Journal of Artificial General Intelligence 6(1) 111-129, 2015
DOI: 10.1515/jagi-2015-0006

Submitted 2015-05-31
Accepted 2015-11-27

A Play on Words: Using Cognitive Computing as a Basis for AI Solvers in Word Puzzles

Thomas Manzini MANZIT@RPI.EDU
Simon Ellis ELLISS5@RPI.EDU
James Hendler HENDLER@CS.RPI.EDU
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180, USA

Editors: Tarek R. Besold, Kai-Uwe Kühnberger, Tony Veale

Abstract

In this paper we offer a model, drawing inspiration from human cognition and based upon the pipeline developed for IBM's Watson, which solves clues in a type of word puzzle called *syllacrostics*. We briefly discuss its situation with respect to the greater field of artificial general intelligence (AGI) and how this process and model might be applied to other types of word puzzles. We present an overview of a system that has been developed to solve syllacrostics.

Keywords: word puzzles, brain teaser, games, cognitive computing, cognition, pipeline, watson, syllacrostic

1. Introduction

Word puzzles are a fun pastime for people of all ages. They range in type from simple word searches to cryptic crosswords and can be simple or can leave even advanced human solvers scratching their heads. The use of a brute force approach to computing solutions to abstract problems in word puzzles can result in lengthy compute times and extreme overuse of memory. Our goal was to attempt to develop a cognitive computing system that is able to use knowledge and reasoning to select fewer candidate answers and to use scorers and a ranking system to determine which are the most relevant in order to solve the puzzle.

We will briefly discuss the syllacrostic as a form of word puzzle, explain the game, rules, and types of clues that someone playing the game will likely encounter. We will then explain the approach which we believe a typical human player might take when playing this type of game. Following a brief review of the Watson pipeline and how it inspired the project, we will then discuss our system and how it attempts to emulate the human problem solving process, as well as how we used techniques and approaches present in IBM's Watson to help improve our system.

2. Background

Creativity generally, not solely in the field of artificial intelligence, is particularly prone to a high degree of subjectivity, not least in the criticism of the artistic merit of creative works but additionally regarding what "is" creativity. This latter remains the subject of intense debate and innumerable articles have been written for positions *pro* and *con*, and even "don't know". The general position is the highly regrettable but equally inevitable one of "I'll know it when I see it", a fearsomely latitudinarian standpoint as it admits and denies with equal measure, but it is currently, unfortunately, one of the few positions to take.

One of the most prominent authors on the subject of music and creativity, David Cope, defines 'creativity' as "the association of two ideas heretofore *not* considered related but now revealed as logically connected" (Cope, 2015): this definition is, arguably, equally vague, and Cope himself is more than willing to admit this. Yet it cannot be denied that we are attempting to quantify an event that is, by another argument, equally unquantifiable. As Cope continues:

> Unfortunately, my definition for creativity, as all the others I've seen, still leaves significant questions unanswered. For example, many would include cooking, exercising, even skydiving as potentially creative activities. … But one might therefore ask whether creativity represents an approach to *anything* that involves unexpected associations between two things not previously considered related.
> … Of course, the same could be said for *who* or *what* can be creative—my two cats, for example, since I've certainly seem them act in very creative ways.
> … Therefore, I underscore my definition by making sure it is understood that *any activity* or *thing* doing that activity can be creative if it 'associates two ideas heretofore not considered related but now revealed as logically connected.' (Cope, 2015)

Another slash through the Gordian knot is provided by Margaret Boden, who continues to assert the nature of creativity in humans as simply an aspect of "normal human intelligence" and something entirely capable of being duplicated by a computational system:

> Creativity isn't magical. It's an aspect of normal human intelligence, not a special faculty granted to a tiny élite. There are three forms: combinational, exploratory, and transformational. All three can be modeled by AI—in some cases, with impressive results. … Whether computers can "really" be creative isn't a scientific question but a philosophical one, to which there's no clear answer. (Boden, 2014)

Yet, while Boden dismisses one set of issues most tidily by (correctly) deeming them a matter of 'mere' philosophy, it introduces another problem: how do we know when a computer is being "creative"? According to Bringsjord, we can't: he, together with Ferrucci and Bello, argued most convincingly that computers *cannot* be *genuinely* creative, at least as far as literature is concerned (Bringsjord, Ferrucci, & Bello, 2001); that said, it is possible to engineer them *to appear to be* creative. In this paper they propose that a machine can only be *truly* creative if its function causes the generation of an artefact whose existence cannot be predicted by the designer(s) and creator(s) of such a machine: they term this the "Lovelace Test". Conversely, it's sufficient (and currently necessary) for a machine to "trick" the observers into believing that the machine is creative by making it *appear* creative, in a manner similar to the 'Turing Test.'[1]

Yet does it really matter if a machine is not genuinely creative but merely *appears* creative? To the untrained eye, the one unbiased by critical examination of an AI system its architecture and hardware infrastructure, the outcome is the same: there has been a form of creative event. The

[1] More recently, Bringsjord argues *for* the position that any general intelligence (human, alien or computational) may be considered creative. (Bringsjord, 2015)

system which we know to be 'smoke and mirrors' is believed to be capable of functioning, and believed to be functioning, in a manner similar to a human. (We recognize here the approach of the Chinese Room. (Searle, 1980))[2] We make these distinctions – between presentation and actuality; between *what is* and *what appears to be* – because we are engineers, philosophers, designers, seekers after information, and because we have not only looked behind the curtain but wrapped ourselves up in it, here in a realm where creativity is not wholly merely in the eyes of the beholders. Fundamentally, however, we believe that computers *can* be creative, just as humans *can* be creative; the *how* and *why* are still under active discussion.

But for others, denied that insight—or perhaps blessed in innocence—the situation is quite different. Let us suppose a machine could successfully be created to search a vast corpus of knowledge, including mythology, fiction, poetry, all of science, whether human, natural or applied, using natural language in response to cryptic, indirect questions and respond correctly to those questions in as short a space of time as possible, and let us further suppose that a human whose abilities were mimicked by that machine saw it in full operation, answering those questions swiftly and correctly: what might that human think? In this paper we consider this type of question specifically in the case of the system and general architecture of IBM Watson (Ferrucci et al., 2010), which famously beat two of the world's best human players of the television trivia quiz *Jeopardy!* at their own game in 2011.

