
Using Watson for Constructing Cognitive Assistants

Ashok Goel

Tory Anderson

Jordan Belknap

Brian Creeden

William Hancock

Mithun Kumble

Shanu Salunke

Bradley Sheneman

Abhinaya Shetty

Bryan Wiltgen

Design & Intelligence Laboratory, School of Interactive Computing, Georgia Institute of Technology, Atlanta, Georgia 30308, USA.

GOEL@CC.GATECH.EDU

TORYS.ANDERSON@GMAIL.COM

JBELKNAP01@GMAIL.COM

BRIAN.CREEDEN@GMAIL.COM

WHANCOCK@GATECH.EDU

MKUMBLE@GATECH.EDU

SHANU@GATECH.EDU

BRADSHENEMAN@GMAIL.COM

SHETTY_ABHINAYA@GATECH.EDU

BRYAN.WILTGEN@GATECH.EDU

Abstract

We describe an experiment in using IBM's Watson tool to construct cognitive assistants in a Georgia Tech Spring 2015 class on computational creativity. The project-based class used Watson to support biologically inspired design, a paradigm that uses biological systems as analogues for inventing technological systems. The students worked in small teams and developed semester-long projects that used Watson to build cognitive assistants for conducting research in support of biologically inspired design. In this paper, we describe this experiment in using Watson, present two of the projects in detail, draw best practices for using Watson, and characterize and critique the Watson tool for constructing cognitive assistants.

1. Background and Motivations

In 2011 IBM's Watson system surprised and delighted much of the world by winning the game show *Jeopardy!* against human opponents. Since then IBM has claimed Watson to be a "cognitive system" that has ushered in an era of "cognitive computing" (Kelly & Hamm 2013). It has published several articles on Watson (e.g., Brown et al. 2013; Ferruci et al. 2010; Kalyanpur & Murdock 2015), including a special issue of the IBM Journal of Research and Development (e.g., Ferruci 2012). In addition, it has incrementally provided access to various versions of Watson, starting with the skeletal Watson Engagement Advisor (n.d.) that is freely available to most faculty and students for educational purposes, and recently, the more robust Bluemix Platform (n.d.) that provides access to various components of Watson as services and is available to most anyone for most any purpose but typically at a charge. Given the large-scale IBM publicity and investment, Watson has come to exemplify cognitive systems in much of the popular media.

Yet, at least outside IBM, our understanding of Watson remains modest. The articles published by IBM often go into enormous algorithmic and representational detail on selected technical issues, but typically leave out both critical elements of the representations and algorithms as well

as the big picture view of how all the components work together as a “cognitive system”, presumably for proprietary reasons. As a result, a small but growing body of academic research has started examining what exactly does Watson do, how does it work, how might it be used, and whether its claimed behaviors can be replicated (e.g., Goel et al. 2015a; Hendler 2013, 2014; Wollowski 2014; Zadrozny et al. 2015a; Zadrozny, de Pavia & Moss 2015).

The core question in this work is what is Watson? At the highest level of description, here are some *possible* answers to this question based on an initial reading of articles published by IBM and various metaphors in the AI literature on cognitive systems:

(i) *Watson is a cognitive system*: The notion of a cognitive system can be traced back to the earliest days of AI (McCarthy et al. 2006) even if the term was not used at the time. While a precise characterization of a cognitive system is hard to find, broadly speaking (Goel & Davies 2011; Langley 2011, 2012), a cognitive system is inspired by cognitive phenomena (what mind does), if not also cognitive processes (how mind works), as well as provides insights into them.

(ii) *Watson is a cognitive architecture*: The notion of a cognitive architecture is central to much research on cognitive systems (Langley, Laird & Rogers 2009). If the original *Jeopardy!* playing Watson is a cognitive system as IBM claims, then we might think of Watson Engagement Advisor as a cognitive architecture equivalent to other cognitive architectures such as Soar (Laird 2012).

(iii) *Watson is a knowledge system*: The notion of knowledge-based reasoning is also central to cognitive systems research (Chandrasekaran 1990; Chandrasekaran & Goel 1988) and again goes to the earliest days of AI. Many practical cognitive systems entail the construction of one or more knowledge bases. There are longstanding and ongoing attempts to construct large-scale knowledge bases such as CYC (Lenat 1995). Given Watson’s performance on *Jeopardy!*, we might think of Watson as a knowledge-based system that uses its knowledge about the world to answer a large variety of questions correctly.

(iv) *Watson is a knowledge system shell*: The notion of a knowledge-based shell goes at least as far back in AI as Emycin (van Melle et al. 1984). Emycin (for Empty Mycin) had the architecture, representations and algorithms of the original Mycin system for diagnosing a class of bacterial diseases (Shortliffe 1976) but without any of Mycin’s knowledge contents. REM is a more recent instance of a knowledge-based shell for instantiating meta-reasoning in intelligent agents (Murdock & Goel 2008). We might think of Watson Engagement Advisor as a knowledge-based shell for question answering.

(v) *Watson is a programming language for cognitive computing*: Programming languages of course permeate most computing. However, a good characterization of cognitive computing is hard to find. One characterization of cognitive computing is the design, development and deployment of cognitive assistants, that is, interactive intelligent agents that assist humans in high-level cognitive tasks. We might then think of Watson Engagement Advisor as a new AI programming language for constructing cognitive assistants that rely on question answering.

(vi) *Watson is a software toolset for cognitive computing*: The notion of a toolset is another common metaphor in much of computing. Again we may characterize cognitive computing as construction of cognitive assistants. We might then think of Watson Engagement Advisor or the Bluemix Platform as a software toolset for constructing cognitive assistants relying on question answering.

