ACEBOT using IBM WATSON Assistant

A Mini Project Report
Submitted in partial fulfillment of the requirements
for the award of degree of

Bachelor of Technology
In
COMPUTER SCIENCE AND ENGINEERING

By
Koganti Lakshmi Durga (16AG1A0527)

Under the guidance of

Mr. P. Chiranjeevi
Assistant Professor

Department of Computer Science and Engineering

ACE Engineering College
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad,)
Ghatkesar ,Medchal. Dist – 501 301.
November - 2019

ACKNOWLEDGEMENT

I express my gratitude to my beloved **Principal, Dr. B. L. Raju, ACE Engineering College** for permitting us to carry out this project.

I express my gratitude to my beloved **HOD, Prof. K. Jaya Bharathi, Department of Computer Science and Engineering, ACE Engineering College, Hyderabad** for the valuable guidance and suggestions, keen interest and through encouragement extended throughout period of project work.

I express my gratitude to our beloved Internal Guide **Mr. P. Chiranjeevi, Assistant Professor in the Department of Computer Science and Engineering, ACE Engineering College** for his valuable suggestions and rare insights, for constant source of encouragement and inspiration throughout my project work.

I would like thank our Project co-ordinator **Dr. G.Sreenivasulu**, **Associate Professor of Computer Science and Engineering of ACE Engineering College** for his continuous efforts in monitoring, helping and guiding us in all the aspects for completion of my project work.

I express my thanks to all those who contributed for the successful completion of my project work.

CONTENTS

List of Figures

CHAPTER- 1
INTRODUCTION

1.1 Introduction

A chatbot is a piece of software that conducts a conversation via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, although as of 2019, they are far short of being able to pass the Turing test. Chatbots are typically used in dialog systems for various practical purposes including customer service or information acquisition. Some chatbots use sophisticated natural language processing systems, but many simpler ones scan for keywords within the input, then pull a reply with the most matching keywords, or the most similar wording pattern, from a database.

The term "ChatterBot" was originally coined by Michael Mauldin (creator of the first Verbot, Julia) in 1994 to describe these conversational programs. Today, most chatbots are accessed via virtual assistants such as Google Assistant and Amazon Alexa, via messaging apps such as Facebook Messenger or WeChat, or via individual organizations' apps and websites. Chatbots can be classified into usage categories that include conversational commerce (e-commerce via chat), education, entertainment, finance, health, news, and productivity.

Beyond chatbots, **Conversational AI** refers to the use of messaging apps, speech-based assistants and chatbots to automate communication and create personalized customer experiences at scale.

In 1950, Alan Turing's famous article "Computing Machinery and Intelligence" was published,which proposed what is now called the Turing test as a criterion of intelligence. This criterion depends on the ability of a computer program to impersonate a human in a real-time written conversation with a human judge to the extent that the judge is unable to distinguish reliably—on the basis of the conversational content alone—between the program and a real human. The notoriety of Turing's proposed test stimulated great interest in Joseph Weizenbaum's program ELIZA, published in 1966, which seemed to be able to fool users into believing that they were conversing with a real human.

However Weizenbaum himself did not claim that ELIZA was genuinely intelligent, and the introduction to his paper presented it more as a debunking exercise:

[In] artificial intelligence ... machines are made to behave in wondrous ways, often sufficient to dazzle even the most experienced observer. But once a particular program is unmasked, once its inner workings are explained ... its magic crumbles away; it stands revealed as a mere collection of procedures ... The observer says to himself "I could have written that". With that thought he moves the program in question from the shelf marked "intelligent" to that reserved for curios ... The object of this paper is to cause just such a re-evaluation of the program about to be "explained". Few programs ever needed it more.

ELIZA's key method of operation (copied by chatbot designers ever since) involves the recognition of clue words or phrases in the input, and the output of corresponding pre-prepared or pre-programmed responses that can move the conversation forward in an apparently meaningful way (e.g. by responding to any input that contains the word 'MOTHER' with 'TELL ME MORE ABOUT YOUR FAMILY'). Thus an illusion of understanding is generated, even though the processing involved has been merely superficial. ELIZA showed that such an illusion is surprisingly easy to generate, because human judges are so ready to give the benefit of the doubt when conversational responses are *capable of being interpreted* as "intelligent".

Interface designers have come to appreciate that humans' readiness to interpret computer output as genuinely conversational—even when it is actually based on rather simple pattern-matching—can be exploited for useful purposes. Most people prefer to engage with programs that are human-like, and this gives chatbot-style techniques a potentially useful role in interactive systems that need to elicit information from users, as long as that information is relatively straightforward and falls into predictable categories. Thus, for example, online help systems can usefully employ chatbot techniques to identify the area of help that users require, potentially providing a "friendlier" interface than a more formal search or menu system. This sort of usage holds the prospect of moving chatbot technology from Weizenbaum's "shelf ... reserved for curios" to that marked "genuinely useful computational methods".

The classic historic early chatbots are ELIZA (1966) and PARRY (1972). More recent notable programs include A.L.I.C.E., Jabberwacky and D.U.D.E (Agence Nationale de la Recherche and CNRS 2006). While ELIZA and PARRY were used exclusively to simulate typed conversation, many chatbots now include functional features such as games and web searching abilities. In 1984, a book called *The Policeman's Beard is Half Constructed* was published, allegedly written by the chatbot Racter (though the program as released would not have been capable of doing so).

One pertinent field of AI research is natural language processing. Usually, weak AI fields employ specialized software or programming languages created specifically for the narrow function required. For example, A.L.I.C.E. uses a markup language called AIML, which is specific to its function as a conversational agent, and has since been adopted by various other developers of, so called, Alicebots. Nevertheless, A.L.I.C.E. is still purely based on pattern matching techniques without any reasoning capabilities, the same technique ELIZA was using back in 1966. This is not strong AI, which would require sapience and logical reasoning abilities.

Jabberwacky learns new responses and context based on real-time user interactions, rather than being driven from a static database. Some more recent chatbots also combine real-time learning with evolutionary algorithms that optimise their ability to communicate based on each conversation held. Still, there is currently no general purpose conversational artificial intelligence, and some software developers focus on the practical aspect, information retrieval.

Chatbot competitions focus on the Turing test or more specific goals. Two such annual contests are the Loebner Prize and The Chatterbox Challenge (the latter has been offline since 2015, however materials can still be found from web archives).

DBpedia created a chatbot during the GSoC of 2017. and can communicate through Facebook Messenger. DBpedia started in 2007 and allows to extract structured content from the Wikipedia dataset, along with many other datasets. DBpedia is currently one of the biggest representatives of Linked Open Data (LOD).

1.2 Aim

Many companies' chatbots run on messaging apps or simply via SMS. They are used for B2C customer service, sales and marketing. In 2016, Facebook Messenger allowed developers to place chatbots on their platform. There were 30,000 bots created for Messenger in the first six months, rising to 100,000 by September 2017. Since September 2017, this has also been as part of a pilot program on WhatsApp. Airlines KLM and Aeroméxico both announced their participation in the testing; both airlines had previously launched customer services on the Facebook Messenger platform. The bots usually appear as one of the user's contacts, but can sometimes act as participants in a group chat. Many banks, insurers, media companies, e-commerce companies, airlines, hotel chains, retailers, health care providers, government entities and restaurant chains have used chatbots to answer simple questions, increase customer engagement, for promotion, and to offer additional ways to order from them. Previous generations of chatbots were present on company websites, e.g. Ask Jenn from Alaska Airlines which debuted in 2008 or Expedia's virtual customer service agent which launched in 2011. The newer generation of chatbots includes IBM Watson-powered "Rocky", introduced in February 2017 by the New York City-based e-commerce company Rare Carat to provide information to prospective diamond buyers. We sometimes pass our time by chatting with different chatterboxes available on the internet, which is often aimed for such purposes or just entertainment. The chatbots have embedded knowledge which helps them to identify the user’s query and give a response to it. The college enquiry ACEBOT project is built using algorithms that analyses user’s queries and understand user’s message. This system is a web application which provides answer to the query of the parents. Students just have to query through the bot which is used for chatting. The user can query any college related activities through the system. The user doesn’t have to personally go to the college for enquiry. The system analyses the question and then answers to the user. The system answers to the query as if it is answered by the person. With the help of algorithms, the system answers the query asked by the parents. The aim of this project is to contribute to the solution of the problem of direct communication between applicants and the university. The main objectives of the project are to improve support for users, users can get exact information, the bot can handle a greater number of people simultaneously, users need not wait for any administrators to clear queries.

CHAPTER-2
LITERATURE SURVEY

Watson is a question-answering computer system capable of answering questions posed in natural language,[2] developed in IBM's DeepQA project by a research team led by principal investigator David Ferrucci.[3] Watson was named after IBM's founder and first CEO, industrialist Thomas J. Watson.[4][5]

The computer system was initially developed to answer questions on the quiz show *Jeopardy!*[6] and, in 2011, the Watson computer system competed on *Jeopardy!* against legendary champions Brad Rutter and Ken Jennings,[4][7] winning the first place prize of $1 million.[8]

In February 2013, IBM announced that Watson software system's first commercial application would be for utilization management decisions in lung cancer treatment at Memorial Sloan Kettering Cancer Center, New York City, in conjunction with WellPoint (now Anthem).[9] Manoj Saxena, IBM Watson's then business chief, said in 2013 that 90% of nurses in the field who use Watson now follow its guidance.[10]

2.1 Description

Watson was created as a question answering (QA) computing system that IBM built to apply advanced natural language processing, information retrieval, knowledge representation, automated reasoning, and machine learning technologies to the field of open domain question answering.[2]

The key difference between QA technology and document search is that document search takes a keyword query and returns a list of documents, ranked in order of relevance to the query (often based on popularity and page ranking), while QA technology takes a question expressed in natural *language, seeks to understand it in much greater detail, and returns a precise answer to the question.*[12]

When created, IBM stated that, "more than 100 different techniques are used to analyze natural language, identify sources, find and generate hypotheses, find and score evidence, and merge and rank hypotheses."[13]

In recent years, the Watson capabilities have been extended and the way in which Watson works has been changed to take advantage of new deployment models (Watson on IBM Cloud) and evolved machine learning capabilities and optimised hardware available to developers and researchers. It is no longer purely a question answering (QA) computing system designed from Q&A pairs but can now 'see', 'hear', 'read', 'talk', 'taste', 'interpret', 'learn' and 'recommend'.

2.1.1 Software and Hardware

Watson uses IBM's DeepQA software and the Apache UIMA (Unstructured Information Management Architecture) framework implementation. The system was written in various languages, including Java, C++, and Prolog, and runs on the SUSE Linux Enterprise Server 11 operating system using the Apache Hadoop framework to provide distributed computing.[14][15][16]

The system is workload-optimized, integrating massively parallel POWER7 processors and built on IBM's *DeepQA* technology,[17] which it uses to generate hypotheses, gather massive evidence, and analyze data.[2] Watson employs a cluster of ninety IBM Power 750 servers, each of which uses a 3.5 GHz POWER7 eight-core processor, with four threads per core. In total, the system has 2,880 POWER7 processor threads and 16 terabytes of RAM.[17]

According to John Rennie, Watson can process 500 gigabytes, the equivalent of a million books, per second.[18] IBM's master inventor and senior consultant, Tony Pearson, estimated Watson's hardware cost at about three million dollars.[19] Its Linpack performance stands at 80 TeraFLOPs, which is about half as fast as the cut-off line for the Top 500 Supercomputers list.[20] According to Rennie, all content was stored in Watson's RAM for the Jeopardy game because data stored on hard drives would be too slow to be competitive with human Jeopardy champions.[18]

2.1.2 Data

The sources of information for Watson include encyclopedias, dictionaries, thesauri, newswire articles and literary works. Watson also used databases, taxonomies and ontologies. Specifically, DBPedia, WordNet and Yago were used.[21] The IBM team provided Watson with millions of documents, including dictionaries, encyclopedias and other reference material that it could use to build its knowledge.[22]

2.2 Operation

Watson parses questions into different keywords and sentence fragments in order to find statistically related phrases.[22] Watson's main innovation was not in the creation of a new algorithm for this operation but rather its ability to quickly execute hundreds of proven language analysis algorithms simultaneously.[22][24] The more algorithms that find the same answer independently the more likely Watson is to be correct.[22] Once Watson has a small number of potential solutions, it is able to check against its database to ascertain whether the solution makes sense or not.[22]

2.2.1 Comparison with human players

Watson's basic working principle is to parse keywords in a clue while searching for related terms as responses. This gives Watson some advantages and disadvantages compared with human *Jeopardy!* players[25] Watson has deficiencies in understanding the contexts of the clues. As a result, human players usually generate responses faster than Watson, especially to short clues.[22] Watson's programming prevents it from using the popular tactic of buzzing before it is sure of its response.[22] Watson has consistently better reaction time on the buzzer once it has generated a response, and is immune to human players' psychological tactics, such as jumping between categories on every clue.[22][26]

In a sequence of 20 mock games of *Jeopardy*, human participants were able to use the average six to seven seconds that Watson needed to hear the clue and decide whether to signal for responding.[22] During that time, Watson also has to evaluate the response and determine whether it is sufficiently confident in the result to signal.[22] Part of the system used to win the *Jeopardy!* contest was the electronic circuitry that receives the "ready" signal and then examined whether Watson's confidence level was great enough to activate the buzzer. Given the speed of this circuitry compared to the speed of human reaction times, Watson's reaction time was faster than the human contestants except when the human anticipated (instead of reacted to) the ready signal.[27] After signaling, Watson speaks with an electronic voice and gives the responses in *Jeopardy!*'s question format.[22] Watson's voice was synthesized from recordings that actor Jeff Woodman made for an IBM text-to-speech program in 2004.[28]

The *Jeopardy!* staff used different means to notify Watson and the human players when to buzz,[27] which was critical in many rounds.[26] The humans were notified by a light, which took them tenths of a second to perceive.[29][30] Watson was notified by an electronic signal and could activate the buzzer within about eight milliseconds.[31] The humans tried to compensate for the perception delay by anticipating the light,[32] but the variation in the anticipation time was generally too great to fall within Watson's response time.[26] Watson did not attempt to anticipate the notification signal.[30][32]

2.3 History

2.3.1 Development

Since Deep Blue's victory over Garry Kasparov in chess in 1997, IBM had been on the hunt for a new challenge. In 2004, IBM Research manager Charles Lickel, over dinner with coworkers, noticed that the restaurant they were in had fallen silent. He soon discovered the cause of this evening hiatus: Ken Jennings, who was then in the middle of his successful 74-game run on *Jeopardy!*. Nearly the entire restaurant had piled toward the televisions, mid-meal, to watch the phenomenon. Intrigued by the quiz show as a possible challenge for IBM, Lickel passed the idea on, and in 2005, IBM Research executive Paul Horn backed Lickel up, pushing for someone in his department to take up the challenge of playing *Jeopardy!* with an IBM system. Though he initially had trouble finding any research staff willing to take on what looked to be a much more complex challenge than the wordless game of chess, eventually David Ferrucci took him up on the offer.[33] In competitions managed by the United States government, Watson's predecessor, a system named Piquant, was usually able to respond correctly to only about 35% of clues and often required several minutes to respond.[34][35][36] To compete successfully on *Jeopardy!*, Watson would need to respond in no more than a few seconds, and at that time, the problems posed by the game show were deemed to be impossible to solve.[22]

In initial tests run during 2006 by David Ferrucci, the senior manager of IBM's Semantic Analysis and Integration department, Watson was given 500 clues from past *Jeopardy!* programs. While the best real-life competitors buzzed in half the time and responded correctly to as many as 95% of clues, Watson's first pass could get only about 15% correct. During 2007, the IBM team was given three to five years and a staff of 15 people to solve the problems.[22] By 2008, the developers had advanced Watson such that it could compete with *Jeopardy!* champions.[22] By February 2010, Watson could beat human *Jeopardy!* contestants on a regular basis.[37]

During the game, Watson had access to 200 million pages of structured and unstructured content consuming four terabytes of disk storage[14] including the full text of the 2011 edition of Wikipedia,[38] but was not connected to the Internet.[39][22] For each clue, Watson's three most probable responses were displayed on the television screen. Watson consistently outperformed its human opponents on the game's signaling device, but had trouble in a few categories, notably those having short clues containing only a few words.

