Verifying the Long-range Dependency of RNN Language Models

Tzu-Hsuan Tseng, Tzu-Hsuan Yang and Chia-Ping Chen

National Sun Yat-sen University

IALP2016 @NCKU Nov 2016

Overview

- Introduction
- Language Model
  - N-gram
  - Recurrent Neural Network (RNN)
  - N-gram + RNN
- Evaluation measure
  - Perplexity
  - Word Prediction Accuracy
- Experiments
- Results
- Conclusion

Introduction

- Language Model (LM)
  - Probability distribution over sequences of words
  - Well-known LMs
    - N-gram
    - Recurrent Neural Network Language Model (RNN LM)
- Compare N-gram model with RNN LM
  - Perplexity
  - Word prediction accuracy
- Analysis on different word position

Language model

- N-gram
- Recurrent neural network (RNN)
- N-gram + RNN

N-gram

- Estimate probability of each word given preceding $N - 1$ words
- Estimated by relative frequency

$$p(w|w_1, \dots, w_{k-1}) = \frac{Count(w_1, \dots, w_{k-1}, \ \ w)}{Count(w_1, \dots, w_{k-1})}$$

- Predict the word by the greatest conditional probability of words

Recurrent Neural Network (RNN)

- Contain input layer, hidden layer and output layer
- An additional loop at the hidden layer

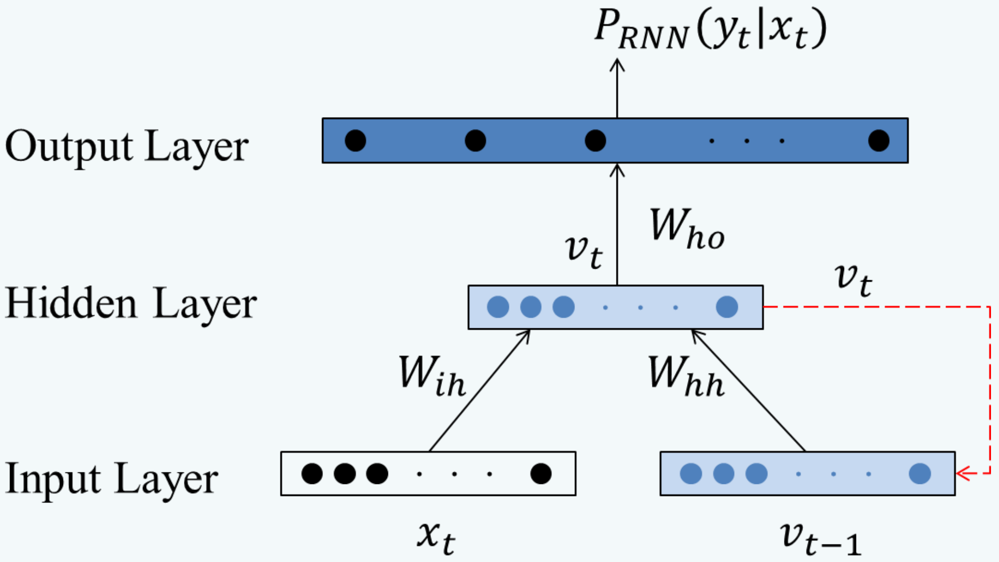


Figure 1. RNN Architecture

Recurrent Neural Network (RNN)

- Suitable for sequential data
- Use One-hot representation in input layer
- Neuron output corresponds to the probability of the word

