Lecture 17: Authoring Systems: Minstrel and Twitter Bots

CM 148
Kathleen Tuite
ktuite@ucsc.edu
May 9, 2015
Automating Creativity and Storytelling

To understand human cognition, it is essential that we understand the process of creativity.

Implementing creativity in computer programs has the potential to greatly increase the scope and power of the modern computer.
Overview

• Minstrel is an author-modeling story generator
  – Tells stories about King Arthur & co.

• Minstrel is a model of creativity
  – Uses a case-based model of problem solving
  – Creativity heuristics: transform-recall-adapt methods (TRAMs)
  – Creative problem solving of Minstrel can be applied elsewhere
High-level assumptions about creativity

- Creativity is driven by a failure of problem solving
- Creativity is an extension of problem solving
- New solutions are created by using old knowledge in new ways
Case-Based Reasoning

• Driven by episodic memory rather than a knowledge base of inference rules
  – How can I map this new problem onto one I already know how to solve?

• Problem-solving process:
  – Recalls past similar problem and its solution
  – Adapts past solution to current problem
  – Assesses the result
• Random aside
  – Google’s self-driving car can only drive on roads that have already been mapped out. Similar to CBR.
  – But! It can sense and adapt to new conditions.
Three types of cases in CBR

• Ossified cases (or rules)
  – Proverbs are a common example
  – Abstracted rules, used when no specific experience is accessible

• Paradigmatic
  – When only one prior experience is relevant, but it doesn’t quite fit and needs to be adapted

• Stories
  – Complex combination of ossified and paradigmatic
  – Unique and full of detail, but with points
kNN is a really simple example CBR

- k nearest neighbor
Ways case-based reasoning can fail

• Fail to find a solution to a similar problem

• Fail to adapt solution

• Fail domain assessment
  – Not efficient enough
  – Not using the right materials
  – Doesn’t meet story constraints, like being consistent or plausible
  – Too boring of a solution (Minstrel)
Tie in to creativity

• Case-based reasoning is used for problem solving

• Creativity is an extension of problem solving (especially when default problem solving fails)

• Need to modify case-based reasoning to account for creative problem solving
Challenges of creativity

• New solutions require knowledge not indexed under current problem
  – Creativity begins when problem solving fails – in CBR, this means that no solutions are found for current problem, or adaptation runs into a dead end

• Creativity involves recasting the problem
  – If trying to solve the current problem is failing, try solving a related problem (search problem space, not just the solution space)

• Adaptation must avoid too much complexity
  – Solving arbitrarily different problems, then trying to adapt that to the original problem, won’t work (limits on creativity)
Failure-driven creativity

• Minstrel gets to be creative when it encounters a problem for which past solutions fail

• It also counts making boring, repetitive solutions a failure and tries to come up with creative new alternatives
Minstrel’s creativity heuristics: TRAMs

• Transform-Adapt-Recall methods

• Process
  – Apply a problem transformation
  – Recall a solution from episodic memory for the transformed problem
  – Adapt the recalled solution back to the original problem domain

• A direct recall from memory is a null TRAM
  – TRAM:Standard-Problem-Solving
Example problem space

A princess kills a dragon

A knight kills a dragon

A knight injures a dragon

A dragon eats a princess

A knight kills a knight
Examples of specific TRAMs

- TRAM: Cross-Domain-Solution
  - **Transform** strategy: find a new problem domain with similar actions and actors, and analogically map current problem to new domain
  - (Recall a relevant solution)
  - **Adapt** strategy: Map any discovered solutions back to the original domain by reversing analogical mapping

One day while out riding, Lancelot’s horse went into the woods. Lancelot could not control the horse. The horse took him deeper into the woods. The horse stopped. Lancelot saw Andrea, a Lady of the Court, who was out picking berries.

John was sitting at home one evening when his dog began acting strange. The dog was scratching at the door and whining for a walk. Finally, John decided to take the dog for a walk. While they were out, John ran across his old friend Pete, whom he hadn’t seen in many years.
Example of TRAM: Cross-Domain-Solution

- **Domains the system knows about**
  - Knights and princesses (King Arthur)
  - Businessmen and friends

- **Original Problem:** create a scene in which a knight accidentally meets a princess
- **Transformed Problem:** create a scene in which a businessman accidentally meets somebody

- Recalled scene: “John was sitting at home one evening when his dog began acting strange. The dog was scratching at the door and whining for a walk. Finally, John decided to take the dog for a walk. While they were out, John ran across his old friend Pete, whom he hadn’t seen in many years.”

