Semi-Supervised QA with Generative Domain-Adaptive Nets

Zhilin Yang Junjie Hu Ruslan Salakhutdinov William W. Cohen
School of Computer Science
Carnegie Mellon University

Motivation

Constructing the training data by semi-supervised method.

Utilize <p,a> to generate **q**.

A Simple Baseline

Giving SQuAD data, regard the context of answer as the question. The simple baseline method leads to substantical improvements when labeled data is limited.

Generative Domain-Adaptive Nets

- Discriminative Model
- Generative Model

Discriminative Model

Gated-attention reader model

Domain Adaptation with Tags

During training the discriminative model, the training data is from two distributions (human-generated and model-generated). These two domains of the data should be treated differently. So it uses a **domain tag** as an additional input to he discriminative model.
d_true represent the domain of human-generated data, **d_gen** represent the domain of model-generated data.
At test time, using the **d_true** tag.

Generative Model

seq2seq model + copy mechanism.
During using label data, MLE. P(q|p,a)
During using unlabel data, RL. P(a|p,q)

Attention: During using unlabel data training generateive model, in order to prevent the generated question close to the distribution of the answer, it appends the **d_true** tag to the generated questions. By this way, the discriminative model regrads them as human-generated question to treat, and make the generated questions close to the distribution of human-generated questions.

Training

Training progress

Algorithm 1 Training Generative Domain-Adaptive Nets

Input: labeled data L, unlabeled data U, #iterations T_G and T_D
Initialize G by MLE training on L
Randomly initialize D
while not stopping **do**
  for $t \leftarrow 1$ to T_D **do**
    Update D to maximize $J(L, \text{d\_true}, D) + J(U_G, \text{d\_gen}, D)$ with SGD
  end for
  for $t \leftarrow 1$ to T_G **do**
    Update G to maximize $J(U_G, \text{d\_true}, D)$ with Reinforce and SGD
  end for
end while
return model D

ps. Two modules are updated alternately.

The objective function

$$max_D J(L, d_true, D) + J(U_G, d_gen, D)$$
$$max_G J(U_G, d_true, D)$$

Model architecture and training