The Text-Minder

Babysitting Your Language Model by Feeding Semantic Constraints

Presenter : Jerry Ding (dalud)

High Level Overview

Goal:

To create a framework for conditional dialogue generation based around a simple “semantic constraint” language.

Broad Approach:

1. Design the SC language syntax & atoms
2. Engineer an automated system to produce semantic constraint graphs from dialogue lines
3. Train transformer model to infer dialogue from context lines & semantic constraint graphs

Investigative Work

Manual inspection of OpenSubtitles data reveals several characteristics of conversational dialogue:

- Frequent and unpredictable changes in subject matter
  - Hard to predict response even with many lines of context
- Non-standard / informal language very common
- Vocabulary from huge variety of topic clusters
  - Scanning text for a small set of keywords doesn't get very far
- Lines too short for wordsense disambiguation
- Less use of compositional structure / complex syntax
- “Many entities, sparsely connected”
  - As opposed to texts with many coreferences and complex storylines

Desiderata for SC Language

1. Human Interpretability
   - A human, or an independently designed expert system, supplies the constraints. Uninterpretable binary data is useless.
2. Isolating (hence non-lexical)
   - Each constraint is a meaningful unit of semantic information. Omitting even few words from a sentence destroys meaning, so semantic constraints go beyond the word / token level.
3. Simple yet Expressive
   - Easier to reason in SC space for domain specific task, and generated sentences are reasonably close to the desired result.

Current Prototype of SC Language

- Only three data types: nouns, keywords, tags

- Nouns are the fundamental building block. Constraints can only anchor to nouns; SC language cannot fill slots with verb phrases / adverbial phrases / other constraints.

- In other words, a sequence of constraints defines a graph: nouns are vertices and each constraint is an edge.
  - Keywords and tags augment the edges. Pairs of vertices can be connected by multiple edges. Self-loops are allowed.

Current Prototype of SC Language

- As verbs / adjectives / adverbs are not allowed, they are represented indirectly by a bag of tags representation.
- A tag is a basic concept from a small vocabulary of 140. They are linked by loose association, not IS-A relationships.

- The manual process used to choose the 140 tags:
  1. An upper ontology was made such their descendants have fair coverage of nontrivial words in the corpus, then the ontology words are tagged.
  2. The vocabulary was clustered based on GLoVe vectors and tags were picked based on the content of the clusters.
  3. The above two steps disfavor common verbs, so the top 50 verbs were added. These are only used directly, not with other tags.

The full set of tags

act
air
alive
analysis
animal
artifact
bad
big
biology
body
change
cold
control
covering
create
damage
data
dead
defense

distance
easy_simple
effort
emotion
emphasis
energy
entertainment
event
expertise
family
female
fiction
fight
financial
food
give
good
group
help

house
intense
leader
legal
less
level
location
love
medical
military
more
move
multiple
name
nature
outfit
ownership
person
piece

politics
quality
quantity
question
receive
religion
resource
rest
say
school
science
sense
situation
small
society
sound
stability
substance
thought

time_period
time_point
touch
trait
use
value
vehicle
vision
water
ask
become
begin
believe
bring
call
come
do
feel
find

give
go
happen
have
hear
help
hold
keep
know
leave
let
like
live
look
lose
make
mean
move
need

provide
put
run
say
see
seem
show
sit
start
take
talk
tell
think
try
turn
use
want
work
write

Constraint types

- **`RELEVANT(noun, n_tags)`**
  - Introduces a noun. Tags relate to the noun's intrinsic meaning.

- **`DESCR(noun, kind, d_tags, parity)`**
  - Appends a modifier to the noun. Modifier is described via tags.
  - Can be an adjective (**`kind=DIRECT`**), an adverb for its actions (**`kind=ACTION`**), a verb for a relative clause (**`kind=THROUGH_ACT`**), etc.

- **`QUESTION`**: The sentence asks a question.

- **`UTTERANCE`**: The sentence makes an interjection.

Constraint types

- `ACTION(subj, act, obj, parity, isreal)`
  - Describes an action or experience. Either the subject or the object can be left empty with keyword `[NONE]`. If the slot is not empty but no noun is a good fit, `[CMPLX]` is used.
  - `act` can be tags for verbs, or be one of `[BE]`, `[POSSESS]`, `[ACT_FOR]`, `[ACT_LINK]`, `[LINKED_TO]`, `[UNK]`
  - `isreal` can be `REAL`, `HYPO`, or `TOPIC` (i.e. the action is part of a subclause whose realness is irrelevant).

