COMP90042 Project 2018: Question Answering

Kaggle username: ycao44 Team Name: Yue Cao Student Number: 843282

1. Introduction

Question answering (QA) is an area which aims to automatically find a short phrase or sentence which precisely answers a natural language question (Prager et al. 2000).

This report aims to build a predominantly retrieval based QA system which can find the answer to a question in the corresponding text span given a question and a document. In this project information retrieval, information extraction, and machine learning technologies are implemented to develop and enhance the QA system.

2. Methodology of Basic QA System

The QA system employs a pipeline architecture which contains three main modules, sentence retrieval module, question processing module and answer extraction module.

2.1. Sentence Retrieval

Given the document ID, this module uses Okapi BM25 based information retrieval function to rank candidate sentences according to the relevance of a question. The result of this module is a set of best matching sentences, which have the highest Okapi BM25 score for the corresponding question.

In the preprocessing step, sentence segmentation algorithm is used to split each passage into sentences. Then each sentence is tokenized and lowercased. Punctuation and stopwords are removed and tokens are stemmed to form the final word representation for each passage.

In this module, each sentence will be treated as a paragraph. The reason why a sentence is chosen as the candidate span for each question is that most questions in the training set are factoid questions which only require simple facts as answers, thus the length of a sentence is appropriate for later named entity extraction and answer ranking. Otherwise, under the limitations of this basic QA system, overlong candidate span could bring extra complexity in the answer processing phase and may result in lower performance.

As for the parameters, k1, k3 and b, in the Okapi BM25 term weighting formulas, under the absence of parameter optimisation, their values are chosen based on existing studies. Thus, k1 = 1.2, k3 = 1.5, and b = 0.75 (Manning, Raghavan & Schütze 2008, p. 232).

While the final answer to each question should be identical to the gold answer, this module will finally obtain the corresponding sentence from the original document set as the best matching sentence.

2.2. Question Processing

Question type detection helps identify answer types, which is useful for later answer extraction module. In the basic QA system, a set of simple handwritten rules combines both 'wh-words' that appear in questions and Stanford 7 classes Named Entity Recognizer (NER) (Finkel, Grenager & Manning 2005).

When	Who/Whose	Where	How many/much	What/Which
Time/Date	Person	Location	Percentage Money	Organisation

Table 2. Question classification types of basic QA

2.3. Answer Extraction

In basic QA system, only named entities from the best matching sentences are treated as candidate answers. After applying Stanford NER on the candidate text span, BIO tags will be used to find the boundaries of named entities. Then the Stanford NE output will be converted into NLTK trees to get continuous chunked named entities. The basic QA will return the first named entity which matches the question type in section 2.2.

3. Error Analysis

1	Sentence retrieval errors: some 'best matching sentences' are not the ones contain the current answers to the questions.
	Inappropriate question type classification due to multiple wh-words or incomplete handwritten rules, especially for **what/which** questions. Unlike explicit intents of 'who', 'when' or 'where' questions, 'what/which' question may seek answers of various types.
3	When more than one named entities match the question type, the returned first NE may not be the correct answer due to the absence of an answer ranking mechanism.
4	Best matching sentences may not contain any named entity, thus simply returns 'unknow' lowers the recall of the system.
5	Numbers except for date, time, percentage and money are not extracted as candidate answers for numerical questions.
6	Some answers are named entities which also appear in the questions, which are unlikely to be the correct answers.

Table 4. error analysis of basic QA system

4. Enhancements to the basic QA

4.1 Enhancement to question processing - question classification using machine learning

The handwritten rules for question classification are coarse and incomplete, which cannot cover all the situations. Therefore, a question type classifier is needed. Unfortunately, the training set does not contain question type annotations.

Label the training set

Hand-labelling the training set requires an enormous amount of work, to do the labelling work more effectively, the Stanford 7 class NER is applied: if an answer is a named entity which appears in the best matching sentence, the named entity type of the answer will be used as its question type label. For answers that are not named entities, their question type will be labelled as 'other'.

Extract features

Five Features used for machine learning include wh-word, the POS tag of wh-word, the subsequently adjacent word (the token follows the wh-word), the POS tag of the adjacent word, and the headword (here the first noun after the question's wh-word) (Jurafsky, D & Martin, JH 2017). The reasons are as follows.

The wh-word contains rich information of the intent of the question. For example, who question usually seeks for a person rather than a place. The POS tag of the wh-word helps to identify whether the wh-word is a wh-determiner (WDT), wh-pronoun (WP), possessive wh-pronoun (WP$), or a wh-adverb (WRB). The adjacent token can work with wh-word as bigrams. For example, 'how many', 'what value'. The POS tag of the adjacent token provides useful information of the constituent of the token. The headword also gives extra information about answer type (Jurafsky, D & Martin, JH 2017). For instance, 'name' is the headword of 'what is the *name* of …'.