2.1 'Cognitive Computing': Collaborative Intelligence

> The computer's techniques for unraveling *Jeopardy!* clues sounded just like mine. That machine zeroes in on key words in a clue, then combs its memory (in Watson's case, a 15-terabyte data bank of human knowledge) for clusters of associations with those words. It rigorously checks the top hits against all the contextual information it can muster: the category name; the kind of answer being sought; the time, place, and gender hinted at in the clue; and so on. And when it feels 'sure' enough, it decides to buzz. This is all an instant, intuitive process for a human *Jeopardy!* player, but I felt convinced that under the hood my brain was doing more or less the same thing. (Jennings, 2011)

Much has been written on the subject of IBM Watson (see e.g. (Ferrucci et al., 2010), (Devarakonda & Tsou, 2015; McMillan & Dwoskin, 2015; Sciales, Rubin, & Martialay, 2015), and not undeservedly: it is a powerful and highly flexible system. But is it creative? is it intelligent? Let us return briefly to Cope's definition: "*any activity* or *thing* doing that activity can be creative if it 'associates two ideas heretofore not considered related but now revealed as logically connected.'" Is it creative? No: it is an information retrieval system which judges possible answers based on the level of support for those answers in its knowledge base. Is it intelligent? It appears so, but that is merely the result of many hours of work on the part of its creators at IBM. And yet Watson – the version which won at *Jeopardy!* or any one of its numerous decedents, including "Watson MD" (McMillan & Dwoskin, 2015), (Devarakonda & Tsou, 2015), the version of Watson used for hydrological research at RPI as part of the Jefferson Project (Sciales et al., 2015), or even "Chef Watson" (Pinel, Varshney, & Bhattacharjya, 2015) – represents a paradigm shift in the field of AI in the manner in which it operates. 'Cognitive

2 It is interesting to note that a counter-argument appears immediately following Searle's original article. "First of all, it is no trivial matter to write rules to transform the 'Chinese symbols' of a story text into the 'Chinese symbols' of appropriate answers to questions about the story. To dismiss this programming feat as mere rule mongering is like downgrading a good piece of literature as something that British Museum monkeys can eventually produce. … Searle's argument itself, sallying forth as it does into a symbol-laden domain that is intrinsically difficult to 'understand', could well be seen as mere symbol manipulation. His main rule is that if you see the Chinese symbols for 'formal computational operations,' then you output the Chinese symbols for 'no understanding at all'." (Abelson, 1980)

computing', it has come to seem, is the way forward, and every day, it seems, brings new applications for IBM's technological brainchild.

Yet beneath all the hype surrounding Watson and its ilk there is indeed some substance to be had. The design of such systems would seem to mimic, after a limited fashion, the method of operation of the human cognitive process, and makes (more) possible some of the concepts introduced by Marvin Minsky in his concept of the "Society of Mind" (Minsky, 1988). We do not presume to suggest that all the issues we have touched so lightly upon here might be resolved, quite literally, by some *deus ex machina* event; we do suggest, however, that such systems might offer new possibilities and new insights in this most intractable and impenetrable of fields.

Interestingly, a recent paper (S. L. Epstein, 2015) offers such a new possibility. Taking the position that AI has come a long way from its 1950s roots, from a focus on implementation at the first IJCAI in 1969 to a more anthropocentric, learning- and modeling-driven approach in current times. "The dramatic changes between 1969 and 2013 were driven, I believe, by our collective fascination with hard problems," Epstein writes, continuing:

> To solve these problems, AI researchers developed a diverse set of representations to model the real world for computers. … Then, to harness these representations, AI researchers built inference mechanisms and search algorithms intended to manipulate that knowledge. … Competition, along with common and exacting evaluation metrics, allows us to see which methods perform best on which data. … As a result, AI's standard for success has become the ability of one system, algorithm, architecture, representation or approach to outperform another. Clearly, we are in search of the best machine intelligence we can construct, without any regard to what people can do. Meanwhile, this clever problem solving has had some unanticipated results.

The "unanticipated results" to which Epstein alludes she then goes on to define as AI as being perceived in the mainstream media as either "flashy but failure-ridden devices" or something to be feared. Epstein's paper attempts to rebuff both perceptions by positing the concept of the "collaborative intelligence", a system which "partners with a person to achieve the person's goals". Epstein specifies the following requirements inherent in a "collaborative intelligence":

- It asks for help when required
- It must be able to model the human view of the world
- It must engage in dialogue with its human partner(s)
- It must be able to perceive and recognize shared vocabulary and shared context
- It can signal its internal state to the user(s), improving transparency and functioning
- It should be aware that a human is very different from a machine

One might believe, based on the above, that Epstein is describing a generally intelligent system, but this is not so; quite the contrary, indeed. "As envisioned here, a CI is not a general intelligence. Each CI would target problem areas in which it could assist people, and provide representations and procedures to support particular human activities." Indeed, the seeds of this concept have already been planted by IBM in their Cognitive Environments Laboratory where numerous smaller subsystems operate synergistically around Watson to provide support to human operators. (IBM, 2015) One could imagine a future time in which the world is crowded with these single-function CI systems, each performing operations on a general pool of data, each returning its analyses to another CI or to a human, each doing its own task, great or small, as part of a much greater whole — such as understanding language, searching world knowledge, comparing it to available information and finding solutions to word puzzles. And, from there, from such a collaborative collection of CIs, it is a small step to imagine the entire supersystem as one large electronic "brain", and an unplanned, unintended, organically-developed practical experiment in artificial general intelligence.

3. Towards a 'Cognitive' Approach

In this section we first outline the format of a syllacrostic puzzle for those unfamiliar with the structure. We then briefly outline and explore our inspirations for the design of our prototype, 'cognitive' syllacrostic solving program SyllaBub, which we will describe later.

3.1 Word Puzzles and Problem-Solving

There are numerous types of word puzzles: popular variants include word searches, crosswords, word placement puzzles and acrostics. Such games can be as simple as a child's word search, easily solvable by both humans and computers alike, or as difficult as a highly abstruse cryptic crossword with no indications for *Across* or *Down*, which are hard for humans to solve and almost impossible for computers. They are a perennially fascinating space for AI work, combining as they do elements of natural language processing, exhaustive search, pattern matching, general problem-solving and, in some cases, planning.

