human to explore the evidence and use this insight to solve their problem or make decisions.

- ▶ The second key element is to have *a more natural interaction between computers and humans*. Until recently, to interact with computers, humans had to adapt the way they work to the computer interface, which was often rigid and inflexible. Cognitive systems provide a much more natural engagement between the computer and the human. Speech recognition, for example, enables the human to interact with the computer by using voice commands.
- ▶ A third key element of cognitive systems is the *use of learning, specifically machine learning*. Machine learning has been pursued for a long time and cognitive systems must go beyond the core foundations of machine learning.
- ▶ The intent is to broaden the potential for learning and the ability of a to *adapt over time with use*, which is a fourth key element of cognitive systems. So as you use these applications, a feedback mechanism captures the results of that interaction and the system must learn from the resulting interaction and evolve automatically over time, improving its performance.

With this base of understanding, you can think about cognitive systems as reaching to provide and, in many cases, already providing these capabilities:

- ▶ **Understand:** Cognitive systems understand imagery, language, and other unstructured data like humans. Cognitive system operationalize virtually all data (structured and unstructured) like humans do.
- ▶ **Reason:** Cognitive systems can reason, grasp underlying concepts, form hypotheses, and infer and extract ideas.
- ▶ **Learn:** With each data point, interaction, and outcome, the cognitive systems develop and increase expertise, and continue to learn, adapt, and improve their expertise.
- ▶ **Interact:** With abilities to see, talk, and hear, cognitive systems interact with humans in a natural way.

### 1.3.1 Solving real life problems with cognitive systems

Cognitive systems drive the use of big data to support business processes. Most big data has no formal organization or structure. Cognitive systems can penetrate the complexity of unstructured data and incorporate the power of natural language processing and machine learning. Cognitive systems create solutions for day-to-day problems.

Cognitive systems create new ways of generating value for consumers and enhance the experience across the purchase lifecycle. For example, a cognitive travel planner can consider language identification, tradeoff analytics, and personality insights to make travel recommendations that best meet customer needs. Another example is the review of massive numbers of insurance policies to obtain policy rules. With these rules, an insurance company can drive standardization, reduce risk, and more broadly learn from the expertise and experience of the underwriters.

Vendors of cognitive systems provide a various offerings that are based on voice commands and the use of data from the Internet. Various vendors provide targeted cognitive systems for industries from healthcare to automotive to finance to insurance.

## 1.4 References

See the following resources:

- ▶ AI and cognitive computing:

<http://research.ibm.com/cognitive-computing/>

- ▶ Computing, cognition and the future of knowing:

[https://www.research.ibm.com/software/IBMResearch/multimedia/Computing\\_Cognition\\_WhitePaper.pdf](https://www.research.ibm.com/software/IBMResearch/multimedia/Computing_Cognition_WhitePaper.pdf)



# Cognitive business and IBM Watson

A *cognitive business* is an organization that creates knowledge from data to expand virtually everyone's expertise, continually learning and adapting to outthink the needs of the market.

These three elements are the root of what becomes possible with cognitive businesses:

- ▶ Grow knowledge from data: Translate expansive, ever-growing data sets into differentiation when you act on game-changing knowledge.
- ▶ Enhance expertise: Redefine industries and professions when you make expertise a scalable resource.
- ▶ Learn and adapt: Transform how you do everything together when you can learn and adapt perpetually.

Cognitive systems are crucial to a successful enterprise. Their capabilities are the key to enabling new kinds of customer engagement, building better products and services, improving processes and operations, making data-driven decisions, and harnessing expertise.

This chapter describes the industry landscape for cognitive computing and introduces Watson, the cognitive computing offering from IBM.

The following topics are covered in this chapter:

- ▶ Landscape of cognitive computing in the industry
- ▶ Introducing IBM Watson
- ▶ References

## 2.1 Landscape of cognitive computing in the industry

Organizations are just “scratching the surface” of cognitive computing capabilities, from improving customer engagement to enhancing research capabilities. The potential value of cognitive computing is boundless. Cognitive systems have three broad capability areas (Figure 2-1).

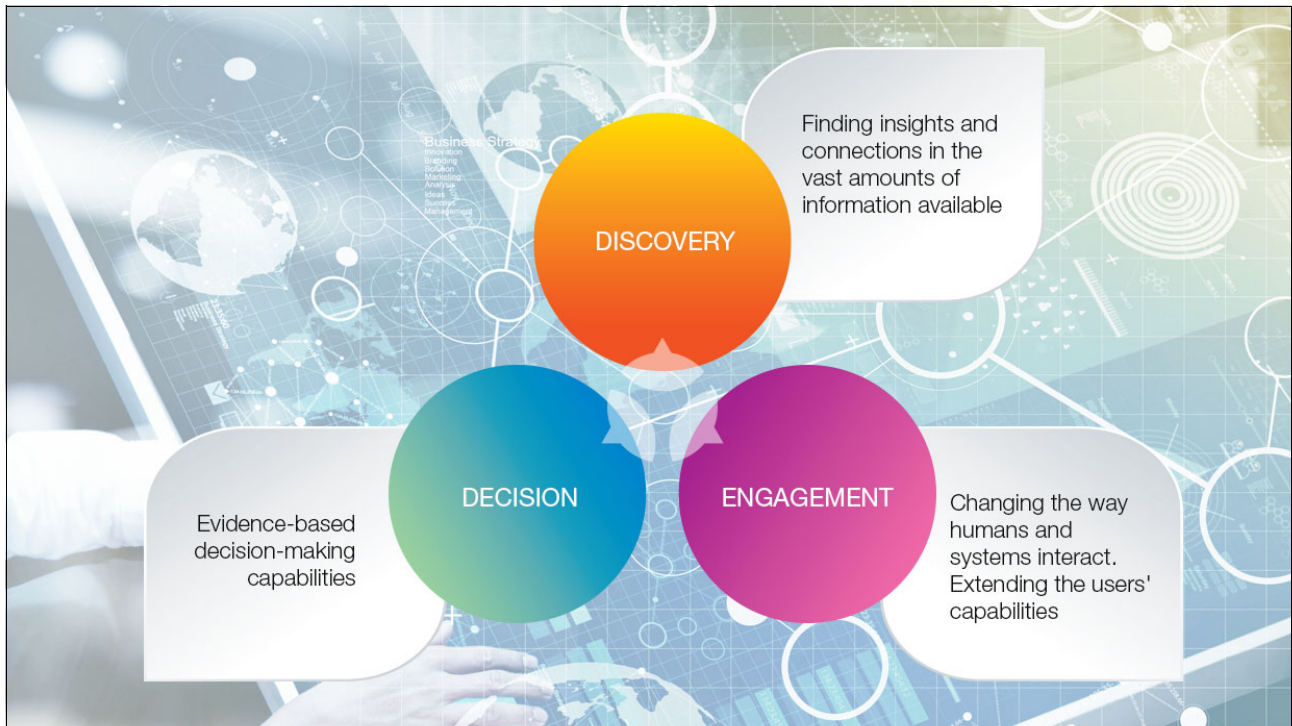


Figure 2-1 Broad capability areas of cognitive systems

The three broad capability areas are as follows:

- ▶ **Discovery:** Finding insights and connections in the vast amounts of available information
- ▶ **Engagement:** Changing the way humans and systems interact, extending the user's capabilities
- ▶ **Decision:** Evidence-based decision-making capabilities

Those capability areas relate to the ways people think and work and drive the use of big data in business processes. Most big data is unstructured data, having no formal organization or structure (Figure 2-2 on page 13).

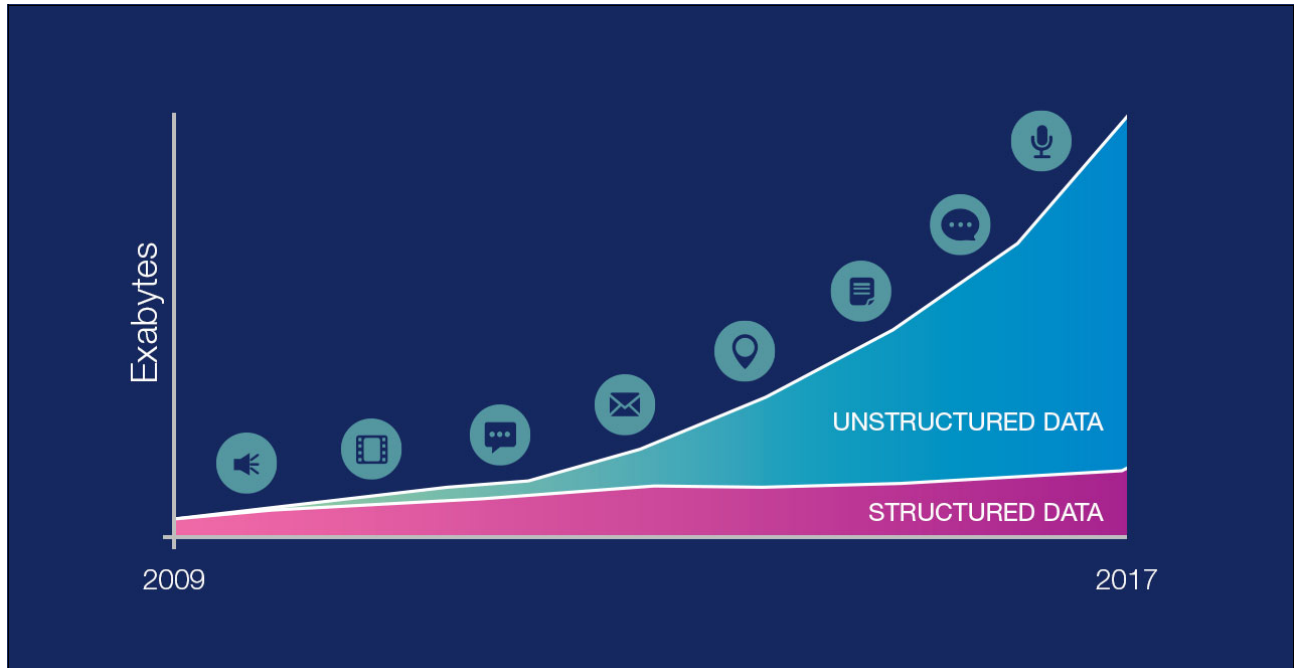


Figure 2-2 Exponential growth of unstructured data drives the need for a new kind of computer systems (cognitive)

Cognitive computing can penetrate the complexity of unstructured data and raise the power of natural language processing and machine learning.

From simple problem solving (for example, voice assistance and language translation) to complex challenges (such as a digital self-service system or healthcare solutions), cognitive computing creates solutions for the day-to-day problems that both organizations and users face.

Vendors that provide cognitive computing offerings take different approaches. For example, some vendors market cognitive appliances to target users (*consumer market*); others develop enterprise cognitive solutions to target businesses (*enterprise market*).

## 2.1.1 Consumer market: Cognitive computing offerings

The *consumer* cognitive market is targeted to users. Cognitive technologies generate new ways of creating value for consumers and enhance their daily experiences. Consumer cognitive products include personal assistants, wearable devices, home automation and more. These products often respond to voice commands and use data from the Internet, personal email, calendar, contacts, and devices that the owner uses to perform the tasks (Figure 2-3.)



Figure 2-3 Cognitive consumers products

Most consumer offerings are based on voice commands and the use of data from the Internet. A cognitive travel planner, for example, can include capabilities such as language identification, tradeoff analytics, and personality insights to make travel recommendations that best meet the customer's needs.

Cognitive technologies can give makers of household goods new ways to create value for consumers. These technologies, derived from the field of artificial intelligence, can enhance the consumer experience across the purchase life cycle, from pre-store planning through the in-store experience, product usage, and post-purchase interaction.

Upon closer examination at what is going on in the consumer space, those products provide customers with services that have access to specific knowledge domains or functions. Widely known examples are as follows:

- ▶ Amazon Echo: A hands-free speaker you control with your voice. Echo connects to the Alexa Voice Service to instantly play music and provide information, news, sports scores, weather, and more.
- ▶ Apple Siri: An intelligent personal assistant and knowledge navigator that lets you use your voice to do a variety of activities such as send messages, make appointments, control the apps in your phone, and update your calendar and to-do list.
- ▶ Google Search: A web search engine that hunts (according to your typed in request) for text in publicly accessible documents located on web servers. The types of information available include books and book reviews, synonyms for words, weather forecasts, time zones and world clock, Google Maps, movie information, airport and airline information, real estate listings, sports scores, and more.

## 2.1.2 Enterprise market: Cognitive computing offerings

Large vendors that offer enterprise cognitive services include a broad range of solutions and tools in their portfolio. Most of them deliver their services and tools on the Internet, taking advantage of the benefits of cloud platforms, such as scalability, accessibility, and flexible billing options.

Also, hundreds of small startups apply artificial intelligence algorithms across industries, from healthcare to automobile to finance to insurance.

Vendors that provide enterprise cognitive services focus on either core cognitive capabilities or they are applying artificial intelligence (AI) algorithms to specific industry solutions.

Natural language processing (NLP), machine learning (ML), and question-answering (QA) technologies provide the foundation for several core cognitive capabilities that are provided by enterprise cognitive services. These core capabilities are used by developers to build cognitive solutions. Core cognitive services available on the market include these examples:

- ▶ Conversation services
- ▶ Text mining, information extraction, and text analytics
- ▶ Machine translation
- ▶ Computer vision and image recognition
- ▶ Speech recognition



Several vendors provide cognitive solutions for industry vertical sectors, as in these examples:

▶ Agriculture

In the agriculture industry, cognitive technologies are used in applications such as crop monitoring, automated irrigation systems, automated harvesting systems and AI guided drone systems.

▶ Banking, Financial services and Insurance (BFSI)

In the BFSI sector, AI technologies are used for wealth management applications such as smart wallet, stock trading, fraud detection, and others.

▶ Manufacturing

In the manufacturing industry, cognitive technologies are using robot-integrated Computer Integrated Manufacturing (CIM) and sensor-assisted machining.

▶ Healthcare

In healthcare applications, cognitive technologies are used in a plethora of applications to reduce drug discovery times, provide virtual assistance to patients, and diagnose ailments by processing medical images, among many others.

▶ Oil and Gas

In the Oil and Gas industry, cognitive technologies are used in the exploration and production (E&P) lifecycle and drill floor automation.

▶ Media and Advertising

Some of the core applications of cognitive computing in this industry are facial recognition, advertising, and customer self-service.

▶ Transportation and Automotive

In this sector, some cognitive applications include these examples:

- Different modes of transport and their interactions
- Intelligent and real-time traffic management and control
- Transport policy, planning, design, and management
- Environmental issues, road pricing, security and safety
- Transport systems operation
- Travel demand analysis, prediction, and transport marketing
- Traveller information systems and services
- Pedestrian and crowd simulation and analysis
- Autonomous driving
- Artificial transportation systems and simulation
- Surveillance and monitoring systems for transportation and pedestrians

### 2.1.3 Delivering cognitive services: Cloud and open source projects

All the big players in the cognitive services market deliver their services and tools on the Internet over *cloud platforms*, as in these examples:

- ▶ IBM delivers Watson cognitive services over IBM Bluemix.
- ▶ Amazon AI services are delivered over Amazon Web Services (AWS).
- ▶ Microsoft AI tools are available over the MS Azure cloud.
- ▶ Google AI services are available in the Google Cloud Platform.

These services enjoy the benefits of cloud platforms, such as availability, scalability, accessibility, rapid deployment, flexible billing options, simpler operations, and management.

Various frameworks of artificial intelligence are delivered under *open source projects*, an environment for quickly creating scalable working machine learning applications. Open source software that is at no cost or at a lower cost than proprietary software provides greater flexibility than commercial software, and it can be modified to meet specific needs. However, most interfaces are not user-friendly and not easy to use. Sometimes, open source requires a steep learning curve that can slow development and deployment. Users are responsible for managing the hardware and software, which can be time-consuming, specially when compared with the cloud delivery model.

### 2.1.4 Cognitive computing and the future of jobs

Due to technological disruptions, high growth is expected for traditional IT and mathematical-based jobs, centered on data analysts and software developers. The same applies across many industries, like financial services, media, entertainment, and professional services, as computing power and big data analytics significantly drive employment growth in each area.

A new generation of skilled developers who understand cognitive computing will be in high demand to build creative solutions for new businesses that do not even exist today. With the wealth of big data and the need for more complex evidence-based decisions, the conventional approaches break or simply fail to keep up with all the available information. Cognitive computing enables people to create a profoundly new kind of value finding answers and insights locked away in volumes of data.

## 2.2 Introducing IBM Watson

IBM Watson is a cognitive system that enables a new partnership between people and computers. It is the cognitive computing offering from IBM.

Watson combines five core capabilities:

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

IBM Watson is at the forefront of a new era of computing: cognitive computing. In summary, Watson can understand all forms of data, interact naturally with people, and learn and reason, at scale.

Data, information, and expertise create the foundation for working with Watson. Figure 2-4 shows examples of data and information that Watson can analyze and learn from, and derive new insights that were never discovered before.

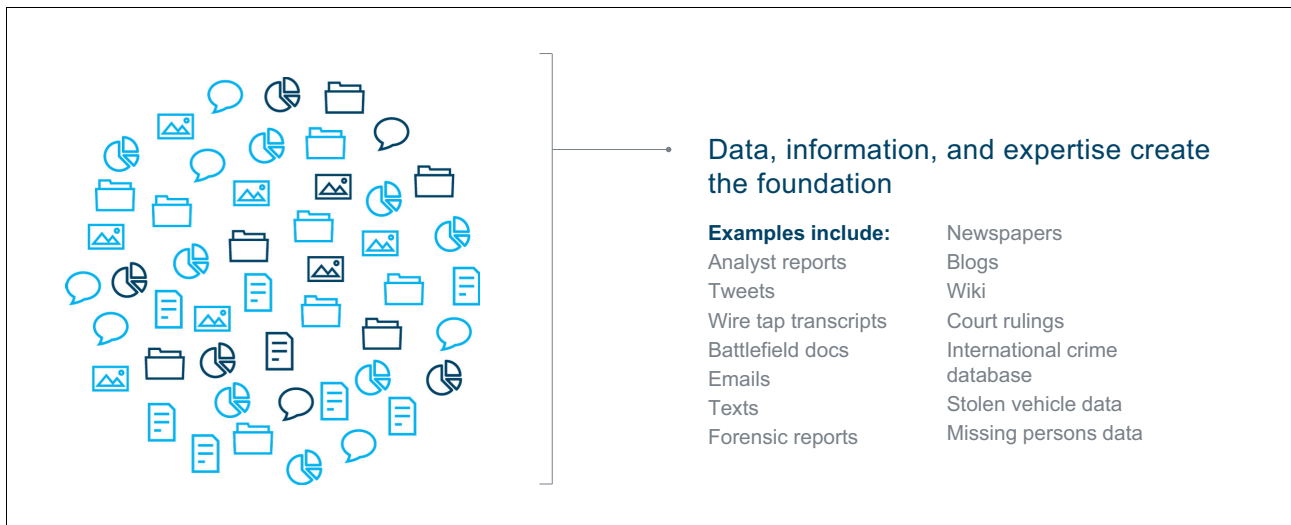


Figure 2-4 Watson relies on collections of data and information

Watson is available as a set of open application programming interfaces (APIs) and *software as a service (SaaS)* industry solutions.

## 2.2.1 Watson APIs: Build with Watson

You can enable cognitive computing features in your applications by using IBM Watson Language, Vision, Speech, and Data APIs. Watson APIs are delivered through IBM Bluemix, which is the *cloud platform as a service (PaaS)* developed by IBM.

The following Watson APIs are currently available:

- ▶ Language:
  - Conversation
  - Document Conversion
  - Language Translator
  - Natural Language Classifier
  - Natural Language Understanding
  - Personality Insights
  - Retrieve and Rank
  - Tone Analyzer
- ▶ Speech:
  - Speech to Text
  - Text to Speech
- ▶ Vision:
  - Visual Recognition
- ▶ Data Insights:
  - Discovery
  - Discovery News

Figure 2-5 shows a view of Watson services in the IBM Bluemix catalog.

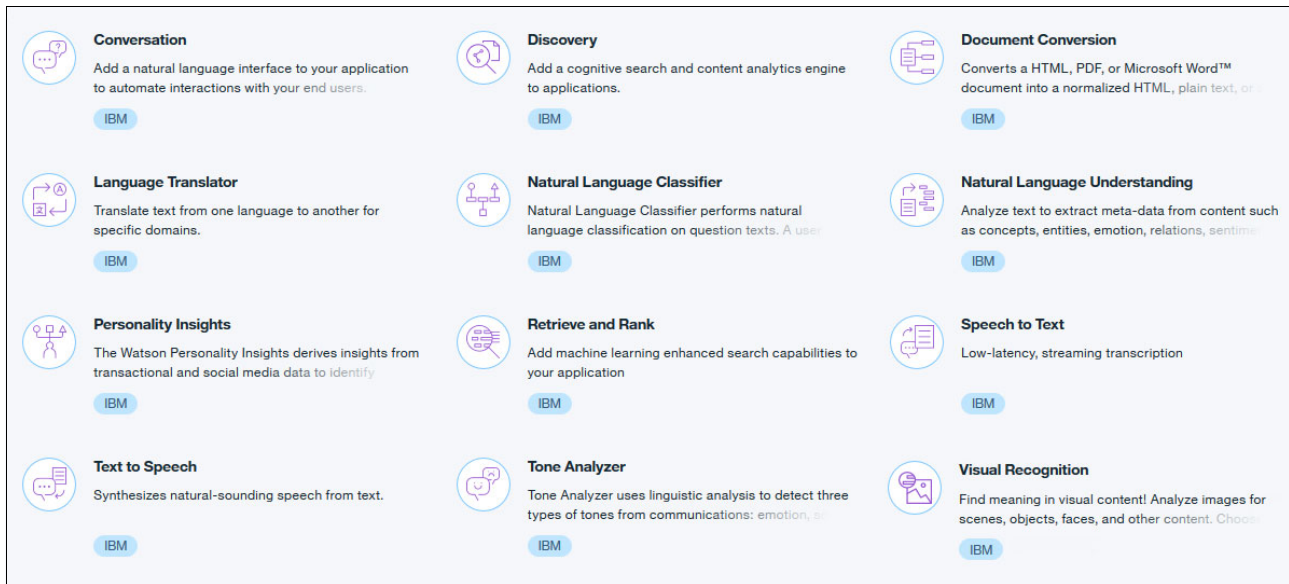


Figure 2-5 Watson services in IBM Bluemix catalog

## 2.2.2 IBM Watson applied to industries, businesses, and science

Today, industries transform their businesses by using data. This way businesses and organizations can predict with greater precision what their customers will want, where traffic congestions will occur, or how diseases will progress.

With cognitive new offerings, IBM is enabling organizations across all industries and of all sizes to integrate new cognitive capabilities into their businesses.

More than 9 billion connected devices operate in the world today, and they generate 2.5 quintillion bytes of new data daily. Making sense of data embedded in intelligent devices is creating a significant market opportunity that is expected to reach \$1.7 trillion by 2020<sup>1</sup>.

IBM Watson offerings apply to industries, businesses, and science (Figure 2-6 on page 19.)

<sup>1</sup> IBM Opens Watson IoT Global Headquarters, Extends Power of Cognitive Computing to a Connected World: <https://www.ibm.com/press/us/en/pressrelease/48443.wss>



Figure 2-6 Cognitive computer applies to industry, businesses, and science

Cognitive computing is being put to work across various industries, helping scientists discover promising treatment pathways or helping law experts discover connections between cases.

Watson has industry expertise to solve your toughest challenges. For more information, see [Watson products and APIs](#).

The following list describes areas where Watson is applied to solve real problems in several industries, businesses, and science:

► IBM Watson Commerce

The IBM paper *Commerce in the Cognitive Era: Unleash the power of Watson Commerce to deepen customer engagement, drive brand loyalty and fuel growth* states: “Watson Commerce combines business expertise with industry-leading solutions embedded with cognitive capabilities, giving commerce professionals the power to create consistent, precise, personalized experiences that customers want and value. Watson Commerce understands, reasons and learns from the collective knowledge of the organization and business trends. Brands gain immediate insight to customer behavior and business performance and can make timely, informed decisions and measurable actions to capitalize on market opportunities before their competitors do.”

For more details, read the full IBM [Watson Commerce Point of View](#).

► IBM Watson Education

With Watson Education, educators and students together can transform the individualized learning experience throughout their lifelong learning journey. Solutions and apps that are infused with analytics and cognitive capabilities help teachers learn about their students holistically. They can shape and drive personalized learning suited to each person, assisted by technology that understands, reasons, learns, and interacts. Students are empowered with “unique to me” constructs that make learning easier and more meaningful. Educators and students build relationships and hubs of collaboration, exchanging and growing expertise to create new possibilities in education that help shape a better future for everyone. See [Transform learning experiences with Watson](#).

► IBM Watson Financial Services

Use the cognitive power of Watson to drive deeper consumer engagement and new experiences, and augment the management of regulatory compliance. Adding Watson cognitive capabilities allows you to go beyond traditional rules-based policy and demographic views to a deeper understanding of customer profitability, preferences, and lifecycle needs so that you can offer new, more personalized offerings and experiences. Watson can also help you transform your approach to managing risk and compliance so that you can stay ahead of an ever-changing regulatory environment. See [Watson Financial Services](#).

► IBM Watson Health

Watson Health cognitive systems understand, reason, and learn, helping to translate information into knowledge to help drive more informed decision-making.

Science is saturated with data. By making connections that might not have been previously considered, Watson can generate new insights that expedite the process of matching subjects to clinical studies, identify promising targets for research, and encourage discovery.

IBM Watson Health solutions can help in the following areas:

- Optimize performance: Drive transformation in your organization.
- Engage consumers: Enhance your patients' chance of success.
- Enable effective care: Support your care team with data and insights to enhance their decision-making.
- Manage population health: Improve patient experiences while helping to reduce cost.

IBM and IBM Business Partners are building solutions for individual patients and larger health populations to benefit as providers share and apply insights in real time.

See the following videos:

- [Introducing IBM Watson Health](#) provides an overview of this offering.
- [Collaboration to Advance Genomic Medicine](#) announces the initiative between the New York Genome Center (NYGC) and IBM to accelerate a new era of genomic medicine with the use of IBM Watson cognitive system.
- [Search for ALS treatments](#) shows how Barrow Neurological Institute is using Watson to help focus its research efforts toward the most promising directions.

Also see [IBM Watson Health](#).

Watson Health provides these specific solutions:

– IBM Watson for Genomics

On average, 75% of cancer patients will not respond to a particular drug in a class of agents. Oncologists are increasingly looking to genomics insights to identify more precise and potentially effective therapies. Now, clinicians across the US can provide precision medicine to cancer patients. See how [Watson for Genomics helps doctors give patients new hope](#) by helping doctor confidence in personalized treatment approaches.

– IBM Watson for Drug Discovery

Watson for Drug Discovery helps researchers identify novel drug targets and new indications for existing drugs. The platform can help researchers uncover new connections and develop new treatments which may lead to new insights and scientific breakthroughs ahead of the competition. See [IBM Watson for Drug Discovery](#).



- IBM Watson Health Patient Engagement

Watson Health Patient Engagement helps identify patients with care gaps and automate personalized interventions, keeping patients engaged and helping them manage their own care between visits. See [Engage patients and consumers](#).

- IBM Watson for Oncology

The amount of research and data available to help inform cancer treatments is growing exponentially. Yet the time care teams have to consume this information, locating insights specific to each patient's unique needs to potentially improve treatment outcomes, is more limited than ever. Watson for Oncology helps physicians quickly identify key information in a patient's medical record, surface relevant articles and explore treatment options to reduce unwanted variation of care.

With Watson for oncology your organization can spend less time searching literature and more time caring for patients. Watson can provide clinicians with evidence-based treatment options based on expert training by Memorial Sloan Kettering physicians. See [Watson for Oncology](#).

- IBM Watson Care Manager

Watson Care Manager provides personalized care plans, automated care management workflows, and integrated patient engagement capabilities to help create more informed action plans. See [Watson Care Manager](#).

- ▶ Watson in the Insurance industry

Insurance regulations and policies are in constant flux. Handling claims requires the intuition of highly skilled assessors, who have to review hundreds of pages of texts, handwritten notes, blogs, and various other sources to keep up with regulation changes and make consistent decisions. Adding to its challenge, the degree of variation in member coverage makes adding help, by training new employees, more difficult.

Insurance companies are teaching Watson to understand the interactions, rules, and processing logic that can apply to policies. Watson is able to analyze structured and unstructured data, reference the right policy information and input documents, and then make insightful recommendations; this can help employees determine whether a claim is eligible and what percentage of the claim should be paid. With Watson, employees can make better decisions and get better results faster.

As the insurance market becomes digital, digitally savvy customers expect insurers to know them in advance when they call, proactively engage with them, and provide them with proactive advice and guidance. Cognitive computing is an opportunity to get closer to customers and generate better understanding, from risk analysis to fraud to security.

See [Employees are making better decisions, faster](#).

- ▶ IBM Watson Internet of Things (IoT)

Internet of Things is changing the way that businesses operate and people interact with the physical world. A cognitive IoT can make sense of all types of data. It can choose its own data sources and decide which patterns and relationships to pay attention to. It uses machine learning and advanced processing to organize the data and generate insights. A cognitive IoT can also evolve and improve on its own through learned self-correction and adaptation.

See the following videos:

- [How it Works: Internet of Things](#) shows how IoT gives us access to the data from millions of devices.
- [Industry 4.0 meets Watson IoT](#) shows how IBM Watson IoT™ is helping companies to connect with everything from personal appliances to manufacturing equipment.

► IBM Watson Cognitive Video

IBM Watson produces videos, applying its visual recognition and tone analyzer capabilities. It analyzes and learns from other videos in order to create new compositions based on the user's requirements.

Visual Recognition understands the content of images: visual concepts tag the image, find human faces, determine approximate age and gender, and find similar images in their collection.

The [IBM Watson Visual Recognition](#) video describes visual recognition and how it works.

Using experimental Watson APIs and machine learning techniques, scientists at IBM Research in collaboration with 20th Century Fox created the first cognitive movie trailer for the movie "Morgan." The system analyzed hundreds of horror and thriller movie trailers. After learning what keeps audiences on the edge of their seats, the cognitive system suggested the top 10 best candidate moments for a trailer from the movie, which an IBM filmmaker then edited and arranged together.

See the following resources:

- [IBM Research Takes Watson to Hollywood with the First "Cognitive Movie Trailer"](#) at the IBM THINK® Blog.
- [Morgan movie trailer](#) video to learn about the collaboration between IBM Research and 20th Century Fox to create the cognitive movie trailer.

► IBM Watson for Cyber Security

Cognitive systems can ease the security analyst's work by providing human-centric communications, such as advanced visualizations and interactive vulnerability analysis. Cognitive systems can spot anomalies and flawed logic and provide evidence-based reasoning.

Cognitive systems shine a light into data that was previously dark to organizational defenses, to uncover new insights, patterns, and previously unseen security contexts. Cognitive systems interpret data, add to their knowledge base from virtually every interaction, weigh probabilities based on a depth of insight, and consider relevant variables to help you take action. Consequently, cognitive security helps to reduce the cost and complexity of dealing with cybercrime. See these videos:

- [Watson for Cyber Security in Action](#) describes how Watson helps a security analyst investigate a particular incident to uncover new patterns and security context never before seen.
- [Teaching Watson the Language of Security](#) illustrates the process of training Watson for Cyber Security by defining the language of security, annotating representative documents, and then manually making correlations between terms. Watson can then, on its own, learn the language and connections needed to respond to questions from security professionals.

For more information, see [Watson for Cyber Security](#).



## 2.2.3 Watson use cases

Examples of how Watson is used in several industries, science, and applications to solve real problems are listed in 2.2.2, “IBM Watson applied to industries, businesses, and science” on page 18.

This section presents two use cases showing organizations that successfully implemented cognitive solutions, based on IBM Watson technology, to enable new kinds of customer engagement, build better products and services, improve processes and operations, make data-driven decisions, and harness expertise.

### **OmniEarth: Cognitive computing shows water consumption patterns from Earth imagery**

OmniEarth Inc. builds scalable solutions for processing, clarifying, and fusing large amounts of satellite and aerial imagery with other data sets.

#### ***The challenge***

Water conservation is a top concern in California, which is susceptible to drought conditions. In April 2015, the state mandated a 25% reduction in urban water consumption over a period of 10 months. It was necessary to understand patterns of water usage, how they relate to the weather, and how to set realistic water budgets.

For the state’s water utilities, achieving this goal requires more than just asking people to use less water. The utilities needed a firm grasp of water usage patterns, where water was being overused, how much water could be saved and through which actions, and where to target consumer outreach and education efforts.

Traditionally, state governments arrive at these statistics by calculating averages across a one-year or multiple-year period, which disregards variations in microclimates and other local circumstances.

By applying its proprietary algorithms and aerial images, OmniEarth was confident it could help the state understand water consumption at an unprecedented, granular level. However, the company needed a solution that would scale its ability to process a large number of images, unlocking the unstructured data within and making it available for analysis.

#### ***The solution***

OmniEarth uses IBM Watson to identify topographical features in satellite images, giving water districts insight into dynamic patterns of water consumption and weather. OmniEarth is using the solution to develop water conservation strategies for drought-stricken areas. OmniEarth is helping water utilities within the State of California to analyze aerial images to monitor water consumption on each parcel of land across the state.

Because a cognitive system can process many more images than a manual process, the solution produces a view of water consumption that is more like a moving picture of the landscape than isolated snapshots. The system is trained to identify important features that signal water usage such as swimming pools, turf lawns, irrigated areas and agricultural zones. It automatically translates unstructured images into structured data that OmniEarth’s proprietary algorithms can analyze.

The system combines aerial images with weather and water consumption data to identify parcels of land that exceed their usage limits and focus their outreach, helping California’s water utilities educate identified consumers about how specific behaviors can yield cost savings and environmental improvements. The granular, localized information also helps

individual water utilities determine realistic water budgets and demand forecasts instead of relying on blanket restrictions based on sweeping statewide averages.

OmniEarth is using IBM Watson to develop water conservation strategies for drought-stricken areas. OmniEarth is helping water utilities within the State of California to analyze aerial images to monitor water consumption on each parcel of land across the state.

OmniEarth uses the IBM Watson Visual Recognition service through IBM Watson Developer Cloud to classify the physical features that are captured in aerial and satellite images (Figure 2-7) and integrates the resulting data with its proprietary analysis tools.



*Figure 2-7 Analyzing the terrain parcel-by-parcel and surfacing insights locked in millions of unstructured images*

### ***Benefits***

The solution enables OmniEarth to process aerial images 40 times faster than was possible with the previous manual methods, with the ability to process 150,000 images in just 12 minutes rather than several hours. The solution also gives OmniEarth significantly greater capacity for analyzing terrain on a massive scale, creating new business opportunities all over the world. Plus, the cognitive technology gives the organization and the State of California deeper insights into satellite imagery, with the ability to analyze the data at a more granular level.

Insight derived from data has the power not only to make regulatory response smarter, but also to improve demand forecasting and other important drought-time decision-making.

### ***What makes the solution cognitive***

Here are several important factors:

- ▶ **Game-changing outcome**

By teaching a cognitive system to recognize visual cues in aerial images, OmniEarth unlocked crucial information and made it available for sophisticated analysis.

- ▶ **Before-after impact**

In the past, OmniEarth had to interpret and tag aerial images in a slow and manual process, taking hours to produce a single batch. By teaching a cognitive system to interpret the images, the company dramatically sped up the process and made it possible to analyze more data more frequently and understand the earth's landscape at a more granular level.

- ▶ **Systems of cognitive discovery, engagement, or decision-making**

The solution provides a system of cognitive decision-making, helping the State of California to extract patterns from images, with enough detail to focus on individual parcels of land and quickly enough that the insights can inform day-to-day actions.

For more information, read this article: [Now you can see the potential hidden in millions of online images.](#)

### **Woodside Energy: Employees instantly access 30 years of experience**

Woodside Energy is Australia's largest publicly traded oil and gas exploration and production company and one of the nation's most successful explorers, developers, and producers of oil and gas.

The article, [How can every Woodside employee instantly access 30 years of experience?](#), describes how Watson on IBM Cloud helps Woodside Energy employees access information.

#### ***The challenge***

Some Woodside employees are based at off-shore facilities for two-week deployments. Their jobs require extreme precision, and conditions must be perfect before any action can be taken. Woodside employees consider every element from weather and wind, to tidal currents, to animal migration patterns. They rely on historical context and procedural information to adapt. Therefore, they needed a way to dramatically simplify the way technical and engineering staff could locate, analyze and learn from the existing body of subject matter knowledge throughout the company.

By discovering information about a particular technical issue, such as how often to maintain certain operating equipment, Woodside could make better decisions while preventing "reinventing the wheel" in expensive and time-consuming tasks.

#### ***The solution***

Working with Watson, Woodside Energy built a customized tool that allowed its employees to find detailed answers to highly specific questions, even on remote oil and gas facilities. Watson ingested the equivalent of 38,000 Woodside documents, which would take a human over five years to read.

This corpus of knowledge evolved into Woodside's cognitive solution powered by Watson on IBM Cloud. Woodside employees can ask questions in natural language, like "What is the maximum weight of a helicopter landing on the platform?" Watson will respond accordingly.

Woodside Energy worked with Watson on IBM Cloud, following five simple steps:

1. Over 38,000 Woodside documents were loaded into Watson on IBM Cloud, the equivalent of 30 years of practical engineering experience.
2. With this data, Watson considers historical context and procedural information on operations, equipment, weather, tidal currents and more.
3. Employees ask Watson a question in natural language, such as “What is the maximum weight of a helicopter landing on the platform?”
4. Watson can find results from previous tests and recommend a course of action.
5. Employees can generate insights and make physical adjustments based on the information.

### ***Benefits***

Now Woodside employees anywhere in the world can access 30 years of expertise and locate technical data to make quicker, smarter, more fact-based decisions.