- Adapted solution: “One day while out riding, Lancelot’s horse went into the woods. Lancelot could not control his horse. The horse took him deeper into the woods. The horse stopped. Lancelot saw Andrea, a Lady of the Court, who was picking berries.”
Problem Solving with TRAMs

- Episodic Memory
- Domain Assessments
  - Boredom Assessments

1. Problem Specification
2. Transform original problem
3. Imaginative memory
4. Adapt past solutions
5. Assess solutions
6. Solution

- Active TRAM
- TRAM: Standard-Problem-Solving & Other TRAMs
Imaginative memory

• Central step in Minstrel: recall a solution from episodic memory

• What if we use creative problem solving on the “recall problem”?  
  – Problem to solve: “find something in episodic memory that matches these features”

• Pool of TRAMs
  – TRAM:Standard-Problem-Solving (get from episodic memory directly) is first
  – If that fails, try other TRAMs, which will adapt memories leading to imagined memories
Imaginative memory: recursive TRAM application

- Leaps in creativity result from combinations of small modifications
Inventing suicide: example TRAM application

- Episodic memory contains two episodes:

Knight Fight
A knight fights a troll with his sword, killing the troll and being injured in the process.

The Princess and the Potion
A lady of the court drank a potion to make herself ill.
Application of TRAM: Generalize-Constraint

- **TRAM: Generalize-Constraint**
  - **Transform**: select and generalize a feature (call it $generalized-feature$) of the scene specification. Use this new scene specification as an index for imaginative recall.
  - **Adapt**: adapt the recalled solution to the current problem by adding $generalized-feature$ back to the recalled scene.

---

**Initial Problem**

- **&Act.47**
  - *Actor*: &Knight
  - *Type*: &M-FIGHT
  - *To*: &Knight
  - *Object*: &Sword.18

**&State.44**

- *Type*: &Health
- *Object*: &Knight
- *Value*: &Dead

**Transformed Problem**

- **&Act.52**
  - *Actor*: &Knight
  - *Type*: &M-FIGHT
  - *To*: &Troll.121
  - *Object*: &Sword.14

**&State.51**

- *Type*: &Health
- *Object*: ???
- *Value*: &Dead

---

**Recalled Episode**

- **&Act.12**
  - *Actor*: &Knight.14
  - *Type*: &M-FIGHT
  - *To*: &Troll.121
  - *Object*: &Sword.14

**&State.14**

- *Type*: &Health
- *Object*: &Troll.121
- *Value*: &Dead

---

**Final Solution**

- **&Act.47**
  - *Actor*: &Knight
  - *Type*: &M-FIGHT
  - *To*: &Knight
  - *Object*: &Sword.18

**&State.44**

- *Type*: &Health
- *Object*: &Knight
- *Value*: &Dead

- **&Act.47**
  - *Actor*: &Knight
  - *Type*: &M-FIGHT
  - *To*: &Knight
  - *Object*: &Sword.18

**&State.44**

- *Type*: &Health
- *Object*: &Knight
- *Value*: &Dead

---

- A knight killed himself by fighting himself with his sword.
- A knight named Lancelot fought a troll with his sword and killed it.
Generalizing features using class hierarchy

• TRAM: Generalize-Constraint tries to use most specific generalization that recalls a solution

• If we completely removed the constraint, can have creativity failures
  – Example: solve “A knight gives a princess something to make her happy.”
  – Transform: “A knight gives somebody something to make them happy.”
  – Recall: “A knight gives a troll a hunk of meat to make it happy.”
  – Adapt: “A knight gives a princess a hunk of meat to make her happy.”
Example using recursive TRAMs

- **TRAM:Similar-‐Outcomes-‐Partial-‐Change**
  - **Transform:** Replace the resulting state caused by an action with a related state.
  - **Adapt:** Replace state change in recalled episode with original state.

- **Apply TRAM:SOPC to “A knight purposefully kills himself” to create “A knight purposefully injures himself”**
- **Recursively apply TRAM:GC to “A knight purposefully injures himself” to create “Someone purposefully injures themselves”**
- **Recall:** “A lady of the court made herself ill by drinking poison”
- **Adapt (using TRAM:GC) to “A knight made himself ill by drinking poison”**
- **Adapt (using TRAM:SOPC) to “A knight killed himself by drinking poison”**
Minstrel is driven by author-level goals

- The preceding slides described the solving of character-level goals
  - The character goals were set by an author to serve the story, but aren’t themselves directly goals about the form and content of the overall story

- Minstrels author goals
  - Thematic goals – the stories illustrate a theme, in Minstrel’s case, Planning Advice Themes (e.g. “A bird in the hand is worth two in the bush.”)
  - Drama goals – goals regarding the unity of action (tragedy, foreshadowing)
  - Consistency goals – motivate and explain story actions
  - Presentation goals – goals about which events must be fully described, and which can be summarized or omitted (in general diegetic goals)
Planning Advice Themes (PATs)