In keeping with this, there is a rich heritage of artificial intelligence solutions in this field. Among the most notable is PROVERB, an agent capable of solving the *New York Times* crossword (Keim et al., 1999; Littman, Keim, & Shazeer, 2002; Shazeer, Littman, & Keim, 1999). (Semeraro, Lops, De Gemmis, & Basile, 2012) addresses another type of word game in which a given word must be discovered by using associations with clue words: for instance, the clues *sin*, *doctor*, *Newton*, *pie* and *New York* would yield the answer *apple*. This system uses a complex "knowledge infusion" system, which attempts to provide the system with "a deeper understanding of the information it deals with." Further, though perhaps moving away from the strict definition of the 'word puzzle' towards a more general-knowledge problem but still operating very much within the field of words and natural language processing, a recently developed system is capable of playing the popular television quiz *Who Wants To Be A Millionaire?* (Molino, Lops, Semeraro, de Gemmis, & Basile, 2015) And then, of course, there is Watson. (Murdock, 2012)

3.2 The Syllacrostic

In selecting a type of puzzle to address we were all too keenly aware of the scope of the problem. While we suggest that the general approach described in this paper might be applied to many different kinds of word puzzles, we felt it was important to show it could properly handle some specific word puzzle type. We settled, then, on attempting to solve syllacrostics as they are representative of many types of word puzzles but are sufficiently simple that we could rigorously explore the solution space and demonstrate clearly how the architecture works.

Syllacrostics consist of a clue to a target word, plus some constraints that the answer must meet: the number of syllables and the number of letters. In these puzzles, a player is given two pieces of information: the clue itself, and a pool of usable syllables: target words must be correct for the clue and be derived using only syllables remaining in the pool. An example of a syllacrostic game and its solution (in parentheses to the right) is shown in Figure 1.

Syllable List

AN AP BLE CAR CES DEN DER DRIV ER FRIC GAR HU IM IST KIN MOR NA NENT
NI OR PA PER POS PREG RA RI SA SI TE TEN TI TILE TION TRY TUS TY VAL VER

1. Heat generating force _ _ _ _ _ _ _ _ (2)	(Friction)
2. Traveling show _ _ _ _ _ _ _ _ (3)	(Carnival)
3. Relatives _ _ _ _ _ _ _ _ _ (3)	(Ancestry)
4. Solidity _ _ _ _ _ _ _ (3)	(Density)
5. Golf club _ _ _ _ _ _ (2)	(Driver)
6. Comedian _ _ _ _ _ _ _ _ (3)	(Humorist)
7. Invincible _ _ _ _ _ _ _ _ _ _ (4)	(Impregnable)
8. School for young children _ _ _ _ _ _ _ _ _ _ _ _ (4)	(Kindergarten)
9. Rear _ _ _ _ _ _ _ _ _ (4)	(Posterior)
10. Equipment _ _ _ _ _ _ _ _ (4)	(Apparatus)
11. Having Many Skills _ _ _ _ _ _ _ _ _ (3)	(Versatile)
12. Relates to the topic _ _ _ _ _ _ _ _ _ (3)	(Pertinent)

Figure 1 Example of a syllacrostic game and its solution.

Each clue provides three different pieces of information: a word or phrase, referring to and indicating the correct answer; a group of blank spaces showing the number of letters in the correct answer, and, lastly, a number in parentheses which shows the number of syllables in the correct answer. The player successfully completes the puzzle by identifying the correct answer, with the correct number of letters and syllables (from the syllable list), for each clue.

Two different forms of the syllacrostic puzzle exist: in one, the use of a syllable removes it from the pool of available syllables; in the other, syllables may be reused as desired. SyllaBub can solve both kinds of syllacrostic, providing the difference is made known before a solution is attempted.

3.3 The Human Factor

As much as we drew inspiration from technical sources, including IBM Watson and Ellis's work on cognitive computing for game AI (Ellis, 2014), significant insight also came from informal observation of several human players attempt to solve syllacrostic puzzles and observing their methodologies. Following several short sessions in which we asked people to solve a syllacrostic puzzle, we recognized two discrete stages generally used by most of the human players: "creative" and "brute-force" candidate answer generation. These two stages formed the core of the model which informs this work.

In the first stage, "creative candidate answer generation", human players appear to take a broader view of the game, where creativity and intuition come into play informed by world knowledge. At this stage, they tend to consider a clue and generate a list of possible words that work as a potential solution. These are then confirmed or rejected based on the constraints of the puzzle (i.e. if the word has the correct number of letters and syllables and if the word can be constructed from the list of available syllables present in the syllable pool). A human player will generally do this for every clue in some sequence, solving obvious clues and leaving others for later. Once the human player has solved all the clues which are immediately obvious they will either repeat this stage until the game is complete or no solutions have been found in a certain number of passes, or move on to the next.

In this second stage the human player will generally attempt to solve the remaining candidate answers by "brute force": that is, taking likely combinations of available syllables and fitting them together in an attempt to create a rational-sounding word. This stage is often entered when three or fewer clues remain, or when the player "blanks on", or is unable to immediately recall, a particular word despite 'knowing that they know it'. Often, the derivation of a word through brute force is sufficient to recall it to the player's memory as the correct answer. Finally, after entering the second stage for the first time and solving a clue, human players often begin to alternate between the two stages, solving a clue and then looking at the syllable box to see if any new words have become obvious.

SyllaBub reflects the basic ideas of the above processes well, designed as it is upon the linear flow model and world knowledge-based evaluation approach used by 'cognitive computing' systems such as IBM Watson.

3.4 'Cognitive Computing'

The Watson system is a "deep question-answering", or "DeepQA", system developed by IBM initially as a tool to play the popular television quiz *Jeopardy!* in 2011. Since that time, Watson has been adapted to fill several roles, including the fields of medicine (Devarakonda & Tsou, 2015), law (Ashley & Walker, 2013) and also *haute cuisine* (Pinel et al., 2015). While the platform has been updated and the software amended to support the different fields of operation, Watson's overall design and method of operation have remained the same. It was this design that served as the basis for SyllaBub.

Watson is a software architecture which performs content analysis and evidence-based reasoning by unifying several different algorithms for the purposes of answering questions. It operates in several stages, as shown in Figure 2. The process begins in the Question Analysis phase: here, the query is examined to draw inferences about the answer, such as the *type* of entity being sought (person, place, object, etc.) and any information about that entity provided by the question. All of the data, including the query itself, then form input queries to the Primary Search, which walks a knowledge base to generate a set of *candidate answers*. Candidates are then scored independently of each other, initially without reference to the context of the question: Context-Independent Scoring includes the type relevance of the candidate and whether the candidate exists or existed in an appropriate time period. The Soft Filtering stage eliminates any obviously irrelevant or incorrect candidate answers from the pool.