Watson helped reduce the time spent searching for expert knowledge by 75%. In a high-risk industry, every action on an offshore platform costs time and money. With Watson on IBM Cloud, Woodside can get a return on their investment and help keep employees safe.

### ***What makes the solution cognitive***

The following Watson APIs are used in this solution:

- ▶ Natural Language Classifier
- ▶ Conversation
- ▶ Retrieve and Rank

Watson ingested Woodside’s entire 30-year base of 38,000 engineering documents and keeps learning with new experiences. Employees communicate with Watson in natural language. From the answers, employees can derive new insights and make better decisions.

## **2.2.4 Watson demonstrations**

Want to try out Watson? Here are some demonstrations that give you a chance to interact with applications built on Watson services.

### **Your Celebrity Match application**

This demonstration uses Watson Personality Insights and Insights for Twitter services:

<https://your-celebrity-match.mybluemix.net>

Enter your Twitter handle to see your personality traits, rated, and see which celebrities have personalities that are most similar to and different from yours.

### **Audio Analysis application**

This demonstration uses the Concepts feature of Watson Natural Language Understanding service coupled with the Speech to Text service in order to provide analysis of the concepts that appear in YouTube videos:

<https://audio-analysis-starter-kit.mybluemix.net/>

## Conversation service

With the Watson Conversation service allows you to understand what users are saying and to respond with natural language:

<https://conversation-demo.mybluemix.net/>

In this demonstration, imagine you are in the driver's seat and Watson is your co-pilot. Watson can understand your commands and respond accordingly. For example, try asking "where is the nearest restaurant" or say "turn on the lights" to see how Watson understands your commands.

This demonstration is trained on a specific set of car capabilities. Click the **What can I ask** button in the demonstration, to display the list of topics that Watson understands (Figure 2-8).

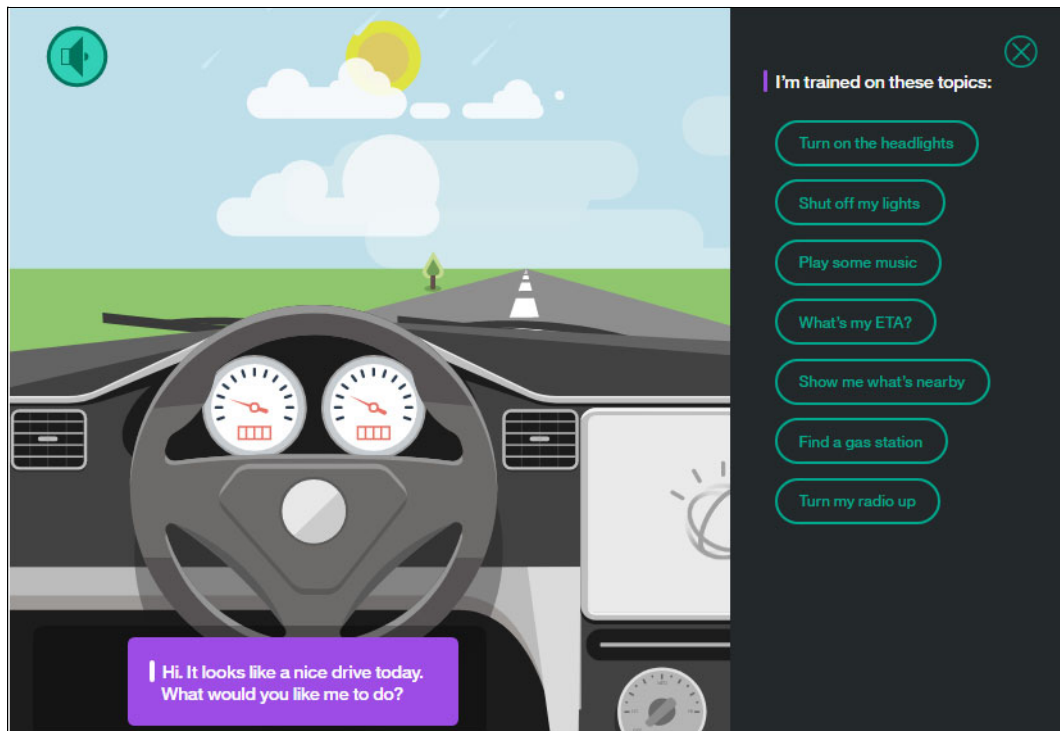


Figure 2-8 Watson Conversation services demonstration commands

## Visual Recognition service

Visual Recognition uses deep learning algorithms to analyze images that can give you insights into your visual content. You can organize image libraries, understand an individual image, and create custom classifiers for specific results that are tailored to your needs:

<https://visual-recognition-demo.mybluemix.net/>

This demonstration has two options:

- ▶ Try  
Try classifies an image. You select either an image that is shown or provide a web address of the image (image URL). Visual recognition will analyze your selection (with the classifiers available) to classify the image and provide you with rated classification results.
- ▶ Train  
Train is used to create a demonstration classifier. To create a temporary trial classifier, select at least three classes from the example image bundles or provide your own image bundles (at least two positive image bundles and if helpful one negative bundle) as specified. Select **Train your classifier**. When the classification is complete, you can use the classifier against images through the Try capability.

## 2.3 References

For more information, see the following resources:

- ▶ IBM Watson: How it works:  
[https://youtu.be/\\_Xcmh1LQB9I](https://youtu.be/_Xcmh1LQB9I)
- ▶ IBM Watson Health: How it works:  
[https://youtu.be/ZPXCf5e1\\_HI](https://youtu.be/ZPXCf5e1_HI)
- ▶ What's Watson working on today?  
<https://www.ibm.com/watson/stories/>



# Introduction to question-answering systems

In 2007, IBM Research took on the grand challenge of building a computer system that could compete with champions at the game of Jeopardy!, a US nationally televised quiz show. In 2011, the open-domain question-answering system, named *Watson*, beat the two highest ranked players in a two-game Jeopardy! match.

Advances in question-answering (QA) technology can help support professionals with critical and timely decision-making in areas such as compliance, healthcare, business integrity, business intelligence, knowledge discovery, enterprise knowledge management, security, and customer support.

This chapter describes the nature of the question-answering challenge represented by Jeopardy!. It provides a high-level overview of the QA system architecture, which was developed for Watson to play the game and which is known as the *DeepQA* architecture.

The following topics are covered in this chapter:

- ▶ The Jeopardy! challenge
- ▶ DeepQA system architecture
- ▶ Exploring the DeepQA pipeline through an example
- ▶ References

## 3.1 The Jeopardy! challenge

The Jeopardy! television quiz show from the US aired for the first time in 1964. The show features three contestants who solve general knowledge clues. The question-and-answer format sets Jeopardy! apart from other trivia games.

For example, a standard trivia question-and-answer format asks the question in the form of a question and provides the answer in the form of a statement:

**Question:** Who defeated Spassky in the World Chess Championship 1972?  
**Answer:** Bobby Fisher.

In Jeopardy!, questions (or clues) are framed as statements, but the answers must be phrased as questions:

**Question (clue):** He defeated Spassky in the World Chess Championship 1972.  
**Answer:** Who is Bobby Fisher?

When Jeopardy! begins, the game board presents six categories. Each category has five questions (clues), so one round of Jeopardy! includes up to 30 questions. Each answer has an associated monetary value (US dollar). The dollar value increases as the questions become more difficult.

During the game, the host reads the clue, and the contestant who presses the buzzer first (in less than 3 seconds) gets to answer the question. If the answer is correct, the contestant accumulates the amount of money associated with that answer, and then chooses the next category to receive the next question (clue). If the contestant answers the question incorrectly, that contestant loses the corresponding amount of money and the two other contestants try to answer the question. The game continues until either no categories remain or time runs out. At the end of the show, the contestant with the highest dollar amount wins the game.

A defining moment in Jeopardy! history was the Jeopardy! and Watson match, where the IBM Watson computer took on the two most successful Jeopardy! champions (Ken Jennings and Brad Rutter).

By early 2010, Watson (in the IBM Lab) was able to beat human Jeopardy! contestants regularly. In early 2011, Watson competed against the best Jeopardy! champions (Ken and Brad). After struggling at first and getting some questions wrong, Watson started answering all the questions correctly, went on a winning streak, and ultimately won the game.

On day two of the competition, Ken was in the lead. The risk was that he could win over Watson. Ken selected an entry in the Legalese category. Watson answered the question correctly and next picked a question that contained a “Daily Double” value. A Daily Double means that if the contestant answers correctly, that contestant receives double the dollar amount added to the current earnings. Watson provided the correct answer, which put Watson far in the lead, enabling Watson to win the competition.

The Watson project was not about playing Jeopardy! It was about doing research in natural language understanding and machine learning, then taking the technology and applying it to solve problems that people really care about. That is why the Jeopardy! game was just the beginning of the IBM investment in cognitive computing. Now, Watson technology is already being applied to solve real-life problems in healthcare, medicine, technical support, finance, government, and many other industries, businesses, and science.



With the ability of Watson to sift through and more deeply analyze piles of information and deliver it on an as-needed basis, learn from previous experiences, and communicate with users in their natural language, Watson is now being used in hundreds of applications around the world.

For more information, see the [Watson and the Jeopardy! Challenge](#) video.

## 3.2 DeepQA system architecture

This section provides an overview of the question-answering (QA) system architecture that was designed for Watson to play the Jeopardy! game. This implementation is known as *DeepQA*. DeepQA is a software architecture for deep content analysis and evidence-based reasoning. It represents a powerful capability that uses advanced natural language processing (NLP), information retrieval, reasoning, and machine learning. The underlying philosophy of the research approach that led to DeepQA is that true intelligence will emerge from the development and integration of many different algorithms each looking at the data from different perspectives. The success of the Watson question-answering system can be attributed to the integration of a variety of artificial intelligence technologies.

The DeepQA architecture views the problem of automatic question-answering as a massively parallel hypothesis generation and evaluation task. DeepQA can be viewed as a system that generates a wide range of possibilities and, for each, develops a level of confidence by gathering, analyzing, and assessing evidence that is based on available data.

The primary computational principle supported by the DeepQA architecture can be summarized in the following points:

- ▶ Assume and pursue multiple interpretations of the question.
- ▶ Generate many plausible answers or hypotheses.
- ▶ Collect and evaluate many competing evidence paths that might support or refute those hypotheses.

The DeepQA architecture was designed in a way that is flexible and allows the integration of a variety of technologies, including machine learning, natural language processing, knowledge representation, reasoning, and other AI technologies.

Each component in the system adds assumptions about what the question might mean, what the answer might be, or why the answer might be correct. DeepQA is implemented on top of a framework called Unstructured Information Management Architecture (UIMA), designed to support interoperability and scale-out of deep analytics. Figure 3-1 on page 32 illustrates the DeepQA architecture at a high level. The remaining sections of this chapter provide more detail about the various architectural roles.

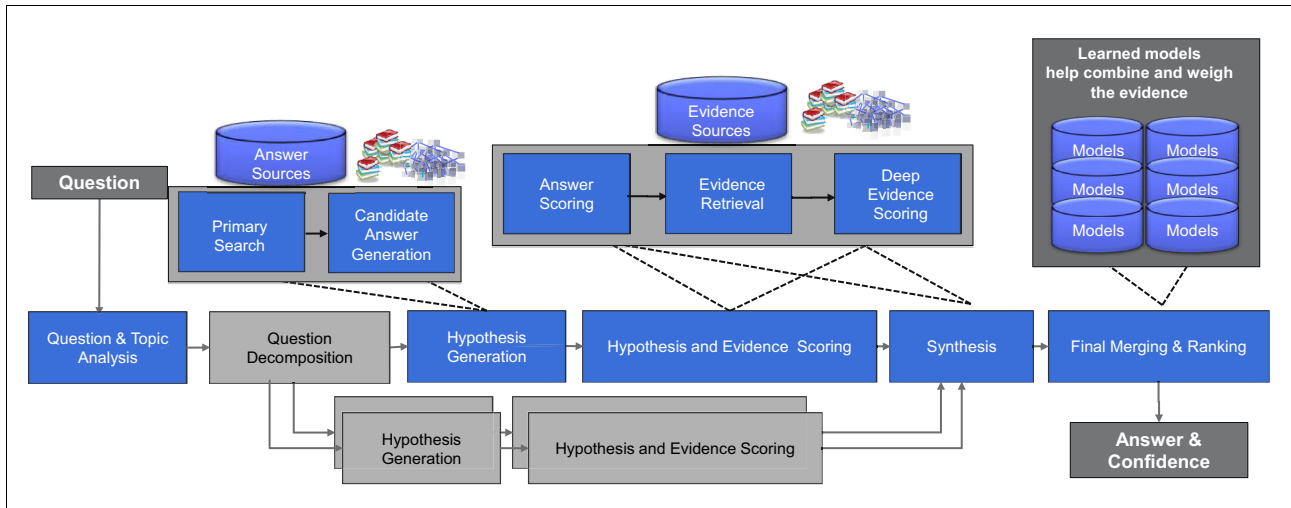


Figure 3-1 DeepQA high-level architecture

The basic assumption of the DeepQA architecture is that both answers and supporting evidence for each answer are gathered from both structured (knowledge bases) and unstructured (text) information, as illustrated in Figure 3-2. DeepQA collects hundreds of possible candidate answers (also called *hypotheses*) and, for each of them, generates evidence by using an extensible collection of natural language processing, machine learning, and reasoning algorithms. These algorithms gather and weigh evidence over both unstructured and structured content to determine the answer with the best confidence.

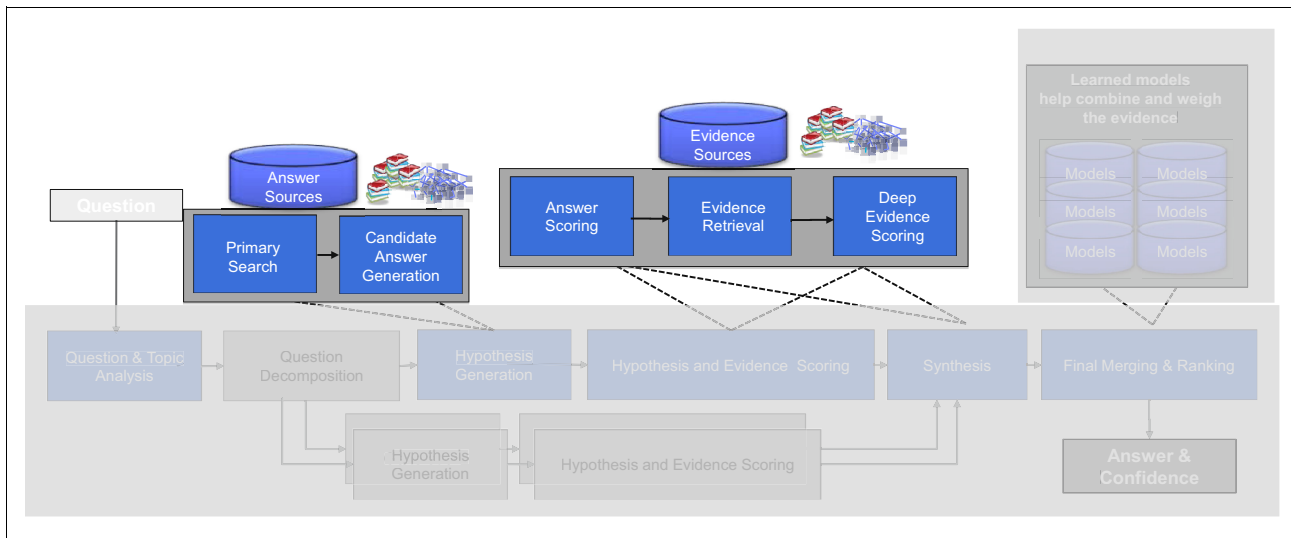


Figure 3-2 Answers and evidence sources: Structured and unstructured data

*Unstructured* data that was used by the Jeopardy! system included a wide range of encyclopedias, dictionaries, thesauri, newswire articles, literary works, text corpora derived from the web, Wikipedia, and so on. *Structured* data included databases, taxonomies, and ontologies, such as DBpedia.

After candidate answers are collected, DeepQA scores each of them and tries to determine the correct one by looking at additional evidence sources, which also come from both unstructured and structured data sources. To this aim, it uses machine learning to weigh the impact of each evidence source on the task of providing a confidence to each answer.

Training is performed by using historical data that is provided by past Jeopardy! games. The resulting models are stored and used when applied (at apply-time) to answer new questions, as illustrated in Figure 3-3.

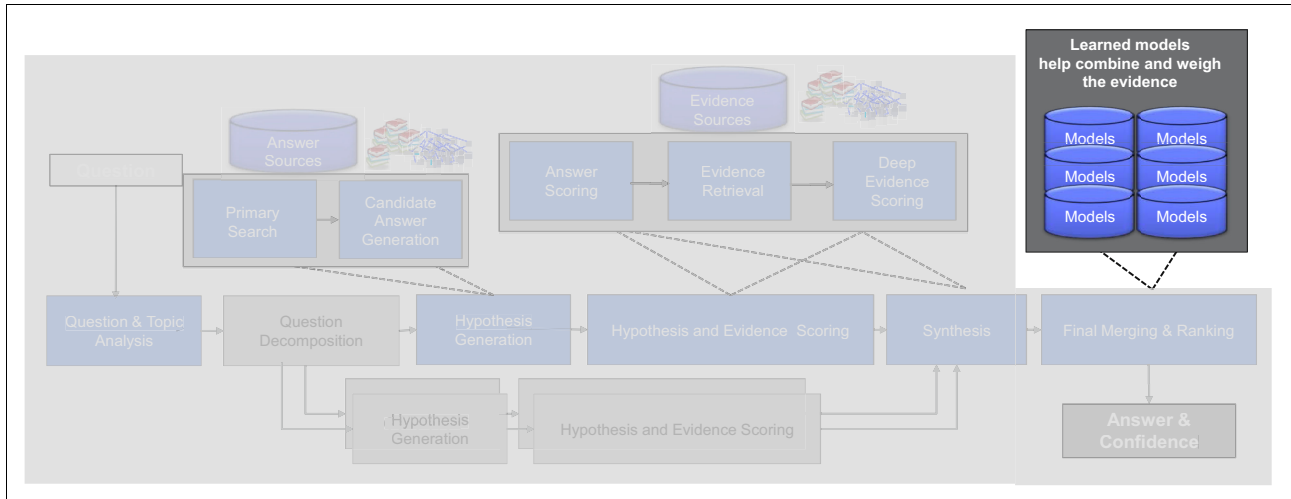


Figure 3-3 Learned models help combine and weigh the evidence

The DeepQA architecture is sophisticated, and describing all its components goes beyond the scope of this document. However, it can be illustrated by the simplified version, known as the *minimum DeepQA pipeline*, shown in Figure 3-4.

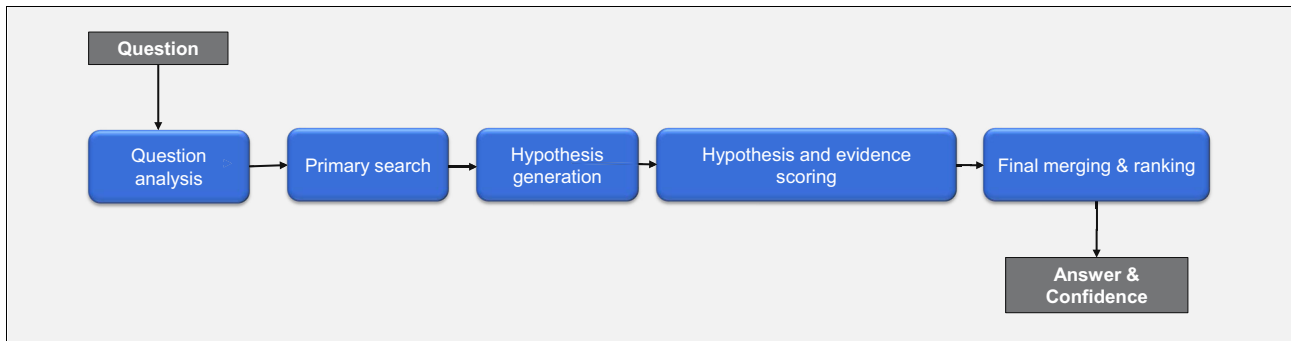


Figure 3-4 Minimum DeepQA pipeline

In spite of its simplicity, this “minimal” architecture can be used to answer factoid questions, which constitute the majority of Jeopardy! questions. The answers from factoid questions are based on factual information about one entity. The questions themselves present challenges in determining what exactly is being asked and which elements of the clue are relevant in determining the answer.

The Minimum DeepQA pipeline (Figure 3-4) receives a question as input, returns an answer and an associated confidence score as output, and includes the following components:

- ▶ Question analysis
- ▶ Primary search
- ▶ Hypothesis generation
- ▶ Hypothesis and evidence scoring
- ▶ Final merging and ranking

### 3.3 Exploring the DeepQA pipeline through an example

This section describes all components of the minimum DeepQA pipeline. The best way to explain DeepQA in action is with an example that describes how Watson would answer a Jeopardy! question.

Figure 3-5 shows the overview flow of the question-answering example described in this chapter. The starting point is the following question (clue): “In 1894, C.W. Post created his warm cereal drink Postum in this Michigan city.”

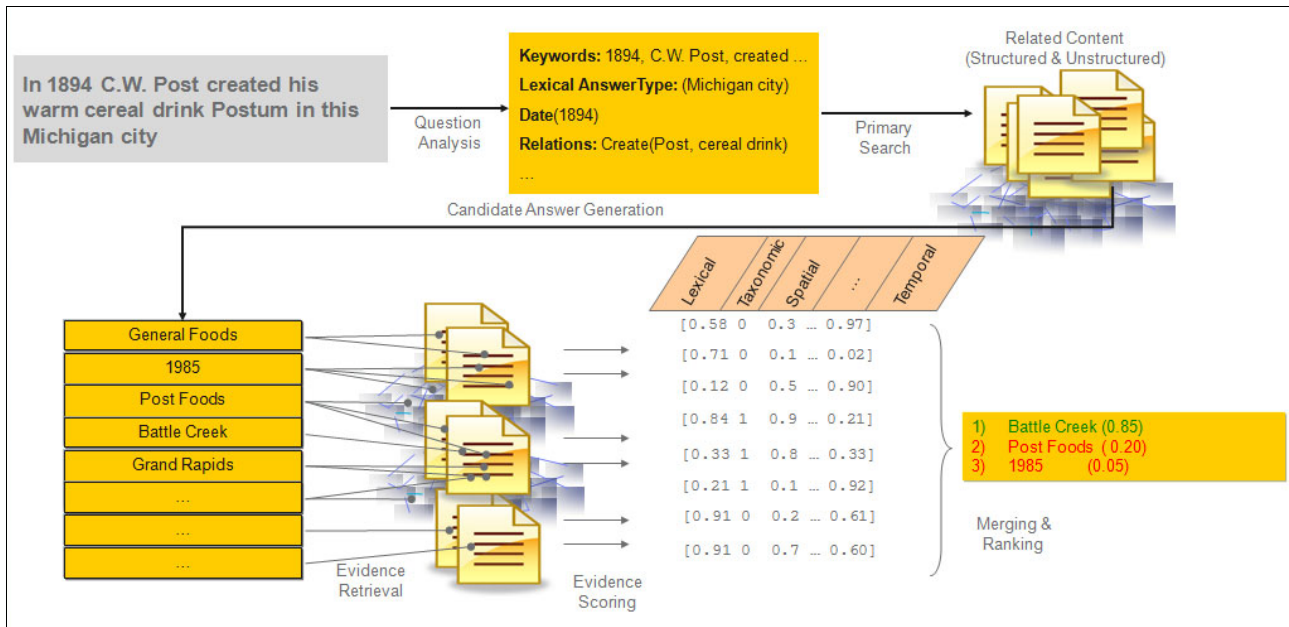


Figure 3-5 Example of how Watson answers a question following the DeepQA architecture

#### 3.3.1 Question analysis

The first task is to analyze the question. This involves parsing the question into its parts of speech and identifying the various roles that the words and phrases in the sentence are playing. This analysis helps to determine what type of question is being asked and what the question is asking for. Question analysis is mostly a natural language processing type of problem and the goal is to gain a good understanding of the question in terms of the entities involved, the relations, the possible categories of the answers, and so on.

After the clue is provided as an input, the first stage in answering it is to perform question analysis. For this stage, DeepQA uses a mix of natural language processing (NLP) analytics, including keyphrase extraction, information extraction, lexical answer type identification, and question classification.

Keyphrase extraction provides a set of keywords to be used to compose a query for primary search. In this example, the keywords identified are: 1894, C.W. Post, and created.

Information extraction (IE) is about identifying entities and relations in the clue. It is implemented by a mix of closed domain and open domain IE technologies, such as statistical and rule-based information extraction systems and semantic parsing. Jeopardy! questions pose an open-domain QA problem, therefore the information extraction approach must be open to handle all possible types and relations. In the example, an entity of type date and relation of type created was identified.

Also important is to identify the type of answer that DeepQA is supposed to generate, called the *lexical answer type (LAT)*. The LAT can be defined as a word in the clue that indicates the type of the answer. In this example, the LAT is Michigan city.

In addition, the question is also classified by a set of categories that usually correspond to slightly different pipelines that are needed in order to answer the question. In the example, the type of the question is a factoid, because the expected answer is an entity.

Figure 3-5 on page 34 shows the result of the question analysis step for the example which is Michigan city as the lexical answer type.

### 3.3.2 Primary search

This task is an information retrieval type of operation. The goal of the primary search is to find a set of possible sources that come from either structured or unstructured data and that contain the candidate answers. The output of primary search is a collection of documents and entities from both corpora and knowledge bases.

The primary search is performed in two ways:

- ▶ From unstructured data. The search is basically a search engine query, combining the keywords extracted by question analysis. The result is a list of text passages. DeepQA uses a combination of different search engines such as Apache Lucene and Indri.
- ▶ From structured data. The search is a SQL or SPARQL query to structured knowledge bases (KBs), returning a list of entities and their names. The question is converted from natural language to a structured query, matching the results of question analysis to the schema of the KBs that are used.

As a result of the primary search, a set of several hundreds of sources are identified both in documents and KBs. In the Jeopardy! settings, an average of 50 sources are collected for each question. These sources will be used to produce the candidate answers.

Figure 3-6 shows a primary search example starting with the keywords in the question and retrieving document sources.

- The keywords (1894, C.W. Post, created, warm, cereal, drink, Postum, Michigan, city) are used to search over millions of documents to find relevant hits.
- 55 documents are found, and 30 passages are found.

Indri Passage Search

Lucene Passage Search

Document Search Results	
Rank	Title
0	General Foods
1	Battle Creek
2	Post Foods
3	Will Keith Kellogg
4	Breakfast Cereal
5	John Harvey Kellogg
6	C. W. Post
7	Kellogg Company
8	Postum
...	...

Passage Search Results	
Rank	Passage
0	C.W. Post came to the Battle Creek sanitarium to cure his upset stomach. He later created Postum, a cereal-based coffee substitute
1	The caffeine-free beverage mix was created by The Postum Cereal Company founder C. W. Post in 1895 and produced and marketed by Postum Cereal Company as a healthful alternative to coffee
2	1895: In Battle Creek, Michigan, C.W. Post made the first POSTUM , a cereal beverage. Post created GRAPE-NUTS cereal in 1897, and POST TOASTIES corn flakes in 1908
3	1854 C. W. Post (Charles William) was born. He founded the Postum Cereal Co. in 1895 (renamed General Foods Corp. in 1922) to manufacture Postum cereal beverage
4	The company was incorporated in 1922, having developed from the earlier Postum Cereal Co. Ltd., founded by C.W. Post (1854-1914) in 1895 in Battle Creek, Mich. After a number of experiments, Post marketed his first product-the cereal beverage called Postum-in 1895
5	...

Figure 3-6 Primary search

### 3.3.3 Hypothesis generation

The goal of this step is to generate possible answers to the question from the document collection that is returned by the primary search; these answers are known as *candidate answers* or *hypotheses*. At this point, quantity trumps accuracy. Therefore, an important approach is to generate a large number (thousands) of possible answers so that the recall is close to 100% to make sure that the correct answer is included. DeepQA identifies and justifies the correct answer in the next steps. The result of the primary search is analyzed by using information extraction (IE) algorithms that are able to identify entities and other relevant terms in the documents that are provided by the primary search. The output of this step is a list of entities extracted from text and knowledge bases. In both cases, they are represented by strings (the entity names). In the case of KBs, the strings also contain links to their corresponding sources in the KB from which they have been extracted (for example, URIs of semantic web resources). All those entities are regarded as competing hypotheses (candidate answers) to be evaluated in the next steps of the DeepQA pipeline.

Candidate answers are identified from the sources following a variety of strategies, including identifying the title of Wikipedia articles, the anchor text of wiki links in the document content, and running Named Entity Recognition and keyphrase extraction algorithms.

The candidate answers are represented as strings, mostly noun phrases for factoid questions. Candidate answers are linked to their sources when possible, such as to their DBpedia URIs. For this example, these items are identified: General Foods, 1985, Post Foods, Battle Creek, and Grand Rapids. In a real case, hundreds of candidate answers are generated and each one is analyzed in the hypothesis and evidence scoring step.

Figure 3-7 shows an example of hypothesis generation results.

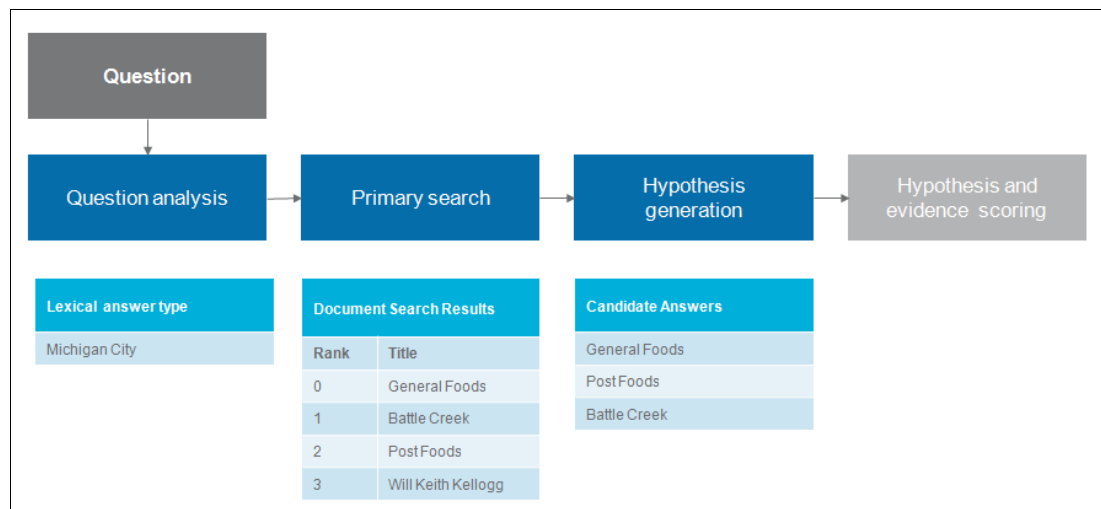


Figure 3-7 Hypothesis generation results

### 3.3.4 Hypothesis and evidence scoring

DeepQA does not simply create a bunch of answers; that approach is not enough. Instead, to win at Jeopardy!, Watson must be able to select the right answer. To this aim, DeepQA treats each candidate answer as a competing hypothesis. Therefore, DeepQA identifies evidence for each candidate answer by using various strategies. Some evidence can be found in a KB (such as, for example, the compatibility between hypothesis types and the answer type that is specified by the clue). Other evidence is found in textual passages of candidate answers.

After evidence is collected, DeepQA triggers a large number of analytics, trying to support and defend all different hypotheses from various perspectives. To achieve this goal, DeepQA uses various strategies. For example, one of the most important sources of evidence for the correctness of an answer is its type. Therefore, DeepQA checks the compatibility between each candidate's type and the lexical answer type that is required by the clue. Another crucial source of evidence is provided by analyzing the collected textual passages to check possible entailment between them and the clue. This family of analytics is referred to as *passage scorers*. To understand those passages, DeepQA uses NLP technology (such as information extraction, entity linking, paraphrasing, and so on) to implement textual entailment analytics. For example, assume the following clue and passage:

- ▶ Clue: In May 1898, Portugal celebrated the 400th anniversary of this explorer's arrival in India
- ▶ Passage: On the 27th of May 1498, Vasco da Gama landed in Kappad Beach

That clue and passage will provide great support for the answer Vasco Da Gama.

Different answer scoring algorithms rate the quality of answers from different points of view. For example, geographic reasoning has been used to assess the proximity between candidate entity and locations contained in the clue, temporal reasoning has been used to assess temporal compatibility, and so on. The DeepQA research team developed hundreds of possible answer scores that together provide a feature space for machine learning algorithms to assess the overall confidence of the answer, which is done in the final step (final merging and ranking).

Most of the candidate answers are wrong because, in Jeopardy!, only one answer is correct. To determine the correct answer, candidate answers undergo a rigorous evaluation process that involves gathering additional supporting evidence for each candidate answer, or hypothesis, and applying a wide variety of deep scoring analytics to evaluate the supporting evidence.

Additional evidence comes from the corpus itself. So, DeepQA performs a search engine query for each candidate against the same index that was used for the primary search, looking for text passage that contains both the candidate answer and some of the terms in the clue.

Answer scoring algorithms determine the degree of certainty that the retrieved evidence supports the candidate answers. The DeepQA framework supports and encourages the inclusion of many different answer scorers, which consider different dimensions of the evidence and produce a score that corresponds to how well the evidence supports a candidate answer for a given question. The scorers analyze the question, the candidate answer and the evidence from different perspectives, and give a score to the features that represent those perspectives.

In the example, a very basic scorer is the *lexical overlap*, showing how many keywords are in common between the question and the supporting evidence for each candidate answer. Another scorer looks for *taxonomic* relations between the candidate answer and the lexical answer type in a large taxonomy of types. For example, is General Foods of type Michigan City? The answer is no, so the score for taxonomy is zero. Another perspective in this example is to analyze geographic proximity between the entities in the candidate answers and the state of Michigan. Battle Creek and Grand Rapids are cities in Michigan so, in these cases, the score is high. This spatial analysis can be done for some entities, for example city and company, but not for others, for example a date. Another perspective is to find *temporal* evidence. In this case, the candidate answer, 1985, is far from the date mentioned in the question (that is, 1894), motivating a low temporal score, while the foundation date of General Foods is close, receiving a high score for this feature.



One of the main challenges of the hypothesis and evidence scoring step is that it requires a massively parallel scale-out framework. In fact, to score hundreds of candidate answers with hundreds of answer scorers, each relying on dozens of supporting textual evidence, DeepQA must be able to execute millions of analytics in parallel in a few seconds. This is entirely handled by UIMA AS (Asynchronous Scaleout)<sup>1</sup>, used as a semantic integration platform that enables scale out on thousands of cores in a massively parallel architecture.

DeepQA combines the features at the end and uses them as features vectors to represent each candidate.

Figure 3-8 shows the results of the hypothesis and evidence scoring step for the example.

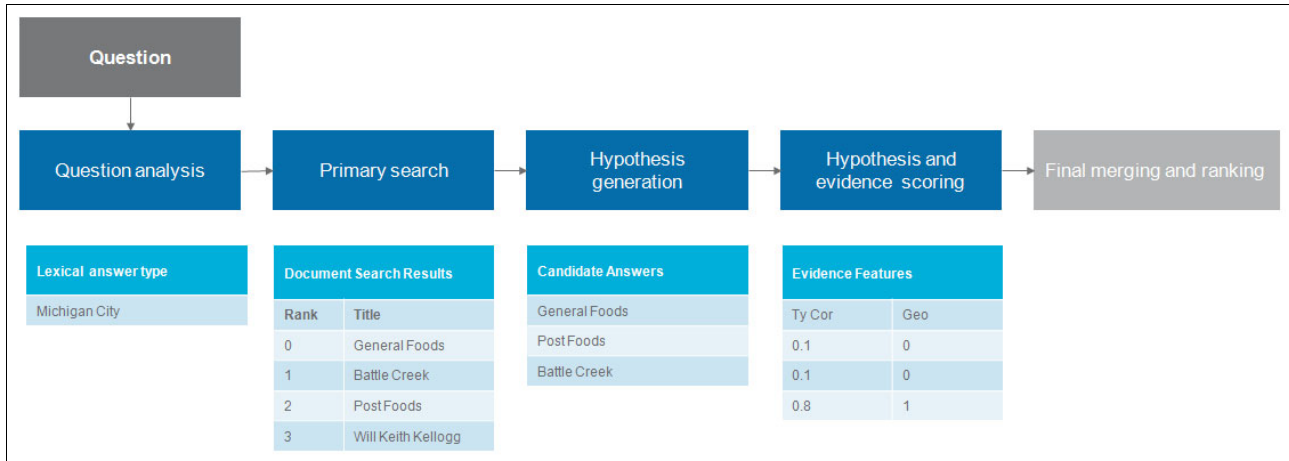


Figure 3-8 Hypothesis and evidence scoring results

### 3.3.5 Final merging and ranking

The final step for DeepQA is to rank each answer based on confidence, which is assessed by aggregating multiple evidence sources and their analysis. This is done by using machine learning techniques, exploiting historical data that contains clues and corresponding answers. Plenty of such data is available from almost 40 years of previous Jeopardy! matches, providing millions of overall data points. Evidence scoring analytics provide a rich feature space for this purpose. This is not about memorizing trivia. By playing thousands of practice games, Watson learns how to weigh, apply, and combine its own algorithms to help decide the degree to which each piece of evidence is useful or not.

These weighed evidence scores are merged together to decide the final ranking for all the possible answers. This is done by training a logistic regression model on the task of providing a confidence score close to 1 (one) for the positive answers, and close to 0 (zero) for all remaining candidate answers.

