The Turing Test

The Turing Test, developed by Alan Turing, distinguishes between a human and a computer in dialogue.

- If I can tell I am interacting with a computer, it has failed the Turing Test.
- If I am convinced I am interacting with a human, yet it’s a computer, then the program has passed the Turing Test.

ELIZA

ELIZA was a computer program written by Joseph Weizenbaum in the mid-1960s

- Simulation of a (Rogerian) therapist.
- User types questions, and ELIZA responds.

http://www.manifestation.com/neurotoys/eliza.php3

Example conversation (cont.)

ELIZA: WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP
USER: Perhaps I could learn to get along with my mother.
ELIZA: TELL ME MORE ABOUT YOUR FAMILY
USER: My mother takes care of me.
ELIZA: WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU
USER: My father.
ELIZA: YOUR FATHER
USER: You are like my father in some ways.
ELIZA: WHAT RESEMBLANCE DO YOU SEE
USER: You are not very aggressive but I think you don’t want me to notice that.
ELIZA: WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE
USER: You don’t argue with me.
ELIZA: WHY DO YOU THINK I DON’T ARGUE WITH YOU

Properties of ELIZA

What do you notice about ELIZA?

- Seems to show some understanding of English.
  - e.g. “What resemblance do you see?”
- Seems to have some understanding of real-world problems.
  - e.g. “Tell me more about your family.”
- Seems to keep track of the conversation.
  - e.g. “Does that have anything to do with...”

Four types of responses

1. A general question or statement
   *Does that trouble you?, Beware of addictions.*
2. A transformed version of the previous statement
   *Your father.*
3. A response relevant to the topic of the previous statement
   *Do you think coming here will help you...?*
A reproduction of an earlier statement
ELIZA: FIRST TELL ME WHY YOU HAVE COME HERE.
USER: To talk to a computer.
ELIZA: TO TALK TO A COMPUTER.
USER: Yes, to talk to a computer.
http://www.cogs.sussex.ac.uk/local/books/computers-and-thought/chap2/node2.html

How ELIZA works

They are many different implementations of ELIZA out there, but they all work in the same general way:

1. **Transform input** to change the point of view (I and you)
2. Perform pattern recognition (pattern matching)
3. Substitute keywords into **canned phrases**, or submit a general canned phrase

choices in ELIZA

User enters: I wouldn’t pay money for a computer.
- money matches one pattern
- computer matches another pattern
⇒ ELIZA picks one pattern at random

Why does ELIZA work so well?

- User assumptions/expectations: if you’re looking for a Rogerian psychotherapist, you may find what you expected.
- In theory, pattern matching can match a lot of user statements, given a large database of patterns.
- The way conversation works was taken into account when creating ELIZA
- Respond with something related to the previous statement; use some of the same vocab the other person did; etc.

How can we tell that ELIZA isn’t a human?

- Lack of initiative
- Lack of knowledge
- ELIZA knows nothing about the real world
- Lack of common sense
- Lack of true grammar/syntax = structure of sentences
- Say you peabrain to it and you might get me peabrain back.
- Need two separate patterns for Computers
- Doesn’t have (much of) a memory

Effect of ELIZA

People became somewhat emotionally involved with the system.
Weizenbaum’s secretary asked him to leave the room while she talked with ELIZA.
PARRY (Colby et al. 1971)

- keeps track of “global” emotional state
- when the anger variable is high, for example, PARRY will choose from a set of “hostile” options.

“With Parry, Dr. Colby established that a computer chip could be programmed to imitate a paranoid schizophrenic.”
http://www.edu-cyberpg.com/Linguistics/Parry.html

How does dialogue work?

- Basic facts about dialogues
- Gricean maxims
- Speech acts
- Plan recognition
- Discourse structure
- Reference

Basic facts about dialogues

- **Turn-taking**: must know when it’s the right time to contribute your turn
  - Comes naturally to humans; overlaps and long pauses are actually somewhat rare
- **Adjacency pairs**: two-part conversational structures where turn-taking is usually quite clear
  - e.g. question/answer, greeting/greeting, request/grant

Grounding

In order to establish common ground, speakers do various things:

- acknowledgments and backchannels = short utterance which indicates the utterance was heard and that the speaker should continue.
  - That’s a nice shirt.
  - Mm-hmm.
- continued attention, completing speaker’s utterance, and so on can also indicate acknowledgment

From Then Until Now

ELIZA was a very simple pattern-matcher, and PARRY and SHRDLU were only somewhat more complex. They’re just “chatterbots”

- There has been a push to add linguistic and real-world knowledge to dialogue systems
- So, instead of focusing on systems, we will focus on what a dialogue system needs to know in order to work.
Gricean maxim 1: Quantity

- Quantity: Be exactly as informative as is required.
  - Make your contribution as informative as is required (for the current purposes of the exchange)
  - How many pencils do you have?
  - Two. → means exactly two (or would have said more)
  - Do not make your contribution more informative than is required.
    (passing by somebody on the way to class)
    How’s it going?
    It’s complicated. Yesterday I was sick ...

