Recurrent Neural Network Encoder with Attention for Community Question Answering

Wei-Ning Hsu, Yu Zhang

MIT CSAIL, Cambridge, MA, U.S.A.

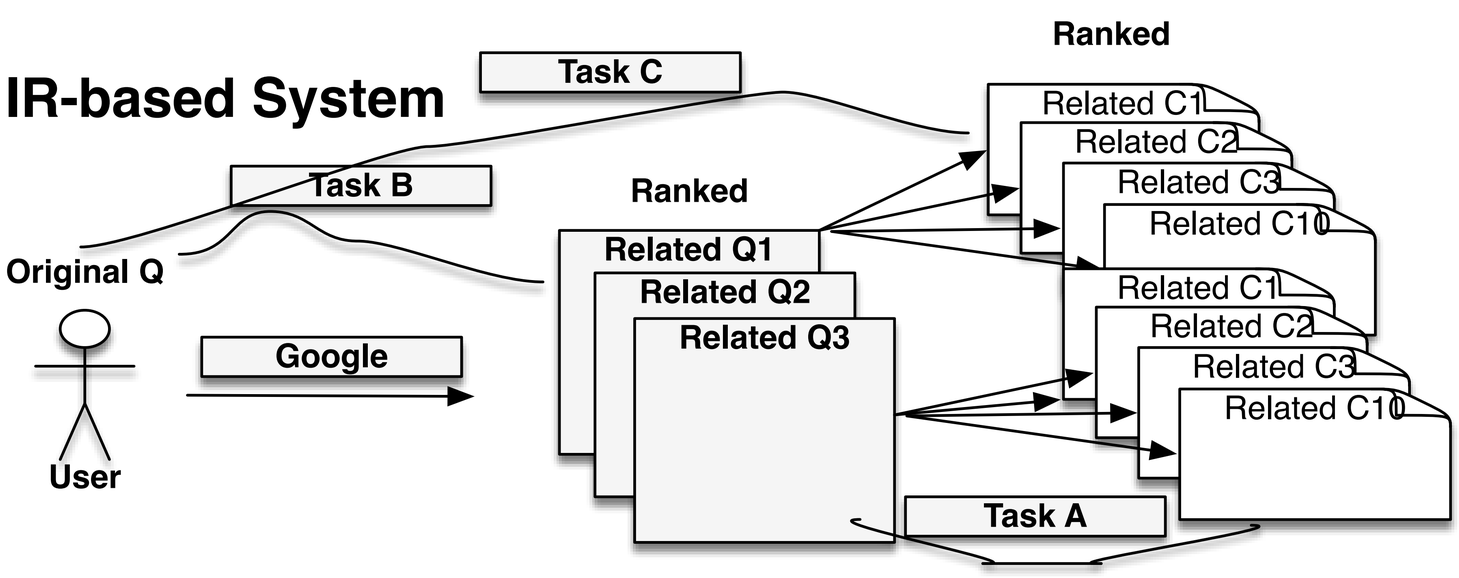
Task of CQA

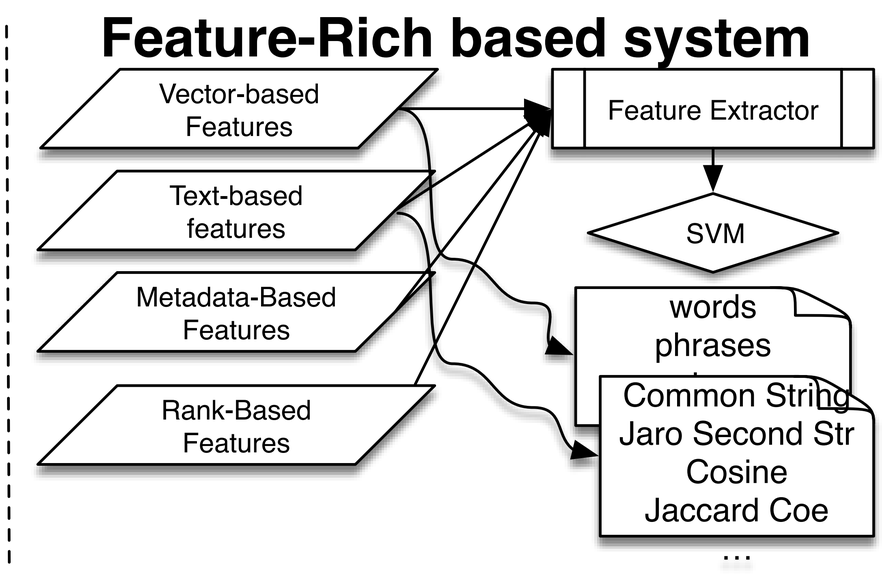
- Given a new question and a large collection of question-comment
- Rank the comment according to its relevance to the new question
  - Task A: Question-Comment Relevance (train 26690, test 5000)
  - Task B: Question-Question Relevance (train 2660, test 500)
  - Task C: Question-External Comment Relevance (train 26690, test 5000)
- Our data is from SemEval 2016 Challenge. Train and test are no overlap.