<sup>1</sup> UIMA: An Architectural Approach to Unstructured Information processing in the Corporate Research Environment, David Ferrucci and Adam Lally, IBM T.J. Watson Research Center: <https://ibm.biz/BdiVks>



After confidence for each answer is generated, answers are finally ranked accordingly, and the top one is selected if the confidence is above a game-specific threshold. The strategy used by Watson to play the Jeopardy! game is an important piece of research in itself. However, this book does not explore that subject.

An important feature of the final ranking is that equivalent answers are grouped together and their supporting evidence is combined. This is called *final merging*. Final merging is typically done by exploiting information in the knowledge bases, such as synonymy or selected semantic relations.

A regularized logistic regression algorithm is used to take each feature vector and assign a single confidence value to the candidate answer. It is trained on thousands of candidate answers and question pairs, each labeled for whether the answer is correct or incorrect with respect to the question, together with their feature vectors, and learning to predict a probability of being a correct answer.

Historical data, composed by several thousands of clues, was used to train the algorithm for the Jeopardy! challenge. During the training phase, the system considers the correct answer (in this case, Battle Creek) as a positive example and gives it a weight of 1 (one). All others are answers returned by hypothesis generation are supposed to be negatives, receiving a weight of 0 (zero). Training is performed over thousands of question-answer pairs.

The generated models learn how to assign different weights to the different features and are applied to unseen question-answer pairs to estimate their confidence. In the example, the candidate with the highest level of confidence is Battle Creek, which then becomes the final answer. Because its confidence is high (0.85), Watson will likely press the game buzzer and attempt to answer the question.

Figure 3-9 shows the results of the final merging and ranking step in the example.

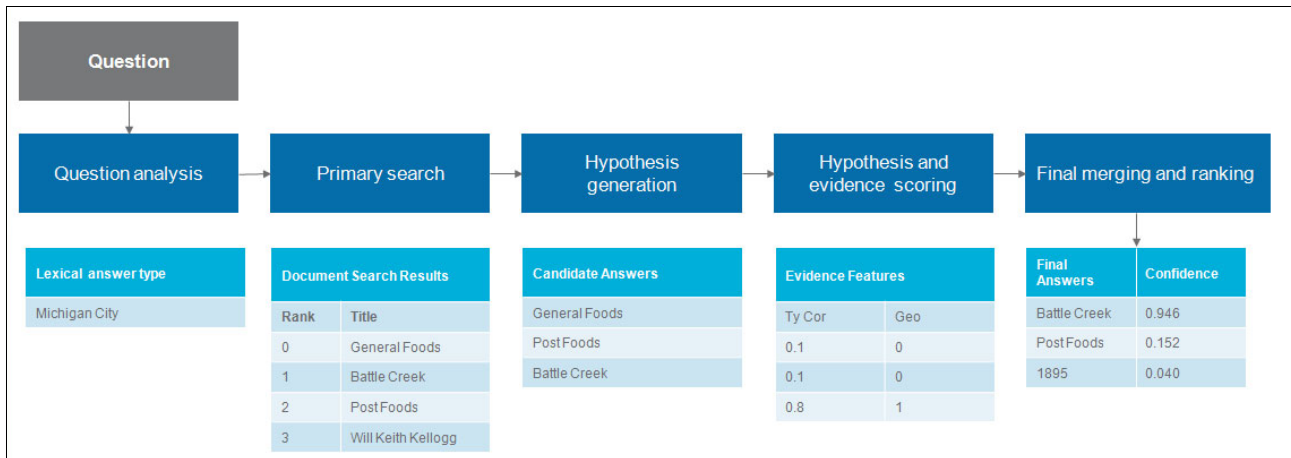


Figure 3-9 Final confidence scores

## 3.4 References

For more information, see the following resources:

- ▶ Building Watson: An Overview of the DeepQA Project:  
<https://www.aaai.org/ojs/index.php/aimagazine/article/view/2303>
- ▶ IBM Watson: The science behind the answer:  
<https://youtu.be/V7B7w7Z15nk>
- ▶ Watson and the Jeopardy! Challenge:  
<https://youtu.be/P18EdAKuC1U>
- ▶ A framework for merging and ranking of answers in DeepQA:  
[http://researcher.watson.ibm.com/researcher/files/us-heq/W\(16\)%20ANSWERS%20MERGING\\_RANKING%2006177810.pdf](http://researcher.watson.ibm.com/researcher/files/us-heq/W(16)%20ANSWERS%20MERGING_RANKING%2006177810.pdf)
- ▶ Semantic Technologies in IBM Watson:  
[http://www.patwardhans.net/papers/GliozzoBPM13.pdf?cm\\_mc\\_uid=&cm\\_mc\\_sid\\_50200000=](http://www.patwardhans.net/papers/GliozzoBPM13.pdf?cm_mc_uid=&cm_mc_sid_50200000=)
- ▶ A Computer Called Watson:  
<http://www.ibm.com/ibm/history/ibm100/us/en/icons/watson/breakthroughs/>
- ▶ Question analysis: How Watson reads a clue:  
[http://researcher.watson.ibm.com/researcher/files/us-heq/W\(4\)%20QUESTION%20ANALYSIS%2006177727.pdf](http://researcher.watson.ibm.com/researcher/files/us-heq/W(4)%20QUESTION%20ANALYSIS%2006177727.pdf)



# Evolution from DeepQA to Watson Developer Cloud

Watson Developer Cloud offers a number of cognitive services as RESTful APIs. The APIs are delivered on IBM Bluemix, which is the platform as a service (PaaS) offering from IBM.

This chapter charts the evolution of the Watson Developer Cloud, from the initial DeepQA architecture, to some of the early products like Watson Engagement Advisor and Watson Oncology Advisor to the suite of APIs that are available today through IBM Bluemix. The primary focus is on highlighting the lessons learned from implementing Watson technology in practical applications and how those lessons influenced the IBM product strategy.

Why IBM decided to commercialize Watson and make cognitive computing a strategic imperative of the company are explored in this chapter.

A quick review of the DeepQA architecture is presented, although more details are available in Chapter 3, “Introduction to question-answering systems” on page 29.

This chapter discusses key capabilities that had to be developed to make question-answering systems practical in real-world applications (and why the DeepQA architecture had to evolve). Also, this chapter examines current Watson Conversation and Discovery services architectures and a reference architecture for implementing those services in production.

The following topics are covered in this chapter:

- ▶ Why commercialize Watson
- ▶ Refresher of DeepQA architecture
- ▶ Evolution to Watson Developer Cloud
- ▶ Watson Conversation service
- ▶ Watson Discovery service
- ▶ Evolution summary
- ▶ References

## 4.1 Why commercialize Watson

Approximately the same time that Watson was competing on the Jeopardy! quiz show, the technology world was being disrupted by three fundamental forces (Figure 4-1):

- ▶ The advancement in machine learning capabilities, opening up new applications of predictive analytics, natural language processing (NLP), speech recognition, and computer vision
- ▶ The ability for these capabilities to be offered through APIs on the cloud, massively decreasing the time-to-value of cognitive computing
- ▶ The amount of data that is available and the potential to harness it beyond traditional analytics

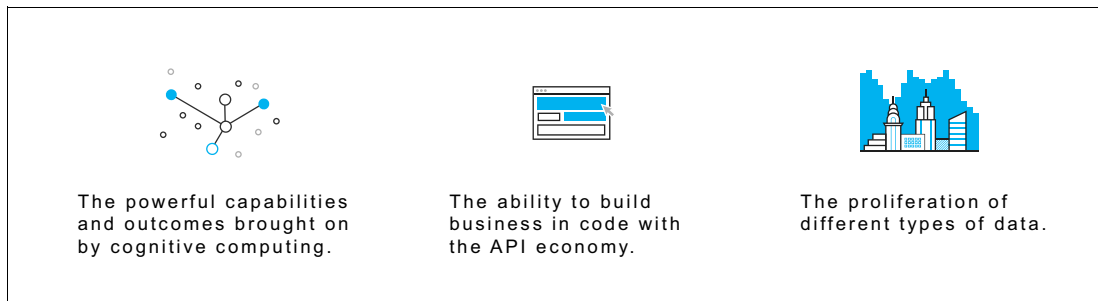


Figure 4-1 Disruption is fueled by three forces

Data has become the competitive advantage and most of it is invisible to traditional computing platforms. In healthcare, each year seems to produce data in an exponential amount compared to prior years. Understanding unstructured data in the form of text documents, images, videos, and raw sensor output provides the vast proportion of the opportunity. Figure 4-2 shows the data growth predictions.

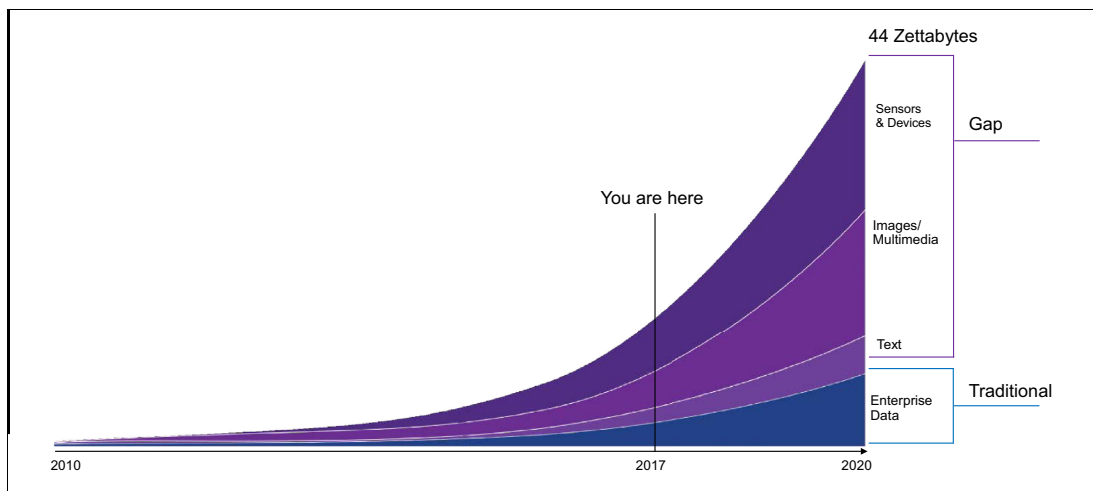


Figure 4-2 Data is the competitive advantage

Figure 4-3 shows data use in several industries<sup>1</sup>.



Figure 4-3 Data and industry sectors

In 2014, IBM formed the Watson Group to commercialize Watson technology. Work done with Oncologists at Memorial Sloan Kettering Cancer Center became the Watson Oncology Advisor. Efforts to organize and unlock the value of text documents became Watson Explorer and Watson Discovery Advisor.

The focus of the remainder of this chapter is on the evolution of the efforts by Watson in Engagement and Conversational artificial intelligence (AI), illustrated in Figure 4-4 on page 44.

<sup>1</sup> Source: <https://ibm.biz/BdiA2N>

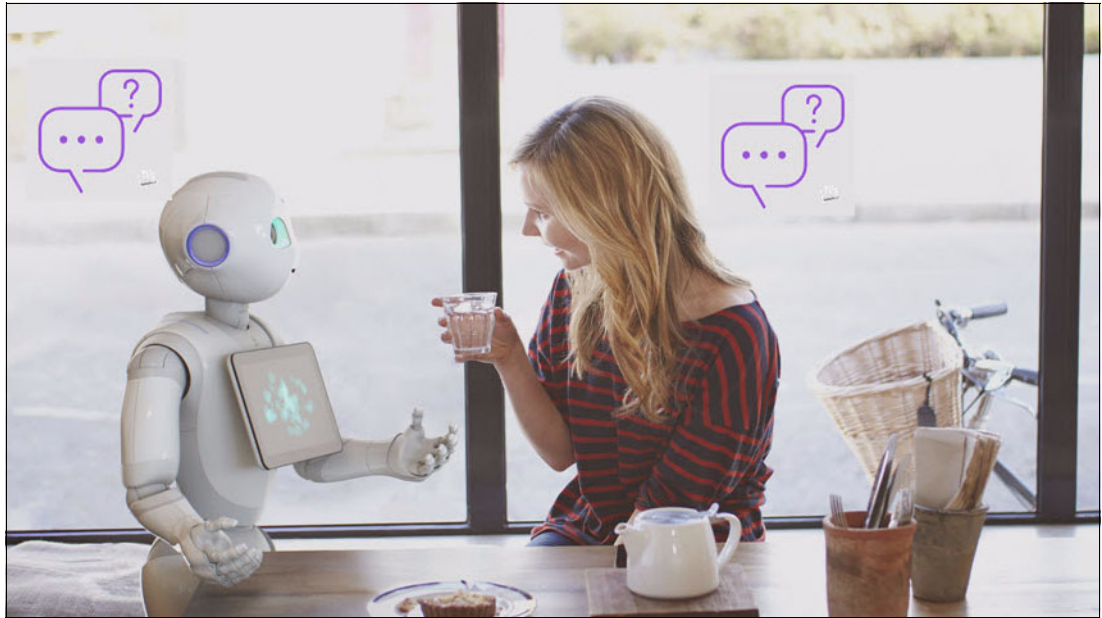


Figure 4-4 Engagement and Conversation

Most companies expect to compete primarily on the basis of customer experience. Poor customer service leads to customer churn and, worse, customers now share those experiences across social media. Young customers expect self-service, on their terms, in the channels of their choice, with no desire to call traditional 1-800 (toll-free) phone numbers. The virtual assistant market alone is expected to grow to USD (\$) 1 billion by 2018. The service robotics and home automation markets are worth tens of billions. So, what will it take to apply DeepQA in the service of these applications?

## 4.2 Refresher of DeepQA architecture

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 primary computational principle supported by DeepQA is to assume and pursue multiple interpretations of the question, to generate many plausible answers or hypotheses, and to collect and evaluate many competing evidence paths that might support or refute those hypotheses (Figure 4-5 on page 45).

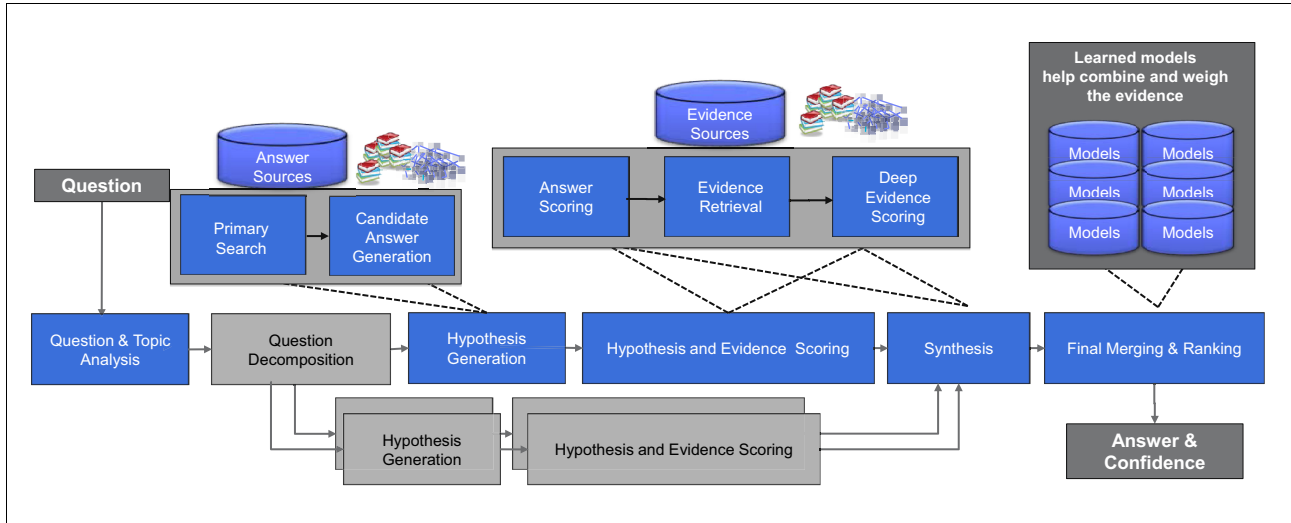


Figure 4-5 DeepQA high level architecture

In the *question analysis* step, parsing algorithms decompose the question into its grammatical components. Other algorithms in this step identify and tag specific semantic entities, such as names, places, or dates. In particular, the type of thing being asked for, if it is indicated at all, will be identified. This is called *lexical answer type (LAT)* and it is the word in the question that indicates the type of answer to look for.

In the *hypothesis generation* step, DeepQA does a variety of broad searches for each of several interpretations of the question. These searches are performed over a combination of unstructured data (natural language documents) and structured data (available databases and knowledge bases) fed to Watson during training. The focus, at this point, is on generating a broad set of hypotheses, which for this application, are called *candidate answers*.

In the *hypothesis and evidence scoring* step, the candidate answers are first scored independently of any additional evidence by deeper analysis algorithms.

In the *merging and ranking* step, the many possible answers are scored by many algorithms to produce hundreds of feature scores. Trained models are applied to weigh the relative importance of these feature scores. These models are trained with machine learning methods to predict, based on past performance, how best to combine all these scores to produce final, single-confidence numbers for each candidate answer and to produce the final ranking of all candidates. The answer with the strongest confidence is Watson's final answer.



## 4.3 Evolution to Watson Developer Cloud

How has the original DeepQA architecture, designed to play the Jeopardy! game, evolved to work in the Customer Service domain? First, consider the question (Figure 4-6). Jeopardy! is an open-domain factoid question-answering problem, Customer Service is a closed-domain problem with multiple potential question-classes, including Factoid, Descriptive, Yes/No, Procedural How To, and Procedural Troubleshooting.

The first insight is that the question distribution in a closed domain problem, such as Customer Service, is different than in Jeopardy!, which makes dealing with multiple question-classes a tenable problem.

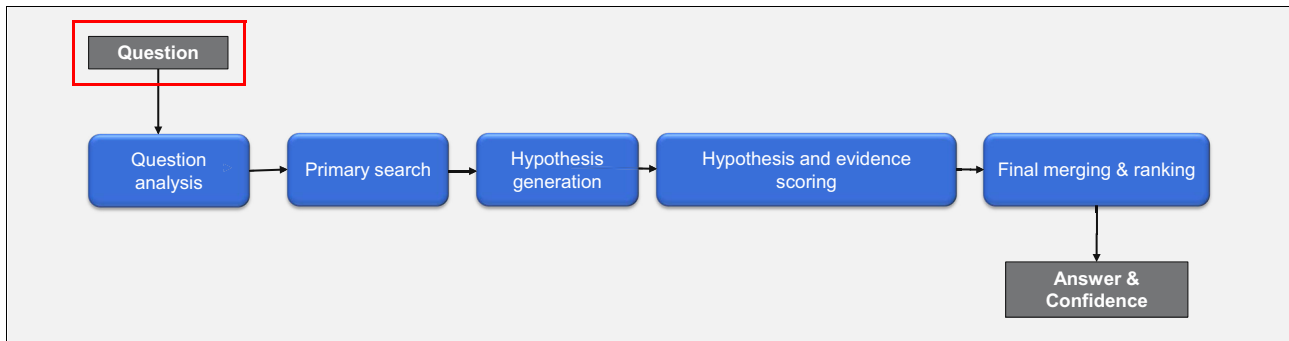


Figure 4-6 DeepQA pipeline: the question

In the Jeopardy! case, no attempt was made to anticipate questions and build databases of answers.

In 13% of the sampled questions, the findings showed no clear indication of the type of answer and the players must rely almost entirely on the context to determine what sort of answer is required.

The remaining 87% is what you see in the graph in Figure 4-7 on page 47. It shows, what is called *long tail distribution*. No set of topics is small enough to focus on that covers enough ground. Even focusing on the most frequent few (the head of the tail to the left) will cover less than 10% of the content.

Thousands of topics, from hats to insects, to writers, to diseases, to vegetables, are all equally fair game. And for these thousands of types, thousands of different questions might be asked and then phrased in a huge variety of ways.

Pre-existing structured knowledge in the form of databases (DBs) or knowledge bases (KBs) is used to help to bridge meaning and interpret multiple natural language (NL) texts. However, because of the broad domain and the expressive language used in questions and in content, pre-built DBs have limited use for answering any significant number of questions. Rather, the focus is on NL understanding.



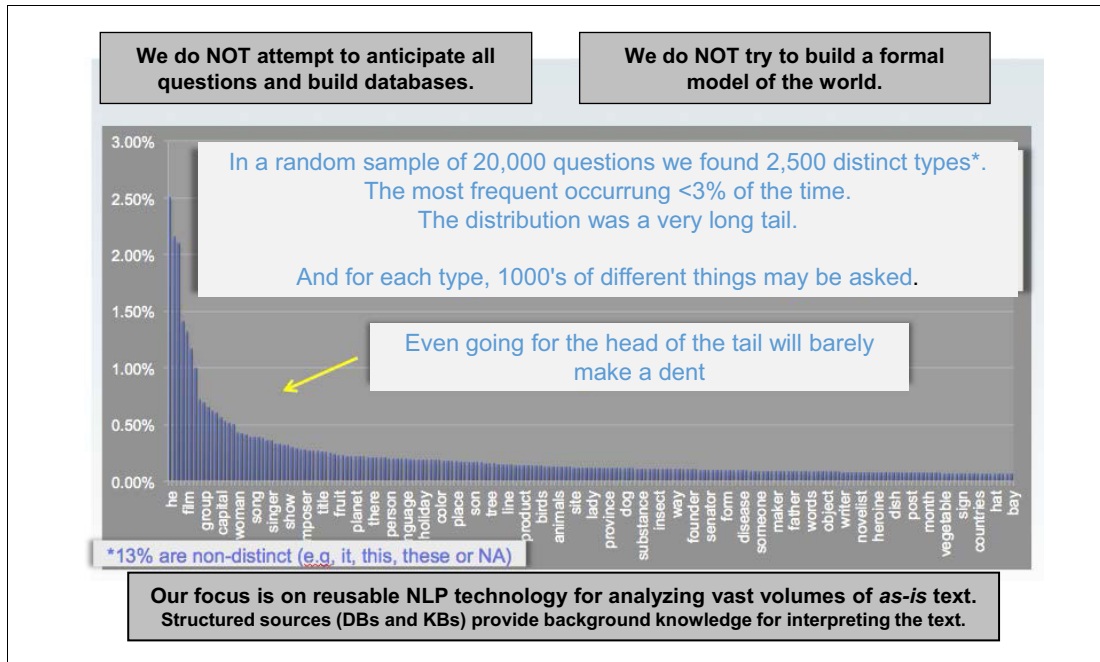


Figure 4-7 Broad domain (long tail distribution)

In the Customer Service domain, you see a much fatter *short head* (Figure 4-8). A typical distribution might be 12 question types representing 80% of questions. These types range from chit-chat (greetings and other social acts) to frequently asked questions. In fact, these are not all questions so they are referred to as *utterances*. Users demand high accuracy (100% in most cases) when dealing in the short head. Therefore, pre-existing knowledge that is mapped to these pre-identified utterances is required.

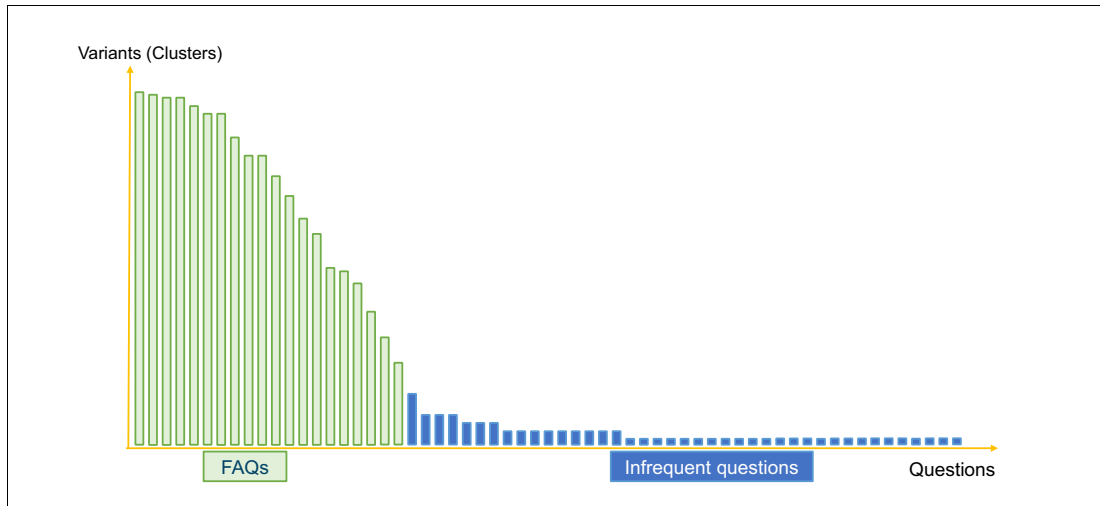


Figure 4-8 The question distribution from a typical customer service use-case

So, if the utterance is recognized, a predefined response should be taken from a structured repository and provided to the user (Figure 4-9).

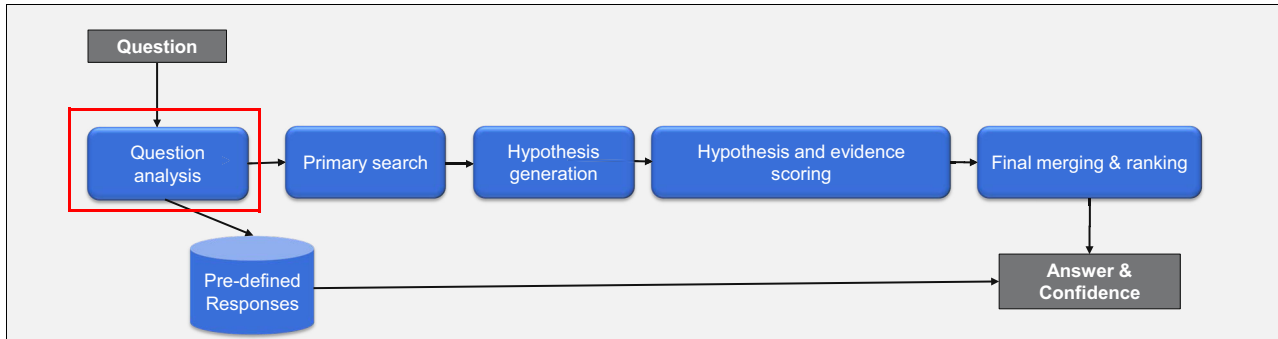


Figure 4-9 DeepQA pipeline: Pre-defined responses for recognized utterance

### 4.3.1 Evolution of question analysis

This section discusses the evolution of *question analysis*.

A typical utterance in a Customer Service domain is “I’m frustrated, I haven’t been able to log into your online billing system” (Figure 4-10).

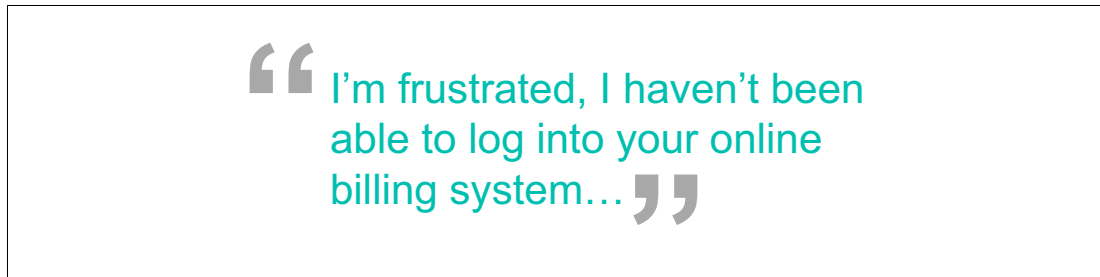


Figure 4-10 Customer is frustrated

In Jeopardy!, which is mostly a factoid question-answering problem, a specific lexical answer type (LAT) can be identified in 87% of questions. As a reminder, the LAT is a specific word in the question that represents the answer type. In the Customer Service domain, having a specific word in the utterance that represents the type is rare. Rather, the question intent must be inferred from the utterance as a whole, as Password Reset in the example shown in Figure 4-11 on page 49.

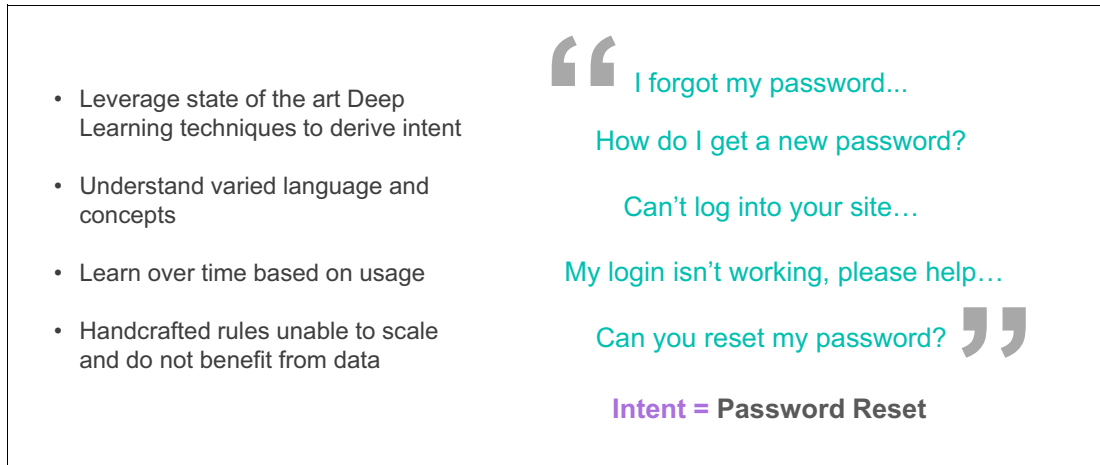


Figure 4-11 Understand the customer's intent

To classify intent (Figure 4-12), Watson *Natural Language Classifier* service was developed. It is a classification service, based on deep learning, that is optimized for short input text and in the order of tens to hundreds of text classes.

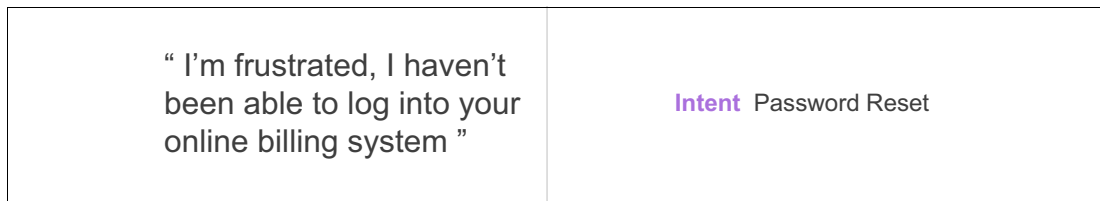


Figure 4-12 Extract other key information from a question: detect intent

As in Jeopardy!, extracting the entities that represent the people, places, and things in the question help to map the intent to the appropriate response. In this example, the entity “Online Billing System” (Figure 4-13) represents the specific platform that requires a password reset. Therefore, the Watson *Natural Language Understanding* service was developed. It can be trained to extract the entities required in a specific domain.

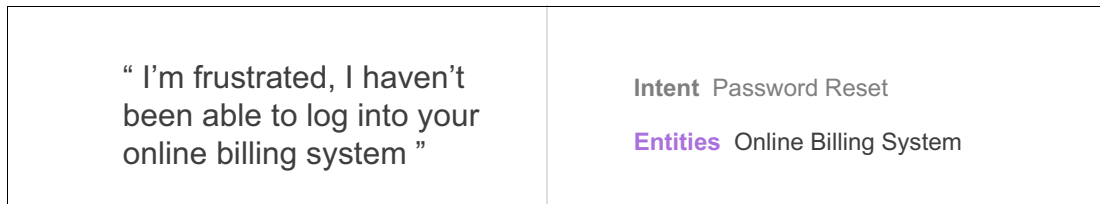


Figure 4-13 Extract other key information from a question: Entities

Another critical facet of question analysis in the Customer Service domain is customer emotion (Figure 4-14 on page 50). A great customer service agent must be able to empathize with the customer; a virtual agent should do the same. The Watson *Tone Analyzer* service uses linguistic analysis to detect three types of tones in text: emotions, social tendencies, and writing style. The Tone Analyzer service can be used to understand emotional context of conversations and communications in order to respond in an appropriate manner.

<p>“ I’m frustrated, I haven’t been able to log into your online billing system ”</p>	<p><b>Intent</b> Password Reset</p> <p><b>Entities</b> Online Billing System</p> <p><b>Emotional Tone</b> Anger</p>
---	---

Figure 4-14 Extract other key information from a question: Emotion

Finally, user context is critical to driving the correct response (Figure 4-15). Bill Smith who is a “gold” member and communicates through a mobile device should get a personalized response. The system must maintain that context through the duration of the interaction and even over disparate interactions. Implied in this requirement is that a single turn interaction is often insufficient. The system must be able to request required context, guide a user through a series of steps, or both.

The Watson *Dialog* service was developed to enable a developer to automate branching conversations between a user and the application. The Dialog service enabled applications to use natural language to automatically respond to user questions, cross-sell and up-sell, guide users through processes or applications, or even “hand-hold” users through difficult tasks. The Dialog service could track and store user profile information to learn more about that user, guide that user through processes based on the user’s unique situation, or pass the user information to a back-end system to help the user take action and get the help needed.

**Note:** The Watson Dialog service was retired and replaced by the Conversation service described in 4.4, “Watson Conversation service” on page 56. References to the Dialog services in this section show the evolution of the Watson Developer Cloud.

<p>“ I’m frustrated, I haven’t been able to log into your online billing system ”</p>	<p><b>Intent</b> Password Reset</p> <p><b>Entities</b> Online Billing System</p> <p><b>Emotional Tone</b> Anger</p> <p><b>Context</b> Bill Smith, 47, Gold Member, High Value</p> <p><b>Context</b> Mobile</p>
---	--

Figure 4-15 Extract other key information from a question: Context

Intent, entities, emotion, and context are combined to drive the next stage of the question-answering pipeline (Figure 4-16 on page 51). In each of the previous scenarios, the correct response could be generated without the need for hypothesis generation, answer scoring, or ranking, thereby ensuring high accuracy and customer satisfaction in the short head of the question distribution.

Question		Answer
How do I reset my password?	— Dialog —>	Guide the user through a set of steps
Someone has stolen my credit card.	— Defect —>	Transfer to human agent
Where is the nearest store?	— Map —>	Application launches map with directions
I need to pay my outstanding invoice.	— App Nav. —>	Bring user to pay bill screen
Can I pay my bills using my credit card?	— Info. Retrieval —>	Bring back an answer

Figure 4-16 Take action: Responses come in different forms

The question analysis step of the pipeline is now a set of highly configurable *microservices* that greatly enhance the ability to customize the system for a particular domain (Figure 4-17).

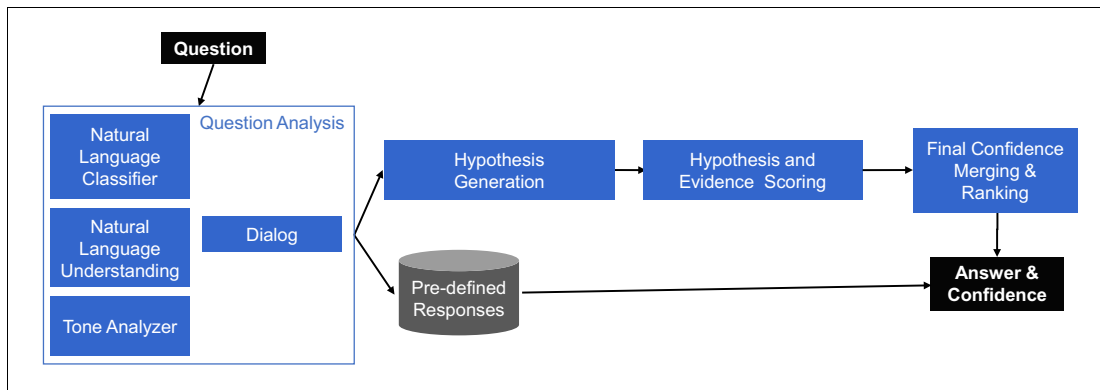


Figure 4-17 Evolution of the DeepQA question analysis step: configurable microservices

In the Customer Service domain, while the short head represents a larger proportion of utterances than the distribution for the Jeopardy! clues, the long tail is still a critical element (Figure 4-8 on page 47). Often, the more complex long tail questions are the most frustrating for both customers and customer support agents.

For example, a more complex utterance that might fall in the long tail of the question distribution is “I reset my password but now I’m getting a sporadic error and only when I log in from my desktop app” (Figure 4-18).

“ I reset my password but now I’m getting a sporadic error and only when I log in from my desktop app... ”

Figure 4-18 Complex utterance in the long tail distribution

The question is unique enough that having a predefined answer is unlikely. As such, a corpus that contains relevant answers must be leveraged (Figure 4-19). In the Customer Service domain, this is likely existing service desk tickets, product manuals, knowledge base articles, and so on.

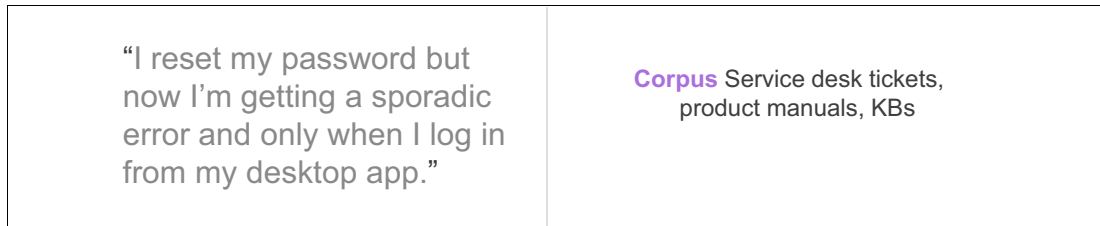


Figure 4-19 Build a long tail solution for complex question: Corpus

These service desk tickets, manuals, and articles might be in many formats: PDF files, Microsoft Word documents, web pages, and others. What is needed is a scalable mechanism to convert those documents into a shared format and the ability to segment those documents into relevant answer units (Figure 4-20). The Watson *Document Conversion* service was designed to do exactly that.