To determine the accuracy of each candidate answer Watson performs an evaluation on the candidates by inserting each of them into the question in the place of the subject (i.e. replacing "who" or "this") and creating a valid declarative statement, then comparing that statement against the documents retrieved during the Primary Search phase. (It is thus important to note that Watson does not score a candidate answer so much as score the evidence which supports that answer.) Lastly, all of the candidates are combined into one pool for ranking and evaluation using its trained machine learning component. At this point Watson selects the candidate with the highest score as its chosen answer to the question. (For more full information on DeepQA, and Watson in particular, we refer readers particularly to (Ferrucci et al., 2010) and (Murdock, 2012).)

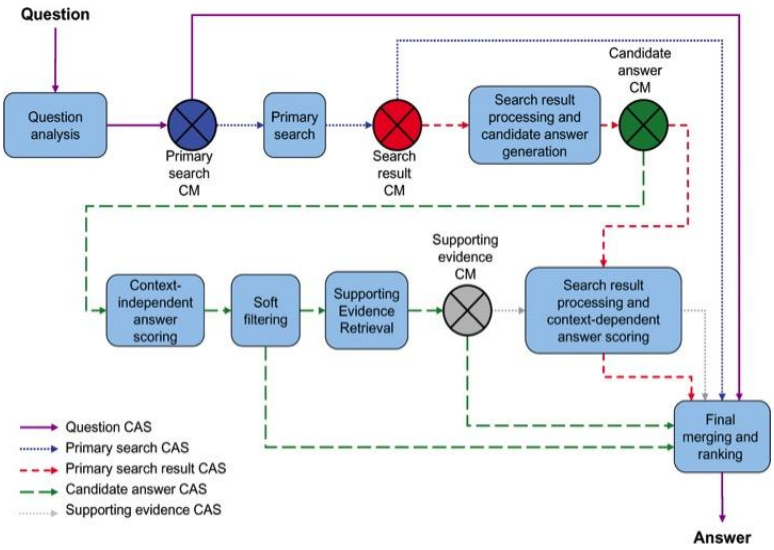

Figure 2 The Watson DeepQA pipeline (E. A. Epstein et al., 2012)

Key to Watson's success, and a major factor in our research, is its use of world knowledge to approach problems, and its use of multiple individual scorers to evaluate that knowledge in various ways before being combined into a single final response. Analysis of the Watson cognitive computing architecture has shown that it functions well when there are many data sources upon which to base a hypothesis and from which to draw candidates. Our work was motivated to see if a similar architecture would perform in situations with limited data and an extremely small scope: in other words, can Watson "scale down" to simpler problems?

4. SyllaBub, a Cognitive Computing-Inspired Syllacrostic Solver

Drawing on inspiration from the sources described above, we developed a prototype system, to solve syllacrostic puzzles, SyllaBub. In this section we describe the architecture of SyllaBub together with some of its functional details.

4.1 Operational Pipeline

SyllaBub has two phases of operation which repeat themselves as required as new information about the game becomes available; an overview of this process may be seen in Figure 3.

The first stage uses a form of "creative" candidate answer generation. The specifics of the candidate answer generation process for this stage will be explained in Section 4.2. We then constrain the candidate answers based on the information acquired from the clue. Attributes such as letter count, syllable count, and whether or not the syllables are available, are all taken into account. At this point, SyllaBub takes the remaining candidates and scores them based on several factors (see Section 4.4) before it ranks the candidates based on the scores and selects the one with the highest rank and score as the solution. This process is repeated for all clues until "creative" candidate answer generation has been attempted for each clue. It should be noted that this phase may result in incorrect solutions or no solutions at all. If there are clues that remain unanswered or that were answered incorrectly, the system moves on to the second stage.

The second stage instead uses brute force candidate answer generation. If the syllacrostic is one where syllables can be reused, this stage and subsequent stages are skipped as there is no additional information that is revealed for each clue. In this stage, we consider all of the remaining syllables that were not removed by use in previously solved clues and attempt to generate words by exhaustively combining these syllables. This generates a significant number of candidates that are constrained by the same attributes as in the previous stage, but additionally by the validity of the word in the English language. At this point we are left with a set of candidate answers that are now potential solutions to the clue. These solutions are then scored by the same scorers used in the previous stage. The candidates are then ranked and the clue with the highest score is selected. This stage can also result in incorrect solutions or no solutions. If this is the case, the system will return to the first stage.

This new repetition of the first stage takes whatever new information was extracted from the second stage and returns to the clues that are still unanswered or were answered incorrectly. This process is repeated until a pass of all the clues is completed and no new information is uncovered. If at any point SyllaBub correctly solves all of the clues the system halts and the outcome is reported.[3]

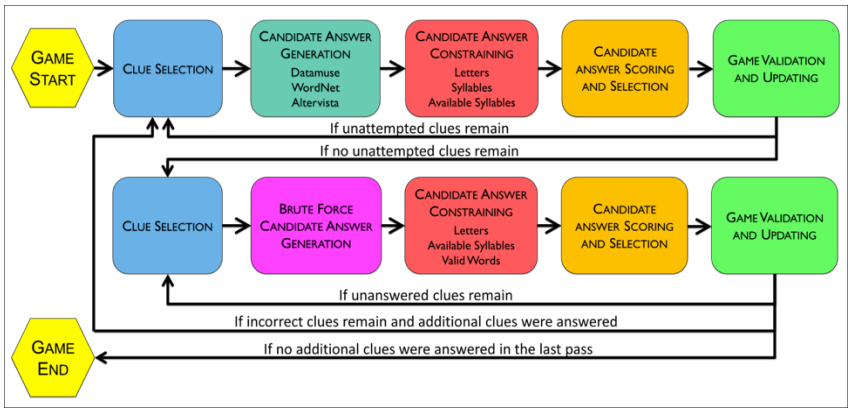

Figure 3 SyllaBub conceptual architecture

4.2 Notation

During the development of SyllaBub we developed a simple notation to describe the various components of the syllacrostic game and the various subcomponents of the solution system, as given below. This notation will be used in the sections discussing the operation of SyllaBub.

- Θ A syllacrostic puzzle represented as a set; $\Theta = \{\Psi, \Omega, \Phi, X\}$
- Ψ The set of all clues in puzzle Θ; $\Psi = \{ \Psi_1, \Psi_2,\ldots, \Psi_n \}$
- Ω The set of all answers in puzzle Θ; $\Omega = \{ \Omega_1, \Omega_2,\ldots, \Omega_n \}$
- Φ The set of all available syllables in puzzle Θ; $\Phi = \{ \Phi_1, \Phi_2,\ldots, \Phi_n \}$
- X The set of all unavailable syllables in puzzle Θ; $X = \{ X_1, X_2,\ldots, X_n \}$

- ψ A single clue, represented as the triple <clue text, number of letters, number of syllables>; $\psi = \Psi_i$ (for i in {1, …, n})

[3] Electronic versions of most puzzle games, including syllacrostics, generally provide immediate right/wrong feedback to a user during play. We used this approach as the validation of the cognitive computing architecture was not dependent on at which point such feedback was provided to the system.

