Biomedical Text Mining about Alzheimer's Diseases for Machine Reading Evaluation

Bing-Han Tsai, Yu-Zheng Liu, and Wen-Juan Hou*

Department of Computer Science and Information Engineering,
National Taiwan Normal University, Taipei, Taiwan
{60047039S,60047061S,emilyhou}@csie.ntnu.edu.tw

Abstract. The paper presents the experiments carried out as part of the participation in the pilot task of Biomedical about Alzheimer for QA4MRE at CLEF 2012. We have submitted total five unique runs in the pilot task. One run uses Term Frequency (TF) of the query words to weight the sentence. Two runs use Term Frequency-Inverted Document Frequency (TF-IDF) of the query words to weight the sentences. The two unique runs differ in the way that when multiple answers get the same scores by our system, we choose the different answer in the different runs. The last two runs use TF or TF-IDF weighting scheme as well as the OMIM terms about Alzheimer for query expansion. Stopwords are removed from the query words and answers. Each sentence in the associated document is assigned a weighting score with respect to query words. The sentence that receives the higher weighting score corresponding to the query words is identified as the more relevant sentence to the document. The corresponding answer option to the given question is scored according to the sentence weighting score and the highest ranked answer is selected as the final answer.

Keywords: question-answering, machine reading, biomedical text mining, QA4MRE

1 Introduction

The machine reading of biomedical texts about Alzheimer's diseases follows the same set up and principles as the QA4MRE, with the difference that it focuses on the biomedical domain. It is important for researchers to perform more efficient processing of Alzheimer-related literature. The task focuses on the reading of single documents and the identification of the answers to a set of questions about information that is stated or implied in the text. Questions are in the form of multiple choices, each having five options, and only one correct answer.

We have submitted total five unique runs in the pilot task. One run uses Term Frequency (TF) of the query words to weight the sentences. Two runs use Term Frequency-Inverted Document Frequency (TF-IDF) of the query words to weight the sentences. The two unique runs differ in the way that when multiple answers get the same scores by our system, we choose the different answer in the different runs. The

last two runs use TF or TF-IDF weighting scheme as well as the OMIM terms about Alzheimer for query expansion.

The paper is organized as follows. Section 2 describes the corpus we use in this experiment. Section 3 introduces the system architecture and methods we propose. We perform and discuss the evaluation results in Section 4. Finally, the conclusions and future directions are drawn in Section 5.

2 Corpus Statistics

2.1 Background Collections

We use three types of background collections provided by the pilot task. The brief introduction of background collections is stated as below.

Open Access Full Articles PMC. 7,512 articles are provided in text format from PubMed Central. These articles have been selected by performing the search and selecting the full articles that belong to the PubMed Central Open Access subset.

Open Access Full Articles PMC, Smaller Collection. There are 1,041 full text articles from PubMed Central. To select these documents, a search by the pilot task was performed on PubMed using Alzheimer's disease related keywords and restricting the search to the last three years.

Elsevier Full Articles. There are 379 full text articles and 103 abstracts from Elsevier. The articles in this subset have been selected from a list of articles provided by Professor Tim Clark from the Massachusetts Alzheimer's Disease Research.

2.2 Test Data

The test set is composed of four reading tests. Each reading test consists of one document, with ten questions and a set of five choices per question. So, there are in total forty questions and 200 choices/options.

2.3 OMIM Term about Alzheimer

1,549 entities and related genes about Alzheimer diseases have been retrieved from OMIM website [1]. We use these terms to do query expansion in Run 4 and Run 5.

3 Method

The main system architecture is illustrated in Fig. 1. The expanded system architecture is pictured in Fig. 2. Fig. 1 is the system architecture adopted in Runs 1, 2 and 3. In Run 4 and Run 5, we use the OMIM terms about Alzheimer as well as other resource to do query expansion. The detailed architecture for OMIM expanded system is shown in Fig. 2. Fig. 2 is the expanded system architecture adopted in Run 4 and Run 5. Part A is the part of the main system architecture in Fig. 1.