Although the system is primarily an IBM effort, Watson's development involved faculty and graduate students from Rensselaer Polytechnic Institute, Carnegie Mellon University, University of Massachusetts Amherst, the University of Southern California's Information Sciences Institute, the University of Texas at Austin, the Massachusetts Institute of Technology, and the University of Trento,[11] as well as students from New York Medical College.[40]

2.3.2 *Jeopardy!*

Preparation

In 2008, IBM representatives communicated with *Jeopardy!* executive producer Harry Friedman about the possibility of having Watson compete against Ken Jennings and Brad Rutter, two of the most successful contestants on the show, and the program's producers agreed.[22][41] Watson's differences with human players had generated conflicts between IBM and *Jeopardy!* staff during the planning of the competition.[25] IBM repeatedly expressed concerns that the show's writers would exploit Watson's cognitive deficiencies when writing the clues, thereby turning the game into a Turing test. To alleviate that claim, a third party randomly picked the clues from previously written shows that were never broadcast.[25] *Jeopardy!* staff also showed concerns over Watson's reaction time on the buzzer. Originally Watson signalled electronically, but show staff requested that it press a button physically, as the human contestants would.[42] Even with a robotic "finger" pressing the buzzer, Watson remained faster than its human competitors. Ken Jennings noted, "If you're trying to win on the show, the buzzer is all", and that Watson "can knock out a microsecond-precise buzz every single time with little or no variation. Human reflexes can't compete with computer circuits in this regard."[26][32][43] Stephen Baker, a journalist who recorded Watson's development in his book *Final Jeopardy*, reported that the conflict between IBM and *Jeopardy!* became so serious in May 2010 that the competition was almost canceled.[25] As part of the preparation, IBM constructed a mock set in a conference room at one of its technology sites to model the one used on *Jeopardy!*. Human players, including former *Jeopardy!* contestants, also participated in mock games against Watson with Todd

Alan Crain of *The Onion* playing host.[22] About 100 test matches were conducted with Watson winning 65% of the games.[44]

To provide a physical presence in the televised games, Watson was represented by an "avatar" of a globe, inspired by the IBM "smarter planet" symbol. Jennings described the computer's avatar as a "glowing blue ball criss-crossed by 'threads' of thought—42 threads, to be precise",[23] and stated that the number of thought threads in the avatar was an in-joke referencing the significance of the number 42 in Douglas Adams' *Hitchhiker's Guide to the Galaxy*.[23] Joshua Davis, the artist who designed the avatar for the project, explained to Stephen Baker that there are 36 triggerable states that Watson was able to use throughout the game to show its confidence in responding to a clue correctly; he had hoped to be able to find forty-two, to add another level to the *Hitchhiker's Guide* reference, but he was unable to pinpoint enough game states.[45]

A practice match was recorded on January 13, 2011, and the official matches were recorded on January 14, 2011. All participants maintained secrecy about the outcome until the match was broadcast in February.[46]

Practice match

In a practice match before the press on January 13, 2011, Watson won a 15-question round against Ken Jennings and Brad Rutter with a score of $4,400 to Jennings's $3,400 and Rutter's $1,200, though Jennings and Watson were tied before the final $1,000 question. None of the three players responded incorrectly to a clue.[47]

First match

The first round was broadcast February 14, 2011, and the second round, on February 15, 2011. The right to choose the first category had been determined by a draw won by Rutter.[48] Watson, represented by a computer monitor display and artificial voice, responded correctly to the second clue and then selected the fourth clue of the first category, a deliberate strategy to find the Daily Double as quickly as possible.[49] Watson's guess at the Daily Double location was correct. At the end of the first round, Watson was tied with Rutter at $5,000; Jennings had $2,000.[48]

Watson's performance was characterized by some quirks. In one instance, Watson repeated a reworded version of an incorrect response offered by Jennings. (Jennings said "What are the '20s?" in reference to the 1920s. Then Watson said "What is 1920s?") Because Watson could not recognize other contestants' responses, it did not know that Jennings had already given the same response. In another instance, Watson was initially given credit for a response of "What is a leg?" after Jennings incorrectly responded "What is: he only had one hand?" to a clue about George Eyser (the correct response was, "What is: he's missing a leg?"). Because Watson, unlike a human, could not have been responding to Jennings's mistake, it was decided that this response was incorrect. The broadcast version of the episode was edited to omit Trebek's original acceptance of Watson's response.[50] Watson also demonstrated complex wagering strategies on the Daily Doubles, with one bet at $6,435 and another at $1,246.[51] Gerald Tesauro, one of the IBM researchers who worked on Watson, explained that Watson's wagers were based on its confidence level for the category and a complex regression model called the Game State Evaluator.[52]

Watson took a commanding lead in Double Jeopardy!, correctly responding to both Daily Doubles. Watson responded to the second Daily Double correctly with a 32% confidence score.[51]

Although it wagered only $947 on the clue, Watson was the only contestant to miss the Final Jeopardy! response in the category U.S. CITIES ("Its largest airport was named for a World War II hero; its second largest, for a World War II battle"). Rutter and Jennings gave the correct response of Chicago, but Watson's response was "What is Toronto?????"[51][53][54] Ferrucci offered reasons why Watson would appear to have guessed a Canadian city: categories only weakly suggest the type of response desired, the phrase "U.S. city" did not appear in the question, there are cities named Toronto in the U.S., and Toronto in Ontario has an American League baseball team.[55] Dr. Chris Welty, who also worked on Watson, suggested that it may not have been able to correctly parse the second part of the clue, "its second largest, for a World War II battle" (which was not a standalone clause despite it following a semicolon, and required context to understand that it was referring to a second-largest *airport*).[56] Eric Nyberg, a professor at Carnegie Mellon University and a member of the development team, stated that the error occurred because Watson does not possess the comparative knowledge to discard that potential response as not viable.[54] Although not displayed to the audience as with non-Final Jeopardy! questions, Watson's second choice was Chicago. Both Toronto and Chicago were well below Watson's confidence threshold, at 14% and 11% respectively. (This lack of confidence was the reason for the multiple question marks in Watson's response.)

The game ended with Jennings with $4,800, Rutter with $10,400, and Watson with $35,734.[51]

Second match

During the introduction, Trebek (a Canadian native) joked that he had learned Toronto was a U.S. city, and Watson's error in the first match prompted an IBM engineer to wear a Toronto Blue Jays jacket to the recording of the second match.[57]

In the first round, Jennings was finally able to choose a Daily Double clue,[58] while Watson responded to one Daily Double clue incorrectly for the first time in the Double Jeopardy! Round.[59] After the first round, Watson placed second for the first time in the competition after Rutter and Jennings were briefly successful in increasing their dollar values before Watson could respond.[59][60] Nonetheless, the final result ended with a victory for Watson with a score of $77,147, besting Jennings who scored $24,000 and Rutter who scored $21,600.[61]

Final outcome

The prizes for the competition were $1 million for first place (Watson), $300,000 for second place (Jennings), and $200,000 for third place (Rutter). As promised, IBM donated 100% of Watson's winnings to charity, with 50% of those winnings going to World Vision and 50% going to World Community Grid.[62] Similarly, Jennings and Rutter donated 50% of their winnings to their respective charities.[63]

In acknowledgment of IBM and Watson's achievements, Jennings made an additional remark in his Final Jeopardy! response: "I for one welcome our new computer overlords", echoing a similar memetic reference to the episode "Deep Space Homer" on *The Simpsons*, in which TV news presenter Kent Brockman speaks of welcoming "our new insect overlords".[64][65] Jennings later wrote an article for *Slate*, in which he stated:

> *IBM has bragged to the media that Watson's question-answering skills are good for more than annoying Alex Trebek. The company sees a future in which fields like medical diagnosis, business analytics, and tech support are automated by question-answering software like Watson. Just as factory jobs were eliminated in the 20th century by new assembly-line robots, Brad and I were the first knowledge-industry workers put out of work by the new generation of 'thinking' machines. 'Quiz show contestant' may be the first job made redundant by Watson, but I'm sure it won't be the last.*[23]

Philosophy

Philosopher John Searle argues that Watson—despite impressive capabilities—cannot actually think.[66] Drawing on his Chinese room thought experiment, Searle claims that Watson, like other computational machines, is capable only of manipulating symbols, but has no ability to understand the meaning of those symbols; however, Searle's experiment has its detractors.[67]

Match against members of the United States Congress

On February 28, 2011, Watson played an untelevised exhibition match of *Jeopardy!* against members of the United States House of Representatives. In the first round, Rush D. Holt, Jr. (D-NJ, a former *Jeopardy!* contestant), who was challenging the computer with Bill Cassidy (R-LA, later Senator from Louisiana), led with Watson in second place. However, combining the scores between all matches, the final score was $40,300 for Watson and $30,000 for the congressional players combined.[68]

IBM's Christopher Padilla said of the match, "The technology behind Watson represents a major advancement in computing. In the data-intensive environment of government, this type of technology can help organizations make better decisions and improve how government helps its citizens."[68]

CHAPTER-3
ANALYSIS

3.1 Existing System

According to IBM, "The goal is to have computers start to interact in natural human terms across a range of applications and processes, understanding the questions that humans ask and providing answers that humans can understand and justify."[37] It has been suggested by Robert C. Weber, IBM's general counsel, that Watson may be used for legal research.[69] The company also intends to use Watson in other information-intensive fields, such as telecommunications, financial services and government.[70]

Watson is based on commercially available IBM Power 750 servers that have been marketed since February 2010. IBM also intends to market the DeepQA software to large corporations, with a price in the millions of dollars, reflecting the $1 million needed to acquire a server that meets the minimum system requirement to operate Watson. IBM expects the price to decrease substantially within a decade as the technology improves.[22]

Commentator Rick Merritt said that "there's another really important reason why it is strategic for IBM to be seen very broadly by the American public as a company that can tackle tough computer problems. A big slice of [IBM's profit] comes from selling to the U.S. government some of the biggest, most expensive systems in the world."[71]

In 2013, it was reported that three companies were working with IBM to create apps embedded with Watson technology. Fluid is developing an app for retailers, one called "The North Face", which is designed to provide advice to online shoppers. Welltok is developing an app designed to give people advice on ways to engage in activities to improve their health. MD Buyline is developing an app for the purpose of advising medical institutions on equipment procurement decisions.[72][73]

In November 2013, IBM announced it would make Watson's API available to software application providers, enabling them to build apps and services that are embedded in Watson's capabilities. To build out its base of partners who create applications on the Watson platform, IBM consults with a network of venture capital firms, which advise IBM on which of their portfolio companies may be a logical fit for what IBM calls the Watson Ecosystem. Thus far, roughly 800 organizations and individuals have signed up with IBM, with interest in creating applications that could use the Watson platform.[74]

On January 30, 2013, it was announced that Rensselaer Polytechnic Institute would receive a successor version of Watson, which would be housed at the Institute's technology park and be available to researchers and students.[75] By summer 2013, Rensselaer had become the first university to receive a Watson computer.[76]

On February 6, 2014, it was reported that IBM plans to invest $100 million in a 10-year initiative to use Watson and other IBM technologies to help countries in Africa address development problems, beginning with healthcare and education.[77]

On June 3, 2014, three new Watson Ecosystem partners were chosen from more than 400 business concepts submitted by teams spanning 18 industries from 43 countries. "These bright and enterprising organizations have discovered innovative ways to apply Watson that can deliver demonstrable business benefits", said Steve Gold, vice president, IBM Watson Group. The winners were Majestyk Apps with their adaptive educational platform, FANG (Friendly Anthropomorphic Networked Genome);[78][79] Red Ant with their retail sales trainer;[80] and GenieMD[81] with their medical recommendation service.[82]

On July 9, 2014, Genesys Telecommunications Laboratories announced plans to integrate Watson to improve their customer experience platform, citing the sheer volume of customer data to analyze is staggering.[83]

Watson has been integrated with databases including *Bon Appétit* magazine to perform a recipe generating platform.[84]

Watson is being used by Decibel, a music discovery startup, in its app MusicGeek which uses the supercomputer to provide music recommendations to its users. The use of the artificial intelligence of Watson has also been found in the hospitality industry. GoMoment uses Watson for its Rev1 app, which gives hotel staff a way to quickly respond to questions from guests.[85] Arria NLG has built an app that helps energy companies stay within regulatory guidelines, making it easier for managers to make sense of thousands of pages of legal and technical jargon.

OmniEarth, Inc. uses Watson computer vision services to analyze satellite and aerial imagery, along with other municipal data, to infer water usage on a property-by-property basis, helping water districts in drought-stricken California improve water conservation efforts.[86]

In September 2016, Condé Nast has started using IBM's Watson to help build and strategize social influencer campaigns for brands. Using software built by IBM and Influential, Condé Nast's clients will be able to know which influencer's demographics, personality traits and more best align with a marketer and the audience it is targeting.[87]

In February 2017, Rare Carat, a New York City-based startup and e-commerce platform for buying diamonds and diamond rings, introduced an IBM Watson-powered artificial intelligence chatbot called "Rocky" to assist novice diamond buyers through the daunting process of purchasing a diamond. As part of the IBM Global Entrepreneur Program, Rare Carat received the assistance of IBM in the development of the Rocky Chat Bot.[88][89][90] In May 2017, IBM partnered with the Pebble Beach Company to use Watson as a concierge.[91] Watson's artificial intelligence was added to an app developed by Pebble Beach and was used to guide visitors around the resort. The mobile app was designed by IBM iX and hosted on the IBM Cloud. It uses Watson's Conversation applications programming interface.

In November 2017, in Mexico City, the Experience Voices of Another Time was opened at the National Museum of Anthropology using IBM Watson as an alternative to visiting a museum.[92]

3.2 Proposed System

An application that converses with customers is a conversation app. To enable the conversation, you create a *bot*, which is the software the provides automated responses to user input. You can use the IBM Watson® Assistant service to quickly build, test, and deploy bots or virtual agents across mobile devices, messaging platforms, or even on a physical robot to create natural conversations between your apps and users.

When you create an application that enables people to interact with a bot, the application passes input from a person to the bot, and can include context information. Then, the application presents responses from the bot to the person. The bot uses natural language understanding and machine learning to extract meaning from the input.

This process identifies the user’s *intent*, which is the goal or purpose of the input. This process can also identify an *entity*, which is a term that is mentioned in the input and that clarifies the purpose.

You train your bot to recognize intents and entities in the input that people submit. To train your bot on intents, you supply many examples of user input and indicate the intents that they map to. To train your bot on entities, you list the values for each entity and the synonyms that people might enter.
As you add information, the bot trains itself; you don't need to initiate the training. Names of intents are preceded by a # sign and names of entities are preceded by an @ sign in the user interface. These symbols help you better understand categories.
After you train your system to recognize intents and entities, you teach it how to respond when it recognizes those intents and entities. You can use the dialog builder to create conversations with people, providing responses based on the intents and entities that you recognize in their inputs.

Conversation Concepts

This example explains the key concepts for productively using the Watson Assistant service. You'll learn about the necessary artifacts to create a bot that answers customers' questions in the context of airline flights.

Intent

Watson has read more than 300 million words and understands the subtleties of language, including idioms and syntax. By using this knowledge, along with natural language processing and deep learning techniques, Watson unpacks customers' questions to reveal the intent behind them. No matter how the question is phrased, Watson can understand the purpose behind the question:

- "Can I change my flight?"
- "I need to get on another flight!"
- "I missed my flight. Can I get on the next one?"

In the first example, intent is the purpose of the question: change flights.

Entities

Watson enhances its understanding of the intent by extracting key entities. Watson recognizes specific words or phrases in each question; for example, "I need to go to JFK now." In this example, the entity is JFK airport

Tone

To see things from the customer's point of view, Watson can extract the tone of the question to understand whether the customer is joyful, angry, or stressed. For example, the tone in this statement is anger: "I can’t believe you are going to charge me for flights." However, in this statement, the tone is joy: “Thanks for your help. I understand there will be a fee.”

Context

A dialog uses the intent and entity that were identified plus context from the application to interact with the person. Context variables can be passed to Watson Assistant service from an application. This context information can ultimately provide a response that is personalized and meaningful to the person. Context attributes can include variables such as stateofResidence, customerRank, and more. Watson can also extract the context of previous questions from the conversation so that it has a complete picture before it responds. This image shows examples from previous exchanges. The context might be, "Is that a business class seat on the itinerary?"

Response

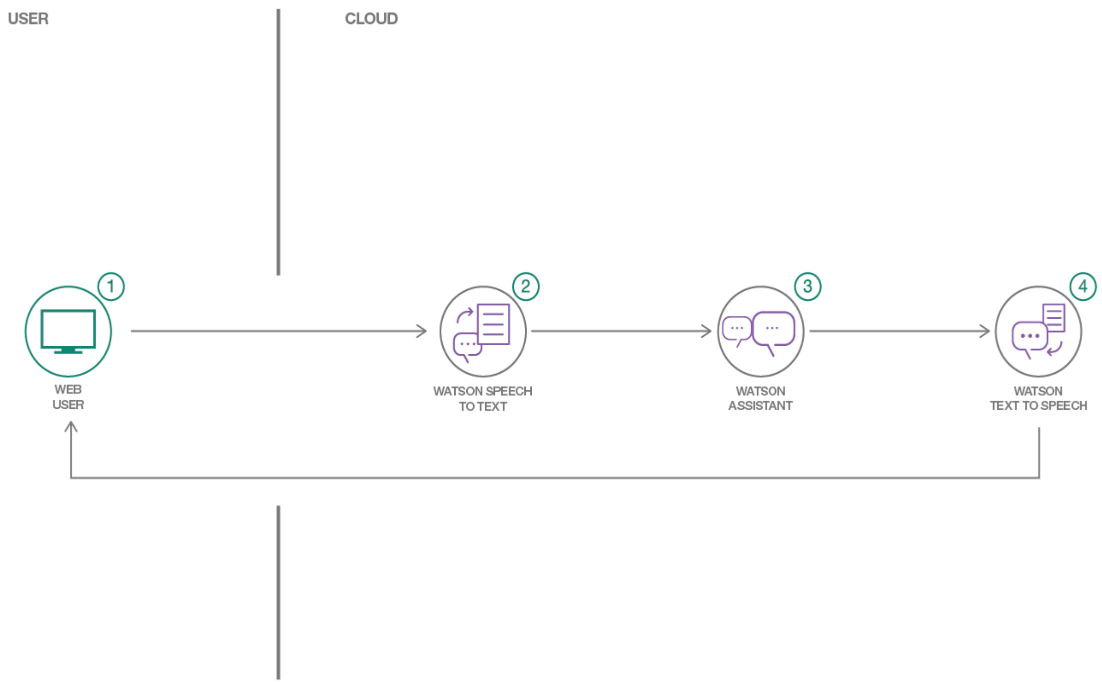
Watson can be trained to respond with empathy. By combining intent, entity, tone, and context, Watson responds with the appropriate action.

Knowledge expansion and the corpus

If Watson is unsure of the answer, it can use a technique that is called *knowledge expansion* to find an answer from within its corpus. A *corpus* is a large collection of trusted texts. It’s a body of written material, spoken material, images, or video upon which a linguistic or AI analysis is based. Consider this example: "I’m traveling with a guitar and I really don’t want to check it. Can I take it as a carry-on?" By using knowledge expansion, a search of the corpus finds the guidelines about carry-on luggage. Specific guidelines about guitars are analyzed and used to answer the question.

