Task: Short Text Conversation (STC)
Team: USTC
System architecture:
- Lexical features: Query-Response Similarity, Query-Post Similarity, Transition-p2c
- Semantic features: EncDec-Forward model, DecDec-Reverse model, Joint-Train model
- Ranking: linear RankingSVM
Results: 0.2867 on Mean nDCG@1, 0.4509 on Mean P+, and 0.4181 on Mean nERR@10

We model this task as learning-to-rank problem, and classify the features into two categories: lexical features and semantic features.

Lexical Features
- Query-Response Similarity
  - Map query and response to their own TF-IDF score vector
  - Calculate cosine similarity between query and response
- Query-Post Similarity
  - Calculate cosine similarity between query and post
- Transition-p2c
  - Model the transition probability between post words' vector and response words' vector
  - Example:
    - Query: Where do we eat?
    - Query-Response: We eat two apples.
    - Query-Post: Where do we go? - We go to the bank.
    - Transition-p2c: Restaurant. (Transition probability between “Where” and “Restaurant”, “eat” and “Restaurant” is higher than normal.)
  - Algorithm:

  Algorithm 1 Transition-p2c Train
  Input: post, rep, post, rep, comment
  Output: transition matrix T
  1: Word segmentation, and get words' vector of post and comment
  2: Initialize: T = zeros(m, n), m = length of post vocabulary set, n = length of comment vocabulary set
  3: IDF score of post set and comment set
  4: for (p, c) in (post-word vector, comment-word vector) do
    5: Get tf-idf score vector: p = tf-idf, c = tf-idf
    6: T = T + p * c
  7: end for
  8: Normalization: for i in [0, m], normalize T[i]

- EncDec-Forward model
  - Motivated by the work in [Vinyals et al. 2015], [Shang et al. 2015] and [Bahdanau et al. 2014], we use the seq2seq model to estimate P(Response | Post).

  Algorithm 2 Transition-p2c Test
  Input: test, query, rep, comment
  Output: transition score
  1: Initialize: score = 0, K = zeroes(m, n)
  2: Get tf-idf score vector of test, query and comment
  3: K = query tf-idf * c tf-idf
  4: for (p, c) in (K, h) do
    5: score = score + K[m][n] + T[m][n]
  6: end for

  Transition-p2c Case Study

- Joint-Train model
  - Generation & Reconstruction: combine P(Response | Post) and P(Post | Response)
  - Decoder1 is regarded as both a decoder and an encoder (Encoder2).

- We use linear RankingSVM to merge all the scores and output a final score for each query and response pair.

Experiments
- We submit 5 runs:
  - USTC-C-R1: Query-Response Similarity + Query-Post Similarity + EncDec-Forward + EncDec-Reverse + Transition-p2c
  - USTC-C-R2: Query-Response Similarity + Query-Post Similarity + EncDec-Forward + EncDec-Reverse + JointTrain
  - USTC-C-R3: Query-Response Similarity + Query-Post Similarity + EncDec-Forward + Transition-p2c
  - USTC-C-R4: Query-Response Similarity + Query-Post Similarity + EncDec-Forward + EncDec-Reverse
  - USTC-C-R5: Query-Response Similarity + Query-Post Similarity + EncDec-Forward

Results:
- Official Results
- Offline Training Set Results

- Transition-p2c Case Study

Conclusions
- The results in training set and cases show the efficiency of the models we proposed.
- The online evaluation is inconsistent with the offline evaluation because of the subset selection problem.