MAKEBELIEVE: Using Commonsense Knowledge to Generate Stories

Hugo Liu, Push Singh
MIT Media Laboratory

AAAI 2002
Student Short Paper
2002.07.31
Edmonton, Alberta, Canada
Agenda

• **What’s in a story?**
  • Previous Approaches in Computational Stories
  • An Approach using Commonsense Knowledge
  • Knowledge Representation
  • Flexible Inference
  • MAKEBELIEVE Architectural Overview
  • Generated story examples, and evaluation
  • Conclusion and Future Work
A simple story example

“John became very lazy at work. John lost his job. John decided to get drunk. He started to commit crimes. John went to prison. He experienced bruises. John cried. He looked at himself differently.”

- Generated by makebelieve
Critics

That was a bad story! No rhetorical style, no depth, no details, and there is only one character!

“It captured the fatalistic essence of tragic man. People can relate and understand John’s experiences. His moment of reflection makes for a powerful ending.”
So... what’s in a story?


Did our story example demonstrate any of these? Arguably, yes! But it wasn’t programmed to having any plot devices!
Agenda

• What’s in a story?

• **Previous Approaches in Computational Stories**
  • An Approach using Commonsense Knowledge
  • Knowledge Representation
  • Flexible Inference
  • MAKEBELIEVE Architectural Overview
  • Generated story examples, and evaluation
  • Conclusion and Future Work
Two (Competing) Computational Approaches

- **Structuralism (ala Klein)**
  - Text structures reflect social structures
  - De Saussure: *story structures are interconnected.*
    *Atomic concepts don’t have meaning*
  - Stories can be produced using real-world inspired story grammars and canned story sequences

- **Transformationalism (ala Dreizin, Dehn, Meehan)**
  - Stories are the result of simulation
  - Expert rules (Dreizin) or narrative goals (Dehn) are applied to atomic story elements such as setting and characters
  - Story can be viewed as problem solving (Meehan)
An Example of Structuralism

• (from Klein 1973 program)
  – WONDERFUL SMART LADY BUXLEY WAS RICH. UGLY OVERSEXED LADY BUXLEY WAS SINGLE. JOHN WAS LADY BUXLEY'S NEPHEW. IMPOVERISHED IRRITABLE JOHN WAS EVIL. HANDSOME OVERSEXED JOHN BUXLEY WAS SINGLE. JOHN HATED EDWARD. JOHN BUXLEY HATED DR. BARTHOLOMEW HUME. BRILLIANT HUME WAS EVIL. HUME WAS OVERSEXED. HANDSOME DR. BARTHOLOMEW WAS SINGLE. KIND EASYGOING EDWARD WAS RICH. OVERSEXED LORD EDWARD WAS UGLY. LORD EDWARD WAS MARRIED TO LADY JANE. EDWARD LIKED MARY JANE. EDWARD WAS NOT JEALOUS. LORD EDWARD DISLIKED JOHN.
Structuralism: Pros and Cons

• Pros
  – High degree of complexity and interconnectedness reflects real social structures
  – ‘Canned’ story grammar and sequences lead to well-formed and believable stories

• Cons
  – Story grammars and complexity are hand-coded
  – Sticking to canned story structures limits creativity and variation in story lines
An Example of Transformationalism

- (from Meehan’s TALE-SPIN program, 1977)
  - ONCE UPON A TIME GEORGE ANT LIVED NEAR A PATCH OF GROUND. THERE WAS A NEST IN AN ASH TREE. WILMA BIRD LIVED IN THE NEST. THERE WAS SOME WATER IN A RIVER. WILMA KNEW THAT THE WATER WAS IN THE RIVER. GEORGE KNEW THAT THE WATER WAS IN THE RIVER. ONE DAY WILMA WAS VERY THIRSTY. WILMA WANTED TO GET NEAR SOME WATER. WILMA FLEW FROM HER NEST ACROSS THE MEADOW THROUGH A VALLEY TO THE RIVER. WILMA DRANK THE WATER. WILMA WASN'T THIRSTY ANYMORE.
Transformationalism: Pros and Cons