Challenges

- Data sparsity: only **2,000** for Task B.
- Informal Language: emoticon, abbreviation, usage of punctuation, typo, grammatical mistakes
- Highly diverse content of comments: “and u work for me shhh!!!”
- Q/C length variation
- Imbalanced labels for Task C (more than **90%** is negative)

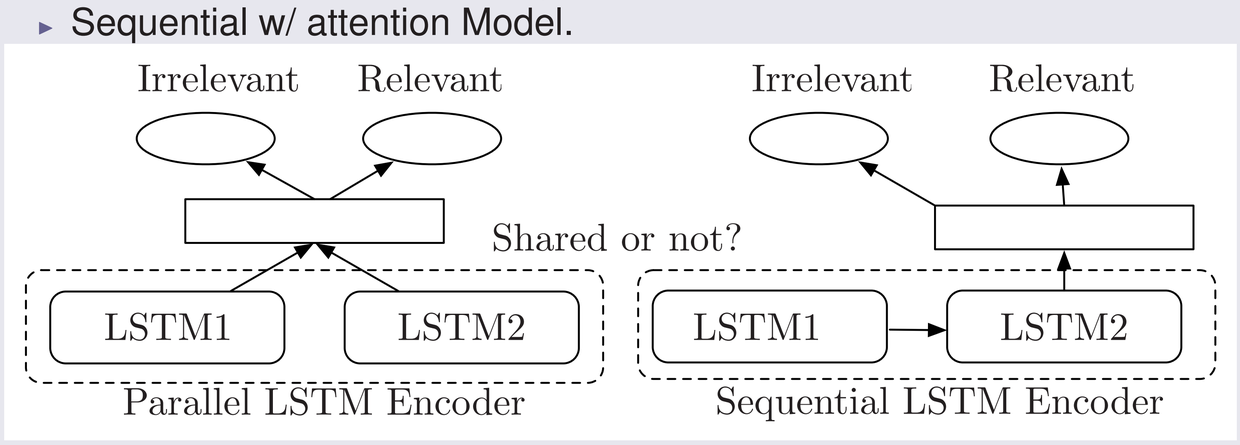
Previous Work



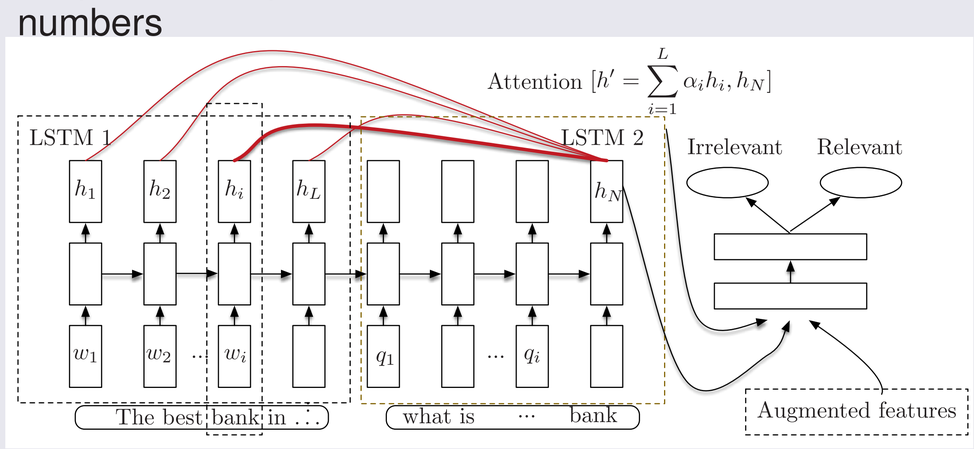


Our Models

- Paralleled LSTM Model, Sequential LSTM Model
- Sequential w/ attention Model.



- Dropout, L2Reg, AdaGrad, AdaDelta, SGD, Hidden node numbers



Experiments

- Preliminary Results

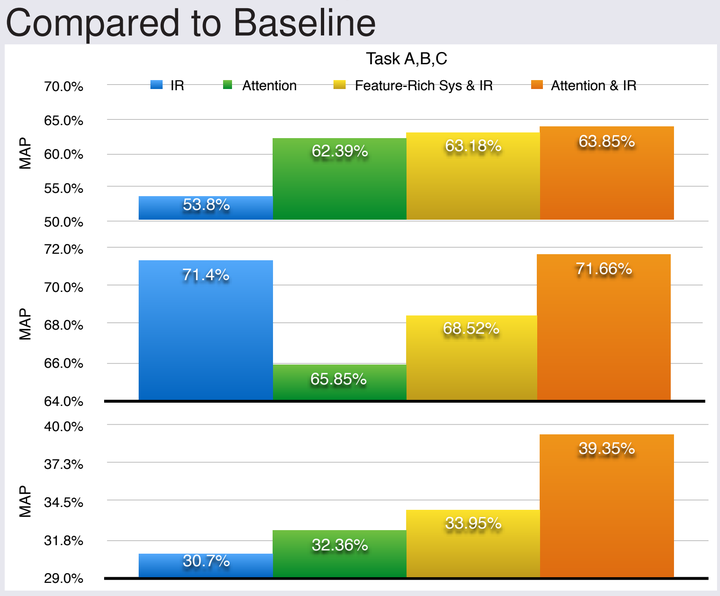
Task A	MAP	F1
Random	0.4860	0.5004
Parallel LSTM	0.6123	0.6091
Seq LSTM	0.6175	0.6063
w/ Attention	0.6239	0.6218