Gricean maxim 2: Quality

- Quality: Try to make your contribution one that is true.
  - Do not say what you believe to be false
  - Do not say that for which you lack adequate evidence
  - Do you know how to drive a stick-shift?
    Yes, I do. I’ve seen my dad do it many times.

Gricean maxim 3: Relevance

- Relevance: Be relevant.
  - Is Gail dating anyone these days?
    Well, she goes to Cleveland every weekend.

Gricean maxim 4: Manner

- Manner: Be perspicuous (easy to understand).
  - Avoid obscurity of expression
  - Avoid ambiguity
  - Avoid unnecessary prolixity
  - Be brief and be orderly

Speech acts

Utterances are equated with actions in a dialogue (Austin 1962)
For example, there are performative verbs, as in these sentences:
- I (hereby) christen this ship The Swarthy.
- I pronounce you man and wife.
- I second that motion.
- I bet you five dollars the Buckeyes will win the NCAA championship this year.

Kinds of speech acts

Searle 1975
- Assertives = assert that something is the case (suggest, boast, conclude)
- Directives = commmand; attempt to get the listener to do something (ask, order, request)
- Commissives = commit to some future course of action (promise, plan, vow)
- Expressives = express psychological state of the speaker about some situation (thank, apologize, welcome)
- Declarations = bring about a different state of the world merely by saying them (performatives)

Indirect Speech Acts

Speech acts can be direct or indirect
- Direct: Form matches meaning.
  e.g. Please take out the garbage. (declarative structure; declarative meaning)
- Indirect: The form and the meaning are different.
  e.g. The garbage isn’t out yet. (declarative structure; imperative/request meaning)
Could you take out the garbage? (interrogative structure; imperative/request meaning)

DAMSL

We can try to define these speech acts and automatically figure out each utterance.
So, people have marked up text with different labels for speech acts.
DAMSL (Dialogue Act Markup in Several Layers) is a scheme of speech acts which people mark up texts with.
- forward-looking functions → question, request, etc., looking for a response
- backward-looking functions → answer, agreement, etc., providing a response
(Note the resemblance to adjacency pairs.)

Using DAMSL

- Using the DAMSL scheme, people can train a system on it, and then attempt to mark a new text with these labels.
- People can also have a dialogue system keep track of the previous forward-looking function to know how to respond.
Plan recognition

One thing a computer can do is to try to match what a speaker says with what it's looking for, i.e. a computer can be trying to fill a template, or schema.

E.g. If I call up a flight system, it knows that there is certain information which needs to be filled in: departure city, arrival city, dates of travel, etc.

Confirmation and repair strategies

Sometimes the system will have to confirm what a user said. E.g. 'So you want to travel from Amsterdam to Utrecht?' (explicit)

'When do you want to travel from Amsterdam to Utrecht?' (implicit) (http://www.compuleer.nl/Nimes.html)

Sometimes the system will have to account for a repair that the human made. E.g. I'd like to take the D Train, um, no, the A Train.

Determining structure

- If we can figure out the discourse structure automatically, the computer is better able to keep track of what's going on.

  ⇒ When you know the structure, you're better able to know what pronouns (and definite noun phrases) refer to.

- Determining structure is not always clear-cut, and humans can disagree as to the correct structure.

Template

BOOK-FLIGHT(Customer, Flight):
DEPARTURE-DATE(Flight) = ?
DEPARTURE-TIME(Flight) = ?
PAYMENT-METHOD(Customer) = ?
NUMBER-OF-REQUESTED-SEATS = ?
NUMBER-OF-OPEN-SEATS(Flight) = ?

Effect: FLIGHT-BOOKED(Customer, Flight) = Yes/No?

- So, when a customer says, I want to leave Dallas on March 3, the DEPARTURE-DATE and DEPARTURE-CITY both get filled in.
- Only when all of the template has been filled in is the flight actually booked.

Discourse structure

Discourse, or dialogue, has some structure to it—not just a bunch of random, loosely-connected statements.