  - ψ[text] The text associated with a specific ψ
  - ψ[letters] The number of letters associated with a specific ψ
  - ψ[syllables] The number of syllables associated with a specific ψ
- ω The correct answer to a clue ψ; $\omega = \Omega_i$ (for i in {1, …, n})
- φ A single available syllable; $\varphi = \Phi_i$ (for i in {1, …, n})
- χ A single unavailable syllable; $\chi = X_i$ (for i in {1, …, n})

The function of SyllaBub may be described using the following terms:

- δ A single candidate answer for clue ψ
- Δ The set of all δ for clue ψ; $\delta = \Delta_n$
- γ The set of scores for candidate answer δ
- Γ The set of all γ for all candidate answers; $\gamma = \Gamma_n$
- γ_i The value returned by a particular scorer
- τ A single scorer that can be applied to any δ
- T The set of all scorers that can be applied to any δ; $\tau = T_n$

The following constraints apply to the above definitions:

- For any $\delta = \Delta_n$ and any $\gamma = \Gamma_n$, γ contains the set of scores for δ.
- $T_i(\Delta_n) = \Gamma_{n_i} \equiv \tau(\delta) = \gamma_i$

4.3 Candidate Answer Generation

As mentioned earlier, in an attempt to emulate the human problem-solving approach SyllaBub uses subsystems to generate possible answers to clues, which we examine here in more detail.

4.3.1 "CREATIVE" CANDIDATE ANSWER GENERATION

The "creative" candidate answer generation process has two phases: a clue analysis phase, in which important keywords are selected from each ψ[text], and a candidate generation phase, using the keywords obtained from clue analysis.

We parse the contents of ψ[text] using the NLTK part of speech (POS) tagger, which uses the Penn State Treebank Tagset (Bird, Loper, & Klein, 2009). For the purposes of SyllaBub, we define keywords as words that are not tagged as "DT", "TO", "CC", "RP", "PRP" or "PRP$". These tags correspond to the following parts-of-speech: "DT" or a determiner, "TO" or the word "to", "CC" or a coordinating conjunction, "RP" or a particle, "PRP" or a personal pronoun, and "PRP$" or a possessive pronoun.

After selecting these keywords SyllaBub continues by attempting to find as many synonyms for these words as possible using the WordNet API (Princeton University, 2010) and the Altervista (Altervista, n.d.) online thesaurus service. WordNet is queried for synonyms and antonyms, as well as synonyms of those synonyms.

Subsequently, this process performs a reverse dictionary lookup using the online Datamuse API (Datamuse, n.d.), which takes a word's definition as a string and returns a number of words that might fit the given definition. ψ[text] is passed to the API in its entirety and any responses captured.

The results of these searches are stored in Δ. Note that it is neither guaranteed nor expected that any $\delta \in \Delta$ will be unique.

4.3.2 "BRUTE FORCE" CANDIDATE ANSWER GENERATION

This subsystem uses the contents of Φ to iteratively form candidates δ using an exhaustive combinatorial search and returns all of the potential permutations of $\varphi \in \Phi$. It terminates when $letter\_count(\delta) \geq \psi$[letters].

4.4 Candidate Answer Constraining

The constraints applied to all δ are largely similar in both phases of the pipeline:

- if δ is an empty string, discard δ
- if $letter\_count(\delta) \neq \psi$[letters], discard δ
- if $syllable\_count(\delta) \neq \psi$[syllables], discard δ
- if a syllable $\delta \notin \Phi$, discard δ

An additional constraint is added for all δ generated in the "brute force" phase: that a WordNet definition can be found for such words. Any δ which fails this test is discarded. This serves to eliminate the number of nonsense words passed to the scoring system.

4.5 Candidate Answer Scoring

The scoring process for all δ is the same regardless of the candidate answer generation phase which generated it. Each δ is processed by five discrete scorers τ, as follows:

- T_{wn}, which uses WordNet
- T_{dm}, the DataMuse scorer
- T_{av}, which generates a score based on the Altervista response
- T_{app}, the appearance scorer
- T_{pos}, the part-of-speech scorer

The first two scorers are known collectively as "the 'black box' scorers" as they use scores generated by external systems, while the others are based on custom processes we designed.

4.5.1 THE "BLACK BOX" SCORERS

We use WordNet to generate the "path similarity" score. This value is calculated remotely by the WordNet server, and is based on the 'fitness' of a δ within a framework of hypernym and hyponym relationships determined by comparing δ with all words in ψ[text]. WordNet returns a normalized value such that $T_{wn}(\delta) = 0 \leq \gamma_{wn} \leq 1$.

The value of the Datamuse scorer is also determined remotely. It represents how closely a passed definition correlates with a given response. The value returned, γ_{dm}, is a real number with an arbitrary range; these values are then normalized based on $\sum\gamma_{dm}, \gamma \in \Gamma$.

4.5.2 ALTERVISTA SCORER

This scorer is very simple and determines if a δ was generated using the Altervista thesaurus service: $T_{av}(\delta) = 1$ if so, else $T_{av}(\delta) = 0$. This scorer was introduced to add additional weight to any δ received from and track how δ were being generated.

4.5.3 APPEARANCE SCORER

The function of this scorer is to weight any δ based on the number of times it has appeared as a candidate in Δ. This is useful because, although it is not always so, it is generally seen that the more times a word appears in a search the more likely it is to be the correct answer. Once γ_{app} is calculated for all δ, the values are normalized based on the maximum appearance score in Γ. The equation that governs the score returned by this scorer is presented in Equation 1; $n(\delta)$ is the number of occurrences of δ in Δ.

$$\gamma_{app} = \frac{\mathrm{n}(\delta)}{\max\left(\Gamma_{0_{app}\ldots i_{app}}\right)}$$

Equation 1 Candidate appearance score

4.5.4 PART OF SPEECH SCORER

The final custom scorer, γ_{pos}, uses the part of speech (POS) of a word. It consists of two components. In the first, all of the words in ψ[text] are tagged using the NLTK parser. The POS of δ is compared to the POS of each word ψ[text], and $T_{pos}(\delta)$ is incremented by the proportion of the particular POS in ψ[text]. In the second, the scorer attempts to match the clue to a set of predetermined clue skeletons where the expected POS is already known; for example, a clue skeleton would be "To *" or "*ly", where "*" is a wildcard. In this instance, we can assume that correct δ is a verb or adverb, respectively, based on the structure ψ[text]. For any ψ[text] that matches these skeletons to the POS of δ, $T_{pos}(\delta)$ is incremented by 1.0.