In this paper, we describe an experiment in using Watson in the Georgia Tech Spring 2015 CS4803/8803 class on Computational Creativity. Given that the Computational Creativity class exclusively used the Watson Engagement Advisor, unless otherwise noted, by Watson we mean

the Watson Engagement Advisor in the rest of this paper. The project-based class started with the goal of using Watson to support biologically inspired design, a design paradigm that uses biological systems as analogues for inventing technological systems. Although all class projects used Watson to develop cognitive assistants, over time they diverged both from one another and, to various degrees, from the original goal. In this paper, we describe this experiment in using Watson, present two of the projects in detail, and draw best practices for using Watson. We characterize and critique Watson as a tool for constructing cognitive assistants, and analyze its use in education about cognitive systems. Throughout this paper, we will refer to the cognitive assistants built using Watson either by their names or generically as Watson+.

2. The Computational Creativity Class

The Georgia Tech CS4803/8803 class on Computational Creativity in Spring 2015 consisted of 24 students, including 21 graduate students and 3 undergraduate senior students. 18 of the 21 graduate students and all 3 undergraduate students were majoring in computer science. While most students in the class had taken at least one prior course in AI, few students had any background in cognitive science, computational creativity, or biologically inspired design.

Most readings and discussions in the class focused on six themes: (1) *Design Thinking* (thinking about ill-structured, open-ended problems with ill-defined goals and evaluation criteria); (2) *Systems Thinking* (thinking about complex phenomena consisting of multiple interacting components and causal processes). (3) *Analogical Thinking* (thinking about novel situations in terms of similar, familiar situations); (4) *Abductive Thinking* (thinking about potential explanations for a set of data); (5) *Visual Thinking* (thinking about images and in images); and (6) *Meta-Thinking* (thinking about one's own knowledge and thinking). The two major creative domains of discussion in the class were (i) Engineering design and invention, and (ii) Scientific modeling and discovery. The choice of these themes and domains reflects cognitive systems research on computational creativity in the Design & Intelligence Laboratory (<http://dilab.gatech.edu/>). The class website provides more details about the course (http://www.cc.gatech.edu/classes/AY2015/cs8803_spring).

3. The Class Projects on Biologically Inspired Design

The class projects focused on biologically inspired design (also known as biomimicry or biomimetics). The paradigm of biologically inspired design views nature as a vast library of robust, efficient, and multifunctional designs, and espouses the use of nature as an analogue for designing technological systems as well as a standard for evaluating technological designs (Benyus 1997; Vincent & Mann 2002). Over the last generation the paradigm has transformed into a movement in engineering and systems design, pulled in part by the growing need for environmentally sustainable development and pushed partly by the perennial desire for creativity and innovation in design. Thus, the study of biologically inspired design is attracting a rapidly growing literature, including publications, patents, and computational techniques and tools (Goel, McAdams & Stone 2014).

By definition, biologically inspired design entails cross-domain analogical transfer from biology to design. In the current state of the art, there are few computational systems that actually perform biologically inspired design. Further, the cross-domain nature of the analogies means that

there are few experts in biologically inspired design: most designers are novices at biology and most biologists are naïve about design. Thus, most designers have neither much knowledge of biological systems nor the ability to easily acquire this knowledge. This presents a major challenge for developing computational techniques and tools for supporting biologically inspired design. In practice, designers typically retrieve biological analogues from the web rather than from their long-term memory. However, this process of *interactive analogical retrieval* itself is challenging because most designers have difficulty in forming good queries that may result in locating biological analogies on the web, finding biological analogues relevant to their problem, recognizing the usefulness of a biological analogue to the problem, and understanding the biological analogue well enough to perform analogical mapping and transfer (Vattam & Goel 2012). Thus, in the current state of the art, the focus is on developing computational tools that can support designers in asking questions, exploring the domain of biology, and finding biological analogues relevant to their design problems. This is why Watson, a question-answering system, *prima facie* appeared to be a potentially useful tool to use for constructing cognitive assistants in support of biologically inspired design.

The process followed by the six teams for using Watson to support biologically inspired design consisted of two phases: an initial learning phase and a later open-ended research phase. The initial learning phase proceeded roughly as follows. (1) The 6 teams selected a *case study* of biologically inspired design of their choice from a digital library called DSL (Goel et al. 2015). For each team, the selected case study became the *use case* in biologically inspired design. Each team was asked to use Watson to build a cognitive assistant in support of its use case. (2) The teams started seeding Watson with articles selected from a collection of around 200 biology articles derived from Biologue. Biologue is an interactive system for retrieving biology articles relevant to a design query (Vattam & Goel 2013). It supports the case retrieval phase of textual case-based reasoning in which the cases are available in the form of textual documents (Weber, Ashley & Bruninghaus 2005). (3) The teams individually generated about 600 questions relevant to their use cases. (4) The teams identified the best answers in their selected biology articles for the 600 questions. (5) The teams trained Watson on the 600 question-answer pairs. (6) The 6 teams evaluated Watson for answering design questions related to their respective use cases.

In the latter open-ended phase each of the 6 teams was free to conduct research as it wished. This led to several additional steps. (7) The 6 teams together grew the number of documents in Watson's knowledgebase to about 600 and the number of questions from 600 to about 1,200. (8) All 6 teams developed custom-made software for their projects. (9) All 6 teams evaluated aspects of their projects. (10) All 6 teams wrote reflective design reports and prepared short videos describing their projects (<https://www.youtube.com/playlist?list=PL44rHkM-p0hu5H7oS3OXYgK9qDkVajyqY>).

The 6 projects turned out to be quite diverse. While all 6 projects started with analyzing use cases in biologically inspired design, one (Watsabi) ended up with agriculture as the task domain. Within biologically inspired design, 2 projects targeted specific domains (resilient materials for BioMaterials and built architecture for SustArch), while the other 3 were domain independent. While one project (Erasmus) integrated Watson with another cognitive tool called Alchemy, another project (Twenty Questions) was inspired by a game. While all 6 projects supported human-computer interaction, 2 of the projects (Watsabi and SustArch) also explicitly supported human-human interaction. While all 6 projects were Internet-enabled applications, 2 (BioMaterials and SustArch) were mobile apps running on the Android smart phone system.

4. The Erasmus Project

The Erasmus system supports exploration of biology in preparation for biologically inspired design. It provides access to snippets from biology articles relevant to a design-related question. It provides designers with a quickly digestible visual map of the biological concepts relevant to a query and the degree to which they are relevant. It also expands the scope of the query to display the concepts contained in a broader set of documents, enabling designers to identify interesting avenues for further inquiry.

