STATEMENTS

PATENT-BOT

James Sham, The Corcoran School of the Arts and Design, The George Washington University, Washington, DC 20052, U.S.A. Email: mrsham@gwu.edu.

Patrick Killoran, 3526 82nd Street, Unit 44, Jackson Heights, NY 11372, U.S.A. Email: killoranstudio@gmail.com.

Neil Rubens, Transport and Telecommunications Institute, Riga, LV-1019, Latvia. Email: neil@hrstc.org.

Brian A. Korgel, McKetta Department of Chemical Engineering and Texas Materials Institute, The University of Texas at Austin, Austin, TX 78712, U.S.A. Email: korgel@che.utexas.edu.

See www.mitpressjournals.org/toc/leon/52/3 for supplemental files associated with this issue.

Submitted: 12 April 2018

Abstract

Patent-Bot is an artificial intelligence (AI) software program that learns language from the patent database to write original patents for submission to the United States Patent Office (USPTO). The program creates thousands of new patent summaries per second. Patent-Bot is itself a piece of intellectual property, which in turn exists to generate more intellectual property. Patent-Bot also invents new words in relation to its future concepts, which appear to test the current linguistic limits of innovation and communication. Patent-Bot was debuted as an interactive art installation in the Omnibus Filing exhibition at the Visual Arts Center, University of Texas at Austin. Omnibus Filing showcased artworks, inventions, prototypes and cross-disciplinary research projects undertaken by teams of scientists, artists and engineers. Patent-Bot has since exhibited at Piksel 17: A festival for Elektronisk Kansk og fri Teknologi, in Lydgalleriet, Bergen, Norway. The project is an ongoing collaboration between the authors spanning a variety of exhibition formats and modes of display.

Patent-Bot was developed in response to questions of how technology might change the concept of innovation. The project initially began as a conceptual exercise: Imagine a patent that would generate patents. A character-based recurrent neural network (char-RNN) was then tasked to write original ideas in patent form. One well-known goal in AI research is to convincingly approximate human communication [1], and the formulaic language of the US patent archive seemed to offer a convincingly approximate human communication [1], and the formulaic language of the US patent archive seemed to offer a high likelihood that Patent-Bot would be able to learn from the existing patent database to express intelligible ideas. Indeed, compare these two patent summaries:

A) “In some embodiments, the triangular flange construction section extends from the bottom end to another before the lower grip legs separate collars from the fixed beams with the arrow tube end from the positioning members in the best position.”

B) “A laser alignment target includes a disk shaped and transparent target body with an annular inclined mounting surface, a back plate having an annular inclined retaining surface, and a toroidal alignment member retained in a generally V-shaped groove formed by the mounting and retaining surfaces.”

Patent-Bot wrote the first summary and the second exists in the USPTO database [2]. They are similar in tone, form and diction, and are perhaps as equally descriptive as they are perplexing.

Patent-Bot, shown in Fig. 1, is primed to invent by entering a seed word or phrase into the interface along with an adjustable “conservatism” parameter, or temperature scale. More creative results are obtained with lower conservatism [3]. Character-based RNNs generate text by analyzing preceding characters and generating the next most likely character based on the text corpus on which the model is based [4]. As an example, the next character following (c,a,_) could be cat or cap. The parameter defining the number of preceding characters considered in the model further improves the cogency of the text, e.g. “baseball cap.” This procedure is repeated to generate the output. The conservativeness parameter defines the chance of “less likely” characters candidates being used, e.g. caz. Priming/seeding with words alters the context of the model; for example, seeding with “animal” or “dog” would increase the probability of generating cat rather than cap.

Fig. 1. Patent-Bot: An artificial intelligence computer program that creates new patent ideas, 2017. (© Patrick Killoran)

One unexpected outcome was that Patent-Bot generates convincing neologisms, which have all the appearance of meaning without actually being in the English language or in the USPTO database. Character RNNs have been used to generate a variety of other kinds of texts, including haiku poetry [5], Shakespearean sonnets [6], jokes [7] and Celtic folk music [8]. Compared to word-based RNNs, char-RNNs are more prone to spelling errors [9]. Word-based models also tend to have more coherent sentences due to the length of the sequence (# of words vs # of sequences) [10]. In the case of Patent-Bot, the lower memory requirement and faster inference speed were the reasons to choose a character-based rather than word-based RNN, as the “vocabulary” of character-based models (letters and special symbols) is significantly smaller than word-based models that typically have a corpus of hundreds of thousands of words [11]. As a character-based AI, Patent-Bot does not know the difference between what is accepted as a word and what is not, and in this case the AI learns the rules of word construction via database analysis within the fuzzy parameters of patent diction. From a creative outlook, “errors” may actually be desirable, which counters the general perspective in machine learning research that errors should be minimized (ideally to zero) [12].