- Themes given to Minstrel
- Taken from *Romeo and Juliet, It’s a Wonderful Life*, assorted proverbs
  - PAT:Hasty-Impulse-Regretted
Minstrel’s creative capabilities

• Creativity principle: *New solutions are created by using old knowledge in new ways*

• Minstrel can produce discoveries that weren’t already in the system

• Scott Turner purposefully starts Minstrel with limited input to show off its creativity
  – The more it knows, the more it gets in trouble and generates inappropriate stories
Modern creative AI systems

• I’m making huge generalizations here:

• Need (at least) two things:
  – A way to **generate** new, creative output
    • By recombining old source material in new ways, possibly from a very large corpus
    • Experiment with different recombination methods, different evolution methods
  – A way to **test** the goodness of the output
    • Is it new?
    • Is it valuable?
    • Does the system judge itself or does it let outsiders judge?
    • Quantify creativity to enable comparisons
      – Is this solution *more* creative than others?
Inappropriate Minstrel stories

• PAT:PRIDE

Once upon a time, a hermit named Bebe told a knight named Grunfeld that if Grunfeld fought a dragon then something bad would happen.

Grunfeld was very proud. Because he was very proud, he wanted to impress the king. Grunfeld moved to a dragon. Grunfeld fought a dragon. The dragon was destroyed, but Grunfeld was wounded. *Grunfeld was wounded because he fought a knight. Grunfeld being wounded impressed the king.*
Inappropriate Minstrel stories

• Knight has a goal of feeding himself
  – Minstrel has no episode in memory of a knight feeding himself

• Applies TRAM:Generalize-Actor
  – Generalizes knight to monster because they are both violent characters
  – Recalls a scene in which a dragon eats a princess

• Result: cannibalism!
  – Knight kills and eats the princess to assuage his hunger
Inappropriate Minstrel stories

• The most famous outputs of systems like Minstrel and another program called Tale Spin are often the “mis-spun tales”, because of their amusement value

• Segue to Twitter
  – Bots with personality
Kathy Butler (@katherynebutler)

- **https://github.com/ElizabethU/The-Biggest-Fan**

- The bot superficially appears to be a weird human who is really determined to be friends with you. It's kind of like the experience of being facebook friends with your parents.

- Tweets random but sensible tweets on a variety of subjects. The subject is read randomly from a text file, and the searches for a tweet with the topic, and steals the text, sometimes with surprising results. Tweets are filtered for undesirable words read from another text file.

Uses **recall** to generate new tweets

Hobbies keyword list: Seattle coffee Seahawks knit css dog yoga camping bike baking yum book through game of thrones sherlock cute inside happy namaste lol mariners ridiculous supernatural pokemon
The Sorting Hat Bot (@sortingbot)

I'm the Sorting Hat and I'm here to say
I love sorting students in a major way

• Let’s try it out (https://github.com/dariusk/sorting-bot)
• Makes use of grammars, rhyming, word pronunciation and stresses, knowledge of nouns/adjectives/adverbs
An army of bots

• @portmanteau_bot
  – Smashes two words together by fusing the sounds and the meanings of its components

• @badjokebot
  – Generates bad jokes where the punchline is a portmanteau

• @HinkyPinkyQs and @HinkyPinkyAs
  – A joke-generation duo of bots
Twitter bot wrap up

• Let’s look at even more! (If we have time)

• Are these bots creative?
  – Are they creating novel solutions that have value?
That’s what she said: Double-entendre identification

Chloé Kiddon and Yuriy Brun
Computer Science & Engineering
University of Washington
Seattle WA 98195-2350
{chloe,brun}@cs.washington.edu

Abstract

Humor identification is a hard natural language understanding problem. We identify a subproblem — the “that’s what she said” problem — with two distinguishing characteristics: (1) use of nouns that are euphemisms for sexually explicit nouns and (2) structure common in the erotic domain. We address this problem in a classification approach that includes features that model those two characteristics. Experiments on web data demonstrated related research has not studied the task of identifying double entendres in text or speech. The task is complex and would require both deep semantic and cultural understanding to recognize the vast array of double entendres. We focus on a subtask of double entendre identification: TWSS recognition. We say a sentence is a TWSS if it is funny to follow that sentence with “that’s what she said”.

We frame the problem of TWSS recognition as a type of metaphor identification. A metaphor is a figure of speech that creates an analogical map.
Watson’s Cookbook

Cognitive Cooking with Chef Watson

Recipes for Innovation from IBM & the Institute of Culinary Education
Required Reading

• Scott Turner on Minstrel
  – Grand Text Auto blog
  – https://grandtextauto.soe.ucsc.edu/2007/10/30/scott-turner-on-minstrel/