Eugene Goostman, a chatbot that passed the Turing Test

Simon Girard, Ewan Knapik

Introduction

- Rule based chatbot
- Vladimir Veselov, Eugene Demchenko and Sergey Ulasen
- devellopment started in 2001
- First ever chatbot to pass the Turing Test?

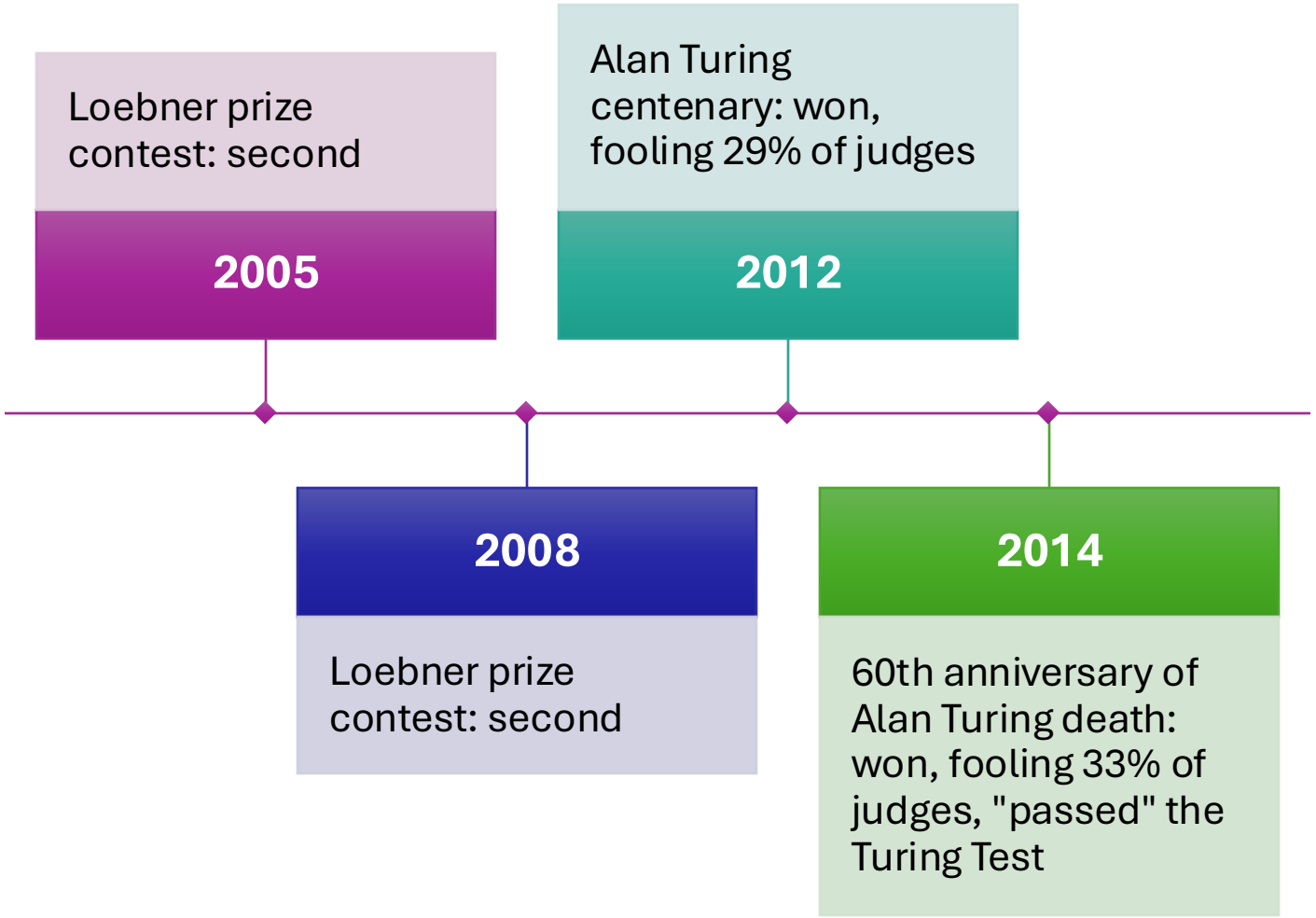
Turing Test

- Proposed by Alan Turing in 1950
- test for machine intelligence
- text-based conversation between human, machine, and judge
- judges has to guess who is the machine
- Highlights idea of machines "thinking" like humans

Who is Eugene Goostman?