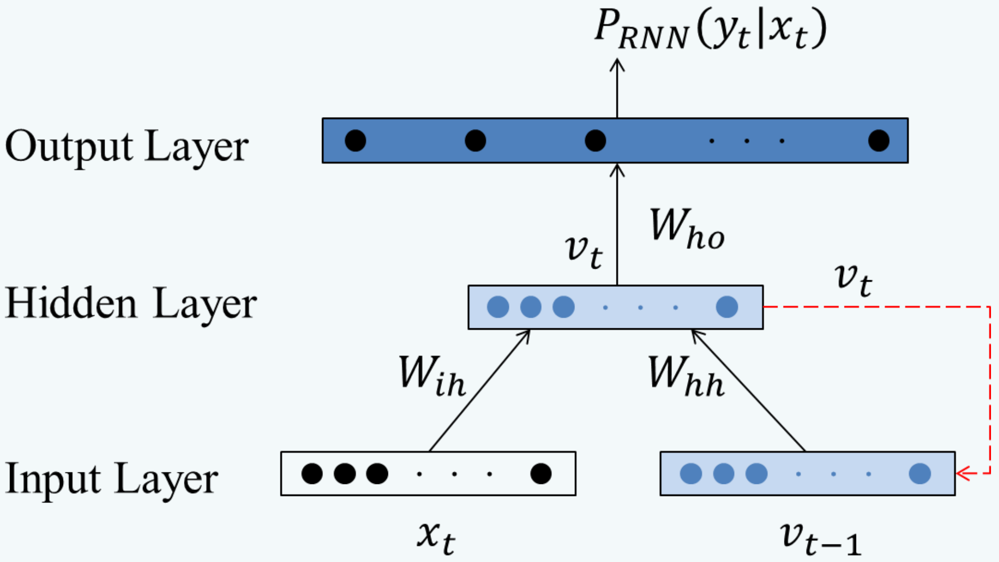


Figure 1. RNN Architecture

N-gram + RNN

- Strength of interpolation method
  - good context coverage
  - strong generalization
- Combine the probability of the RNNLM with N-gram model
- The interpolated LM probability :

$$p(w_i|h) = \lambda \cdot p_{ng}(w_i|h) + (1 - \lambda) \cdot p_{rnn}(w_i|h)$$

Evaluation measure

- Perplexity
- Word prediction accuracy

Perplexity

- Perplexity is an evaluation measure for language models
- A low perplexity means that the model is good at predicting words

$$PPL = p(D|M)^{-\frac{1}{N}}$$

$p(D|M)$: data likelihood
N : number of words
D : text set
M : language model

Word Prediction Accuracy

- Use the greatest probability word as predicted word
- Compare the predicted word with the actual word
- Calculate the number of accurate words

$$Accuracy = \frac{correct\ prediction\ of\ word}{Number\ of\ word}$$

Experiments

Experiments

Datasets

- Penn Tree Bank(PTB)
- AMI meeting corpus(AMI)

Dataset	Sample sentences	Vocabulary size	Number of words	
PTB	now the field is less <unk> he added there is no asbestos in our products now	9999	train	887521
			validation	70390
			test	78669
AMI	OKAY YEAH UH MAYBE TO AS UH IT	11883	train	802824
			validation	94953
			test	89666

Table 1. Sample sentences and statistics of the datasets

Experiments

Evaluation of word position p

- Use only probability of word position p in the sentence rather than entire text to calculate results

- Use the subset of the test set, with sentence of length at least p

Word position	Testing data
4	no it was **n’t** it ‘s also **costly** some circuit breakers **installed**
5	no it was n’t **black** some circuit breakers installed **after**

Table 2. Illustration of word position

Experiments

System Implementation

- N-gram
  - trigram model
  - KN smoothing
- RNN LM
  - 1 hidden layer
  - 200 hidden units
- Interpolated model
  - Weight : 0.5

