Question Answering System

Pei-Yun Sun (Student ID: 667816)
Kaggle Username: peiyuns, Team Name: Pei-Yun Sun

1. Introduction

The Question Answering (QA) System is used to find the answer to a question within the given document. In this report, two designs of the QA System will be introduced: a basic rule-based method, and an advanced method with deep-learning.

2. Preprocessing

2.1. Method

Before applying basic or advanced method, the documents need to be preprocessed, and the information retrieval (IR) techniques can be used for finding the best paragraph for each question. (Schütze, H. et al., 2008)

First, each document is treated as a collection, and each paragraph is treated as a pseudo-document. For each paragraph, a bag of words is obtained by tokenizing the whole paragraph, lower-casing and lemmatizing each word, removing punctuations and stop words, and finally counting the frequency of each term.

Next, a TF-IDF matrix can be formed for each document and normalise each row by the vector length (Euclidean norm), where each row corresponds to a paragraph, and each column corresponds to a term.

However, the normalised TF-IDF matrix is a sparse matrix (many zeros), to solve the sparsity, it can be converted into the inverted index where the posting lists are made up of (paragraph_index, weight) pairs, and therefore both storage and querying will be more efficient.

Then, the score of each paragraph for a query can be obtained by adding up the weight of each query keyword in the paragraph. So, the paragraph with the highest score will be selected as the best paragraph for that query.

2.2. Evaluation

The best paragraph for each query can be predicted with the inverted index by iterating through the development set. The resulting accuracy is about 73.68% by comparing the prediction with the “answer_paragraph”.

3. Basic method

3.1. Method

The same method for finding the best paragraph can also be applied to find the best sentence. After obtaining the best sentence, which is a smaller span, filtering might be a simple and efficient method to find the answer tokens.

The best sentence can first be filtered by removing the stop words and punctuation. Then, by observing the answers in the development set, it can be concluded that the answers follow 2 simple rules:

1. Mostly nouns or numbers, and sometimes foreign words
2. Often not containing the terms which appeared in the question

Therefore, the more specific answers could be obtained by removing the question words and the words that are not nouns, numbers, or foreign words from the answer.

3.2. Evaluation (Error Analysis)

The average f-score, recall, and precision evaluated with the combination of training and development set, and the Kaggle score of the test set by using the advanced method is shown below.

F-score	0.0914	**Recall**	0.2571	**Precision**	0.0633	**Kaggle**	0.18149

The scores calculated with development set only is shown below. (to compare with advanced)

F-score	0.0834	**Recall**	0.2421	**Precision**	0.0570	**Kaggle**	0.18149

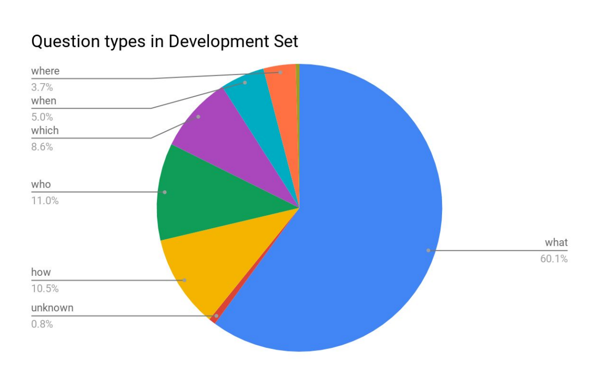
The basic method has a good average recall, but a very low average precision, which means that many of the predicted answers contain the words of the golden standard, but it contains too many redundant words. Therefore, the answers can be improved by using an advanced filtering method.

4. Advanced method

4.1. Method

In this section, in order to improve the prediction, a more sophisticated filtering method will be introduced. This method is based on categorizing the question types and predicting the expected answer type with *spacy* named entity tagging.

First, it founded that the questions in the development set can be categorized into 7 main question types: “what”, “who”, “how”, “which”, “where”, “when”, and “or”, others will be just labelled as “unknown” question types.