- 13 years old ukrainian boy
- excuses lack of knowledge
- excuses poor english grammar
- old enough for meaningful conversation

Accomplishments



Features

- Spellchecker and typos correction
- context sensitive pattern matching
- logical or math expression interpreter
- integration with external database(s)

Inner working

- Input processed:
  - Type corrected
  - Tokenized
- Treematcher:
  - Find best matching pattern
  - Activate selected tree
  - Allows to find best fitting answer
  - Can include data from databases in answer

Problems encoutered

- Logical thinking
- Sense of humor
- Language choice
- Association game
- Learning knowledge

Controversy

- Not the first to pass a Turing Test
  - Pc Therapist III, 1991, Loebner Prize: 5 out of 10 judges
  - Cleverbot, 2011, Techniche: 59.3% of 1334 votes

- Turing Test doesn't test ability to think
- Test programmer's ability to trick

Comparison with today's technology

- Rule-based chatbots are far outdated
- Too much work to set up knowledge database
- Not as performant as LLM

- Inner working easier to understand
- Not as ressource demanding

Conclusion

- Pushed boundaries of rule-based chatbot
- Maybe not the first to pass the Turing Test, open to debate
- Turing Test not really about machine intelligence

Eugene Goostman, a chatbot that passed the Turing Test

Simon Girard,Ewan Knapik

May 26, 2025

1 Introduction

Eugene Goostman is a chatbot created by the 3 programmers Vladimir Veselov, Eugene Demchenko and Sergey Ulasen. It's development started in 2001, and by the year 2014 it was certified to have passed the Turing Test, fooling 33% of the judges into thinking it was human, after only 5 minutes of questioning.

2 Who is Eugene Goostman?

Eugene Goostman was designed as a 13 years old Ukrainian boy. Those characteristics were chosen to give some context to Eugene answers. The age of 13 has been chosen has a way to excuse the lack of knowledge of the bot, but still make it old enough to have a conversation with. It has also been chosen to indulge forgiveness in some of his grammatically incorrect answers, a smart way of covering some of the problems of chatbot development at that time.

3 Accomplishments

Eugene Goostman has competed in a few Turing Test contests over the years:

1. 2005 Loebner prize contest: in which it scored second.
2. 2008 Loebner prize contest: in which it again scored second.
3. 2012 Alan Turing centenary: won the contest, fooling 29% of the judges. The competition was considered the biggest ever hosted in its domain, and featured 5 bots, 25 hidden humans, and 30 judges.
4. 2014 60th anniversary of Alan Turing death: won first place, by fooling 33% of the judges, and deemed the first ever chatbot to pass the Turing Test, even though this statement is controversial.

4 Inner Workings

Eugene Goostman present some interesting features:

1. Spellchecker and typos correction: hard-coded possible mistake words, linked with the correct spelling. This allows for correct pattern matching, as the input is then closer to what the system expects. A clear increase in accuracy has been noticed after implementing this feature.

2. context-sensitive pattern matching: The answer of the chatbot depends on the theme of the question, and leads then to further refinement of the answer.

3. random or defined response: Answers can be defined as random, so that a same answer pattern would not be used multiple times, leading to a more believable conversation.

4. logical or math expression interpreter: The system can process simple logical and math operation, as to make it more believable and not break when asked specific questions.

5. integration with external database(s): The knowledge used by Eugene Goostman can be increased by linking it with other databases.

The framework used is fairly modular, and can be used to create rule based applications, like a document parser or some calculation application.
The logical architecture of the framework is described as a simple chain of command. An action is defined, and a signal is defined for this action.

The input is first processed, typo corrected, and then separated into token. The treematcher chose the best matching pattern for the input sentence, and activates the corresponding tree. The input is then processed through a tree of choices, as to find the best fitting answer. Each branch of the tree analyses the structure of the sentence in a reg-ex way, and provides a fitting, pre-chosen sentence that can include data fetched from databases if the input requires it (information about population of a city).

The 3 programmers have worked together on 3 different aspects of the knowledge database: common knowledge (science, news, encyclopedia), personal knowledge (name, age, occupation, family), and behavioral knowledge (etiquette, short questions, remarks). This work distribution allowed them to work alone, reducing friction, and then joining them all in the finished product.

Because the system is modular, rules can be inserted anywhere, and the response of the system doesn't have to be text-based, it can be any action.

5 Problems Faced During Elaboration

5.1 Categories of Judges

The chatbot had to be ready for several categories of judges, going from picky ones, that will ask trap questions, to really talkative ones, who will write more

than they will read. The hardest part was to take all those kinds of person into consideration, and create specific rules to answer each of them. The next few points will explain some of the strategies used to trick those judges

5.2 Logical Thinking

Some trick questions that you can ask a chatbot to break it are logical questions, that are really easy for a human to understand, but really hard to program into a rule based chatbot. Indeed, as those old rule based chatbot couldn't really think, each question had to be manually programmed into the bot, associated with the correct answer. Some pattern could be used for most questions, like "How many [parts] do [number] [objects] have in common?", for which data can be fetched from a database. As the number of such pattern wouldn't be too hard to program, the database needed to answer most question would be really painstaking to create. That is in part the reason for which most chatbot fail at logical thinking, they are created by amateurs that don't have neither the time nor resources to create such a database.

5.3 Sense of Humor

The use of humor might also be a way to trick the judges, as it was not frequent for chatbot of the time to use elaborate humor. An external database could be used to get the pattern of some multi-lines jokes, like "knock-knock". An understanding of irony could also be noticed by the bot, notably when the user formulates a very polite phrase, that could imply some sort of humor, to which the bot could answer accordingly. Some simple tricks could also be used to imitate some kind of humor, like when presented with the pattern "I am [...]", the bot could answer something like "we are all [...] in a way". The tricky part about programming humor is that there exists a lot of different kinds of humor, and hard-coding all of the in the chatbot would be a painstaking and quite impossible task.