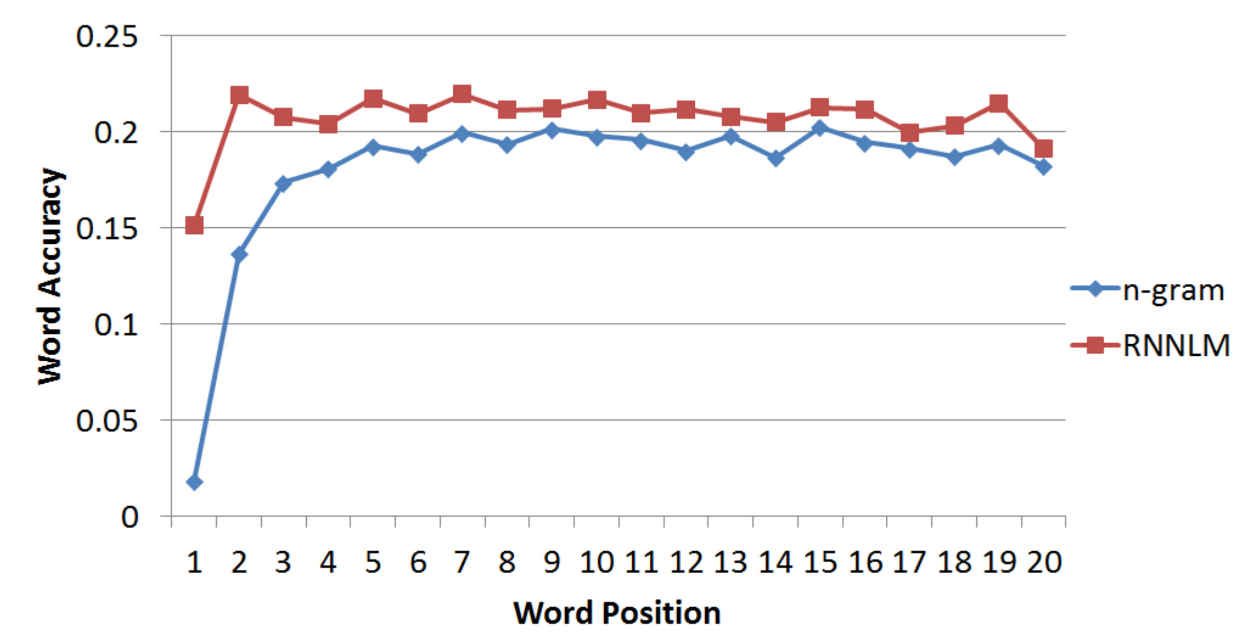
Results

Results

Word Prediction Accuracy

PTB

AMI



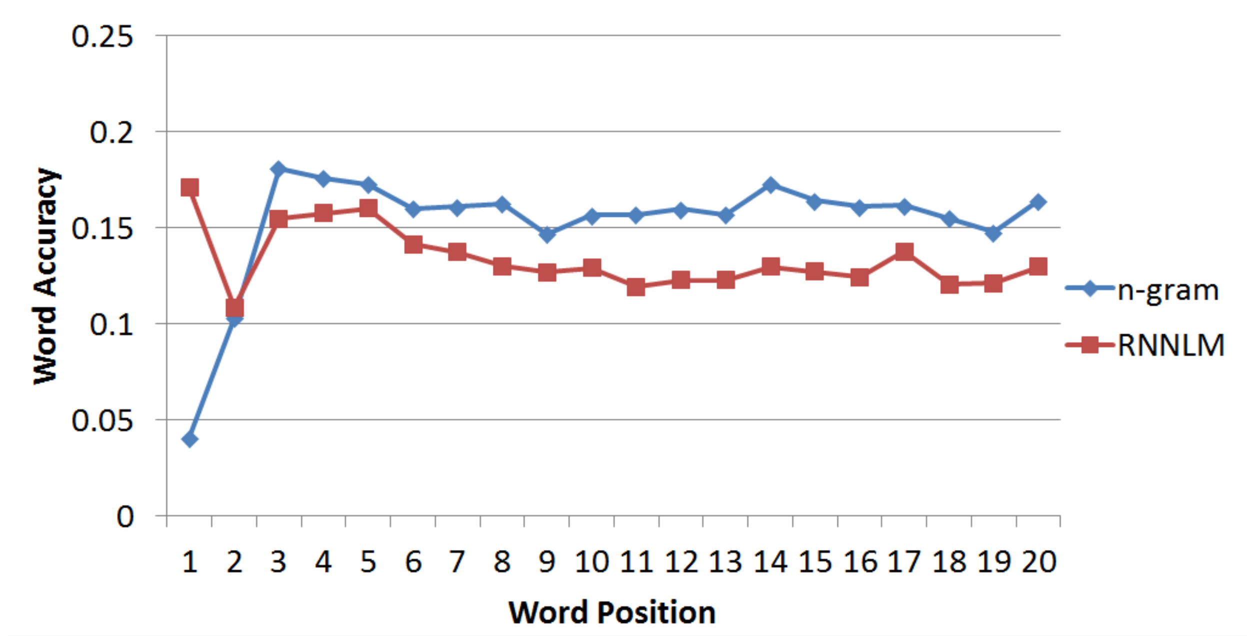
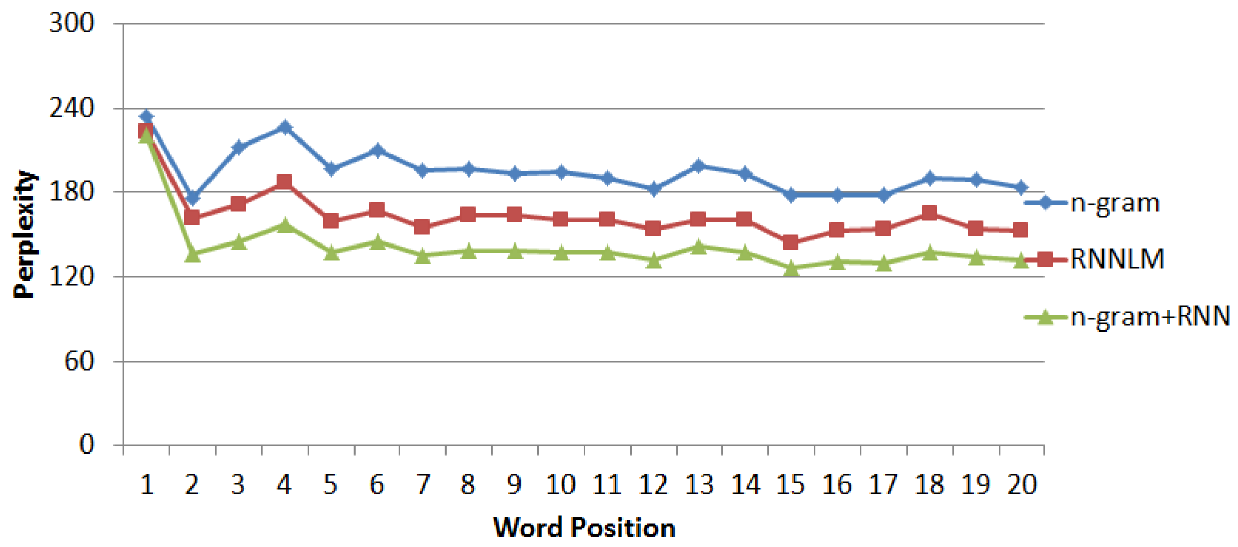