- This is even more the case when the speakers are trying to achieve some purpose, or task, like giving directions or fixing a car → task-based dialogues
- Every discourse has a discourse purpose = a reason for talking (Grosz and Sidner 1986)

The problem is trying to figure out the intentions of the speakers.

Subdialogues

And within the overall dialogue, there are subdialogues

- E.g. in order to book a flight, you must ask me where I want to go.
- Or I may have to correct you in a subdialogue (I said Columbus, not Columbia)

Inferring Intention

Computers can try to infer what a user intended, or what a user is intending to do next.

⇒ Convert an indirect speech act into something a computer can use.

Can you give me a list of flights from Atlanta?

⇒ If the user asks if I'm capable of doing something, it probably wants me to do that thing.

Reference

We call what a word refers to its reference = using expressions to refer to things

- Indefinite noun phrases: new to the discourse, e.g. a book, some books
- Definite noun phrases: something identifiable to the listener, e.g. the book, those books
- Pronouns: replacement for a noun phrase, e.g. it, she, etc.

Coreference

These different kinds of noun phrases can refer to the same thing, i.e. coreference

I saw a monkey yesterday. The dumb thing was wily, and it kept throwing things at me.

⇒ a monkey, the dumb thing, and it all refer to the same real-world creature.
Preferences for Pronoun Reference

We know they all co-refer, but how can a computer tell? How does it go about solving the task of pronoun resolution?

- Syntactic and Semantic Constraints
- General Preferences/Heuristics

The task is to find the previous noun phrase (NP) in the discourse which the current pronoun (or definite NP) refers to.

General Preferences/Heuristics

After filtering out unwanted referents due to agreement and so on, we have various heuristics we can use to find the pronoun’s referent.

- Recency: pronoun refers to most recent possible NP
- John bought a cookie, and I bought a cake. Mary ate it.
- Subject over Object: prefer matching the pronoun to a previous subject of a sentence over the sentence’s pronoun. John hit Bill. Then I asked him to come with me.

Syntactic and Semantic Constraints

- Number, gender, and person agreement: co-referents must agree in all of these properties.
  - John has a new car. It’s red. (number)
  - John has a new car. He (=John) is attractive. (gender)
  - John and I have new cars. We/They love these cars. (person)
- Syntactic constraints: reflexives (himself, herself, ...) generally refer to subject of sentence
  - John bought him/herself a new car.

Resolving Pronouns

So, we can combine all this knowledge into a system for resolving pronouns.

1. Take all nouns in a sentence and make a list of possible referents.
2. Rank the nouns in terms of recency, frequency, subjecthood, and so on.
3. In the next sentence, try to match the pronoun with something in that list, starting with the most probable.
4. Rule out any “match” which violates agreement or other syntactic and semantic constraints.

How do we get our knowledge about how dialogue works?

Look at human-human interaction, i.e. how humans have interacted with computers.

Look at human-computer interaction, i.e. how humans have interacted with computers.

Also HH psycho-linguistic experiments (Jurafsky & Martin, 18.4 (p. 707-712))

Human-human and human-computer interaction

We look at corpora — large texts of collected data, often annotated with linguistic properties.

- e.g. A corpus of dialogues about giving directions from a map.
- Natural dialogues = A record of two (or more) humans speaking with one another, often about a task
- Wizard of Oz dialogues = A person talks to a “computer”, which is actually a human (thus the WOZ analogy)

Natural dialogues

Pluses:

- Natural user tasks and needs
- Easy to set up and record

Minuses:

- These record human-human interactions, not human-computer interactions
  - People often speak differently to a computer, will often adapt to the way a computer is talking to them.

Selectional restrictions

- Selectional restrictions: verbs pick out what kinds of nouns they can have for subjects and objects.
  - e.g. drive needs a human object and a drivable object.
  - John parked his car in the garage. He had driven it around for a bit.

Note that these restrictions can be extended: The White House said yesterday ...

Components

Dialogue System

Introduction

Early Systems

ELIZA

How Does Dialogue Work?

Subject over Object: prefer matching the pronoun to a human object and a drivable object.

Parallelism: Back-to-back sentences with similar structures can help pick out the referent.

Jim went with Carl to the supermarket. And I went with him to the gas station.

Corpora

Working on a large text of collected data, often annotated with linguistic properties.

e.g. A corpus of dialogues about giving directions from a map.