5.4 Language

English was obviously an easy choice to use, as it would get the bot more recognition across the world, but it came with some pros and cons. One of the advantages of the English language is that a sentence with specific words can only be written in a single way, words have a defined place for the sentence to make sense. Indeed, in Slavic languages, from which the 3 programmers are coming, words use declensions, and can be placed at multiple spots in the sentence, as only the ending of the words helps in understanding their role in the sentence. That allowed them to use pattern-matching to understand the user's input, and not sentence parsing which would have been far more complicated. Moreover, the use of declensions would mean that all of the possibles cases of each word had to be put together in a database, that would mean an incredible amount of work. On the other hand, a lot of words in English are small and

similar (bag, fag, jag, lag, nag, rag, wag, beg, leg), and made the typo correction part of the input processing to require context to determine the correct form of the word. That wouldn't have happened in the programmers' languages, as Slavic languages tend to have longer words, easier to differentiate.

5.5 Association Game

Some questions that can be asked to the chatbot, to test his thinking abilities,are related to lexical fields, and related meanings of the words. One such question would be "What subject is wrong in this group:house, hut, sparrow, bungalow?". The way that those question may be answered by a machine is by associating each word with one or multiple categories, that way the system has a way of knowing how to classify the elements.

5.6 Common Sense

This element is one of the most tricky to implement, because a lot of question that would be common sense for a human, as for example question about his past or habits, would be too long to code into a chatbot. It would need an entire database containing information about its life, which would not be feasible. A solution would be to give a few predetermined, evasive answers to escape the question, or vague enough so that it can serve as an answer to a wide range of questions, while not being that precise. For example, when asked about its past life, the bot could answer that it doesn't like to talk about that period, or that it was too vast to have a clear and precise answer, escaping the question.

5.7 Learning Knowledge

The chatbot should be able to remember a few information during the dialogue, as not to seem like it doesn't remember past answers. The system is made, so that small amount of information can be memorized, but the nature of the rules-directed chatbot makes it so that it can lead to some odd situation. For example, if the judge tell the bot that "John lives in Kraków", and then ask "Where does John live?" the bot can easily answer that John indeed lives in Kraków, as it can remember this information. The issue become apparent as the bot doesn't have a deep understanding of words' semantic: If then the judge says that "John lives in poverty" and asks the question "Where is John", the bot will answer that "John is in poverty", mistaking it for a place.

6 Controversial Win

The statement that Eugene Goostman was the first chatbot to pass the Turing Test has been met with a lot of criticism. Indeed, a few other chatbots managed to trick a higher percentage in the previous decades; Cleverbot, in 2011, managed to trick 59.3 of its judges (1.334 votes) into thinking it was human.

PC Therapist III, in 1991, won the Loebner price by tricking 50% of the judges (only 10) into thinking that it was human.

The difference with these events was that the 2014 competition allowed the conversations to be unrestricted. Indeed, when PC Therapist III passed the test, the questions that the user could ask had been chosen beforehand.

But even more than that, even the meaning of the Turing Test can be questioned. What was actually being tested during these competitions was the ability of programmers to create a chatbot that can mimic human writing patterns. Is that really what Alan Turing had in mind? Those chatbots are not actually thinking, just applying a set of predetermined rules made to trick people into thinking that they are capable of reflexion.

7 Conclusion

Eugene Goostman was sure an impressive technological feat of it's time, and pushed the boundaries of what a rule-based chatbot was capable of. But to say that it was the first ever chatbot to pass the Turing Test can be questioned in more ways than one. One could say that the rules of the Turing test are not defined enough, and that other bot passed it before. It could also be argued that the Turing test is not a good metric of how smart a system is, but only measure how good the programmers were at mimicking human patterns.

Eugene Demchenko and Vladimir Veselov wrote themselves:

> "The imitation game" is exciting, amusing and highly intelligent – but it is nothing but a game. Do not expect that passing it means anything more than that some bot was luckier than the rest, or that there were more simpletons among judges this time.

Nowadays, with the emergence of LLM like ChatGPT, Deepseak, and many others, the Turing Test is yet again used as a benchmarking tool. If these new models have only just passed the Turing Test, can we still claim that earlier, more primitive systems like Eugene Goostman truly passed it in the first place?

References

[1] Eugene Demchenko and Vladimir Veselov (2008). *Who Fools Whom? The Great Mystification, or Methodological Issues on Making Fools of Human Beings*. In Robert Epstein, Gary Roberts, and Grace Beber (Eds.), *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Springer.

[2] Cleverbot.com (2007). *Chatbots 3.0 - Vladimir Veselov - Eugene Goostman the Bot* Retrieved from https://www.youtube.com/watch?v=ubfVPjQq-CA https://www.youtube.com/watch?v=vrhCKtqSib8

[3] Cleverbot (2025). *Cleverbot comes very close to passing the Turing Test.* https://www.cleverbot.com/human

[4] Chatbots.org (2009). *PC Therapist III: Loebner prize 1991.* https://www.chatbots.org/awards/winner_nomination/pc_therapist_iii/