11791 Homework 3 Writeup

Yueran Yuan

Note

One difference in my file structure is that my baseline aae and my as-aae are a little different. As a result I've named the baseline **hw3-yuerany-aae.xml** and the as-aae **hw3-yuerany-aae-as.xml**

Overview

I deployed my aae locally and accessed the scnlp annotator remotely in my UIMA AS project. To augment my answer scores with NamedEntity, I computed the percentage of named entities from the question that were found in an answer. This and the Ngram scores from HW2 were aggregated in a weighted sum and used as the final answer score. I also replaced the local Stanford Tokenizer with the remote ClearTK tokens to try to improve performance.

Both speed and precision decreased in the new pipeline.

AS Pipeline

The following describes the new pipeline created in accordance to the requirements of homework 3. Sections novel/modified from hw2 are in *italics*

1. *Collection Reader using FileSystemCollectionReader provided by uima*
2. *scnlp annotator hosted by the course staff to annotate with ClearTK annotations*
    1. *client interfaces with remote service*
    2. *remote service returns annotations*
3. Modified hw2 annotator (includes NamedEntity) deployed on local machine as an AS service
    1. TestElementAnnotator converts document into question and answer annotations
    2. *TokenizeAnnotator converts the ClearTK tokens computed remotely into tokens in our annotator format. This replaces the need for using Stanford NLP locally.*
    3. NgramAnnotator uses the tokens to compute Ngrams over the next
    4. NgramScorer uses a specified (parameter) Ngram evaluation algorithm to score a sentence using Ngrams
    5. *NamedEntityScorer uses the NamedEntity information collected from scnlp remotely to compute a separate score. More details below*
    6. *AggregateScorer aggregates the scores from the above 2 scorers into a single answerScore annotation for each answer*
4. *CasConsumer takes the place of the Evaluator and prints the ranked answerScores and* precision@N *to the console*

Token Annotator

The Stanford Tokenizer is removed and clearTK tokens computed remotely are used instead. The hope is that this will increase speed because the remote machine is expected to be faster than my local machine.

NamedEntity Scorer

The scorer uses named entities mention annotations from the scnlp remote service to compute a list of named entities for each sentence. Then a scorer checks whether each named entity found in the question could also be found in the answer. The score for a given answer is the percentage of named entities from the question that occurs at least once in the given answer.

From examination of the annotations, I can see that the annotations are not very precise. Therefore, I've given this scorer a relatively low confidence (0.3). In future projects that focus on evaluation, I might use regression to computationally arrive at an appropriate confidence value for this score.

Aggregate Scorer

After the NgramScorer and NamedEntity scorers have been run, there multiple answerScore annotations per answer. These scores are read by the Aggregate Scorer and aggregated for each answer. The scores are summed and weighted by their confidence. To make sure the resulting score is less than 1.0, the summed scores are divided by the total count of scores.

The answerScore annotations which are not generated by this phase are then removed so as not to polute the Cas when the the annotations are evaluated by the TA.

Results Compared Against HW2

HW2 output:

```
Question: Booth shot Lincoln?
+ 0.28 Booth shot Lincoln.
- 0.21 Lincoln shot Booth.
+ 0.17 Booth assassinated Lincoln.
- 0.14 Booth was shot by Lincoln.
+ 0.10 Lincoln was shot by Booth.
- 0.10 Lincoln assassinated Booth.
- 0.08 Booth was assassinated by Lincoln.
+ 0.05 Lincoln was assassinated by Booth.
Precision at 4: 0.50
Question: John loves Mary?
+ 0.28 John loves Mary.
+ 0.11 John loves Mary with all his heart.
- 0.08 John doesn't love Mary.
- 0.05 Mary doesn't love John.
+ 0.03 Mary is dearly loved by John.
Precision at 3: 0.67
Average Precision: 0.58
```

H2 Time: ~0.8sec

HW3 output:

```
Question: Booth shot Lincoln?
+ 0.29 Booth shot Lincoln.
- 0.26 Lincoln shot Booth.
+ 0.23 Booth assassinated Lincoln.
- 0.22 Booth was shot by Lincoln.
- 0.19 Booth was assassinated by Lincoln.
```

```
+ 0.05 Lincoln was shot by Booth.
- 0.05 Lincoln assassinated Booth.
+ 0.02 Lincoln was assassinated by Booth.
Precision at 4: 0.50
Question: John loves Mary?
+ 0.21 John loves Mary.
- 0.12 John doesn't love Mary.
- 0.10 Mary doesn't love John.
+ 0.06 John loves Mary with all his heart.
+ 0.02 Mary is dearly loved by John.
Precision at 3: 0.33
Average Precision: 0.42
```

HW3 time: ~2.0sec

Discussion

As we see from the above, the HW2 pipeline performs better and is faster than the HW3 pipeline. The speed differences are likely a result of the latency created by the network communications in UIMA AS and also by the inherent overhead of doing those network communications. Critically, we note that it did not help to do remote tokenization instead of local tokenization. I ran the HW3 pipeline with local tokenization and did not notice any time differences.

The performance hit can be attributed to the new NamedEntity scores. This is two-fold. First, as we noted earlier, the NamedEntity annotations are fairly inaccurate (sometimes tagging an entire sentence as one named entity) and do not provide much additional information. Second, from this dataset, we see that all named entities are show with the same surface form and every named entity occurring in the question occurs in every answer choice. As a result, even if the NamedEntity tags were perfect, we should not expect a performance gain in this dataset.

That said, it's conceivable that in other data sets NamedEntity annotations could increase the score. For instance, Lincoln and Booth both have many names e.g. Abraham and it would be important to be able to recognize when a sentence is referring to the same person by another name.