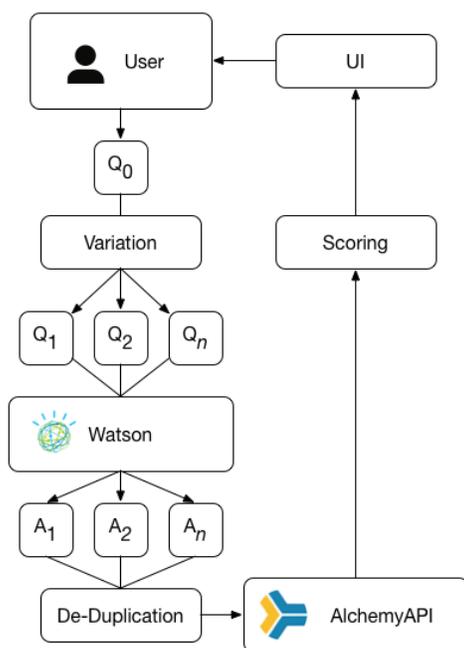


Figure 1. The Erasmus architecture.

4.1 Architecture

Figure 1 illustrates Erasmus’ architecture implemented within a browser-based, server-client application. Watson, at the core of the architecture, provides a natural language front-end to a powerful information retrieval service. AlchemyAPI, also an IBM technology, performs information extraction on the relevant text produced by Watson, identifying concepts that encapsulate the topics addressed in each document (Turian 2013). Erasmus integrates multiple components: a front end that accepts a question from the user, a variant generator that creates multiple variants of the question; Watson—which uses the query to perform a lookup on its document corpus; AlchemyAPI—which extracts the major concepts from each responsive document; a term frequency inverse

document frequency (TFIDF) filter to eliminate highly similar spans of text; and a comparison system that scores conceptual relevancy between question variants and answers.

4.2 Process

A query begins when a user enters their question into a text field in the Erasmus front-end. The string is then passed to a Python script hosted on a private server, which uses the natural language toolkit (NLTK) to generate ten grammatically acceptable variants on the user’s query. The script separates the word tokens by their grammatical function (e.g., subject, verb, object) and then re-fits existing tokens into new configurations using grammatical templates. We expanded the scope of the resulting concept space by generating small variations on verbiage and word order, which enables Watson to include documents from its corpus that it might have considered irrelevant. This allows the user to perceive new concepts for further exploration.

Once the NLTK script has generated ten variants on the user’s question, it passes this set to our Watson instance, which is hosted remotely by IBM. Each request is transmitted in a separate POST call and returned in JSON format. For each of the ten questions posed to Watson, Erasmus

Erasmus

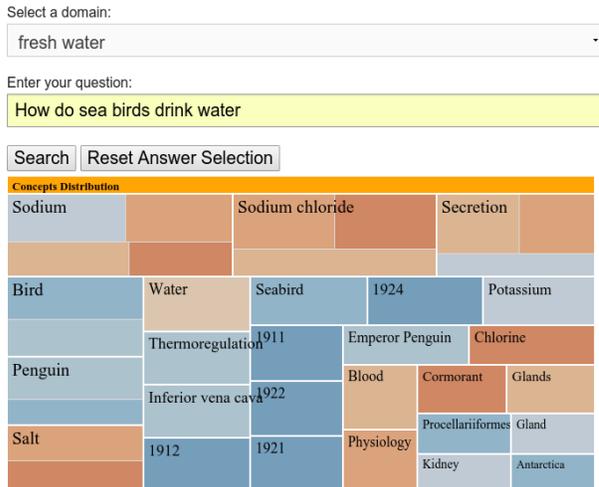


Figure 2. The concept neighborhood treemap: Erasmus’ initial response to a question.

extracts the top ten answer candidates, ordered by Watson’s confidence that they are the correct response to the question posed. Because Erasmus passes multiple requests to Watson using slight variations on query language, Watson’s internal deduplication methods are thwarted. We clean the resulting answer set of duplicates by converting each answer string into TFIDF vectors, then calculate the cosine similarity among Watson’s top ten answer candidates. Once Erasmus has matched its ten question variants with Watson’s ten best answer candidates for each, we pass each set to AlchemyAPI for concept extraction. AlchemyAPI returns a set of concepts for each question and answer candidate, along with a relevance score between 0 and 1, where a score of 1 indicates maximum relevancy to the

document.

With this information, Erasmus scores each answer for its conceptual overlap with the question. Erasmus calculates this score by determining the degree of similarity between a question’s concept set and the concept set of its answers. Common concepts are then weighted by each concept’s relevance score and aggregated into a single value that represents the degree of similarity in the question-answer pair. Erasmus then ranks answers by their score, eliminating all but the top five responses. This limits the scope of the concepts presented to the user, but makes their relationship to each of the questions more easily digestible in a treemap.

Erasmus’s visualization of the resulting concepts and their relationship to each question attempts to honor Shneiderman’s (1996) information visualization task taxonomy: overview, zoom, filter, details-on-demand, relate, history, extract. In the initial display, users are shown all concepts generated by their query. The display area occupied by each concept is determined by that concept’s share of the aggregate relevance score. Within each concept, colored blocks represent the share of that concept accounted for by each of the five answers. Users can click to expand each concept, which then shifts the display to represent only those answer blocks relevant to the selected concept alongside the relevant text. This is depicted in Figure 2. By zooming in and out of a particular concept space, users can locate concepts that appear relevant to their research, diving into an area of interest and reading the span of text that Watson has identified as germane.

In addition to extracting text from PDF-formatted research papers, we contributed the HTML of 382 Wikipedia articles related to a narrow domain: the desalination of water. We selected and extracted HTML from the relevant articles using Scrapy (<http://www.scrapy.org>), a Python library for generating custom web crawlers. We instructed our spider to begin at the Wikipedia article for desalination and crawl to a link depth of two, which produced a diverse selection of technical and non-technical information.

