Question Answering Using Aspect Model

Abstract: This approach initially aimed at doing better QA by using the Aspect Model for the words that do not occur in the wordneti.e. the non-wordnet words. The other words i.e. the wordnet words were supposed to be handled using wordnet. But as a first stab, we planned to try out using only Aspect model. The Aspect-based learning gives somewhat satisfactory parameter values. But it falters while inducing similarity between questions and passages. Therefore, we concluded by using term vector based approach coupled with aspect parameters for question answering.

1 Proposed approach for Question Answering

Our proposal is to do QA using aspect model in an unsupervised manner. To this end, we thought it appropriate to use *question-types* as the aspects. The exact question-types were fixed based on the following statistics from TREC-9.

1.1 Usage Statistics of different cue-phrases in questions

The overall question answering system that we propose is depicted in figure 4.

```
1: z = {how, when, what, who, how_many, which, where}
```

Figure 1: 7 clusters for different question types

Cluster Names	Count
when did	20
how much	6
which	13
why	2
who	115
what	363
where	71
how many	27
how	54
when	48
in which	3
who is	27
who was	30
where does	3
how large	1
what does	15

A is a candidate answer passage, Q is a query.

To find $P(Q|A) = \sum zP(z)P(Q|z)P(A|z)$

Training passages (psg): All passages from corpus(A) and all correct question-answer pairs (Q-A)

EACH Q-A pair is worth $\Lambda > 1$ **times an ordinary passage**

Initialisation: $P(z) = \frac{1}{31}$, $P(psg|z) = \frac{1}{Npsg}$, $P(w|z) = \frac{1}{(Nw)(1+31)}\ if\ w \neq z\ else\ \frac{1}{1+31}$

while Paramaters don't converge **do**

  E Step: $P(z|psg, w) = \frac{P(z)P(psg|z)P(w|z)}{\sum x^{'} P(z^{'})P(psg|z^{'})P(w|z^{'})}$

  M Step1: $P(w|z) = \dfrac{\sum psgn(psg, w)P(z|psg, w)}{\sum w, psg^{'} n(psg^{'}, w)P(z|psg^{'}, w)}$

  M Step2: $P(psg|z) = \dfrac{\sum wn(psg, w)P(z|psg, w)}{\sum w, psg^{'} n(psg^{'}, w)P(z|psg^{'}, w)}$

  M Step3: $P(z) = \dfrac{\sum psg, wn(psg, w)P(z|psg, w)}{\sum psg, wn(psg, w)}$

end while

Figure 2: Training the Aspect Model

Folding in Q:

while Paramaters don't converge **do**

  E Step: $P(z|Q,w) = \frac{P(z)P(Q|z)P(w|z)}{\sum x^{'} P(z^{'})P(Q|z^{'})P(w|z^{'})}$

  M Step2: $P(Q|z) = \frac{\sum wn(Q,w)P(z|Q,w)}{\sum w,Q^{'} n(Q^{'},w)P(z|Q^{'},w)}$

end while

Find

for all A's **do**

  $P(Q|A) = \sum zP(z)P(Q|z)P(A|z)$

end for

Pick 'A' with highest $P(Q|A)$

Figure 3: Folding-in followed by finding the correct answer (Initial proposed algorithm)

To find $P(Q|A) = P(WQ1,...W_{Q_m}|W_{A_1}...W_{A_n})$

$P(WQ1,...W_{Q_m}|W_{A_1}...W_{A_n}) = \frac{P(W_{Q_1},...W_{Q_m},W_{A_1},...W_{A_n})}{P(W_{A_1}...W_{A_n})}$

$\frac{P(WQ1,...W_{Q_m},W_{A_1},...W_{A_n})}{=} \sum_{z} P(z)P(W_{A_1},...W_{A_m}|z)P(W_{Q_1},...W_{Q_m}|z)$

Figure 4: An alternate unexplored approach

1. Preprocessing :
    - identify POS tagging (already being done)
    - Do stemming and morphing
    - Collocation detection and addition
2. Initialization
3. Construct the dyadic data
4. Loop through
5. Online :
    - Folding-in
    - Inferencing

2 Results

Question :
What is Francis Scott Key best known for?
Passage :
Tag anthem to them like a poem, and translate for them the following words and phrases : ; The star-spangled banner : Spangled means adorned with shining or glitte ring objects ; sequins , for example , are spangles. ; What so proudly we hailed : Hailed means cheered for or applauded.
cluster with highest probability:
what(0.00011066)