Question Type	**Answer Type (NE)**	
who	PERSON	
where	GPE, LOC, FAC	
when	DATE, TIME	
how - much	QUANTITY, CARDINAL, MONEY	
how - number	QUANTITY, CARDINAL, PERCENT	

For some simple question types, the expected answer type can be easily predicted by the table on the above.

For the ambiguous types: “what”, and “which”, “how”, the subtype will be required. (Moldovan D., 2000) For the first two question types, the subtypes can be defined as the first noun came after “what” or “which”. For the third type, “how”, the subtypes could be: “much” if the question if “how much”, “number” if it seems to be asking for a quantity, and “unknown” for others.

For the special question type, “or”, the questions are in the form “... A or B ?”, and the answers are usually one of “A” or “B”. Therefore, the answers for “or” questions can be improved by using “A B” as the answer instead of the contents in the best sentence.

The question types “what” and “which”, the answer NE type can be predicted by training a multiclass classifier. In this QA system, Decision Tree Classifier used since it has better performance than the Gaussian Naive Bayes Classifier for “what” questions.

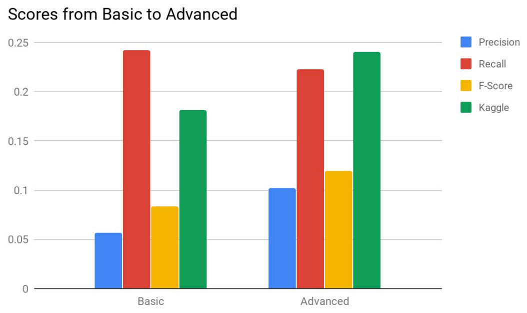
Model	Score of predicting “what”	Score of predicting “which”
Decision Tree	0.4565	0.6111
Gaussian Naive Bayes	0.0168	0.6111

4.2. Evaluation

The following table is the average f-score, recall, and precision evaluated with the development set, and the Kaggle score of the test set by using the advanced method.

F-score	0.1197	**Recall**	0.2224	**Precision**	0.1020	**Kaggle**	0.2405

By evaluating both basic and advanced methods with the development set, it is founded that, the advanced method has better f-score, precision, and Kaggle score, but a slightly lower recall than the basic method. This is because some of the answer tokens may ben incorrectly filtered by the advanced method. There is a trade-off between precision and recall (Buckland M. et al., 1994), in this case, the recall is lowered, but the precision is doubled, so the advanced method is better.



5. Conclusion

The QA system may be further improved by considering the synonyms of the query terms (Hermjakob, U. et al., 2002), learning the question patterns (Ravichandran, D et al., 2002), or using the Deep Learning methods (Yu A. et al., 2018). The synonyms might improve the accuracy of finding the best matching paragraphs and sentences since query terms could also match with its synonyms in the documents rather than the exact same terms only. On the other hand, as the current accuracy of predicting is relatively low, learning the question patterns and using the deep learning models could help with predicting the answer types.

Reference

Buckland, M., & Gey, F. (1994). The relationship between recall and precision. *Journal of the American society for information science, 45*(1), 12.

Hermjakob, U., Echihabi, A., & Marcu, D. (2002). Natural Language Based Reformulation Resource and Wide Exploitation for Question Answering. In *TREC* (Vol. 90, p. 91).

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R., & Rus, V. (2000, October). The structure and performance of an open-domain question answering system. *In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (pp. 563-570). Association for Computational Linguistics.

Ravichandran, D., & Hovy, E. (2002, July). Learning surface text patterns for a question answering system. *In Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 41-47). Association for Computational Linguistics.

Schütze, H., Manning, C. D., & Raghavan, P. (2008). Introduction to information retrieval (Vol. 39). Cambridge University Press.

Yu, A. W., Dohan, D., Luong, M. T., Zhao, R., Chen, K., Norouzi, M., & Le, Q. V. (2018). QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *arXiv preprint arXiv:1804.09541.*