COMP90042 Project 2018: Question Answering

Shimei Zhao(ShiMeiZhao) 849781. Chaoyi Han(hhhhph123) 890593. Jiyu Chen(jiyuc) 908066.

1. Introduction

Two Question Answering Systems were implemented and compared their performances in finding concise answers to factoid questions from 440 pieces of Wikipedia documents. One system leverages TF-IDF, rule-based and term-closeness methods. Another system leverages distributed semantics, machine learning and term co-occurrence methods.

2. Method

2.1 System Overview

- TF-IDF, rule-based and term-closeness based QA system:
    - Apply paragraph retrieval and ranking based on BM25 model.
    - Apply rule-based pattern match answer type recognition on query.
    - Apply POS tagging on each word in the paragraphs and extract noun-phrase chunks.
    - Extract a list of potential answers given answer type.
    - Rank answers by calculating the closeness between answer term and question keywords in paragraph boundary.
    - Finalize answer and output result.
- Distributed semantics, machine learning and term-co-occurrence based QA system:
    - Train word2vec model on given Wikipedia documents.
    - Transform word embeddings to sentence embeddings by taking the average of word embeddings weighted by term TF-IDF created by Le et al[5].
    - Apply paragraph retrieval and ranking based on semantic cosine similarity.
    - Apply logistic regression and rule-based model in answer type classification on query.
    - Apply named-entity-recognition in answer extraction.
    - Rank answers based on semantic similarity and number of co-occurrence times with query terms in a sentence boundary.
    - Finalize answer and output result.

2.2 Error Analysis

This section will summarize drawbacks and errors in previous system versions.

- Paragraph Retrieval:

  Initially, basic TF-IDF model were built on paragraph level and its performance was tested on different Top-K level. (e.g. Five ranked paragraphs will be retrieved on a Top-5 level). The retrieval tests on Top-1, Top-2, Top-5 and Top-10 level were shown in Table 2.2.1.

Top-K	1	2	5	10
True Relevant paragraphs	703	828	875	901
Total paragraphs	1000	1000	1000	1000
Paragraph Retrieval Recall	70.3%	82.8%	87.5%	90.1%
Paragraph Retrieval Precision	70.3%	41.4%	17.5%	9.01%
Answer Extraction Accuracy	6.8%	6.85%	6.83%	6.82%
Running times	6 mins	29 mins	1.7 hours	7.1 hours

Table 2.2.1

Table 2.2.1 shows basic TF-IDF model is low-accuracy in retrieving true relevant paragraph on Top-1 level, and time consuming, low-precision on higher Top-K level. Considering a concise answer extraction will be applied to the retrieved paragraphs and high level of Top-K lead to low-precision, Okapi BM25 is applied to weight TF-IDF and improve precision in later system version. The illustration will be in section 2.3.

- Named Entity Recognition:

  Initially, 7-class Stanford NER model was applied to named-entity recognition. The problem is many terms were false recognized. For instance, 'John Smith' is recognized as "OTHER" instead of its true tag "PERSON". Moreover, with only 7 named-entity classes, many terms are labeled as "OTHER" which caused low-precision in later answer extraction. Hence, we changed the NER model into industrial-strength spaCy[6] NER model with an accuracy of 85.85% and contains 18 entity classes.

- Answer Type Recognition:

  Rule-based pattern match method was firstly used in answer type classification. However, most queries are tagged as "OTHER" and simple pattern match is unpromising in dealing situations when two types of terms overlap in the same query. For instance, "What composer…?" contains patterns indicate both "what, OTHER" and "composer, PERSON". To solve this, a pure semantic based machine learning method is implemented. However, since factoid questions contains very explicit features on pattern level such as "when" indicate "DATE" while semantic features are easily biased by noises in the sentence. Therefore, a semantic machine learning created by Li et al[7] and rule-based pattern match hybrid classifier was implemented.

- Answers contain Stop-words:

  Gold answers contain stop-words while text preprocessing removes them from corpus. A statistical analysis is applied to gold answers calculating stop-words existence rate. The existence rate on training data is 0.16 and on development data is 0.14 which significantly influence the answer's consistency.

- Noun-phrase Chunk Extraction:

  A part of our work is merging contiguous words with the same tags and extract them as noun-phrase chunks. However, due to the named-entity recognition errors, we cannot always get the correct noun-phrase chunk.

2.3 Enhancement and Technique

This section will summarize enhancements and techniques in two final system versions.

- Enhanced Paragraph Retrieval Model:

  Okapi BM25 model increase the accuracy of paragraph retrieval. The performance to the original TF-IDF method on Top-1 level is shown in Table 2.3.1.

	TF-IDF	Okapi BM25
Accuracy	0.71	0.83

Table 2.3.1

Paragraph retrieval based on BM25 model out-performs sentence2vec distributed semantics model. The performance comparison of two methods on Top-1, Top5 and Top-10 level is shown in Figure 2.3.2.

