A Conversational Neural Language Model for Speech Recognition in Digital Assistants

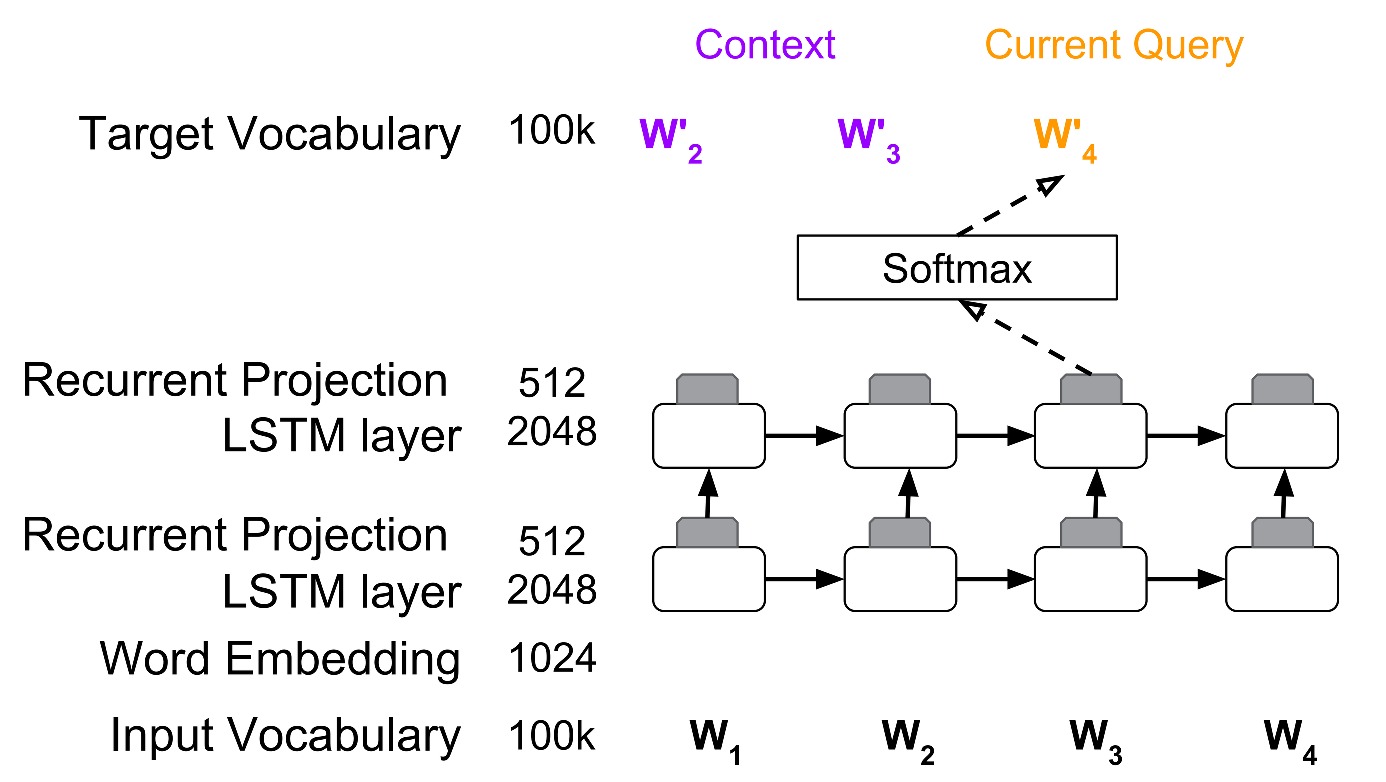
Eunjoon Cho and Shankar Kumar

1. Introduction

- Conversations with digital assistants are centered around topics
- Can a recurrent neural network language model make use of context in a conversation to improve ASR?
- Prior work in modifying network architectures to incorporate the speaker turn/context
- This work:
  - Given the previous queries with/without agent responses, can we improve the language model for the current query?
  - Uses a standard LSTM language model architecture
  - Achieves a 4% relative WER reduction on Google Assistant

2. Conversational Context LSTM LM

$$P(w_1, w_2, \ldots, w_T|C) = \prod_{i=1}^{T} p(w_i|w_1 w_2, \ldots, w_{i-1}, C)$$



3. Types of Contexts

- Queries Only: Each query is a treated as a sentence
- Query History: Prepend each query with previous 2 queries spoken within 5 minutes, separated by turn boundaries
  What is the weather today? <t>
  How about tomorrow ? <t>
  Will it be windy ?
- Query History with Agent Responses: 3 queries with agent responses within 5 minutes, separated by turn boundaries
  What is the weather today? <t>
  It is cloudy with a high of 55 and a low of 32 <t>
  How about tomorrow ? <t>
  It will be sunny with a high of 60 and a low of 40 <t>
  Will it be windy? <t>
  Yes, it will be windy with 16 m/hr winds coming from the west

4. Speech Recognition setup

- Training data: Anonymized queries/responses from *Google Assistant* in US English
  - 16.9B tokens from sequences with responses and 6.3B tokens from sequences without responses
  - LSTM LM has a vocabulary of 100k tokens
- LSTM LM rescoring on lattices generated using a 5-gram LM
  - 2nd pass interpolation weight of 0.5
  - LSTM LM initialized using tokens from previous queries with/without the agent responses
  - Previous queries are from the ASR output to simulate an actual system
- Test sets
  - Testset A has 16k tokens sampled from *Google Assistant* traffic
  - Testset B is a subset of Testset A with 12.6k tokens with exactly 2 previous query/response pairs per utterance

5. Do previous queries help?

Model	Testset A	Testset B
No context	11.9	12.5
w/ query context	11.6	12.2

- Using previous queries improves recognition
- Gains are mostly from question answering type conversations
- Common corrections are acoustically confusable words: *two/too*.
- If previous query includes a number, the contextual model prefers a number

6. Do previous responses help?

Model	Testset A	Testset B
Only queries	11.6	12.2
Queries + Responses	11.5	12.1

- Wins on short queries such as *no*, where agent response is useful
- Question words (e.g. *what*) had more errors wrt baseline
- Hypothesis: Model is trained on both queries/responses and sees less proportion of question words than baseline trained on queries only
- Two approaches to address the mismatch:
  - Restrict LSTM LM vocabulary to words from queries only
  - Add a recency bias for queries by presenting responses first followed by queries.
    $\text{query}_1, \text{response}_1, \text{query}_2, \text{response}_2, \text{query}_3, \text{response}_3$
    $\Rightarrow \text{response}_1, \text{response}_2, \text{response}_3, \text{query}_1 \text{query}_2, \text{query}_3$

Model	Testset A	Testset B
Vocab from queries	11.6	12.3
Priority on queries	11.4	12.1

- Restricting the vocabulary to queries does not help
- Recency bias on queries helps!

6. Conclusions

- Strategies for training a standard LSTM LM on conversation data from a digital assistant
- Experimented with a variety of inputs for training the model
- Obtained a 4% relative improvement in error rate on *Google Assistant*