**Question Answering System Project Report**

Subject: COMP 90042 Web Search and Text Analysis

Date: 24th May 2017

Team Name in Kaggle Competition: Mainframe

Team Members: Kuan Qian (Jack) [686464]

Zequn Ma [696586]

Yueni Chang (Nia) [884622]

**Introduction**

In this project, we have made our attempts on building a question answering (QA) system. Starting with an implementation of the basic QA system described in specification, we have made error analysis and introduced some enhancements to the system. The basic and enhanced system are then evaluated using several different metrics.

The QA system takes 3 datasets in JSON format, for training, development and testing respectively. Each dataset is a JSON array, with each item being a "trial", that is, a set of documents (sentences) and questions. For training and development sets, answer to each question is also provided, including its content and label of the sentence containing the answer. The system takes questions along with a set of documents as input, and provides the answers as output.

**Basic QA System**

For the first part of the project, a basic QA system has been made. This does not only help us understanding the problem, but also provides a reasonable baseline evaluation for further development.

The basic system, as described in the specification, is formed by three main components. Firstly, the sentence retrieval component attempts to find the most relevant sentence to a certain question. The retrieved sentences are then used as the input for entity extraction component, which aims to extract named entities in the sentences. The entities are finally ranked as potential answers to the question, with the top answer returned as the final output. Implementation details for the basic QA system are not included for brevity of the report, the system is designed and implemented to be consistent with the specification.

**Error Analysis**

**1. Sentence Retrieval**

64% sentences retrieved in the basic QA system, while testing against the development set, are correct answer-containing sentences for corresponding questions. We are not sure if this function can be easily improved. On the other hand, 64% is quite acceptable considering the performance baseline. If entity recognition and answer ranking can correctly utilise a third of these sentences, it is possible to achieve an accuracy of 20% roughly.

**2. Entity Extraction**

While running against the development set, only 19.3% of answers are extracted as entities, there is significant space of improvement for entity extraction component. The basic QA system only uses Stanford NER as entity extraction tools.

**3. Question type**

By inspecting the results from running the QA system on the training set, many questions had been identified as wrong types with the simple rule based classification. Especially for questions starting with "what" or "which", it was difficult to classify the question as correct type. It was also an issue with number types, questions asking for answer types such as year, money, measurements are all classified as "NUMBER" types. If these types can be distinguished, it was expected to increase the accuracy of our QA system.

**4. Answer ranking**

In many occasions, the few answers ranked on the top are of same rank. The ranking rules in basic QA system doesn't handle answers of same rank very well. Basic ranking may prefer a wrong answer, this calls for a more specific ranking method.

> - NER has significant room of improvement (only 24% identified in the correct (and retrieved) sentence, 29% in any retrieved sentence), among the failed cases:

> - About 70% that can possibly be extracted

> - 10/35 partial match (measurements, symbols such as ‘,’ and ‘-’)

> - Common words (noun phrases) not extracted

> - Date not fully extracted

> - Identifying more entities could possibly bring much more false positives, a good ranking algorithm will be required

> - Over 90% cases where correct sentence is retrieved in top 20, but only 60% as top 1. Consider improving sentence retrieval, or interpolate sentence ranking with other properties after rule 2

> - 16% when both of above are correct (correct sentence retrieved as top 1, in which correct answer is extracted by NER), but drops to 12% after applying answer ranking. With current implementation:

> - Rule 1: No significant problem

> - Rule 2: Question types not always correctly identified (88% accuracy for sample of 50 in training set, manually assessed), or entity type not correctly identified.

> - Take sentence retrieval ranking: Some correct sentences not ranked as top 1, whose answers will be ranked lower to those in higher ranked sentences and passes Rule 1&2

> - Rule 3: Not always helpful, sometimes prefers wrong answer. (stats?) All sorts of entities with type of ‘OTHER’, may confuse the ranker.

**Enhancement**

**1. More types**

First obvious enhancement was to use a 7-class NER model.

**2. Type classification**

As more answer types are brought into the system, the basic rule based classification is no longer sufficient. Therefore, we

**3. POS tagging in answer extraction and tokenization method**

**4. Unsuccessful attempts**

Answer classification

Syntactic distance

Answer extraction by selected pattern

> Rank answers using a classifier (from textbook, features at p8) FAILED

> Use ML classifier + NER for question type classification (along with more types)

**Evaluation**

> Metrics: Exact match, partial match, MRR...

> Test for: Sentence retrieval, NER, answer ranking

> For rule 1/2/3, could be useful taking random as baseline

**Future Improvements**

**Predicates translation with dependency parsing**

**Answer ranking system**

> - Take advantage of semantic information

> - Apply RNN for a “translation” from answers to questions (ref. TREC 2016 CMU paper)

> - Other approaches such as learnt pattern matching (ref. Learning Surface Text Patterns for a Question Answering System)

**Conclusion**

A question answering system has been built for this project, including a basic implementation as well as several enhancements. Although some of the enhancement attempts, such as applying syntactic distance and performing classification on extracted entities directly, does not improve the results, other approaches do seem to contribute. Applying machine learning to classify question type results in..., and.... With the ... enhancements combined, the system achieved an accuracy ... for the development set, and ... as the evaluation result on Kaggle. We believe the system can be further improved by applying more advanced approaches such as ... and ....