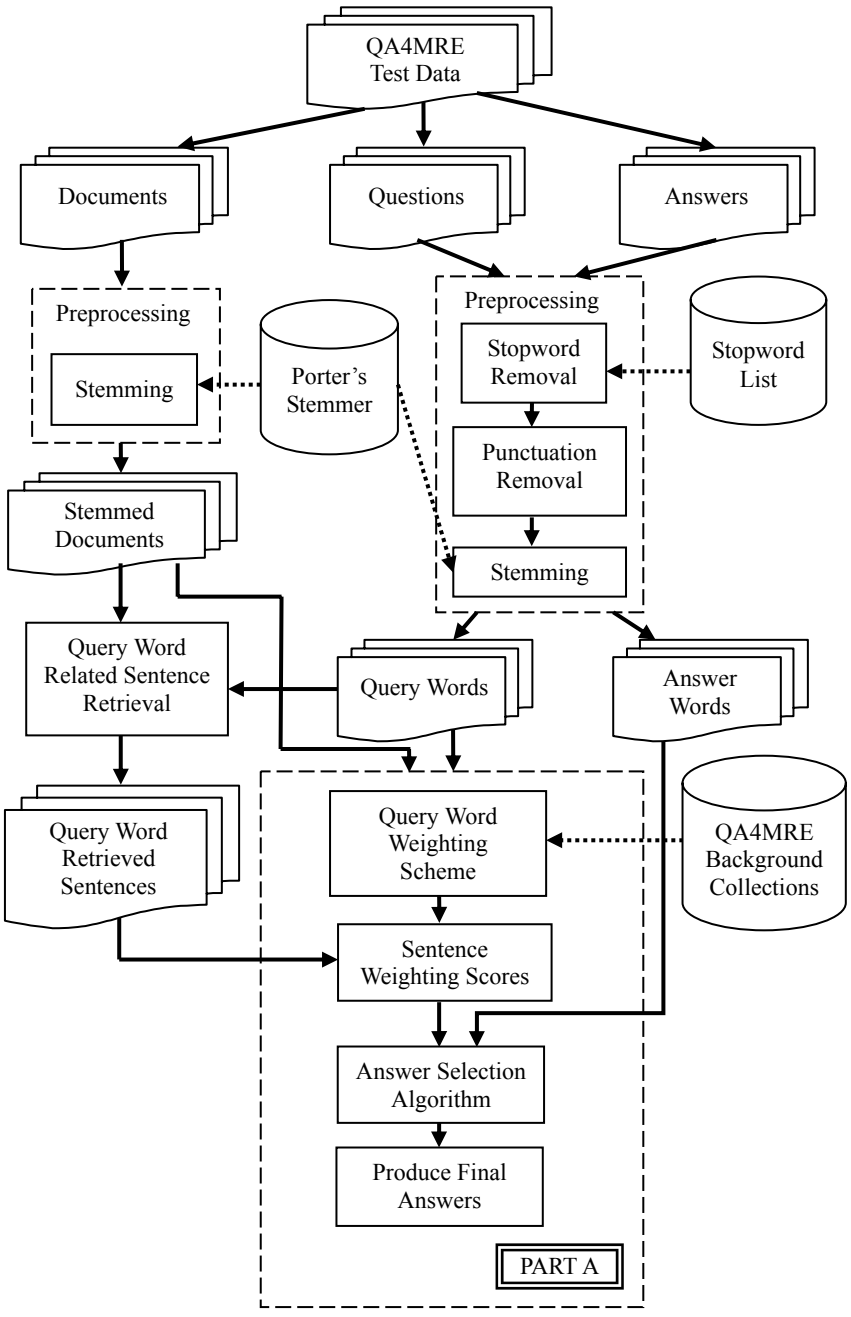

Fig. 1. Main system architecture

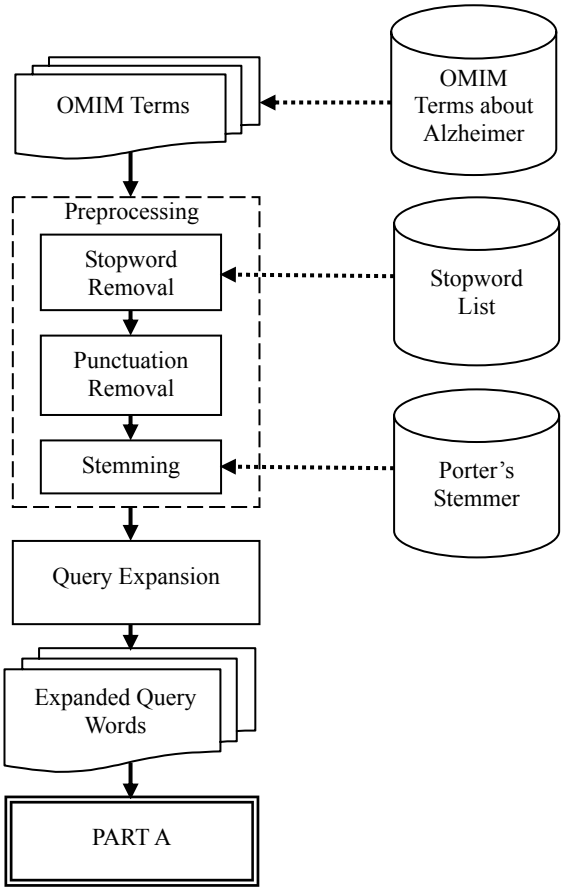

Fig. 2. Expanded system architecture for OMIM background knowledge. Part A is illustrated in the bottom part of Fig. 1.

Let's explain the details about Fig. 1. Because QA4MRE test data is provided in XML format, we have to do some format cleaning work. Hence, we first split it to three parts: (1) documents, (2) questions and (3) answers.

3.1 Preprocessing

After splitting QA4MRE test data to three parts, we need to do some processes so as not to cause implicit query handling during searching. They are described as follows.

Stopword Removal. The Stopwords are removed from each question and answer option using a stopword list [2].

Punctuation Removal. Punctuation characters are removed from the questions and answers. For example, "http://wt.jrc.it/" and "doug@nutch.org" are rephrased as "http wt jrc it" and "doug nutch org", respectively.

Stemming. Standard Porter stemming algorithm [3] is used to stem words in documents, questions and answers.

The remaining words in the question and answer are identified respectively as the query words and answer words.