4.3 Illustrative Examples

Below are excerpts from a conversation between a user and Erasmus. For brevity, we have included only five questions and truncated the some of the answers.

How do sea birds drink water?

Marine birds can drink seawater because their cephalic 'salt' glands secrete a sodium chloride (NaCl) solution more concentrated than seawater. Salt gland secretion generates osmotically free water that sustains their other physiological processes. Acclimation to saline induces interstitial water and Na move into cells.

What are salt glands?

Schmidt-Nielsen et al. (1958) were the first to demonstrate SGS from a marine bird. They gave a cormorant (*Phalacrocorax auritus*) seawater orally and soon after observed in the bird's nares drops of a concentrated sodium chloride (NaCl) solution produced by the nasal 'salt' glands (Schmidt-Nielsen, 1960). When a bird drinks sea-water its w_{Na} , plasma osmolality (Osm) increase due excessive osmolyte content (mainly Na and Cl). The salt glands of marine (and some falconiform and desert birds) secrete excess NaCl via the salt glands in less water imbibed with it, thus generating free water needed to return Osm to its preload value. However, salt gland secretion is the culmination of a series of highly integrated interactions among the gut, kidneys, hindgut, salt glands, and supporting organs. Some aspects of these organs in Pekin ducks have been reviewed by Simon (1982)...

What is osmosis?

Reverse osmosis (RO) is a water purification technology that uses a semipermeable membrane to remove larger particles from drinking water. This membrane technology is not considered a proper filtration method. In reverse osmosis, an applied pressure is used to overcome osmotic pressure, a colligative property, that is driven by chemical potential, a thermodynamic parameter. Reverse osmosis can remove many types of molecules and ions from solutions, including bacteria, and is used in both industrial processes and the production of potable water. The result is that the solute is retained on the pressurized side of the membrane and the pure solvent is allowed to pass to the other side....

Tell me more about semipermeable membrane.

The simplest case is a red blood cell bounded by a semipermeable membrane and immersed in plasma. If the ions and molecules in the cytoplasm of the red cell and in plasma differ in concentrations for one or more of the ions or molecules for which the membrane is not permeable, then the impermeant ions or molecules will press (exert a pressure, a solute pressure) against both the inside and outside of the red cell membrane. The side of the membrane receiving the greater solute pressure will. Cell Biochemistry and Biophysics be displaced, thereby causing water and permeant ions and molecules to flow into the solution on that side

What other membranes exist in nature?

The presence of AQP 3 along the plasma membranes of basal epithelial cell of FW-acclimated eels (gill arch and primary filament epithelia), is similar to the sub-surface location of this protein which has been observed in rat urinary bladder epithelia as well as in rat and human skin (Matsuzaki et al., 1999; Matsuzaki et al., 2000). In rats, the role for AQP3 along the plasma membranes of sub-surface epithelial cells has been suggested to be related to the protection of the cells from dehydration. Such a role for mammalian AQP3 furthermore, is strengthened by the fact that AQP3 expression in rats is controlled by the osmolality of the extracellular fluid (Matsuzaki et al., 1999; Matsuzaki et al., 2000). In the eel, the location of AQP 3 along the plasma membranes of basal epithelial cell of FW-acclimated eels also suggests water movement across epithelial cells but its role could be somewhat different. It is hypothesised here that this water movement across the basal cells of the branchial epithelium could occur in order to release water entering apically from the external hypo-osmotic FW environment....

5. The Watsabi Project

The Watsabi system guides users to find answers to difficult and highly contextual questions related to the domain of agriculture. In bridging an online community with IBM's Watson, Watsabi brings together the real personal experience of farmers with the rich datasets of published verifiable information such as World Bank Agriculture (2015). This results in both reduced overhead due to trial and error, and increased creativity inspired by hard to find data.

5.1 Architecture

Watsabi uses a website that allows usage on almost any device connected to the internet. The architecture draws from existing online software-as-service stacks, with a backend/frontend distinction. On the backend we selected Laravel to handle our API and database management. Laravel out of the box provides powerful API management capabilities, and was able to work for our system with minimal configuration. On the frontend we use AngularJS, a modern JavaScript framework that provides a powerful templating engine. Angular couples effectively with Laravel. It features a service to aid in calling RESTful endpoints, as well as a powerful templating engine for generating a dynamic view for our website. Angular makes HTML development much easier and faster than with traditional backend frameworks such as PHP. This has allowed us to more rapidly develop extensible prototypes.

Watson interacts with Watsabi most directly via Angular's AJAX GET calls to Watson's JSON API, which provides rich responses to the user's natural language queries. Angular allows for simple parsing and utilization of the returned information, which provides the basis for the user chat experience. Watson's answers also include several identifying components which, together, form the basis of our intended Laravel (database) support by which community interaction is paired with Watson answers.

5.2 Process

When a user enters into the forum section, she can browse existing conversations, or start a new one. If she has a particular topic in mind, she can use the search function to narrow down existing conversations by relevant keywords. We also drew inspiration from community moderated websites, and implemented a voting system. If a user finds a question or answer to be particularly insightful, they can vote for it. Voting raises the profile of that particular question/answer, thus making it more visible to the community so that others may use its information. Voting also serves another important function. A highly rated question/response signals to Watson that it should weight the information contained higher, thus ensuring that future conversations with Watson will include the newly relevant information. Once an answer has reached a specified number of upvotes, Watson will be trained on that question/answer pair (this all occurs without the knowledge of the user). Once Watson is trained on this question/answer pair, it is removed from the forum page.

A user can also enter into the conversation system. This involves researching topics directly with Watson. The user can start a new conversation, or return to a previous one that has been saved by the system. Users then type a question into the "ask Watson" field. There are three components to this page.

- Chat box. The chat box is where the user's questions and Watson's responses are shown in a chat like format. As the user asks more questions they are added to the conversation.