As an example, consider the clue ψ with ψ[text] = "Tall flowery plant". This clue consists of a phrase of two adjectives and a noun. Given δ is a noun, $T_{pos}(\delta)$ would initially yield $\gamma_{pos} = 0.333$ as the POS of δ matched the POS of one of the three words in ψ[text]. ψ[text] would then be examined further to determine if it matched any of the predefined clue skeletons. If there was a match, then the score would be increased by 1.0 for every match. The score would then be normalized following the calculation of γ_{pos} for all δ.

4.5.5 FINAL SCORE CALCULATION

Once all scores have been calculated for all δ, the scores are normalized between zero and one, summed and averaged across Γ. This score, which we will call $Final(\delta)$, is the deciding score to be computed for a δ. This is done for all δ, and at the end of the process all δ are ranked, and the δ with the highest $Final(\delta)$ is selected as the solution. This δ is then returned to the syllacrostic game for validation. The equation for calculating final scores is shown in Equation 2.

$$Final(\delta) = \frac{\sum_{i=0}^{m} \gamma_i}{m} \; (0 \leq \gamma_i \leq 1)$$

Equation 2 Final score

5. Results and Discussion

In this section we present SyllaBub's performance in terms of accuracy and time and then outline some of its strengths and weaknesses. Results were obtained using a system with an Intel Core i5-

3360M 2.80 GHz processor and 8 GB of RAM. All tests were performed with minimal additional system load. The test battery used a bank of 374 clues across 32 distinct games drawn from various different sources. SyllaBub was programmed in Python.

5.1 Accuracy and Performance

The current version of SyllaBub answers questions with an accuracy of 95.187%. A breakdown of this value is presented in Table 1. The "creative" candidate answering process solves roughly 80% of the clues that are presented SyllaBub; of the remaining clues, 14.706% are solved by the brute force candidate answering process, while only a relatively small number remain unanswered.

	Number	**%**
Clues solved by the "creative" candidate answer generation process	301	80.481
Clues solved by the brute force candidate answer generation process	55	14.706
Clues unsolved by either technique	18	4.813
Total clues solved	356	95.187

Table 1 Accuracy of the system across all games

SyllaBub is able to complete all 374 clues across 32 games in 804.228 seconds, with an average clue and game solution time of 2.15s and 25.132s respectively. Additional information is included in Table 2. The vast majority of the time is spent in the "creative" candidate answer process: this is expected, as this process makes multiple requests from the internet, adding significant amounts of latency. Additional time spent outside the two candidate answering processes is predominantly spent on file access and required system 'housekeeping' operations.

Process	**Time (s)**	**Time (%)**
Time spent in the "creative" candidate answer generation process	772.05	95.999
Time spent in the brute force candidate answer generation process	30.078	3.740
Other	2.099	0.261
Total	804.228	

Table 2 Time usage of the system across all games

5.2 Strengths and Weaknesses

SyllaBub performs well on the majority of clues presented to it, and is particularly good at finding solutions to one-word clues. This may seem counter-intuitive, as it might appear less information is available to find the correct answer: but since there is only one word typically means that the answer is a synonym, meaning that the answer can be achieved by a fairly straightforward lookup.

The most difficult clues to answer are those where the clue is a more cryptic phrase. For example, the clue 'Bring on the germs' might have the answer 'contaminate': this type of clue, for which a degree of lateral thinking is required, is very difficult for SyllaBub to answer since, while there is a connection between the clue and the answer which may be evident to a human, this connection is not easily revealed by standard lexical analysis techniques.
Currently SyllaBub's most common reason for getting a clue incorrect is not that it selects the incorrect response but because the correct response is not made available by the candidate answer generation subsystem. While the brute force candidate answer generation is able to find some otherwise undiscoverable answers, SyllaBub still generally fails on certain answers requiring proper nouns and or those that require multiple word phrases.

5.3 Machine Learning

SyllaBub currently weighs all values returned from the different scorers equally in the final merging and ranking phase. Although we explored the use of a simple machine learning model, it was removed from the final model as the systems already in place provided sufficient distinction between the correct answer and the incorrect answers. While a more complicated game or problem would almost certainly benefit from the addition of a machine learning component, due to the effectiveness of the system as in its current form, the amount of overhead introduced with the addition of this component outweighed its benefit. Additonally, it was noted that the introduction of machine learning *reduced* SyllaBub's overall accuracy by 5–10%: this could be because of insufficient training data. This might also seem to suggest that there is a minimum complexity of problem for which a machine-learning component can make meaningful discriminations.

5.4 Further Improvements

We outline below several ideas which will be implemented in future versions of SyllaBub.

5.4.1 Additional Sources

SyllaBub's current largest issue is not that it selects the wrong answer but rather that it is unaware of the correct answer. Providing it with additional sources (thesauruses, encyclopedias, etc.) would increase overall accuracy, although at the cost of considerably increased processing time; the impact of this could be managed by a degree of offline preprocessing of common data as was used extensively in the IBM Watson *Jeopardy!* player.

5.4.2 Improved Scoring Algorithms

Refinement of already existing scorers and the addition of new scorers would almost certainly improve the system's performance. This could include scorers capable of addressing some of the more cryptic clues, with the intention of solving them in the initial "creative" phase and without having to resort to the "brute force" subsystem. An example of such a clue, which we encountered during testing, was "Bring on the germs," for which the answer was "Contaminate": such clues require a degree of lateral thinking and are difficult for SyllaBub to solve.

5.4.3 BACKTRACKING FOR CONFLICT RESOLUTION

As it stands, the current version of SyllaBub has the distinct advantage of being told, when it has made its selection of which of the candidate answers it prefers, whether or not that clue is correct. This gives SyllaBub a significant boost to performance: it does not have to backtrack once it has solved a clue as it can be certain that it has used the correct syllables. An extended system could use a decision tree to store attempted solutions to a clue as a node in the tree, with each decision weighted by the scores of the candidate answers. Conflict resolution would initially be predicated on the respective scores of the conflicting candidates, with the candidate with the highest probability (based on natural language models of probability and a persistent knowledge base) would be retained in this iteration.

