Data Recombination for Neural Semantic Parsing

Presented by: Edward Xue

Robin Jia, Percy Liang

Intro

- Semantic Parsing: The translation of natural language into logical forms
- RNNs have had much success recently
  - Few domain specific assumptions allows them to be generally good without much feature engineering
- Good Semantic Parsers rely on prior knowledge
  - How do we add prior knowledge to an RNN model?

Sequence to Sequence RNN

- Encoder
  - Input utterance is a sequence of words: $x = x_1, \ldots x_m \in V^{(in)}$
  - Converts to sequence of context sensitive embeddings: $b_1, \ldots, b_m$
  - Through a bidirectional RNN
    - Forward direction: $h_i^{\mathrm{F}} = \mathbf{LSTM}(\phi^{(\mathrm{in})}(x_i), h_{i-1}^{\mathrm{F}})$
  - Each embedding is a concatenation of the forward and backward hidden state

Sequence to Sequence RNN

- Decoder: Attention based model
  - Generates output sequence one token at a time: $y = y_1, \ldots y_n \in V^{(out)}$
  -

$$s_1 = \tanh(W^{(s)}[h_m^{\mathrm{F}}, h_1^{\mathrm{B}}]). \quad (2)$$

$$e_{ji} = s_j^\top W^{(a)} b_i. \quad (3)$$

$$\alpha_{ji} = \frac{\exp(e_{ji})}{\sum_{i'=1}^{m} \exp(e_{ji'})}. \quad (4)$$

$$c_j = \sum_{i=1}^{m} \alpha_{ji} b_i. \quad (5)$$

$$P(y_j = w \mid x, y_{1:j-1}) \propto \exp(U_w[s_j, c_j]). \quad (6)$$

$$s_{j+1} = \mathbf{LSTM}([\phi^{(\mathrm{out})}(y_j), c_j], s_j). \quad (7)$$

Attention Based Copying: Motivation

- Previously just chose next output word using a softmax over all words in the output vocabulary
- Does not generalize well for entity names
- Entity names often correspond directly to output tokens: eg “iowa” -> iowa

Attention Based Copying

- At each time step j also allow the decoder to copy any input word directly to the output, instead of writing a word to the output

$$P(a_j = \texttt{Write}[w] \mid x, y_{1:j-1}) \propto \exp(U_w[s_j, c_j]), \tag{8}$$

$$P(a_j = \texttt{Copy}[i] \mid x, y_{1:j-1}) \propto \exp(e_{ji}). \tag{9}$$

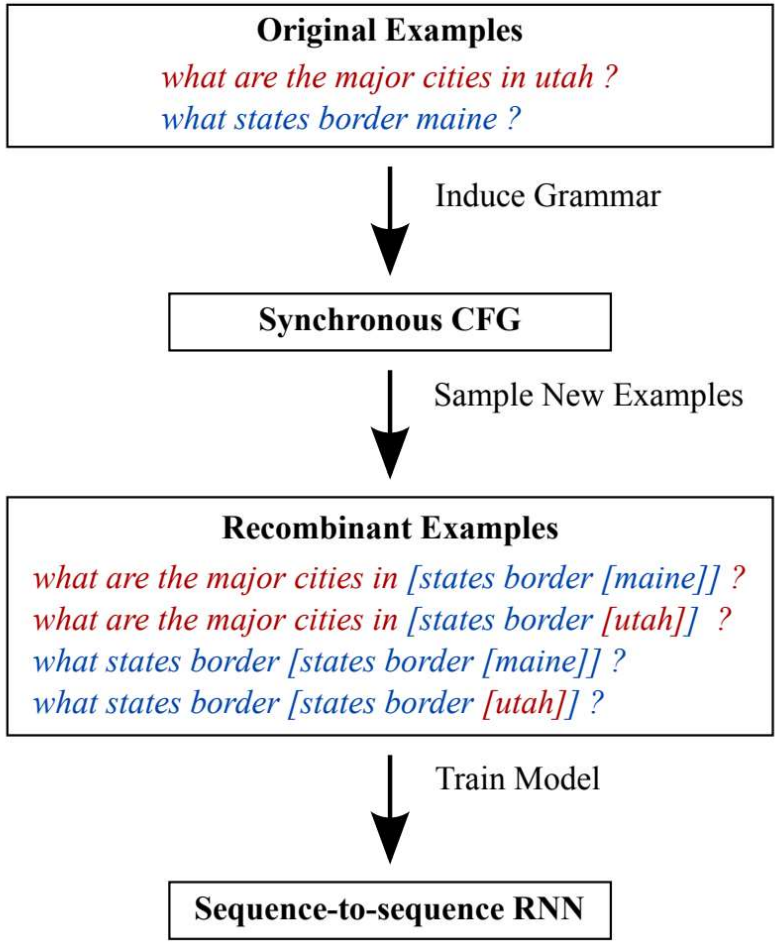
Attention Based Copying Results

	GEO	ATIS	OVERNIGHT
No Copying	74.6	69.9	76.7
With Copying	85.0	76.3	75.8

Table 1: Test accuracy on GEO, ATIS, and OVERNIGHT, both with and without copying. On OVERNIGHT, we average across all eight domains.