Question:
Who is the fastest swimmer in the world? Passage :
ag meters from the finish to win in 22.10 seconds. Jager was second in 22.31 , well off his world record of 21.81. Rudolph was fourth in 22.61 . Biondi , who has complained because swimmers are not being paid for competing here while athletes in some other sports.
cluster with highest probability:
who

Question :
Who is Coronado ? Passage :
Tag Jim Shore , who is handling the case. ; Both Cano , who is reputed to be a member of the prison gang Nuestra Familia , and Coronado – an admitted leader of the Fresno-based prison gang Cluster with highest probability:
who

2.1 Observation:

The passage containing a question phrase gets pulled to the cluster corresponding to that question phrase. This is many times undesirable, since *the occurence of a cue-phrase in a passage and in the question are not in the 'same sense'*. In fact the MRR obtained (0.178) is worse than asymmetric tf-idf.

2.2 Conclusion from the intial experiment

- Gives an intuition about why Hoffman **??** in his paper on PLSI, resorts to ad-hoc vector space cosine-similarity-like measures despite using Aspect to learn the parameters. In other words, a pure aspect model does not allow us to account for the proximity between question and answer based on exact match between tokens in the question and the answer.

- It seems futile to consider the occurence of 'cue-phrases' in the passages. This is because, once a 'cure-phrase' occurs in a passage, the passage becomes completely biased towards that cluster. And more often than not, the cue-phrase is not used in the passage in the sense in which it is supposed to provide an answer to the corresponding question type. In other words, it seems important that we start distinguishing between questions and answers.
- This raises an issue; to avoid hand tweaking of the form mentioned above, does it make more sense to actually come up with clusters whose representatives are unambiguous with respect to the task at hand ? For instance, in QA, it is not that every reference to *what* in a passage is associated with
- The clusters are supposed to do magic for us by completely representing the space of 'relevant terms', by filtering out noise. But it is found that the MRR decreases from 0.30 (as given by asymmetric TFIDF) to 0.178
- The values of $p(t|c)$'s have started making some sense.

IR baseline: IR technology is widely accessible, and forms our baseline. We construct 250-byte windows of text as passages and compute the similarity between these passages and the query. Because we would not like to penalize passages for having terms not in the question (provided they have at least some query terms), we use an asymmetric TFIDF similarity. Under this measure, the score of a passage is the sum of the IDFs of the question terms contained in the passage. If D is the document collection and D_t is the set of documents containing t, then one common form of IDF weighting (used by SMART again) is

$$\text{IDF}(t) \quad = \quad \log \frac{1+|D|}{|D_t|}. \tag{1}$$

The IR baseline MRR is about 0.3.

Our Modified Approach: Despite the failure of aspect-based QA, we do not use pure asymmetric TF-IDF, since the $p(t|c)$ parameters seem to make some sense. First of all, we classify the question into one of the clusters(question-types). Therefore, our c gets fixed when the question is seen. According to our measure, the score of a passage is the sum of the IDFs of the passage terms contained in the question and the sum of the $p(t|c)$ values of the passage terms not present in the question.

$$\text{Score}(passage) \quad = \quad \sum_{t \in passage \cap question} \log \frac{1+|D|}{|D_t|} + \sum_{t \in passage question} p(t|c) \tag{2}$$

The MRR we got with this was 0.258. We hope to improve this with some minimal effort.

3 Appendix

```
addLabeledDocToCurrM : Add question-answer pairs for the first time
normalizeToProbabilities("current")
if Semisupervised then
    for each question do
        getScoresForUnknownDoc : fold-in iteration for new question
        addUnlabeledDOcToNextM : Now all test documents are known to EM
    end for
end if
addLabeledDocToNextM
normalizeToProbabilities("reset")
writeNextMparamsToCurrent
Main ASPECT EM iterations
for each passage do
    if (AnswerToAQuestion(passage)) or (passageFromCorpus(passage)) then
        addToNextM : use the question to which it was answer along with the answer
    end if
end for
normalizeToProbabilities("next")
writeNextMparamsTOCurrent
if not(semisupervised then
    for each test doc do
        getScoresForUnknownDoc
        evaluateUnknownDoc : print evaluation
    end for
end if
```

Figure 5: Aspect Model : Training and QA

4 Design of the Aspect Module