- My ideas. Next to the chat box, there is a place where the user can save answers they found particularly useful. To save the answer, the user must click on the pin next to

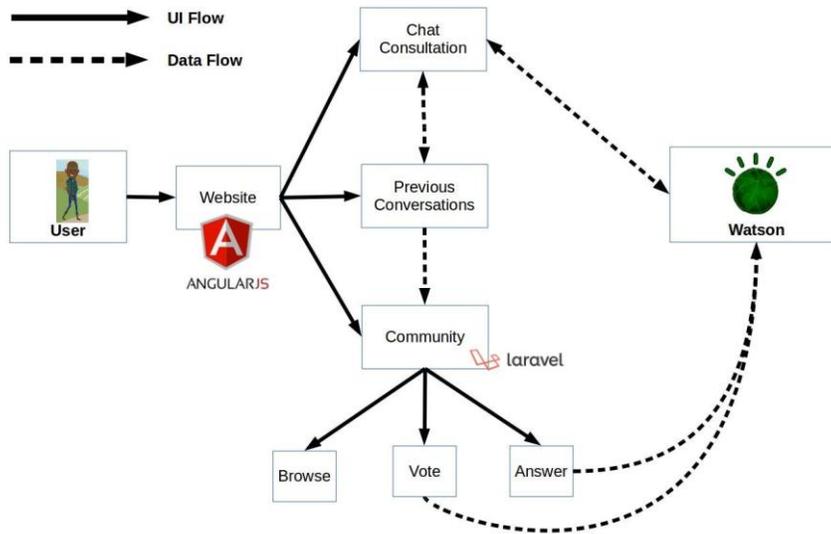


Figure 3. Computational Process in Watsabi

Watson’s response. This is to allow the user to shorten the interaction with Watson to only include the relevant and informative responses.

- Watsabi’s Brain. To the right of the chat box there is an evidence list. This includes the links to documents that Watson pulled the returned answer from. If

the answer was pulled from "Growing Bananas in Rwanda" a link to this document would pop up.

We trained Watson on a broad set of articles relating to agriculture and water systems. We initially focused on water scarcity problems and specifically uploaded documents related to water collection in dry areas. We then began to look more towards particular agricultural practices in various climates, and trained Watson on articles relating to these topics.

5.3 Illustrative Examples

Let us consider a specific use case scenario: A farmer from Rwanda might know for sure that she wants to plant bananas this season. She knows that bananas require a lot of land, and that she could possibly grow something else alongside of them, but she does not know what. She speaks to other farmers in the area, but they cannot help. She logs onto the Watsabi system and searches the forums for bananas, but nothing relevant comes up. She asks Watsabi “what can I intercrop with bananas?”. Watsabi polls its knowledge base. It cannot come up with a definitive answer, but it provides five relevant articles with important excerpts highlighted. Our user sees an interesting passage that mentions ginger and bananas, and asks Watsabi “in which climate is ginger viable?” And “when should I plant ginger?”. She finds that Ginger can indeed be planted successfully in the shade of banana trees. She goes online to find the current price of ginger and determines that it will be profitable. She marks that she has found an answer to her question, and the information becomes available to anyone else using the system with a similar problem.

Below are excerpts from a conversation between a user and Watsabi. For brevity, we have included only three questions and truncated the some of the answers.

What can I intercrop with bananas?

Intercropping of bananas with other crops is a common soil fertility improvement effort, which when it occurs, is sometimes a secondary benefit to the primary purpose of the practice. The following crops can be intercropped with bananas: Perennial crops such as coffee can be intercropped with bananas for provision of shed especially at establishment. They also benefit the banana by recycling nutrients from deep capture, because of their rooting system, via litter fall. Bean is the annual crop most commonly associated with banana. The two crops are compatible in a multi-story system as bean does not compete with banana above ground and is more shade tolerant than most other food crops. Maize and sweet potatoes are intercropped with the young banana plants and are phased out at canopy closure. Fruit trees like jack fruit and papaw are established with the plantation as sources of fruit and to serve as windbreaks.

Should I intercrop coffee with bananas?

In Uganda and Rwanda, during a perception study on intercropping coffee and banana, Jassogne et al. (2013) and Deraeck (2011) found that farmers would intercrop for six major reasons. These reasons, in decreasing order of importance, are: (i) cash and food are provided from the same piece of land under conditions of land scarcity, and there is increased income; (ii) coffee in banana shade is more resilient to drought; (iii) banana provides in situ mulching material for both crops; (iv) bananas in the intercrop provide motivation to manage coffee well when it is too young to produce; (v) coffee under shade gives larger berries; and (vi) additional animal feed is supplied from bananas grown in coffee fields.

Are there benefits to intercropping?

The first reason for intercropping indicates that intercropping coffee and banana increases food security of the farmer. In a situation where land becomes scarce, intercropping cash and food crops is a way to manage risks. The farmer gets food from his or her bananas throughout the year, as well as a small continuous income from the surplus. Furthermore, once or twice a year, depending on the location in Uganda, the smallholder farmer gets a 'cash boom' from the coffee. The study by van Asten et al. (2011) showed that for Robusta and Arabica farmers in Uganda, intercropping coffee with banana was more profitable than either monocrop. This was the case even though the bananas in a Robusta-banana intercropped field suffered and their yield value decreased - while the opposite occurred in an Arabica-banana intercropped field (Fig. 18.1). These contrasting effects of intercropping on banana yields are because Robusta coffee trees are more demanding in terms of space than Arabica trees and so are more competitive when intercropped with bananas. Intercropping had no significant effect on coffee yields....

6. Discussion

There are several dimensions to our analysis of the above experiment on using Watson in the Georgia Tech CS4803/8803 class on Computational Creativity in Spring 2015.

