Using Watson for Enhancing Human-Computer Co-Creativity

Ashok Goel, Brian Creeden, Mithun Kumble, Shanu Salunke, Abhinaya Shetty, Bryan Wiltgen

Design & Intelligence Laboratory, School of Interactive Computing, Georgia Institute of Technology
goel@cc.gatech.edu, brian.creeden@gatech.edu, mkumble3@gatech.edu, salunkeshanu91@gmail.com, shetty_abhinaya@yahoo.co.in,
bryan.wiltgen@gatech.edu

Abstract

We describe an experiment in using IBM's Watson cognitive system to teach about human-computer co-creativity in a Georgia Tech Spring 2015 class on computational creativity. The project-based class used Watson to support biologically inspired design, a design paradigm that uses biological systems as analogues for inventing technological systems. The twenty-four students in the class self-organized into six teams of four students each, and developed semester-long projects that built on Watson to support biologically inspired design. In this paper, we describe this experiment in using Watson to teach about human-computer co-creativity, present one project in detail, and summarize the remaining five projects. We also draw lessons on building on Watson for (i) supporting biologically inspired design, and (ii) enhancing human-computer co-creativity.

Background, Motivations and Goals

Creativity is one of humanity's most special traits and resources as well as goals and ideals. Humans are creative as individuals, as societies, and as a species. It is human creativity that leads to visual and performance arts, scientific discoveries and technological inventions, social and cultural movements as well as political and economic revolutions. Successful business organizations value creative people. While it was the increase in human productivity due to the use of the assembly line by Ford Motor Company in the early twentieth century that was responsible for the production of the first commonly affordable automobiles, it was Henry Ford's creativity that led to the introduction of the assembly line in his eponymous company (Dasgupta 1996). All this naturally raises several questions for computational creativity: What is creativity? Can creativity be enhanced? Can creativity be taught? How might a computer aid human creativity? How might we enhance human-computer co-creativity where the creativity emerges from interactions between humans and computers?

Goel, the first author of this paper, conducts research on computational creativity. In late 2014, IBM gave him free access to a version of Watson (Brown et al. 2013; Ferruci et al. 2010) in the cloud called the Watson Engagement Advisor. In Spring 2015, he used Watson in the Georgia Tech CS4803/8803 class on Computational Creativity with Wiltgen as the teaching assistant. The project-based class used Watson to support biologically inspired design, a design paradigm that uses biological systems as analogues for inventing technological systems. The twenty-four students in the class self-organized into six teams of four students each, and developed semester-long projects that built on Watson to support biologically inspired design. In this paper, we describe this experiment in using Watson to teach about human-computer co-creativity, present the project of one team, consisting of Creeden, Kumble, Salunke and Shetty, in some detail, and summarize the remaining five projects. We also draw some lessons about "best practices" for using Watson for (i) supporting biologically inspired design, and (ii) enhancing human-computer co-creativity.

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.

According to the course description handed out to the students at the start of the class, the *learning goals* were (1) To become familiar with the literature on computational creativity (concepts, methods, tasks), (2) To become familiar with the state of art in computational creativity (systems, techniques, tools), (3) To learn about the processes of designing, developing and deploying interactive/autonomous creative systems from ideation to realization, (4) To acquire experience in designing an interactive creative tool, and (5) To become an independent thinker in computational creativity. The observable *learning outcomes* were (i) To be able to analyze/critique developments

in computational creativity like an expert, (ii) To be able to identify problems in addressing creative tasks, and (iii) To be able to design, develop, and document creativity tools from ideation to realization for addressing creative tasks. The *learning assessments* were (a) Daily Quizzes, (b) Two class presentations (1 individual, 1 team), and (c) Design projects (4 deliverables that later became 6). Also, according to the course description, the *learning strategies* consisted of (1) Personalized learning (reading, analyzing, critiquing papers in computational creativity), (2) Collaborative learning (presenting, discussing, critiquing papers on computational creativity), (3) Learning by example (case studies of computational creativity), (4), Learning by doing (small group exercises), (5) Project-based learning (collaborative project), (6) Peer-to-peer learning (sharing and critiquing work), and (7) Learning by reflection (documenting and critiquing own work).

