Acceptance Testing Artificial Intelligence – AIQ

Ken Pugh ken@kenpugh.com https://kenpugh.com

How do you apply test-first to an artificial intelligence application[1]? That was a question posed to me during one of my ATDD/BDD workshops. Here are my thoughts. I got inspiration from recent articles in *IEEE Spectrum* and *Communications of the AC.* I welcome additional suggestions from you. The examples here represent some of the areas in which AI is currently used. Another article will explore acceptance testing for autonomous vehicles.

Introduction

There are many types of AI programs. They range from neural networks that learn by training and configuration to fully programmed applications that might be written in Prolog, C++, or other languages. The programmed portions of an application could be tested using unit testing to ensure the components worked as designed. However, it's more difficult to unit test an application that learns by training. But acceptance tests can be created for them. One might consider the results of these tests as a form of Artificial Intelligences Quotient (AIQ)

Image Identification

One test that can be applied to many AI programs is a simple "how many did you get right". For example, a common application of AI is to identify objects in an image. Often the application is trained with one set of images that have been identified. Then it is tested with a second set of images to see if it correctly identifies them. Suppose the goal was to correctly images that contained lions, tigers, and bears. An acceptance test could be that 99% or 99.9% of the test images were correctly identified. The test set might also include images that were not lions, tigers, and bears to check that application correctly determined "none-of-the-above" for all these pictures or 99% or 99.9% of them. One common test instance is an image of white noise (random data), which should not be identified as anything.

A similar type test can be applied to identification of images such as ones that are pictures of defects in a manufacturing item. For example, the test could be no more than .1% false positives (identifying an image that did not portray a defect) and no more than .1% false negatives (not identifying a defect in an image that did portray one). If the identification had a result that meant "unsure, check manually", then a test might be no more than .1% "unsures".

Causality / Explanability

An issue with many AI programs is that they come up with an answer, but not with an explanation of how that answer was achieved. One might identify a tiger but cannot explain it was due to its stripes or

[1] . The article Advanced Topic - Succeeding with AI in SAFe - Scaled Agile Framework gives an excellent introduction to AI applications.

a bear because it was standing on two legs. For AI programs that do explain their answers, the explanation can be tested for plausibility and logic.

Business Value

AI is being used to make suggestions on videos to watch, items to buy, or other content to read. One test is to see what percentage of the time do users select a suggestion. Another is what percentage do they act on the suggestion once they select it (watch the video, buy the item, or read the content). A third is how the overall user actions fit into the hypothesis that suggestions deliver business value (such as more user engagement or more orders). Does the cost of providing suggestions deliver the expected return to the business?

Complex Algorithms

There are many complex algorithms that some people categorize to be artificial intelligence. One of them is determining the shortest or fastest route to a destination. Complex code tends to have components that can be tested – e.g., determining the time to traverse a distance when traveling at a given speed. Their overall output – e.g., the shortest route – can be checked against an oracle – answers created by manual or alternative implementations - to see if the algorithm is acting as desired. The tests can involve either exact duplication of the oracle (e.g., the same route) or a percentage comparison (e.g., no longer than 102% of the oracle.

Games

AI programs can play Chess, Go, and other games. Some learn by playing against themselves. Most of the games that have been programmed are those with discrete rules. A test can be the percentage of games won against other opponents – either computer or human.

Speech Recognition

Acceptance tests for speech recognition applications can follow on the same lines as image identification, e.g., the percentage of correctly recognized words or sentences. The tests can include inputs that have different frequencies, timing, and sound quality. Random noise can be used to see if it is identified as a word or just noise.

The Intelligence Test

There are several IQ tests for humans. One could have an application whose purpose is to score high on an IQ test. The acceptance test is the desired IQ that the application should achieve.

The Turing Test

Wikipedia describes the Turing Test as a “Test of a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human. Turing proposed that a human evaluator would judge natural language conversations between a human and a machine designed to generate human-like responses.“ The evaluator sees a text conversation between the two. “If the evaluator cannot reliably tell the machine from the human, the machine is said to have passed the test.”

Since man-machine communication has evolved from text, I propose a new test. An AI program, competing in Fortnite against humans, wins the game.