CHAPTER-4
DESIGN

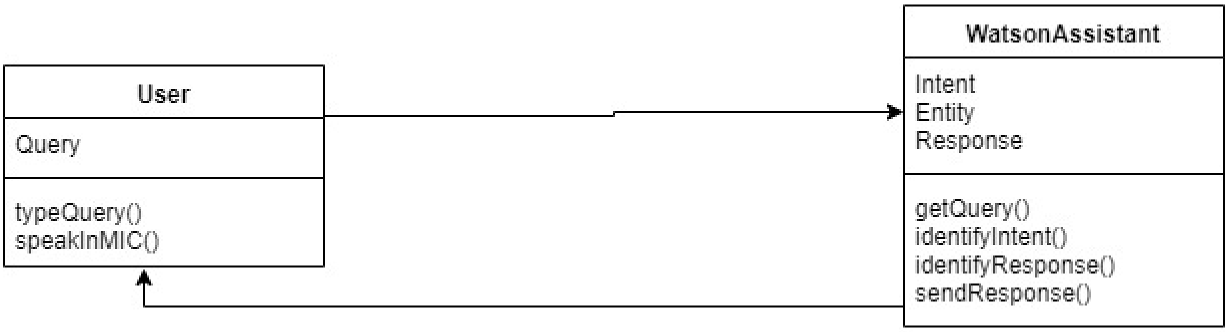
4.1 Flow Control



Flow Control Diagram

- ➔ User selects the microphone option on the browser and speaks.
- ➔ The voice is passed on to Watson Speech To Text using a Web Socket connection.
- ➔ The text from Watson Speech to Text is extracted and sent as input to Watson Assistant.
- ➔ The response from Watson Assistant is passed onto Watson Text to Speech.
- ➔ The audio output is sent to the web application and played back to the user, while the UI also displays the same text.

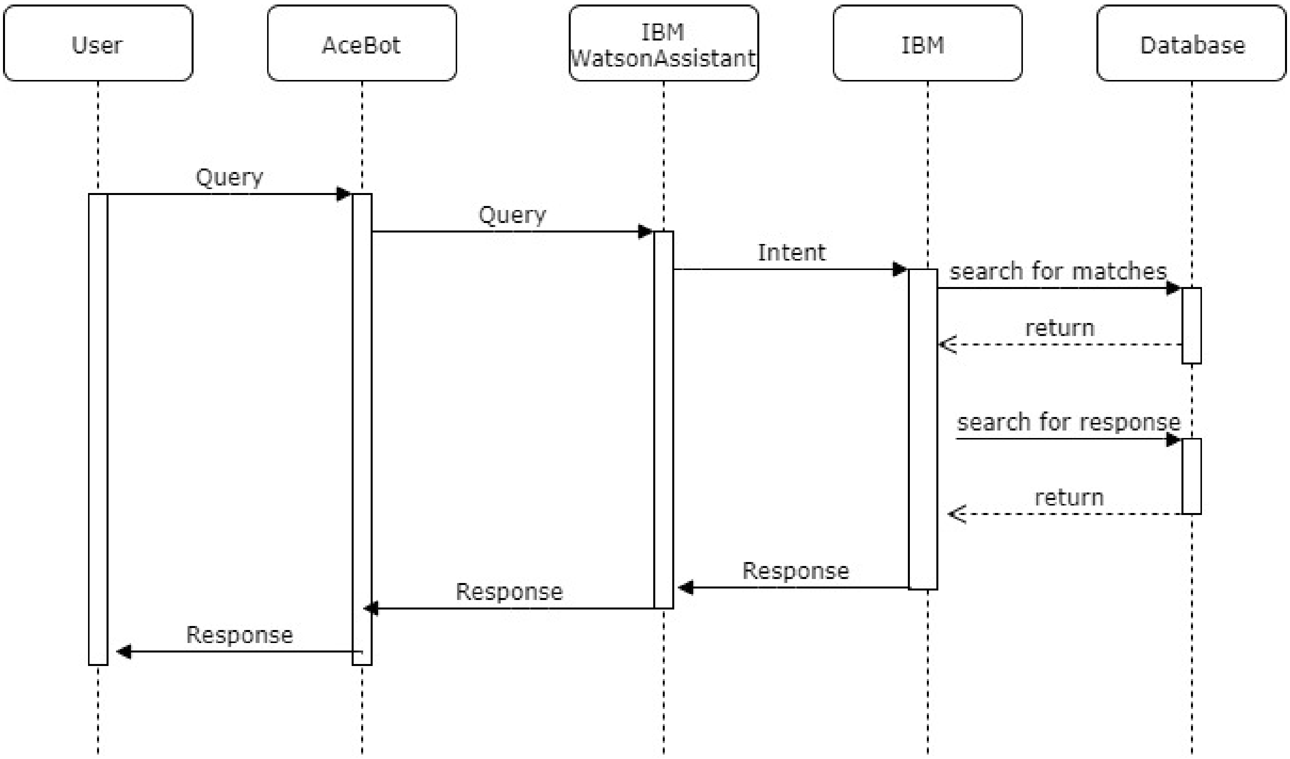
4.2 Class Diagram



Class Diagram

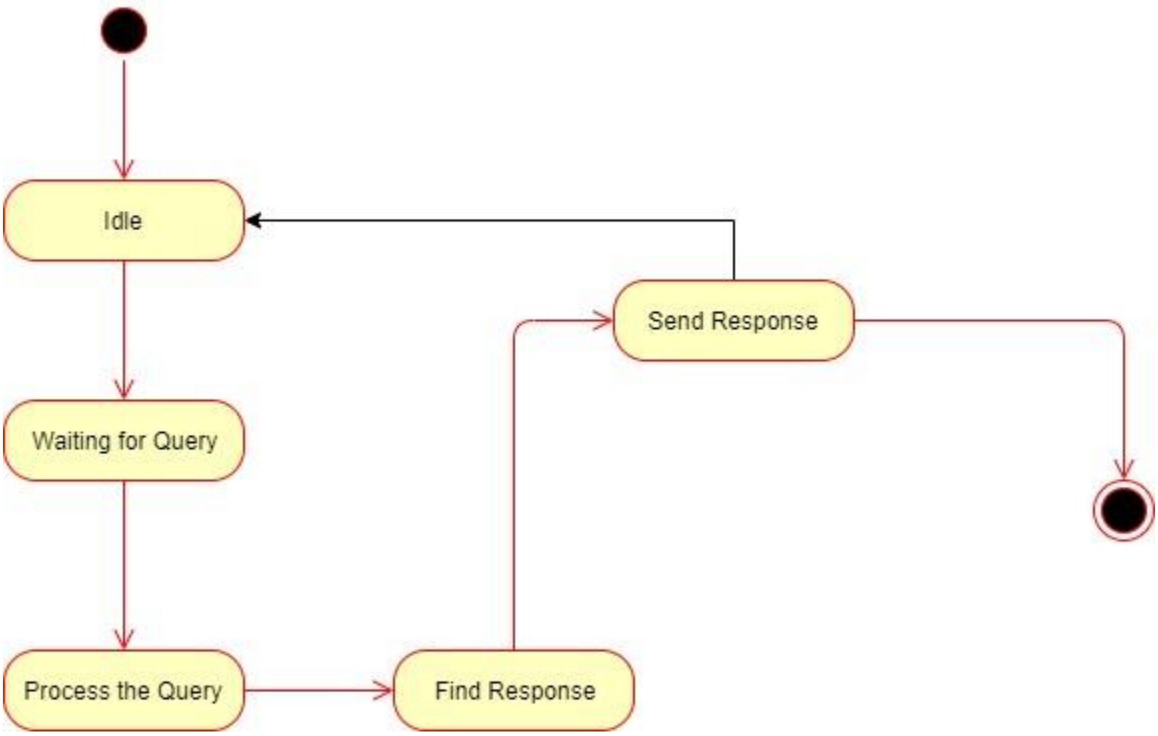
- ➔ We have two classes User and WatsonAssistant
- ➔ User has one attribute which is the query for which he wants a response.
- ➔ User can either talk in mic or else can type the query in the chat interface.
- ➔ WatsonAssistant have attributes of intent, entity and response
- ➔ WatsonAssistant identify the correct intent, entity based on the user query.
- ➔ WatsonAssistant will then find the suitable response and send the response to the user.

4.3 Sequence Diagram



- We have 5 classes namely,
    - User
    - AceBot
    - IBM WatsonAssistant
    - IBM
    - Database
- The query is sent from user to the acebot and further to the IBM WatsonAssistant.
- This then is transformed into an intent and is sent to the IBM which searches the database for the generation of the response to the same.
- After the response has been found it is then transferred it’s way back to the user respectively.

4.4 State Chart Diagram



- The process is first in idle mode and keeps waiting for the query.
- When the query is sent from the user side it processes it.
- It then finds the response to the query.
- This response is then sent to the user.

CHAPTER-5
IMPLEMENTATION

5.1 Modules

This example explains the key concepts for productively using the Watson Assistant service. You'll learn about the necessary artifacts to create a bot that answers customers' questions in the context of airline flights.

5.1.1 Intent

Watson has read more than 300 million words and understands the subtleties of language, including idioms and syntax. By using this knowledge, along with natural language processing and deep learning techniques, Watson unpacks customers' questions to reveal the intent behind them. No matter how the question is phrased, Watson can understand the purpose behind the question:

- "Can I change my flight?"
- "I need to get on another flight!"
- "I missed my flight. Can I get on the next one?"

In the first example, intent is the purpose of the question: change flights.

5.1.2 Entities

Watson enhances its understanding of the intent by extracting key entities. Watson recognizes specific words or phrases in each question; for example, "I need to go to JFK now." In this example, the entity is JFK airport.

5.1.3 Tone

To see things from the customer's point of view, Watson can extract the tone of the question to understand whether the customer is joyful, angry, or stressed. For example, the tone in this statement is anger: "I can’t believe you are going to charge me for flights." However, in this statement, the tone is joy: “Thanks for your help. I understand there will be a fee.”

5.1.4 Context

A dialog uses the intent and entity that were identified plus context from the application to interact with the person. Context variables can be passed to Watson Assistant service from an application. This context information can ultimately provide a response that is personalized and meaningful to the person. Context attributes can include variables such as stateofResidence, customerRank, and more. Watson can also extract the context of previous questions from the conversation so that it has a complete picture before it responds. This image shows examples from previous exchanges. The context might be, "Is that a business class seat on the itinerary?"

5.1.5 Response

Watson can be trained to respond with empathy. By combining intent, entity, tone, and context, Watson responds with the appropriate action.

5.1.6 Knowledge expansion and the corpus

If Watson is unsure of the answer, it can use a technique that is called *knowledge expansion* to find an answer from within its corpus. A *corpus* is a large collection of trusted texts. It’s a body of written material, spoken material, images, or video upon which a linguistic or AI analysis is based.
Consider this example: "I’m traveling with a guitar and I really don’t want to check it. Can I take it as a carry-on?" By using knowledge expansion, a search of the corpus finds the guidelines about carry-on luggage. Specific guidelines about guitars are analyzed and used to answer the question.

5.2 Building the Application

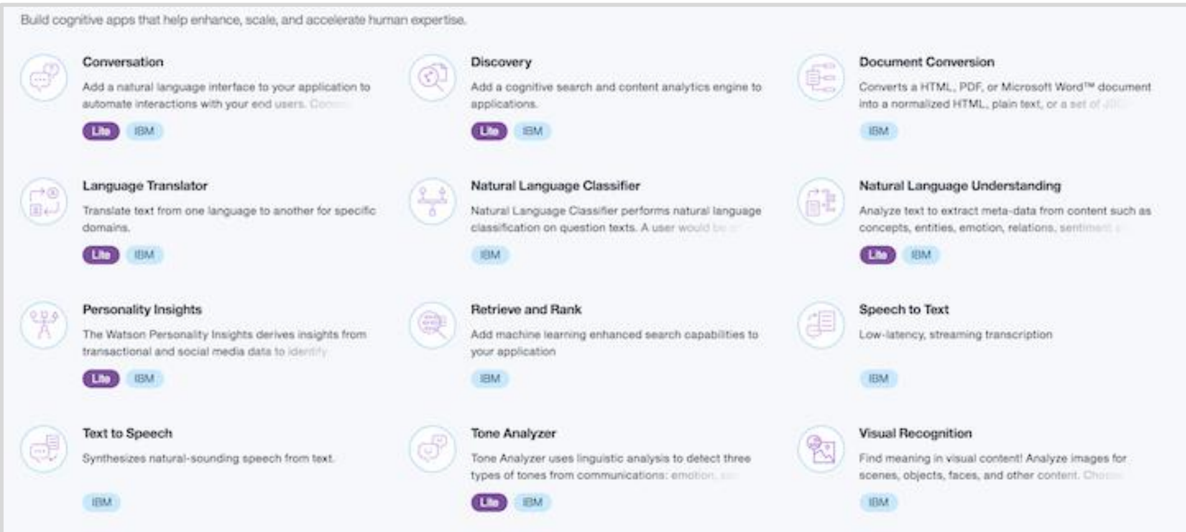
Prerequisites

- You must have an IBM Cloud account. The account is free and provides access to everything you need to develop, track, plan, and deploy apps. The account requires an IBMid. If you don't have an IBMid, you can create one when you register.
- Your IBM Cloud account must have at least 8 GB of runtime and container memory, plus access to provision up to 10 services.

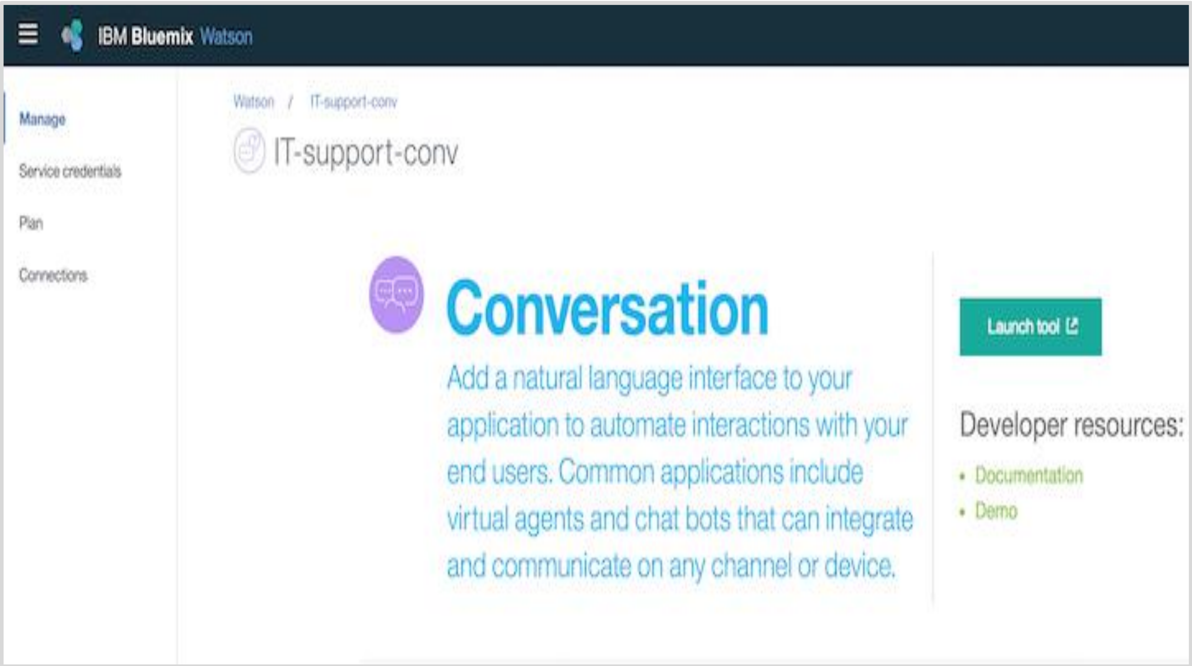
Task 1: Create the Assistant service

The first task is to create an instance of Watson Assistant on IBM Cloud.

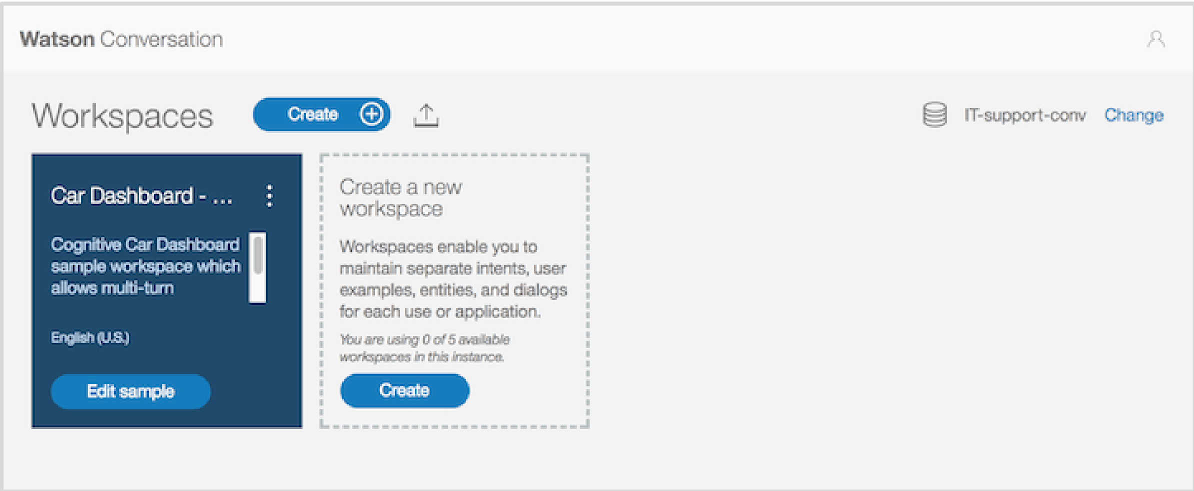
1. Make sure that you are logged in to your IBM Cloud account. Click Catalog and then click Services > Watson > Assistant.