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

The Class Projects

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). This paradigm has inspired many famous designers in the history of design including Leonardo da Vinci, appears in a wide variety of design domains ranging from architecture to computing to engineering to systems. However, over the last generation the paradigm has become a movement in engineering and systems design, pulled in part by the growing need for environmentally sustainable development and pushed partly by the 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).

Biologically inspired design covers at least three of the six core processes of creativity enumerated above. By definition, biologically inspired design engages design thinking and systems thinking; also by definition, it engages cross-domain analogical transfer from biology to design. These core processes of creativity present major challenges for developing computational techniques and tools for supporting biologically inspired design. For example, the cross-domain nature of 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, a computational tool for supporting biologically inspired design aimed at designers, for example, must enable them to navigate the unfamiliar domain of biology.

The general design process followed by the 6 design teams for using Watson to support biologically inspired design may be decomposed into two phases: an initial learning phase and a latter 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*. (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). (3) The teams generated about 600 questions relevant to their use cases. (4) The teams identified the best answers in their 200 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 from 200 to about 500 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).

It is noteworthy that all 6 projects significantly evolved from the initial learning phase to the latter open-ended research phase. In particular, in the initial phase, the 6 projects tended to view Watson as an interactive tool for aiding human creativity. However, in the latter open-ended phase, each of the 6 projects at least to some degree evolved into interactive tools for enhancing human computer co-creativity. For example, the Erasmus project de-

scribed below augmented Watson with another tool called AlchemyAPI as well as custom-made context-specific pre- and post-semantic processing for iteratively asking and answering questions. Thus, the creativity did not reside solely in the human user of Watson; instead creativity emerged out of interactions between the user and Watson.

The Erasmus Project

The Erasmus system both supports human creativity by affording access to snippets from biology articles relevant to a design-related question and enhances human-computer co-creativity by enabling designers to generate new questions. Erasmus takes advantage of Watson's natural language information retrieval abilities to significantly reduce the amount of time designers spend acquiring the foundational knowledge necessary to determine whether a biological design concept might contain insights into a particular design problem. It provides researchers with a quickly digestible visual map of the 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.

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. Al-

Figure 1: The Erasmus architecture.

chemyAPI, 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 packages and presents this information to the user in a simple front-end.

Watson's strengths lie in its ability to accept a natural language question from its user and accurately identify responsive spans of text from documents within its corpus. The answer returned, however, depends on the documents to which Watson has access, the way in which those documents are annotated and formatted, the questions used to train the instance of Watson being used, and the phrasing of the question asked by the user. The accuracy of this answer varies widely and, from the perspective of a user unfamiliar with Watson's mechanics, in surprising ways.

Erasmus attempts to alleviate some of Watson's opacity and to complement its strengths toward the particular needs of biologically inspired design researchers by mediating interactions between Watson and the user. It integrates multiple components, as outlined in Figure 1: 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. The system then returns the most relevant documents, along with their matching concepts and weighted relevancy, to the front-end.
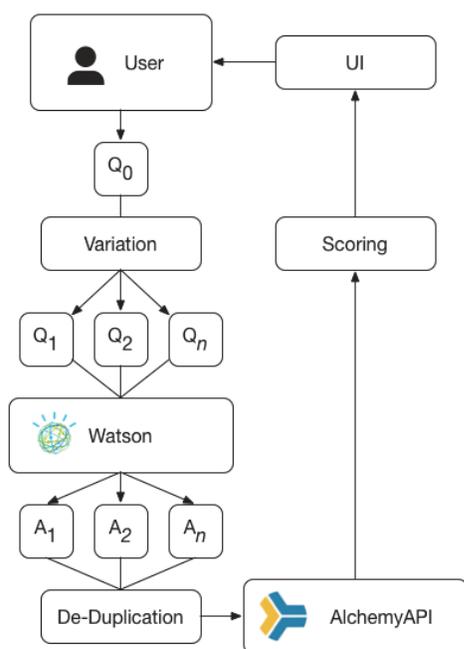
Process

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. While it is possible to request that Watson's API return multiple responses to a user query—rather than the single answer in which Watson is most confident—we are able to further expand the scope of the resulting concept space by generating small variations on verbiage and word order, which forces Watson to include documents from its corpus that it might have considered irrelevant. While this level of filtering is desirable when a Watson user is seeking a single, direct answer to a question, the creative nature of design research benefits from responses that contain less-relevant material (Thudt, Hinrichs & Carpendale 2015). 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 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 de-duplication 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. If multiple answers are highly similar, we retain the answer that provides the most detail.