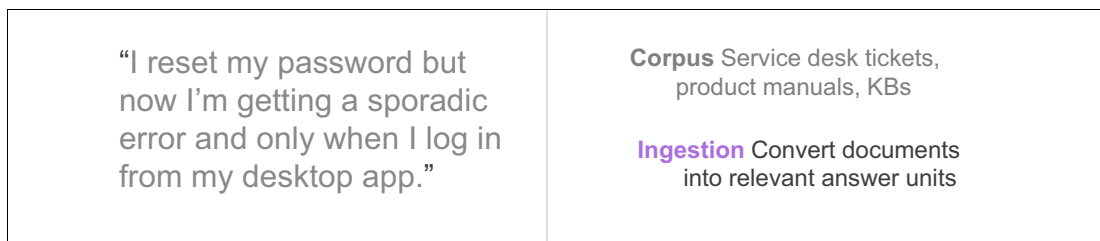


Figure 4-20 Build a long tail solution for complex question: Ingestion

Finally, just like in the DeepQA pipeline, executing primary search and score and rank candidate answers is necessary (Figure 4-21). The Watson *Retrieve and Rank* service is an Apache Solr cluster in the cloud with a custom query builder optimized for natural language queries, a set of feature scorers to evaluate query/candidate answer overlap, and a machine learning-based ranker that can be trained with questions in the specific domain.

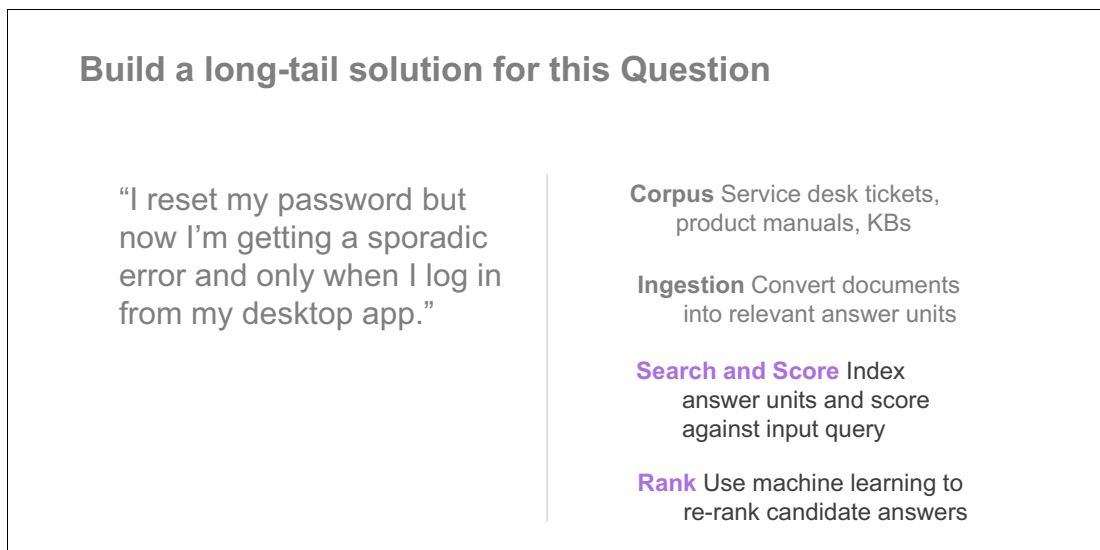


Figure 4-21 Primary search and score and rank

Retrieve and Rank becomes the solution for the long tail (Figure 4-22). Note, the evidence generation and scoring steps are removed from DeepQA. Making cognitive systems explicative and evidence-based is an important aim of IBM, especially in domains like healthcare. In customer service applications, the requirements for speed of implementation to reduce time-to-value outweigh the potential increase in performance that evidence generation might provide. As always, tradeoffs exist when you implement cognitive solutions in practice.

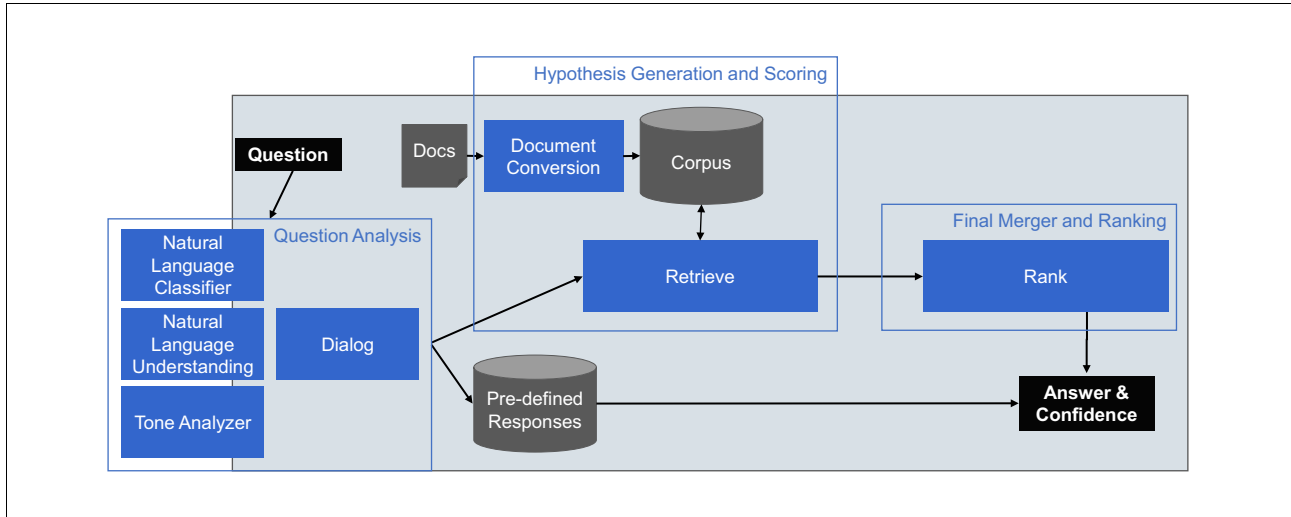


Figure 4-22 DeepQA pipeline; evolution to Watson Developer Cloud: Retrieve and Rank

As these services are implemented in production applications, handling multiple modes of communication is necessary (Figure 4-23). Speech is common in mobile, home automation, and wearable applications. Being able to understand and translate between languages is critical to expanding globally. And finally, the ability to understand images and videos can greatly expand the potential of engagement applications in domains, like robotics and connected cars.

<p>“I reset my password but now I’m getting a sporadic error and only when I log in from my desktop app.”</p>	<p><b>Speech</b> Convert speech to text and text to speech</p> <p><b>Languages</b> Understand and translate between languages</p> <p><b>Images</b> Interpret image content</p>
---	--

Figure 4-23 Deal with multi-modal communication

Watson *Speech to Text*, *Text to Speech*, *Visual Recognition*, and *Language Translator* services were developed to round out machine perception capabilities of Watson in addition to the NLP tools highlighted in the question analysis phase (Figure 4-24).

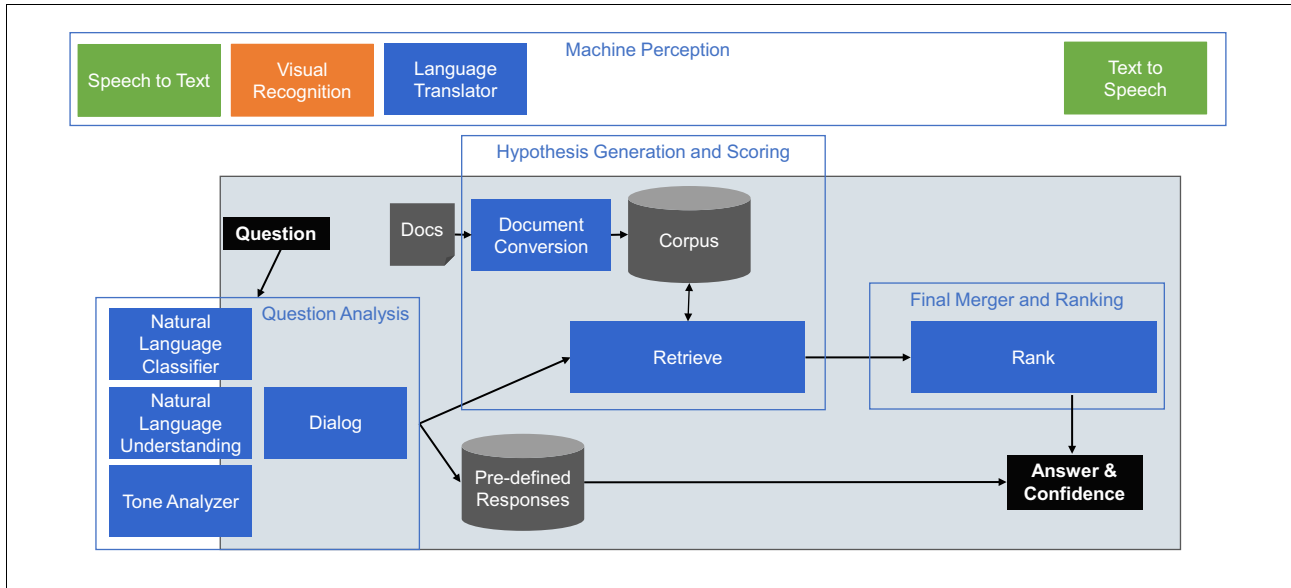


Figure 4-24 DeepQA pipeline; evolution to Watson Developer Cloud: Watson services overview

### 4.3.2 Microservices and robust tooling evolved from DeepQA

Cognitive solutions consist of two primary components (Figure 4-25 on page 55):

- ▶ Machine perception layer
- ▶ Decision making layer

Machine perception enables the cognitive system to extract signals from the world around it, making visible the 80% of data that was invisible to traditional computing systems. Those signals feed into the decision making layer that can be based on traditional business process logic or more advanced predictive analytics.



Figure 4-25 shows the suite of microservices that evolved from DeepQA and the efforts to implement Watson in real applications.

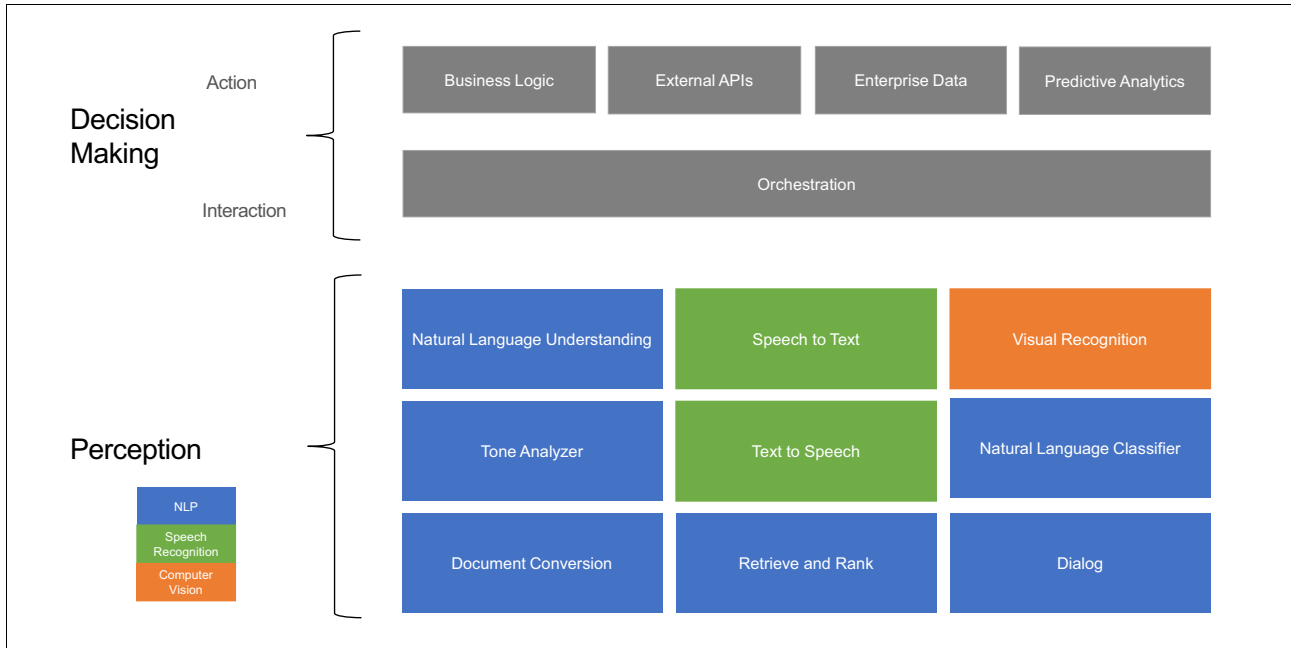


Figure 4-25 Watson Developer Cloud in approximately 2015

The next step in the evolution of Watson Developer Cloud was to orchestrate these machine perception capabilities aligned around key use cases and to provide robust tooling to enable configuration and domain adaptation all in the aim of decreasing time-to-value for clients.

## 4.4 Watson Conversation service

Most of this chapter has been discussing the engagement use case, which became Watson *Conversation* service.

Watson Conversation service allows you to quickly build, test, and deploy a bot or virtual agent across mobile devices, messaging platforms such as Slack, or even on a physical robot. Conversation has a visual dialog builder to help you create natural conversations between your apps and users, without requiring any coding experience.

Figure 4-26 shows Watson today.

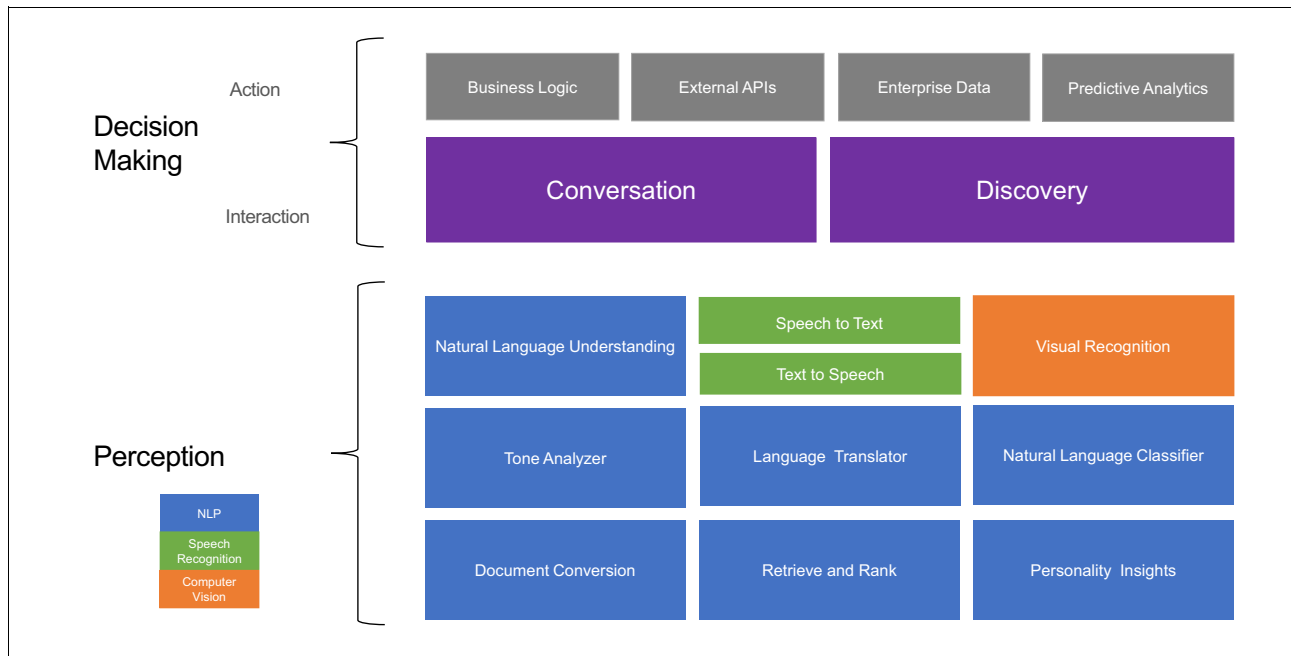


Figure 4-26 Conversation and Discovery services: Watson Developer Cloud today (2017)

Watson Conversation orchestrates the capabilities represented by Natural Language Classifier, Natural Language Understanding, and Dialog and exposes them through a single tool. Additional capabilities, like Tone Analyzer, Speech to Text, and Text to Speech integrations, are planned in the future (although developers can orchestrate these services themselves in the applications layer).

Figure 4-27 on page 57 shows the Conversation user interface (UI); Figure 4-28 on page 57 shows the reference architecture for Conversation.

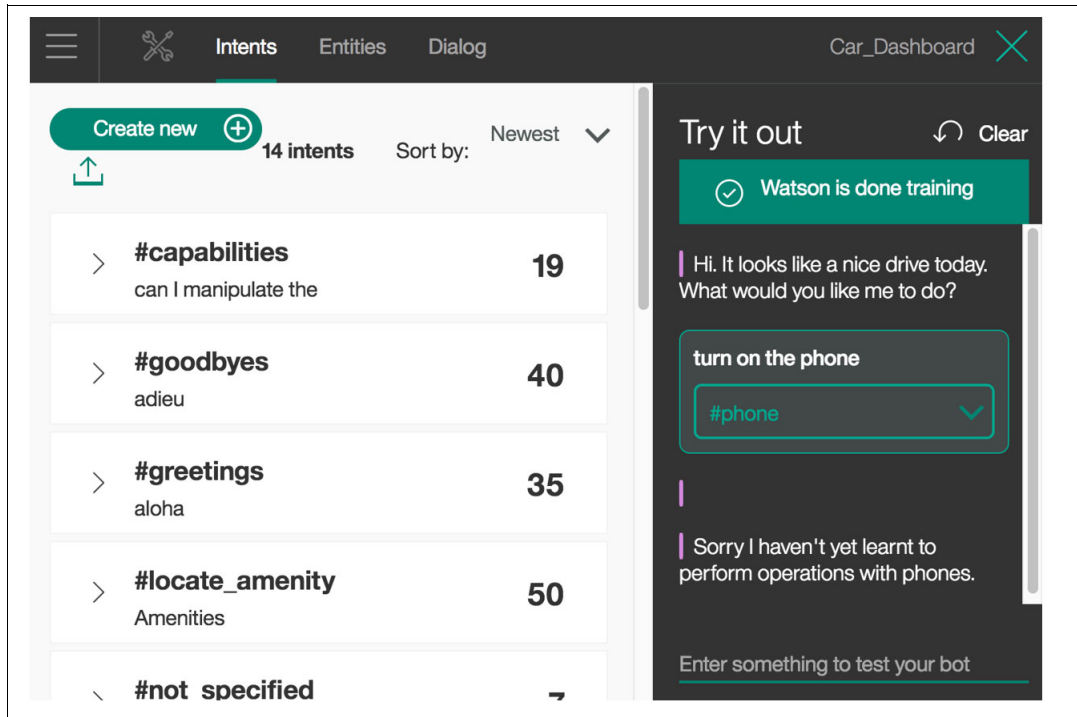


Figure 4-27 Conversation service: User interface

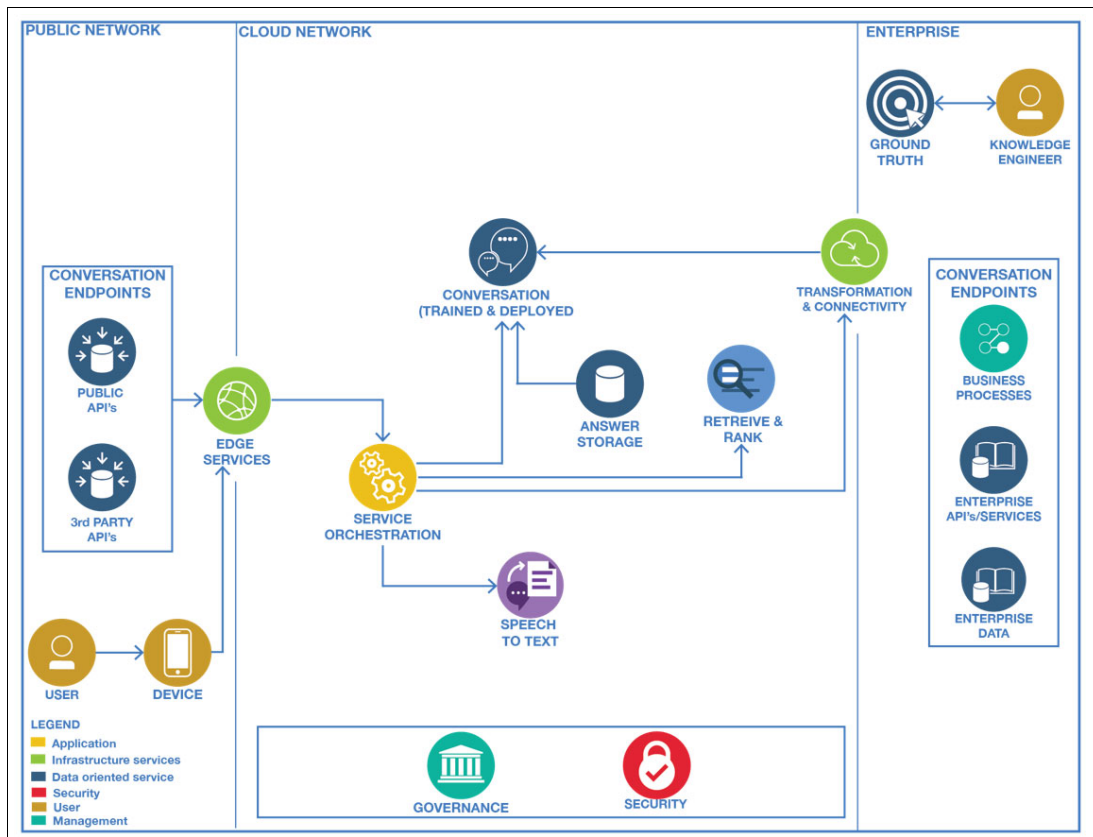


Figure 4-28 Conversation: Reference architecture

## 4.5 Watson Discovery service

Watson *Discovery* service (Figure 4-26 on page 56) allows users to extract value from unstructured data by converting, normalizing, and enriching it. By using a simplified query language, a user can explore the data or quickly tap into pre-enriched data sets, such as the Discovery News collection. Discovery News primarily includes English language news sources that are updated continuously, with over 300,000 new articles and blogs added daily, from more than 100,000 sources.

Watson Discovery orchestrates the capabilities represented by Document Conversion, Retrieve and Rank, Natural Language Understanding, and others, exposing them through a single tool.

Figure 4-29 shows the Watson Discovery UI; Figure 4-30 on page 59 shows a reference architecture for Discovery.

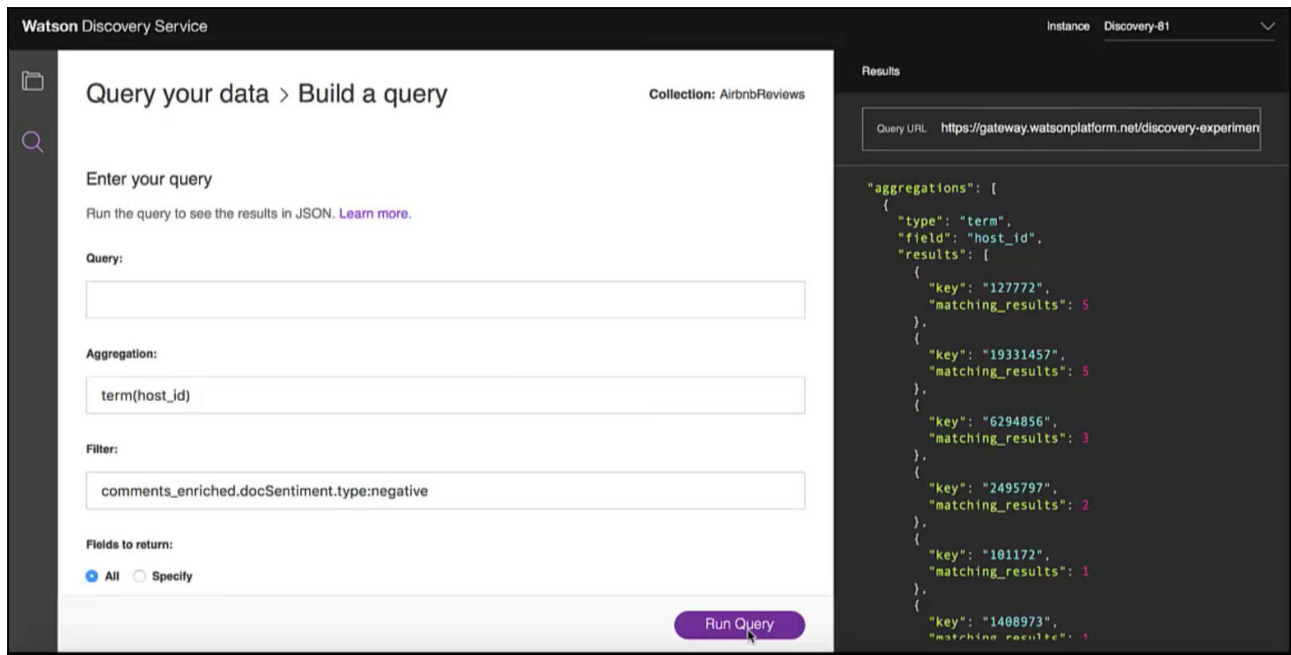


Figure 4-29 Discovery service: User interface

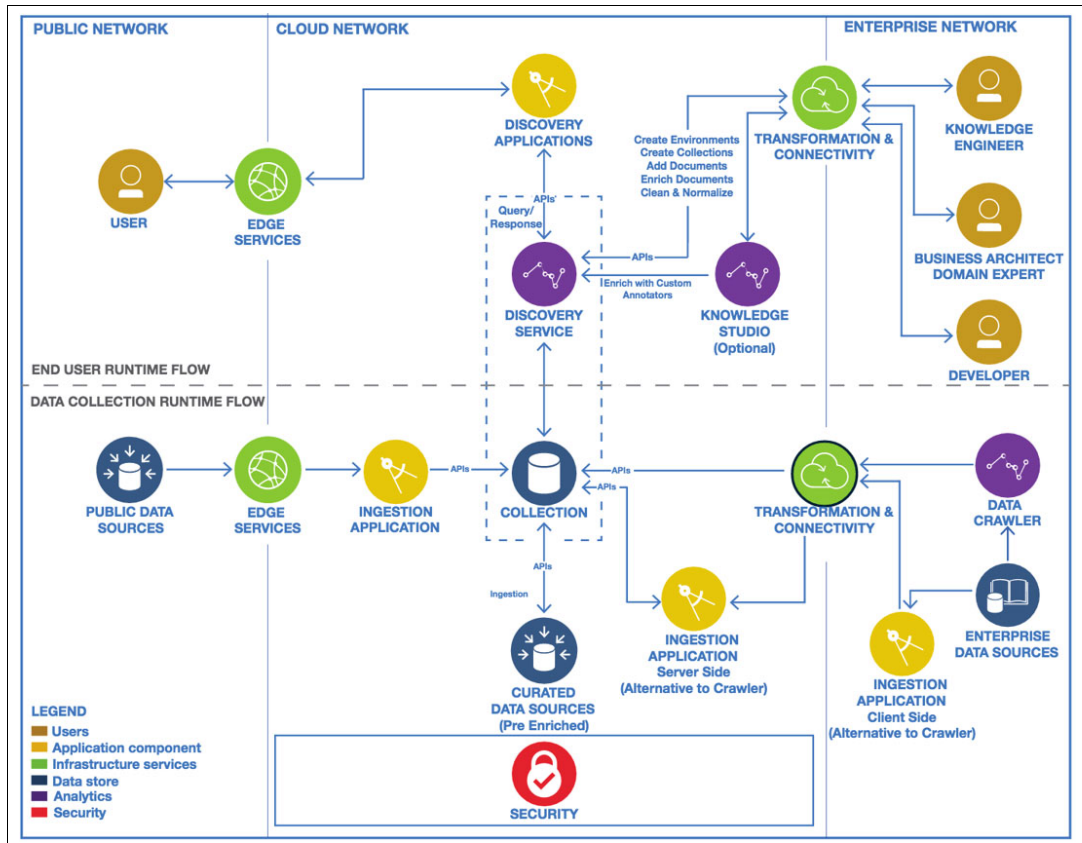


Figure 4-30 Discovery: Reference architecture

## 4.6 Evolution summary

Bringing it all together, the goal of the Watson Developer Cloud is to provide the most robust and flexible platform for building cognitive applications in deep industry domains (Figure 4-31). The microservices architecture enables developers to envision a broad range of potential applications by mixing and matching services. Conversation and Discovery provide clear direction around the most important use cases. Watson Knowledge Studio provides the tools to teach Watson the unique characteristics of your domain. Although the original DeepQA architecture and technologies have evolved and modernized, the “fingerprints” can be found throughout the Watson Developer Cloud.

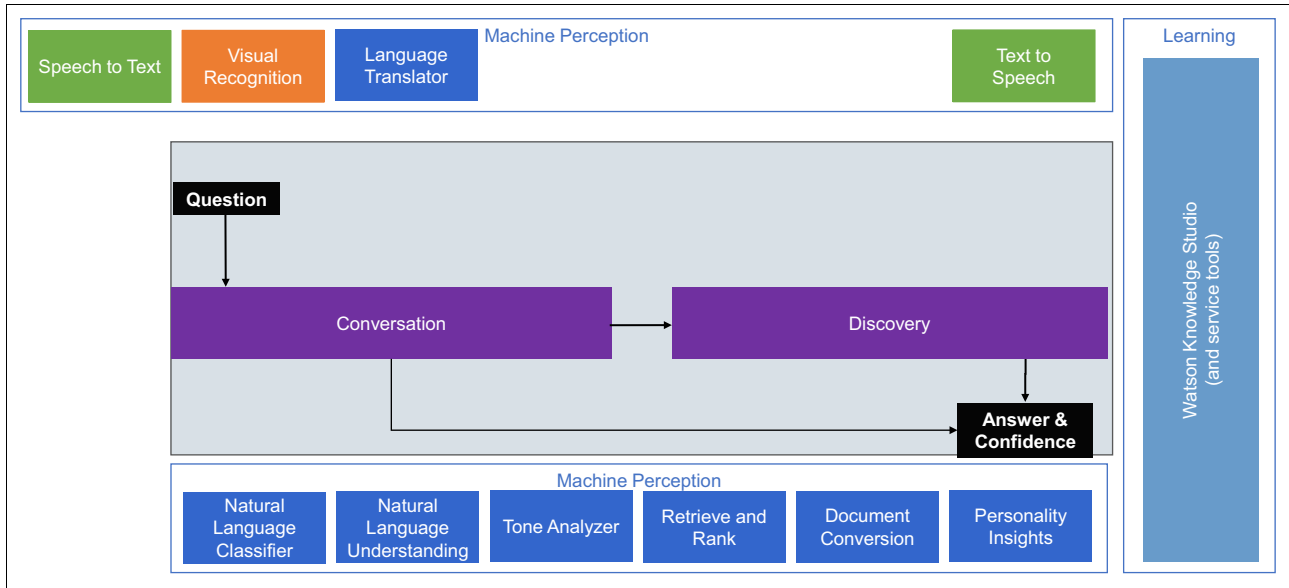


Figure 4-31 DeepQA pipeline, evolution to Watson Developer Cloud: Summary

## 4.7 References

- ▶ Watson Developer Cloud: Starter kits:  
<https://www.ibm.com/watson/developercloud/starter-kits.html>
- ▶ Watson services catalog:  
<https://www.ibm.com/watson/developercloud/services-catalog.html>



## Domain adaptation

One key element of cognitive systems is the capability to learn and adapt overtime. Rather than being explicitly programmed, cognitive 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 computer 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. This task is known as *domain adaptation*.

This chapter introduces the concept of domain adaptation and the processes that must be followed to adapt the various Watson services to specific domains.

The chapter lists the Watson services that can be trained, that is, can be adapted to specific domains, and those that cannot. For the Watson services that can be trained, this chapter provides an overview of the process that you have to follow to train each service.

The chapter also has an overview of Watson Knowledge Studio and the Watson services that required a model generated with Watson Knowledge Studio for adaptation to a new domain.

This chapter covers the following topics:

- ▶ Introduction to domain adaptation
- ▶ IBM Watson Developer Cloud and domain adaptation
- ▶ Watson Knowledge Studio

## 5.1 Introduction to domain adaptation

Like humans, cognitive systems need to 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 about 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, cognitive 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).

*Supervised learning* is one of the main styles of machine learning. Input data (also known as training examples) comes with a label, and the goal of learning is to be able 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.

Supervised learning is fairly common in classification problems because the goal is often to train the system to learn a classification system. For example, a spam classifier is a system that looks at the content of the email subject line to assess whether the received email is or is not spam. A spam classifier is trained with a labeled training set that contains enough samples of spam and legitimate emails. After being trained, the classifier should be able to distinguish spam emails from legitimate emails. Another example, input data could be past transactions for all customers of a bank. Transactions are labeled as either fraud or legitimate after processing them. The goal of learning is to predict for each new transaction whether it is a fraud or not.

Another way to adapt Watson to new domains is to ingest *knowledge bases*. A knowledge base (KB) is a data structure that represents structured information that can be consumed by a computer system to perform inferences. Knowledge bases can be implemented in several ways, ranging from simple dictionaries to first-order logic statements. Knowledge bases and supervised learning complement each other. In general, the more the provided knowledge, the less the needed training for the system to reach good performance on a task. Finding the right mix is the key for a successful domain adaptation process.

Adapting a cognitive system to a new closed-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 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.



## 5.2 IBM Watson Developer Cloud and domain adaptation

IBM Watson Developer Cloud is a cloud-hosted marketplace where application providers of all sizes and industries are able to tap into resources for developing applications powered by Watson services. Developers can combine the Watson services (and other services available in IBM Bluemix) with additional logic to build cognitive applications (Figure 5-1). Watson Developer Cloud includes a developer toolkit, educational materials, and access to Watson APIs. 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 infused with Watson cognitive computing intelligence.

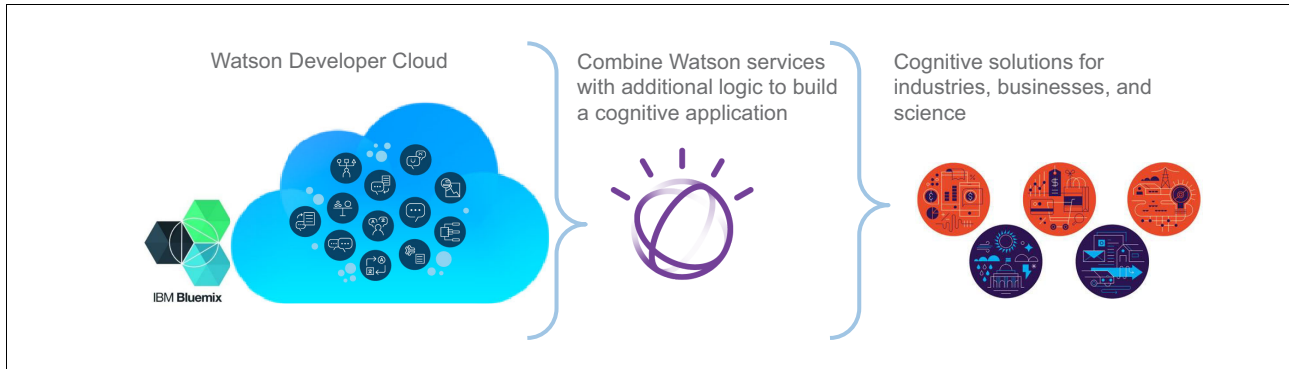


Figure 5-1 Building Cognitive Applications with IBM Watson Services

Information extraction analytics provided by Watson APIs are open domain, meaning that they are able to recognize named entities belonging to basic types such as company, person and location, but they are not provided with the ability of recognizing more specific distinctions, such as names of banks, insurance companies and their products. To become a subject matter expert in a specific industry or domain, some Watson services must be trained.

Figure 5-2 shows the Watson services that cannot be trained by the user; for these services, IBM is in charge of the required training.

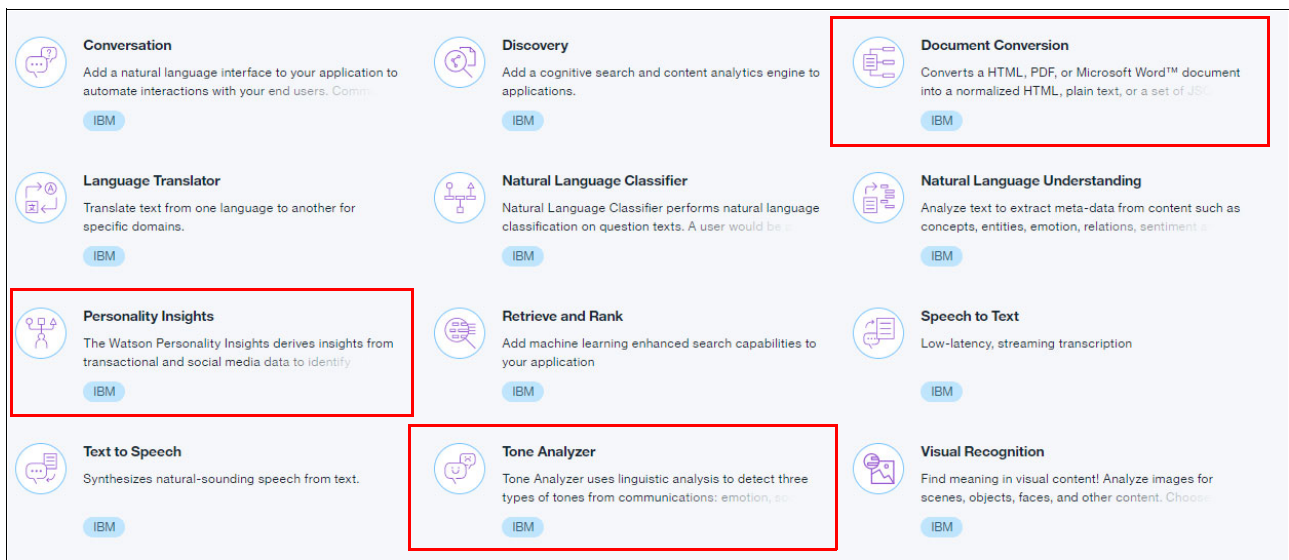


Figure 5-2 Watson services that cannot be trained by users

Because most of the Watson services are based on a supervised machine learning approach, a possibility is to train them, providing manually labeled data. For example, to enhance Watson understanding of financial text, a corpus of financial news can be ingested in Watson, and subject matter experts can be asked to label occurrences of banks, insurances, products, and their relations.