• Pros
  – More creativity by the free application of rules and goals to story elements
  – Viewing the story as a “trace” of how a character solves a problem gives purpose to the story line

• Cons
  – Story-telling as problem solving limits the development of the plot line
  – Rules and narrative goals have to be hand-coded
  – Freely applying rules to story elements may result in unforeseen “nonsensical” story steps.
MAKEBELIEVE: Approach

- Inherits from both structuralism and transformationalism
- Transformationalist qualities
  - Simulates stories from individual story steps
  - “Rules” affect how story steps are selected
- Structuralist qualities
  - Makes use of commonsense corpus knowledge about how real world events can be causally linked
  - At a low level, story steps are based on ‘canned’ real-world structures
Agenda

• What’s in a story?
• Previous Approaches in Computational Stories

• An Approach using Commonsense Knowledge
  • Knowledge Representation
  • Flexible Inference
  • MAKEBELIEVE Architectural Overview
  • Generated story examples, and evaluation
  • Conclusion and Future Work
Story generation:
chaining causal events

• In MAKEBELIEVE, a story is a causal chain of events experienced by a character(s).

• Given a *large* set of causal relations from The Open Mind Commonsense Corpus, e.g.
  “Something that can happen if you drive a long time is you might fall asleep”,
  “If you sleep you might have a dream.”
  …. (etc)

• Find ways to link the steps together into a storyline, e.g.
  John was driving for a long time. John fell asleep. John had a dream…. 
What makes a story good or bad?

- The quality and coherence of the story depend solely on how events are linked together (selection of next_story_step).
- There are local constraints i.e. match the effect of one statement to the cause of another statement.
- And there are global constraints (plot goals, coherence), etc.
  - Coherence: If John fell asleep while driving and had a dream, eventually he will wake up and maybe crash?
More on commonsense

- Source of commonsense is the Open Mind Commonsense corpus (OMCS)
- Knowledge is gathered through amateur web teachers (b/c everyone has commonsense to teach)
- OMCS has close to ½ million semi-structured English sentences about commonsense
- **Pitfalls:** using English as a representation leads to word-sense ambiguity, reference ambiguity, not fully parseable.
- **Advantages:** easily gathered, English representation is flexible, i.e. compared to Cyc
Screenshot of OMCS

Teaching computers the stuff we all know

Welcome Hugo, to Open Mind! You have entered 100 items

Search: [ ] Open Mind

Other Activities: Information · Preferences · Logout

Cause and effect

We can all predict the effects of our ordinary actions and of ordinary events in the world, like eating will make you less hungry, or that things fall when they are not supported, but no computer can do this. Please teach Open Mind more about such cause and effect.

Something that might happen as a consequence of reading the newspaper is you get the news.

Teach Open Mind! · Give me a new template
Story Steps Using Open Mind

• From OMCS Ontology
  – Subset of 15,000 sentences describe causation

• Examples of ontological relations
  – A consequence of bringing in a verdict is that the defendant is nervous.
  – Something that might happen when you act in a play is you forget your lines.
  – A consequence of eating too fast may be indigestion.
Agenda

• What’s in a story?
• Previous Approaches in Computational Stories
• An Approach using Commonsense Knowledge

• Knowledge Representation
• Flexible Inference
• MAKEBELIEVE Architectural Overview
• Generated story examples, and evaluation
• Conclusion and Future Work
Using transframes as a representation

- Transframes are two-slot frames which capture a change.
- There is a before and an after state, which are further decomposed into verb-object-modifier tuples.
- Example
  - “The effect of keeping things orderly and tidy is living a better life.”
  - VERB: “keep”
  - OBJ: “things”
  - MODS: (“orderly”, “tidy”)
  - EFFECT: “living a better life”
- We normalize OMCS causal knowledge into transframes using a broad coverage syntactic parser
Fuzzy inference

• We infer the connectedness of the EFFECT of a transframe to the CAUSE of another using lexical semantic resources

• Heuristics:
  – Scoring function on the semantic proximity of verbs, nouns and modifiers
  – WordNet nymic relations for nouns and modifiers
  – Levin verb classes commonality measure for verbs

