Deliverable #3

C.J Hsu

Ryan Bielby

Outline

- Improvements for whole system
- Improvements for frontend System
- Improvements for backend System
- Evaluation
- Discussion
- Future work

Improvements for Whole System

- Implement a question classifier based on the data of UIUC
- Abandon the two-stage QC procedure, pop up some refined labels to first stage (21 class labels in the first stage)
  - e.g., “OTHER”, “ENTY:animal”,” ABBR”,

  ”NUM:date” , “ENTY:sport”, etc.
- Testing accuracy is 0.946 on TREC 10 data
- Incorporating statistical QC system doesn’t imply that we abandon the rule-based QC system

Improvements for Frontend System

- Implement sentence and word tokenizer ourselves instead of using NLTK
- Incorporate NER on snippets to identify “Person”, “Organization” and “Location”
- New pipeline of frontend system: “Voting”, “Filtering”, “Combining”, “Scoring”, “NER_Filtering” and “Reranking”

Improvements for Frontend System (con’t)

- NER_filter only works for HUM, LOC; we incorporate WordNet to verify the results on question types like “ENTY:religion”, “ENTY:sport”, “ENTY:animal”, etc.
- This mechanism explains why we try to refine the Question Type class labels
- Predicting the type of question is easy, how to verify the corresponding answer is hard. Try to focus on the question types that you could deal with.

Improvements for Backend System

- Question Series Handling: Replace the pronoun of every question by its question context
- Query Expansion: Perform NER on question and do not separate the ENTITY into unigrams when issuing the query
  - result is not good, still under investigation

Evaluations on whole system

	D2	D3
strict	0.1031	0.1325
lenient	0.2383	0.3153

Discussions

- A lot of n-grams from snippets are unnecessary, why does Jimmy Lin not try to avoid generating them? A bunch of simple solutions could be directly applied to avoid them.
- Why do we only incorporate three ENTITY types of openNLP when performing NER on snippets?

Future Work

- Improve runtime by running multiple questions simultaneously
- Look into ways of incorporating equal parts web-redundancy and information from documents in the corpus