Question Generation for Question Answering

Nan Duan†, Duyu Tang†, Peng Chen§, Ming Zhou†
Microsoft Research Asia†
Microsoft Research and AI Group§

Contributions

- Two types of question generation approaches are proposed
- It shows how to leverage the generated questions to improve existing question answering systems

Goals

- The training data should need few or no human efforts and reflect commonly-asked question intentions(acquiring large scale high-quality training data from Community-QA (CQA) website)
- The questions generated is normal sentence (two question generation models)
- The generated questions be helpful to QA tasks (integrate QG approach into an end-to-end QA task)

Question Generation

- Question Pattern Mining
- Question Pattern Prediction
- Question Topic Selection
- Question Ranking

Question Pattern Minging

question pattern: [w1,w2,...,#,...,wL]
Mining the frequently-asked question patterns from CQA.

Approaches:

- question cluster
- enumerate all valid n-gram (<=7) as question topic candidates.Then assign an score (word-frequency based)
- For each question cluster, select the one with the highest importance score as the question topic, and remove it from each question to form a question pattern.
- It calls each removed question topic as a historical question topic of its corresponding question pattern.

Ranking	Question Pattern	Historical Question Topics					
1	how to #	boil eggs	ombre hair	taxidermy	freeze potatoes	photoshop	...
2	who is #	henry ford	sandusky	sally ride	john glenn	pericles	...
3	what is #	eosinophils	carbohydrates	mitochondria	cilia	bacteria	...
...	...	...					
528	who is wife of #	steve jobs	moses	bin laden	roy orbison	charles stanley	...
...	...	...					
584	who is author of #	hunger games	ben hur	naruto	seabiscuit	snow white	...
...	...	...					

Question Pattern Prediction

Training data construction

For each QA pair<Q,A>, if Q can be matched by a frequently-asked question pattern Q_p, and the matched question topic Q_t of Q based on Q_p exists in A, create it as traing instance $<A, Q_p, Q_t>$.
ps. A is S.

Retrieval-based QP Prediction

Input: <S, Q_p>

Output: score

Count the similiarty of S(A) and the candidate Q_p, then ranking for the candidates.

Implements

Att-CNN encoding

Generation-based QP Prediction

Implements

Att Seq-to-Seq Model

Inference

Beam search to output the top-N question patterns

This approach can generate question pattern out of the candidate pattern set.

Question Topic Selection

For Q_p from retrieval-based method

Candidates generation:(1) Entities as question topic candidates, which are detected on FreeBase.(2) Noun phrase as question topic candidates, which are detected based on Stanford Parser.

Counting the score for each candidate question topic:

$$s(Q_t, Q_p) = \frac{1}{N} \sum_k \#(Q_p^{t_k}) dist(v_{Q_t}, v_{Q_P^{t_k}})$$

$Q_P^{t_k}$ denotes the k^{th} historical question topic of Q_p.

For Q_p from generation-based method

Supposing the placeholder # is the i^{th} word in Q_p , then selecting the j^{th} word in S as the question topic, which satisfies the following constraint:

$$w_j = \arg\max_{w_{j'} \in \mathcal{S}} \alpha_{ij'} = \arg\max_{w_{j'} \in \mathcal{S}} \frac{exp(e_{ij'})}{\sum_{k=1}^{|\mathcal{S}|} exp(e_{ik})}$$

ps.Limitation: only one word

Question Ranking

For each candidate generation question.

Features:

- question topic selection score
- QA matching score
- word overlap between Q and S
- question pattern frequency

All features are combined a linear model:

$$p(Q|S)=\sum\limits_{i}\lambda_i\cdot h_i(Q,S,Q_p,Q_t)$$

Question Generation for QA

$$\hat{A} = argmax_A P\{(A|Q) + \lambda QQ(Q, Q_{max}^{gen})\}$$

$$QQ(Q, Q_{max}^{gen}) = argmax_{i=1,2,\ldots,10} sim(Q, Q_i^{gen}) p(Q_i^{gen})$$

Motivation

The questions generated from correct answers are more likely to be similiar to labeled questions than questions generated from wrong answers.