Figure 2. Word prediction accuracy against word position

Results

Perplexity

PTB



AMI

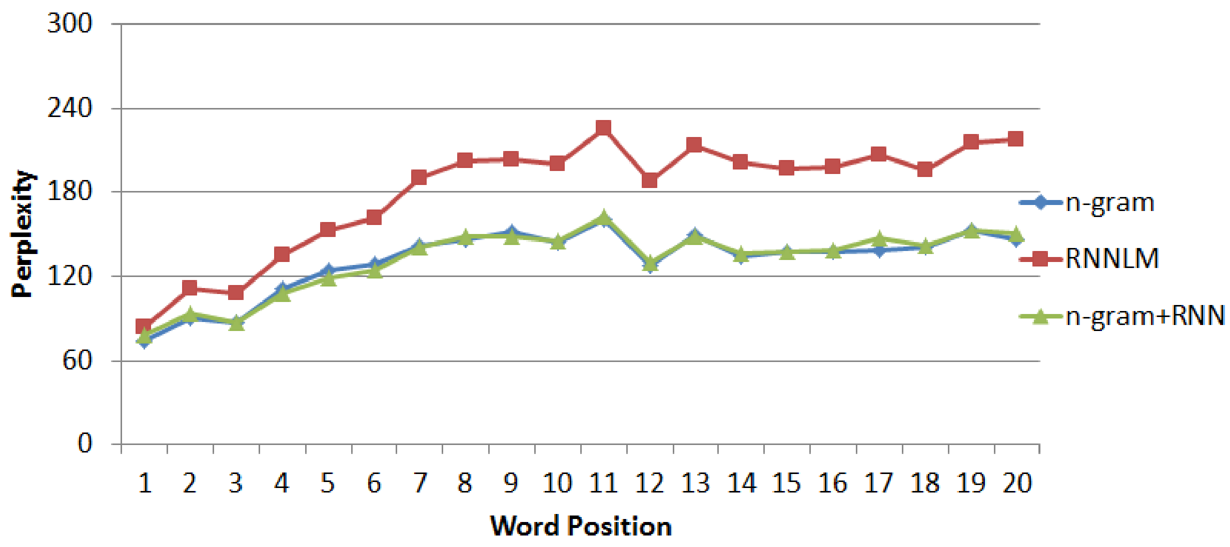


Figure 2. Perplexity against word position

Conclusion

- RNNLM always get better performance than n-gram in PTB, but it is opposite in AMI

- PTB contains written sentences, and AMI contains colloquial sentences

- RNNLM may be affected by data property and lead to worse performance than n-gram