Project Presentation

Team Prashnottar*
Abishek Krishnan, u1261980
Monesha Murdeshwar, u1266063

* प्रश्नोत्तर - A hindi word which means “question and answering”

Contents

Overview

Our python3 based question & answering system is based on Quarc[1] paper

A Rule-based Question Answering System for Reading Comprehension Tests

Ellen Riloff and **Michael Thelen**
Department of Computer Science
University of Utah
Salt Lake City, Utah 84112
{riloff,thelenm}@cs.utah.edu

Libraries used: spaCy and NLTK

Methodology

- We developed a scoring system which ranks each sentence in the given story, corresponding to the question type.
- Steps

  1 - Baseline scoring for each sentence for a given question

  2 - Sentence scoring based on question type - (who, what, when, where, why,how)

  3 - Answer phrase extraction from sentence with highest score

Methodology (cont) - Baseline scoring

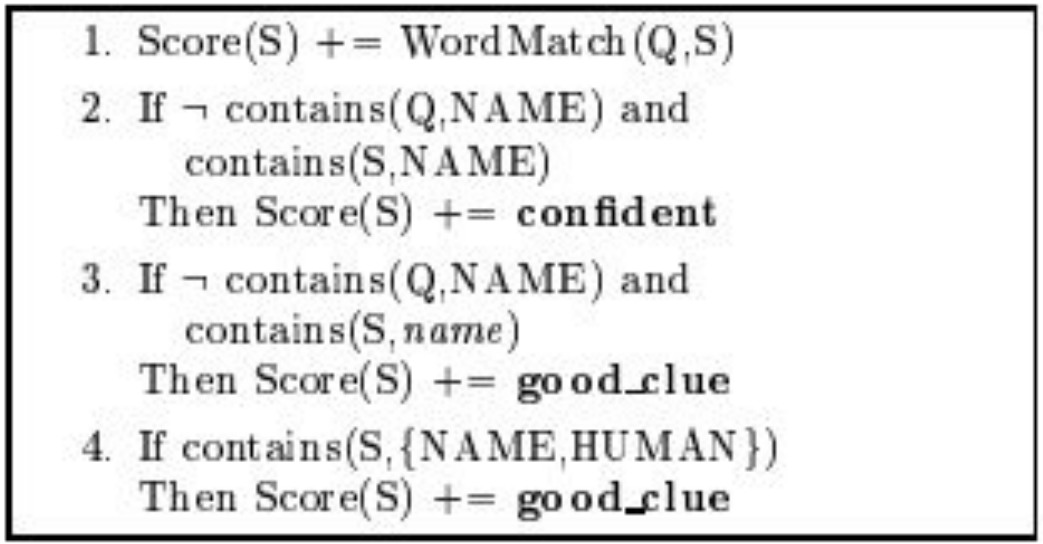
- Given a question, for every sentence
  - preprocess them to remove stopwords, case sensitivity

- Compute the cosine similarity between the resultant vectors, we scale it by a factor of 100, to match our scoring system

- POS tagged the resultant vectors, and add “5” points whenever a verb phrase matches

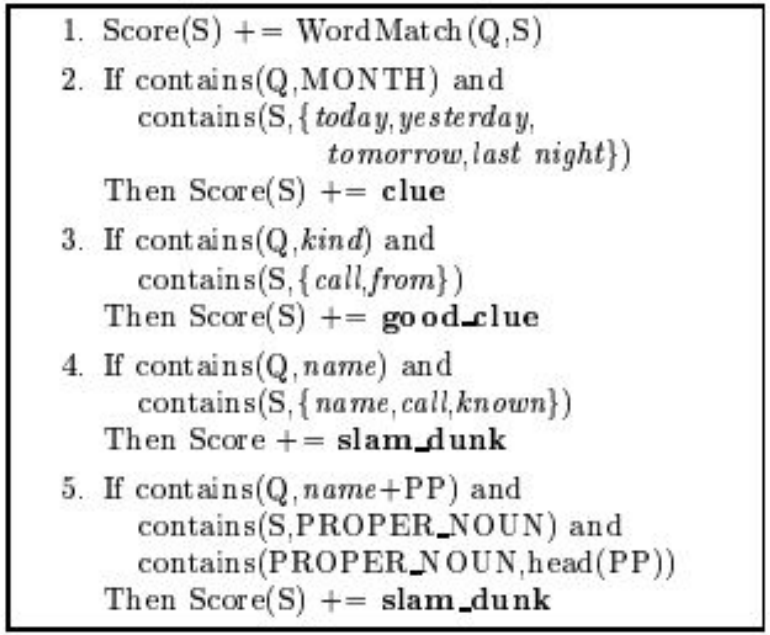
Methodology (cont) - Scoring question - “who”

