D. Lopresti, G. Nagy, Adapting the Turing Test for Declaring Document Analysis Problems Solved, Procs. DAS02, Gold Coast, Australia, Feb. 2012.

Adapting the Turing Test for Declaring Document Analysis Problems Solved

Daniel Lopresti
CSE Department
Lehigh University
Bethlehem, PA 18055, USA
Email: lopresti@cse.lehigh.edu

George Nagy
ECSE Department
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
Email: nagy@ecse.rpi.edu

***Abstract*—We propose to adapt Turing's seminal 1950 test for machine intelligence to evaluating progress in document analysis systems. Our premise is that a problem can be considered solved if automated and human solutions to the underlying task are indistinguishable to a skeptical human judge. For the domain-specific problems of concern here, we reformulate the test to keep the interaction between judges and human/machine participants to graphical user interfaces that do not require natural language processing, a notable difference from Turing's original formulation. Examples of tasks that may lend themselves to such tests include detecting or identifying specific document components such as logos, photographs, tables, as well as writer and language identification. The administration of the test would be facilitated by commercial crowd-sourcing systems such as Amazon Mechanical Turk, as well as research platforms such as the Lehigh Document Analysis Engine (DAE) that accept arbitrary documents for input, record test results, and provide for trusted execution of submitted programs.**

***Keywords*-digital image analysis; evaluation; Turing Test; imitation game;**

I. Background

In a recent ICDAR paper, we observed that open problems are defined differently in document image analysis than is customary in the physical sciences, theoretical computer science, or mathematics [1]. Problems in DIA are stated in terms of automation of an application area (*e.g.*, postal address reading) or a scientific subfield (*e.g.*, image compression). We suggested that the notion of a successful solution may be based on (1) the relative accuracy of automated vs. expert solutions (given specific data and degree of manual tuning); (2) the distinguishability of automated output from human output (a Turing Test); (3) the degree of current community interest (via conferences and journals); and/or (4) economic considerations. Because interest in automating certain tasks has been evolving rapidly, reaching a consensus on these issues may accelerate progress and symbiosis with allied disciplines.

Here we focus on the idea of adapting Turing's seminal 1950 test for machine intelligence [2] to the task of deciding when a document analysis problem can be declared "solved." A Turing (or Turing-like) Test offers intuitive appeal. Document analysis is a subfield of artificial intelligence, and the Turing Test – despite some issues – has survived as an inspirational (if rarely applied) measure of when a machine can be deemed intelligent. In document image analysis, we strive to produce algorithms that match what a human would do when faced with the same input – *i.e.*, the two outputs are indistinguishable, which can be viewed as the same underlying thesis behind the Test. In contrast to automated methods that attempt to quantify the similarity between complex output representations (many of which result in computational problems that are NP-complete), the Turing Test yields a simple statistical evaluation. While a number of criticisms have been levied against the Test, for our intended purpose the primary concern is that it appears impractical to take Turing's philosophical vision and make it operative in a DIA context. The goal of this paper is to explore this idea further and to lay the groundwork for overcoming the technical issues. Our hope is that the test we propose can serve as a practical determination of when a DIA problem has been solved.

To briefly summarize the main points from Turing's original paper, the test (which he called the "imitation game") involves locking a human player in one room and a machine player in another. A human judge queries both players with a series of questions, evaluating their answers and asking new questions based on the results of previous questions. At some point, when enough evidence has been collected to establish a degree of confidence, the judge is permitted to declare which player is human and which is machine. If the judge can do no better than random chance over a number of trials, the machine is said to have passed the Turing Test. An important implicit assumption is that all entities are attempting to do their best: the human player to appear human, the machine player to appear human, and the judge to accurately distinguish the two. *I.e.*, it is not sufficient to fool an inept judge, and employing an unqualified (or adversarial) human player is similarly pointless.

Turing's test is conversational and heavily dependent on natural language understanding as well as shared knowledge. He noted that he was not concerned about the need to fool the judge regarding the machines physical appearance. A teletype communication link is employed to circumvent challenging (at the time) issues in speech recognition/synthesis. Just as Turing employed conventions to "shape" the problem into what he considered a tractable framework, we propose a set of our own conventions to arrive at formulations we believe will work for evaluating document analysis techniques.