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 can score each answer for its conceptual overlap with the question, a process depicted in Figure 2. 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
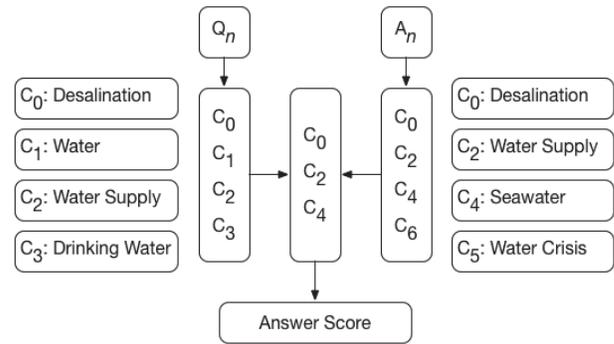

Figure 2: Erasmus's answer scoring process.

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 information visualization task taxonomy: overview, zoom, filter, details-on-demand, relate, history, extract (Shneiderman 1996). 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 3. By zooming in and out of a particular concept space,
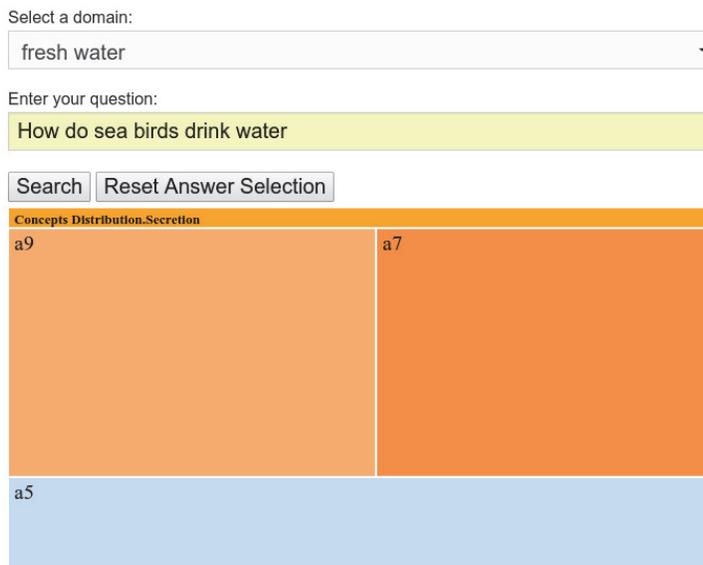

Figure 3: An Erasmus treemap, zoomed to display each answer's share of a concept and the related text.

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 the absence of a structured ontology of biological systems, Watson's information retrieval capabilities were essential to creating a functional prototype. 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 were careful to select literature that specifically covered natural desalination systems (e.g., seabirds, mangroves). We gave Watson no explicit metadata that might have allowed it to identify the desalination processes that were the focus of our searches, but it performed adequately with only the loosely structured header markup of the Wikipedia files to guide it.

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.

Illustrative Example

To compare Erasmus' user experience with Watson's—unaided by AlchemyAPI and our filtering processes—we ran a query in both systems from the fictional perspective of a biomimetic researcher intent on learning more about how seabirds desalinate the water that they drink. Our user began their search with a generic query: "How do sea birds drink water?"

Our instance of Watson responded with seventy-seven answers, of which only the top five are accessible from the Watson UI, as depicted in Figure 4. Users must scroll through the single text field to read all of the material presented, which—even with only five answers at an average of 671 words per answer—is a daunting task.

While Erasmus does not address Watson's full seventy-seven-document result set, it expands its scope from the top five answers to the top ten, as scored by Watson's confidence metric. Erasmus displays the embodied concept neighborhood in a treemap, as depicted in Figure 5. In this case, the treemap is composed of twenty-six concept blocks, each made of anywhere from one to four answer blocks. The relative size of each concept block and each answer block represents our scoring mechanism's estimation of its importance to the concept neighborhood.