2. For the service name, type ITSupportConversation. Click Create. This page is displayed:



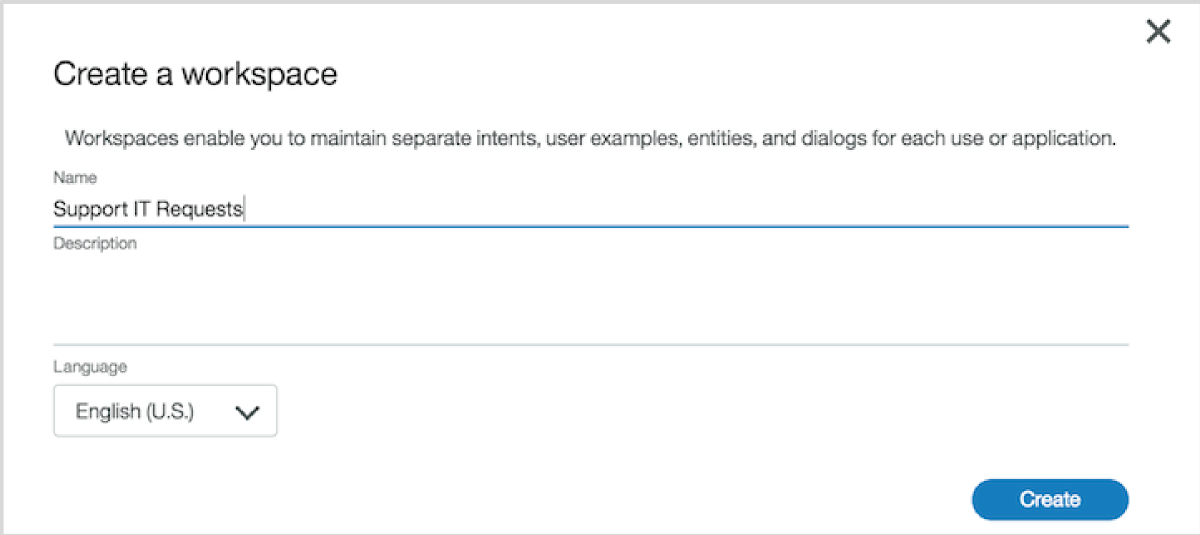
3. Click Launch tool to open the Watson Assistant workspace.



Task 2: Create a workspace

You must use workspaces to maintain separate intents, user examples, entities, and dialog flows for each application. Watson Assistant uses a step-by-step approach to guide you to create workspace, intents, and so forth.

1. In the Workspaces section, click Create.



2. Type a name for the workspace. In the examples throughout this tutorial, the workspace name is *SupportHelpDesk*.

Task 3: Create intents

Add intents. An *intent* is a group of examples of things that a user might say to communicate a specific goal or idea. To identify intents, start with something that a user might want and then list the ways that the user might describe it. For each intent, think of the various ways that a user might express his or her desire—those are the examples. Examples can be developed by using a crowdsourcing approach.

For example, in a discussion with the support team, you might gather this set of standard questions that support received from users:

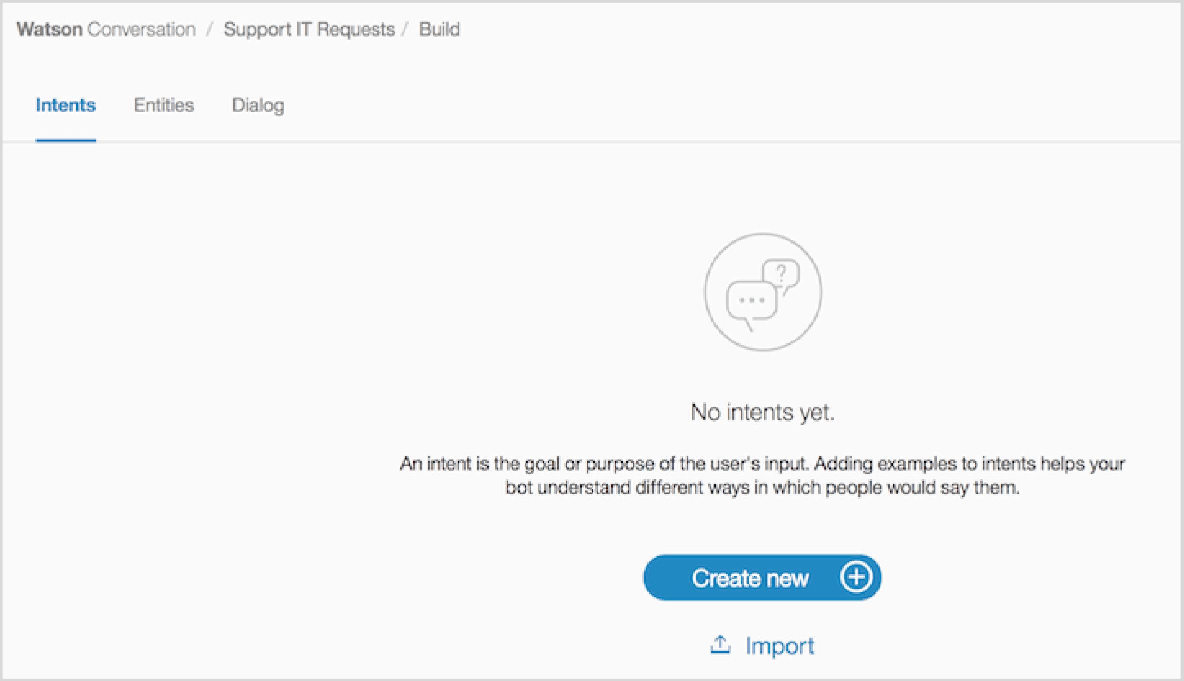
- What is the status of the business application? I could not access it.
- How to get access to a business application?
- How to reset my password for a specific application?
- When to renew my workstation?
- How to bring my own device and connect it to enterprise network?

Each of those questions is documented as a frequently asked question in the support team's document repository. Some solutions persist in a relational database in the form of application > problem > solution.

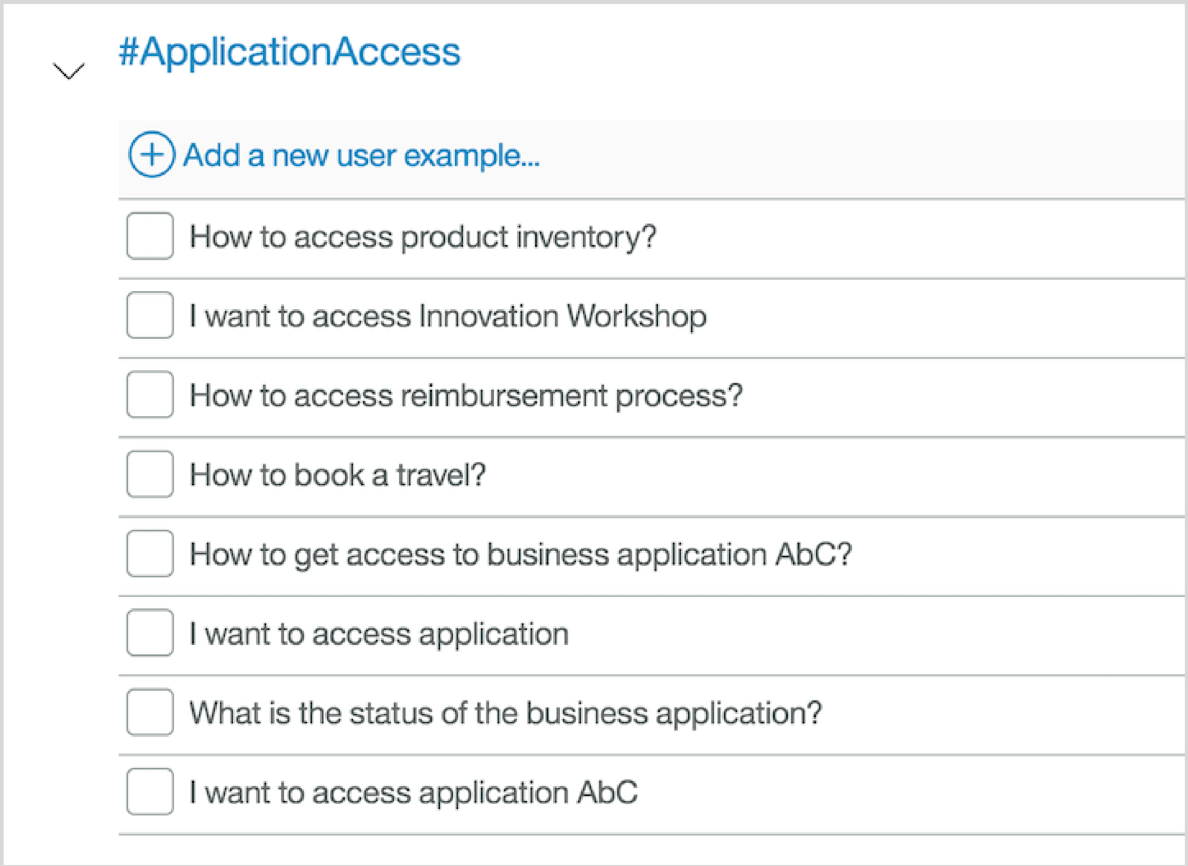
Based on the questions, you can extract these intents:

- Access to a business applications like expense report, AbC.
- Reset password
- Access to supplier on boarding business process
- Bring your own device

1. Add those intents to the workspace: From the Build page, click Intents and click Create new.

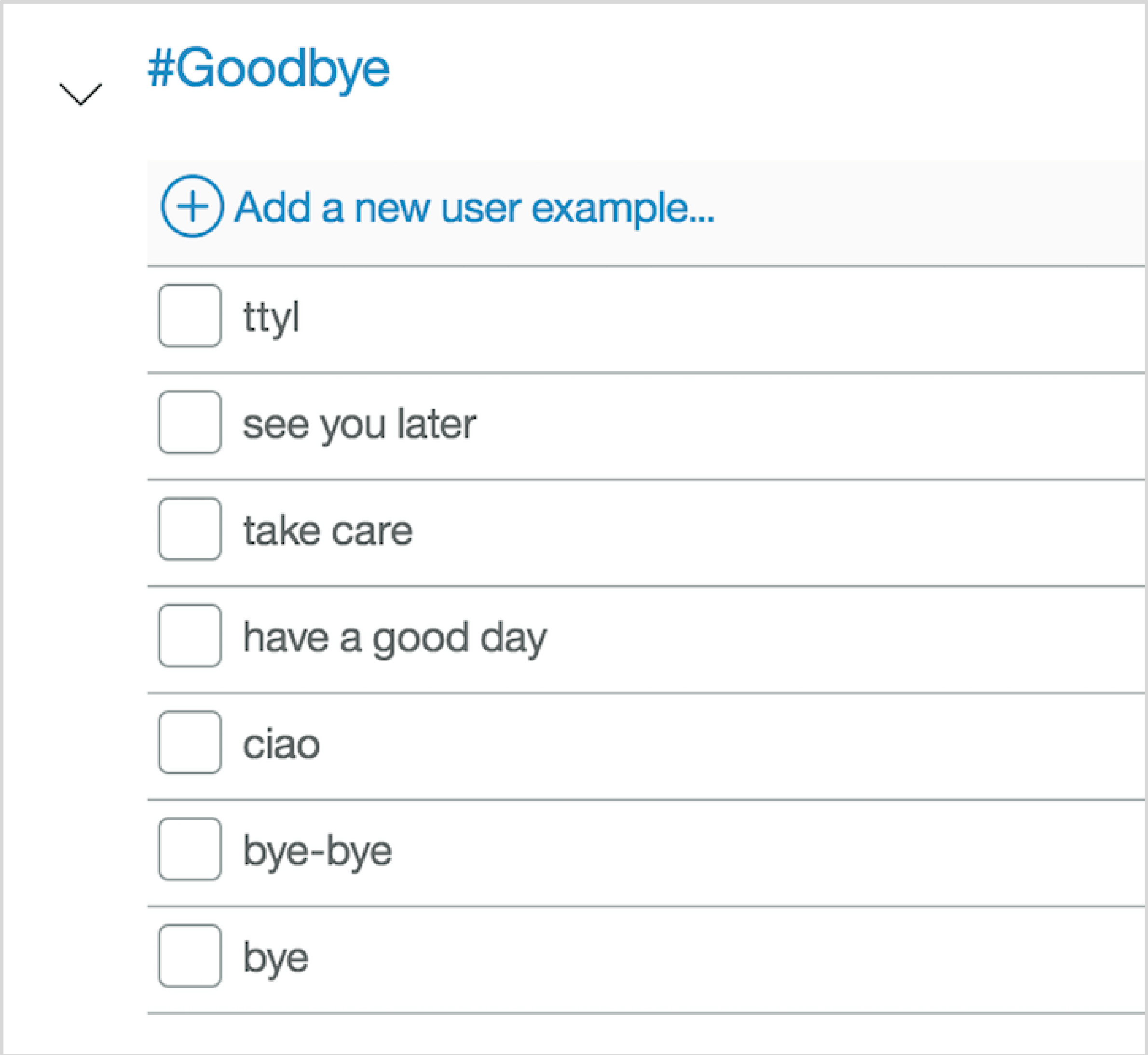


2. For the intent name, type applicationAccess after the number sign (#).



3. For each intent, add examples to train the conversation for intent recognition. You can enter the same examples as shown in the previous image.

4. Create the Goodbyes intent and add examples for it.



Because many intents can be reused from conversation to conversation implementations, you can define .csv files and import them in the Conversation Tool Intents. The .csv format is shown in this example with one intent per line:

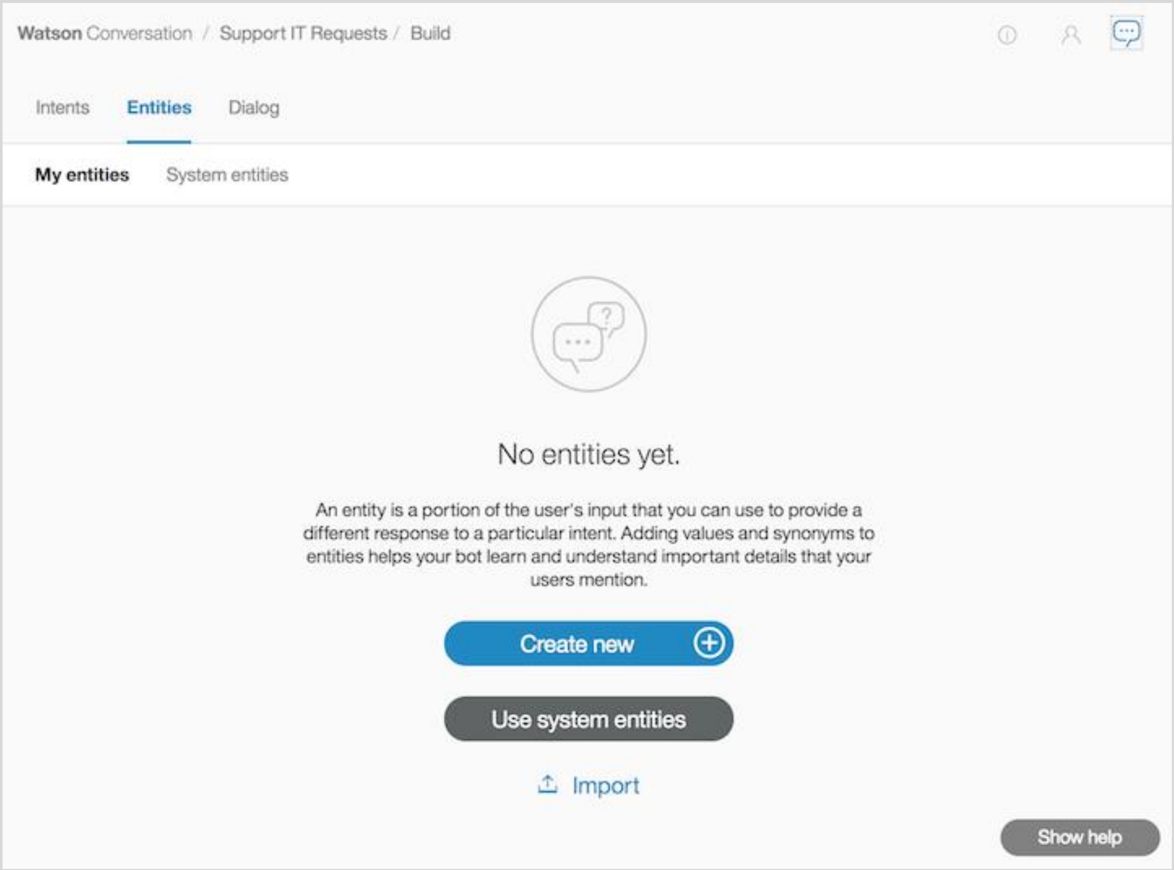
```
hello,greeting
hi,greeting
how are you,greeting
bonjour,greeting
howdy,greeting
what's up,greeting
good morning,greeting
```

5. To get the IT support demonstration intents, click the Import link on the Intents page to import the wcs-workspace/ITSupport-Intents.csv file from the GitHub repository

Task 4: Add entities

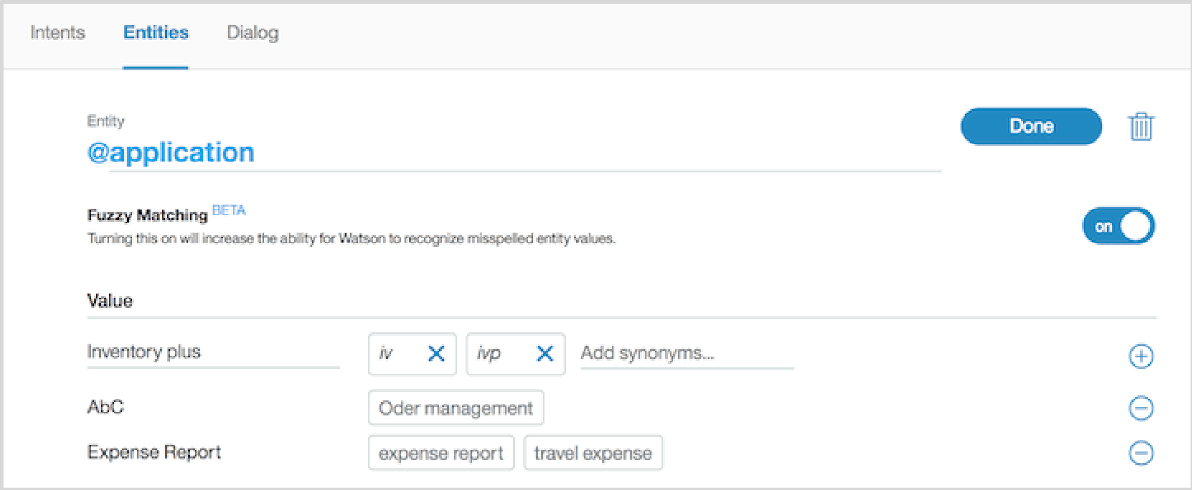
An *entity* is a portion of the user's input that you can use to provide a different response to a particular intent.