For example, we might allow the judge to present various page images to the two players and ask questions about them. If a machine could answer such questions as well as a human, we would consider it "intelligent" with respect to the task at hand and, by extension, we might say that the underlying document analysis problem has been solved. Whether such a paradigm gains acceptance is for the community to decide. For now we simply pose the question and in the remainder of the paper address various points regarding implementation of this idea. It is good to keep in mind, however, that Turing's original formulation as well as our variations allow anyone to serve as judge, so the most skeptical member of the community is given the opportunity to confirm (or disprove) the assertion at stake. This is one of the features that make the Turing Test so compelling.

While often discussed in philosophical terms, the Turing Test has rarely been implemented. A notable example is the "Long Bet" between industry titans Mitch Kapor and Ray Kurzweil where, with a $20,000 wager on the line, the details of the deciding test are spelled out in some amount of detail [3]. Another concrete, albeit restricted, example is the concept of a CAPTCHA (Completely Automated Public Turing Test To Tell Computers and Humans Apart) now being used both to protect online services from automated attacks as well as a way of crowd-sourcing the collection of ground-truth for pattern recognition problems [4], [5], [6]. There are, however, important differences between a CAPTCHA (or reCAPTCHA) and what we are proposing here. Most notably, the judge in the case of a CAPTCHA is also a machine, and we might well anticipate significant skepticism in allowing one machine to declare that another machine is indistinguishable from a human on some task. We seek a more compelling standard of evidence.

In the next section, we discuss limitations with current approaches to performance evaluation, most of which can be viewed as highly simplified (and in some sense defective) versions of the Turing Test. We follow this by a section reviewing the properties of Turing's original test that must be maintained, and those that can be dispensed with in the interests of practicality. We then describe how we propose to adapt Turing's test to the matter at hand: deciding when a given problem has been solved. This section is followed by the presentation of several examples that we consider amenable to such an approach. We conclude with a discussion of open questions and future research.

II. Issues in Performance Evaluation

Existing approaches to performance evaluation have served us well up to a point. We believe, however, that they possess fundamental limitations that will prevent us from being able to decide when a problem has been solved. This provides the motivation for proposing a new paradigm based on the Turing Test.

At the risk of over-generalizing, the "standard" procedure is to assemble a dataset that represents an ill-defined sample of an ill-defined population. So-called "ground-truth" is then collected. This is the intended output from the algorithm we are trying to develop. The specifier of the truth must be led to an understanding of what the algorithm is trying to produce by the way in which the truthing protocol is defined. In other words, the human is trained to "think" like the machine.

Then the algorithm is run on the same page images (often repeatedly) and the output representation is compared to the truth, typically using an algorithmic technique. This yields an accuracy figure. It is questionable whether such measures relate in a meaningful way to the quality of the result with respect to its usefulness in a downstream application. We have termed the unattainable goal of 100% accuracy as the "endless pursuit of perfection."

It has been argued that a more appropriate approach might be to measure the work a human must do to "correct" the output of the algorithm via a suitable user interface. In this case, there is no pre-defined truth, but rather the human is shown the original document and then uses his/her knowledge to fix what the machine produces. However, even this approach is ill-defined – unless the human has a deep understanding of the intended application, and the interface is well-tuned to the problem, he/she may be led to make many unnecessary corrections, skewing the evaluation.

For certain document analysis problems (*e.g.*, determining voter intent from markings made on an optical scan ballot), it is easy to argue there is not just one ground truth, but rather a set of possible interpretations, any of which can be considered reasonable. In such cases, where human experts disagree, an endless cycle of editing would never converge. Contrast this situation with the conceptually simple, unambiguous decision to be made in the Turing Test.

We should note that we are not proposing to replace all forms of performance evaluation in this way – rather, this new approach is targeted specifically at the question: "Is this problem solved?"

There has been, and continues to be, other significant progress in the area of performance evaluation. In this category we place the growing interest in competitions at conferences and workshops in our field [7], as well as new systems such as the Lehigh Document Analysis Engine which provides third party certification of test results on previously unseen data [8]. Both of these are most certainly also steps in the right direction which complement what we are proposing here.

III. Salient Properties of the Turing Test

As we have noted, the "imitation game" originally proposed by Turing is not appropriate for our purposes as it stands. Here we list both those aspects of the test that can be discarded in the interests of practicality and those that must be preserved. Our intent is to avoid issues in previous approaches to performance evaluation which make it impossible to declare a problem has been solved.

