Humor as a Creative Search Problem

Lydia B. Chilton
University of Washington
Seattle, WA
hmslydia@cs.washington.edu

Daniel S. Weld
University of Washington
Seattle, WA
weld@cs.washington.edu

James A. Landay
Stanford University
Stanford, CA
landay@cs.washington.edu

ABSTRACT
Generating humor is a problem that has long been thought to be one of the hardest challenges for machine intelligence. It requires a type of intelligence that is considered essentially human: language ability, theory of mind, framing and reframing issues, logic, world knowledge and common sense. We believe that a promising first step towards computer-generated humor is to decompose the task into simple steps that humans can do independently.

We take the view of humor as a search problem that presently we need the creativity of humans or groups of humans to solve. To pose humor as search problem, we define the starting state and goal state and intermediate states we must find a path through. That path roughly maps to flare-and-focus creativity technique that is evaluated and performed iteratively until the goal is reached. The key step in the focus stage where we employ design patterns to find a path to a joke. We present our evaluation comparing crowd-generated humor to professional humor from The Onion. We discuss how to generalize generating humor to other forms of problem solving.

INTRODUCTION
Generating humor is a problem that has long been thought to be one of the hardest challenges for machine intelligence. It requires a type of intelligence that is considered essentially human: language ability, theory of mind, framing and reframing issues, logic, world knowledge and common sense. We believe that a promising first step towards computer-generated humor is to decompose the task into simple steps that humans can do independently.

To decompose humor we treat it as a search problem that has a starting state, a goal state, and a set of intermediate states that we must find a path through. To simplify the problem we narrow our goal of generating humor to generating a particular kind of humor: namely, a type of news satire that appears regularly in The Onion, a popular online humor site. The starting state is \is a real news headline, and the goal state is a joke in the form of a man-on-the-street style reaction to the headline. See Figure 1 for an example from the Onion.

Figure 1. Example of a headline (starting state) and a man-on-the-street style humorous response (goal state) from The Onion.

To inform our model of what the nodes on the search path would be, we did an extensive search of the humor literature. We incorporated definitions of humor from philosophers ranging from Plato who thought all jokes were insults to Daniel Dennett who proposes that jokes demonstrate violate our most deeply held assumptions. We also included the advice on the humor process from comedians ranging from Bob Hope to Judd Apatow. The advice mentions humor as containing truth and pain, and as rhetorical devices such as exaggeration and reversal of arguments. There are many diverse opinions of what humor is and how to generate it. Nobody has found "the answer" yet. Our position is not to decide who is right and who is wrong, rather to treat all the advice as being a valid or partially valid path through the space of jokes.

SEARCH PROCESS
The path from the starting state (a headline) to the goal state (a joke) is modeled after the flare and focus pattern from the creativity literature Both the flare and the focus stages are built out of smaller intermediate steps on the path from headline to joke. For simplicity, we will describe the steps as they were applies linearly. In reality, they are applied

iteratively. For example, when the focus stage fails to find a path to a joke, we understand what type of entities and valences we need to brainstorm in the flare stage, thus learning from iteratively applying the stages.

In the flare stage, the headline is decomposed into its individual entities and aspects; each part is analyzed for emotional valence, for potential reasons for those emotional reactions. For example, if use the headline in Figure 1 as the starting state we can then apply various intermediate states in the flare stage as follows:

1) **Decompose the headline** into its entities and aspects. Example, *Entities: Uber, Panic Button, Passengers, Chicago, Chicago Passengers. Aspects: adding a feature to an app, adding a safety feature to a product, panic etc.*

2) **Determine a valence** of each entity and aspect, and a reason. Example: *adding a safety feature to a product is positive because it helps keep customers safe and it shows the company cares about the passenger's safety*

3) **List associated entities or aspects** that are closely tied to entities and aspect in (1) and that have a strong valence, and list the reason if not obvious. Examples: *panicking (related to 'panic button') is negative, Uber drivers (related to Uber) can be negative because drivers have been accused of harming passengers.*

Note that all the tasks in these intermediate states are currently outside the reach of machine intelligence but people are able to do them quickly and easily.

In the focus stage, we look for connections between the outputs of the flare stage and find ways to connect them. This is difficult because there are many paths from many entities and aspects generated in the flare stage. To reduce the search space we introduce design patterns for jokes. Design patterns are paths from the output of the flare stage (the "entities" we will call these), to a joke that follows intermediate states that humans can easily apply such as:

1. **Reverse** an expectation. For example, if you expect the reaction to an article to be positive, find something negative to say about it.
2. **Exaggerate** an aspect. Exaggerate something negative to heighten the clarity of the argument, exaggerate a positive entity potentially to find a contradiction.
3. **Point of View**. Who would agree with this?
4. **Negative Point-of-View**. Who would disagree with this?
5. **Reuse** information in a different context

For example, consider this headline: "*New Smartphone Dating App 'The League' Matches Elite Professional Singles."* An obvious associated entities is "people who would use this app." This is also a leads to a **positive point of view** we could use: an elite person, which has a negative valence. If we **exaggerate** the negative valence, we can make the point of view an elitist. We expect an elitist to like this app. Thus we **reverse** that expectation and find a reason for them to not like the app. What would that reason be? Here, we want to **reuse** the fact that they are an elitist. We **exaggerate** that in a negative way. Maybe it's not elite enough for them. The Onion makes two jokes with that design pattern:

1. "How elite are we talking? They aren't allowing Vassar alums in, are they?"
2. "How can people who don't have a personal matchmaker honestly call themselves elite?"

This is an example of our design pattern called "Point-of-View Reversal" because we use a point of view that has the reverse of the expected reaction, and then justify it. Two other design patterns we employ are "Tone Reversal Contradiction" and "Lens Shift." The Uber driver joke from Figure 1 is a Tone Reversal Contradiction. It first identifies a contradiction – that the panic button is both positive and negative. It is positive because it means the company cares about passenger safety, but it is negative because it indicated there is a safety problem they have been sweeping under the rug. To bring both those aspects out at once the speaker starts in a positive tone ("*It's nice knowing…*") and sharply veers into a blatantly negative tone ("[*not] properly screening its drivers.*").

The three design patterns we identified cover 70 of the 84 Onion jokes we analyzed. We do not expect them to cover all of them. Design patterns are not meant to be perfect solutions, only to guide the user to a solution to a problem. We can deviate from them as the circumstance demands it. There are many paths through the search space of jokes. Design patterns are just a heuristic to cut down the search time.

EVALUATION

Our evaluation is currently in progress. Early pilot evaluations show we are 37% as funny as The Onion. We expect this number to increase dramatically as we continually improve the training and examples (particularly of the design patterns), as we optimize the number of iterations to perform, and as we incorporate more diverse opinions and ways of seeing. We evaluate our work by asking people to read a headline and six humorous responses: three are generated from our search process, three are from The Onion. Each is judged separately, and we can count how many times The Onion's responses are funny and compare it to the number of times our responses are funny. Moreover, by analyzing the instances where The Onion is funny and we are not, we can iteratively improve our design patterns and identify missing states in our search space.