QA System with Word Cluster

Introduction

Word clusters and word vectors are commonly used in the NLP to avoid sparsity problems in statistics. In this project, we compared different methods of word clusters and vectors such as pre-trained Brown Clusters, Twitter word clusters, word2vec and Glove word vectors from external resources, generate overall understanding and write detailed descriptions. When we looked through all the previous projects, we found it might be meaningful for us to implement word clusters for QA systems in the project 3 to improve the performance of scoring the answering sentences. We mainly relied on counting the overlap parts related to the question and scoring each sentences in the passage retrieval. Word clusters can help us get more candidates.

Basic Observation and Analysis

1. Pre-trained Brown Clusters

This clustering produces a clustering as well as a hierarchical tree. This is a hard clustering, meaning that no element is found in multiple clusters. The generative goal is to maximize

$\frac{1}{n} log \prod_{i=1}^{n} P(C(w_i)|C(w_{i-1}))P(w_i|C(w_i)) = \frac{1}{n} log \prod_{i=1}^{n} (transition\ probability)(emission\ prob)$

$= \sum_{c,c'} P(c,c') log \frac{P(c,c')}{P(c)P(c')} + \sum_{w} P(w) log P(w)$ where the first term is the mutual information

However, these are insensitive to sentence structure, and finds a local optima due to greedy merging.

For example, some clusters end up with words that aren't able to be otherwise clustered, as it isn't always clear how to merge clusters.

The tree structure is established as a binary encoding where the length of the identical left bits are a measure of the word similarity. Some examples of this are:

01110001111010 hands

01110001111011 clock

This correctly identifies that hands are similar to clock. However, it doesn't always work well, for example:

00000000101111101101101010 ridiculous

00000000101111101101101011 Scotland

Implies that Scotland and ridiculous are similar words. (http://brownclusterings.tumblr.com)

Low cluster counts and small input corpora both lead to poor quality Brown clustering

One example (from rcv1.1M-c5120-p1.paths)

000110	in	20155
000111000	about	1503
0001110010	around	467
00011100110	above	170
00011100111	below	161
0001110100	into	1298
00011101010	under	697

This is a cluster of relative location words. This could be useful for finding adjacency information, but at the cost of throwing out the specific relation, for example the sentence “The rug is below the couch, and the cat is on the couch”. If we ask what is above the couch, it is ambiguous if it is the cat or the couch, but since we can answer multiple words, we can just respond with “cat rug” which doesn’t answer the question, but gets the response in the list.

2. Twitter Word Clusters

Twitter word clusters is a hierarchical word clusters along with annotated corpora and web-based annotation tools. It is produced by an unsupervised HMM: Percy Liang's Brown clustering implementation on Lui and Baldwin's langid.py-identified English tweets. It contains more than 56 million English tweets (837 million tokens), 1000 hierarchical clusters over 217 thousand words and form separated words into binary tree structure which also includes wrong spelling identification.
For example:

010001110 (24)	think assume reckon thnk fink can't! thinkk haven't! thik thinkkk thinnkrekon tihnk thinl fnk thynk htink thinkkkk thinc thnik thibk thimk thiknthonk

3. Word2Vec

Word2Vec is an efficient word vector training tool which enables words to change into real vectors. We can easily use the result word embedding to get text similarity based on vector calculation. There are two core training models in word2vec which are Continuous Bag-of-Words Model(CBOW) and skip-gram. CBOW uses the context of a certain word to predict the word while skip-gram use the context to predict the word. Besides, the Hierarchical Softmax algorithm and the negative sampling contribute to the code efficiency.
In word2vec, the cosine distance reflects the similarity between two words. This value lies in the interval from -1 to 1, which means the similarity is 1 when two words are the same. The difference of two words implies the relation. If Paris - France = X - Germany(X refers to an unknown word), the word2vec will give you the answer of Berlin based on its cosine similarity calculation of vectors based on the large corpora.