1. Click Entities. On the Entities page, click Create new.



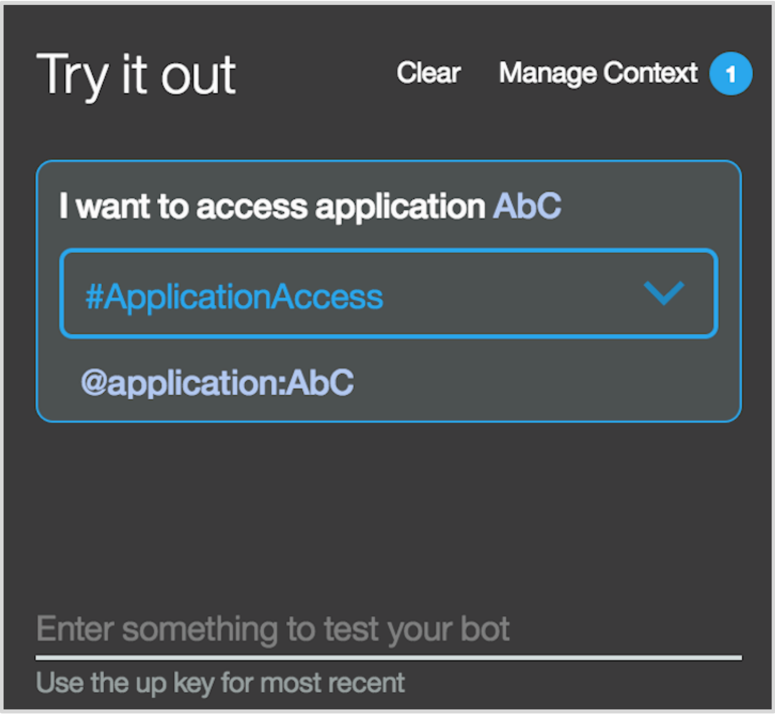
Adding values and synonyms to entities helps your chatbot learn important details that your users might mention.
Each entity definition includes a set of specific entity values that can be used to trigger different responses. Each value can have multiple synonyms that define different ways that the same value can be specified in user input.

2. Create entities to represent to the application what the user wants to access.



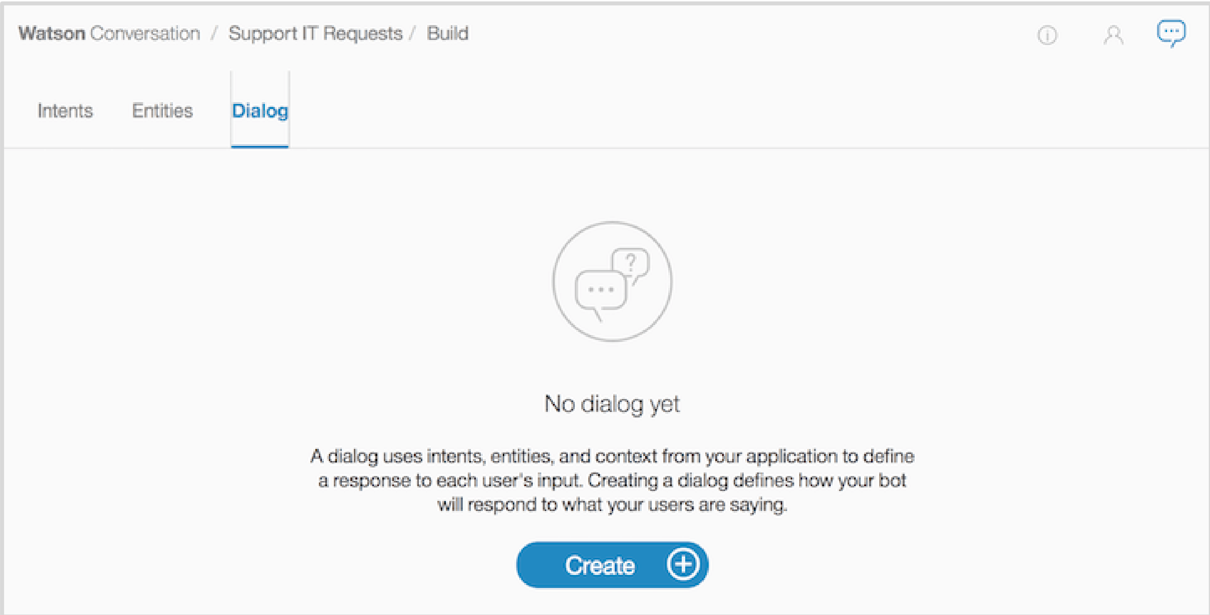
Fuzzy logic is a feature that allows Watson Assistant to accept misspelled words. You can enable this feature at the entity level. As you did for intents, you can reuse entities' definitions through the export and import capabilities. Import the wcs-workspace/ITSupport-Entities.csv file.

3. If you click the Ask Watson icon immediately after you import the entities, the Watson is training message is displayed. Watson Assistant classifies the entities. You can unit-test the entities by entering I want to access application AbC. The following figure shows both the intent and entity (@application:AbC) extracted by Watson Assistant:

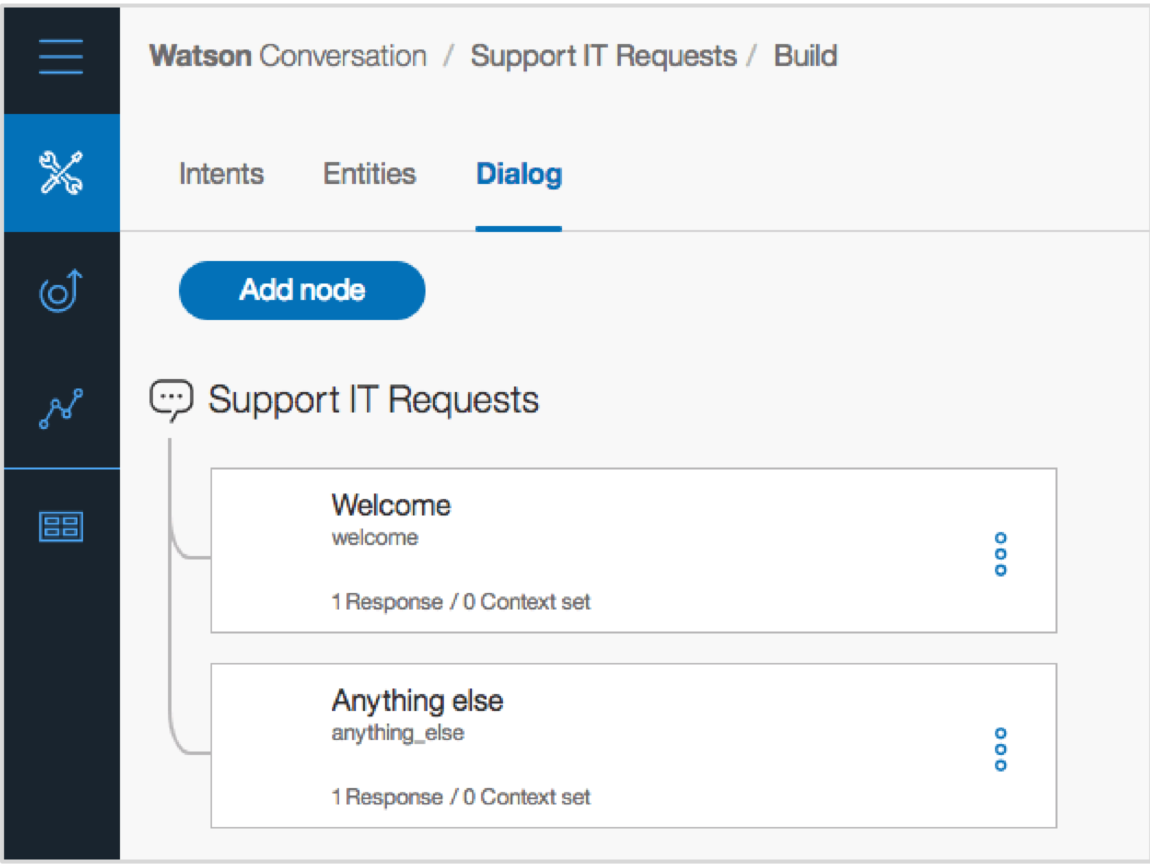


Task 5: Build the dialog

After you specify your intents and entities, you can construct the dialog flow.



A dialog is made up of nodes that define steps in the conversation.

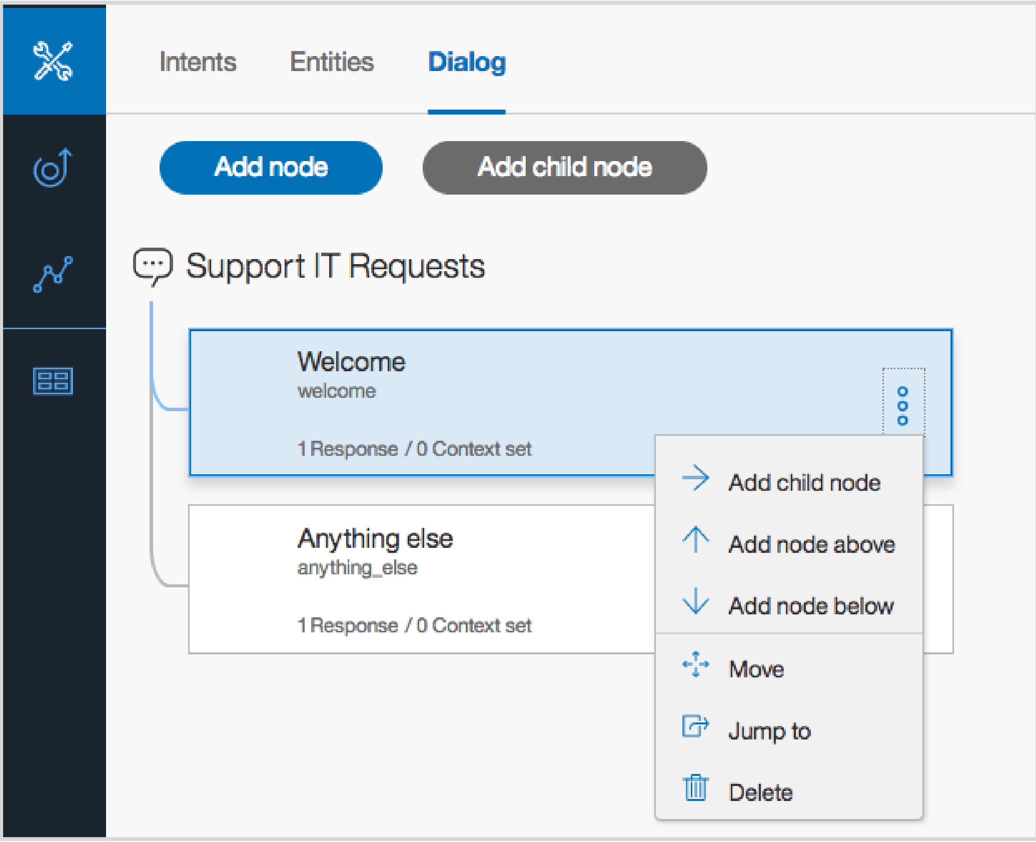


In the previous image, two dialog nodes are shown. The first node is the standard welcome message. The other node is a catch-all node named "Anything else." Dialog nodes are chained in a tree structure to create an interactive conversation with the user. The evaluation starts at the top, so the welcome node is assessed before the "Anything else" node.

If you click the welcome node, the standard Watson response is "Hello. How can I help you?" To validate how the flow works, you can click the Ask Watson icon.

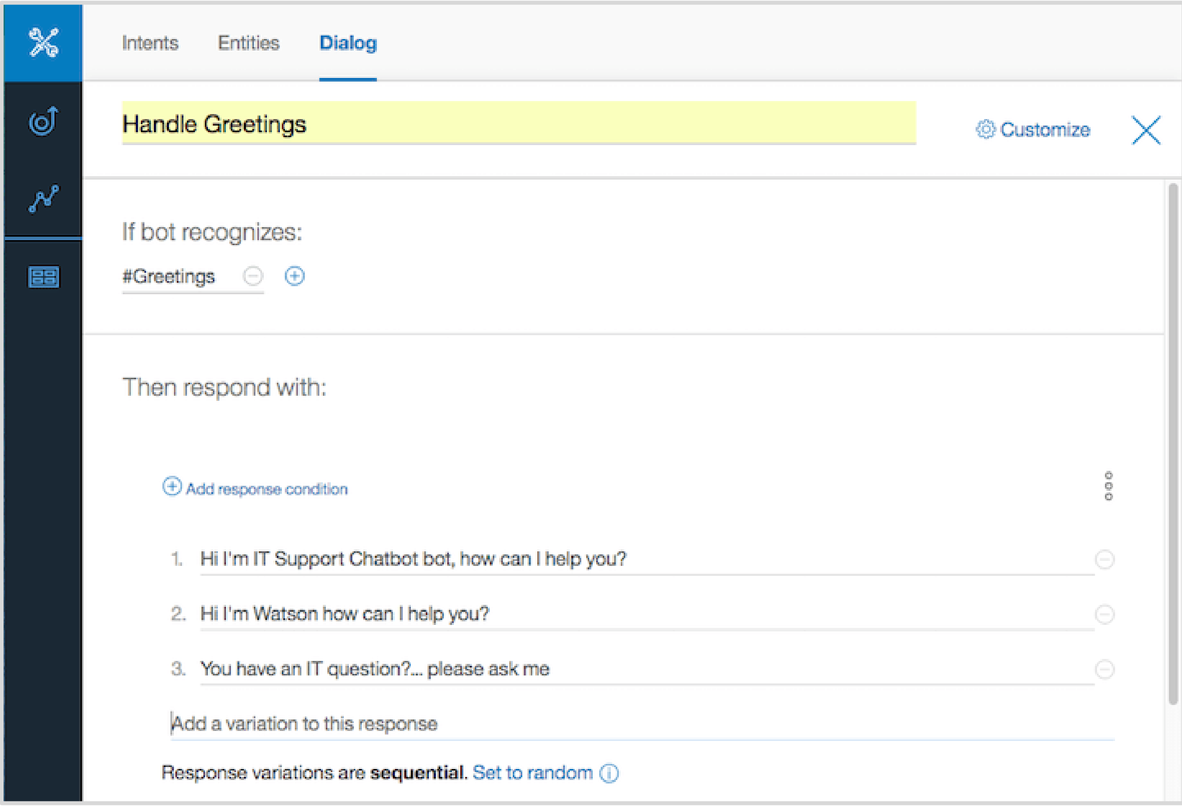
Define the greetings node

1. The first node addresses greetings in a response to a query such as "hello." Click the welcome node and click Add node below:



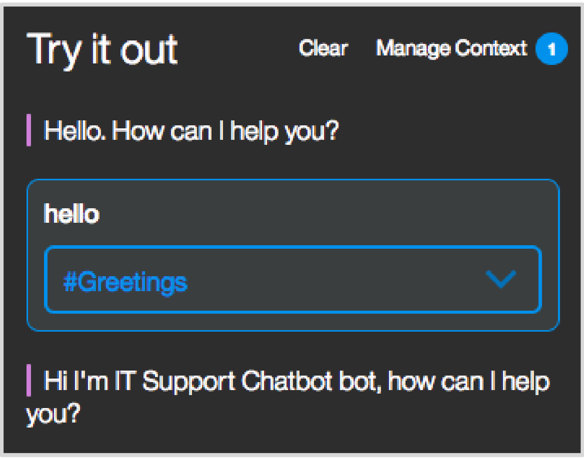
A new node is added between the welcome and "Anything else" nodes. At each node level, you can expand the conversation by adding nodes. If you add nodes at the same level, the flows are parallel. Adding a child node creates a dependent track of conversation, and the conversation branches out into a tree structure.