Image recognition can also be trained to recognize items, for example, company logos in pictures. Natural Language Classifier can be trained to identify financial news from blog posts. Machine translation can be trained to reduce the error in translating financial news across languages. The more training data is collected, the better the expected accuracy of the Watson services in recognizing entities and relations of the desired type.

Figure 5-3 shows the Watson services that you can train in order to adapt them to a closed-domain.

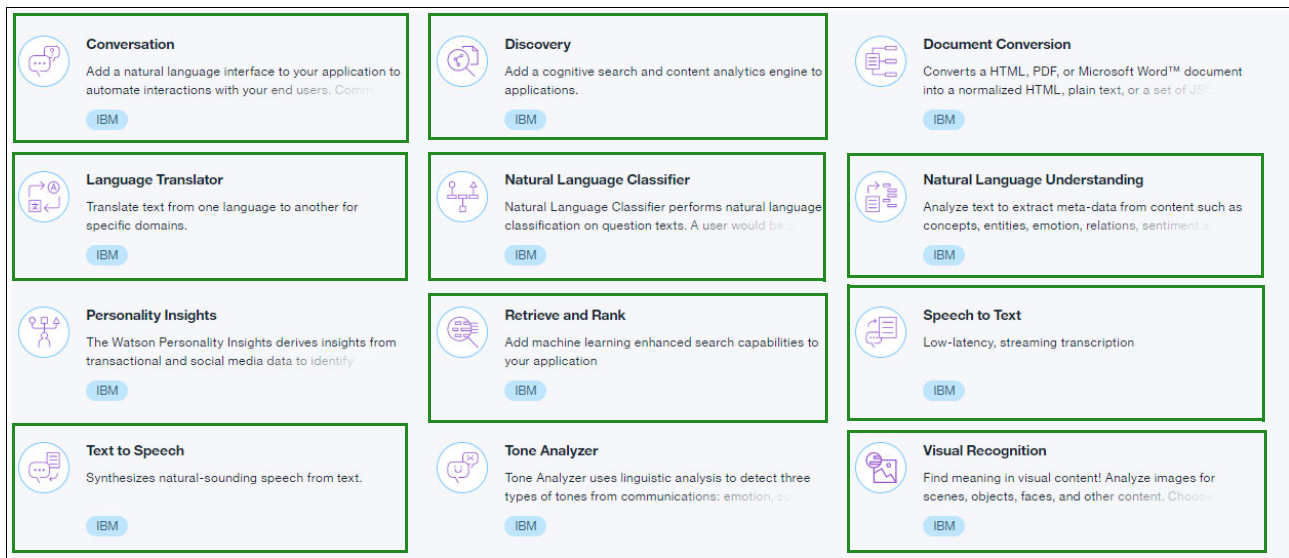


Figure 5-3 Watson services that can be trained by users

## 5.2.1 Watson Conversation

With the IBM Watson Conversation service, you can create an application and user agents that understand natural language input and communicate with your users simulating a real human conversation. Conversation service uses deep learning techniques to respond to your customers in a way that simulates a conversation between humans. With the advent and increasing popularity of chatbots, the Conversation service can be created once and can be made available across various chat platforms, such as Facebook Messenger, Slack, Twitter Direct Messages (DMs), and so on.

The service provides web tooling that enables you to configure how your bot behaves by using three key concepts (Figure 5-4 on page 65):

- # intents**            The purpose of a user’s input; what the user wants to achieve.
- @ entity**            A term or object that is relevant to the intent. Provides the context.
- Dialog**                Enables the service to respond to users; based on intents and entities.

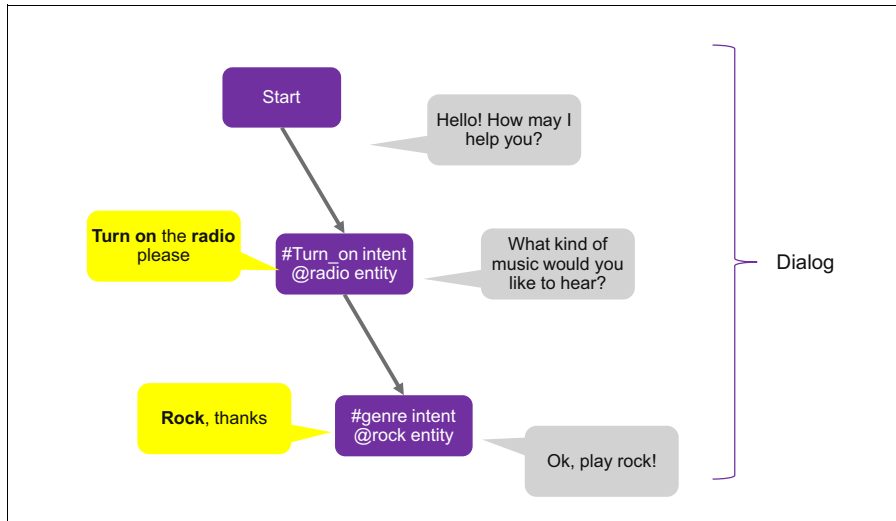


Figure 5-4 Conversation service: Showing intent, entity, dialog

You must train the service to recognize what the user is saying and respond accordingly. You do that by providing examples of how the user might phrase an *intent*, for example, to turn something off, ask for directions to a train station, ask what the weather will be tomorrow, and so on.

Next, you train the service to recognize key pieces of input that will determine how it should respond to users. These *entities*, marked with the at sign (@), are categories of words and phrases that influence the service responses. They might consist of specific appliances your users might need to control, places they might want directions to, and so on.

Finally, you use *dialog* to enable the service to respond to users based on the intent and entities recognized in their queries. Using dialogue nodes you can instruct the service to give simple answers when it detects certain intents, to ask clarifying questions when it is missing key information, or to guide users through more elaborate processes using advanced capabilities of the Conversation service.

One of the major challenges in developing a conversational interface is anticipating every possible way in which your users will try to communicate with your chatbot. The *Improve* component of the Conversation service provides a history of conversations with users. You can use this history to improve your chatbot's understanding of user input.

Figure 5-5 shows an overview of the steps to adapt the Conversation service.

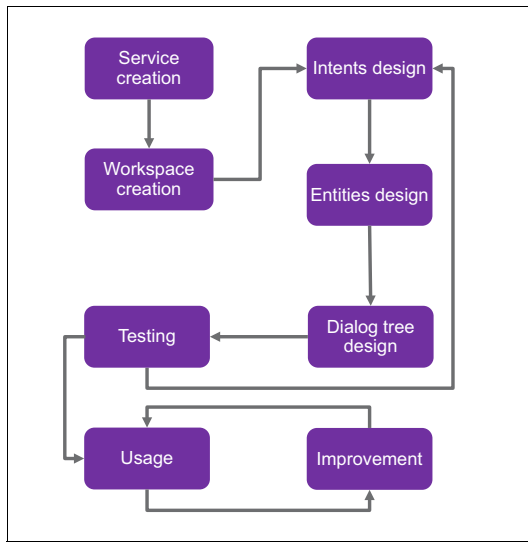


Figure 5-5 Conversation service adaptation

These are the steps for using the Conversation service and adapting it to your domain:

1. Create a Conversation service instance.
2. Create a workspace in a Watson Conversation service instance.
3. Train the Conversation service instance to recognize concepts in the user input (intents and entities):
  - Train the Conversation service instance with natural language examples of each possible *intent*. At least five examples are required for minimal training. However, providing many examples will give more accurate results, especially if they are varied and representative of possible input from users.
  - Train the Conversation service instance with natural language examples of each possible *entity*. Add as many synonyms as you expect your user to possibly use. The *Improve* interface will allow you to refine this process later, adding more synonyms as you test your dialog.
4. Create a conversation flow of the stages of the dialog. Use logical conditions evaluating the concepts identified in the user's reply.
5. Test your dialog in the embedded chat in the Conversation workspace. You can monitor how the Watson Conversation service interprets the dialog, what intents and entities it detects, and improve its training data in real time.

For more information, see the following resources:

- ▶ [Conversation](#)
- ▶ [Watson Conversation Service Overview](#) video
- ▶ *Building Cognitive Applications with IBM Watson Services: Volume 2 Conversation*, SG24-8394

## 5.2.2 Watson Language Translator

The Watson Language Translator service translates text from one language to another. It can be used by any application that can benefit from real-time, domain-specific translation capabilities.

The following linguistic models are provided with the Watson Language Translator service:

- ▶ News

Targeted at news articles and transcripts. Translate English to and from Arabic, Brazilian Portuguese, French, German, Italian, Japanese, and Spanish. You can also translate Spanish to and from French.

- ▶ Conversational

Targeted at conversational colloquialisms. Translate English to and from Arabic, Brazilian Portuguese, French, Italian, and Spanish.

- ▶ Patents

Targeted at technical and legal terminology. Translate Brazilian Portuguese, Chinese, and Spanish to English.

The Language Translator service can be trained over time to provide better accuracy when running translations. To do this, the service needs to learn from previous translations. Watson Language Translator takes specific terms and phrases into account, such as the names of people or products to ensure that they are translated correctly.

Watson Language Translator service provides a mechanism to train the service to perform domain-specific translations by customizing the existing models. It provides a useful function to update the existing models to add context and improve their quality from situation to situation.

For example, perhaps you are creating a translator for customer support and you have company-specific terms that you want handled in a certain way in conversations. Or you are creating a way for your engineers in one country to look up patent data in another language, and you usually 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.

The provided translation models are updated by adding an input source file. Customization depends on *type* and *content* of the input.

The inputs are in either of the following file formats:

- ▶ Translation Memory Exchange (TMX) file, which has a [specific format](#). TMX is an XML specification that is designed for machine-translation tools.
- ▶ Plain text file, which holds a large body of text

Both files must be UTF-8 encoded. Watson Language Translator service currently provides three ways of inputting a source to customize the translation models:

► **Forced glossary**

Forced glossary is a collection of terms and phrases with their translations in the target language. Forced glossaries *replace* the existing terms with their translation from those in the input file. Forced glossaries are used in the TMX format with the Language Translator service.

► **Parallel corpus**

Parallel corpus is used for a wide range of applications outside of Language Translator, including building a new translation model from scratch. In the scope of the Language Translator service, it contains pairs of terms or phrases that serve as alternate translation suggestions that you want the translation service to consider. It is used to enhanced a provided model to add terms and contexts in the form of phrases that might not be present in original model.

In contrast to the forced glossary, parallel corpuses are used to train the existing models, adding the terms and phrases from the input file to the existing training data rather replacing it. They do not override the original domain data.

The parallel corpus is also used in the TMX format with Language Translator. To successfully train a custom model, a parallel corpus document must contain a minimum of 5,000 term and translation pairs.

► **Monolingual corpus**

Monolingual corpus is a UTF-8 encoded plain text file that contains a body of text in the target language and that is related to what you are translating.

A monolingual corpus serves as a language sample that the service can evaluate and use to improve overall translation quality, for example, to make it more human-like, fluent, and natural. To successfully train a custom model, a monolingual corpus document must contain a minimum of 1,000 sentences.

Figure 5-6 shows an overview of the Language Translator adaptation flow.

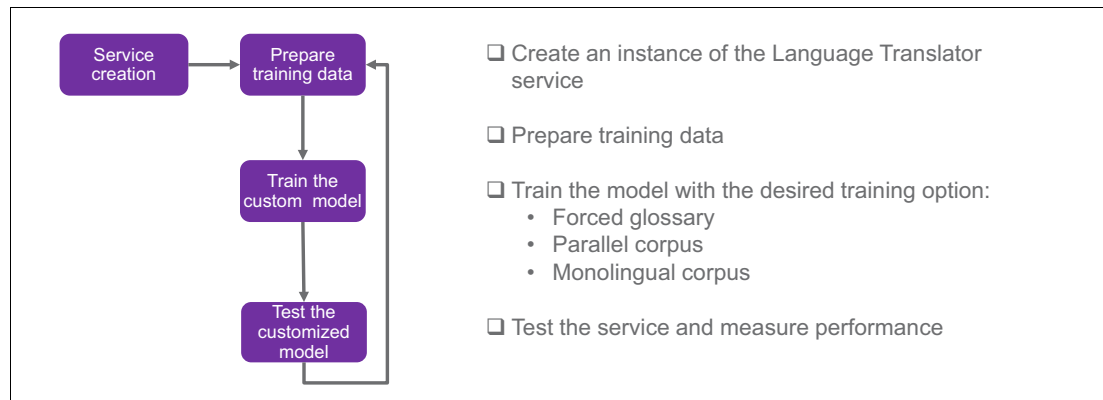


Figure 5-6 Language Translator service adaptation

For more information, see the following resources:

- [Language Translator](#)
- [Language Translator Service by IBM Watson](#) video
- *Building Cognitive Applications with IBM Watson Services: Volume 5 Language Translator*, SG24-8392

## 5.2.3 Watson Natural Language Classifier

The Natural Language Classifier service applies cognitive computing techniques to return best matching predefined classes for short text inputs, such as a sentence or phrase.

Figure 5-7 provides an overview of the four steps that are included in the process of creating and using the classifier.

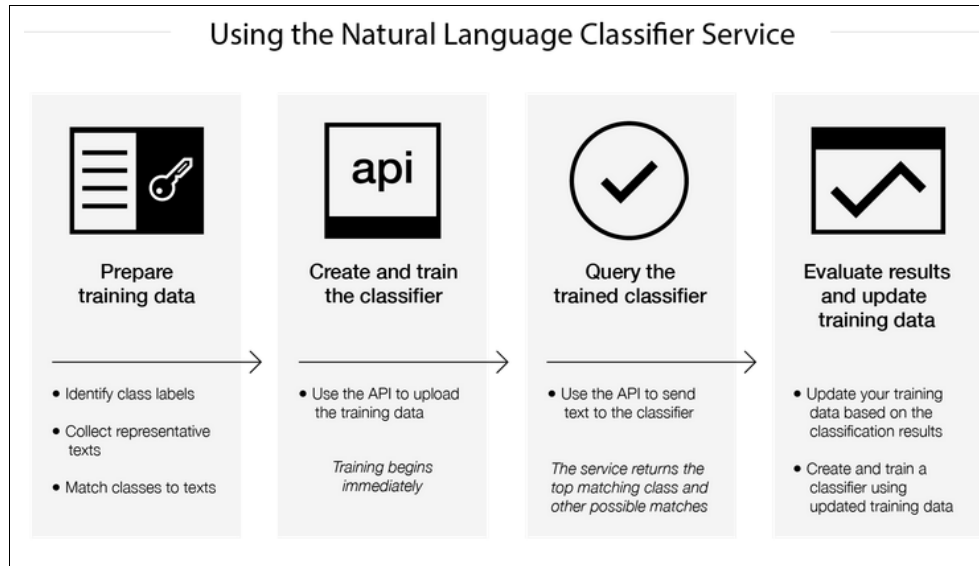


Figure 5-7 Using the Natural Language Classifier service: Process steps

To use the Natural Language Classifier service in your application, you must train the classifier following these steps:

1. Prepare training data
2. Create and train the classifier
3. Query the trained classifier
4. Evaluate results and update the data

### Prepare training data

To prepare the training data, follow these steps:

1. Identify class labels.

Class labels represent the result labels that describe the intent of the input text. Class labels are the output of a trained classifier.

2. Collect representative text.

Gather representative texts for each class label for training purposes. These texts show the classifier examples for each class and serve as training data. These examples should be similar to the actual text input that will be provided to the classifier in production.

3. Match classes to text.

Create the training data by matching text with their respective classes. To train the classifier, you prepare a training CSV file that is used when the classifier is created.

Table 5-1 shows the input text and corresponding class label for a simple example.

Table 5-1 Training data to create a CSV file

Input text	Class label
How much does it cost to get an occupational health card	Health
What are steps required to get a health card	Health
I want to be immune from Hepatitis B	Health
I need to know regulations for importing animal/veterinary products into the Markets	VeterinaryHealth
Where can I adopt a pet from a shelter	VeterinaryHealth
Where can someone obtain health cards for veterinary	VeterinaryHealth
How to get a post mortem report for my pet	VeterinaryHealth

Example 5-1 shows the CSV file created from Table 5-1.

Example 5-1 Training data in CSV format

---

```

How much does it cost to get an Occupational health card,Health
What are steps required to get a health Card,Health
I want to be immune from Hepatitis B,Health
I need to know regulations for importing animals/veterinary products into the
Markets,VeterinaryHealth
Where Can I adopt a pet from a shelter,VeterinaryHealth
Where can someone obtain Health cards for veterinary,VeterinaryHealth
How to get a post mortem report for my pet,VeterinaryHealth

```

---

### Create and train the classifier

Before you can create a classifier, a Natural Language Classifier service instance must be created. After creating the Natural Language Classifier service instance, create a classifier that is associated with the service instance. Specify the classifier name and training CSV file, and then upload the training CSV file.

This step is called *bootstrap classification*. The bootstrap classification (Figure 5-8), can be validated by subject matter experts (SMEs) for accuracy by using other data, called test data, and if necessary correcting classification problems.

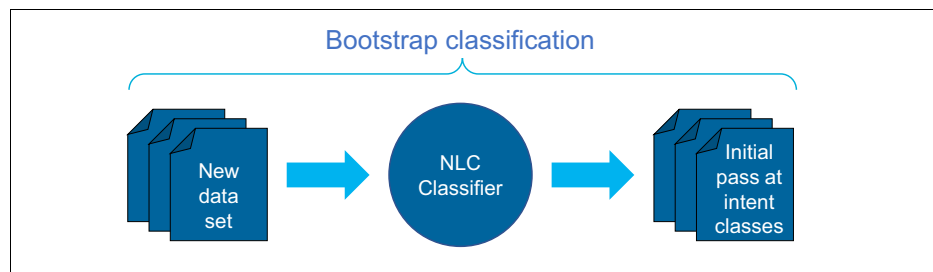


Figure 5-8 Bootstrap classification

This step is highly sensitive to good training data provided from the prepare data step and can be continuously improved depending on the target accuracy level, using other data sets.



## Query the trained classifier

After the classifier is trained, you can query it. The API returns a response that includes the name of the class for which the classifier has the highest confidence. Other class-confidence pairs are listed in descending order of confidence. The confidence value represents a percentage, and higher values represent higher confidences.

## Evaluate results and update the data

The first approach to evaluation is validation by SMEs and adjusting the classifier if accuracy is not aligned with the desired outcome. You can also include customer feedback, providing a way for users to input their feedback about the classification results. Figure 5-9 illustrates an overview of the process.

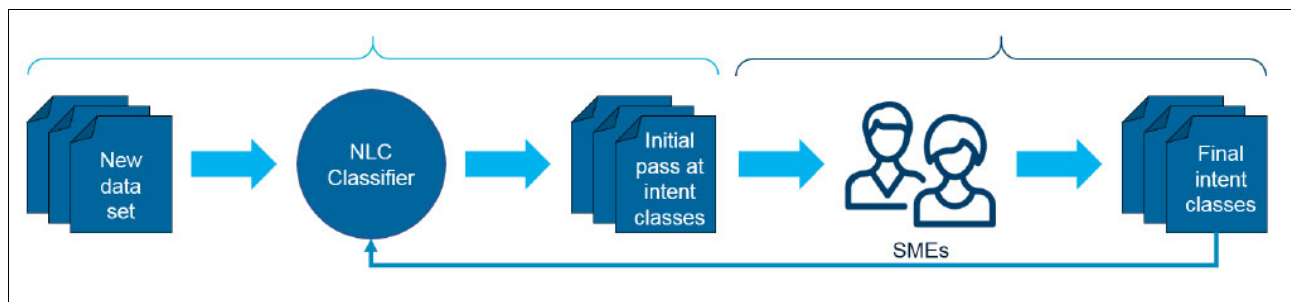


Figure 5-9 Manual validation of classifier

The objective of this step in the process is to improve the results returned by the classifier:

1. Detect wrong or weak confidence cases for user input text.
2. Change or restructure user's phrases into generic representative text.
3. Match text to their corresponding class label.
4. Add new text to the original training data and create a new classifier.
5. Repeat this cycle when quality of classification drops to a certain lower limit.

For more information, see the following resources:

- ▶ [Natural Language Classifier](#)
- ▶ [IBM Watson Natural Language Classifier](#) video
- ▶ *Building Cognitive Applications with IBM Watson Services: Volume 4 Natural Language Classifier*, SG24-8391

## 5.2.4 Watson Retrieve and Rank

Retrieve and Rank service can surface the most relevant information from a collection of documents. The purpose of the Retrieve and Rank service is to help you find documents that are more relevant than those that you might get with standard information retrieval techniques.

The primary users of the Retrieve and Rank service are customer-facing professionals, such as support staff, contact center agents, and field technicians. Examples of using Retrieve and Rank might include an experienced technician who can quickly find solutions from dense product manuals and a help desk agent who can also quickly find answers to improve average call handle times.

The Retrieve and Rank service is ready for immediate use as delivered by IBM but it can also be customized to improve the results.

The Retrieve and Rank service combines two information retrieval components in a single service: the power of *Apache Solr* and a sophisticated *machine learning* capability. This combination provides users with more relevant results by automatically reranking them by using these machine learning algorithms. The Retrieve component is based on Apache Solr. The Rank component uses machine learning techniques and it is the component that can be trained by the user to perform adaptation to a specific domain.

Figure 5-10 shows the process of creating and using the Retrieve and Rank service.

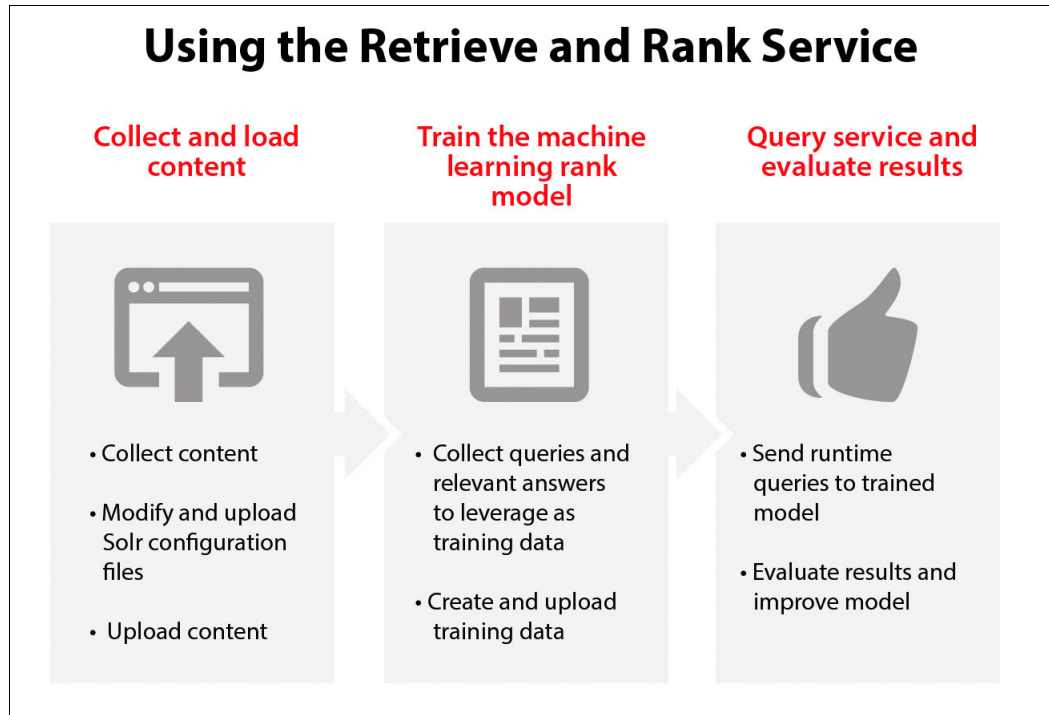


Figure 5-10 Using the Retrieve and Rank service

Figure 5-11 shows an overview of the steps for using the Retrieve and Rank service.

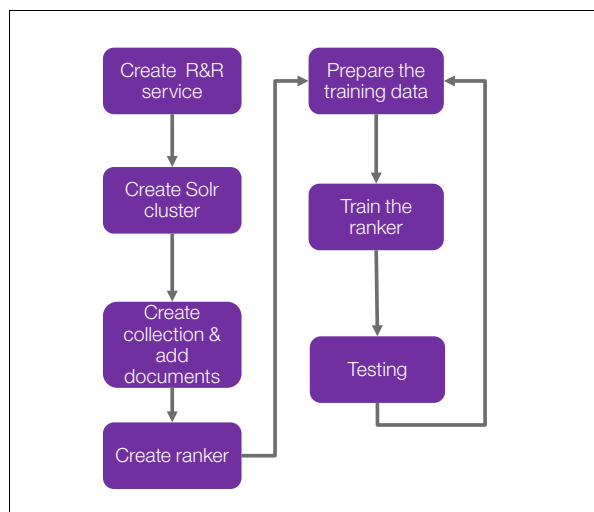


Figure 5-11 Adaptation of the Retrieve and Rank service

Perform these steps:

1. Create the Retrieve and Rank service instance in Bluemix.
2. Create a Solr cluster.

A Solr cluster manages your search collections, which you will create later.

3. Create a collection and add the documents that you will search.

A Solr collection is a logical index of the data in your documents. A collection is a way to keep data separate in the cloud. In this stage, you create a collection, associate it with a configuration, and upload and index your documents.

4. Create and train the ranker.

To return the most relevant documents at the top of your results, the Retrieve and Rank services uses a machine learning component called a *ranker*. You send queries to the trained ranker. The ranker learns from examples before it can rerank results from queries that it has never seen before. Collectively, the examples are referred to as *ground truth*.

Varies ways exist to train the Retrieve and Rank service: manually using the API, in semi-automatic mode using provided scripts, or by using a web UI. Regardless the training method used, the user always has to build the ground truth to drive the training.

5. Retrieve some answers.

While you are waiting for the ranker to finish training, you can search your documents. This search, which uses the *Retrieve* component of the Retrieve and Rank service, does not use the machine learning ranking features. It is a standard Solr search query. Your query returns the 10 most relevant results from Solr.

6. Rerank the results.

After the ranker finishes training, query the ranker to review the reranked results, now that the ranker is trained. The query returns your reranked search results in JSON format. You can compare these results against the results you got with the simple search in step 5.

After evaluating the reranked search results, you can refine them by repeating steps 4, 5, and 6. You can also add new documents, as described in step 3, to broaden the scope of the search. Repeat the process until you are completely satisfied with the results. This can require multiple iterations of refining and reranking.

For more information, see [Retrieve and Rank](#).

## 5.2.5 Watson Visual Recognition

The Watson Visual Recognition service uses deep learning algorithms to analyze images for scenes, objects, faces, and other content. The response includes keywords that provide information about the content. A set of built-in classes provides highly accurate results without training.

You can also train and create a custom classifier. With a custom classifier, you can train the Visual Recognition service to classify images to suit your business needs. By creating a custom classifier, you can use the Visual Recognition service to recognize images that are *not* available with pre-trained classification.

The Watson Visual Recognition service can learn from example images that you upload to create a new classifier. Each example file is trained against the other files uploaded when you create the classifier and positive examples are stored as classes. These classes are grouped to define a single classifier, but return their own scores.

Figure 5-12 shows the process for using the Watson Visual Recognition service with a custom classifier.

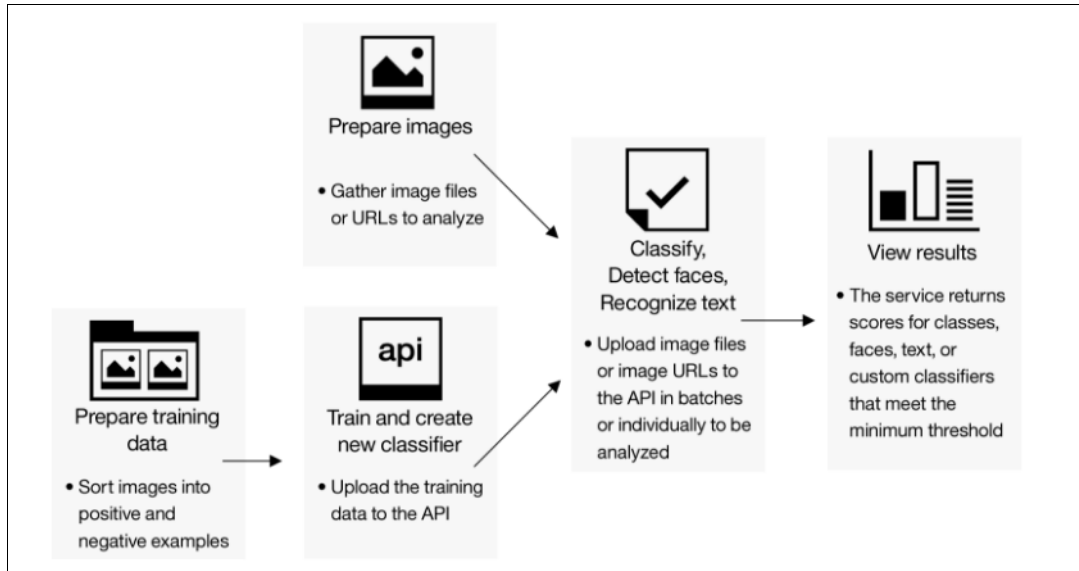


Figure 5-12 Visual Recognition process with custom classifier

A new custom classifier can be trained by several compressed (.zip) files, including files containing positive or negative examples of images (.jpg or .png). You must supply at least two compressed files, either two positive example files or one positive and one negative example file.

Figure 5-13 shows an example of training images to create a specialized Visual Recognition classifier to recognize and classify breed of dogs. The user prepares ZIP files with positive examples for dog breeds such as Beagle, Golden Retriever, and Husky. The user may also prepare a ZIP file containing negative examples of animals that are not dogs, for example, cats, lions, jaguars, and so on.

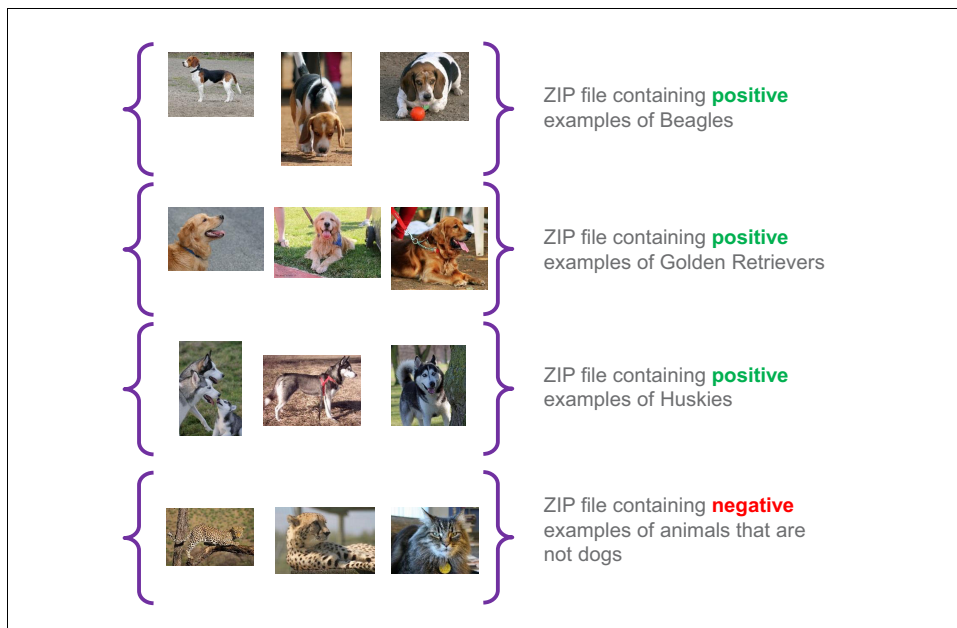


Figure 5-13 Example of Visual Recognition training images

Compressed files containing positive examples are used to create *classes* that define what the new classifier is. 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. No limit exists for the number of positive example files that you can upload in a single call.

The compressed file containing negative examples is not used to create a class within the created classifier, but does define what the new classifier is not. Negative example files should contain images that do not depict the subject of any of the positive examples. You can specify only one negative example file in a single call.

Figure 5-14 shows steps for creating and training a specialized Visual Recognition classifier.

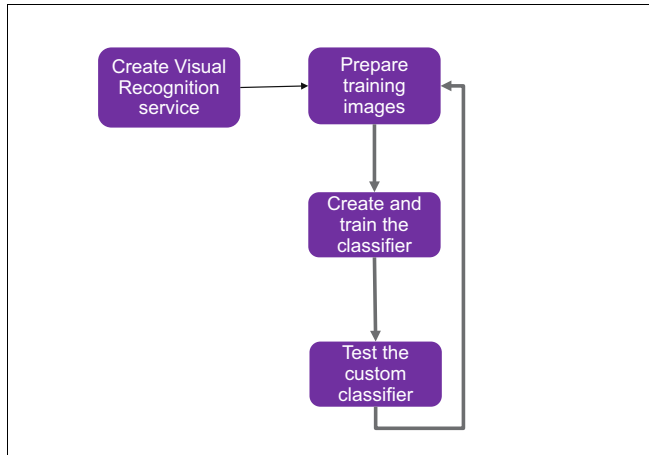


Figure 5-14 Adaptation of the Visual Recognition service

Figure 5-14 shows these steps:

1. Prepare training data.  
Gather image files to use as positive and negative example training data.
2. Create and train the classifier.  
Specify the location of the training images and call the Visual Recognition API to create the custom classifier.
3. Test the custom classifier.  
Classify images with the new custom classifier and measure the classifier performance.

For more information, see these resources:

- ▶ [Overview of the IBM Watson Visual Recognition service](#)
- ▶ [Guidelines for training classifiers](#)
- ▶ *Building Cognitive Applications with IBM Watson Services: Volume 3 Visual Recognition, SG24-8393*

## 5.2.6 Watson Speech to Text

The Speech to Text service converts speech into readable text according to the language that the user specifies. The service is capable of transcribing speech from various languages and audio formats to text with low latency. This service uses speech recognition capabilities to convert Arabic, English, Spanish, French, Brazilian Portuguese, Japanese, and Mandarin speech into text.

The Speech to Text service was developed for a broad, general audience. The service's base vocabulary contains many words that are used in everyday conversation. This general model provides sufficiently accurate recognition for a variety of applications, but it can lack knowledge of specific terms that are associated with particular domains.

To customize Speech to Text for a particular domain, a new language model is needed to provide the nuances in that domain in terms of vocabularies and word pronunciations.

With the language model customization interface, you can improve the accuracy of speech recognition for domains such as medicine, law, information technology, and others. Customization lets you expand and tailor the vocabulary of a base model to include domain-specific data and terminology. After you provide data for your domain and build a custom language model that reflects that data, you can use the model with your applications to provide customized speech recognition.

The typical usage model for working with Speech to Text customization includes these steps:

1. Create a custom language model.

You use the **POST /v1/customizations** method to create a new custom language model.

2. Add data from corpora to the custom language model.

The preferred way to add data (domain-specific words) to a custom model is by adding one or more corpora to the model.

A corpus is a plain text document that uses terminology from the domain in context. The service builds a vocabulary for a custom model by extracting terms from corpora that do not exist in its base vocabulary. You can add multiple corpora to a custom model.

You use the **POST /v1/customizations/{customization\_id}/corpora/{corpus\_name}** method to add a corpus to a custom model.

Example 5-2 shows an abbreviated corpus for the healthcare domain. A corpus file is typically much longer.

*Example 5-2 Abbreviated sample corpus for health care domain*

---

How Is Coronary Microvascular Disease Treated?  
If you're diagnosed with coronary MVD and also have anemia, you may benefit from treatment for that condition.  
Anemia is thought to slow the growth of cells needed to repair damaged blood vessels.  
What causes autoimmune hepatitis?  
A combination of autoimmunity, environmental triggers, and a genetic predisposition can lead to autoimmune hepatitis.  
What research is being done for Spinal Cord Injury?  
The National Institute of Neurological Disorders and Stroke NINDS conducts spinal cord research in its laboratories at the National Institutes of Health NIH.  
NINDS also supports additional research through grants to major research institutions across the country.  
What is Osteogenesis imperfecta OI?  
Osteogenesis imperfecta OI is a rare genetic disorder that, like juvenile osteoporosis, is characterized by bones that break easily, often from little or no apparent cause.

---

3. Add words to the custom language model.

You can add custom words to a model individually as well as via corpora. Although adding corpora is the preferred means of adding words to a custom language model, you can also

add custom words to the model directly. In addition, you can use the same methods to modify custom words extracted for the model from corpora. The methods let you specify the pronunciation of words and how they are displayed in a speech transcript.

Example 5-3 shows the `POST /v1/customizations/{customization_id}/words` method to add multiple words at one time. You pass a JSON object that provides information about each word through the body of the request. The example adds two words to the custom model: HHonors and IEEE.