Users can click on any of the concept blocks to zoom into that block. In this instance, we have selected "secretion." Once zoomed, the treemap displays the concept block alongside the answer text associated with secretion, as depicted in Figure 3. This allows users to more easily visualize the full concept neighborhood and to investigate topics of interest without engaging in the onerous task of reading the full response set.
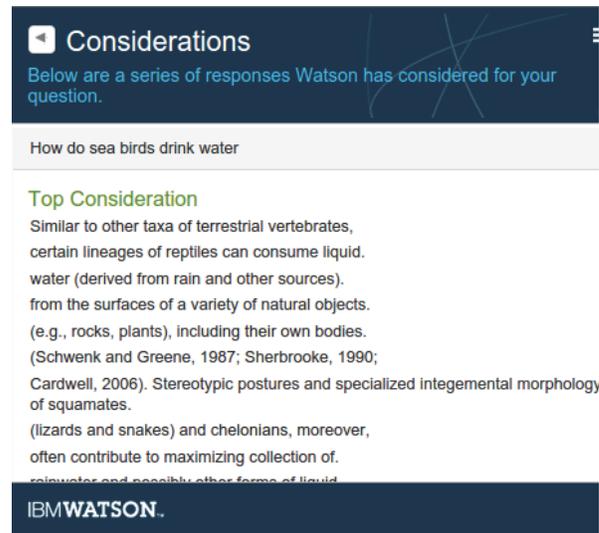

Figure 4: The Watson UI's response. The answer in which Watson has the most confidence, which is not topical in this case, is displayed first.

Evaluation

Although the particulars of Watson's operation remain a trade secret, we can infer some broad principles and make a few educated guesses. Watson appears to operate by chunking the documents in its corpus into sub-documents, delineating them by their HTML header tags and then scoring the sub-documents for relevance. Scoring seems to work by matching word tokens between the query and answer candidates, accounting for frequency.

The corpus attached to our Watson instance contains a broad set of knowledge relating to biological desalination systems, but the unstructured nature of the text made it difficult for Watson to identify spans containing the details of specific desalination processes. As we demonstrated in our example, we asked Watson how seabirds desalinate water, but its top answer related to reptiles. In our tests, Watson's full answer set often contained the span of text that we had hoped to retrieve, but it was frequently buried beneath irrelevant results. We ran several tests in which we manually selected a document for its high degree of topicality, but we were rarely able to coax Watson into producing our target at or near the top of its answer set.

Watson's precision and recall when retrieving Wikipedia articles was noticeably superior to its ability to find correct answers from research publications. Part of its trouble was likely due to the lack of structure: all of the research documents in our corpus were originally contained in .pdf files, which required us to extract their text with PDFMiner (http://www.unixuser.org/~euske/python/pdfminer/) and then manually review the result for errors. This strategy is not scalable, but a team with access to an advanced optical character recognition tool might be able to automatically add markup to research texts, which should at least match

Watson's performance on the similarly encoded Wikipedia HTML.

We were entirely thwarted by information encoded in graphs and figures, which Watson had no way to comprehend. These images were both common and critical to understanding the core findings in nearly every topical research paper.

Best Practices

Given our experience with Watson, it seems wise to seed the system with a large and diverse corpus of structured and semi-structured documents in a marked-up format that supports Watson's chunking strategy. The most obvious source for pre-formatted documents is Wikipedia. By extracting the HTML source of relevant Wikipedia pages, we achieved a significant jump in performance. Other document types—such as .pdf and .doc files—should be enriched to delineate sub-sections. Watson often returned sub-documents that were too large to be useful, so some automated means of chunking the already-structured Wikipedia input might further improve Watson's output.

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, Watson can be unexpectedly sensitive to the presence of keywords in a query string or certain styles of grammatical construction. Users who experiment with synonyms for potential keywords and variations in query grammar are likely to receive greater precision and recall.