Build and train the question classifier

To avoid error propagation, only 9471 questions which labelled with named entity types are used to build the question classifier. Two machine learning models are implemented, Naïve Bayes and Logistic Regression. For questions in development set which labelled as named entity types, the Logistic Regression classifier performs slightly better than the Naïve Bayes model, the results are shown below.

Classifier	Logistic Regression	Naïve Bayes
Accuracy	0.803	0.788

Table 3. The accuracy of classifiers on NE type queries

4.2 Enhancements to answer extraction

4.2.1. Utilise POS tag to obtain extra candidate answers

For sentences do not contain any named entity, any token follows prepositions('IN' POS tag) and is not 'the' will be extracted as candidate answers. Because preposition is often used to link nouns, pronouns and phrases, which may be the correct answers.

For Num type questions, which start with 'how many', 'how much' etc, obtaining numerical candidate answers ('CD' POS tag) from best matching sentence.

4.2.2. Conduct answer ranking

Answer ranking will be conducted based on a few metrics. First, candidate answers which has appeared in the question will be ranked lowest. Second, candidate answers match the question type will be ranked higher. Third, for candidate answers belong to the same question type, if a named entity appears in both best matching sentence and the question, the candidate answer which is closer to the named entity will be ranked higher. Because, answers are more likely to sit close to a content word. It is noted that the distance between a NE to itself will be '1000' as penalise.

5. Evaluation

5.1. Evaluation Metrics

In this section, accuracy and micro-average F1 score will be used.

5.2. Results and discussion

The measurements of basic QA system and enhancements are shown in Table 4. The final QA system makes use of some additional syntactic features and has get improved average F-score. It is proved that it is important to understand the syntactic constituents for both documents and question to correctly answer a question.

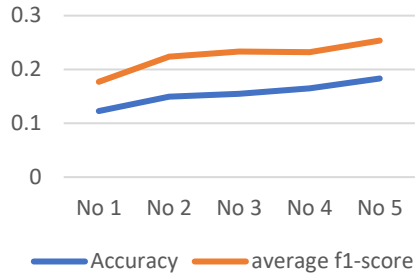

Figure 1. The measurements on development set
(*No refers to the corresponding system in Table 4)

Despite incorrect identified match text span errors, a large part of errors come from insufficient interpretation of the syntactic or semantic relationships between question and expected answer. For example, ordinal number phrases, such as 'the second largest', cannot be identified under the current answer extraction module. Because it's belongs to adjectives other than any named entity nor cardinal numbers. answers, it is difficult to obtain under the final QA system.

6. Conclusion and Future Work

In this report, I have built a basic QA system and made enhancements to develop a final IR-based QA system based on error analysis. The final QA system performs well on questions which answers are named entities. But for more complex questions, the performance is unsatisfied.

For further improvements, building a pattern-based approach for answer extraction may be helpful. Beside, semantic information could also be utilised to improve answer type detection classifier, such as 'what' question.

No	Combinations	QA system & enhancements	Accuracy of development set	Average F1 score of development set	Average F1 score of testing set (Kaggle)
1	1	2. Basic QA system	0.12270	0.17704	0.16213
2	1+2	4.2.2. Conduct answer ranking	0.14940	0.22398	0.18761
3	1+2+3	4.1. Apply logistic regression	0.15460	0.23314	0.19144
4	1+2+3+4	4.2.1. apply IN POS tag	0.16466	0.23193	0.22617
5	1+2+3+4+5 (the final QA)	4.2.1. apply CD POS tag	0.18317	0.25368	0.24804

Table 4. The performance of basic QA and enhancements

References

Finkel, JR, Grenager, T & Manning, C 2005, Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363-370.

Kadam, S, 2017, NLP: QUESTION CLASSIFICATION USING SUPPORT VECTOR MACHINES [SPACY][SCIKIT-LEARN][PANDAS], viewed 22 May 2018, <https://shirishkadam.com/2017/07/03/nlp-question-classification-using-support-vector-machines-spacyscikit-learnpandas/>.

Jurafsky, D & Martin, JH 2017, Chapter 28 Question Answering, *Speech and Language Processing* 3rd edition.

Manning, CD, Raghavan, P & Schütze, H 2008, *Introduction to Information Retrieval*, Cambridge University Press.

Prager, J, Brown, E, Coden, A, & Radev, D 2000, 'Question-answering by predictive annotation', *Question-answering by predictive annotation*, pp. 184-191.