Data Recombination

- This framework induces a generative model from the training data
- Then, it samples from the model to generate new training examples.
- The generative model here is a Synchronous CFG



Data Recombination

Start with training set D of (x, y) pairs.

$\hat{p}(x, y)$ is the empirical distribution

Fit a generative model $\tilde{p}(x, y)$ to $\hat{p}$ which generalizes beyond with recombinat examples

To train model, maximize $E[\log p_\theta(y \mid x)]$

Data Recombination

- Synchronous CFG
  - Set of Production rules $X \rightarrow \langle \alpha, \beta \rangle$
- The generative model is the distribution over the pairs (x,y) defined by sampling from G
- SCFG is only used to convey prior knowledge about conditional independence structure
- Initial grammar generated as $ROOT \rightarrow \langle x, y \rangle$

Data Recombination: Grammar Induction Strategies

- Abstracting Entities
  - Abstracts entities with their types
- Abstracting Whole Phrases
  - Abstracts both entities and whole phrases with their types
- Concatenation
  - For any k >=2, CONCAT-K creates two types of rules
  - ROOT going to a sequence of k SENT's
  - Then for each ROOT -> <α,β> in the input grammar, add rule SENT-> <α,β> to the output grammar

Examples

(“*what states border texas ?*”,
`answer(NV, (state(V0), next_to(V0, NV), const(V0, stateid(texas))))`)
(“*what is the highest mountain in ohio ?*”,
`answer(NV, highest(V0, (mountain(V0), loc(V0, NV), const(V0, stateid(ohio)))))`)

Rules created by ABSENTITIES

ROOT → ⟨ “*what states border* STATEID *?*”,
`answer(NV, (state(V0), next_to(V0, NV), const(V0, stateid(`STATEID `))))`⟩
STATEID → ⟨ “*texas*”, `texas` ⟩
ROOT → ⟨ “*what is the highest mountain in* STATEID *?*”,

```
answer(NV, highest(V0, (mountain(V0), loc(V0, NV),
                        const(V0, stateid(STATEID ))))))⟩
```

STATEID → ⟨“*ohio*”, `ohio`⟩

Rules created by ABSWHOLEPHRASES

ROOT → ⟨ “*what states border* STATE *?*”, `answer(NV, (state(V0), next_to(V0, NV), STATE ))`⟩
STATE → ⟨ “*states border texas*”, `state(V0), next_to(V0, NV), const(V0, stateid(texas))`⟩
ROOT → ⟨ “*what is the highest mountain in* STATE *?*”,
`answer(NV, highest(V0, (mountain(V0), loc(V0, NV), STATE )))`⟩

Rules created by CONCAT-2

ROOT → ⟨SENT$_1$ `</s>` SENT$_2$, SENT$_1$ `</s>` SENT$_2$⟩
SENT → ⟨ “*what states border texas ?*”,
`answer(NV, (state(V0), next_to(V0, NV), const(V0, stateid(texas))))` ⟩
SENT → ⟨ “*what is the highest mountain in ohio ?*”,
`answer(NV, highest(V0, (mountain(V0), loc(V0, NV), const(V0, stateid(ohio)))))` ⟩

Datasets

- GeoQuery (GEO): questions about US geography paired with answers in database query form. 600/280 split.
- ATIS: queries for a flight database paired with corresponding database queries. 4473/448 split
- Overnight: Logical forms paired with natural language paraphrases over eight different subdomains. For each domain, random 20% as test, the rest split into 80/20 training/development set

Experiments: GEO and ATIS

	GEO	ATIS
Previous Work		
Zettlemoyer and Collins (2007)		**84.6**
Kwiatkowski et al. (2010)	88.9	
Liang et al. (2011)[2]	91.1	
Kwiatkowski et al. (2011)	88.6	82.8
Poon (2013)		83.5
Zhao and Huang (2015)	88.9	84.2
Our Model		
No Recombination	85.0	76.3
ABSENTITIES	85.4	79.9
ABSWHOLEPHRASES	87.5	
CONCAT-2	84.6	79.0
CONCAT-3		77.5
AWP + AE	88.9	
AE + C2		78.8
AWP + AE + C2	**89.3**	
AE + C3		83.3