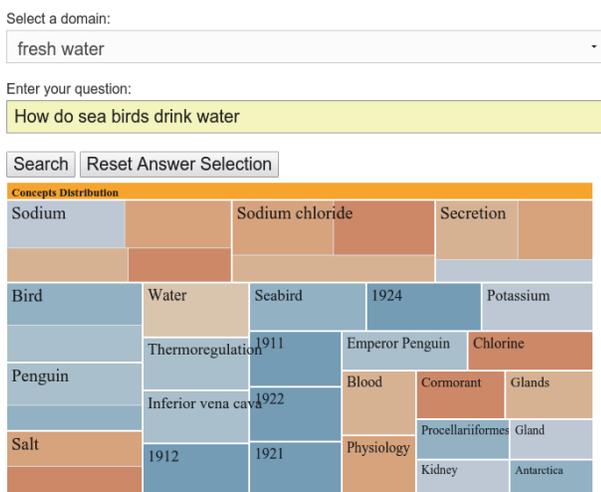*Figure 5: The concept neighborhood treemap, Erasmus's initial response to a user question.*

The Remaining Five Projects

As previously mentioned, there were 6 team projects in the Georgia Tech Computational Creativity class in Spring 2015. We have just described the Erasmus project in detail. Now we briefly summarize the remaining 5 projects. We also highlight some best practices of using Watson identified by the 5 teams.

Watson BioMaterial

This project focused on materials in the context of biologically inspired design. Specifically, this project is an Android app that allows a human user to search for materials relevant to her. This can be done in two ways. (1) A user can submit an unstructured search query that allows the user to search for a material based on a feature. (2) A user can submit a structured search query that allows the user to search for materials based on two or more features and also based on a related material.

Twenty Questions[1]

This team developed an interactive website modeled after the game 20 Questions. As in the 20 Questions game, the user's interaction with the website is structured in rounds. For each round, the user asks questions to the system, and the system returns snippets from the top five articles retrieved by Watson as answers. The user can then select the article that she finds most relevant and decide whether or not to continue. If the user decides to continue, the system will then suggest a set of keywords that the user can use for future searches. Then another round starts, and so on. The system leverages both the questions and its tracking of the context (such as user feedback about its responses) to drive the user towards a desirable document.

Ask Jill

This team made Jill, an interactive website that supports researchers conducting literature reviews. A user can go to the Jill website and write her research paper within its interface. As she does so, she can highlight text and use that text as a query to Watson. The site will return paper snippets from relevant papers in Watson's corpus. The user can then select a retrieved paper to add to her research paper. The Jill project builds on the Watson-powered retrieval by leveraging the site's ongoing context with the user. Jill records both the queries and the results returned by Watson, and it allows a user to favorite retrieved papers. Both of these features allow the user to store and conveniently retrieve the context of his or her work.

---

[1] We have simplified this project's name.

SustArch

The SustArch project is both a research tool and a community space for sustainable architecture, specifically as that field relates to biologically inspired designs for passive temperature control. SustArch, an Android app, allows a user to research this topic using a Watson-powered search engine. The authors describe an interesting hierarchical question-linking strategy that they used to train Watson, which they propose could allow Watson to potentially return multiple biological systems for a given high-level search. The team has also designed and mocked up a community space that they call the "Marketplace." In the Marketplace, a user would be able to browse, buy, sell and discuss designs created by herself or other users.

Watsabi

Although they started with biologically inspired design in mind, the Watsabi team wound up developing an interactive website to help people answer agricultural questions. A human user can go to the site, ask a Watson-based engine questions about agriculture, and the site will attempt to retrieve an answer. Similar to Jill, the Watsabi site will also record these ongoing interactions and allow the user to identify meaningful responses. If the Watson part of the site fails to retrieve a good enough answer, the user can then go to a forum where she can conduct question asking/answering with other users. Alongside this, the Watsabi team designed a way in which the forum aspect of their site could drive additional training of Watson. If an answer in the forum receives enough "upvotes," it will automatically be used to train Watson and be removed from the forum. Unfortunately, the team was unable to implement this because we lacked access to a training API to Watson, but this design nevertheless addresses how one might sustainably train Watson to support a user base.

Best Practices

Most of the final project reports from these five teams explicitly wrote a section on best practices for Watson. Here, we highlight some of what was mentioned.

(1) Well annotated (or structured) data—either the need for it or how time-consuming it is to produce—was a common topic among the projects. In particular, the Watsabi team mentions exploring automation to address the time needed, and they suggest that their crowd-sourcing approach may help overcome it.