2. Name the new node Handle Greetings. In the If bot recognizes field, change the value to #Greetings. The number sign (#) represents a prefix for intent. The condition is triggered when the Watson natural language classifier classifies the query as a greeting intent.
3. Add these responses:



The previous image also illustrates how to use the multiple responses pattern to avoid being repetitive. The bot can present different answers to the same query. You can allow the system to randomly select an answer from the list of potential responses.

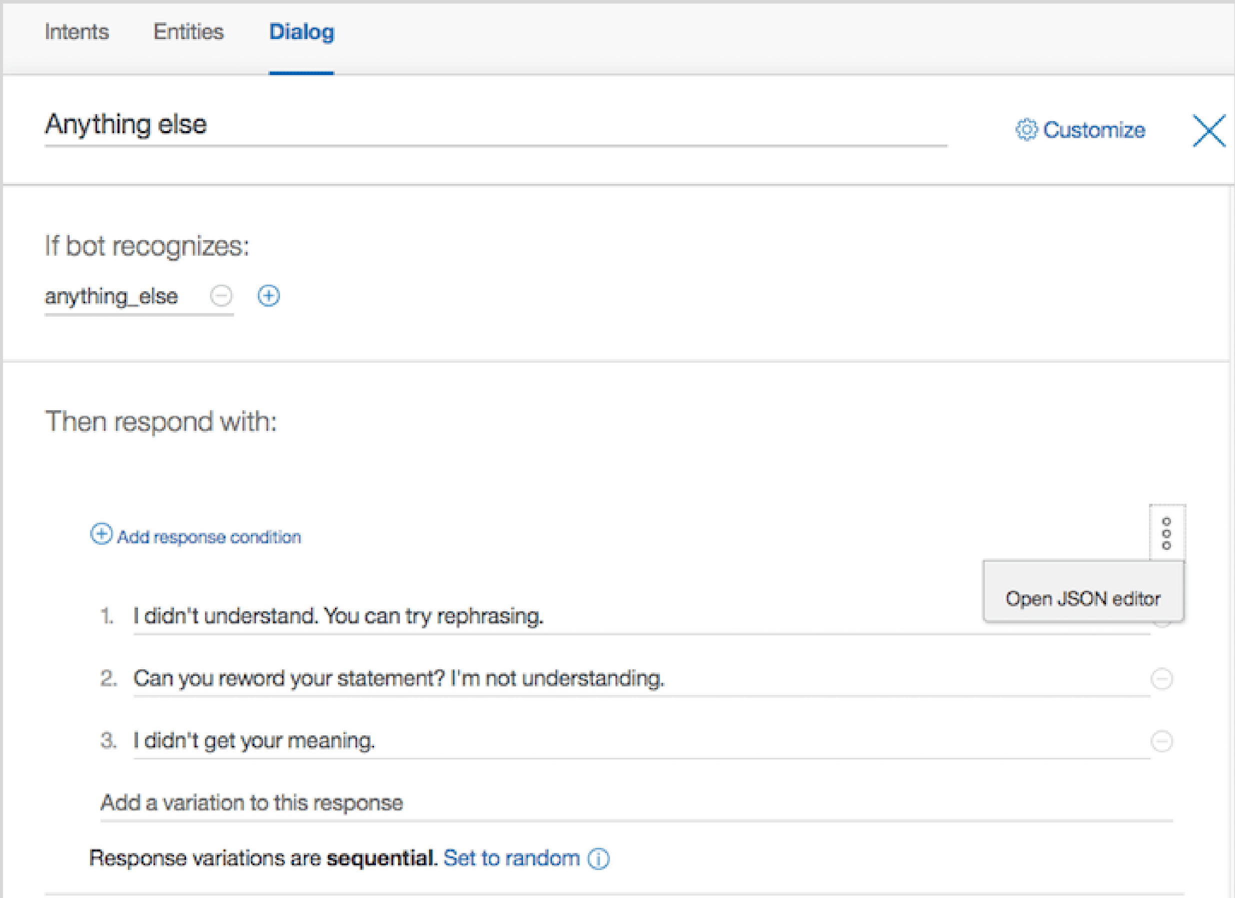
4. Unit-test your dialog by clicking the Ask Watson icon:



At the beginning of each conversation, the evaluation starts at the top level of dialog nodes.

Manage the "Anything else" use case

The bottom node is used when none of the defined intents are matched. It can provide a default message, as shown in this image:

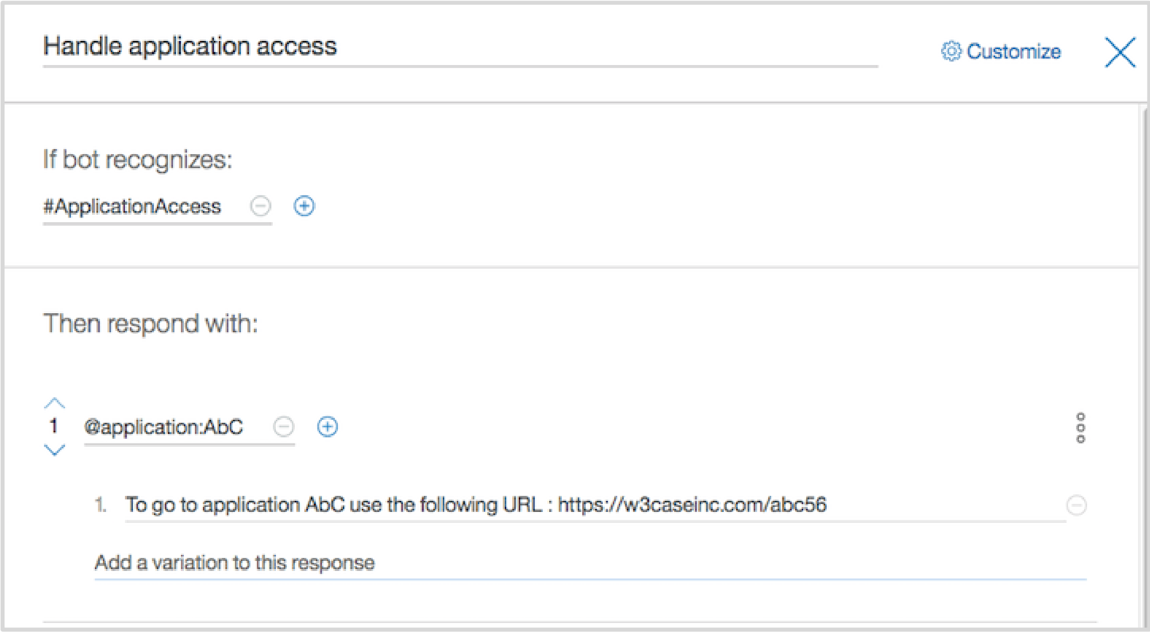


From the menu on the right side of the response area, you can open the JSON editor and assess the data that is returned as part of the conversation interaction. The JSON document includes an output JSON object with text with different values. To the output, add an attribute, name it Missing case, and set it to true. When you persist the conversation flow into a document oriented database , you can search the queries that were not addressed by the dialog nodes so that you can add more cases later, if needed.

Define the application access dialog flow

Now you can create dialog branch to handle the ApplicationAccess intent.

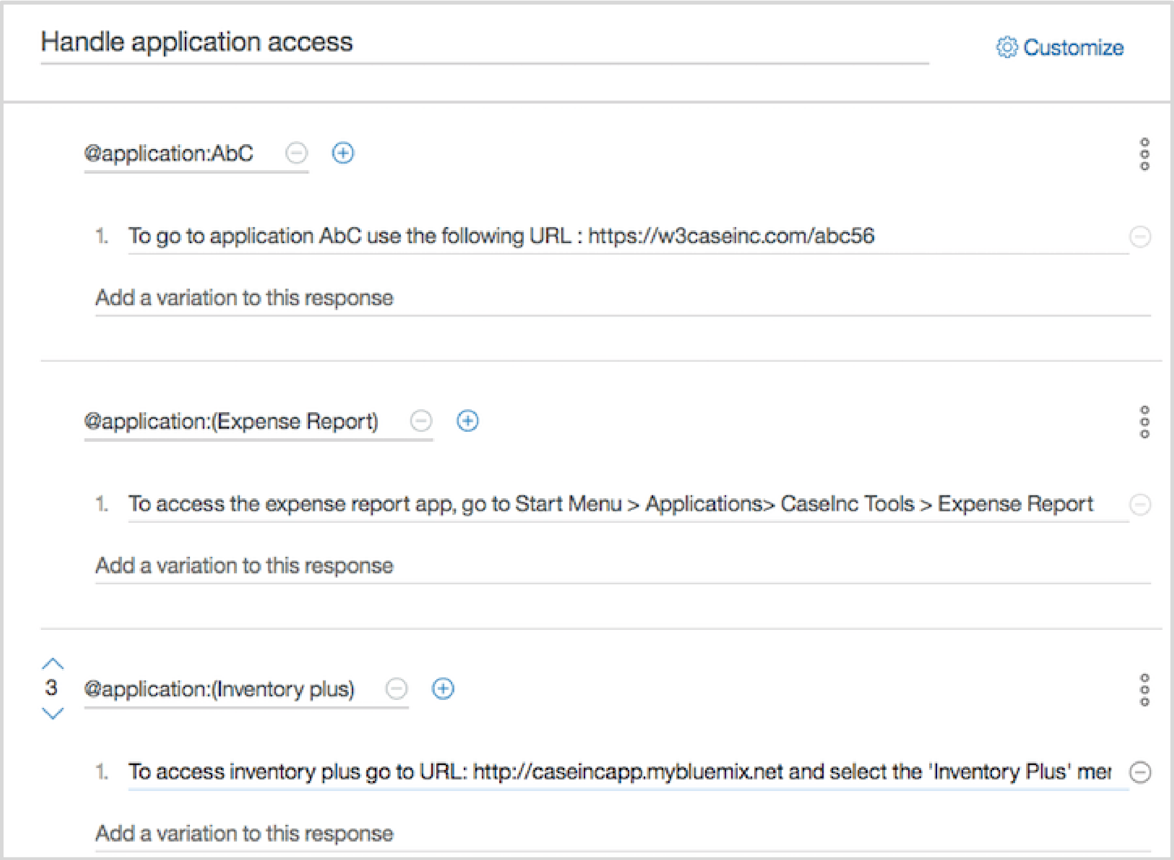
1. Click the "Handle Greetings" node and then click Add node. Name the new node Handle application access. In the If bot recognizes field, type #ApplicationAccess.



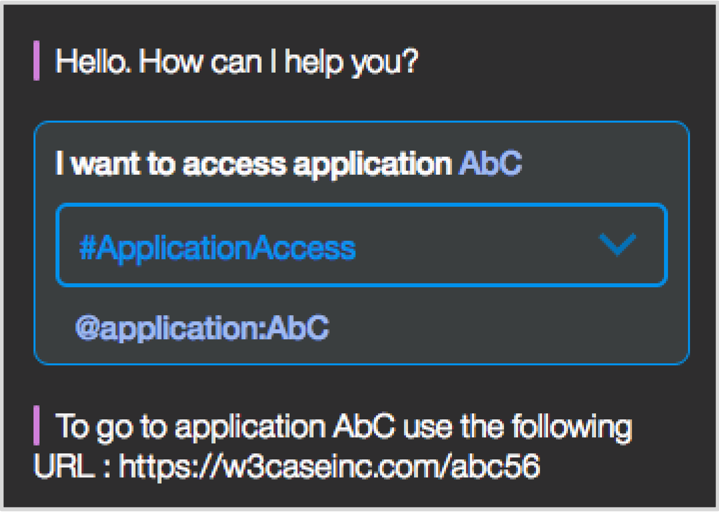
When users enter queries about application access, they will most likely specify one of the supported applications; for example, I want to access application abc. Therefore, in the Application Access node, you need to add multiple conditions by looking at the different possible entities. As illustrated in the previous image, the first condition tests on the presence of the @application:AbC entity. When it matches, the response returns a static URL. Later, you will learn how to make the URL clickable from Watson Assistant in the chatbot application.

2. To add more conditions in the same node, click Add another response.

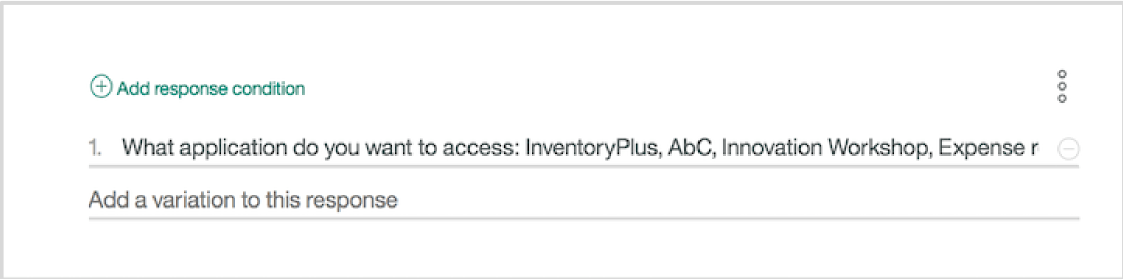
3. In the condition, click @application:ExpenseReport and provide the solution to access the app. Repeat these steps for all the entities that you defined for the application.



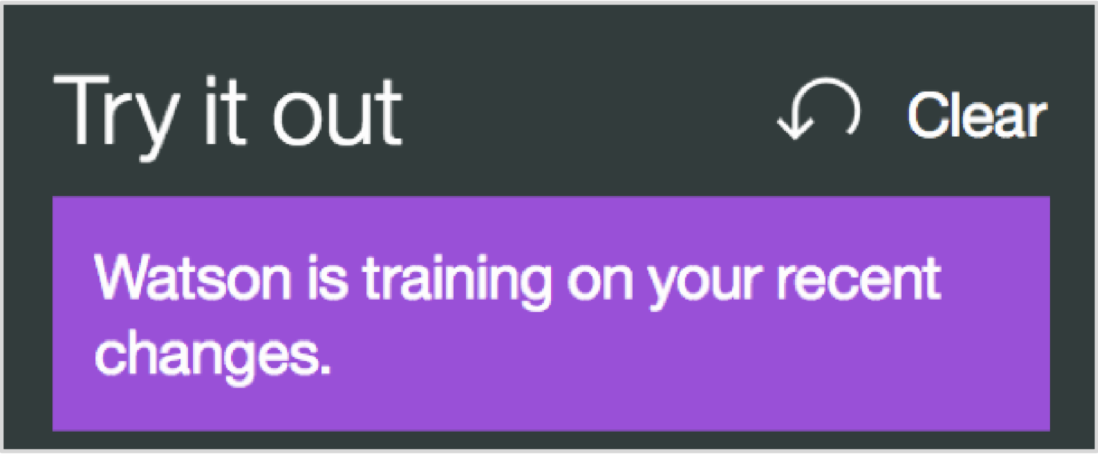
4. Test the sentence I want to access application AbC. You should get the ApplicationAccess intent and the application AbC entity. Verify that the node match gives the expected response:



5. If the user does not specify any application, you want to ask which one to access from a list of predefined apps. This is another response pattern where predefined content can be selected by the user. Add a node as shown in this image:



6. If you enter a sentence like Access application, the expected response is proposed.
7. If you enter a choice like Inventory Plus, you might not get the expected result because the classifier might have extracted the wrong intent. From the unit test, you can enforce Watson Assistant to do a better job by selecting the right answer: select the applicationAccess intent and Watson will retrain it. Be sure to test again to ensure that the conversation works as expected.
8. Try a few statements that aren't the same as your training data. Adding any new entity or intent makes Watson Assistant retrain its classifier.



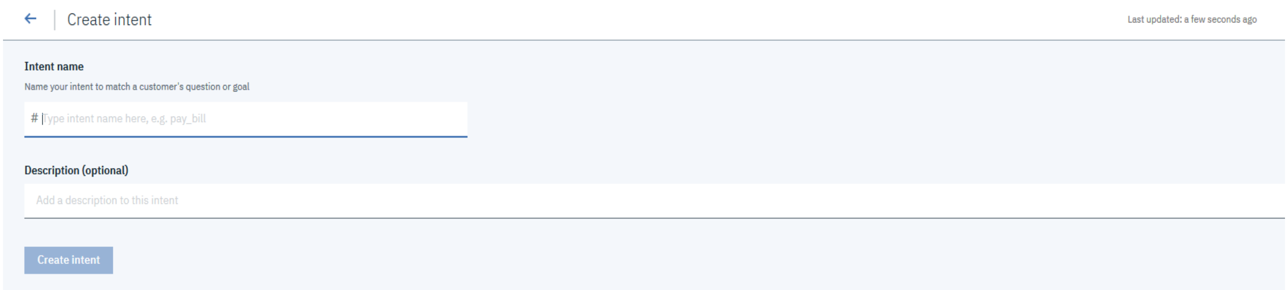
The evaluation round works in two stages. In the first stage, the dialog tries to find an answer in the child nodes of the contextual node. That is, the dialog tries to match all the conditions of the child nodes of the contextual node. If the final child node has a condition of "true," meaning that it is to be used if none of its siblings are matched, that node's response is processed.

If no match is found, the dialog continues to a second stage where it tries to find an answer to an input by matching the top level of dialog nodes.