This approach would also mitigate the problem of SyllaBub choosing incorrect syllables to construct a word. For example, a candidate answer might be "sovereign": while the correct sequence of syllables (as defined by the game) might be "['SOV', 'ER', 'EIGN']", the solver may choose to select the incorrect sequence "['SO', 'VER', 'EIGN']". This results in incorrect constraints and decisions later on in the pipeline and consequently incorrectly solved clues. This problem could be solved by simply entering the distinct syllable structure as a new potential node to be explored in a decision tree.

5.4.4 PUZZLE VARIATIONS

More complex variants of the syllacrostic require that an acrostic passphrase constructed from the first and last letters in the correct answers be used to confirm a solution. While we investigated such a scorer, we decided to focus primarily on deriving the correct answers using the clues as obtaining the correct answers would necessarily entail obtaining the correct acrostics.

6. From SyllaBub to AGI

> Concepts in the brains of humans acquired the property that they could get rolled together with other concepts into larger packets, and any such larger packet could then become a new concept in its own right. In other words, concepts could nest inside each other hierarchically, and such nesting could go on to arbitrary degrees. (Hofstadter, 2008)

It is fair to say that a syllacrostic is not a particularly complex puzzle; it is equally fair to say that it could be solved, with a certain minimum of fuss and effort, by a simple combinatoric algorithm. So by extension a fair question might be: why go to the time and trouble of building a 'cognitive computing' version at all?

SyllaBub was built as a conceptual prototype to explore the applicability of the 'cognitive computing' approach: that is, using a modular, largely linear, highly extensible architecture conceptualized upon a hypothetical model of a small subset of human mental processes which uses world knowledge as a primary input source and answer confirmation. (The final part, the ability to tune its behavior to improve its performance or accuracy, was omitted for reasons discussed above.) By extension, the same principles which can be used to solve the syllacrostic can be used to solve other problems in the domain of word puzzles: almost all of them beyond the simplest or most mechanically-based require world knowledge to complete, understanding of the input clue in some form or other, and the ability to process and manipulate data to find possible answers and, ideally, learn from its successes and failures to improve its performance.

Considering briefly a more extreme example of word puzzle, the cryptic crossword, we can see the principles of the humble syllacrostic nestling within it, the requirement for world knowledge and understanding of that knowledge, and the ability to derive the information

required to solve a clue. But that is not all. Cryptic crosswords are notoriously abstract and complex, and require considerable additional information to solve, including:

a) literal data contained represented by the text of the clue itself
b) meta-information about the type of clue (e.g. anagram, word fragment) provided by a word or phrase in the clue
c) recognising cultural references indicated by a 'theme' or 'setting' for the puzzle (e.g. summer, Christmas, poetry)
d) experience of the Setter's habits, idioms, thought processes, and, in certain cases, current circumstances

For example, the clue "Swap large numbers in protest from Lincolnshire town: no wind farm! (3, 2, 5)" requires the following chain of deduction:

a) NIMBY ("Not In My Back Yard") protests against renewable energy sources
b) Louth, a town in Lincolnshire, UK where protests were under way
c) L is 50 and M is 1,000 in Roman numerals
d) 'Swap' is a meta-instruction to the reader

Thus "NIMBY LOUTH (swapping M/1000 for L/50) became NIL BY MOUTH to someone with cancer of the oesophagus in a Cambridge hospital." (Stephenson, 2013)

It would not be true or fair to suggest that a system whose sole purpose is to solve clues to cryptic crosswords is a general intelligence, but it is fair to say that it is considerably closer to being one. Perhaps a system which is capable of *generating* cryptic crosswords, not merely solving them, might be a step closer still, requiring as it would some form of creative ability and the insight into the mind of the opponent – that is, the solvers of the crosswords it sets – which permits it to make its puzzles sufficiently challenging but still capable of being solved. If nothing else we would expect that a system which could generate cryptic crosswords would be able to solve them. In an ideal world it would be able to solve any type of crossword or word puzzle it might encounter by recognizing familiar components in unfamiliar combinations and generating, through some form of reasoning or quasi-intuitive process, a method of solving that unfamiliar problem based on its own sets of abilities. This, surely, is the basis of general intelligence in any form: the ability to of an agent (artificial or natural) to adapt as necessary to the requirements of the changing situation based on its own analysis of the situation, tuning its reaction to and interaction with the external situation based on its degree of success.[4]

7. Future Work

SyllaBub was an interesting project but ultimately self-limiting in scope. We have been begun working on a system which can solve a more advanced type of crossword puzzle, *acrostic crosswords*. In this type of crossword there are two grids: one is for answers to 'straight' (non-cryptic) clues, while the other is for a quotation which is built up from the letters in the clue grid; additionally, the name of the quotation's author is usually given by reading down the column of first letters of clues. This type of crossword makes much greater use of world knowledge, and appears to require much more complex and interesting reasoning and inference techniques.

4 For a more complete discussion of this area, see e.g., (Meehl & Sellars, 1956), (Searle, 1980), (Nikolić, 2015).

We are also interested in how well the Watson-type architecture might be suited to other AGI-related tasks. Beyond merely extending the current system as we have suggested, a more ambitious use of such an architecture could be to play other, more general types of games, particularly those whose search space is extremely large. This might be, for example, because of the combinatorial complexity of the game's mechanics, or because there exists a high degree of imperfect information in the game (or both). Such a cognitive computing-based game AI system would be novel and extremely interesting to develop, and is another focus of our future work.[5]

8. Conclusion

'Cognitive computing' is rapidly becoming a significant force in the field of computer science, with IBM's Watson system very much its vanguard. In this paper we have presented SyllaBub, an agent capable of solving simple word puzzles, designed using the 'cognitive computing' pattern and using a human-inspired model to perform its tasks. We believe that such small systems have real significance to the larger field of artificial general intelligence, representative as they are of the many types of low-level problem-solving of which an AGI must be capable.

Acknowledgements

This work was supported in part by a grant to Rensselaer Polytechnic Institute from the Walt Disney Imagineering Research and Development organization, the IBM SUR grant that provided the Watson system to Rensselaer, and the Rensselaer Endowment in its support of RPI's Tetherless World Constellation. We would also like to acknowledge and thank the reviewers for their thorough and insightful comments.

References

Abelson, R. P. (1980). Searle's argument is just a set of Chinese symbols. *Behavioral and Brain Sciences*, *3*(3), 424–425.