Using automation

- In principle, sentences *could* be manualled labeled.
- But without an army, cannot get a big dataset from this...
  - The labeling task is pretty complicated
  - May need multiple annotators / line to cover all bases
  - Language models train best with millions of sentences

- Automatic labeling:
  - Will be fast and can be more comprehensive
  - Will have errors or glaring omissions
  - Can attach semantic information to new text on the fly
  - Hope to have useful “signal to noise ratio”

Automatic tag generation

To generate the tags for a target word

1. Upper ontology words are sorted based on WordNet search distance (edge distances are hand-tuned)
2. The tags are taken from the top ontology words until a distance cap or # of tags cap is reached
3. Any tags associated with assigned GLoVe cluster are added
   - These clusters are conservatively labeled; clusters with ambiguous semantic meaning have no tags attached

– Different parts of speech for the same spelling are handled as separate words, and will have different tags

– The outputs are noisy; to increase recall the system appends more tags than what one would manually pick

Tags example

- If I manually labeled them:
  - **Hunter**: person, expertise, animal, food, name
  - **Radioactive**: science, bad, substance
  - **Appoint**: give, receive, control, leader

- Generated by automated system:
  - **Hunter**: use, person, time_period, animal, name, society, artifact
  - **Radioactive**: bad, distance, less, hot, science, good
  - **Appoint**: use, receive, say, leader, control, artifact, change

- Note: People (and algorithms) likely disagree on tags, and tags are not independent so similarity between tags is relevant.

Automatic SC graph generation

To generate the semantic constraints:

1. Sentences are syntactically parsed via AllenNLP's dependency parser.
2. The parse tree is recursively visited, and semantic constraints are added to a list based on many hand crafted rules.
3. The non-nouns in the constraints are replaced with tags.

Semantic constraints example

Sentence: **I want you to examine it for explosives, all right?**

- If I manually labeled it (w/o tags):
  - `RELEVANT[each noun],`
  - `ACTION[I, want, [CMPLX], POS, REAL],`
  - `ACTION[you, examine, it, POS, TOPIC],`
  - `ACTION[you, [ACT_FOR], explosive],`
  - `QUESTION`

- Automated system matches my guess (!)

SC Conditioned Language Model

- Language model uses transformer networks.
- Two components: an encoder and a decoder
  - Both are causal like GPT-2, unlike BERT.
  - Semantic constraint graph is converted into a sequence of embeddings, with knowledge of previous SC graphs in context.
  - Each token in line is generated with knowledge of previous tokens and previous SC graphs embeddings.
- Semantic constraints and tags are shuffled + randomly dropped out during training to assist with generalization.

SC Conditioned Language Model

- Still training. Latest checkpoint has perplexity 3.8
  - With dropout still active in SC graphs and neural net layers
  - Unclear whether this improves or worsens if higher quality SC graph is used (without fine-tuning), both are plausible
  - Hasty comparisons vs. unconditioned models:
    - 2016 SOTA for OpenSubtitles has perplexity 17
    - Back-of-napkin estimate: perplexity >10 for tuned medium GPT-2
  - Only trained on one epoch ... the data set is huge!

Example sentence + SC graph

They're geeks.
The reason no one's here is because they can all talk to each other without even speaking.
I know why I got shot.
Harvey Wratten needed to get himself out of jail.
[To do that, he got Gatehouse to buy up a huge amount of drugs and pass them on to Customs.] (Original)
[To do that, he got John to buy up a huge amount of drugs and pass them on to the guards.] (Customized)

(RELEVANT, he), (RELEVANT, john),
(DESCR, amount, DIRECT, huge, POS), (RELEVANT, amount),
(RELEVANT, drug), (RELEVANT, they), (RELEVANT, guard),
(RELEVANT, that),
(ACTION, he, pass, they, POS, REAL),
(ACTION, amount, [LINKED_TO], drug, POS, REAL),
(ACTION, [NONE], do, that, POS, REAL),
(ACTION, john, buy, amount, POS, TOPIC),
(ACTION, he, get, [CMPLX], POS, REAL),
(ACTION, he, [ACT_LINK], guard, POS, REAL)

Recall, the network only sees the verbs or adjectives above as bags of tags, not as words.

Some generator outputs

[To do that, he got John to buy up a huge amount of drugs and pass them on to the guards.]

He lost the entire amount of drugs that had been given to the guards.

If John was paying them off, he got them to buy you a whole amount of drugs that were later found in the civil servant's wrist.

He got away from the guards, done the whole amount of drugs, and made big bucks laundering the money to the fbi and the johns and the sex repels that went along with them.

He passed a huge amount of drugs to the wrong guards, and Garland Raysshaw, tried to broker a deal that did that.

It wasn't hard to find these, and they look surprisingly good. There's also many outputs that were not as good (they still generally discussed drugs as a relevant part of an escape from prison).

The above sentences generally only satisfy a subset of the specified constraints. But they clearly are aware of the non-noun constraints (e.g. huge amounts, paying, he got [CMPLX], etc.)

Future work

- Get estimates for a model without SC graphs, and for a model with only nouns as input
- Find a quantitative measure of output variability given fixed context and SC graphs
- Study the system in various qualitative ways
  - Difficult for a purely quantitative metric to capture many of the desiderata for the system
- Improve automatic tagging / find a better representation
  - I think tag replacement is the most lossy step in the system