*Example 5-3 Adding multiple words to a custom language model*

---

```
curl -X POST -u {username}:{password}
--header "Content-Type: application/json"
--data '{"words": [
  {"word": "HHonors", "sounds_like": ["hilton honors", "h honors"],
  "display_as": "HHonors"},
  {"word": "IEEE", "sounds_like": ["i triple e"]}]}'
https://stream.watsonplatform.net/speech-to-text/api/v1/customizations/74f4807
e-b5ff-4866-824e-6bba1a84fe96/words"
```

---

#### 4. Train the custom language model.

After you add words to the custom model (from corpora, individually, or both), you must train the model on the custom words in your domain-specific vocabulary. Training prepares the custom model for use in speech recognition. The model does not use the new words you add until you train it on the new data. You train a custom model by using this method:

```
POST /v1/customizations/{customization_id}/train
```

#### 5. Use the custom language model in a recognition request.

After you train your custom model, you can use it with a recognition request. The results of the request reflect the enhanced vocabulary available with the custom model.

For more information, see the following resources:

- ▶ [About Speech to Text](#)
- ▶ [Using customization](#)
- ▶ [Speech to Text API reference](#)
- ▶ *Building Cognitive Applications with IBM Watson Services: Volume 6 Speech to Text and Text to Speech*, SG24-8388

## 5.2.7 Watson Text to Speech

Watson Text to Speech is a speech synthesizer API that converts written text into audible speech. It is multilingual, so it accepts text as input and outputs an audio file in various languages. The input text can be plain text or written in Speech Synthesis Markup Language (SSML). Additionally, it outputs various speaking styles, pronunciation, pitch, and speaking rate. The *Voices* feature synthesizes text to audio in a variety of languages, including English, French, German, Italian, Japanese, Spanish, and Brazilian Portuguese. The service offers at least one male or female voice, sometimes both, for each language and different dialects, such as US and UK English and Castilian, Latin American, and North American Spanish. The audio uses appropriate cadence and intonation.

When you synthesize text with the Text to Speech service, the service applies language-dependent pronunciation rules to convert the ordinary (orthographic) spelling of each word to a phonetic spelling.

The service's regular pronunciation rules work well for common words. However, they might yield imperfect results for unusual words such as special terms with foreign origins, personal or geographic names, and abbreviations and acronyms. When your application's lexicon includes such words, the service can produce imperfect pronunciations. To address this issue, the service provides a customization interface that you can use to specify how it pronounces unusual words that occur in your input.

The customization interface of the Text to Speech service lets you 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*, or just a *custom model*. Each entry in a custom voice model consists of a word/translation pair. A word's translation tells the service how to pronounce the word when it occurs in input text.

The customization interface provides methods to create and manage your custom models, which the service stores permanently. The interface includes methods to add, modify, delete, and query the words and translations in a custom model.

The typical usage model for working with Speech to Text customization includes these steps:

1. Create a custom model.

To create a new custom model, you use the **POST customizations** method. A new model is always empty when you first create it; you must use other methods to populate it with word/translation pairs.

2. Add words to a custom model.

After a custom model is created, the next step is to add contents in the form of word/translation pairs to define how specified words are to be pronounced during synthesis. The definitions override the service's default regular pronunciation rules.

To add a single word/translation pair to a custom model, you use the following method, where you specify the word to be added in the URL of the method:

```
PUT customizations/{customization_id}/words/{word}
```

You provide the translation for the word as a JSON object with a single translation attribute. Adding a new translation for a word that already exists in a model overwrites the word's existing translation.

You specify the translation for a word in a custom voice model via one of two methods: *sounds-like* or *phonetic*. You can use both methods for entries in the same custom model, but a single custom model can include no more than 20,000 entries.

Example 5-4 adds the word *IEEE* to a custom model by using the *sounds-like* method.

*Example 5-4 Adding the word IEEE to a custom voice model by using the sounds-like method*

---

```
curl -X PUT -u {username}:{password}
--header "Content-Type:application/json"
--data "{\"translation\":\"I triple E\"}"
"https://stream.watsonplatform.net/text-to-speech/api/v1/customizations/{customization_id}/words/IEEE"
```

---

3. Use the custom model.

After you create a custom model and populate it with word/translation pairs, you use it by passing its GUID with the `customization_id` query parameter of the HTTP GET or POST synthesizer method or the WebSocket synthesizer method.



Example 5-5 uses the HTTP GET version of the `synthesize` method to generate a Waveform Audio File Format (WAV) file named `ieee-orig.wav` with the default pronunciation for IEEE that is based on the service's regular pronunciation rules.

*Example 5-5 Using the custom model*

---

```
curl -X GET -u {username}:{password}
--header "Accept: audio/wav"
--output ieee-new.wav
"https://stream.watsonplatform.net/text-to-speech/api/v1/synthesize?text=IEEE&customization_id={customization_id}"
```

---

For more information, see the following resources:

- ▶ [About Text to Speech](#)
- ▶ [Understanding customization](#)
- ▶ [Using customization](#)
- ▶ [Text to Speech API reference](#)
- ▶ *Building Cognitive Applications with IBM Watson Services: Volume 6 Speech to Text and Text to Speech*, SG24-8388

## 5.2.8 Watson Natural Language Understanding

Watson Natural Language Understanding is a new rendition of the `AlchemyLanguage` API, which was deprecated. With Natural Language Understanding service, you can analyze semantic features of input text and extract metadata from content such as categories, concepts, emotion, entities, keywords, relations, semantic roles, and sentiment.

By default, Watson Natural Language Understanding is trained on an open domain. With custom annotation models developed by using IBM Watson Knowledge Studio, you can further customize the service to identify domain-specific entities and relations in your content. When deployed to a Natural Language Understanding service instance, the custom model overrides the standard entity detection model.

The differences between Watson Natural Language Understanding and `AlchemyLanguage` are documented in [Migrating from AlchemyLanguage](#).

The languages supported by Watson Natural Language Understanding are documented in [Supported languages](#).

Figure 5-15 on page 80 depicts the flow that organizations can use to analyze their unstructured data, using Watson Natural Language Understanding. The figure also shows the option to deploy a custom model with Watson Knowledge Studio.

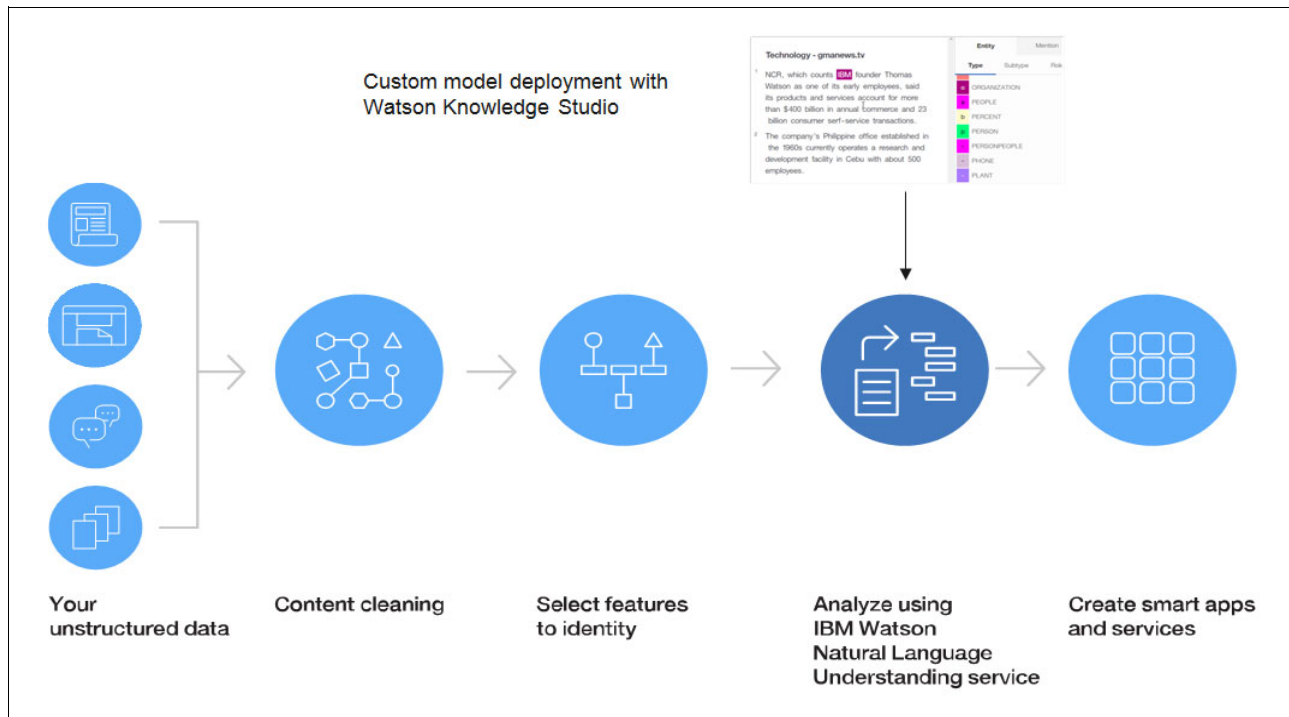


Figure 5-15 Analyzing unstructured data with Watson Natural Language Understanding

To create a custom model with Watson Knowledge Studio, see 5.3.2, “Example: Creating a machine learning model” on page 84. To deploy the custom model to a Natural Language Understanding service instance, see 5.3.3, “Deploying a machine-learning annotator to Watson Natural Language Understanding” on page 104.

For more information, see the following resources:

- ▶ [Overview of the IBM Watson Natural Language Understanding service](#)
- ▶ [Customizing](#)
- ▶ [Getting started with custom models](#)
- ▶ *Building Cognitive Applications with IBM Watson Services: Volume 7 Natural Language Understanding, SG24-8398.*

## 5.2.9 Watson Discovery

The Watson Discovery service provides developers with the ability to rapidly add a cognitive, search and a content analytics engine to applications in order to identify patterns, anomalies, trends, and insights that drive better decision making.

The Watson Discovery service provides a pipeline for ingesting, enriching and storing vast amounts of unstructured data. Watson Discovery service allows you to run queries by using its query API. In addition, Watson Discovery service includes a new feature that provides the ability to improve search results by training using documents with prior relevancy labels (relevancy ranking). These documents can be from your past customer interactions, chat logs, and forum responses. The relevancy ranking feature enables custom training based on ranking provided as part of the ground truth.

Enrichments in the Watson Discovery service can be customized for your domain by training a custom model using Watson Knowledge Studio and deploying it to the Discovery service.

Figure 5-16 depicts the flow to ingest, enrich, and query unstructured data by using the Watson Discovery service.

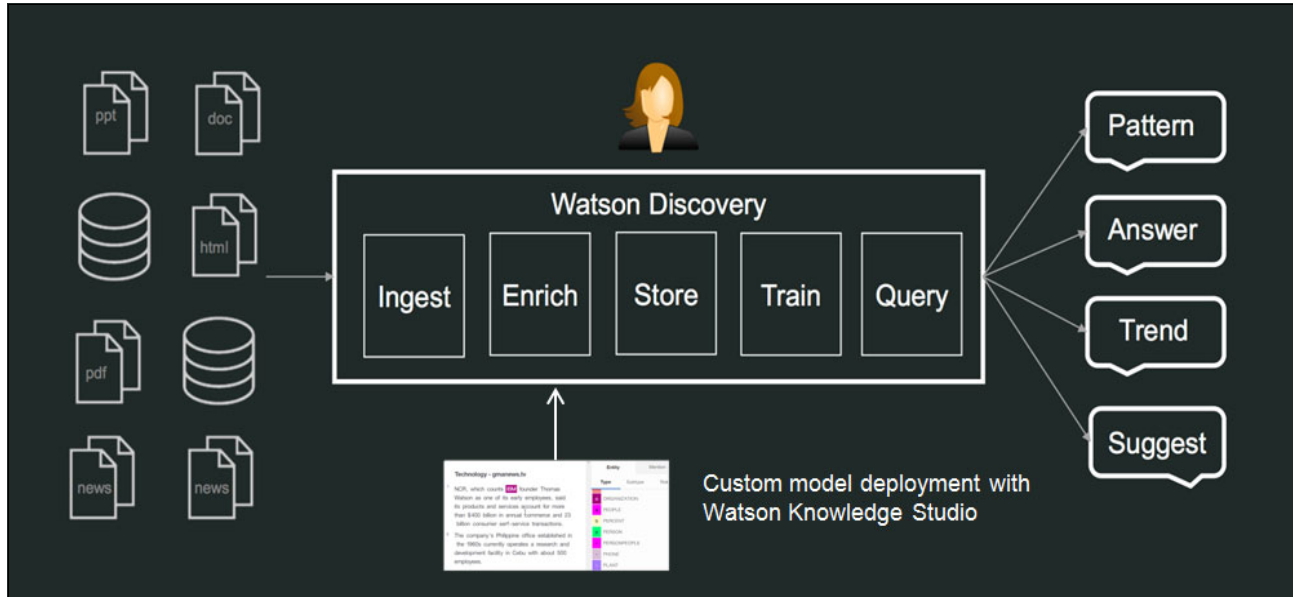


Figure 5-16 Ingest, enrich, and query unstructured data with Watson Discovery service

To create a custom model with Watson Knowledge Studio, see 5.3.2, “Example: Creating a machine learning model” on page 84. To deploy the custom model to the Watson Discovery service, see 5.3.4, “Deploying a machine-learning annotator to Watson Discovery” on page 107.

For more information, see the following resources:

- ▶ [Discovery](#)
- ▶ [Watson Discovery Service Overview](#) video

## 5.3 Watson Knowledge Studio

It used to be impossible for anyone without a PhD in machine learning to teach a computer to understand specialized terminology. If you have embraced cognitive computing you know the value of gaining insights from your unstructured data. But for specialized areas, in order to pinpoint the information you need, a cognitive system needs to be taught the unique language used. For example, you need to teach the system what is a *virus*, a *crash*, or a *worm* for your industry or domain and what are the relationships between these entities.

Traditional technology has relied on only programming or creating rules to customize natural language systems. Now with IBM Watson Knowledge Studio you can teach some Watson services about your specific interest area faster through examples without writing any code. Instead of requiring PhDs in AI, experts in any field can train Watson.

Watson Knowledge Studio is a cloud-based application that enables developers and domain experts to collaborate and create custom annotator components for unique industries and domains. These annotators can identify mentions and relationships in unstructured data and be easily administered throughout their lifecycle using one common tool. Annotator components can be deployed directly to IBM Watson Explorer, Watson Natural Language Understanding and Watson Discovery.

Watson Knowledge Studio users have trained Watson to identify critical content in order to stop cybersecurity attacks before disaster strikes, understand software support requests, and track down information about diseases. Across financial, legal, technical, and environmental industries applications using Watson Knowledge Studio can identify new opportunities, trends, and threats.

### 5.3.1 Watson Knowledge Studio domain adaptation overview

The primary purpose of Watson Knowledge Studio is to help create a model that understands domain specific linguistic nuances, meanings, and relations. It also provides a rule-based model to find entities in documents. These capabilities help Watson to become a subject matter expert in a given domain or industry.

The core to understanding domain expressed in natural language is often referred to as information extraction. Information extraction can be performed through both rule-based annotators and machine learning annotators. Rule-based annotators work on a codify set of rules, patterns and dictionaries. Machine learning represents a paradigm shift from traditional rule-based approaches. Watson Knowledge Studio is currently a machine learning tool which can use existing rule-based annotators dictionaries to train a machine learning model that can be used to adapt to the new domain.

Watson Knowledge Studio helps to train the model that identifies entities and relations specific to a domain through three main features:

- ▶ Mention detection, which is the most important ingredient for identifying entities and relationships of interest
- ▶ Relation detection
- ▶ Coreference resolution

Watson Knowledge Studio provides an interface to define these various entity types and relation types that you are interested in extracting from your information sources. It also provides the ability to train your machine learning annotator to be able to extract instances of the same entity types and relation types.

Domain adaptation starts with raw and representative documents from the domain. First you upload these documents to Watson Knowledge Studio and then use the Ground Truth Editor tool to label mentions of different entity types and relation types that are found in these documents. These annotations become the *gold standard*, also known as *ground truth*. The machine learning model is then trained based on this ground truth that has been established.

After the model is trained, you have to evaluate its performance based on a set of documents which act as the text corpus. If the model performs satisfactorily, then you can deploy the model. However if it does not perform satisfactorily then you go through more iterative cycles until the performance of the machine learning annotator is acceptable.

Figure 5-17 shows the typical workflow for creating a machine-learning annotator component in Watson Knowledge Studio.

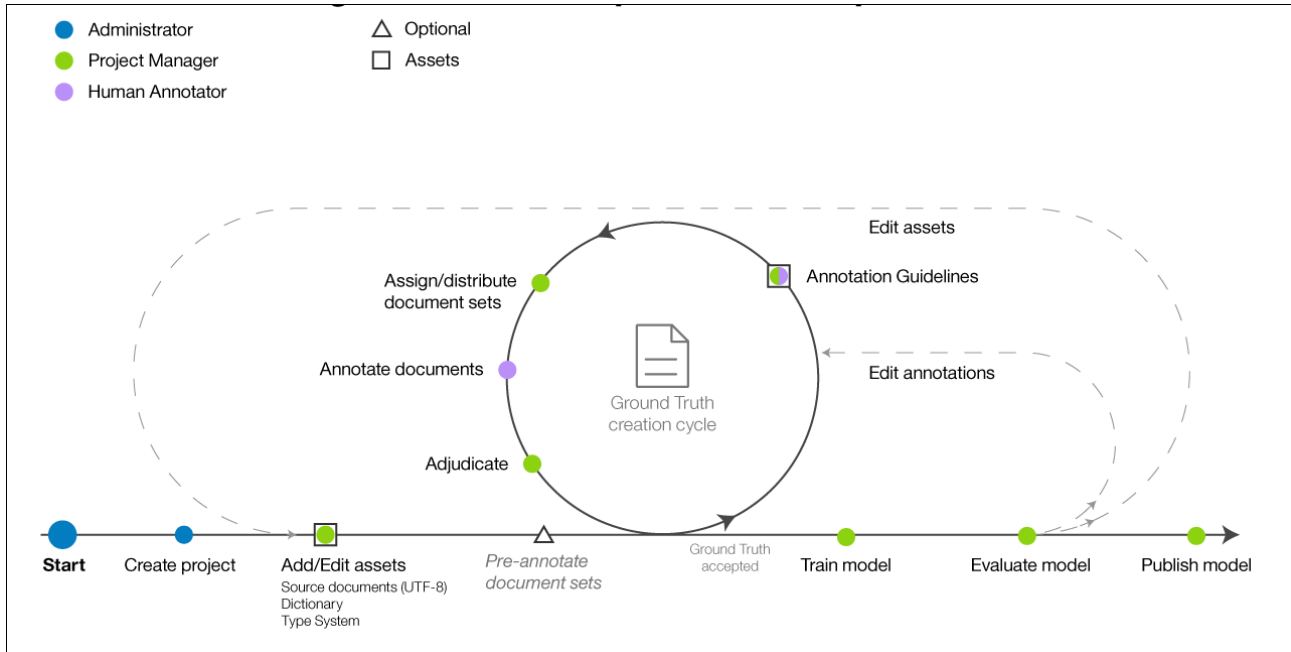


Figure 5-17 Machine-learning annotator component development workflow

All steps are performed by the project manager, except for the Annotate documents step, which is performed by the human annotator. Because human annotators are often subject matter experts, they might also be consulted during the creation of project resources, such as the type system.

Table 5-2 describes the implementation steps that are required in order to create a machine learning model with Watson Knowledge Studio and deploy it to Watson services.

Table 5-2 Creating a machine learning model with Watson Knowledge Studio overview

Step	Description
Assign user roles	An annotator component requires input from subject matter experts, project managers, and users who can understand and interpret statistical models. A user account must be created for each user who needs to log in to Watson Knowledge Studio (see “Assign user roles” on page 85).
Create a project	A project (see “Create a project” on page 85) contains the resources that are used to create the annotator component, including these: <ul style="list-style-type: none"> <li>▶ Type system (see “Create a type system” on page 86).</li> <li>▶ Source documents (see “Add documents for annotation” and “Create and assign annotation sets” on page 89).</li> <li>▶ Dictionaries (see “Add a dictionary” on page 87).</li> </ul>
Optional: Pre-annotate documents	Pre-annotate documents according to the terms in the project dictionaries or based on rules that you define (see “Optional: Pre-annotate with a dictionary-based annotator” on page 90).
Annotate documents	The project manager assigns annotation tasks to human annotators, configures the inter-annotator agreement threshold, and provides annotation guidelines for the human annotators to follow (see “Create an annotation task” on page 92).  Human annotators use the Ground Truth Editor to manually annotate documents (see “Annotate documents” on page 94).

Step	Description
Adjudicate and promote documents	Accept or reject the ground truth that was generated by human annotators and adjudicate any annotation differences to resolve conflicts. Accepted documents are promoted to ground truth (see “Adjudicate conflicts and promote documents to ground truth” on page 98).
Train the model	Create a machine learning annotator component (see “Create a machine-learning annotator” on page 100).
Evaluate the model	Evaluate the accuracy of the annotator component (see “Create a machine-learning annotator” on page 100).
Publish the model	Export or deploy the model (see 5.3.3, “Deploying a machine-learning annotator to Watson Natural Language Understanding” on page 104 and 5.3.4, “Deploying a machine-learning annotator to Watson Discovery” on page 107).

### 5.3.2 Example: Creating a machine learning model

This section provides an example to help you understand the process for building a machine-learning model that you can deploy and use with other Watson services (Watson Natural Language Understanding and Watson Discovery).

**Note:** For instructions and the artifacts to implement this example, see these tutorials:

- ▶ [Tutorial: Creating a project](#)
- ▶ [Tutorial: Creating a machine-learning model](#)

Implementing this example involves the following steps:

1. Assign user roles
2. Create a project
3. Create a type system
4. Add a dictionary
5. Add documents for annotation
6. Create and assign annotation sets
7. Optional: Pre-annotate with a dictionary-based annotator
8. Create an annotation task
9. Annotate documents
10. Adjudicate conflicts and promote documents to ground truth
11. Create a machine-learning annotator

## Assign user roles

The creation of an annotator component requires input from subject matter experts, project managers, and users who can understand and interpret statistical models. A user account must be created for each user who needs to log in to Watson Knowledge Studio.

Invite people to fill these roles:

- ▶ Human annotators

The human annotator is someone, typically a subject matter expert, who reviews domain documents to identify entities and relationships of interest to the domain. These users interact with the application in a limited way; they use the Ground Truth Editor to annotate a set of documents that have been assigned to them.

- ▶ Project manager

The project manager is someone who helps to facilitate the creation of annotator components by performing such tasks as creating project artifacts, and training, creating, and deploying models. For projects that build machine-learning annotators, they also manage the document annotation process by assigning document review tasks to human annotators, adjudicating annotation conflicts, and approving documents to add to the ground truth.

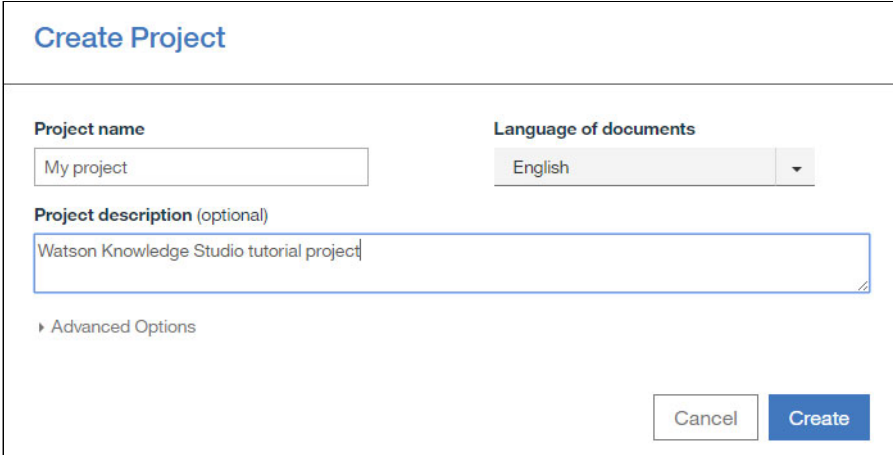
## Create a project

For each model that you want to build and use, you create a single project that contains the artifacts and resources needed to build the annotator component. You then train the annotator component to produce a custom model that can be deployed to an external service for use.

You can create a *machine-learning* model or a *rule-based* model. This example shows you how to create a machine-learning model. Machine-learning models use a statistical approach to finding entities and relationships in documents. This type of model can adapt as the amount of data grows.

The custom model will be used by other Watson services such as Natural Language Understanding and Discovery.

Figure 5-18 shows the Create Project window.



The screenshot shows a 'Create Project' dialog box. At the top, the title 'Create Project' is displayed in blue. Below the title, there are three main input areas: a 'Project name' text box containing 'My project', a 'Language of documents' dropdown menu currently set to 'English', and a 'Project description (optional)' text area containing 'Watson Knowledge Studio tutorial project'. Below the description text area is a link that says 'Advanced Options' with a right-pointing arrow. At the bottom right of the dialog, there are two buttons: a white 'Cancel' button and a blue 'Create' button.

Figure 5-18 Create Project window

After you click **Create** the project is created and opens automatically. You can now start configuring the project resources, such as the type system.

## Create a type system

A type system defines things that are interesting in your domain content that you want to label with an annotation. 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 annotator process manager typically works with subject matter experts for your domain to define the type system.

You can create or import a type system. To start a project, you might want to import 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.

To import a type system in this example, from within your project click **Type System** → **Import** → select or drag JSON file → **Import** (Figure 5-19).

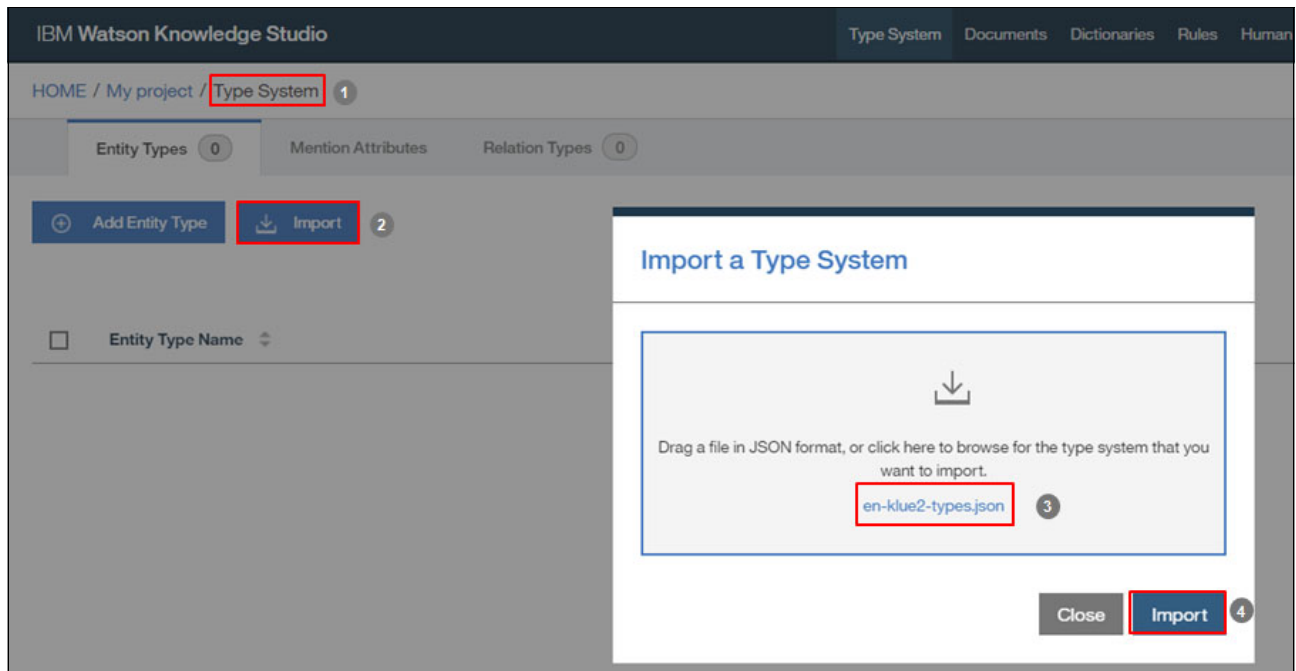


Figure 5-19 Import a type system



The imported type system is listed in the table (Figure 5-20).

The screenshot shows the IBM Watson Knowledge Studio interface. The top navigation bar includes 'Type System', 'Documents', 'Dictionaries', 'Rules', 'Human Annotation', and 'Annotator Component'. The breadcrumb path is 'HOME / My project / Type System'. Below the navigation, there are tabs for 'Entity Types' (52), 'Mention Attributes', and 'Relation Types' (2177). There are buttons for 'Add Entity Type', 'Import', and 'Export'. A search bar is present with the placeholder text 'Enter text to filter entries'. The main content is a table with the following columns: 'Entity Type Name', 'Roles', 'Subtypes', and 'Action'. The table lists several entity types with their respective roles and subtypes.

Entity Type Name	Roles	Subtypes	Action
<input type="checkbox"/> ORDINAL	ORDINAL ANIMAL		<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/> MONEY	MONEY AWARD	OTHER UNSPECIFIED	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/> EVENT_VIOLENCE	EVENT_VIOLENCE		<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/> PEOPLE	PEOPLE		<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/> TITLEWORK	TITLEWORK	OTHER	<a href="#">Edit</a> <a href="#">Delete</a>

At the bottom of the table, there is a pagination control showing 'First', '1', '2', '3', '4', '5', '...', and 'Last'.

Figure 5-20 Imported type system

### Add a dictionary

A dictionary groups together words and phrases that should be treated equivalently by an annotator component. In machine learning, a dictionary groups together words and phrases that share something in common.

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 an annotator component.

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. For 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 annotator correctly annotates occurrences of Tuesday, Wednesday, and the other days of the week in unseen documents at run time.

You can create dictionaries in Watson Knowledge Studio by manually adding individual entries. Watson Knowledge Studio also supports the ability to import several types of dictionary files.

To create a dictionary for this example, from within your project click **Dictionaries** → **Add** (plus icon) → enter dictionary name (Test dictionary) → **Save** (Figure 5-21).

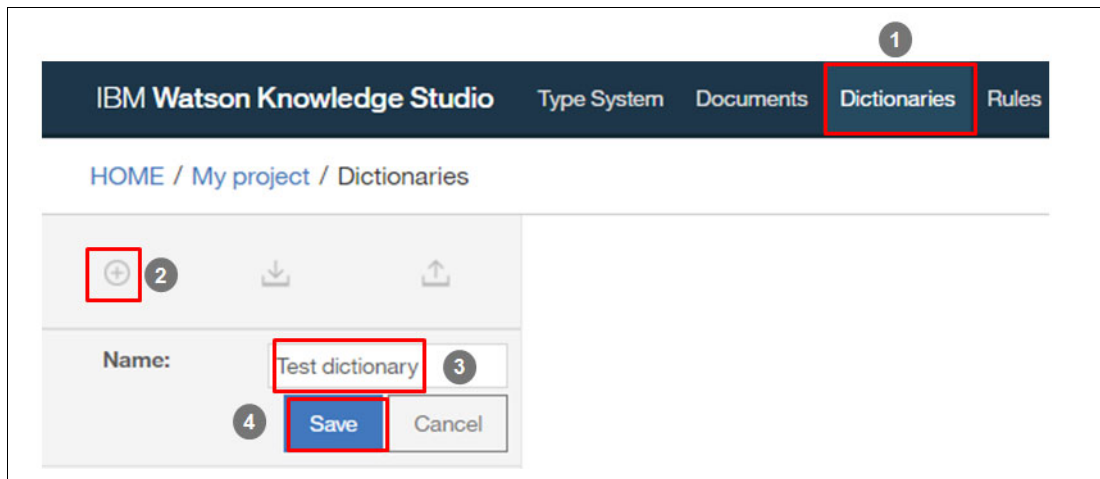


Figure 5-21 Create dictionary

The new dictionary is created and automatically opened for editing.

Next you add entries to the dictionary from a CSV file. In the dictionary pane, click **Import** → select or drag the CSV file → **Import** (Figure 5-22).

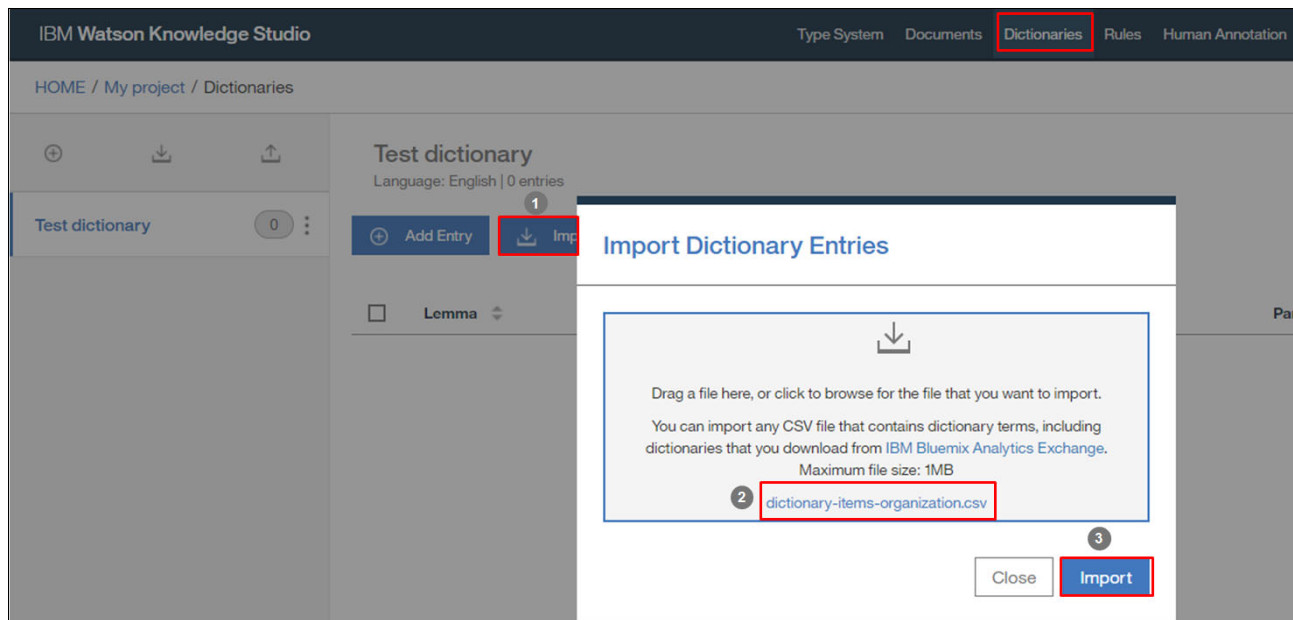


Figure 5-22 Import dictionary entries

The terms in the file are imported into the dictionary. After you create a dictionary, you can use it to speed up human annotation tasks by pre-annotating the documents.

At this point you created a project and added artifacts to it (type system and dictionary). The next steps help you understand the process for building a machine-learning model that you can deploy and use with other Watson services.

## Add documents for annotation

Documents serve a different purpose depending on whether you are creating a machine-learning annotator or a rule-based annotator. This example shows how to create a machine learning annotator. To train a machine-learning annotator component, you must add documents that contain subject matter knowledge, such as journal articles or other texts that are industry-specific, to your project. Before doing this task, you must obtain representative documents, written in natural language, for training the machine-learning model.

To add documents to a project in Watson Knowledge Studio that can be annotated by human annotators, within your project click

**Documents** → **Import Document Set** → select or drag the .csv file with your document collection → **Import** (Figure 5-23).

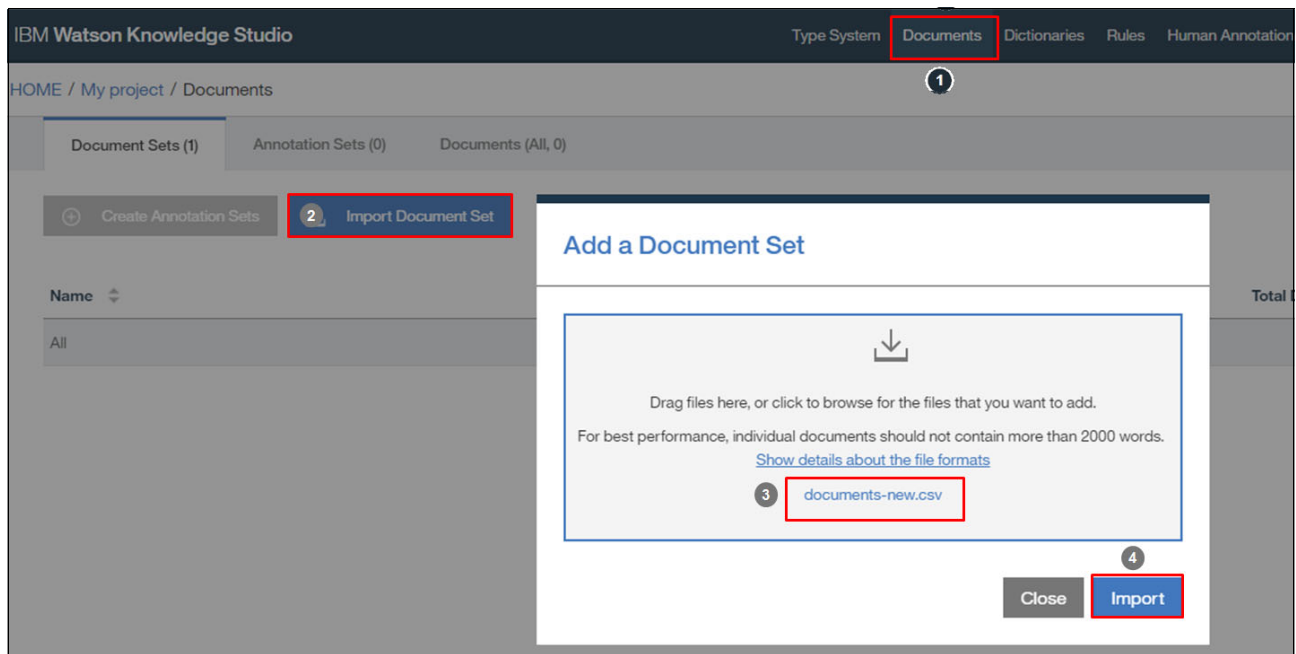


Figure 5-23 Add a document set

The imported file is displayed in a table.