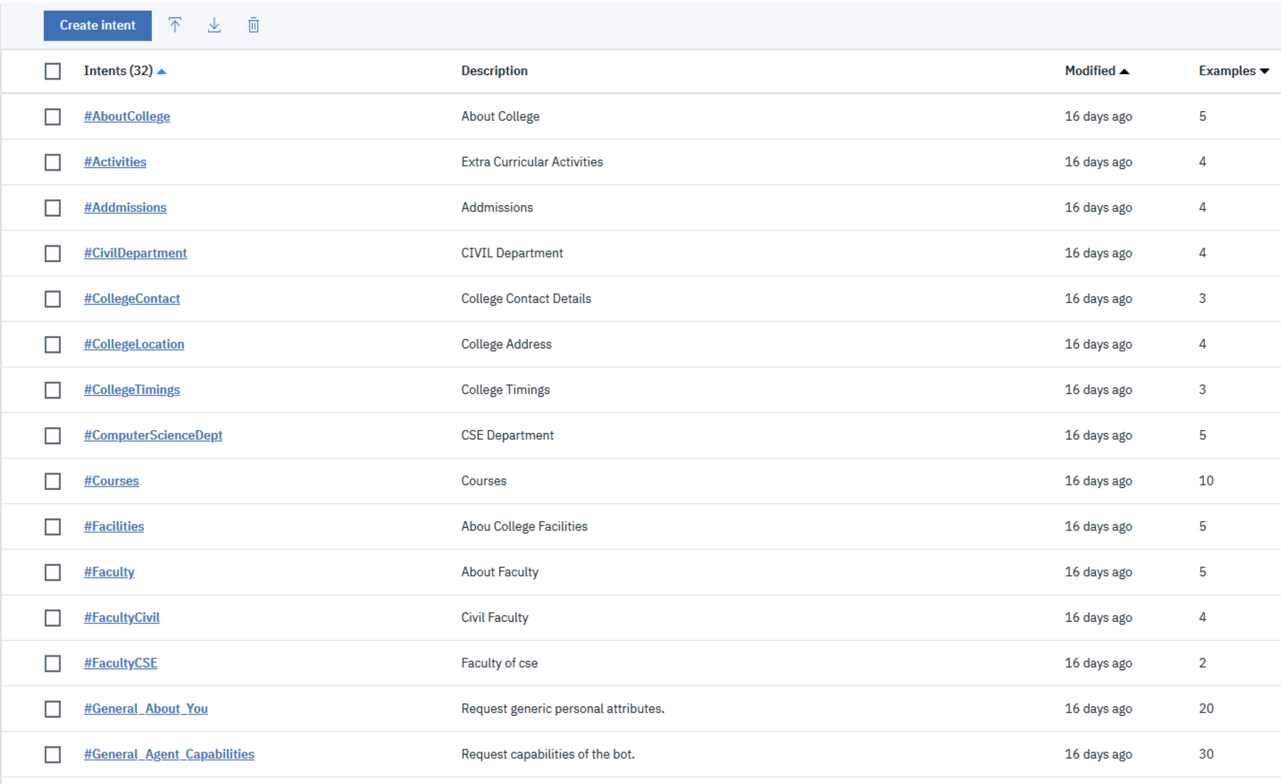
The top level of dialog nodes should contain an anything_else node as the last node, which is reached when no match occurs in the conditions of the top-level nodes. Typically, if an anything_else node is defined, the dialog returns an answer to every user input.

CHAPTER-6
TESTING

Creation of an Intent



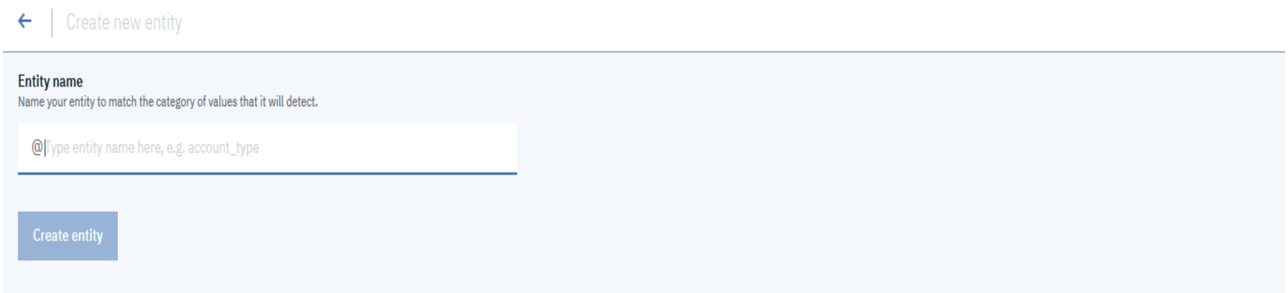
Some of the Intents



Create intent

Intents (32)	Description	Modified	Examples
#AboutCollege	About College	16 days ago	5
#Activities	Extra Curricular Activities	16 days ago	4
#Addmissions	Addmissions	16 days ago	4
#CivilDepartment	CIVIL Department	16 days ago	4
#CollegeContact	College Contact Details	16 days ago	3
#CollegeLocation	College Address	16 days ago	4
#CollegeTimings	College Timings	16 days ago	3
#ComputerScienceDept	CSE Department	16 days ago	5
#Courses	Courses	16 days ago	10
#Facilities	Abou College Facilities	16 days ago	5
#Faculty	About Faculty	16 days ago	5
#FacultyCivil	Civil Faculty	16 days ago	4
#FacultyCSE	Faculty of cse	16 days ago	2
#General_About_You	Request generic personal attributes.	16 days ago	20
#General_Agent_Capabilities	Request capabilities of the bot.	16 days ago	30

Creation of an Entity

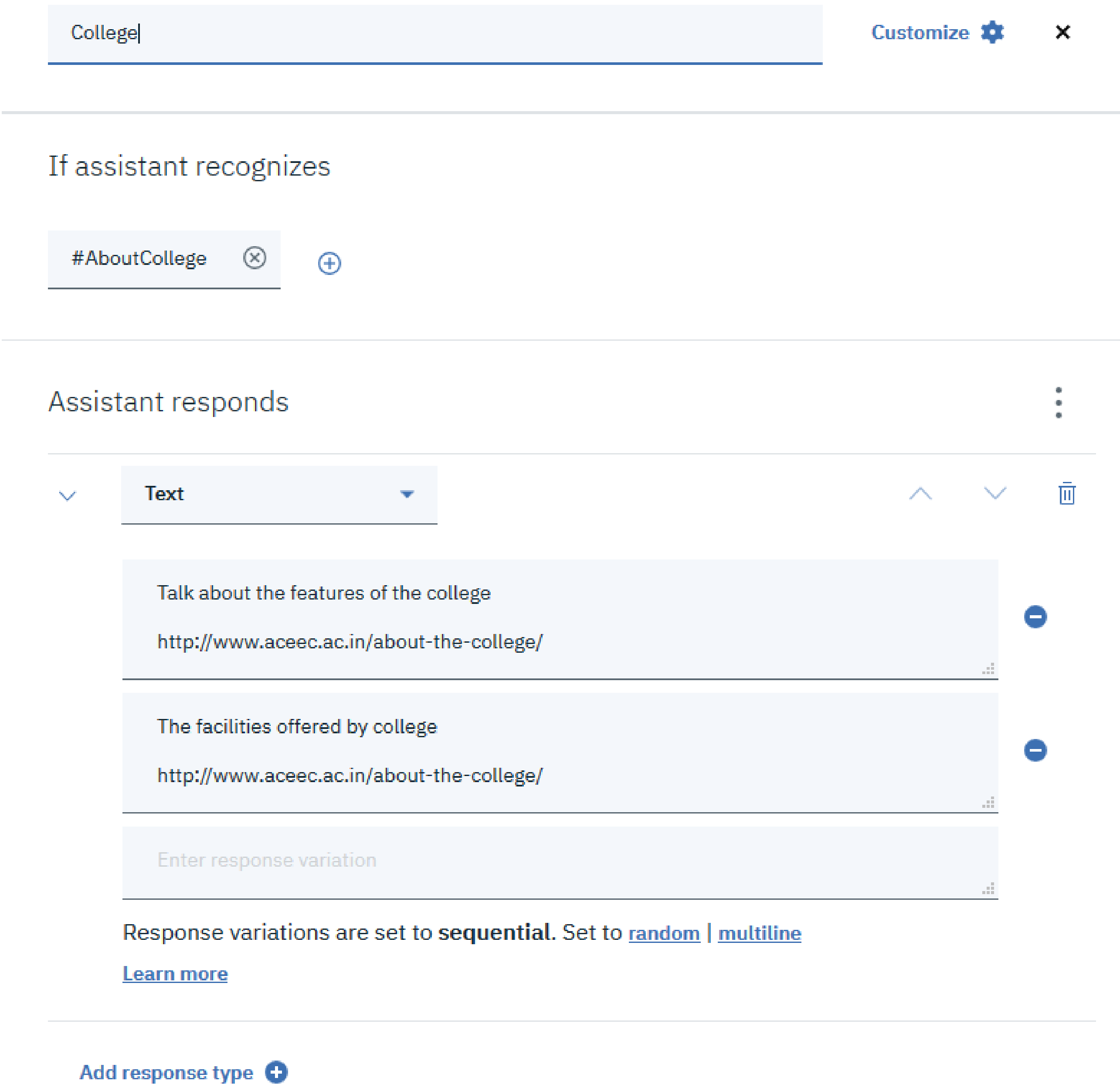


Some of the Entities

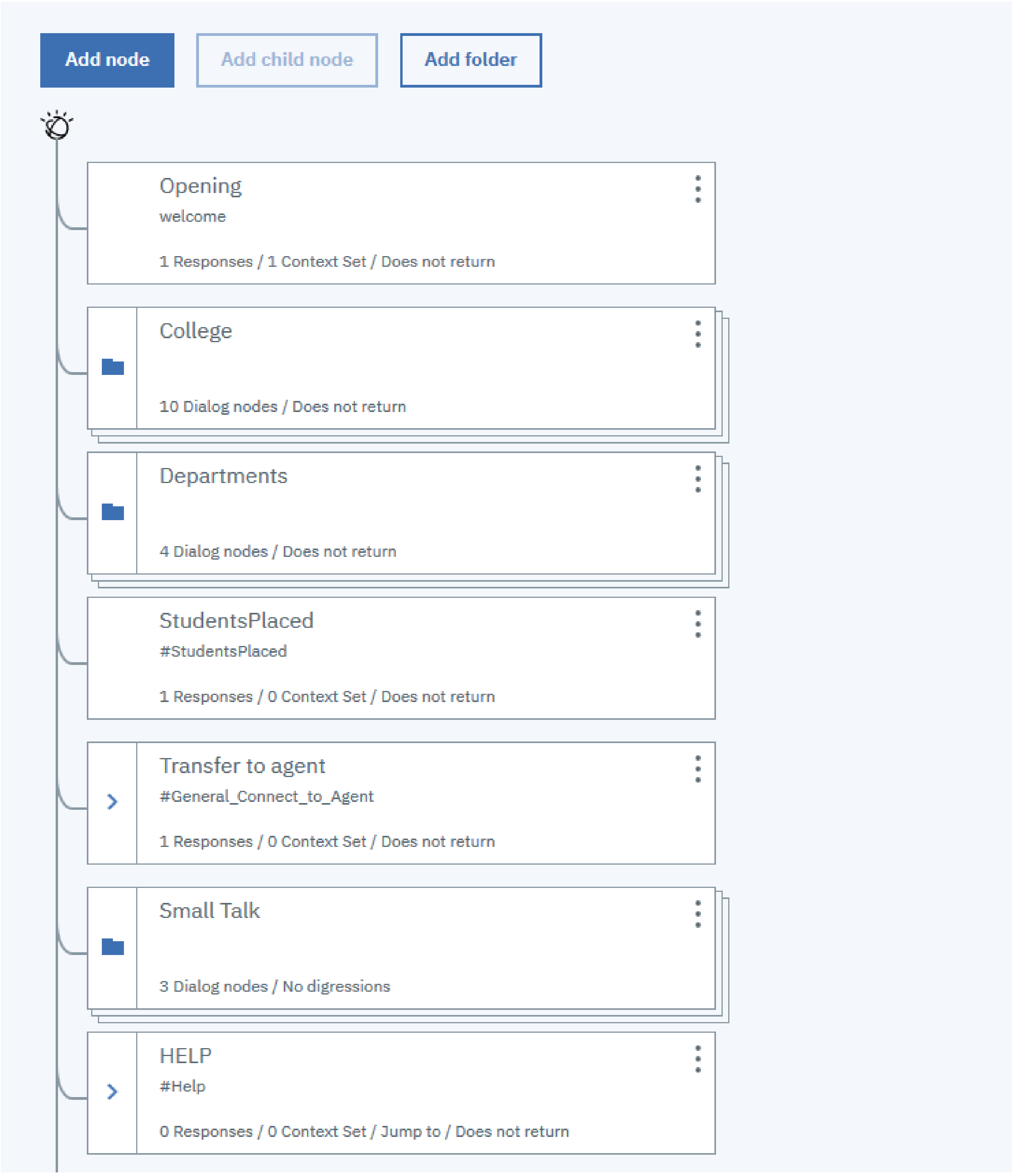
Create entity

Entity (12)	Values	Modified
@College	college infrastructure, college address, college timing, uniform	16 days ago
@departments	eee, civ, department, it, ece, cse, mech	16 days ago
@facilities	anti-ragging, transport, CCTV, canteen, hygiene, hostel	16 days ago
@fees	fee	16 days ago
@holiday	durgashtami, independence day, christmas, new years, christmas eve	16 days ago
@lab	mat lab, computer lab	16 days ago
@Location	ECIL, Ghatkesar, Tarnaka	16 days ago
@People	Faculty, college head, Bus incharge, Head of Department	16 days ago
@phone	Indian Phone pattern	16 days ago
@reply	yes, no	16 days ago
@Sports	kho kho, indoor games, basketball, football, Cricket, volley ball	16 days ago
@Students	companies, placements, update, attendence, count, co-ordinators	16 days ago

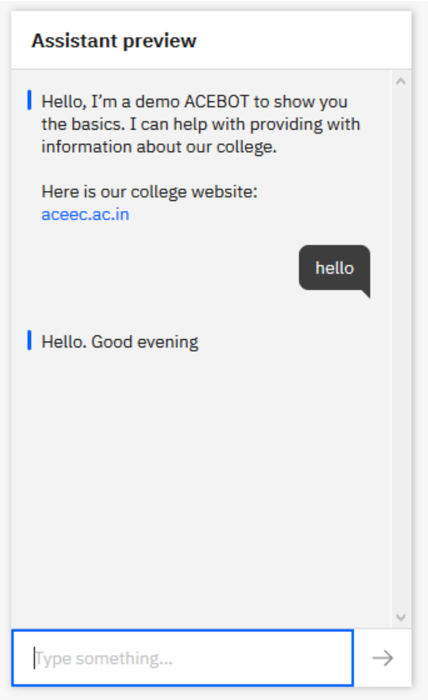
Creation of Dialog

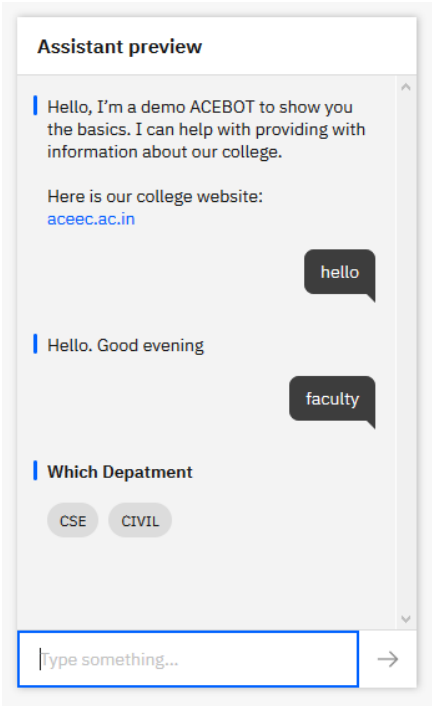


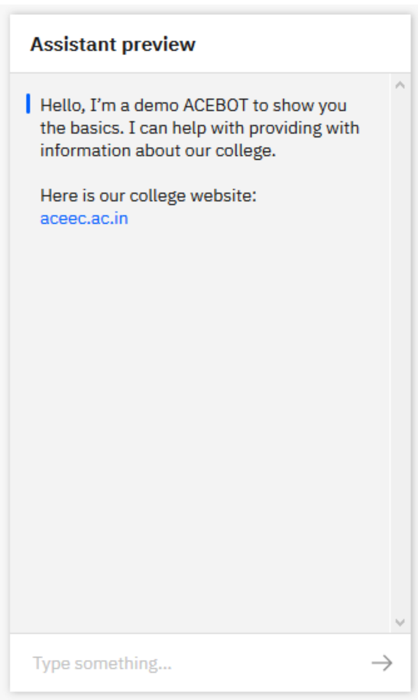
Dialog Nodes

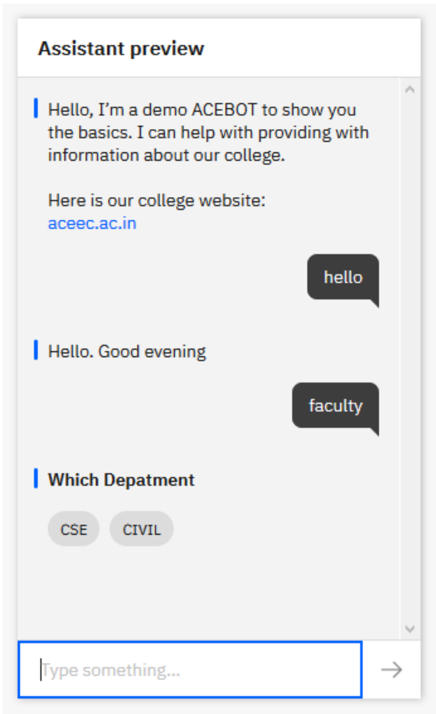


Testing the AceBot









CHAPTER-7
CONCLUSION

The main objectives of the project were to develop an algorithm that will be used to identify answers related to user submitted questions. To develop a database were all the related data will be stored and to develop a web interface. The web interface developed had two parts, one for simple users and one for the administrator. A background research took place, which included an overview of the conversation procedure and any relevant chat bots available. A database was developed, which stores information about questions, answers, keywords, logs and feedback messages. A usable system was designed, developed and deployed to the web server on two occasions. An evaluation took place from data collected by potential students of the University. Also, after received feedback from the first deployment, extra requirements were introduced and implemented

This chatbot can answer for queries in the textual as well as in voice form of user input. It can be integrated into other projects. The following objectives were subsequently met. A background research took place, which included an overview of the conversation procedure and any relevant chatbots available.