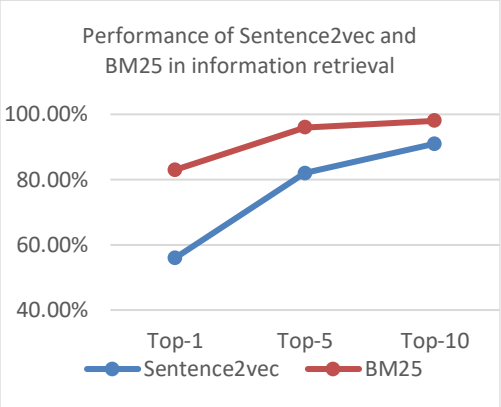

Figure 2.3.2

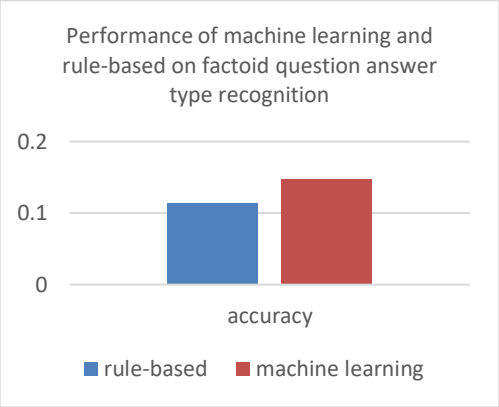

Figure 2.3.3

- Hybrid Answer type Recognition:

  Rule based answer type recognition is unpromising and expensive as the texture form of queries are diverse. A pure machine learning and semantic based classifier is also unpromising with factoid questions because some words as “when” obviously indicate an answer type of “DATE” while the classifier may be biased by other words occur in the text. Hence, we implemented a hybrid rule-based and machine learning recognizer and compared their performances on the second QA system. Figure 2.3.3 shows the machine learning answer type recognizer is better.

- Enhanced named-entity recognition:

  In order to get more precise entity tags, we changed named entity recognition package from Stanford NER to spaCy [6] which contains 18 NER tags. The performance of two NER methods on the second QA system is shown in Table 2.3.4.

	Stanford-NER	spaCy-NER
System Accuracy	0.08	0.11

Table 2.3.4

- Solution to Answer type of OTHER:

  Most of OTHER answer types are denoted by the term of “what” and most of valid OTHER type answers are either nouns or noun-phrase chunks. Hence, the strategy in OTHER type answer extraction is by applying POS tagging to each term in the text and reform them as individual noun or phrases end up with noun. The performance

on the first QA system shown in Table 2.3.5.

	Before POS tagging	After POS tagging
System Accuracy	0.9%	14.4%

Table 2.3.5

- Answer Ranking:

  Two systems use two answer ranking strategies:

  - Term closeness:

    Here we define the key words as the nouns, verbs and number in the query. First of all, we find the index positions of the key words and potential answer in the paragraphs, then we add up all absolute values of key words index minus potential answer index. The result is a dictionary of potential answer and its "closeness", we chose the potential answer that is closest to query as expect answer.

  - Term co-occurrence:

    Query terms and their denoted answer are very likely occurring in the same sentence. Hence, answers will be ranked by number of times of co-occurrence with query key terms in a sentence boundary. Initially, same punishment values were given to each answer. Then a loop process will start on the retrieved paragraph in sentence level for each potential answer. When a term in the query co-occur with an answer, answer's punishment value will minus the query term weight and answer weight based on TF-IDF. After the process, answer having the least punishment value will be ranked highest and finalized as system output.

3. Conclusion

Two systems are compared on same test set and scored on Kaggle. Their performances are shown in Table 3.1. QA I denote TF-IDF, rule-based and term-closeness based QA system, QA II denote Distributed semantics, machine learning and term-co-occurrence based QA system.

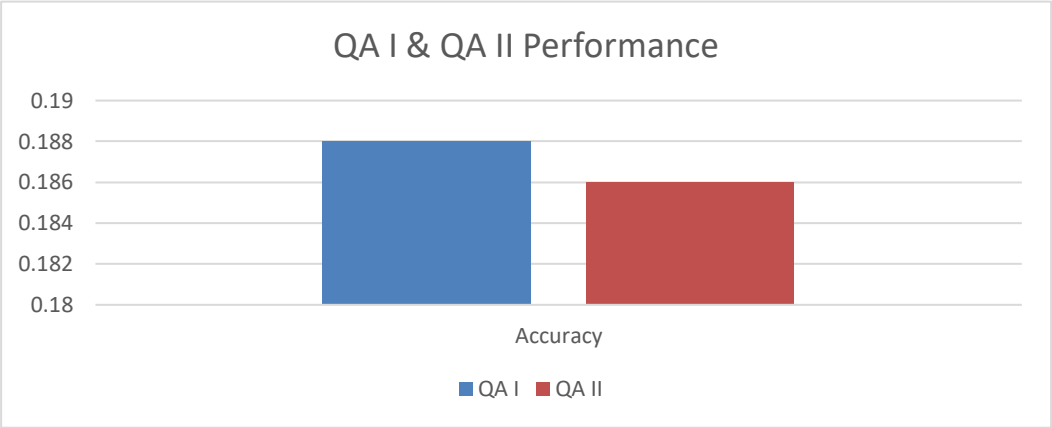

Table 3.1

To sum up, QA I slightly out-performance QA II in finding concise answers to factoid questions on test dataset.

Reference:

[1] Khalid, M.A. and Verberne, S., 2008, August. Passage retrieval for question answering using sliding windows. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering* (pp. 26-33). Association for Computational Linguistics

[2] Answer Extraction and Passage Retrieval for Question Answering Systems(Waheeb Ahmed)

[3] Jinguji, D., Lewis, W.D., Efthimiadis, E.N., Minor, J., Bertram, A., Eggers, S., Johanson, J., Nisonger, B., Yu, P. and Zhou, Z., 2006, October. The University of Washington's UWclmaQA System. In *TREC*.

[4] Larson, T., Gong, J.H. and Daniel, J., Providing A Simple Question Answering System By Mapping Questions to Questions.

[5] Le, Q. and Mikolov, T., 2014, January. Distributed representations of sentences and documents. In *International Conference on Machine Learning* (pp. 1188-1196).

[6] spaCy, an open-source software library for advanced Natural Language Processing. https://spacy.io

[7] Li, X. and Roth, D., 2002, August. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1* (pp. 1-7). Association for Computational Linguistics