6.1 Products from the experiment

A common theme that emerged from the 6 projects was the use of Watson+ as an *intelligent research assistant*. While the original *Jeopardy!*-playing Watson system answered a series of questions, the questions were largely independent of one another. However, as scientists we know that insight rarely arises out of a single question or answer. Instead, science typically reaches deep insights only through sustained systematic research engaging a series of questions and answers. The power of this research process depends on the quality of questions scientists generate in a given context and the insights the answers provide into the problems of interest. The

two factors are highly related: the insightfulness of the answers depends in part on the quality of questions, and the quality of (subsequent) questions depends partly on the insights of the (preceding) answers. Thus, the issue in using Watson as an intelligent research assistant becomes how can we enable people to converse with Watson to develop deep insights into a situation or a problem? This requires adding both semantic processing to Watson and context to the human interaction with it. All six projects in the Georgia Tech Spring 2015 CS4803/8803 class on Computational Creativity explored different ways of adding context and semantics to user interactions with Watson.

At the end of the Spring 2015 term, we shared short videos of the six projects with IBM. As a result, IBM invited us to join its commercial ecosystem and encouraged us to spawn a startup around the concept of Watson+ as an intelligent research assistant for biologically inspired design. In the summer of 2015 we won a small NSF I-Corps grant to explore the concept in the marketplace. Customer discovery in early fall of 2015 showed much interest in the concept of an intelligent research assistant for biologically inspired design, but not (yet) a paying customer. We continue customer discovery at the time of the writing of this paper.

6.2 Lessons from the experiment

Our class' experience with Watson helped us to identify several best practices.

- (a) Watson is strong at answering specific questions related to expert documents it has been trained on but it does not perform any better than existing search engines for simple and generic searches.
- (b) Watson performs best with structured textual documents: documents divided into short, topically contained paragraphs separated by meaningful headers. This may require significant data engineering.
- (c) An obvious source for structured textual documents of this kind is Wikipedia. By extracting the HTML source of relevant Wikipedia pages, we achieved a significant jump in performance. However, Watson often returned sections that were too large to be useful. Thus, an automated process for segmenting the already structured Wikipedia input might further improve Watson's output.
- (d) Other document types—such as .pdf and .doc files— need to be significantly enriched to delineate sub-sections.
- (e) When training Watson, question-answer pairs should be chosen carefully. Watson seems to benefit from training examples that cover a breadth of topics both within and around the target domain. Watson also seems to benefit from “bridge” questions, which link the responsive text from existing questions to each other. With these links established in its corpus, Watson can more easily locate similar questions when faced with a previously unseen query. Depending on how thoroughly an instance has been trained.
- (f) Putting enough expert documents into the corpus is a daunting task, especially if includes parsing, structuring, and labeling documents. As Watsabi illustrates, crowdsourcing may be way out of this problem of data engineering if a clever method is used to automatically curate the training documentation.
- (g) Watson can be very (and often unexpectedly) sensitive to the presence of keywords in a query string or certain styles of grammatical construction. Use of synonyms for potential keywords and variations in query grammar are likely to receive greater precision and recall.

6.3 Results from the experiment

Irrespective of its merits as a cognitive system, Watson was an excellent vehicle for teaching about cognitive systems. The Georgia Tech CS4803/8803 class on Computational Creativity in Spring 2015 in general received very positive student responses in the anonymous opinion surveys conducted by the university. Student motivation and engagement were very high throughout the semester and produced projects of better quality than in previous offerings of the class. Pedagogical techniques such as reflective critiques of Watson and comparisons with well established cognitive systems provided powerful lessons about cognitive systems, making their core characteristics more transparent. Almost all students in the class found working with Watson a unique learning experience that was simultaneously motivating and engaging, productive and creative, and successful and satisfying.

7. Watson at the Task and Knowledge Levels

As we mentioned in the introduction, despite the many IBM articles on some technical issues, the high-level design of Watson is still quite opaque. Given the above experiment in using Watson, we seek to develop a task- and knowledge-level description of Watson that may provide some insights into its high-level design. A task-level description of a cognitive system specifies the tasks the system addresses, the high-level processes the system uses to accomplish the tasks, and the subtasks these processes spawn, without detailed specifications of the algorithms and the representations that realize the processes (Marr 1977). Similarly, a knowledge-level description of a cognitive system specifies the goals of the system, the body of knowledge the system possesses, and the processes that use the knowledge to achieve the goals, again without detailed specifications of the algorithms and the representations (Newell 1981). It is important to restate that by Watson here we mean the Watson Engagement Advisor.

7.1 Watson is a question answering system

We already know that Watson is a question answering system. In particular, Watson takes a question as input, and has the goal of giving a ranked list of answers above a threshold of confidence as output. Here the confidence threshold is a tunable parameter of the system.

7.2 Watson views question answering as a memory task

One can adopt several epistemic views of the question answering task: question answering as a natural language processing task, as a knowledge-based reasoning task, as a machine learning task, etc. Watson views question answering as a *memory task*. That is, Watson first assumes a massively populated memory of answers. Then, given a question, Watson views its task as retrieving all answers above a preset confidence threshold that match the question.

This view of question answering is similar in some respects to the stance of cognitive systems research on case-based reasoning (Riesbeck & Schank 1989; Kolodner 1993). The case-based reasoning view of intelligence asserts that (a) the problems intelligent agents encounter tend to recur, perhaps with small variations, (b) expertise lies in acquiring a memory of cases that encapsulate solutions to known problems, (c) similar problems often have similar solutions, and (d) given a new problem, memory retrieves the most similar cases that either provide a solution to the problem or provide an almost correct solution that simply needs to be tweaked to fit the new problem. Of course there are also important differences. For example, case-based reasoning also

entails modifying the retrieved solution to the new problem; Watson however does not adapt the answers it retrieves. Nevertheless, the general epistemic view of question answering as a memory task has important implications for Watson's processing of questions and answers.

7.3 Watson is a textual case retrieval system

We already have alluded to textual case-based reasoning in which cases are available as textual documents. As in textual case-based reasoning, Watson views each textual document as a case. Further, Watson views each document segment (e.g., a section) as a subcase, thus organizing each document in a case-subcase hierarchy. It is important that each subcase be small otherwise Watson will retrieve long texts as answers. Each subcase in this hierarchy has its own index and it is important that these indices are designed well enough to discriminate among textual cases. Thus, an index such as "Conclusions" is not very useful because a very large number of textual cases likely have the same or similar indices. At present, this segmentation of a textual case into subcases and labeling of all the subcases should be done manually to optimize performance. Thus, using Watson may require significant knowledge engineering.