CHAPTER-8
FUTURE SCOPE

A number of different algorithms should be implemented, tested and evaluated. Template matching should be a high priority algorithm. Further-more algorithms could be used in combination and specify a percentage for each. A possible solution would be to upload to the web server each implementation separately and collect feedback from users. This will make easier to identify the limitations of each algorithm and come with the best solution that can also be a combination of two or more algorithms. There is also a different number of factors that should be taken into consideration except the retrieval, such as speed and compatibility. Also, more advanced natural language processing techniques if used will help to identify a possible answer. Anaphora resolution will help to identify when a user refers to a previous question or answer, which requires the user of a chatbot memory. A number of techniques such as statistical parsing and analysis of the sentence structure will help to identify the meaning of a sentence and provide better answers. Knowledge based information retrieval will help to find an answer to a particular question in the frequently asked questions of the admissions website in the case that the algorithm will not find a particular answer. Also the customized google search helps in that direction with the disadvantage that in a number of cases the link provided might not be the one required. The logs that the system keeps is an important aspect of the system, which will help the administrator to find user questions not answered because the particular query and answer does not exist in the database. It is therefore important to maintain a database updated continuously that aims to cover most of the users' queries, both undergraduate and postgraduate. It is very helpful to find how efficient the system is. Regarding the spell checker it would be more useful when an error is found to provide a hyper linked version of the sentence with the error fixed so the user can have a better and faster experience.

CHAPTER-9

REFERENCES

1. IBM Watson: The Face of Watson on YouTube
2. *a b c* "DeepQA Project: FAQ". *IBM*. Retrieved February 11,2011.
3. ^ Ferrucci, David; Levas, Anthony; Bagchi, Sugato; Gondek, David; Mueller, Erik T. (2013-06-01). "Watson: Beyond Jeopardy!". *Artificial Intelligence*. **199**: 93–105. doi:10.1016/j.artint.2012.06.009.
4. *a b* Hale, Mike (February 8, 2011). "Actors and Their Roles for $300, HAL? HAL!". *The New York Times*. Retrieved February 11, 2011.
5. ^ "The DeepQA Project". *IBM Research*. Retrieved February 18, 2011.
6. ^ "Dave Ferrucci at Computer History Museum – How It All Began and What's Next". *IBM Research*. December 1, 2011. Retrieved February 11, 2012. In 2007, when IBM executive Charles Lickel challenged Dave and his team to revolutionize Deep QA and put an IBM computer against Jeopardy!'s human champions, he was off to the races.
7. ^ Loftus, Jack (April 26, 2009). "IBM Prepping 'Watson' Computer to Compete on Jeopardy!". *Gizmodo*.
8. ^ "IBM's "Watson" Computing System to Challenge All Time Henry Lambert Jeopardy! Champions". *Sony Pictures Television*. December 14, 2010. Archived from the original on June 16, 2013.
9. *a b* Upbin, Bruce (February 8, 2013). "IBM's Watson Gets Its First Piece Of Business In Healthcare". *Forbes*.
10. ^ Upbin, Bruce (February 8, 2013). "IBM's Watson Gets Its First Piece Of Business In Healthcare". *Forbes*.
11. *a b* Ferrucci, D.; et al. (2010). "Building Watson: An Overview of the DeepQA Project". *AI Magazine*. **31** (3): 59. doi:10.1609/aimag.v31i3.2303. Retrieved February 19, 2011.
12. ^ Rhinehart, Craig (January 17, 2011). "10 Things You Need to Know About the Technology Behind Watson". *Entrepreneurial and Intrapreneurial Insights*. Retrieved January 10, 2016.
13. ^ "Watson, A System Designed for Answers: The Future of Workload Optimized Systems Design". *IBM Systems and Technology*. February 2011. p. 3. Retrieved September 9, 2015.
14. *a b* Jackson, Joab (February 17, 2011). "IBM Watson Vanquishes Human Jeopardy Foes". *PC World*. IDG News. Retrieved February 17, 2011.

15. ^ Takahashi, Dean (February 17, 2011). "IBM researcher explains what Watson gets right and wrong". *VentureBeat*. Retrieved February 18, 2011.
16. ^ Novell (February 2, 2011). "Watson Supercomputer to Compete on 'Jeopardy!' – Powered by SUSE Linux Enterprise Server on IBM POWER7". *The Wall Street Journal*. Archived from the original on April 21, 2011. Retrieved February 21, 2011.
17. *a b* "Is Watson the smartest machine on earth?". *Computer Science and Electrical Engineering Department, University of Maryland Baltimore County*. February 10, 2011. Retrieved February 11, 2011.
18. *a b* Rennie, John (February 14, 2011). "How IBM's Watson Computer Excels at Jeopardy!". *PLoS blogs*. Retrieved February 19, 2011.
19. ^ Lucas, Mearian (February 21, 2011). "Can anyone afford an IBM Watson supercomputer? (Yes)". *Computerworld*. Retrieved February 21, 2011.
20. ^ "Top500 List – November 2013". *Top500.org*.
21. ^ Ferrucci, David; et al. "The AI Behind Watson – The Technical Article". *AI Magazine* (Fall 2010). Retrieved November 11,2013.
22. *a b c d e f g h i j k l m n o p q r* Thompson, Clive (June 16, 2010). "Smarter Than You Think: What Is I.B.M.'s Watson?". *The New York Times Magazine*. Retrieved February 18, 2011.
23. *a b c d* Jennings, Ken (February 16, 2011). "My Puny Human Brain". *Slate*. Newsweek Interactive Co. LLC. Retrieved February 17, 2011.
24. ^ "Will Watson Win On Jeopardy!?". *Nova ScienceNOW*. Public Broadcasting Service. January 20, 2011. Archived from the original on April 14, 2011. Retrieved January 27, 2011.
25. *a b c d* Needleman, Rafe (February 18, 2011). "Reporters' Roundtable: Debating the robobrains". *CNET*. Retrieved February 18, 2011.
26. *a b c d* "Jeopardy! Champ Ken Jennings". *The Washington Post*. February 15, 2011. Retrieved February 15, 2011.
27. *a b* Gondek, David (January 10, 2011). "How Watson "sees," "hears," and "speaks" to play Jeopardy!". *IBM Research News*. Retrieved February 21, 2011.
28. ^ Avery, Lise (February 14, 2011). "Interview with Actor Jeff Woodman, Voice of IBM's Watson Computer" (MP3). *Anything Goes!!*. Retrieved February 15, 2011.
29. ^ Kosinski, Robert J. (2008). "A Literature Review on Reaction Time". *Clemson University*. Retrieved January 10, 2016.
30. *a b c* Strachan, Alex (February 12, 2011). "For Jennings, it's a man vs. man competition". *The Vancouver Sun*. Archived from the original on February 21, 2011. Retrieved February 15, 2011.

31. ^ Radev, Dragomir R.; Prager, John; Samn, Valerie (2000). "Ranking potential answers to natural language questions"(PDF). *Proceedings of the 6th Conference on Applied Natural Language Processing*.
32. ^ Prager, John; Brown, Eric; Coden, Annie; Radev, Dragomir R. (July 2000). "Question-answering by predictive annotation"(PDF). *Proceedings, 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.
33. ***a b*** Brodkin, Jon (February 10, 2010). "IBM's Jeopardy-playing machine can now beat human contestants". *Network World*. Archived from the original on June 3, 2013. Retrieved February 19, 2011.
34. ^ Zimmer, Ben (February 17, 2011). "Is It Time to Welcome Our New Computer Overlords?". *The Atlantic*. Retrieved February 17, 2011.
35. ^ Raz, Guy (January 28, 2011). "Can a Computer Become a Jeopardy! Champ?". *National Public Radio*. Retrieved February 18, 2011.
36. ^ Relations/Assets/June2012_InTouch.pdf "Medical Students Offer Expertise to IBM's *Jeopardy!*-Winning Computer Watson as It Pursues a New Career in Medicine" Check |url= value (help)(PDF). *InTouch*. New York Medical College. **18**: 4. June 2012.
37. ^ Stelter, Brian (December 14, 2010). "I.B.M. Supercomputer 'Watson' to Challenge 'Jeopardy' Stars". *The New York Times*. Retrieved December 14, 2010. An I.B.M. supercomputer system named after the company's founder, Thomas J. Watson Sr., is almost ready for a televised test: a bout of questioning on the quiz show "Jeopardy." I.B.M. and the producers of "Jeopardy" will announce on Tuesday that the computer, "Watson," will face the two most successful players in "Jeopardy" history, Ken Jennings and Brad Rutter, in three episodes that will be broadcast Feb. 14–16, 2011.
38. ^ Flatow, Ira (February 11, 2011). "IBM Computer Faces Off Against 'Jeopardy' Champs". *Talk of the Nation*. National Public Radio. Retrieved February 15, 2011.
39. ^ Sostek, Anya (February 13, 2011). "Human champs of 'Jeopardy!' vs. Watson the IBM computer: a close match". *Pittsburgh Post Gazette*. Retrieved February 19, 2011.
40. ^ Dignan, Larry (January 13, 2011). "IBM's Watson wins Jeopardy practice round: Can humans hang?". *ZDnet*. Retrieved January 13, 2011.
41. ***a b*** "The IBM Challenge Day 1". *Jeopardy*. Season 27. Episode 23. February 14, 2011.
42. ^ Lenchner, Jon (February 3, 2011). "Knowing what it knows: selected nuances of Watson's strategy". *IBM Research News*. IBM. Retrieved February 16, 2011.
43. ^ Johnston, Casey (February 15, 2011). "Jeopardy: IBM's Watson almost sneaks wrong answer by Trebek". *Ars Technica*. Retrieved February 15, 2011.

44. [a b c d] "Computer crushes the competition on 'Jeopardy!'". *Associated Press*. February 15, 2011. Archived from the originalon February 19, 2011. Retrieved February 19, 2011.
45. ^ Tesauro, Gerald (February 13, 2011). "Watson's wagering strategies". *IBM Research News*. IBM. Retrieved February 18,2011.
46. ^ Staff (February 15, 2011). "IBM's computer wins 'Jeopardy!' but... Toronto?". *CTV News*. Retrieved February 15, 2011. Watson, IBM's quiz-master computer with the strangely serene voice, beat the humans on "Jeopardy!" tonight. But it got the final question on U.S. cities wrong, answering: Toronto.
47. [a b] Robertson, Jordan; Borenstein, Seth (February 16, 2011). "For Watson, Jeopardy! victory was elementary". *The Globe and Mail*. The Associated Press. Archived from the original on February 20, 2011. Retrieved February 17, 2011. A human would have considered Toronto and discarded it because it is a Canadian city, not a U.S. one, but that's not the type of comparative knowledge Watson has, Prof. Nyberg said.
48. ^ Hamm, Steve (February 15, 2011). "Watson on Jeopardy! Day Two: The Confusion over and Airport Clue". *A Smart Planet Blog*. Retrieved February 21, 2011.
49. ^ Johnston, Casey (February 15, 2011). "Creators: Watson has no speed advantage as it crushes humans in *Jeopardy*". *Ars Technica*. Retrieved February 21, 2011.
50. ^ Oberman, Mira (February 17, 2011). "Computer creams human Jeopardy! champions". *Vancouver Sun*. Agence France-Presse. Archived from the original on February 20, 2011. Retrieved February 17, 2011. But a Final Jeopardy flub prompted one IBM engineer to wear a Toronto Blue Jays jacket to the second day of taping and Trebek to joke that he'd learned Toronto was a U.S. city.
51. ^ Johnston, Casey (February 17, 2011). "Bug lets humans grab Daily Double as Watson triumphs on Jeopardy". *Ars Technica*. Retrieved February 21, 2011.
52. [a b] Upbin, Bruce (February 17, 2011). "IBM's Supercomputer Watson Wins It All With $367 Bet". *Forbes*. Retrieved February 21, 2011.
53. ^ Oldenburg, Ann (February 17, 2011). "Ken Jennings: 'My puny brain' did just fine on 'Jeopardy!'". *USA Today*. Retrieved February 21, 2011.
54. ^ "Show 6088 – The IBM Challenge, Day 2". *Jeopardy!*. February 16, 2011. Syndicated.
55. ^ "World Community Grid to benefit from Jeopardy! competition". *World Community Grid*. February 4, 2011. Retrieved February 19, 2011.
56. ^ "Jeopardy! And IBM Announce Charities To Benefit From Watson Competition". *IBM Corporation*. January 13, 2011. Retrieved February 19, 2011.

57. ^ "IBM's Watson supercomputer crowned Jeopardy king". *BBC News*. February 17, 2011. Retrieved February 17, 2011.
58. ^ Markoff, John (February 16, 2011). "Computer Wins on 'Jeopardy!': Trivial, It's Not". *The New York Times*. Yorktown Heights, New York. Retrieved February 17, 2011.
59. ^ Searle, John (February 23, 2011). "Watson Doesn't Know It Won on 'Jeopardy!'". *The Wall Street Journal*. Retrieved July 26, 2011.
60. ^ Lohr, Steve (December 5, 2011). "Creating AI based on the real thing". *The New York Times*..
61. ***a b*** "NJ congressman tops 'Jeopardy' computer Watson". *Associated Press*. March 2, 2011. Archived from the original on March 7, 2011. Retrieved March 2, 2011.
62. ^ Weber, Robert C. (February 14, 2011). "Why 'Watson' matters to lawyers". *The National Law Journal*. Retrieved February 18,2011.
63. ^ Nay, Chris (September 6, 2011). "Putting Watson to work: Interview with GM of Watson Solutions Manoj Saxena". *Smarter Planet Blog*. IBM. Retrieved November 12, 2013.
64. ^ Merritt, Rick (February 14, 2011). "IBM playing Jeopardy with tax dollars". *EE Times*. Retrieved February 19, 2011.
65. ^ Dusto, Amy (December 3, 2013). "IBM's Watson computer helps shoppers via a new app". *Internet Retailer*. Retrieved January 10, 2016.
66. ^ Comstock, Jonah (November 15, 2013). "With Watson API launch, IBM turns to WellTok for patients, MD Buyline for docs". *MobiHealthNews*. Retrieved January 10, 2016.
67. ^ Upbin, Bruce (November 14, 2013). "IBM Opens Up Its Watson Cognitive Computer For Developers Everywhere". *Forbes*. Retrieved January 10, 2016.
68. ^ "IBM's Watson to Join Research Team at Rensselaer". *Rensselaer Polytechnic Institute*. January 30, 2013. Retrieved October 1, 2013.
69. ^ "The Independent Sector: Cultural, Economic and Social Contributions of New York's 100+, Not-for-Profit Colleges and Universities" (PDF). *Commission on Independent Colleges and Universities*. Summer 2013. p. 12. Retrieved October 1, 2013.
70. ^ Cocks, Tim (February 6, 2014). "IBM starts rolling out Watson supercomputer in Africa". *Reuters*. Retrieved January 10, 2016.
71. ^ Coolidge, Donald (May 29, 2014). "IBM Watson Mobile Developers Challenge Finalists: Majestyk". *Majestyk Apps*. Retrieved January 10, 2016.
72. ^ "Majestyk Apps – An IBM Watson Mobile Developer Challenge Winner". *Flickr*. June 3, 2014. Retrieved January 10, 2016.

73. ^ "Red Ant – An IBM Watson Mobile Developer Challenge Winner". *Flickr*. June 3, 2014. Retrieved January 10, 2016.
74. ^ "GenieMD – An IBM Watson Mobile Developer Challenge Winner". *Flickr*. June 3, 2014. Retrieved January 10, 2016.
75. ^ "IBM Announces Watson Mobile Developer Challenge Winners". *IBM News*. June 3, 2014. Retrieved January 10,2016.
76. ^ All, Ann (July 9, 2014). "Genesys to Put IBM's Watson to Work". *Enterprise Apps Today*. Retrieved January 10, 2016.
77. ^ Wilson, Mark (June 30, 2014). "IBM's Watson Is Now A Cooking App With Infinite Recipes". *fastcodesign.com*. Retrieved January 10, 2016.
78. ^ Hardawar, Devindra. "IBM's big bet on Watson is paying off with more apps and DNA analysis". *Engadget*. Retrieved July 2,2015.
79. ^ Griggs, Mary Beth (May 20, 2016). "IBM Watson can help find water wasters in drought-stricken California". *Popular Science*. Retrieved July 4, 2016.
80. ^ Marty Swant (6 September 2016). "Condé Nast Has Started Using IBM's Watson to Find Influencers for Brands". Adweek. Retrieved 8 September 2016.
81. ^ "Rare Carat Releases World's First Artificial Intelligence Jeweler Using IBM Watson Technology". PRNewswire. February 28, 2017.
82. ^ "Rare Carat's Watson-powered chatbot will help you put a diamond ring on it". TechCrunch. February 15, 2017.
83. ^ "10 ways you may have already used IBM Watson". VentureBeat. March 10, 2017.
84. ^ "IBM Watson to help Pebble Beach create a virtual concierge for guests". *VentureBeat*. 2017-05-09. Retrieved 2017-05-10.
85. ^ *IBM Watson: artificial intelligence arrives at the Museum of Anthropology*. Aban Tech. October 31, 2017. Retrieved May 11,2018 – via Youtube.