Michael NLP Literature Review

Natural Language Inference

Facebook Infersent

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Decomposable Attention Model for Natural Language Inference

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Question Generation

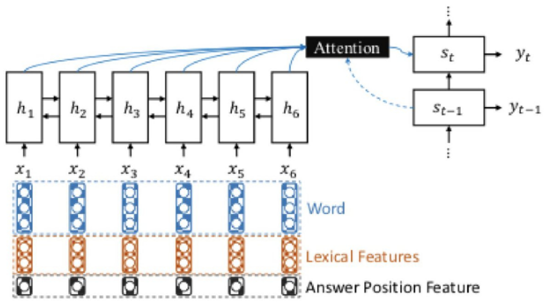
Neural Question Generation from Text, A Preliminary Study

Input:
Text sentence(not the passage or paragraph) and Answer position (answer-aware input representation)

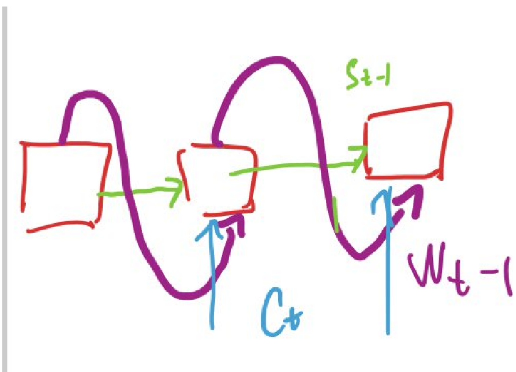
Output:
Generated Answer-aware question

Architecture:

Lexical Features:
NER and POS
Answer Position Feature:
BIO (begin, continue, not included in the answer)



For the decoder part, each RNN cell should look like this



w: word vector *selected from last step*
s: hidden state
c: context vector by attention

This is essentially a generative language model.
They use a *copy mechanism* in decoder which generates a probability p of copying a word from source sentence in order to cope with unknown words.

Evaluation Metrics:

1. BLEU-4 Score (13.29%)
    a. More on Bleu Score: ⚓Bleu Score: Michael Machine Learning Notes
2. Precision and Recall for question type
    a. types: when...what...how...
    b. Precision: for all generated T type question, how many of them are actually T?

Conclusion:
Three things that largely improve performance (by ablation analysis):

1. The use of answer position
2. The use the lexical features
3. The use of copy mechanism

Implementation:
https://github.com/magic282/NQG/blob/master/seq2seq_pt/s2s/Models.py

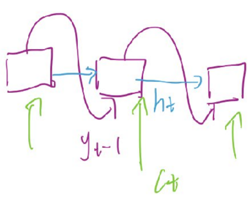
My Thoughts:
This model is easy to understand and can serve as a good baseline model.
Inspired by the question type metric, I think maybe we can train different models for different types. (type is limited, 4~5)
One problem is that we may also need to provide answer positions during inference time.

Learning to Ask: Neural Question Generation for Reading Comprehension

Input:
For **basic model**, the input is just the sentence that contains the answer
For **paragraph-level model**, the inputs are the sentence and the truncated paragraph containing it
Sometimes the answers may span 2 or more sentences, we just concatenate the sentences.
Preprocessing: CoreNLP - tokenization, lower case and sentence splitting

Output:
Generated question

Architecture:
Encoder is just a Bi-LSTM sentence encoder. Suppose the generated vector of the sentence is s and generated vector of the truncated context paragraph is s'.
For decoder,



Goal: find $\hat{y} = \arg\max_{y} P(y|x)$

$$P(y|x) = \prod_t P(y_t | x, y_{<t})$$

$$P(y_t | x, y_{<t}) = \text{softmax}(f(h_t, c_t))$$

where f is just W matrices

$$h_t = LSTM(y_{t-1}, h_t)$$

- Initialization: we initialize the first hidden state of the decoder using "s" for basic model and "s + s'" for paragraph-level model.
- Beam search is utilized: ⚓Beam Search: Michael Machine Learning Notes
- To cope with UNKs, we copy the word in the source sentence with the highest attention score at this step

Evaluation Metrics:

1. BLEU Score
2. METEOR and ROUGE

Side notes:
They randomly sample some sentence-questions pairs from the test set and categorize them to 4 categories:

- w/ sentence: only requires the sentence tot answer the question
- w/ paragraph: requires the sentence and context paragraph tto answer the question
- w/ article: requires the sentence and entire article to answer the question
- no ask-able: maybe annotation error, cannot answer

Implementation:
Original Implementation in Torch/Lua: https://github.com/xinyadu/nqg
Tensorflow implementation: https://github.com/yanghoonkim/neural_question_generation

My Thoughts:

Inspired by the motivation part of rule-based method vs NN method: is BLEU score a syntactic measurement instead of semantic? If so, how about applying our question matching (between generated and reference sentences) model as one of the performance metrics for Question generation?

A Generator-Evaluator Framework for Question Generation

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Reinforcement learning Perspective

- Some basics: ⚓Reinforcement Learning: Michael Machine Learning Notes
- Reinforcement Learning with NLP

Word Coverage Mechanism

- Motivation: TODO
- More Details on their own paper: Modeling Coverage for Neural Machine Translation

Resources:
Implementation:

Question Answering

Stanford Attentive Reader

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Stanford Attentive Reader ++

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Bidirectional Attention Flow for Machine Comprehension

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:
https://github.com/jojonki/BiDAF

Dynamic Coattention Networks For Question Answering

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

QANet

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

FusionNet

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Text Generation

Tutorial on Variational Autoencoders

Input:
Output:
Architecture:

Evaluation Metrics:
Side Notes:
Resources:
[Highly Recommend!] University of Waterloo Deep Learning Lecture by Dr. Ali Aghosi
Implementation:

Generating Sentences from a Continuous Space

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

A Hybrid Convolutional Variational Autoencoder for Text Generation

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Chatbot

Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Building Dialogue Using Generative Hierarchical Neural Network Models

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

A Persona-Based Neural Conversation Model

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Deep Reinforcement Learning for Dialogue Generation

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

A Diversity-Promoting Objective Function for Neural Conversation Models

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Adversarial Learning for Neural Dialogue Generation

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Learning through Dialogue Interactions by Asking Questions

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Dialogue Learning With Human-In-The-Loop

Input:
Output:
Architecture:
Evaluation Metrics:
Side Notes:
Resources:
Implementation:

Classic NLP

Attention Is All You Need

Motivation:
Traditional models for NMT task are usually many-to-many RNNs. There are two drawbacks here:

1. Long term dependency problem
2. The sequential property precludes parallelization

Input:
sentence from source language
Output:
sentence from target language
Architecture:
Transformer Explanation. TODO
Side Notes:
Resources:

1. A great blog post on Attention mechanism
2. Soft vs Hard attention
3. Local vs global attention