CS 6340 Final Project
Question Answering System

Team: Wise Ferrets
Anirudh Narasimhamurthy(u0941400)
Vulpala Shiva Priya(u1011589)
KEY APPROACH

❖ A rule based question answering system primarily modeled from the rules used in Quarc[1] with few modifications.
❖ Separate set of rules for each question type: WHO, WHAT, WHEN, WHERE, WHY, HOW, WHICH.
❖ In addition there were also questions of type WHOM and WHOSE which made use of the same rules used for WHO type questions.
QUESTION AND SENTENCES PARSING

- Given story was parsed and sentences were obtained primarily based on the occurrence of full stop.
- The sentences were then run through stop word removal process. Used Reuters-RCV1 stop list of 25 words.

![Figure 2.5: A stop list of 25 semantically non-selective words which are common in Reuters-RCV1.](image)

- Maintained two lists: Complete_sentence_list and Stop_words_free_Sentence_list and used it accordingly during rule processing.
- Question was also parsed from the questions file removing the question ID and storing just the required questions of the story in a list.
Stanford NER and POS Tagging

- Every sentence in the story was also initially passed through Stanford’s NER to get named entities.
- We used the 7-class model: Location, Person, Organization, Money, Percent, Date, Time.
- Global lists for all the 7-classes were maintained for each story.
- Also used the POSTagging module for picking out NP or verbs from candidate sentences.
WordMatch Function

- The most important or key function used for finding the candidate answer sentence(s).
- Verb matches weighted more heavily than non verb matches.
- Used NLTK’s `word_tokenize()` to get tokens from question and sentence.
- First step is to look for an exact word match. If found a score of 3 points was given for every matching word in question and sentence.
- If there was no exact word match, then for the set of unique tokens from question and sentence, we used `WordNetLemmatizer()` for finding the verb match.
- Verb match was given a score of 6 points.
Different rules were applied to each question type and final score for each sentence was the sum of scores for each individual rules.

Scoring was done in an identical fashion as provided in the paper, with a clue getting 3 points, good clue getting 4 points, confident clue getting 6 points and slam dunk getting 20 points.

We also had custom scoring rule of 10 points in scenarios when, we were more than confident and less than being a slam dunk answer.
Answer Selection (Generic Rule)

- After scoring all the sentences in the story, find the sentence with the highest score as the answer sentence.
- Based on the question type, different rules/checks are applied to get different named entities. If they are present, then they are displayed as the answer.
- If no such entities exist or are rather not identified by Stanford NER and the other custom rules also did not help in narrowing down, then the answer is the complete sentence minus the words which appear in the question.
- Slightly helps with the precision for few sentences where displaying the entire sentence would lead to a very low precision.
WHO Rules

- Used the exact same set of rules given in the paper for processing WHO questions.
- NER’s PERSON entity output + custom-made profession list was used to check for the existence of NAME in the question and sentence.
- Additional Rule: Because WHO questions generally have a NAME as the answer, if question contains a profession and the sentence also has a profession name in it, then we reward those sentences as a confident clue.

Figure 2: WHO Rules

1. \( \text{Score}(S) += \text{WordMatch}(Q,S) \)
2. If \( \sim \text{contains}(Q,\text{NAME}) \) and \( \text{contains}(S,\text{NAME}) \) Then \( \text{Score}(S) += \text{confident} \)
3. If \( \sim \text{contains}(Q,\text{NAME}) \) and \( \text{contains}(S,\text{name}) \) Then \( \text{Score}(S) += \text{good.clue} \)
4. If \( \text{contains}(S,\{\text{NAME},\text{HUMAN}\}) \) Then \( \text{Score}(S) += \text{good.clue} \)
ANSWER Selection

- Based on the score, the sentence(s) with maximum score was added to the candidate list.

- If candidate list had only one sentence, and question did not have a NAME, then the PERSON name or PROFESSION names which appear in the sentence were selected for answer.

- Else based on POS Noun Tagging (‘NN,NNS,NNP,NNPS’) those words were used in the answer sentence.

- Also sentences of the type ‘Who is X?’ and ‘Who defeated X?’ were also appropriately handled. The former generally has profession name as the answer and the latter has a PERSON name which is not appearing in the question as the Answer.

- If multiple sentences were there, then the sentence which appeared first was chosen as the answer sentence and answer was appropriately displayed.
WHAT Rules

- Used the rules given in the Paper but did not implement the 5th rule which checks for Proper Noun and Head noun.

- Instead we found that having a verb match rule which checks if any of the verbs in the question and sentence exactly match, they are much more stronger indication of an answer.

- Also helps to overcome questions where word match gives a high score for another sentence which might not be the correct answer.