7.4 Watson sets up meta-memories for retrieving textual cases

To help retrieve textual cases matching a question, a user needs to manually train Watson to set up meta-memories in the form of question clusters. In this process, the user first specifies a question and then specifies the answer to the question in the form a textual subcase. Next, the user specifies another form of the same question with the same answer, and so on. In general, the larger the number of questions in a cluster for a given answer, the better Watson's recall and precision should be. Thus, once again, using Watson may require significant knowledge engineering. Some of this effort can be automated by automatically generating variations of questions.

7.5 Watson uses limited processing of input questions

Watson's processing of input questions remains opaque. It seems to be using only limited lexical and syntactic processing to generate a vector that identifies keywords and their placement in the question. We found that additional processing of questions as in Erasmus improved Watson's performance.

7.6 Watson uses a weak notion of similarity in case retrieval

Given the vector representation of the input question, Watson classifies the question into one or more meta-memories. Again Watson's mechanism for this question classification remains opaque. However, it appears to use a weak notion of similarity for this classification, invoking all relevant meta-memories which point to matching answers to the input question.

7.7 Watson uses stronger notions of evidence for case scoring

Given multiple answers for the input question, Watson calculates a confidence value in each answer, ranks the answers by their confidence scores, and returns all answers above a user specified threshold of confidence. The original version of Watson that played *Jeopardy!* appeared to use multiple sources of information for calculating the confidence in an answer. The Watson Engagement Advisor appears to be using a single mechanism that calculates the confidence value

based on the match between the vector representation of the input question and the vector representation of the answer itself.

8. Conclusions

We now return to the question we raised in the introduction: what is Watson? It is interesting to note that despite IBM's massive publicity of Watson, this is still a legitimate question because much of Watson is still opaque. Our analysis suggests that Watson, and Watson Engagement Advisor in particular, views question answering as a memory task similar to the case-based reasoning view of intelligence. Watson answers questions through textual case retrieval and uses meta-memories for retrieving the textual cases.

We can now examine each of the possible answers for the question "what is Watson" we posited in the introduction:

(i) *Is Watson a cognitive system?* Despite IBM's claims, this is a not an easy question to answer in part because there are few precise characterizations of a cognitive system. In the introduction, we characterized a cognitive system as a system that is inspired by cognitive phenomena (what mind does), if not also cognitive processes (how mind works), and that provides insights into cognitive phenomena and processes. On one hand, question answering is a cognitive phenomenon; on the other, there is little evidence that the design of Watson was inspired by cognitive processes. Instead, the focus of the Watson project appears to have been on engineering a system that performed well enough at *Jeopardy!* to beat most human players. On one hand, the design of Watson supports the view that intelligence comprises multiple strategies (Hendler 2014). There is also the clear parallel between Watson and textual case-based reasoning. Could it be that through an engineering feat, the Watson project has rediscovered some of the principles of cognition? On the other hand, Watson provides little evidence for any reasoning about the questions put to it or about the answers it provides. It does not, for example, modify any of the retrieved answers to fit the question.

(ii) *Is Watson a cognitive architecture?* Irrespective of whether or not Watson is a cognitive system, it is not a cognitive architecture in part because it does not have the generality of a cognitive architecture. Watson may be good at answering some types of questions in some contexts, but it is not very good at most anything else.

(iii) *Is Watson a knowledge-based system?* Yes, the original Watson that played *Jeopardy!* clearly was a knowledge-based system.

(iv) *Is Watson a knowledge system shell?* This is an apt description of Watson Engagement Advisor. Indeed, some aspects of the Watson Engagement appear to resemble the shells for case-based reasoning developed in the early 1990s such as the ReMind shell developed by the now defunct *Cognitive Systems Inc.* (Altoff et al. 1995).

(v) *Is Watson a programming language for cognitive computing?* In the introduction, we had characterized cognitive computing as the construction and use of cognitive assistants. Watson Engagement Adviser is a new programming language for constructing cognitive assistants that rely on question answering. The language provides high-level constructs for question answering through textual case retrieval including structures for building the case memory and meta-memory.

(vi) *Is Watson a software toolset for cognitive computing?* Watson Engagement Adviser is a software toolset for constructing cognitive assistants. However, the notion of a toolset is more

applicable to the Bluemix Platform that deliberately decomposes Watson into components and enables the user to compose and apply the components in various ways.

Acknowledgements

Goel, the first author of this paper, was the primary instructor for Georgia Tech CS4803/8803 class on Computational Creativity in Spring 2015; Wiltgen, the last author was the teaching assistant for the course; Creeden, Kumble, Salunke and Shetty developed the Erasmus system; and Anderson, Belknap, Hancock and Sheneman developed the Watsabi system. We are grateful to IBM for providing us with access to Watson Engagement Advisor as well as two Faculty Fellowships in support of this research. We thank Richard Darden, Pamela Induni, J. William Murdock, Armen Pischdotchian at IBM, and especially James Spohrer, the Director of IBM's Institute for Cognitive Systems, for their support of our work. Finally, we thank the anonymous reviewers of the previous version of this paper for their constructive and very useful critiques. This paper is a significant expansion of Goel et al. 2015a.