Plan Recognition

Reference

GRICEAN MAXIMS

Basic Facts

SHRDLU

PARRY

TRIPS

Corpora

Working on a large text of collected data, often annotated with linguistic properties.

e.g. A corpus of dialogues about giving directions from a map.

Plan Recognition

Reference

GRICEAN MAXIMS

Basic Facts

SHRDLU

PARRY

TRIPS
Wizard of Oz dialogues

"Computer" is actually a human in disguise.

Pluses:
- Provide insights on human-computer interaction (HCI) without having to set up a computer which is capable of HCI.
- Allow freedom in the range of tasks you can cover.

State transition networks

Well-structured dialogue can be modeled with a state transition network.

*E.g.* Dialogue system where the user only has to answer questions → very constrained
- State 1: I know nothing, so I need to ask for the user's name.
  - If the user gives me a name, go to State 2.
  - If the user does not give me a name, go back to State 1.
- State 2: I know the user's name, so I ask for the user's birthday.
  - If a birthday is given, go to State 3.
  - If a non-birthday is given, go to State 2.
  - If the user claims their name was wrong, go back to State 1.
- State 3: I know the user's birthday and name, so I ask for other things.
  - If information is given, go to State 4.
  - If the user claims their birthday was wrong, go back to State 2.

Finite-state automata

A state transition network is basically just a finite-state automaton (FSA) = network of states and transitions between states. (Also called a finite-state machine.)

- By moving through the FSA, you try to match an input string.
- The idea is that you can move from state to state, when conditions on the arcs are met. E.g. Move from state 0 to state 1 if I encounter the letter c. (Otherwise, do nothing.)

**FSAs continued**

We represent states with nodes (circles) and transitions with arcs (arrows) in a picture. (Note that the numbers on the nodes are just for our convenience; they don't really "mean" anything.)

The FSA above matches (or generates) the string color or the string colour. The double lines around node 6 means that this is a final state = can end at this state. I.e. If you match colo, you are not done. (you have to get to a double circle or else it fails.) What regular expression does this match?

**Drawing FSAs**

Things to remember:
- You have to have a single start node, but you can have more than one end node.
- The nodes don't necessarily have to go in order from left to right.
- FSAs are what are used to match regular expressions. Logically, they are equivalent.

**Frame-based systems**

Dialogues with a little more complexity—e.g. the user can ask questions, too—can be modeled with frame-based systems = systems based on frames, or templates. We saw these earlier:

```
BOOK-FLIGHT(Customer, Flight):
  DEPARTURE-DATE(Flight) = ?
  DEPARTURE-TIME(Flight) = ?
  PAYMENT-METHOD(Customer) = ?
  NUMBER-OF-REQUESTED-SEATS = ?
  NUMBER-OF-OPEN-SEATS(Flight) = ?

... Effect: FLIGHT-BOOKED(Customer, Flight) = Yes/No?
```
Components of a dialogue system

- Interface (media/mode)
- Discourse manager
- Context manager
- User model
- Knowledge sources
- Language Interpreter & Generator

Discourse manager

a.k.a. Dialogue manager
- Decide whose turn it is
- Decide what items need attention
- Feed input and output to the linguistic systems.

Context manager

Pay attention to the surrounding context:
- What time it is
- Where the user is, what the conditions are there
- If the user is looking at something (e.g. a map), what does that map look like?

Knowledge sources

Keep track of knowledge about the world the system might need
- If the user is looking at something (e.g. a map), what does that map look like?
- A database of flights: what flights are being offered by this airline?
- A listing of road closings

Language Interpreter & Generator

And of course, we need to convert what the user says into some form of meaning representation (natural language understander). And we need to convert the system's (meaning) response into some form of language (natural language generator). A great deal of knowledge about how sounds, words, and sentences are formed is needed.

User model

Who is the user? What is the user thinking?
Try to deduce various facets of the user:
- Beliefs = what do they believe to be true about this conversation?
- Intentions = what are they trying to do?
- Capabilities = what are they capable of doing?

Interface (media/mode)

If it's a spoken dialogue system, need an automatic speech recognizer.
If it's a written dialogue system, simply need a keyboard. Can also consider the aesthetics of the system, if someone is communicating with a talking picture on a computer screen, for instance.
Example system: TRIPS

TRIPS: extension of the earlier TRAINS system at the University of Rochester

- Human works with TRIPS in order to construct plans in a crisis situation: a hurricane is approaching the island of Pacifica, and the people must be evacuated.
- Features:
  - Task is clear
  - Quality of task is easily measured
  - Can vary the complexity of the task
  - Complex model of collaborative problem-solving