- Additional Rules to handle “What is X?” type definition questions. Questions of those type had the answer enclosed within {} or ()

```
1. Score(S) += WordMatch(Q,S)
2. If contains(Q,MONTH) and
   contains(S,{today,yesterday,
                tomorrow,last night})
   Then Score(S) += clue
3. If contains(Q,kind) and
   contains(S,{call,from})
   Then Score(S) += good_clue
4. If contains(Q,name) and
   contains(S,{name,call,known})
   Then Score += slam_dunk
```
Answers to WHAT type questions were generally lengthy or in some cases the whole sentence.

So not a lot of filtering was done apart from removing words which appear in the question from the answer sentence.

(Localized rule:) Few of the what questions seemed to have a per cent value as the answer and so if the answer sentence contained “per cent”, then the value before it was displayed along with the per cent as the answer.
WHEN Rules

- Used the exact same rules provided in the paper.
- For TIME entity, we used the output from both the DATE and TIME class of Stanford NER and also had a custom list of time related words.
- Also maintained a list of prepositions which generally occur with sentences describing time for filtering the answers from the candidate sentence.
- We had also implemented the DATELINE rules provided in the paper, but it did not seem to be a candidate for any of the sentences provided in TestSet1.
Answer Selection

- Once the answer sentence was identified, the month and time values if any present in the sentence were taken out and stored in list.

- If both of them were empty, then the entire sentence minus the words in the question was printed as the answer.

- Else if time_list was not empty, then words appearing immediately before and after them are also part of the answer. Ex: 10 hours ago, 5 days ago, few months, ten years.

- Time prepositions list which was used included: ['over', 'period', 'within', 'inside', 'under', 'ago', 'through', 'past']

- Time related numbers and isdigit() function was also used to identify or extract potential answer words.
WHERE Rules

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

Figure 5: WHERE Rules

- In addition to the 3 rules, we had also added one more rule to reward sentences which contains the word “from” with 6 points, if the question also contained the word “from”.

- These proved slightly helpful in answering “Where is X from?” questions, which were quite a few in the small corpus.

- LOCATION was obtained from the NER’s LOCATION entity and we used a list of 30 location prepositions from [http://www.englisch-hilfen.de/en/grammar/prepositions_place.htm][2] plus a few other related prepositions to the corpus.
Answer Selection

- If the candidate answer contains a LOCATION entity and that entity is not present in the question, then we simply print that word alone.
- Else we filter out the locations appearing in the question and then display the other LOCATION values.
- If neither of them work, we display the entire sentence minus the words that appear in the question.
- We tried doing a POS Tagging and then displaying the ‘NNP or NNPS’ words, but it did not seem to do too well for many cases. The approach of printing the entire sentence provides a good recall but poor precision but this overhead seemed to give better scores.
WHY Rules

- Used the same exact rules as provided in the paper.
- Added one more rule of Verb match between the question and sentence which seemed to help out in few situations, when the sentence with the best score could be the answer instead of the one after it.
- The generic rules seemed to be doing a good job with this corpus which we had.
The answer to Why questions are generally lengthy when compared to the other question types.

If the answer sentence contained any of the following words: ‘so’, ‘because’, ‘to’, then the final answer was the sentence starting from this particular word till the end of sentence.

This seemed to be doing a very good job as even questions with moderate difficulty were answered correctly with full precision and accuracy sometimes.

Ex: because jumping, known as "taking big air," is such a big part of the sport.

This was one of the answers to the questions in the last story in testset2 and this question received a precision and recall of 1.

When multiple candidate sentences were present, the sentence which appeared at a later point in time was selected as the answer sentence.
HOW Rules

❖ Based on the rules for the other question types, similar analogy was applied to get a set of rules for HOW questions.

❖ The WordMatch function was one of the key scoring rules. The other three rules are as follows:

❖ If question contains “many” and sentence contains an expression of number, then it is confident score.

❖ If the question contains “much” and the sentence contains an expression of measurement or number, then it is a confident score.

❖ If the expression contains “long” or “often” and the sentence contains an expression of time, then it is a confident score.
Answer Selection

- Lists containing words for much, many and often were maintained.
- Regular expression for a number match was also used for picking out the numbers for the answer to how “many” type questions.
- If all the lists returned empty, then the entire sentence minus words in question was displayed as the answer.
- When multiple candidate sentences are there, the sentence which came first was selected as the answer.
The QA system seemed to perform well for “What”, “How”, “When” and “Why” type questions particularly well.

The problem of finding the correct sentence was reasonably solved well but picking out the answer or improving the precision was the biggest challenge. For “what” type questions, it was able to find the correct sentence like more than 7/10 times but displaying the entire sentence makes the precision go for a toss.

“Who” and “Where” questions did well for few cases but the corpus had answers which were not generalizing to a PERSON or LOCATION at all times and the Stanford NER also had few issues where it couldn’t correctly identify them as PERSON or LOCATION.