Task B	MAP	F1
Random	0.5595	0.4691
Parallel LSTM	0.5128	0.2452
Seq LSTM	0.5620	0.4299
w/ Attention	0.5723	0.4334

Task C	MAP	F1
Random	0.1383	0.1277
Parallel LSTM	0.2473	0.0057
Seq LSTM	0.2356	0.0115
w/ Attention	0.2837	0.1449

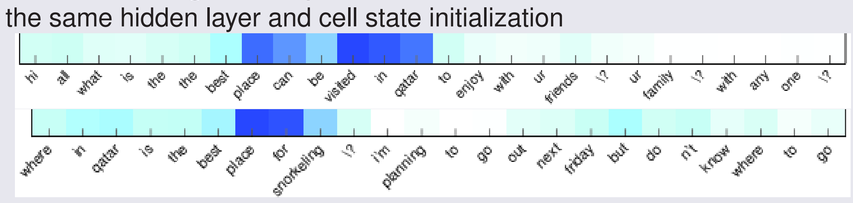
Very Low F1

- Task B: only 2,000 pairs for train
  - + pretraining (SNLI, taskA): **1%** MAP, **0.*X*%** F1, more stable
  - + adding additional pairs (e.g. if both rel1 and rel2 are related, then (rel1,rel2)=1): **0.5%** MAP, **15.2%** F1
- Task C: imbalanced labels
  - + pretraining (SNLI, taskA): **1%** MAP
  - + multi-task: **0.5%** MAP, **2%** F1
  - + adding taskA’s data: **0.5% 6%** F1
- Augmented feature can improve all the tasks’ MAP (**8**% on B, **4**% on C).
- Compared to Baseline

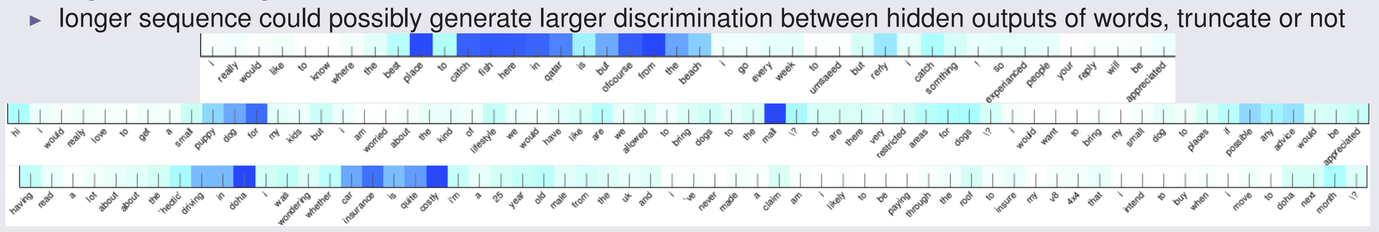


Qualitative Analysis of Attention Mechanism

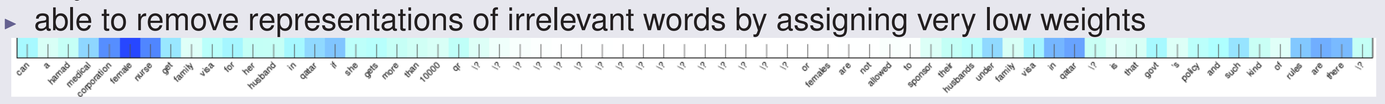
- Short Sentences
  - successfully assign larger weights to keywords
  - initial words have medium weights despite their irrelevance
    - might result from the same hidden layer and cell state initialization



- Long Sentences
  - amazingly pick keywords as well!
  - assign lower weights to initial words
    - longer sequence could possibly generate larger discrimination between hidden outputs of words, truncate or not



- Noisy Sentences
  - able to remove representations of irrelevant words by assigning very low weights



Conclusion

- Attention improves the performance
- Complementary with traditional feature-based system
- The more data, the better results
- Pretraining from external data yield more stable results
- Only limited data: just Google it

Future Work

- Incorporate more meta data-feature (i.e. user, topic, date)
- Preprocess data with morphological normalization and OOV mapping
- Ensemble different systems for better performance