## Create and assign annotation sets

An annotation set is a subset of documents from an imported document set that you assign to a human annotator. The human annotator annotates the documents in the annotation set. To use inter-annotator scores later to compare the annotations that are added by each human annotator, you must assign at least two human annotators to different annotation sets. You must also specify that some percentage of documents overlap between the sets.

To create annotations sets, within your project, click **Documents** → **Create Annotation Sets** → for each new annotation set you are creating, specify the required information → **Generate** (Figure 5-24).

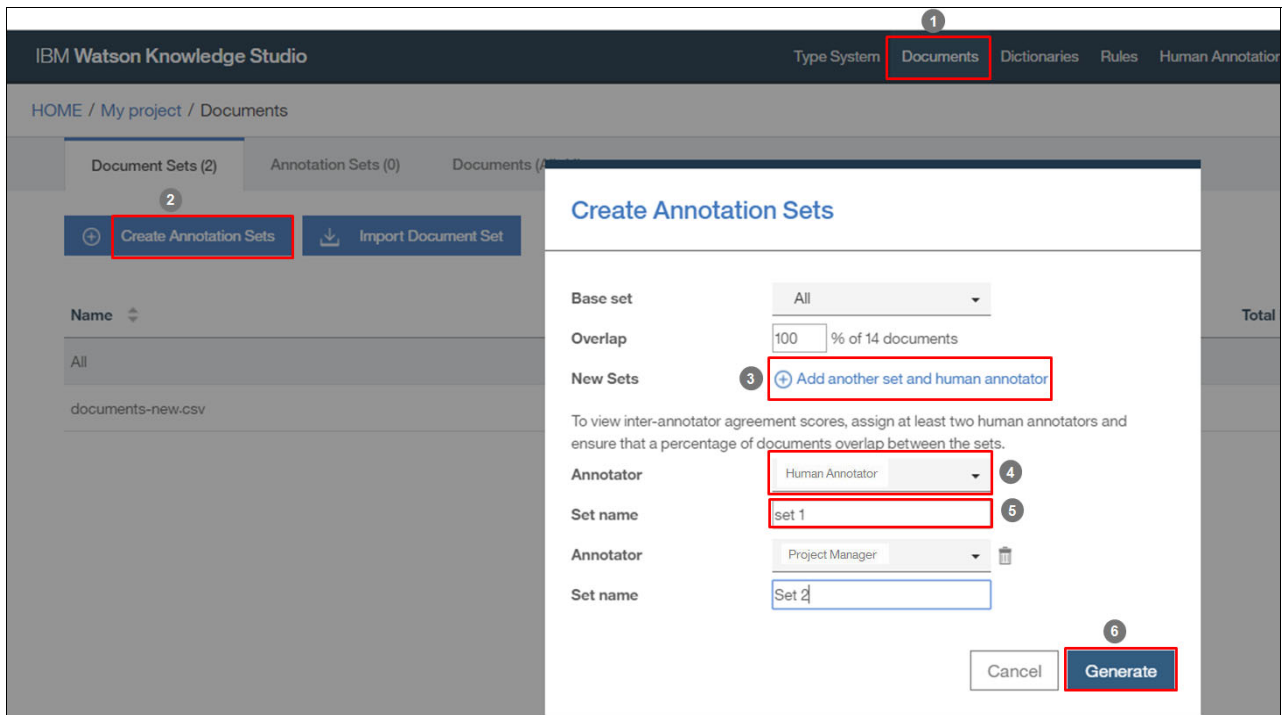


Figure 5-24 Create annotation sets

The new annotation sets are created and now are listed in the Annotation Sets tab of the Documents page (Figure 5-25).

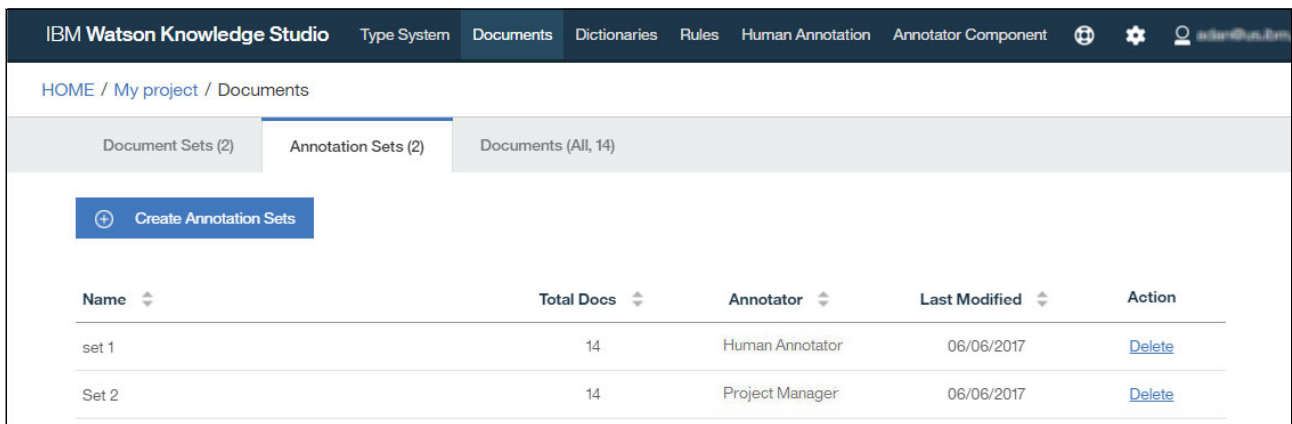


Figure 5-25 Annotation sets

### Optional: Pre-annotate with a dictionary-based annotator

Pre-annotating documents is an optional step. However, it is a valuable step because it makes the job of human annotators easier later. To pre-annotate documents, use these steps:

1. Within your project, click **Annotator Component**.
2. Under the description of the **Dictionary** annotator type, click **Create this type of pre-annotator**.

- Map the ORGANIZATION entity type to the Test dictionary you created in “Add a dictionary” on page 87.  
Click **Edit** for the ORGANIZATION entity type name, and from the list, select **Test dictionary** as the dictionary (Figure 5-26).

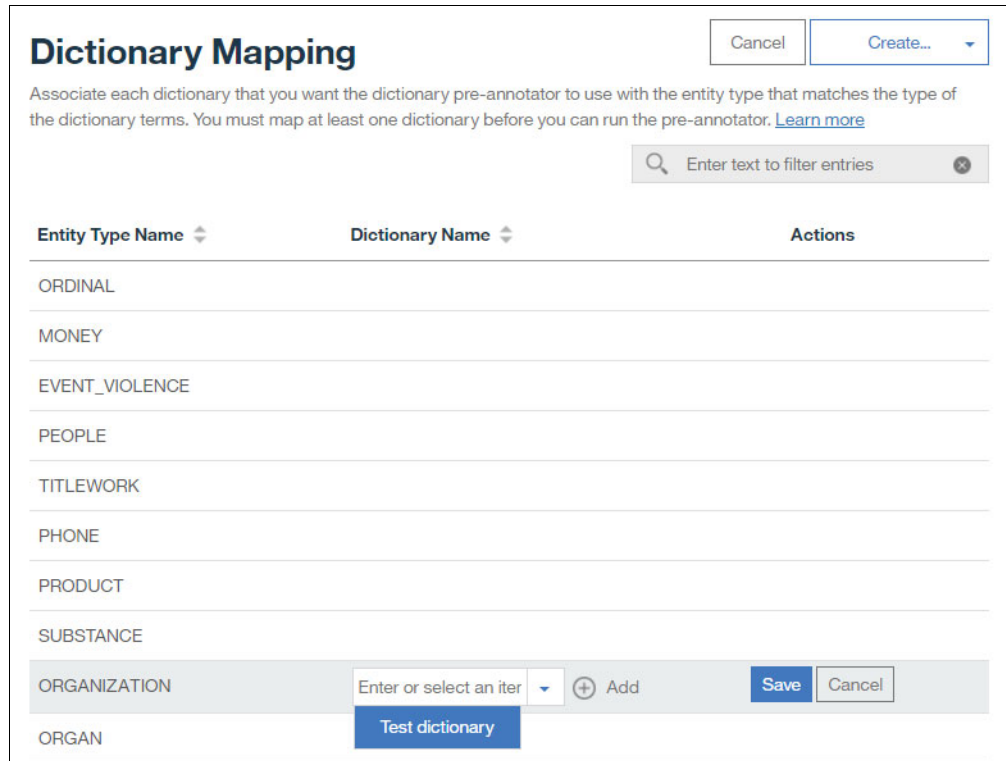


Figure 5-26 Select dictionary for mapping

- Click the plus sign (+) beside the dictionary name to add the mapping, and then click **Save** (Figure 5-26).
- Click **Create** and then, from the menu, select **Create & Run** (Figure 5-27).

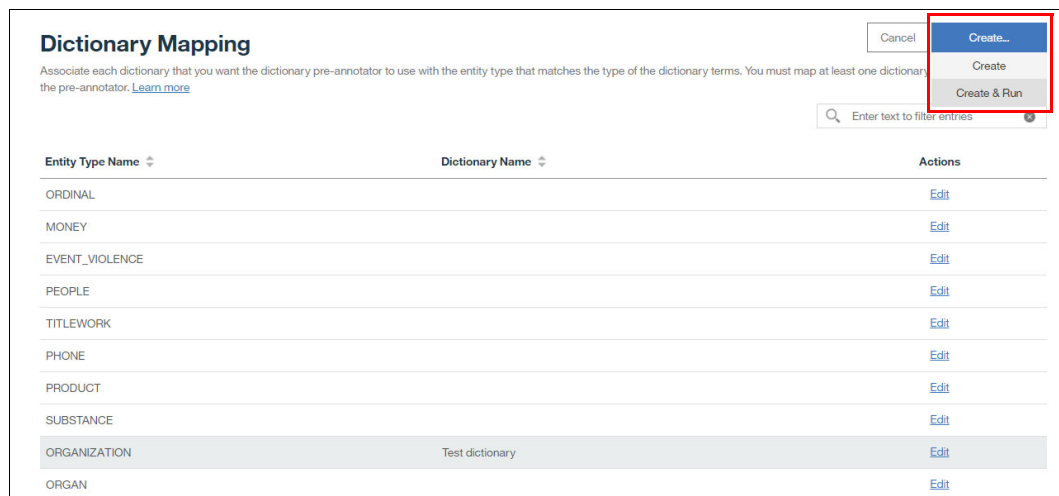


Figure 5-27 Create & Run

6. On the Run Annotator page, click the check boxes to select both of the annotation sets that you created earlier in “Create and assign annotation sets” on page 89 (not including the base set).
7. Click **Run** (Figure 5-28).



Figure 5-28 Run annotator

The documents in the selected sets are pre-annotated using the dictionary annotator you created. The annotator component is added to the Annotator Component page; you can later use the same annotator to pre-annotate additional document sets by clicking **Run** (Figure 5-29).

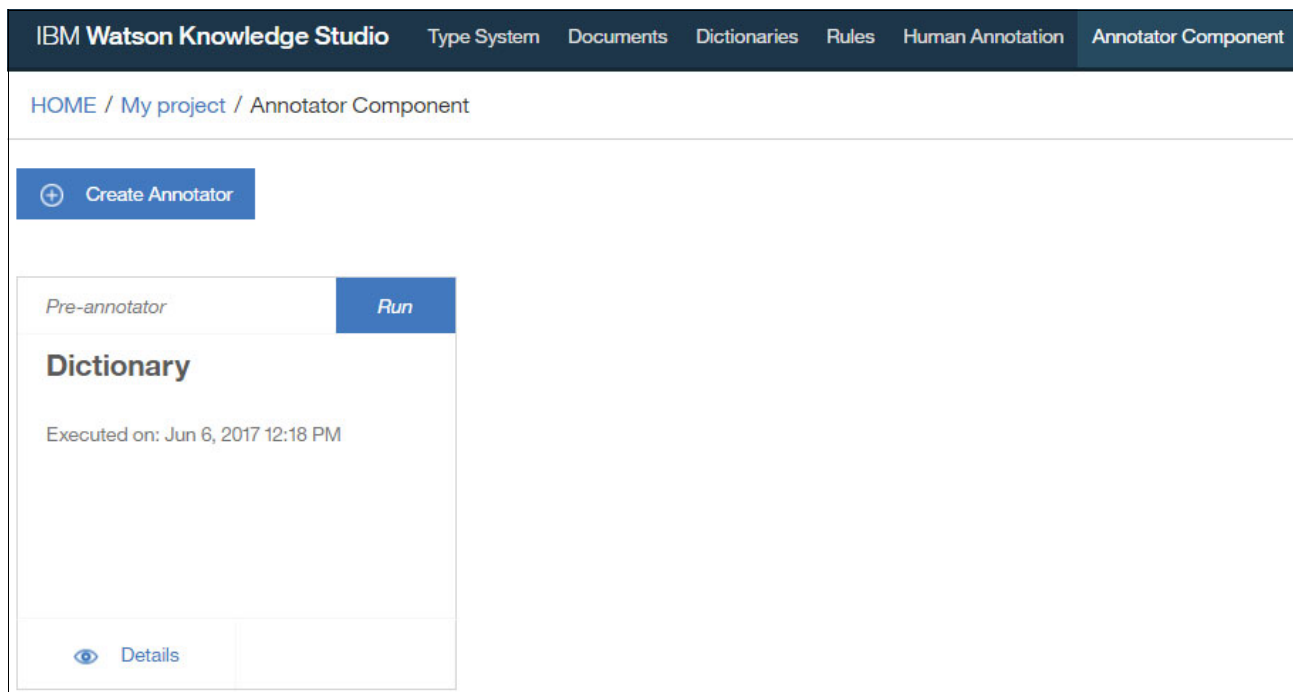


Figure 5-29 Annotator component

### Create an annotation task

Before human annotators can begin adding annotations to documents, the annotation process manager must create an annotation task. The annotation task specifies which documents are to be annotated. Use annotation tasks to track the work of human annotators in Watson Knowledge Studio.

Within your project, click **Human Annotation** → **Add Task** → Specify the details for the task → **Create** (Figure 5-30).

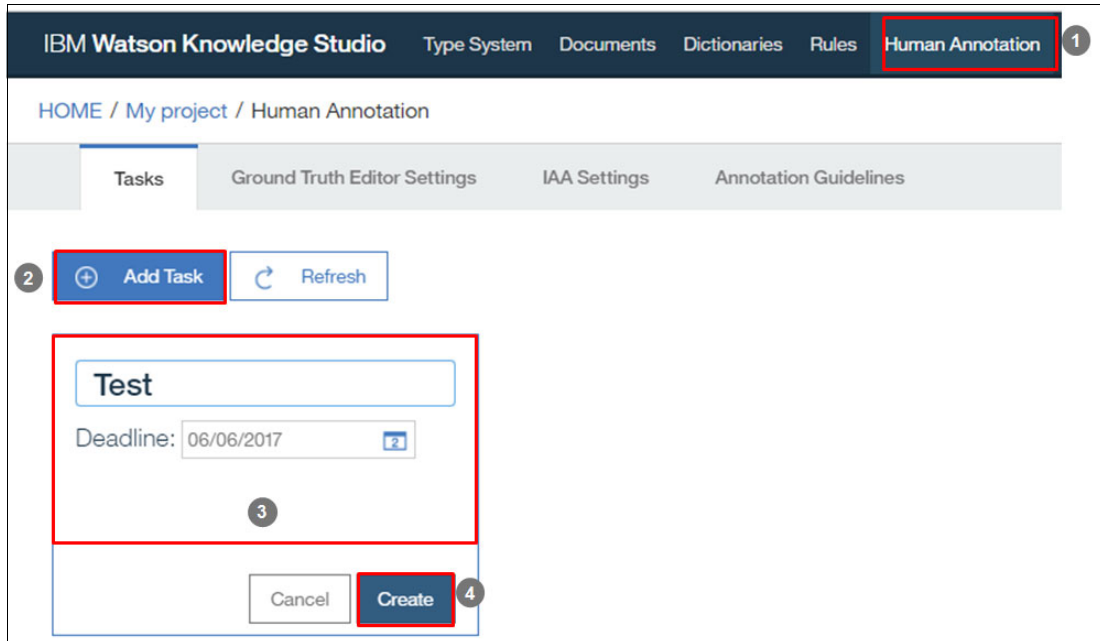


Figure 5-30 Add annotation task

In the Add Annotation Sets to Task window, click the check boxes to select both of the annotation sets that you created in “Create and assign annotation sets” on page 89 and then click **Create Task** (Figure 5-31).

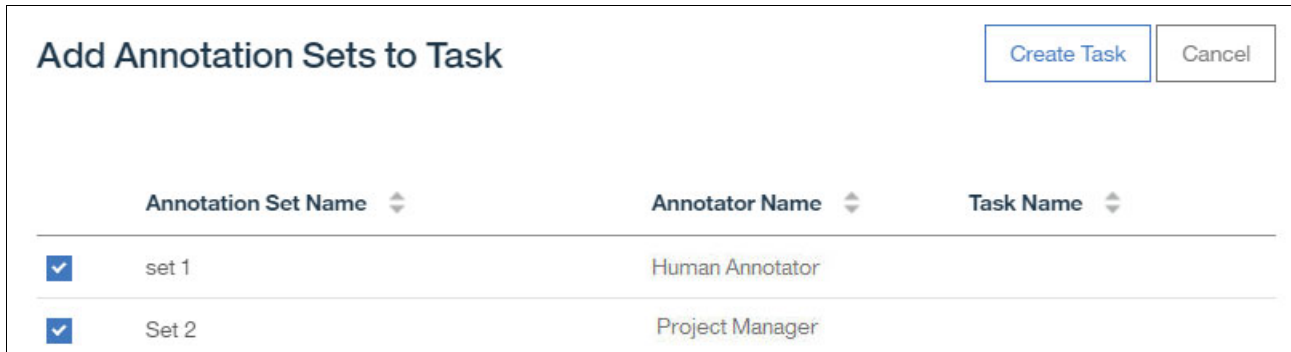


Figure 5-31 Add annotation sets to task

The Test task is now listed on the Tasks tab of the Human Annotation page. Click the **Test** task to open it. You can use this view (Figure 5-32) to see the progress of human annotation work, calculate the inter-annotator agreement scores, and view overlapping documents to adjudicate annotation conflicts.

Annotation Set Name	Annotator Name	Status	
Set 1	Human Annotator	IN PROGRESS	<a href="#">View</a>
Set 2	Project Manager	IN PROGRESS	<a href="#">View</a>

Figure 5-32 Test annotation task status

### Annotate documents

When a human annotator annotates a document, the document is opened in the Ground Truth Editor. The Ground Truth Editor is a visual tool that human annotators use to apply labels to text.

The goal of human annotation is to label mentions, relations, and coreferenced mentions so that the machine-learning annotator can be trained to detect these patterns in unseen text. At a minimum, use the tool to annotate entity mentions. If the application that will use the resulting model does not need to find and extract coreferences and relation mentions, then you do not need to annotate coreferences and relation mentions.

Complete the following steps to use the Ground Truth Editor to annotate documents in Watson Knowledge Studio:

1. Log in to Watson Knowledge Studio as a human annotator who is assigned to the annotation task you created “Create an annotation task” on page 92.
2. Open the project named **My project**.
3. Within the project, click **Human Annotation** in the banner or navigation menu.
4. Open the Test annotation task you created in “Create an annotation task” on page 92.



- Click **Annotate** next to a document set that is assigned to the user ID you are logged in with (Figure 5-33).

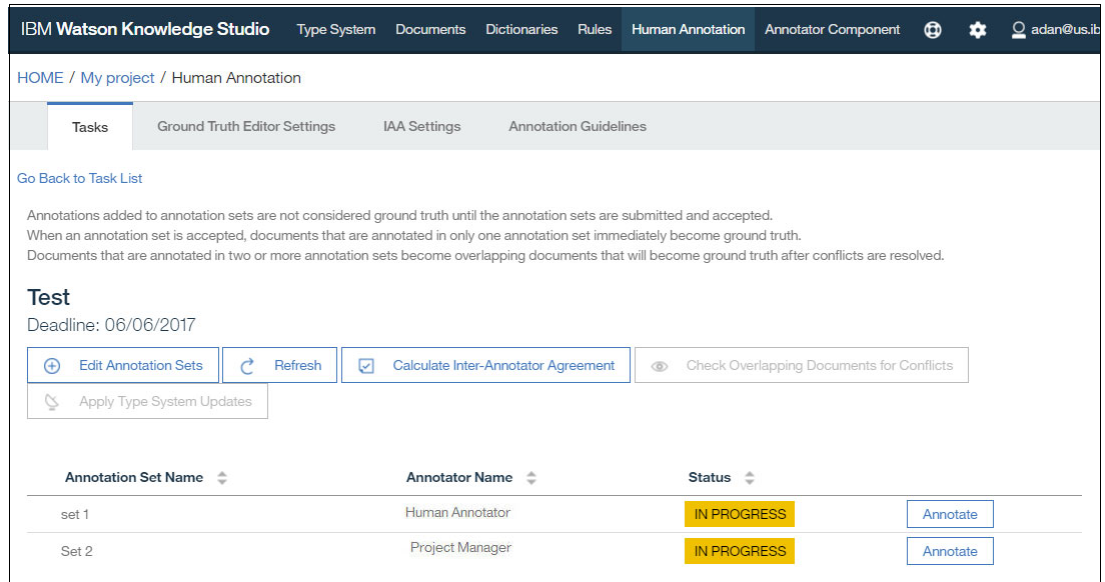


Figure 5-33 Annotate

- The Ground Truth Editor opens; you can preview each document in the document set (Figure 5-34).

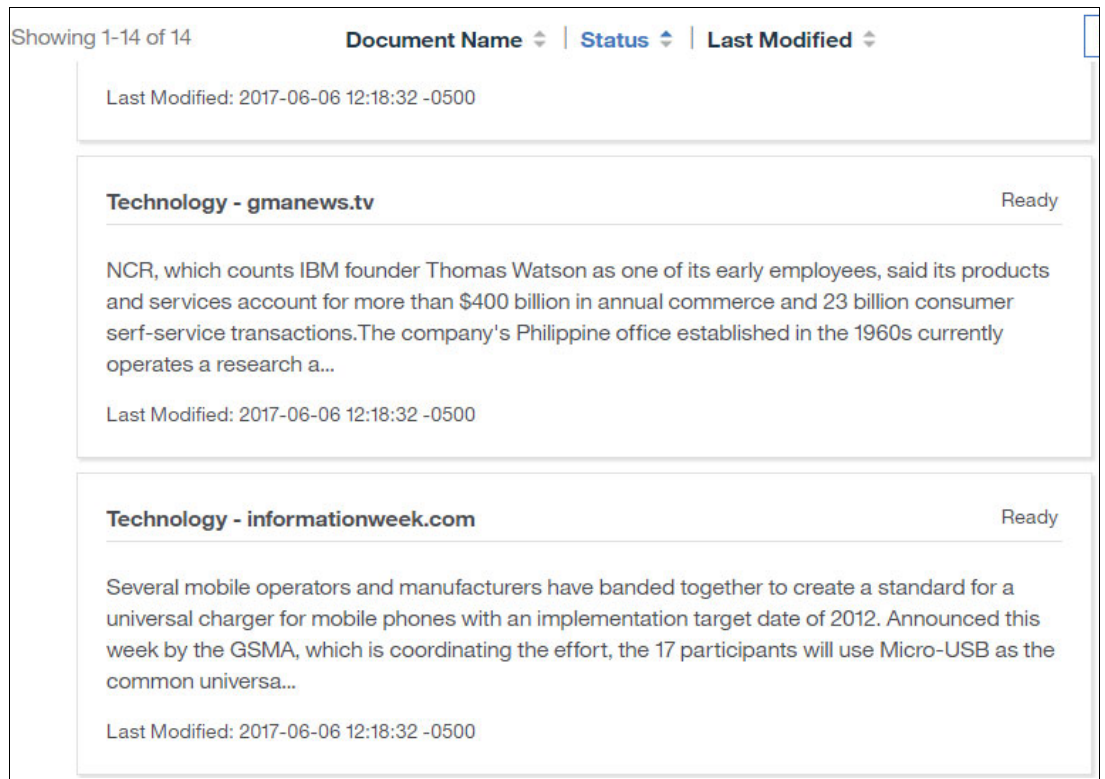


Figure 5-34 Documents in Ground Truth Editor

7. Scroll to the Technology - gmanews.tv document and click to open it for annotation. Note that the term IBM was already annotated with the ORGANIZATION entity type (Figure 5-35); this annotation was added by the dictionary pre-annotator in “Create and assign annotation sets” on page 89. This pre-annotation is correct, so it does not need to be modified.

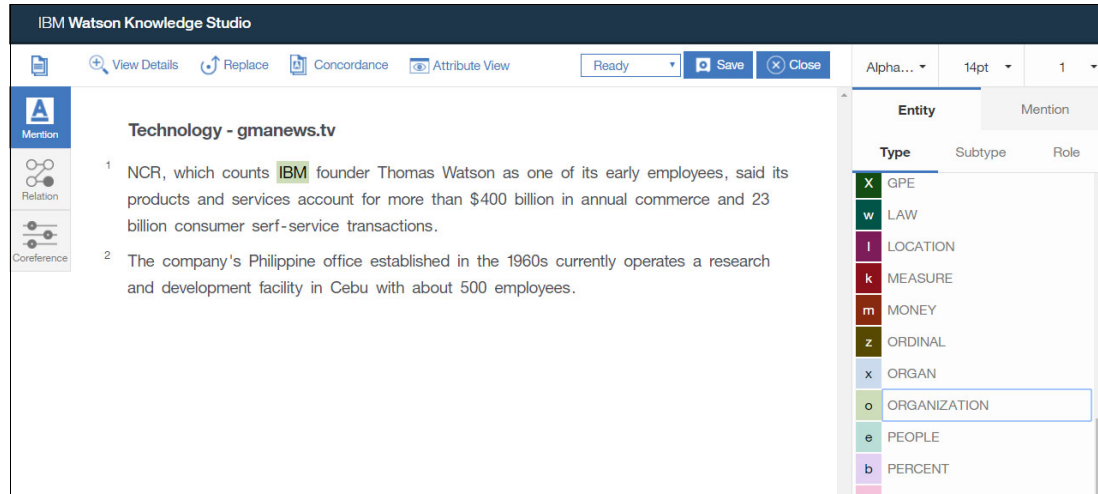


Figure 5-35 Pre-annotated document

8. Annotate a mention (Figure 5-36):
  - a. Click the **Mention** icon to begin annotating mentions.
  - b. In the document body, select the text **Thomas Watson**.
  - c. In the list of entity types, click **PERSON**. The entity type PERSON is applied to the selected mention.

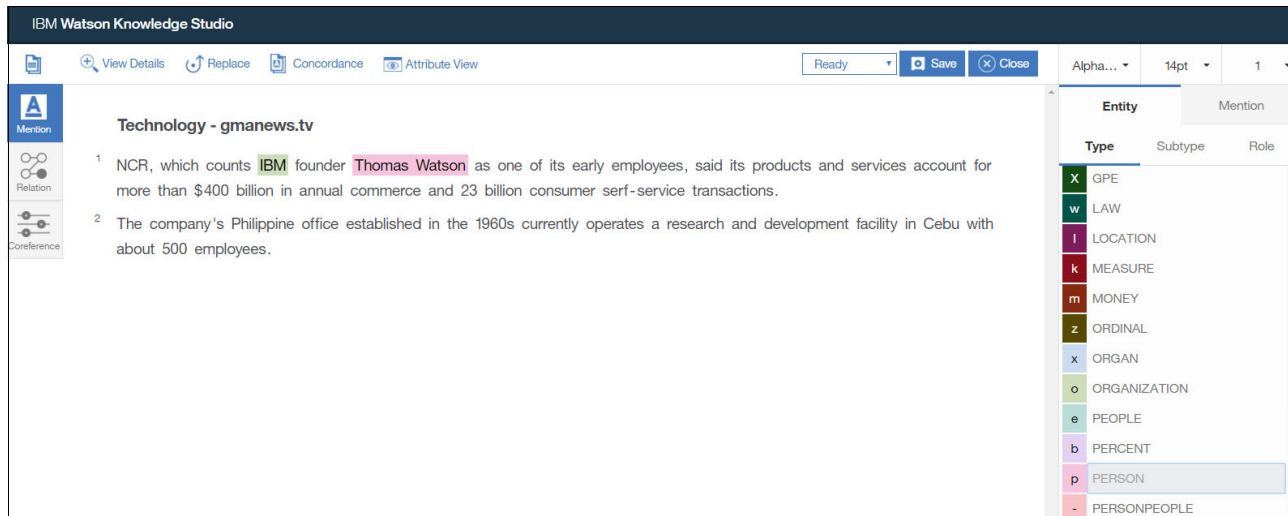


Figure 5-36 Annotate a mention

9. Annotate a relation:

- a. Click the **Relation** icon to begin annotating relations.
- b. Select the **Thomas Watson** and **IBM** mentions (in that order). To select a mention, click the entity-type label above the text.
- c. In the list of relation types, click **founderOf**. The two mentions are connected with a founderOf relationship (Figure 5-37).



Figure 5-37 Annotate a relation

- d. Click **Completed** from the menu, and then click **Save** (Figure 5-38).

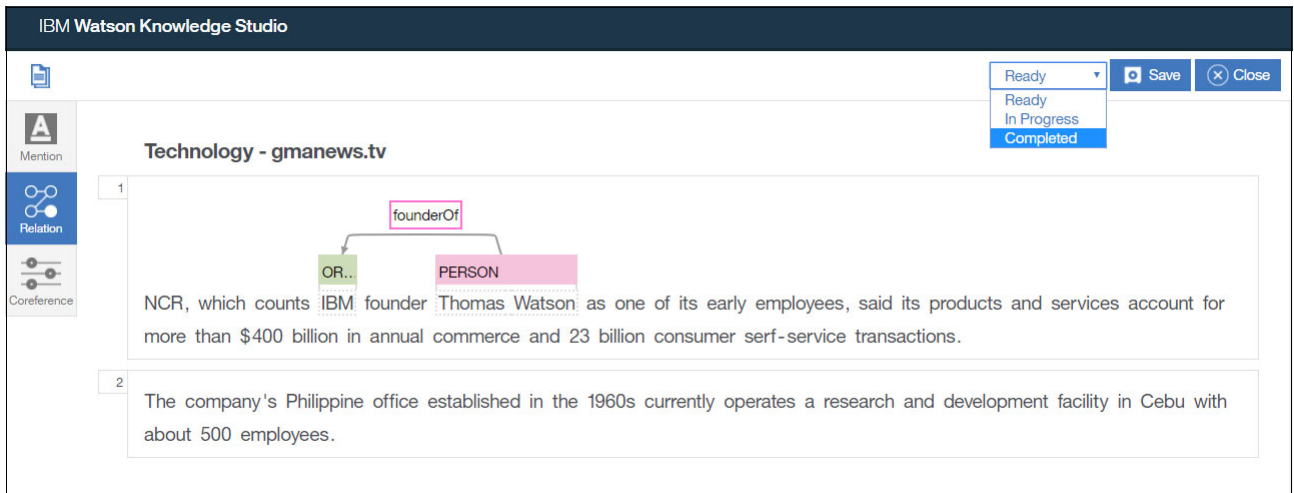


Figure 5-38 Complete document annotation

10. Return to the list of documents and click **Submit All** to submit the documents for approval. Click **OK** to confirm that you want to submit all assigned documents.
11. Close the Ground Truth Editor.

**Note:** In a real-life scenario, this annotation process is repeated by all human annotators for the annotation sets assigned to them.

## Adjudicate conflicts and promote documents to ground truth

To determine whether different human annotators are annotating overlapping documents consistently, review the inter-annotator agreement (IAA) scores.

Watson Knowledge Studio calculates IAA scores by examining all overlapping documents in all document sets in the task, regardless of the status of the document sets. The IAA scores show how different human annotators annotated mentions, relations, and coreference chains. A good idea is to check IAA scores periodically and verify that human annotators are consistent with each other.

To analyze the IAA scores, complete the following steps:

1. Log in to Watson Knowledge Studio as the administrator.
2. Click **Human Annotation** → **Tasks** → **Test** → **Calculate Inter-Annotator Agreement** (Figure 5-39).

The screenshot shows the IBM Watson Knowledge Studio interface. The top navigation bar includes 'Type System', 'Documents', 'Dictionaries', 'Rules', 'Human Annotation', and 'Annotator Component'. The user is logged in as 'adan@us.ibm.com'. The breadcrumb trail is 'HOME / My project / Human Annotation'. The main content area has tabs for 'Tasks', 'Ground Truth Editor Settings', 'IAA Settings', and 'Annotation Guidelines'. Below the tabs, there is a 'Go Back to Task List' link and a paragraph explaining that annotations are not considered ground truth until accepted. A 'Test' section shows a deadline of '06/06/2017' and several buttons: 'Edit Annotation Sets', 'Refresh', 'Calculate Inter-Annotator Agreement' (highlighted), and 'Check Overlapping Documents for Conflicts'. There is also an 'Apply Type System Updates' button and 'Accept'/'Reject' buttons. A table below shows two annotation sets:

Annotation Set Name	Annotator Name	Status	
<input type="checkbox"/> set 1	Human Annotator	SUBMITTED	<a href="#">View</a>
<input type="checkbox"/> Set 2	Project Manager	SUBMITTED	<a href="#">View</a>

Figure 5-39 Calculate IAA

- View IAA scores for mention, relations, and coreference chains by clicking the first drop-down. You can also view agreement by human annotator pairs. In general, aim for a score of 0.8 out of 1, where 1 means perfect agreement. Because documents were not actually annotated in this example, the scores are shown as N/A (Figure 5-40).

HOME / My project / Human Annotation

Tasks | Ground Truth Editor Settings | IAA Settings | Annotation Guidelines

[Back to Test Task](#)

### Review IAA for Test

Mention | Pair

Values in the "All" column are Fleiss kappa scores. The range is below 1, and values can be negative. Values in the other columns are F1 scores. The range can be from 0 to 1. The closer a value is to 1, the higher the agreement rate is (1 means perfect agreement).

Summary	All	Human Annotator & Project Manager
Overall Statistics	1	1

Entity Type	All	Human Annotator & Project Manager
AGE	N/A	N/A
ANIMAL	N/A	N/A
AWARD	N/A	N/A

Figure 5-40 Review IAA scores

- After you review the scores, decide whether to accept or reject annotation sets that are in a Submitted status. To accept the annotation sets, select the check box and click **Accept** (Figure 5-41).

Project Manager ject / Human Annotation

Tasks | Ground Truth Editor Settings | IAA Settings | Annotation Guidelines

[Go Back to Task List](#)

Annotations added to annotation sets are not considered ground truth until the annotation sets are submitted and accepted. When an annotation set is accepted, documents that are annotated in only one annotation set immediately become ground truth. Documents that are annotated in two or more annotation sets become overlapping documents that will become ground truth after conflicts are resolved.

### Test

Deadline: 06/06/2017

Annotation Set Name	Annotator Name	Status
<input checked="" type="checkbox"/> set 1	Human Annotator	SUBMITTED
<input checked="" type="checkbox"/> Set 2	Project Manager	SUBMITTED

Figure 5-41 Accept annotation sets

- Overlapping documents are included in two or more annotation sets. Click **Check for Conflicts** to open a document and check whether different human annotators applied different annotations to it. Resolve any inconsistencies that you find. If you know that no conflicts exist, click **Accept** to add the document to the ground truth (Figure 5-42).

**Overlapping Documents in Task Test**

These documents are included in two or more annotation sets. Click Check for Conflicts to open a document and check whether different human annotators applied different annotations to it. Resolve any inconsistencies that you find. If you know that no conflicts exist, click Accept to add the document to the ground truth.

Document Name	Annotation Set Name	Actions
Technology - ip-telephony.tmcnet.com	set 1, Set 2	<a href="#">Check for Conflicts</a> <a href="#">Accept</a>
Technology - brighthand.com	set 1, Set 2	<a href="#">Check for Conflicts</a> <a href="#">Accept</a>
Technology - computerworld.com	set 1, Set 2	<a href="#">Check for Conflicts</a> <a href="#">Accept</a>
Technology - gamasutra.com	set 1, Set 2	<a href="#">Check for Conflicts</a> <a href="#">Accept</a>

Figure 5-42 Check for conflicts or accept add documents to ground truth

After all documents in the annotation set are accepted, the status is COMPLETED (Figure 5-43).

The screenshot shows the IBM Watson Knowledge Studio interface. The top navigation bar includes 'Type System', 'Documents', 'Dictionaries', 'Rules', 'Human Annotation', and 'Annotator Component'. The main content area is titled 'HOME / My project / Human Annotation' and has tabs for 'Tasks', 'Ground Truth Editor Settings', 'IAA Settings', and 'Annotation Guidelines'. Under the 'Tasks' tab, there is a 'Go Back to Task List' link and a paragraph explaining that annotations are not ground truth until accepted. Below this is a 'Test' section with a deadline of '06/06/2017'. A toolbar contains buttons for 'Edit Annotation Sets', 'Refresh', 'Calculate Inter-Annotator Agreement', and 'Check Overlapping Documents for Conflicts'. At the bottom, a table shows two annotation sets, both with a 'COMPLETED' status and a 'View' button.

Annotation Set Name	Annotator Name	Status	View
set 1	Human Annotator	COMPLETED	<a href="#">View</a>
Set 2	Project Manager	COMPLETED	<a href="#">View</a>

Figure 5-43 Status is COMPLETED

After you resolve the annotation conflicts and promote the documents to ground truth, you can use them to train the machine-learning annotator.