• Not perfect, but helps overcome the brittleness of precise inference over such a (relatively) small dataset

• Also has the effect of lending creativity to the storyline
Agenda

• What’s in a story?
• Previous Approaches in Computational Stories
• An Approach using Commonsense Knowledge
• Knowledge Representation
• Flexible Inference

• MAKEBELIEVE Architectural Overview

• Generated stories & evaluation
• Conclusion and Future Work
1. The user enters seed sentence for the story
2. The sentence is parsed into verb-objs-modifier form to be compatible with transframes
3. Fuzzy inference matches this initial event to the CAUSE slot of a new transframe
4. The EFFECT slot of the new transframe is parsed and treated as the second story step.
5. After each step of inference, elements of the current story step are modified by analogy and synonymy (using lexsem resources). The intention is to introduce variation.
6. A global manager evaluates the chain of events to make sure it is free of cycles and contradictions. If necessary, it can backtrack to explore other storylines. Narrative goals are currently being added here.

7. In cases where the inference chain is completely stuck, users are asked to enter the next story step.

8. The user’s protagonist + frames of the inference chain + their corresponding sentences are used to generate the story text.
<table>
<thead>
<tr>
<th>Story #1 (length = 10)</th>
<th>Story #3 (length = 9)</th>
<th>Story #5 (length = 7)</th>
<th>Story #2 (length = 6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>John went shopping at the mall</td>
<td>John watched TV</td>
<td>Mary went to the zoo</td>
<td>David fell off his bike</td>
</tr>
<tr>
<td>John bought new clothes</td>
<td>John fell asleep</td>
<td>Mary learned about animals</td>
<td>David scraped his knee</td>
</tr>
<tr>
<td>John dressed better</td>
<td>John had a dream</td>
<td>Mary experienced enlightenment</td>
<td>David cried like a baby</td>
</tr>
<tr>
<td>John made a good impression on a girl</td>
<td>John dreamt about his day</td>
<td>Mary felt superior</td>
<td>David was laughed at</td>
</tr>
<tr>
<td>John bought items for her</td>
<td>John imagined he met a girl</td>
<td>Mary became a snob</td>
<td>David decided to get revenge</td>
</tr>
<tr>
<td>John spent a large amount of money</td>
<td>John went out with the girl</td>
<td>Mary was disliked</td>
<td>David hurt people</td>
</tr>
<tr>
<td>John needed to go to work</td>
<td>John had fun</td>
<td>Mary felt ashamed of who she was</td>
<td></td>
</tr>
</tbody>
</table>
Preliminary Evaluation

• 18 users were asked to judge creativity, quality, and coherence of 5 five-line stories each, generated as they interacted with the agent.

• Users were instructed to keep seed sentence simple. Stories that required user intervention were redone (on average, 2 restarts per user).

• Mean Scores:
  – Creativity: 4.1/5
  – Quality: 3.6/5
  – Coherence: 2.3/5
Agenda

• What’s in a story?
• Previous Approaches in Computational Stories
• An Approach using Commonsense Knowledge
• Knowledge Representation
• Flexible Inference
• MAKEBELIEVE Architectural Overview
• Generated stories & evaluation

• Conclusion and Future Work
Conclusions

- MAKEBELIEVE generates stories by casually chaining statements from a corpus of commonsense knowledge
- Fail-soft approach to story generation
- Demonstrates and helps to validate a new knowledge source, OMCS, used in a novel way (OMCS is a generic commonsense corpus)
- Commonsense brings valuable aspects of structuralism to story simulation
Limitations

- Ambiguity of OMCS representation makes it difficult to resolve bindings
  - Precludes multiple character stories at this time
- Without deeper semantic understanding of story step, chaining is brittle
- Realm of commonsense somewhat limits the genre of generatable stories
- Longer stories (>10 steps) reveal lack of plot devices. Users will intentionalize plot devices into shorter stories
Future work, pointers, etc.

- Try simulation approach using Cyc instead of OMCS
- With disambiguated word senses and references, it may be possible to involve multiple characters
- Add global constraints i.e. narrative goals, etc.
- For further papers on commonsense apps,
  - google for “hugo liu”