4. Glove Word Vectors

The GloVe model is trained on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. It is essentially a log-bilinear model with a weighted least-squares objective.
Take the official example:

Prob and Ratio	k = solid	k = gas	k = water	K = fashion
P(k|ice)	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
P(k|steam)	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-4}	1.8×10^{-5}
P(k|ice)|P(k|steam)	8.9	8.5×10^{-2}	1.36	0.96

In the table, the first row means ice co-occurs more frequently with solid and the second row means steam co-occurs more frequently with gas. The third row checks the relations with two words given a word k condition. When k = solid, the value of 8.9 means the ice occurs more frequently with k = solid than steam. If we compared this value with others, we may find it is the maximum value and it may be a vertical vector pair and the word solid correlates well with ice. The minimum value of 8.5×10^{-2} may also means a vertical pair and word gas correlates well with steam. When the value is approximately 1, it means the word k correlates neither ice nor steam well.

Sparsity Problem

The sparsity problem in our QA system is that the words in question may occur very few times in the sentences. So when we calculate overlap words, it is very possible that the exact word in the question does not show up, even in the sentence that contain the right answer. For example, the question is “Who is the leader of of the group?” The right answer may be included in the sentence “The chairman is Joe.” But if we just count the overlap words, the score for this sentence can be very low for that the word “leader” does not show up in this sentence. But in fact, they mean same thing here. With the help of word cluster, we know that chairman and leader are in the same word cluster. So this sentence can have a relatively higher score.

Improvement over previous system

1. Noun phrase improvement

After analyzing the results of project3, we first found that the implement of calculating overlapped words between question noun phrases and sentence words could be improved.

For example, if the question is “Where is the paper clip?”, our last method will first extract noun phrase “paper clip” from the question. Then it will split every sentence into independent words and try to calculate the overlapped ones. But the problem is, it’s not possible to find overlapped words when the noun phrase has words more than one. So instead, we change the noun phrase as “paper” and “clip”. The result significantly outperforms last one.

2. Word cluster improvement

In project 3, our system will count overlapped words between question and sentence to select candidates. We use word cluster to get more candidates and tried two design choices here.

Word cluster: *Brown word cluster dep-brown-data-master/en/standard/paths_1000_min50*

Code implement: In BrownCluster.java file, given a word, we can return an arraylist of all the words in its cluster. By changing the parameters, we can choose the range of the cluster. For example, for word 001001011, we are able to get all the words with 001001011 or its left and right children 0010010110 and 0010010111 or grandchildren. By using upsearch function, we can even get cluster like all the children(limited level) of 001001.

2.1 Design choice one: use word cluster twice to score and select candidates

At first, we applied word cluster on two steps in our QA system like Fig1: score first time(counting overlapped words between noun phrases in question and sentence words) and score second time(for selected candidates, counting overlapped words between all non-function words in question and them). But after we exhaustively tried several different parameters values, we still can not get any MRR > 0.263. So we turned to second design choice.

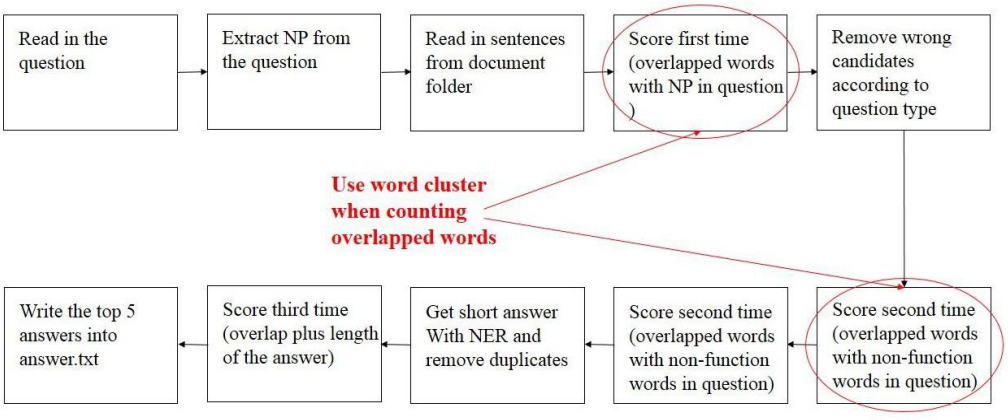

Fig1 design choice one flowchart

2.2 Design choice two: use word cluster only on second scoring

We figured the reason of our failure may because that word cluster for noun words are not suitable for our system. The first scoring is counting overlapped words between noun phrases in question and sentence words. So we does not apply word cluster here. The second scoring is to count overlapped words between all non-function words in question and selected candidates.So we just apply the word cluster here(more detailed analysis in part 3 of the Experiments). The flowchart is like below:

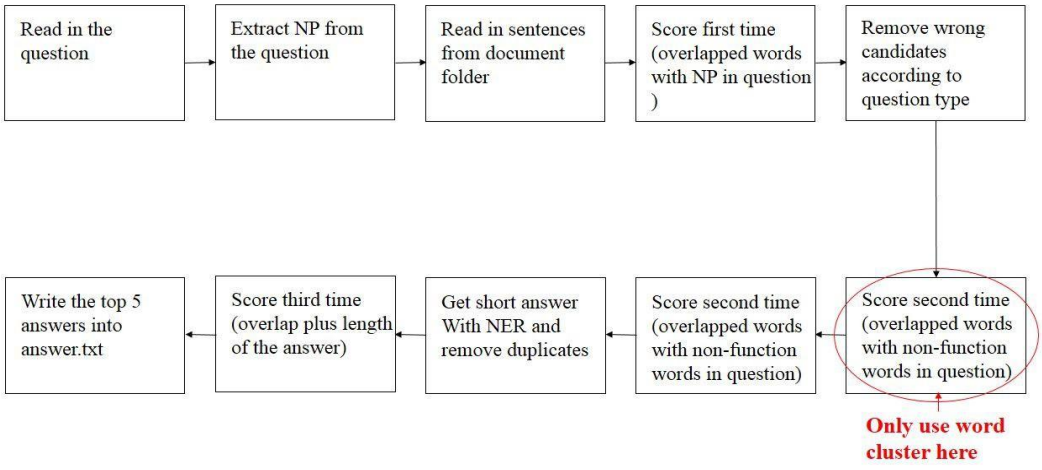

Fig2 design choice two flowchart

Experiments

1. Noun phrase improvement

For the same other parameters like project 3: 25 (number of nouns recorded in the first pass), 15 (number of words kept in the final pass) and 1.0 (final pass weight). We got MRR = 0.176 last time and MRR = 0.263 after this noun phrase improvement.

2. Word cluster improvement.

First, we set all the other parameters as experiment of noun phrase improvement. And we change the upsearch value, downsearch value and clusterfactor. Our results shows that the bigger of the word cluster range results in the worse result. So we just restrict the word cluster within the same code. The clusterfactor means that if the sentence contains the same words in the question, we add 1 point for each word for the sentence.If the words in the cluster appears in the questions, we add 0.9 or 0.8 or 0.7(clusterfactor) for each word for scoring the sentence performance.

Design choice one:

Cluster factor	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
MRR	0.263	0.255	0.255	0.255	0.255	0.251	0.239	0.240	0.237	0.237

Then we tried design choice two with same clusterfactor 0.8:

Without word cluster	0.263
Design choice one	0.237
Design choice two	0.268

The result are finally better than the one without word cluster. We than changed the value of number of nouns recorded in the first pass from 25 to 50 and got MRR = 0.271. If we changed the clusterfactor here to 1, the system will give us MRR = 0.272.

3.Analysis

From the results above, we can see that the noun phrase improvement are very successful. The reason has been stated in the first part of Improvement of previous work. As for the word cluster, it improves a little bit on the overall performance of QA system. But in design choice one, the results are even worse than original system.

This may be caused by that word cluster for noun words are not suitable for our system. For example, if the question is “Who is the leader of India?” The word cluster of India is country names like America or China. But if we give sentences like “The leader of America” high score, it will mess up our candidates pool. Meanwhile, word cluster of verbs or other stuff may be helpful for our system. The sentence “He said he likes dogs” is very similar with “He announced that he likes dogs.” We can see from this example again that “He said he likes cats” means very different from the original sentence.

Back to our system, the first scoring is counting overlapped words between noun phrases in question and sentence words. The second scoring is to count overlapped words between all non-function words in question and selected candidates. These candidates already have relatively more overlapped noun words with question. Now applying word cluster will take other part of speech words like verbs into consideration. So the second design choice is more reasonable.