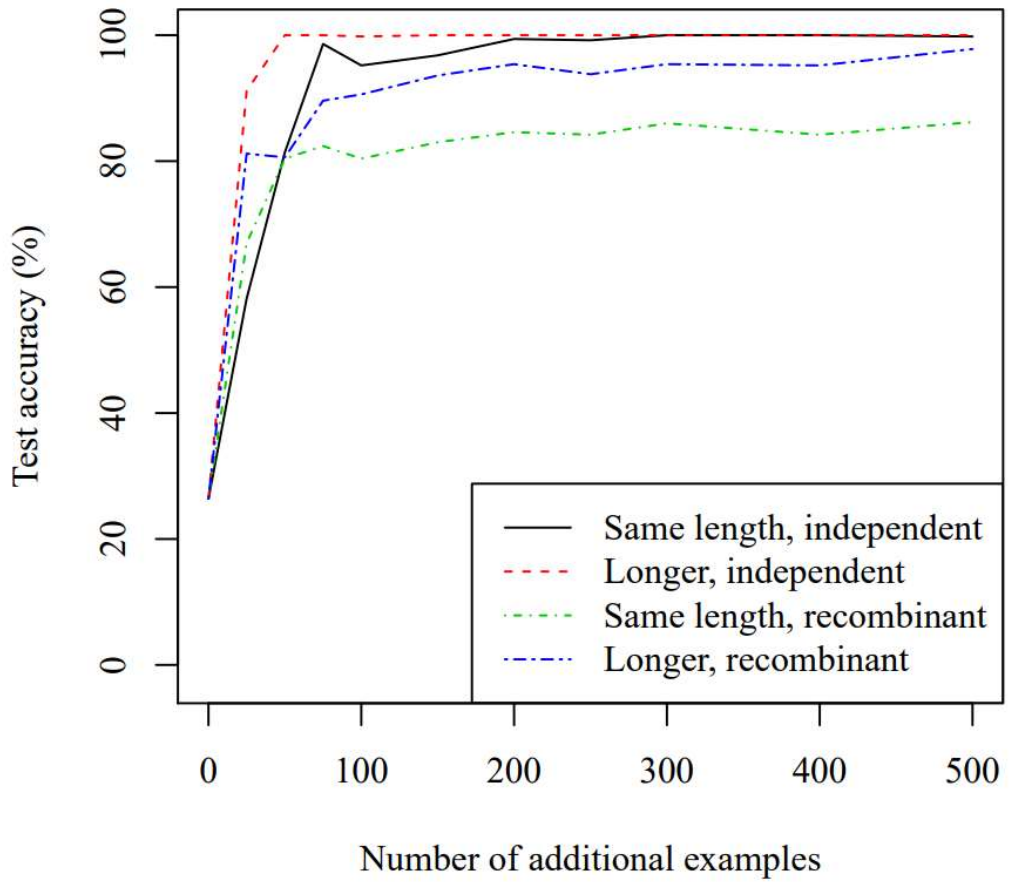
Table 2: Test accuracy using different data recombination strategies on GEO and ATIS. AE is ABSENTITIES, AWP is ABSWHOLEPHRASES, C2 is CONCAT-2, and C3 is CONCAT-3.

Experiments: Overnight

	BASKETBALL	BLOCKS	CALENDAR	HOUSING	PUBLICATIONS	RECIPES	RESTAURANTS	SOCIAL	Avg.
Previous Work									
Wang et al. (2015)	46.3	41.9	74.4	54.0	59.0	70.8	75.9	48.2	58.8
Our Model									
No Recombination	85.2	58.1	78.0	71.4	76.4	79.6	76.2	81.4	75.8
ABSENTITIES	86.7	60.2	78.0	65.6	73.9	77.3	79.5	81.3	75.3
ABSWHOLEPHRASES	86.7	55.9	79.2	69.8	76.4	77.8	80.7	80.9	75.9
CONCAT-2	84.7	**60.7**	75.6	69.8	74.5	80.1	79.5	80.8	75.7
AWP + AE	85.2	54.1	78.6	67.2	73.9	79.6	**81.9**	**82.1**	75.3
AWP + AE + C2	**87.5**	60.2	**81.0**	**72.5**	**78.3**	**81.0**	79.5	79.6	**77.5**

Table 3: Test accuracy using different data recombination strategies on the OVERNIGHT tasks.

Experiments: Effects of longer examples



Conclusions

- Data Recombination seems to provide better test accuracy in lieu of more training examples
  - Would this generalize well?
- Attention Based Copying is useful for certain datasets

Thank you