Patent-Bot appears to invent words as containers for future ideas, words that currently are not assigned meaning but have the form of meaning. For instance, Patent-Bot created the word incessive to describe the type of flow encountered in a display device it invented, as in “the flow mode is incessive in conjunction with a . . . .” Incessive sounds like a word; however, Patent-Bot created this word independently—it does not appear in the patent database and is not accepted in the Oxford English Dictionary or Merriam-Webster’s Collegiate Dictionary. Yet there are people who have used it. President George H.W. Bush wrote about China’s “incesive propaganda” [13], and the US House of Representatives Subcommittee on Asian and Pacific Affairs on Taiwan wrote in one of its reports,

290 LEONARDO, Vol. 52, No. 3, pp. 290–291, 2018
doi:10.1162/LEON_a_01719 ©2019 ISAST

Downloaded from http://direct.mit.edu/leon/article-pdf/52/3/290/1579104/leon_a_01719.pdf by guest on 13 July 2021
“Because of its incessive conflicts with warlords and the Japanese, the Nationalist Party had to adopt a Soviet-type model” [14]. Although it may be a misspelling in some cases—of the word incessant—there is a Wiktionary entry that provides a unique definition of incessive: “intense and active,” deriving from the Latin root incessus + -ive. And this is the way many people will use the word. The Wiktionary page goes on to list examples of the term’s use in a variety of texts.

Fig. 2. Patent-Bot approximates new words to describe future ideas based on the formal parameters of the English language, 2017. (© Brian A Korgel)

So, how does Patent-Bot “know” or sense the meaning of incessive? How does the program sense the meaning of the hundreds of other words it generated, like stereosonic, messhobous, carger, translucence, electrolysisitc, elongatement, moveminal, vertebral, chemistration and bundalancy, to name a few? Patent-Bot also has an appropriate sense of linguistic categories, creating and using new words as nouns, verbs adjectives and adverbs.

For generating patent ideas, the ability to create new and unexpected concepts seems to be fundamentally tied to the freedom to generate terminology and descriptive language. Along these lines, Patent-Bot conceived of the term photodiothings. A photodiode is used in solar cells and optical detectors. Patent-Bot utilized a partial representation of this word and added the childlike catchall thing to encompass a sweeping range of new ideas. The words it creates also reflect trends in the USPTO database, as in this case, for photodiode-related technologies. It seems that it would be a broad piece of intellectual property to own the patent claim to photodiothings.

Patent-Bot seems to fill lingual gaps with words that very easily could be words but have not yet been used or invented. Ultimately, however, a human reader must assess to what degree the new word accomplishes the task of a potential new word. In this respect, Patent-Bot highlights how knowledge production is relational. The AI must share the creative act with a human subject. As a specific illustration of this kind of collaborative process, many of these new words became the subject of another artwork in which the words were visually presented as text-based wall objects to be read along with audio-based dialogues that used the new words and highlighted their potential utility (Fig. 2) [15,16]. This emergent-language immersion revealed the collaborative exchange of ideas at work between human and AI within the Patent-Bot framework.

While the original intention behind Patent-Bot was to replicate and automate some form of patentable creativity as a conceptual exercise, the actual results have presented philosophic challenges to our notions of creativity codification within the parameters of patent law. At the crux of it is a chicken-or-egg paradox about language: What comes first, the new concept, or the new language to describe the concept? Patent-Bot recognizes statistical patterns in the language used in the patent database and generates text with only the 26 letters of the alphabet, spaces and punctuation as its basis. It creates a plethora of new words. With the wide use of bots to speed communication between people, one wonders if AI will spark a significant increase in the rate of language evolution. Patent-Bot is (at least) a step ahead, producing words that will likely contain concepts and ideas that have yet to be invented.

References and Notes

4. Russell and Norvig [3]
10. Graves [9].
11. Graves [9].