(2) Training Watson is another topic that appeared in more than one report. As previously mentioned, the Watsabi team proposed a crowd-sourced approach to training. The Watson BioMaterials team proposed a similar feedback loop, but there the app developers were in charge of improving the knowledgebase. The Jill team proposes a strategy to training that we summarize here in three steps: (i) train Watson on all possible questions that the corpus papers can answer; (ii) train Watson on alternative versions of those questions; and (iii) if the first two steps prove insufficient after testing, add additional papers to the corpus and train on those papers.

Discussion and Conclusions

The 6 projects on using Watson for supporting human-computer co-creativity were 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 AlchemyAPI, 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 amd SustArch) were mobile apps running on the Android smart phone operating system. The variety of these projects indicates both the range of potential applications of Watson as well as the range of opportunities available for using Watson as an educational tool.

A common theme that emerged from the 6 projects was the use of Watson as an *intelligent research assistant* for realizing human-computer co-creativity. 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. The 6 projects explored different ways of adding context and semantics to user interactions with Watson.

This brings us to some of the limitations of Watson, and in particular to the limitations of the Watson Engagement Advisor. It is important to note that using Watson requires significant knowledge engineering. First, Watson needs to be seeded with natural language articles. Second, the natural language articles need to be well structured and/or an-

notated by semantic tags. Third, a developer needs to train Watson, which consists of (a) developing a taxonomy of questions of potential interest and (b) pairing the best answers to questions in the taxonomy. While the students in the Computational Creativity class found these knowledge engineering tasks manageable, they were quite frustrated that while the Watson Engagement Advisor provided a "dashboard" to ask questions and get answers from Watson, it did not provide any insights into the internal workings of Watson behind the dashboard. This limited students' understanding and use of Watson. We recommend that IBM consider releasing a more transparent and accessible version of Watson for education. On the other hand, most students in the class also found working with Watson a unique learning experience that was simultaneously motivating and engaging, productive and creative, and successful and satisfying.

While in this experiment we used Watson as an intelligent research assistant to support teaching and learning about biologically inspired design and computational creativity, we believe that Watson can be used in a large number of educational settings. Wolloski (2014) and Zadrozny et al. (2015) describe two other experiments with using Watson in a classroom. We are presently exploring the use of Watson as a cognitive system for answering frequently asked questions in an online class. Goel, the first author of this paper, teaches an online course CS 7637 Knowledge-Based AI: Cognitive Systems (KBAI for short; Goel & Joyner 2015) as part of Georgia Tech's Online MS in CS program (http://www.omscs.gatech.edu/). The online course uses Piazza (https://piazza.com) as the forum for online class discussions. The classroom discussions on the Piazza forum in the KBAI course tend to be both extensive and intensive, attracting about ~6950 and ~11,000 messages from ~170 and ~240 students in the Fall 2014 and the Spring 2015 classes, respectively. We believe that the large numbers of messages in the discussion forums are indicative of the strong motivation and deep engagement of the students in the KBAI classes. Nevertheless, these large numbers also make for significant additional work for the teaching team that needs to monitor all messages and answer a good subset of them. Thus, using the questions and answers from the Fall 2014 and Spring 2015 KBAI classes, we are developing a new Watson-powered technology to automatically answer frequently asked questions in future offerings of the online KBAI class.

References

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.

Dasgupta, S. (1996) *Technology and Creativity.* Oxford University Press.

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.

Goel, A., & Joyner, D. (2015) An Experiment in Teaching Cognitive Systems Online. Georgia Tech Technical Report GT-IC-15-01, July 2015.

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).*

Shneiderman, B. (1996) The eyes have it: A task by data type taxonomy for information visualizations. In Procs. IEEE Symposium on *Visual Languages, 1996.*

Thudt, A., Hinrichs, U., & Carpendale, S. (2015) A Modular Approach to Promote Creativity and Inspiration in Search. In *Procs. 2015 ACM SIGCHI Conference on Creativity and Cognition*.

Turian, J. (2013). *Using AlchemyAPI for Enterprise-Grade Text Analysis*. Technical report, AlchemyAPI.

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.

Wollowski, M. (2014) Teaching with Watson. In *Procs. Fifth AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI'14),* pp. 3044-3049.

Zadrozny, W., Gallager, S., Shalaby, W., & Avadhani, A. (2015) Simulating IBM Watson in the Classroom. In *Procs. 46th ACM Technical Symposium on Computer Science Education (SIGCSE' 15),* pp. 72-77.