Properties of the original Turing Test we must preserve:

- Human judgment is applied to determine a simple machine/human distinction and nothing more complex than this. Automated evaluation (*i.e.*, a computation to determine whether a response is more similar to some predetermined human or machine output) is ruled out.
- A judge may ask any number of questions before making a determination. A "question" here is a challenge that requires a response from the player. For document analysis applications, this will normally consist of a page image to be processed.
- A judge gets to decide which questions to use, and must be free to conduct the questioning of the players without constraint on the choice, sequence, and number of questions.
- A series of such evaluations, with anyone being allowed to serve as judge or as the human player, is conducted before declaring a problem "solved" (if/when the success of the best-performing judges is statistically no better than random).

Properties we do not need to preserve:

- Interaction between a judge and the players via a natural language question-and-answer process. Instead, we propose to employ standard graphical user interface paradigms (including the ability to upload image files, and visual inspection of process outputs). This is justifiable since we are not attempting to demonstrate general intelligence.
- Open-ended domains of discourse. Note that abandoning this point replaces Turing's original motivating question "Can machines think?" with our own question of interest "Is this problem solved?"

Properties that can be modified under some provisos:

- The same players and judge must continue to participate from the start of a given test until the final verdict for that test. It may be possible to dispense with this property if a trusted third-party is involved in the evaluation, unless answers to later questions depend on information gleaned from previously-seen page images.

IV. Adapting the Turing Test Paradigm

It has been understood from the start that there are significant challenges in creating an operative instance of the Turing Test. As noted in the preamble to the "Long Bet" between Kapor and Kurzweil cited earlier [3], Turing was very specifically nonspecific about many aspects of how to administer the test. Their approach is to specify concretely many of the necessary details: each of three Turing Test judges is to conduct an online interview ("chat") with each of four human players as well as the machine for two hours. At the end of these interviews, the judges indicate whether or not each candidate is human and also rank them from "least human" to "most human." The machine is said to pass the Turing Test if it fools two or more judges and if its median rank is equal to or greater than at least two of the human players.

This level of specificity is appropriate for implementing our idea as a competition at a conference and would be an interesting option to explore. Here, however, we describe a more open approach that leverages the rapid rise of Internet services and crowdsourcing, as embodied by the Lehigh DAE server [8]. We consider this to be a more powerful demonstration of the concept, as will soon become clear.

If implemented as a full-scale Turning Test, a judge would be able to provide any page image to a player and pose any query about it. Declaring that a machine has "passed" the test involves multiple judges conducting a number of rounds and determining whether the success of all judges (or, perhaps, of the best judge) is no better than random guessing. This level of generality raises serious technical issues, however, and is not necessary in our intended application.

A key issue is the need to focus on the specific DIA problem that the machine (algorithm) is claiming to address. How can we eliminate out-of-scope querying by a judge, such as submitting a half-tone document to a binarization program? To prevent such mischief, the creators of the machine under test should be required to formulate a formal specification for within-scope entities (usually document images) that constitute a challenge. All challenge documents must be validated according to these specifications. Therefore, the universe of queries available to a judge must consist either of arbitrary documents with a way to filter those that are out-of-scope at test time, or of samples drawn by the judge from a large pre-filtered population of within-scope documents.

As noted, the judge should be permitted to ask a series of questions, basing each on those that preceded it, until he/she becomes certain of a decision. In the original Turing Test, the inquisition takes the form of a free-form conversation. In our case, we must prevent the human from signaling to the judge in a way that is clearly impossible for a special-purpose document processing program. Both questions and answers must therefore obey a restrictive domain-specific syntax, somewhat like in a multiple-choice test. (For the convenience of the human player and the judge, the interface may include provisions to facilitate entering and displaying answers in a human-friendly format.)

Even in our more restricted case, a clever inquisitor might "probe" both players by selecting query documents to explore areas of perceived weakness. While inherently less conversational, this still conforms to the paradigm outlined by Turing and, in fact, provides substantial rigor to the test.

We envision our modified Turing Test running on an open server such as the DAE platform. Users (members of the research community) can volunteer at any point in time to serve as the judge or the human player to test a preregistered algorithm on some specific task. The need to pair a judge with a human player can likely be addressed through a crowdsourcing system that provides micropayments to recruit subjects such as Amazon Mechanical Turk. We believe it is important that the human player be "live" and not simply a pre-recorded set of opinions (unlike traditional approaches to performance evaluation).

