1. Title:
   1. Decomposing Humor: An Iterative Process for Generating News Satire
   2. Decomposing Humor: Crowdsourced Problem Solving
   3. Crowd-Created Humor: A Case of Decomposing Creative Problem Solving
2. Abstract
3. Introduction
   1. Paragraph 1 – Setting the stage: Why is this hard, why is this important?
      1. Crowdsourcing goes places that computers cannot go: language ability (transcription, translation), common sense and world-knowledge (image labeling, categorization), emotional comprehension (sentiment labeling).
      2. The biggest of these is humor. It uses all these human abilities.
      3. Language – say things without saying them
      4. IN ADDITION to these human abilities that crowdsourcing has already tackled is the new challenge: Understanding of the audience – theory of mind
         1. Unstated assumptions, must know what other people will assume.
         2. Reveal in surprised fashion – must know how to incept surprise in people.
   2. Important: language understanding problem.
      1. Good Example of problem solving and intelligence
      2. Generating News Satire is a challenge, and potentially commercially viable. Prevalent – Jon Stewart, The Onion. But that’s not why we’re taking on the challenge.
      3. It’s a language generating problem that is short (unlike a novel), so we can iterate faster, and is semi-evaluatable. Hard to pose language as a maximization problem. Jokes can be. Perhaps that makes them too distinct for real language problems. Hopefully comedy is just a complex form of communication, perhaps skewed to certain emotions like surprise and skewed away from more dramatic emotions like greif.
         1. Who is better Shakespeare, Hemingway, or Tom Clancy?
            1. It’s hard to get an objective measure.
      4. Non-literal
      5. Emotional processing
      6. Linguistic processing
      7. By generating rather than analyzing we hope to bring insight into
      8. Generating and Understanding are related but not the same. Children do both. Generating is harder, but has more flexibility.
      9. Entities: yes, it involved entities
      10. Refers to them in Oblique ways.
      11. Meaning – sarcasm – we mean the opposite of what we say. Meaning and syntax are divorced.
4. Contributions
   1. Decomposition of a hard task.
   2. Iterative process driven by Decision Theory?
      1. Extensible
      2. Discussion of aspects generalizable to other domains.
         1. Iterative design in applicable to everything ☺
   3. (Some result)?
5. Related Work
   1. Humor (What is it?) Philosophy
      1. Historical: Comedy and Drama
         1. King’s Problems, and nobel fights
         2. Ordinary people’s problems. Not so nobel. Much more about getting laid.
         3. Comedy takes second fiddle in today’s society.
         4. Perhaps because when chosing between the joke and the moral, comedy usually picks the moral.
      2. Satire
      3. Defining humor
         1. Hard, possibly impossible as an evoluationary process
         2. Philosphers: Many many things identified as correlated with or causing humor: long list. Insult (Plato), (Surprise)Hume, Lens (Kant, Schopenhauer), Assumption Violation (Dennet and others), Two Scripts (Raskin and others),
         3. Practitioners:
            1. Comedy is Truth and Pain
            2. Winning (don’t need this?)
            3. Iterative Process
            4. Study other comedians so you can write for them
            5. Inspiration is wonderful, but you can’t depend on it. (Greg Dean). They do use processes that decompose humor. Thus the idea of decomposing humor is sanctioned by the experts. We are not perverting their art (in this way.)
         4. Daniel Dennett and Raskin (Dennett argues his theory is a more general case)
         5. We don’t take sides – between philosophers and practitioners. We see the truth in all of these. We have taken bits of philosophical and practical wisdom and applied them where they have served us.
      4. Humor involves emotion. Dennet says we think we our emotions, and this plays a part.
   2. Creativity
      1. Iterative?
      2. Crowd Creativity
         1. Niki’s Analogies
         2. Joy’s work on Motif and thing with Mira
   3. NLP work in Humor, sentiment?
      1. Puns (which are not humor!)
      2. Dan Jurafsky – high level analysis?
   4. Peer Assessment & Self Assessment
      1. Rubrics
   5. Crowdsourcing
      1. Using experts?
      2. Flash Crowds (iteration at the heart of the process) – also Cascade, although not the heart of the algorithm. Also while loops in TurKit.
6. Lessons
   1. People can’t just list assumptions (and follow 2-script theory)
      1. Break down that process
   2. People do better with Responding to stuff, than cold generation (Mindstorms)
   3. Learning while doing. This is a property of life. Not just this algorithm?
   4. Hard Analysis problem. (Early meteor examples)
      1. Very low agreement, many aspects being judged at once.
      2. Try to break things into single aspects at a time.
         1. This jives with modern learning theory of teaching one skill at a time. IXP, Duolingo, etc.
7. Problem
   1. **Our goal is to generate humor on par with The Onion.** (At least 50% as funny as the Onion).
   2. Obviously, humans are capable of generating humor. Unlike other human skills like transcribing audio, it is highly challenging to do it on demand, every time, and to do it subjectively well. We propose not an algorithm, but **a process**.
      1. Nobody is ever stuck.
      2. Fights fixation (when it is bad)
      3. Provides inspiration when it lacks
   3. Decompose not to the level that a computer can do it, but to a level that humans can do it. Do one thing at a time.
   4. Iterative process we need to have something akin to Design, Prototype, Evaluate
      1. Evaluate jokes for funniness, evaluate assumptions as shared, evaluate logic as sound
      2. Design – come up with parts of utterances, in reaction to SOMETHING. Designs are a reacting. If you’re designing a better X, you’re reacting to existing X.
      3. Prototyping is about learning from your designs (utterances).
         1. Do other people agree with your assumptions?
         2. ?????????
8. Solution
   1. Describe the Process
      1. Box and pointer Diagram of all the pieces
      2. Screen shots of the UI
      3. UI design decisions that were hard
9. Deployment
   1. Many iterations? Done over two weeks?
   2. Groups of 3?
   3. Review other people’s work?
   4. Analysis for learning?????
10. Results
    1. What things are important to test?
11. Discussion
    1. Generalizable?
       1. To other forms of humor
          1. Other sections of The Onion:
             1. Harder to compare – what are they reacting to?
          2. Less linguistic humor
       2. To other forms of communication
          1. Understanding your audience
       3. Yes in terms of iterative process
       4. Yes in terms of there being some break down
       5. Which steps are important?
       6. The “Design Patterns” will change. Probably will change.
          1. Hardest part. We found this breakdown (which is by no means perfect,) as a quote from Charlie Chaplin about humor, taking the theory of emotions as all having valence, and Dennett’s Epistemic Emotions.
          2. Every field has design patterns.
             1. Civil Engineering, and oh yeah, Software. There are even patterns in how people write research papers, beyond just the section headlines.
             2. Many researchers have patterns for introductions.

Start with why the problem is hard, and important.

For related work, there is the differenting yourself from competing papers so reviews won’t ding you for that.

For citing and referencing how you work fits into to larger research agendas, so they think you are relevant and have a frame for evaluating your work.

* + - * 1. Step one of transferring to the new domain would be an intense literature search for design patterns. I do not know of a way of formulating this problem better, decomposing it or automating it. Nevertheless, it my philosophy that anything can be broken down. Believing anything else is tantamount to magic.
        2. Step 2 is testing those design patterns against data. Literature is wrong. Didn’t realize they were in a special case, not totally broken down.
        3. A design pattern is only good if it’s useful for something. Test it on people.
      1. Also studying the Onion to get frequencies, and dependencies.
    1. The Goal will change

1. Conclusions