### Create a machine-learning annotator

When you create a machine-learning annotator, you select the document sets that you want to use to train it. You also specify the percentage of documents that are to be used as training data, test data, and blind data. Only documents that became ground truth through approval or adjudication can be used to train the machine-learning annotator.

To create the machine-learning annotator component, perform the following steps:

- Log in to Watson Knowledge Studio as the administrator.
- On the Annotator Component page, click **Create Annotator**.

3. Click **Create this type of annotator** in the machine-learning annotator section (Figure 5-44).

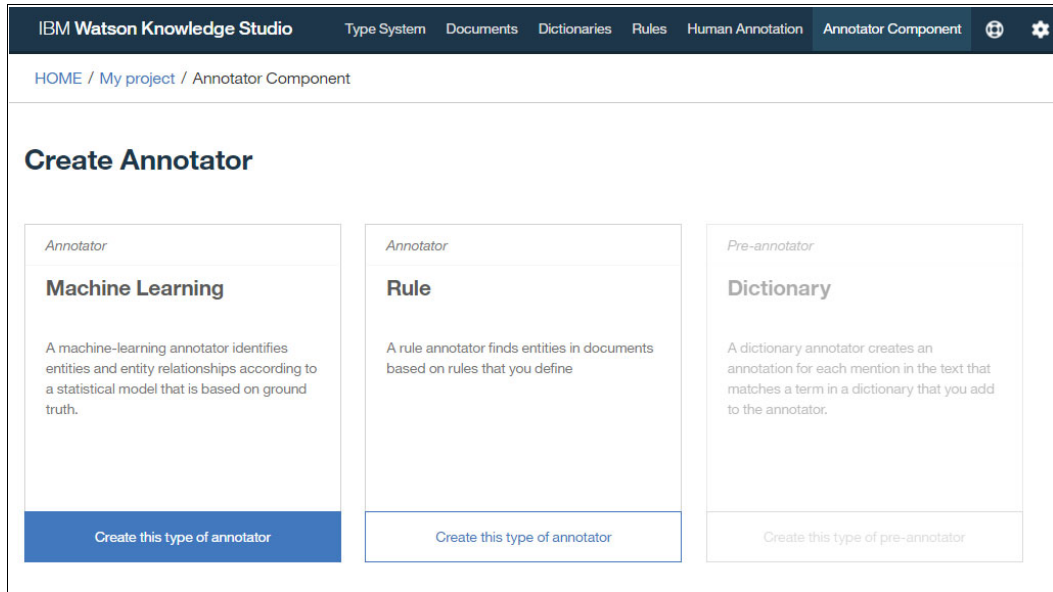


Figure 5-44 Create machine learning annotator (1 of 3)

4. Select the document sets that you want to use for creating a machine-learning annotator. Click the check mark next to each document set name. Use the default values for creating your testing, training, and blind data. Click **Next** (Figure 5-45).

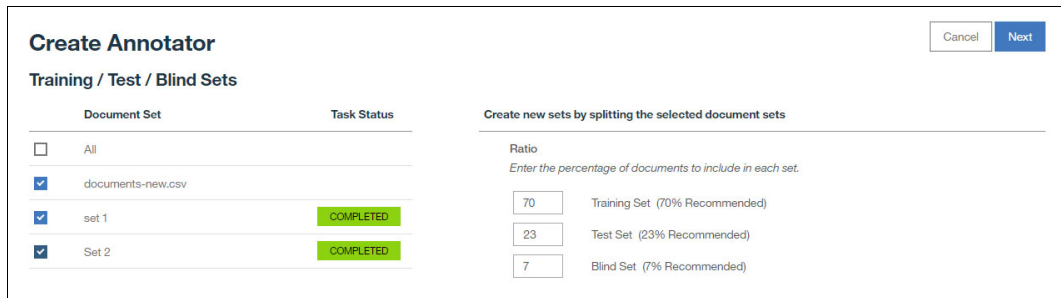


Figure 5-45 Create machine learning annotator (2 of 3)

5. Click **Train & Evaluate** (Figure 5-46).

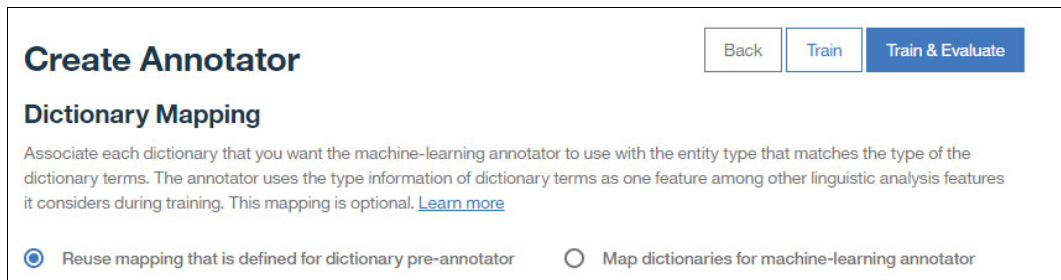


Figure 5-46 Create machine learning annotator (3 of 3)

The training process starts. Training can take several minutes, or even hours, depending on the number of human annotations and the total number of words across documents.



- After the machine-learning annotator is trained, you can export it by clicking **Export** or you can view detailed information on its performance by clicking **Details** (Figure 5-47).

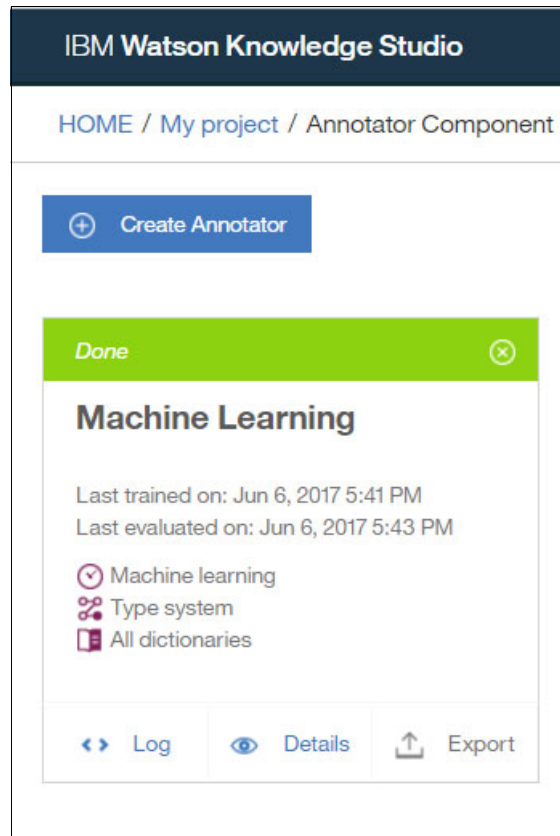


Figure 5-47 Trained machine learning annotator

- By exploring the performance metrics, you can identify ways to improve the accuracy of the machine-learning annotator. Click **Details** → **Model Settings** → **Training / Test / Blind Sets** (Figure 5-48).

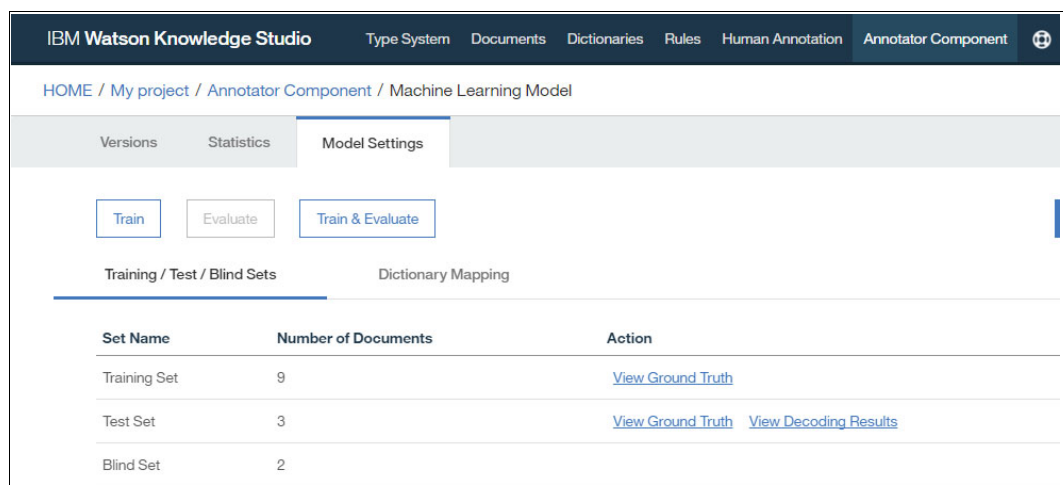


Figure 5-48 View Ground Truth and Decoding Results



8. Click one of the following actions:
  - Click **View Ground Truth** to see the documents that human annotators worked on.
  - Click **View Decoding Results** to see the annotations that the trained machine-learning annotator created on that same set of documents.
9. On the Statistics page (Figure 5-49), view details about the precision, recall, and F1 scores for the machine-learning annotator. You can view these scores for mentions, relations, and coreference chains by using the radio buttons. You can analyze performance by viewing a summary of statistics for entity types, relation types, and coreference chains.

Entity Types	F1	Precision	Recall	% of Total Annotations	% of Corpus Density (by number of words)	% of Documents that Contain This Type
AWARD	N/A	N/A	N/A	0% (0/11)	0% (0/807)	0% (0/3)
CARDINAL	N/A	N/A	N/A	0% (0/11)	0% (0/807)	0% (0/3)
<b>Overall Statistics</b>	<b>0.9</b>	<b>1</b>	<b>0.82</b>	<b>100% (11/11)</b>	<b>1% (11/807)</b>	<b>33% (1/3)</b>

Figure 5-49 Machine-learning annotator statistics page

10. On the Versions page (Figure 5-50), you can take a snapshot of the annotator and the resources that were used to create it (except for dictionaries and annotation tasks). For example, you might want to take a snapshot before you change the annotator. If the statistics are poorer the next time you run it, you can promote the older version and delete the version that returned poorer results.

Version	Base	Creation Date	Entity Types	Relation Types	Description	Action	Status
1.0		Current Version	0 (0 / 0)	N/A		Take Snapshot	N/A

Figure 5-50 Machine-learning annotator Versions page

The machine-learning annotator is created and trained. At this point, you have a custom machine-learning model that you can use with other Watson services.

### 5.3.3 Deploying a machine-learning annotator to Watson Natural Language Understanding

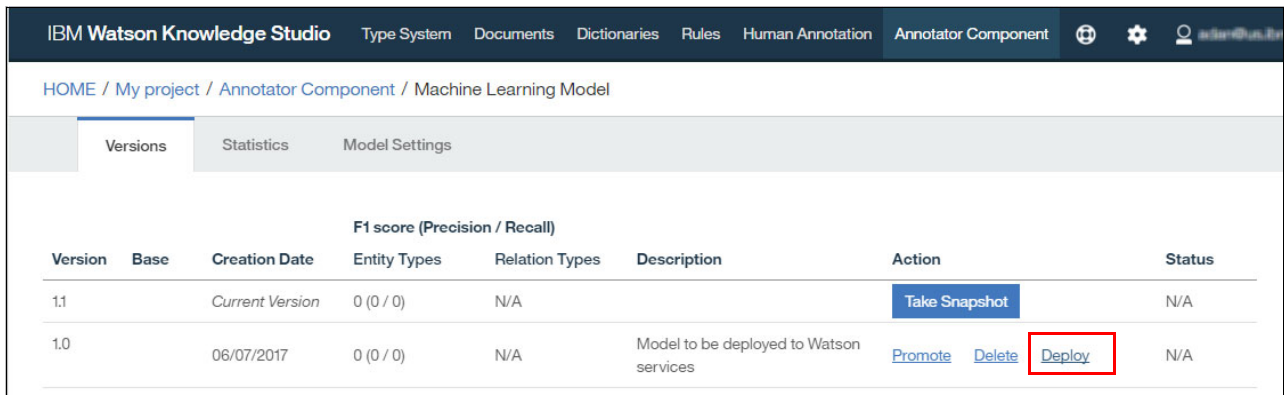
When you are satisfied with the performance of the annotator component, you can deploy a version of it to Watson Natural Language Understanding. This feature enables your applications to use the deployed machine-learning annotator to analyze semantic features of text input, including entities and relations. The custom model overrides the standard entity detection model.

**Note:** A trial Bluemix account enables use of *one* custom model published through Watson Knowledge Studio. You must know the Bluemix space and instance names that are associated with your account.

To deploy a machine-learning annotator to the Natural Language Understanding service, complete the following steps:

1. Log in as a Watson Knowledge Studio administrator or project manager.
2. Within your project, click **Annotator Component**.
3. In the Machine-Learning Annotator, click **Details**.
4. On the Versions tab (Figure 5-51), find the version of the model that you want to deploy.

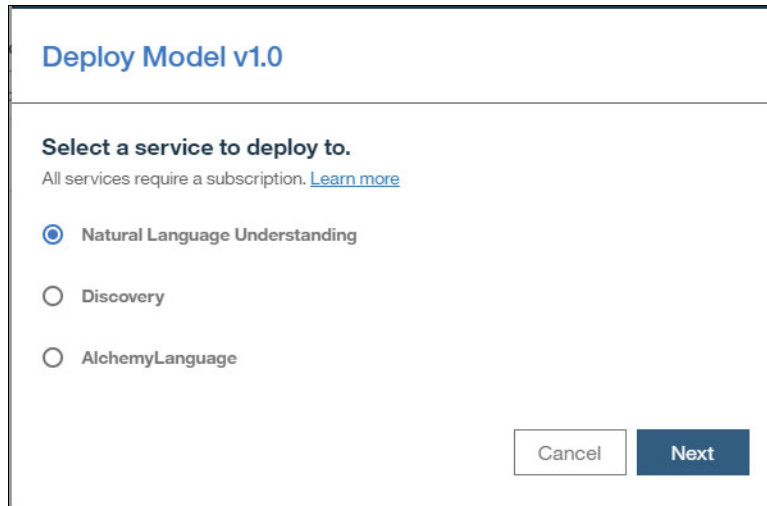
If only one working version of the model exists, create a snapshot of the current model. This versions the model, which enables you to deploy one version, while you continue to improve the current version. The option to deploy does not appear until you create at least one version. Click **Deploy**.



IBM Watson Knowledge Studio							
Type System	Documents	Dictionaries	Rules	Human Annotation	Annotator Component		
HOME / My project / Annotator Component / Machine Learning Model							
Versions		Statistics	Model Settings				
Version	Base	Creation Date	F1 score (Precision / Recall)		Description	Action	Status
			Entity Types	Relation Types			
1.1		Current Version	0 (0 / 0)	N/A		Take Snapshot	N/A
1.0		06/07/2017	0 (0 / 0)	N/A	Model to be deployed to Watson services	Promote Delete <b>Deploy</b>	N/A

Figure 5-51 Deploy machine learning model to Watson service

5. Choose to deploy it to Natural Language Understanding and then click **Next**. (Figure 5-52).



**Deploy Model v1.0**

**Select a service to deploy to.**  
All services require a subscription. [Learn more](#)

Natural Language Understanding

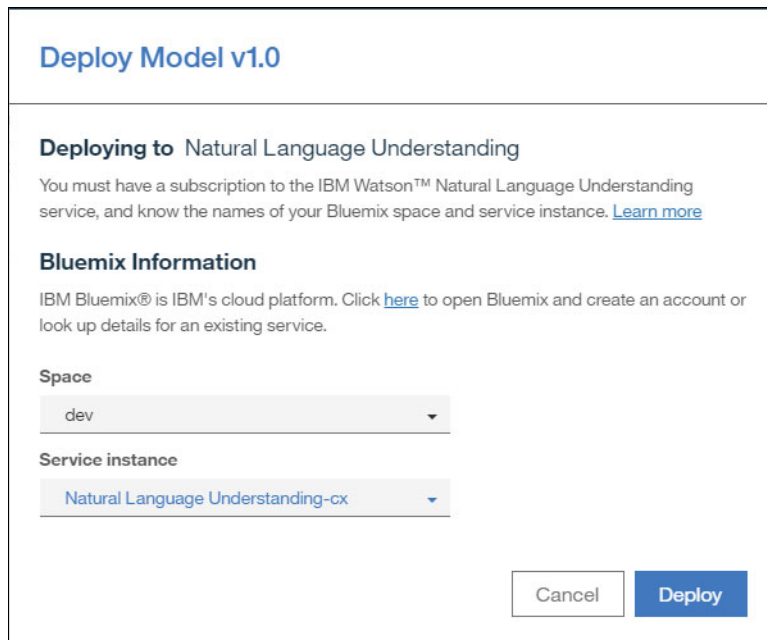
Discovery

AlchemyLanguage

Cancel Next

Figure 5-52 Deploy custom model to Natural Language Understanding service

6. Provide the IBM Bluemix space and instance names, and then click **Deploy** (Figure 5-53).



**Deploy Model v1.0**

**Deploying to Natural Language Understanding**  
You must have a subscription to the IBM Watson™ Natural Language Understanding service, and know the names of your Bluemix space and service instance. [Learn more](#)

**Bluemix Information**  
IBM Bluemix® is IBM's cloud platform. Click [here](#) to open Bluemix and create an account or look up details for an existing service.

**Space**  
dev

**Service instance**  
Natural Language Understanding-cx

Cancel Deploy

Figure 5-53 Specify space and NLU service instance name

7. The model deployment starts (Figure 5-54). Note the Model ID and then click **OK**.

**Model ID:** Make a note of the model ID. You will provide this ID to the Natural Language Understanding service in order to enable the service to use your custom model.

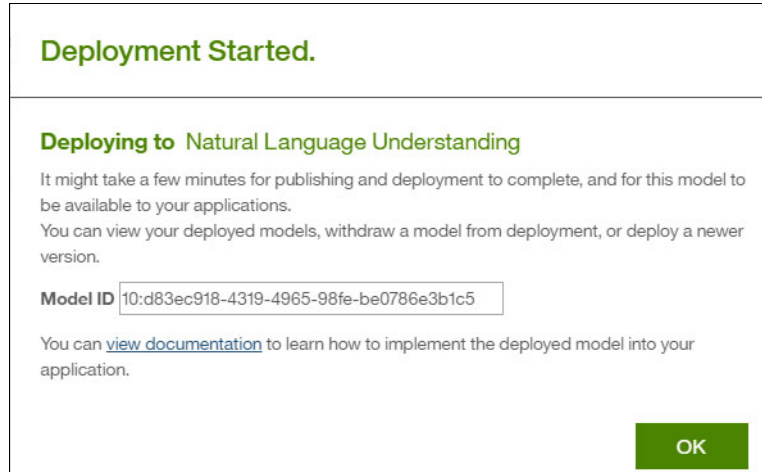


Figure 5-54 Deployment started

8. On the Versions tab, and under the Status column, click the version that you deployed. The Deployment Status window opens (Figure 5-55). If the model is still being deployed, the status indicates publishing. After deployment completes, the status changes to available if the deployment was successful, or error if problems occurred.

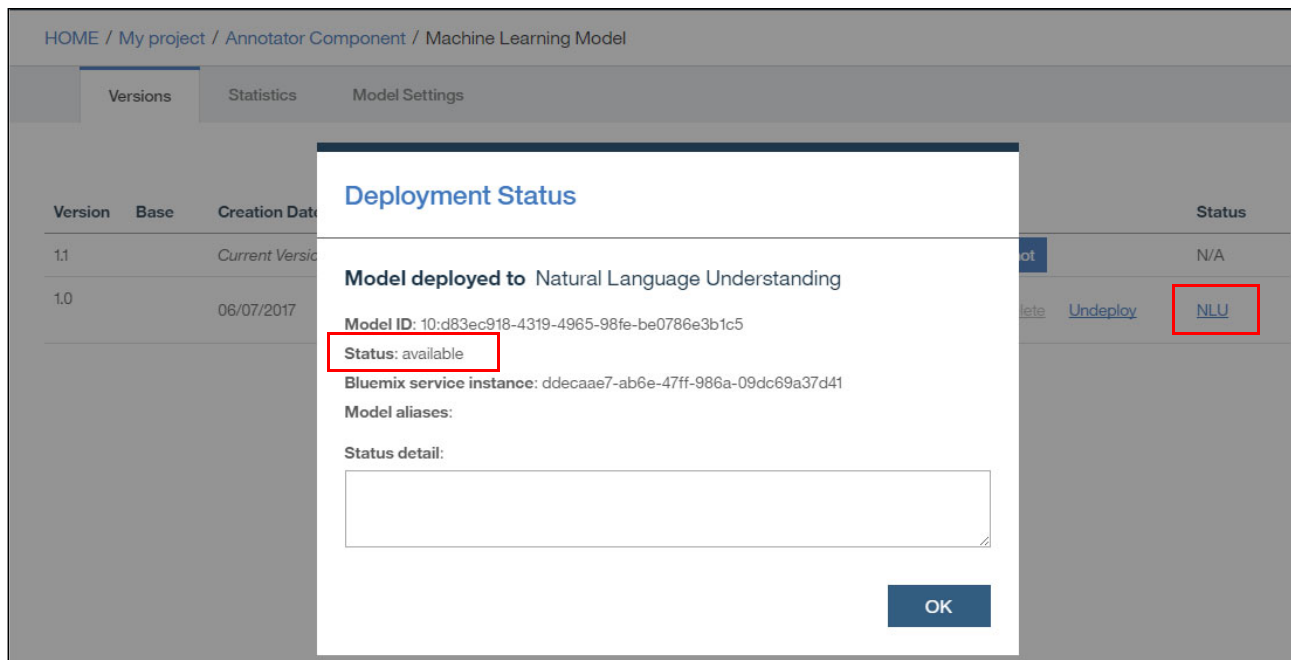


Figure 5-55 Deployment status

9. Click **OK**.

### 5.3.4 Deploying a machine-learning annotator to Watson Discovery

When you are satisfied with the performance of the annotator component, you can deploy a version of it to IBM Watson Discovery. This feature enables your applications to use the deployed machine-learning model to enrich the insights that you get from your data to include the recognition of concepts and relations that are relevant to your domain.

You must have administrative access to a Watson Discovery service instance, and know the IBM Bluemix space and instance names that are associated with it.

To deploy a machine-learning annotator to Watson Discovery, complete these steps:

1. Follow steps 1 on page 104 through 4 on page 104.
2. Choose to deploy the service to **Discovery** and then click **Next** (Figure 5-56).

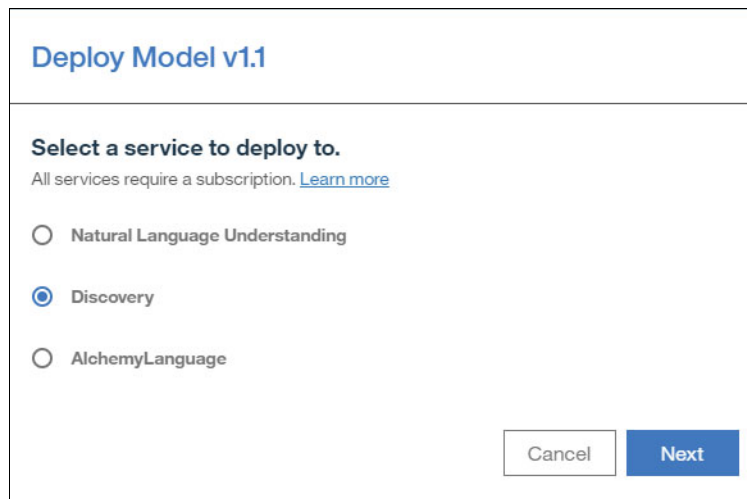


Figure 5-56 Deploy custom model to Watson Discovery service

3. Provide the IBM Bluemix space and instance names, and then click **Deploy** (Figure 5-57).

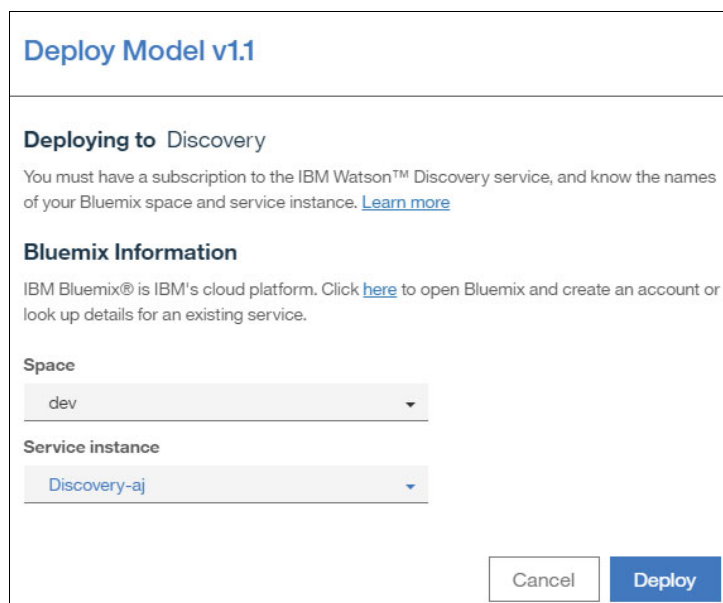


Figure 5-57 Specify space and Discovery service instance name

- The model deployment starts (Figure 5-58). Note the Model ID and then click **OK**.

**Model ID:** Make a note of the model ID. You will provide this ID to the Discovery service in order to enable the service to use your custom model.

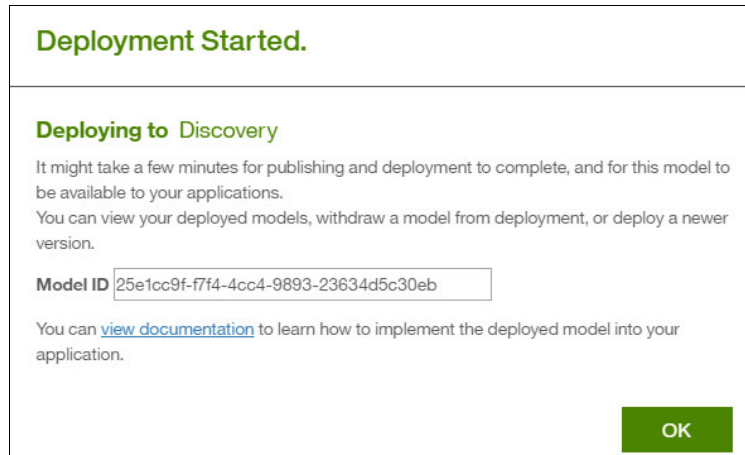


Figure 5-58 Deployment started

- On the Versions tab, find the version that you deployed and click under **Status**. If the model is still being deployed, the status indicates publishing. After deployment completes, the status changes to available if the deployment was successful, or error if problems occur (Figure 5-59).

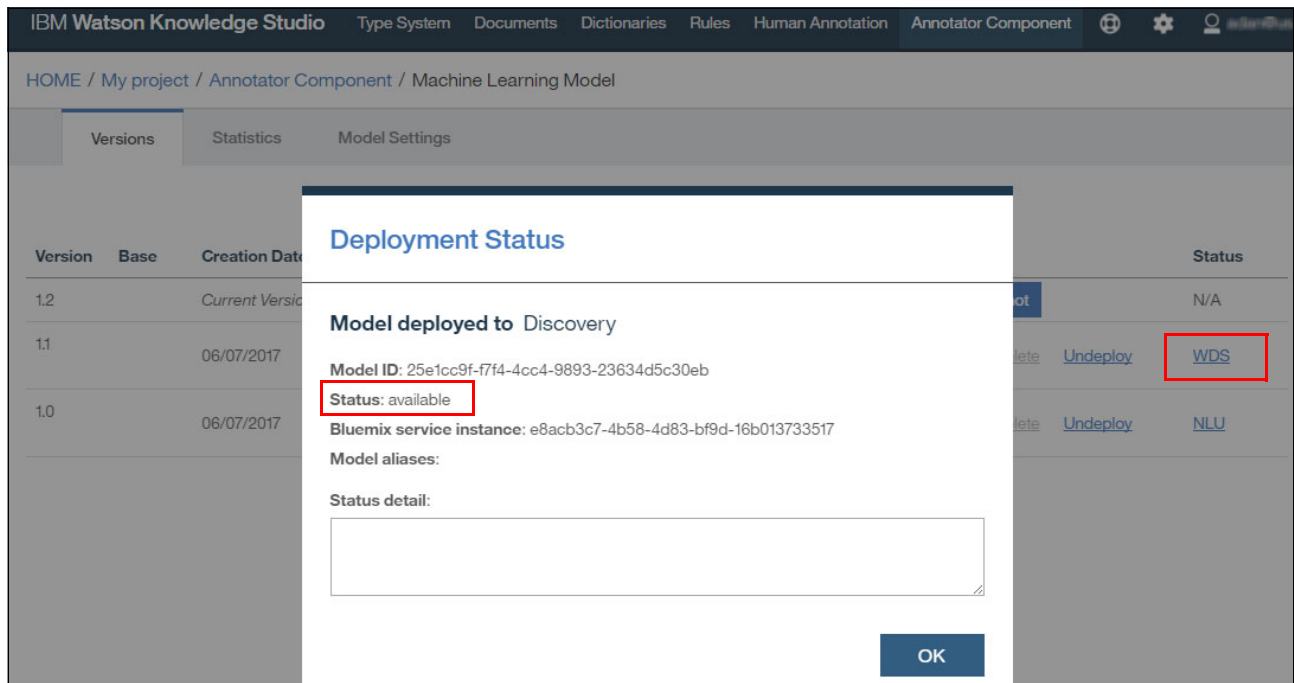


Figure 5-59 Deployment status

To use the deployed model, you must provide the model ID when it is requested during the Discovery service enrichment configuration process. For more details, see the [Discovery service documentation](#) (Integrating with IBM Watson Knowledge Studio).

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

## IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

The volumes in the *Building Cognitive Applications with IBM Watson APIs* series:

- ▶ *Volume 1 Getting Started*, SG24-8387
- ▶ *Volume 2 Conversation*, SG24-8394
- ▶ *Volume 3 Visual Recognition*, SG24-8393
- ▶ *Volume 4 Natural Language Classifier*, SG24-8391
- ▶ *Volume 5 Language Translator*, SG24-8392
- ▶ *Volume 6 Speech to Text and Text to Speech*, SG24-8388
- ▶ *Volume 7 Natural Language Understanding*, SG24-8398

You can search for, view, download or order these documents and other IBM Redbooks, Redpapers™, Web Docs, draft and additional materials, at the following website:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Online resources

These websites are also relevant as further information sources:

- ▶ Watson products and APIs:  
<https://www.ibm.com/watson/products.html>
- ▶ Transform Learning Experiences with Watson:  
<https://www.ibm.com/watson/education/#>
- ▶ IBM Watson Commerce Point of View:  
<https://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=UVW12550USEN>
- ▶ Watson Financial Services:  
<https://www.ibm.com/watson/financial-services/>
- ▶ Introducing IBM Watson Health video:  
[https://youtu.be/yV\\_6sd32oW0?t=34](https://youtu.be/yV_6sd32oW0?t=34)
- ▶ Collaboration to Advance Genomic Medicine video:  
[https://youtu.be/xQvdR\\_iUDhI](https://youtu.be/xQvdR_iUDhI)
- ▶ Search for ALS treatments video:  
<https://youtu.be/F-qBLH6EfR8>

- ▶ IBM Watson Health:  
<https://www.ibm.com/watson/health/>
- ▶ Watson for Genomics helps doctors give patients new hope:  
<https://www.ibm.com/watson/health/oncology-and-genomics/genomics/>
- ▶ IBM Watson for Drug Discovery:  
<https://www.ibm.com/watson/health/life-sciences/drug-discovery/>
- ▶ Engage patients and consumers:  
<https://www.ibm.com/watson/health/value-based-care/patient-engagement/>
- ▶ Watson for Oncology:  
<https://www.ibm.com/watson/health/oncology-and-genomics/oncology/>
- ▶ Watson Care Manager:  
<https://www.ibm.com/watson/health/value-based-care/watson-care-manager/>
- ▶ Employees are making better decisions, faster:  
<https://www.ibm.com/watson/stories/insurance-with-watson.html>
- ▶ How it Works: Internet of Things video:  
<https://www.youtube.com/watch?v=QSIPNh0iMoE&feature=youtu.be>
- ▶ Industry 4.0 meets Watson IoT video:  
<https://www.youtube.com/watch?v=vjaISgnKN3Q&feature=youtu.be>
- ▶ IBM Research Takes Watson to Hollywood with the First “Cognitive Movie Trailer” video:  
<https://www.ibm.com/blogs/think/2016/08/cognitive-movie-trailer/>
- ▶ Morgan movie trailer video:  
<https://www.youtube.com/watch?v=gJEzuYynaiw&feature=youtu.be>
- ▶ Watson for Cyber Security web page:  
[https://www.ibm.com/security/cognitive/?drive\\_source=cognitivesecurity](https://www.ibm.com/security/cognitive/?drive_source=cognitivesecurity)
- ▶ Watson for Cyber Security in Action video:  
<https://www.youtube.com/watch?v=MYZ0IdK4o1M&feature=youtu.be>
- ▶ Teaching Watson the Language of Security video:  
<https://www.youtube.com/watch?v=kao05ArIiok&feature=youtu.be>
- ▶ IBM Watson Visual Recognition video:  
[https://www.youtube.com/watch?v=n3\\_oGnXkMAE&feature=youtu.be](https://www.youtube.com/watch?v=n3_oGnXkMAE&feature=youtu.be)
- ▶ Now you can see the potential hidden in millions of online images:  
<https://www.ibm.com/blogs/watson/2016/08/now-can-see-potential-hidden-millions-online-images/>
- ▶ How can every Woodside employee instantly access 30 years of experience?:  
<https://www.ibm.com/watson/stories/woodside.html>
- ▶ Medtronic and IBM Watson Health partner to develop new ways to tackle diabetes:  
<http://www.medtronic.com/us-en/about/news/ibm-diabetes.html>
- ▶ A Chef Takes a Fresh Approach to Diabetes:  
<https://medium.com/cognitivebusiness/a-chef-takes-a-fresh-approach-to-diabetes-4235fad1f222>



- ▶ IBM Watson Health Showcases Progress Tackling Diabetes at American Diabetes Association's 76th Scientific Sessions:  
<http://www.ibm.com/press/us/en/pressrelease/49904.wss>
- ▶ Conversation:  
<https://www.ibm.com/watson/developercloud/conversation.html>
- ▶ Watson Conversation Service Overview video:  
<https://youtu.be/1rTl1WEbg5U>
- ▶ Language Translator:  
<https://www.ibm.com/watson/developercloud/language-translator.html>
- ▶ Language Translator Service by IBM Watson:  
<https://www.youtube.com/watch?v=bYtVaQxJ994>
- ▶ Natural Language Classifier:  
<https://www.ibm.com/watson/developercloud/nl-classifier.html>
- ▶ IBM Watson Natural Language Classifier:  
<https://youtu.be/h1ZiUIvYdD8>
- ▶ Retrieve and Rank:  
<https://www.ibm.com/watson/developercloud/retrieve-rank.html>
- ▶ Overview of the IBM Watson Visual Recognition service:  
<https://www.ibm.com/watson/developercloud/doc/visual-recognition/index.html>
- ▶ Guidelines for training classifiers:  
<https://www.ibm.com/watson/developercloud/doc/visual-recognition/customizing.html>
- ▶ About Speech to Text:  
<https://www.ibm.com/watson/developercloud/doc/speech-to-text/index.html>
- ▶ Using customization (Speech to Text):  
<https://www.ibm.com/watson/developercloud/doc/speech-to-text/custom.html>
- ▶ Speech to Text API reference:  
<https://www.ibm.com/watson/developercloud/speech-to-text/api/v1/#introduction>
- ▶ About Text to Speech:  
<https://www.ibm.com/watson/developercloud/doc/text-to-speech/index.html>
- ▶ Understanding customization (Text to Speech):  
<https://www.ibm.com/watson/developercloud/doc/text-to-speech/custom-intro.html>
- ▶ Using customization (Text to Speech):  
<https://www.ibm.com/watson/developercloud/doc/text-to-speech/custom-using.html>
- ▶ Text to Speech API reference:  
<https://www.ibm.com/watson/developercloud/text-to-speech/api/v1/>
- ▶ Migrating from AlchemyLanguage:  
<https://www.ibm.com/watson/developercloud/doc/natural-language-understanding/migrating.html>

- ▶ Supported languages:  
<https://www.ibm.com/watson/developercloud/doc/natural-language-understanding/index.html#supported-languages>
- ▶ Natural Language Understanding documentation:  
<https://www.ibm.com/watson/developercloud/doc/natural-language-understanding/index.html>
- ▶ Customizing (Natural Language Understanding):  
<https://www.ibm.com/watson/developercloud/doc/natural-language-understanding/customizing.html>
- ▶ Getting started with custom models:  
<https://www.ibm.com/watson/developercloud/doc/natural-language-understanding/customizing.html#getting-started-with-custom-models>
- ▶ Discovery:  
<https://www.ibm.com/watson/developercloud/discovery.html>
- ▶ Watson Discovery Service Overview:  
<https://youtu.be/9ks-cEG6KPs>
- ▶ Tutorial: Creating a project:  
[https://www.ibm.com/watson/developercloud/doc/wks/tutorials.html#wks\\_tutintro](https://www.ibm.com/watson/developercloud/doc/wks/tutorials.html#wks_tutintro)
- ▶ Tutorial: Creating a machine-learning model:  
[https://www.ibm.com/watson/developercloud/doc/wks/tutorials.html#wks\\_tutml\\_intro](https://www.ibm.com/watson/developercloud/doc/wks/tutorials.html#wks_tutml_intro)
- ▶ Integrating Discovery service with IBM Watson Knowledge Studio:  
<https://www.ibm.com/watson/developercloud/doc/discovery/integrate-wks.html>

## Help from IBM

IBM Support and downloads

[ibm.com/support](https://www.ibm.com/support)

IBM Global Services

[ibm.com/services](https://www.ibm.com/services)









SG24-8387-00

ISBN 073844264X

Printed in U.S.A.

Get connected