- Who type - rules based on NER tagging and a “Names” list
  - Rules based on



Methodology (cont) - Scoring question - “what”

- What type - rules based on NER, POS tagging and a “months” list
  Rules based on



Methodology (cont) - Scoring question - “when”

- When type - rules based on NER tagging

1. If contains(S,TIME)
   Then Score(S) += **good_clue**
   Score(S) += WordMatch(Q,S)
2. If contains(Q,*the last*) and
   contains(S,{*first*,*last*,*since*,*ago*})
   Then Score(S) += **slam_dunk**
3. If contains(Q,{*start*,*begin*}) and
   contains(S,{*start*,*begin*,*since*,*year*})
   Then Score(S) += **slam_dunk**

Methodology (cont) - Scoring question - “where”

- Where type - rules based on NER, POS tagging

1. Score(S) += WordMatch(Q,S)
2. If contains(S,LocationPrep)
   Then Score(S) += **good_clue**
3. If contains(S, LOCATION)
   Then Score(S) += **confident**

Methodology (cont) - Scoring question - “why”

- Why type - rules based on previous and next sentence structure

1. If S ϵ BEST
   Then Score(S) += **clue**
2. If S immed. precedes member of BEST
   Then Score(S) += **clue**
3. If S immed. follows member of BEST
   Then Score(S) += **good_clue**
4. If contains(S, *want*)
   Then Score(S) += **good_clue**
5. If contains(S, {*so*, *because*})
   Then Score(S) += **good_clue**

Figure 6: WHY Rules

Methodology (cont) - Scoring question - “How”

- Subdivided into types

```
if question_words.__contains__("much"):
    if "PERCENT" or "MONEY" or "ORDINAL" in tagged_word_label_list:
        score += 10

if question_words.__contains__("many"):
    if "QUANTITY" or "ORDINAL" or "CARDINAL" in tagged_word_label_list:
        score += 10

if question_words.__contains__("old") or question_words.__contains__("often"):
    if "DATE" in tagged_word_label_list:
        score += 10
```

Methodology (cont) - Scoring question - “How”

```
if question_words.__contains__("tall") or question_words.__contains__("large") or question_words.__contains__(
        "high") or question_words.__contains__("deep"):
    if "QUANTITY" in tagged_word_label_list:
        score += 10

if question_words.__contains__("long"):
    if "QUANTITY" or "DATE" or "TIME" in tagged_word_label_list:
        score += 10

if question_words.__contains__("far"):
    if "QUANTITY" or "TIME" in tagged_word_label_list:
        score += 10
```

Methodology (cont) - Scoring question
- Answer phrase extraction - who, what & why

```
if current_question_type == "who":
    for e in ner_tagged_best_sentence.ents:
        if e.label_ == "PERSON" or e.label_ == "NORP" or e.label_ == "ORG":
            best_extracted_sentence += e.text + " "

elif current_question_type == "what":
    best_extracted_sentence = ""

elif current_question_type == "why":
    best_extracted_sentence = ""
```

Methodology (cont) - Scoring question
- Answer phrase extraction - where & how

```
elif current_question_type == "where":
    for e in ner_tagged_best_sentence.ents:
        if e.label_ == "PRODUCT" or e.label_ == "WORK_OF_ART" or e.label_ == "ORG" or e.label_ == "LOC" or e.label_ == "GPE"
            best_extracted_sentence += e.text + " "
```

How type - had answer extraction rules based on the sub-divided question types

Results

Precision - 0.3300 (103.95 / 315)

Recall - 0.5968 (188.00 / 315)

Fscore - 0.425

References

[1] - https://www.cs.utah.edu/~riloff/pdfs/quarc.pdf

Thank you
Questions?