Mysterious Meerkat, QA System, CS6340, Fall 2015

Debjyoti Paul, deb@cs.utah.edu
School of Computing, University of Utah

QA SYSTEM STRUCTURE

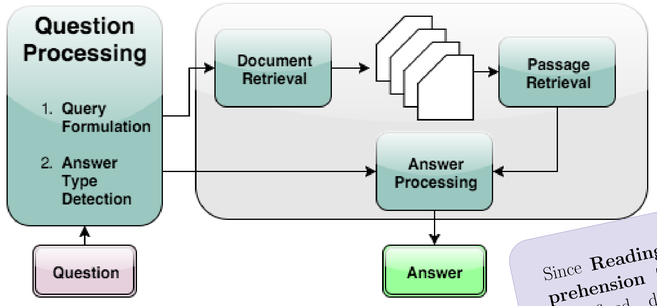

Fig. 1: Basic Question Answering System

Since **Reading Comprehension Test** has a defined domain of search space, our QA system does not require **Document Retrieval module**. Our focus is mainly on building an efficient **Answer Processing** module.

INTRODUCTION

Human beings have an inherent tendency to seek information. In the world of internet, useful information is free flowing. However, we are more interested in getting specific answer to queries rather than just gathering relevant information.

Question Answering (QA) system is a specialized form of Information Retrieval. QA lies at the intersection of **Natural Language Processing, Information Retrieval, Information Extraction, Machine Learning, Knowledge Representation, Logic and Inference, Sematic Search**. **QA Systems** are needed everywhere, be it medical science, learning systems for students and personal assistants.

Here, in this project, we are more concerned in answering quesions for Reading Comprehension tests where domain of search space is defined.

Note:
Preprocessing module is not shown in diagram. In preprocessing step, the passage is passed through an **Anaphora Resolution system**(i.e. **BART-Beautiful Anaphore resolution Toolkit**) to resolute pronominal coreferences with a hypothesis, that will facilitate Answering Module to find answer.

AFTERTHOUGHTS

Things went well

1. Question classifier to determine answer type as HUM:ind, HUM:gr, LOC:city, NUM:date, NUM:money etc. is the key to enhance performance.
2. Using regex to find number, date and currency such as 'dd[st|nd|rd|th] [Month]' in date, 1800-2100 for years and $100.0 or $ 0.50 for currency has helped.
3. The idea of measuring the length of the shortest path in the semantic ontology between two words has been beneficial. Also, extracting verb from question and finding matching verbs or its synset has improved system performance.

Scope for improvement

1. Coreference resolution has little effect on my QA system, though it might be required to revamp scoring function to observe its impact.

Did you know?
First **Neural Network** based Factoid QA on passage was published last year (2014), by M. Iyyer of UMD.

MODULES

Question Processing Module

1. Question Classifier
2. Answer Type Detector

Answer Processing Module

1. Named Entity Extractor
   - Human
   - Location
2. Sentence Similarity module
   - *Verb and theme Extractor*
   - *Synset and Hypernym matching module*
3. Answer type rule based module
   - *Human Individual / Group*
   - *Number Count / Money / Date / Period / Size / Weight etc.*
   - *Location country / city / other*
   - *Reasoning answer*

Answer Formulation Module

1. Number, Name, Location extractor

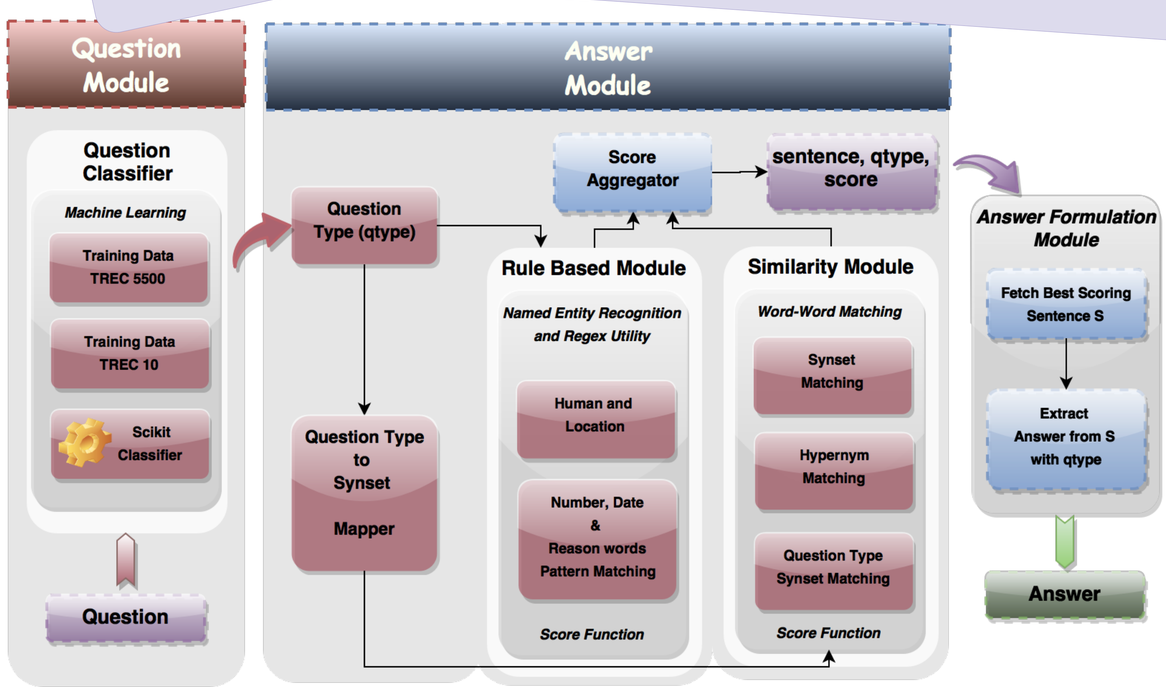

Fig. 2: Inside Answering Module

Did you know?
In 2000, Semantic and Rule based QA system gained a lot attention; researchers around the globe gathered and contributed to form a QA research roadmap. Our instructor **E. Riloff** has also made contribution in it.

CITATIONS

1. Question Classifiers is based on the Paper **Learning Question Classifiers: The Role of Semantic Information**, Xin Li, Dan Roth, *Natural Language Engineering*, 2004
2. Some concept of Sentence Similarity Module is borrowed from Paper **Sentence Similarity Based on Semantic Nets and Corpus Statistics** by Yuhua Li, David McLean et. al.
3. Coreference Module: **BART Beautiful Anaphora Resolution Tool** by Massimo Poesio et.al.
4. Worth mentioning our NLP friend **NLTK** and **Scipy**

Did you know?
SIRI was primarily a QA project started in **2003** funded by **DARPA** till 2008, though it became public knowledge only after being integrated with **Apple iOS4** in 2010.

SCORING $f(x)$

1. Rule Based scores $0 \leq f_1(x) \leq 1$
2. **0.25** for best matching, **0.15** for probable and **0.05** for 'may be' answers.
3. Scoring function of Similarity Module $0 \leq f_2(x) \leq 1$
4. Hypernyms matched words score is less than the synsets matched words $f_2(hypernym) < f_2(synset)$

$$\text{ScoreAggregator} = f_1(x) * (1 + f_2(x))$$