How can we assure that the judges and the human players are giving it their best? One means of achieving this goal is by allowing anyone – even (or especially) the most ardent skeptic – to serve as judge or player. We can also deter some adversarial or careless humans (bad judges, bad players) who would make an algorithm look good by compensating them more highly if they are more successful. To avoid biased participants whose honesty cannot be bought, we need a number of tests with different combinations of judges and players before rendering a decision.

By openly publishing traces of all tests conducted on an algorithm, other researchers can be encouraged to follow along and render their own opinions. In this way, the behaviors of judges and players will themselves be subject to scrutiny. Ultimately, the community will determine which tests were conducted fairly and therefore can be used in computing the statistics that answer the question at hand.

Is it necessary to "lock" the two players in a room, even if only virtually? Yes, we believe this is necessary to make sure the machine does not "cheat" via access to human assistance. For example, the DAE server provides the functionality to run algorithms on a trusted third-party machine.

What about tasks that are natural for machines but very tedious for humans? Clearly it makes no sense to ask human players to try to perform the same search functions over billions of documents that google does so well. We could "dumb down" the algorithm drastically by, say, running it on very slow hardware, but this seems pointless. This suggests only that some tasks are not suited for evaluating this way, not that the basic idea of a Turing-like Test is flawed.

On the other hand, we certainly must worry about the machine being too fast even for tasks that can be readily performed by humans (Turing was also concerned about this same issue). An algorithm might be able to locate text lines on a page in a fraction of a second, while a human would take orders of magnitude longer. It is important that the machine be tripped up by its lack of accuracy, not by its excessive speed. Hence, to support a real-time evaluation, each algorithm must be encoded with an awareness of roughly how long a human would take to perform the same task and a built-in delay function in returning its results.

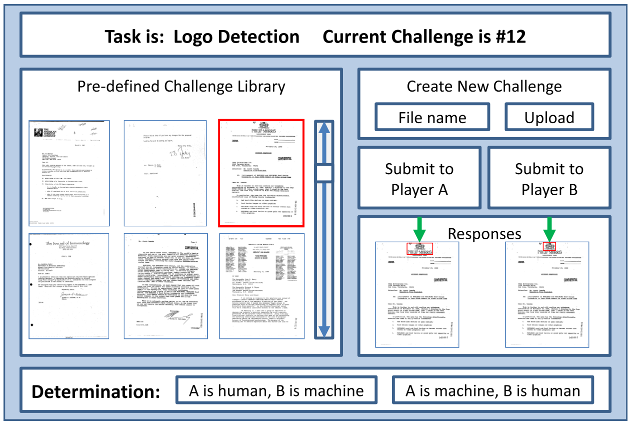

Figure 1. Mock-up of user interface for judge's control panel.

V. Examples

In this section we describe several tasks we believe would work well under the paradigm we have described.

1) *Logo detection*. The players are given a page image and must highlight (*e.g.*, draw a bounding box around) any logos that might appear on the page (Fig. 1). The judge is allowed to provide the input page images by (a) uploading a bitmap to the server, (b) selecting from a large library provided by an independent third party (*e.g.*, the DAE server), (c) pointing the players to an online bitmap, say from some digital library. Here the specifications may allow practically any page image.
2) *Logo identification*. Very much like (1), only the output is item numbers from a long list of company names (including "none of these"). Other similar problems include detecting and labeling photos or handwritten annotations on scanned pages.
3) *Table recognition*. Given an input table, provide the data from a specified cell in the table. With the right user interface, expressing the question would not require natural language.
4) *Writer ID*. Given two handwriting samples provided by the judge, determine whether or not they came from the same writer. This one should also be easy using a simple graphical interface. It could be modified to require searching a (small) database to determine the author of a given sample. Answers could be given as a Top-N list.
5) *Text transcription*. Here it is necessary to distinguish two tasks: maintaining the correct reading order and performing the actual transcription. However, this is a task where existing automated evaluation techniques are probably adequate.
6) *Language ID*. This is one task where the machine might do better than the human! (Turing makes note of such scenarios in his original essay, since even at that time, arithmetic was faster and more accurate using

computers.)