Altervista. (n.d.). Thesaurus web service. Retrieved November 2, 2015, from http://thesaurus.altervista.org

Ashley, K. D., & Walker, V. R. (2013). Toward constructing evidence-based legal arguments using legal decision documents and machine learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Law* (pp. 176–180). Rome, Italy: ACM, New York, NY. Retrieved from http://dl.acm.org/citation.cfm?id=2514622

Bird, S., Loper, E., & Klein, E. (2009). *Natural Language Processing with Python*. Sebastopol, CA: O'Reilly Media.

Boden, M. A. (2014). Computer models of creativity. *AI Magazine*, *30*(3).

[5] A brief outline of such a system appeared in (Ellis, 2014): note that this describes a Watson-based AI for *board games*: to our knowledge, no other such system is in development, though work has been done in other areas of game AI research, such as real-time strategy (RTS) games (Dannenhauer & Muñoz-Avila, 2013) and 'serious games'.

Bringsjord, S. (2015). Theorem: General intelligence entails creativity, assuming . . . In T. R. Besold, M. Schlorlemmer, & A. Smaill (Eds.), *Computational Creativity Research: Towards Creative Machines* (1st ed., pp. 51–63). Paris: Atlantis Press.

Bringsjord, S., Ferrucci, D., & Bello. (2001). Creativity, the Turing Test, and the (better) Lovelace Test. *Minds and Machines*, *11*, 3–27.

Cope, D. (2015). Computational Creativity and Music. In T. R. Besold, M. Schorlemmer, & A. Smaill (Eds.), *Computational Creativity Research: Towards Creative Machines* (pp. 309–326). Atlantis Press.

Dannenhauer, D., & Muñoz-Avila, H. (2013). Case-based Goal Selection Inspired by IBM's Watson. In *Twenty-First International Conference on Case-Based Reasoning (ICCBR 2013)*. Saratoga Springs, NY: ICCBR.

Datamuse. (n.d.). Datamuse API.

Devarakonda, M., & Tsou, C. (2015). Automated Problem List Generation from Electronic Medical Records in IBM Watson. In *Proceedings of the 27th Conference on Innovative Applications of Artificial Intelligence* (pp. 3942–3947). Austin, TX: AAAI.

Ellis, S. (2014). When Watson Gets Board: Cognitive computing as a basis for AI in tabletop games. *IBM Cognitive Systems Colloquium 2014*. Yorktown Heights, NY: IBM Research.

Epstein, E. A., Schor, M. I., Iyer, B. S., Lally, A., Brown, E. W., & Cwiklik, J. (2012). Making Watson fast. *IBM Journal of Research & Development*, *56*(3/4), 15:1–15:12.

Epstein, S. L. (2015). Wanted: Collaborative intelligence. *Artificial Intelligence*, *221*, 36–45. doi:10.1016/j.artint.2014.12.006

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., … Weltz, C. (2010). Building Watson: An Overview of the DeepQA Project. *AI Magazine*, (Fall), 59–79.

Hofstadter, D. (2008). *I Am a Strange Loop*. New York, NY: Basic Books.

IBM. (2015). A Symbiotic Cognitive Experience: Human-computer collaboration at the speed of thought. Retrieved November 2, 2015, from http://researcher.ibm.com/researcher/view_group.php?id=5417

Jennings, K. (2011). My Puny Human Brain. Retrieved from http://www.slate.com/articles/arts/culturebox/2011/02/my_puny_human_brain.html

Keim, G. A., Shazeer, N., Littman, M. L., Agarwal, S., Cheves, C. M., Fitzgerald, J., … Weinmeister, K. (1999). Proverb: The Probabilistic Cruciverbalist. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence* (pp. 710–717). AAAI.

Littman, M. L., Keim, G. a., & Shazeer, N. (2002). A probabilistic approach to solving crossword puzzles. *Artificial Intelligence*, *134*, 23–55. doi:10.1016/S0004-3702(01)00114-X

McMillan, R., & Dwoskin, E. (2015). IBM Crafts a Role for Artificial Intelligence in Medicine. Retrieved November 2, 2015, from http://www.wsj.com/articles/ibm-crafts-a-role-for-

artificial-intelligence-in-medicine-1439265840

Meehl, P. E., & Sellars, W. (1956). The concept of emergence. In H. Feigl & M. Scriven (Eds.), *Minnesota Studies in the Philosophy of Science, Volume I: The Foundations of Science and the Concepts of Psychology and Psychoanalysis* (pp. 239–252). Minneapolis, MN: University of Minnesota Press.

Minsky, M. (1988). The Society of Mind. Retrieved from http://aurellem.org/society-of-mind/

Molino, P., Lops, P., Semeraro, G., de Gemmis, M., & Basile, P. (2015). Playing with knowledge: A virtual player for "Who Wants to Be a Millionaire?" that leverages question answering techniques. *Artificial Intelligence*, *222*, 157–181. doi:10.1016/j.artint.2015.02.003

Murdock, J. W. (Ed.). (2012). This is Watson. *IBM Journal of Research & Development*, (3/4).

Nikolić, D. (2015). Practopoiesis: Or how life fosters a mind. *Journal of Theoretical Biology*, *373*, 40–61. doi:10.1016/j.jtbi.2015.03.003

Pinel, F., Varshney, R. R., & Bhattacharjya, D. (2015). A Culinary Computational Creativity System. In T. R. Besold, M. Schorlemmer, & A. Smaill (Eds.), *Computational Creativity Research: Towards Creative Machines* (1st ed., pp. 327–346). Atlantis Press.

Princeton University. (2010). About WordNet.

Sciales, J., Rubin, J., & Martialay, M. (2015). Internet of Things Turning New York's Lake George into "World's Smartest Lake."

Searle, J. R. (1980). Minds, Brains, and Programs. *Behavioral and Brain Sciences*, *3*(3), 417–457.

Semeraro, G., Lops, P., De Gemmis, M., & Basile, P. (2012). An Artificial Player for a Language Game. *IEEE Intelligent Systems*, *27*(5), 36–43. doi:10.1109/MIS.2011.37

Shazeer, N. M., Littman, M. L., & Keim, G. A. (1999). Crossword Puzzles as Constraint Satisfaction Problems. In *Proceedings of the 16th National Conference on Artificial Intelligence* (pp. 156–162). AAAI. Retrieved from http://www.cs.columbia.edu/~evs/ais/finalprojs/steinthal/

Stephenson, H. (2013, November 26). Araucaria: "He never dumbed down." *The Guardian*. Manchester, UK. Retrieved from http://www.theguardian.com/crosswords/2013/nov/26/araucaria-guardian-crossword-setter-puzzles