Relation extraction for commonsense causal reasoning

Ben Hixon
University of Washington
CSE 573

Goal: commonsense cause-and-effect reasoning

- We want to answer questions like

  Premise: The man lost his balance on the ladder.

  What happened as a result?

  Alternative 1: He fell off the ladder.

  Alternative 2: He climbed up the ladder.

- Choice of Plausible Alternatives (COPA) Task: 1000 cause and effect questions in this format
  - Humans: 99% accuracy
  - Best performing algorithm: ~65% accuracy
  - Random choice: 50% accuracy
  - *There's room for improvement*

More examples

The farmland needed irrigation. A canal was constructed. A flood occurred.	The man hated his new haircut. He wore a hat. He grew a beard.
My favorite song came on the radio. I covered my ears. I sang along to it.	The woman won the lottery. She bought a yacht. She joined a church.

Proposed method: use relation extraction

- Relation extraction finds the semantic relations in text
- Open IE finds semantic relations in open-domain free text
- Input=“`The homeowners disliked their nosy neighbors.`”
  Output=“`0.76: (The homeowners; disliked; their nosy neighbors)`”

- Hypothesis: *relation pairs that more frequently co-occur in a large text corpus will be more causally connected*

- If `(I; poured; coffee),(I; added; milk)` co-occurs more often than `(I; poured; coffee),(I; voted for; Obama)`, then first pair more causally connected

Solve low recall with relation pair similarity score

- Open IE has high precision but low recall:
  - The relations it extracts are usually correct
  - But it misses out on a lot of true relations
- Probably won't find relations identical to the COPA relations

Hypothesis: a relation pair is also causally connected if *similar* relation pairs frequently co-occur in a large corpus

Example:

If

`(Princess Di; was; famous)` & `(The press; chased her)`

frequently co-occur, then

`(the woman; became; famous)` & `(photographers; followed; her)`

should also be causally connected

Methodology

1. I ran a relation extractor, OLLIE, on all the COPA questions

2. I obtained results for OLLIE run on a subset of the Gigaword corpus (for time constraints I used only 2500 of the 1.2 million articles)

3. I preprocessed each relation by lower-casing and removing stop-words so `(The homeowners; disliked; their nosy neighbors) -> (homeowners; disliked; nosy neighbors)`

4. I found all co-occurring giga relation pairs for which one relation was similar to a copa premise and the other relation in the pair was similar to one of the copa alternatives. Co-occur defined as occuring within +-2 sentences.

5. For each COPA question, I chose the alternative for which more similar relations co-occured with the premise

2 simple similarity scores (for now)

- Relation Similarity Score #1:
  $[arg1,pred,arg2]_1 \sim [arg1,pred,arg2]_2$ if each corresponding element pair has a single word in common

- Relation Similarity Score #2: (Slow!)
  $[arg1,pred,arg2]_1 \sim [arg1,pred,arg2]_2$ if in each corresponding element pair there exists words with a high enough wordnet similarity score

Results

1000 articles gave 55,748 distinct relations

- SS #1: 17 questions answerable, 53 % accuracy

2500 articles gave me 142,374 distinct relations

- SS #1: 22 questions answerable, 59% accuracy

Restricted to the first 5000 relations:

- SS #3: 10 questions answerable, 6/10 correct

  (and it still had to run overnight)

In the coming week

- More data: Increase number of relations

- Better pre-processing of relations: stemming (drinking -> drink), replace named entities with their class, e.g.
  'Princess Di' -> 'female',
  'John'-> 'male',
  'Boeing' -> 'business' or 'company'.
  (Does there exist available software to do this?)

- More relation similarity score functions.

- Improve speed? (Python not the fastest language; Profile code)

- Can I cluster the relations using unsupervised learning?

Wordnet similarity score

- Wordnet: words arranged in synsets, groups of synonyms, hypernyms (of a), and hyponyms (has a)

- Wu-Palmer Similarity: Return a score denoting how similar two word senses are, based on the depth of the two senses in the taxonomy and that of their Least Common Subsumer (most specific ancestor node).