References

- Althoff, K-D., Bareiss, R., Manago, M., & Auriel, E. (1995) *A Review of Industrial Case-Based Reasoning Tools*. Oxford.
- Benyus, J. (1997) *Biomimicry: Innovation Inspired by Nature*. New York: William Morrow.
- Brown, E., Epstein, R., Murdock, J.W., & Fin, T-H. (2013) Tools and methods for building Watson. *IBM Research Report RC 25356*.
- Chandrasekaran, B. (1990) What Kind of Information Processing is Intelligence? In *Foundations of Artificial Intelligence – Sourcebook*, D. Partridge & Y. Wilks (editors), Cambridge University Press.
- Chandrasekaran, B., & Goel, A. (1988) From Numbers to Symbols to Knowledge Structures: Artificial Intelligence Perspectives on the Classification Task. *IEEE Transactions on Systems, Man, and Cybernetics* 18(3):415-424, 1988.
- Ferrucci D, Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., Schlaefer, N., & Welty, C. (2010) Building Watson: an overview of the DeepQA project. *AI Magazine*, 31:59–79.
- Ferrucci, D. (2012) Introduction to "This is Watson". *IBM Journal of Research and Development* 56(3-4):1-15.
- Goel, A., Creeden, B. Kumble, M., Shetty, S., & Salunke, S. (2015a) Using Watson for Enhancing Human-Computer Co-Creativity. In *Procs. AAAI 2015 Fall Symposium on Cognitive Assistance*, Arlington, Virginia, November 12-14, 2015.
- Goel, A., & Davies, J. (2011) Artificial Intelligence. In *Handbook of Intelligence*, R. Sternberg & S. Kauffman (editors), Chapter 23, pp. 468-484, Cambridge University Press, 3rd Edition, 2011.
- Goel, A., McAdams, D., & Stone, R. (2014) *Biologically Inspired Design: Computational Methods and Tools*, Springer.
- Goel, A., Zhang, G., Wiltgen, B., Zhang, Y., Vattam, S., & Yen, J. (2015) On the Benefits of Digital Libraries of Analogical Design: Documentation, Access, Analysis and Learning. *AI for Engineering Design, Analysis and Manufacturing*, 29(2).
- Hendler, J. (2013) Watson at RPI. Retrieved April 1st, 2016, from <http://www.slideshare.net/jahendler/watson-summer-review82013final>.
- Hendler, J. (2014) Watson: An Academic Perspective. Retrieved April 1st, 2016, from <http://www.slideshare.net/jahendler/watson-an-academics-perspective>.
- Kalyanpur, A., & Murdock, J.W. (2015) Unsupervised Entity-Relation Analysis in IBM Watson. In *Procs, Third Annual Conference on Advances in Cognitive Systems (ACS 2015)*, Atlanta, GA, 2015.
- Kelly, J., & Hamm, S. (2013) *Smart Machines: IBM's Watson and the Era of Cognitive Computing*. Columbia University Press.

- Kolodner, J. (1993) *Case-Based Reasoning*. Morgan Kaufmann.
- Langley, P. (2011) Artificial Intelligence and Cognitive Systems. *AISB Quarterly*.
- Langley, P. (2012). The cognitive systems paradigm. *Advances in Cognitive Systems*,1, 3–13.
- Langley, P, Laird, J. & Rogers, S. (2009) Cognitive Architectures: Research Issues and Challenges. *Cognitive Systems Research* 10(2): 141-160.
- Laird, J. (2012) *The Soar Cognitive Architecture*. Cambridge, MA: MIT Press..
- Lenat, D. (1995) CYC: A Large-Scale Investment in Knowledge Infrastructure, *CACM* 38 (11): 33-38.
- Marr, D. (1977) Artificial Intelligence: A Personal View. *Artificial Intelligence* 9: 37-48.
- McCarthy, J, Minsky, M, Rochester, N., & Shannon, C. (2006) A Proposal for a Dartmouth Summer Research Project on Artificial Intelligence (August 31, 1955), *AI Magazine*,
- Murdock, J.W., & Goel, A. (2008) Meta-Case-Based Reasoning: Self-Understanding for Self-Improvement. *Journal of Experimental and Theoretical Artificial Intelligence* 20(1): 1-36, 2008.
- Newell, A. (1981) The Knowledge Level: AAAI Presidential Address. *AI Magazine* 2(1).
- Riesbeck, C., & Schank, R. (1989) *Inside Case-Based Reasoning*. Lawrence Erlbaum.
- Shneiderman, B. (1996) The eyes have it: A task by data type taxonomy for information visualizations. In *Procs. IEEE Symposium on Visual Languages, 1996*.
- Shortliffe, E. (1976) *Computer-Based Medical Consultations*. Elsevier.
- Turian, J. (2013). *Using AlchemyAPI for Enterprise-Grade Text Analysis*. Technical report, AlchemyAPI.
- van Melle, W., Shortliffe, E.H., and Buchanan, B.G. (1984). EMYCIN: A Knowledge Engineer's Tool for Constructing Rule-Based Expert Systems. In Buchanan and Shortliffe (Eds.), *Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, pages 302-313.
- Vattam, S., & Goel, A. (2012) Interactive Analogical Retrieval. In *Procs. First Annual Conference on Cognitive Systems, 2012*.
- Vattam, S., & Goel, A. (2013) Biological Solutions for Engineering Problems: Cross-Domain Textual Case-Based Reasoning in Biologically Inspired Design. In *Procs. 21st International Conference on Case-Based Reasoning*, pp. 343-357.
- Vincent, J., & Mann, D. (2002) Systematic technology transfer from biology to engineering." *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 360(1791): 159-173.
- Watson Bluemix Platform (n.d.) <http://www.ibm.com/cloud-computing/bluemix/>; last accessed May 27, 2016.
- Watson Engagement Advisor (n.d.) http://www.ibm.com/smarterplanet/us/en/ibmwatson/engagement_advisor.html; last accessed May 27, 2016.
- Weber, R., Ashley, K., & Bruninghaus, S. (2005) Textual case-based reasoning. *The Knowledge Engineering Review*, 20(03), 255-260 (2005).
- Wollowski, M. (2014) Teaching with Watson. In *Procs. Fifth AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI'14)*, pp. 3044-3049.
- World Bank Agriculture (2015) Retrieved 2015-03-13 from <http://data.worldbank.org/topic/agriculture-and-rural-development>.
- Zadrozny, W., Gallager, S., Shalaby, W., & Avadhani, A. (2015a) Simulating IBM Watson in the Classroom. In *Procs. 46th ACM Technical Symposium on Computer Science Education (SIGCSE' 15)*, pp. 72-77.
- Zadrozny, W., de Pavia, V., & Moss, L. (2015) Explaining Watson: Polymath Style. In *Procs. 29th National Conference on AI (AAAI-2015)*.