Tasks that might not be appropriate for this approach include those that provide fine-grain intermediate results to downstream programs, such as binarization or character segmentation. Also to be avoided are tasks where machines are already incontestably superior to humans, *e.g.* searching for a word in coded text.

VI. Human and Machine Learning

Turing says nothing about how to establish competence of judges. Fooling 100 inept judges has no value if one experienced judge is able to reliably distinguish human from machine. Still, some DIA requires acquired skills, so we could recruit 9th graders, high-school graduates, BA/BS degree recipients, and DAS workshop participants.

Turing did not envision one player seeing the interactions with the other. Clearly some learning might be possible purely based on what a single player sees in the traditional model. But things change dramatically if players can observe each other. In this case, clearly the judge cannot ask the same question of both players because the player to go second will have an obvious advantage. This does not have to be as straightforward as the machine simply repeating the human's response – rather, even the machine can take an input like this and refine it. It might be interesting to imagine how a human would refine a machine's output vs. the other way around, although it is not obvious that a judge would be able to use that information.

Both human and machine could conceivably learn from one another (imagine a scenario where the human player did not understand the task at first, and learns it by seeing the machine's output). This may be considered a kind of co-training. Of course, either player could also deliberately mislead the other.

To assess the state of machine learning, evaluation could eventually include a sequence of "similar" documents where the machine should see the judge-human interaction (or corrections through a non-NLP interface) on the first document, take it into account in processing the second document, and so forth. Learning/adaptation (here case-based supervised learning) is one of the processes that truly distinguishes humans from machines, even in DIA. The premise here is that the original machine performance is worse than human performance, but the machine may catch up.

This paradigm has a different goal than the tests discussed so far. We might be pleased to see the machine improve, but if we can identify it based on its early clumsy responses, then it still loses the test. However, if the tests take place with various judges over an extended period of time, then perhaps it is fair to declare the problem solved if by the end, the machine is indistinguishable from the human.

It is interesting to note that Turing concludes his 1950 paper with a forward-looking discussion of machine learning as perhaps the most viable way of building machines that can pass his test.

VII. Discussion

The essential difference between the above proposal and current methods of evaluation – including competitions – is that, instead of formulating detailed criteria for each task to rank machine performance relative to human performance (often in the form of questionable ground truth on hoary data), we propose a simple and universal binary criterion for channeling document research to unsolved problems.

We are not proposing a replacement for all forms of performance evaluation, nor is the Turing Test appropriate for all tasks we might seek to automate. Rather, this paradigm is intended to answer one specific question – "Is this problem solved?" – in situations where human-level accuracy and speed are the ultimate goals.

Suggested modifications of the original Turing Test include selection and reward of the judge and human players, and restrictions of both questions and answers to the scope of software designed for a specific DIA task. Publishing traces of all tests conducted on a particular algorithm will allow the community to form a consensus as to which to trust, thereby providing the statistical basis to decide when, in fact, the problem under study has been solved.

Acknowledgment

Daniel Lopresti acknowledges support from a DARPA IPTO grant administered by Raytheon BBN Technologies.

References

[1] D. Lopresti and G. Nagy, "When is a Problem Solved?," *Proceedings of the Eleventh International Conference on Document Analysis and Recognition*, September 2011, Beijing, China, pp. 32-36.

[2] A. M. Turing, "Computing Machinery and Intelligence," *Mind*, vol. 59, no. 236, October 1950, pp. 433-460.

[3] M. Kapor and R. Kurzweil, "A Long Bet: By 2029 no computer – or 'machine intelligence' – will have passed the Turing Test," http://longbets.org/1/.

[4] L. von Ahn, M. Blum and J. Langford, "Telling Humans and Computers Apart Automatically," *Communications of the ACM*, vol. 47, no. 2, February 2004, pp. 57-60.

[5] D. Lopresti, "Leveraging the CAPTCHA Problem," *Proceedings of the Second International Workshop on Human Interactive Proofs*, Berlin: Springer- Verlag, 2005, pp. 97-110.

[6] reCAPTCHA, http://www.google.com/recaptcha

[7] ICDAR 2011 Competitions,
http://www.icdar2011.org/EN/column/column26.shtml

[8] B. Lamiroy and D. Lopresti, "An Open Architecture for End-to-End Document Analysis Benchmarking," *Proceedings of the Eleventh International Conference on Document Analysis and Recognition*, September 2011, Beijing, China, pp. 42-47.