Also, we expand some key words for the questions. For example, when facing with the word "experiment" in the question, we expand the related word "show" to the question. It is because we think words "experiment" and "show" are highly related each other.

3.2 Retrieving Query Word Related Sentences

After extraction of the query words, we use it to retrieve sentences from the documents. If a query word exactly matches with words in a sentence, then we view it as the relevant sentence and retrieve it.

3.3 Query Word Weighting Scheme

Each query word is assigned a weight to determine its importance for the sentences. We use TF and TF-IDF depending on different runs as the weights of the query words. In Run 1 and Run 4, we use TF to weight the query words. The remaining runs use TF-IDF to weight the query words.

TF Weighting. The formula of TF weighting is listed in Equation (1):

$$TF_{Q_i} = 1 + \frac{f_{Q_i}}{\max\limits_{Q_i} f_{Q_i}} \tag{1}$$

where TF_{Q_i} is the term frequency of query word Q_i. f_{Q_i} is the number of Q_i appearing in the stemmed document. We assume that the weight of each query word has a baseline of 1. If a query word doesn't exist in the document, the formula will give TF_{Q_i} a value of 1.

TF-IDF Weighting. The formula of Inverted Document Frequency (IDF) is listed in the following.

$$IDF_{Q_i} = \begin{cases} \log_2 \frac{N}{n_{Q_i}} & \text{if } n_{Q_i} \neq 0 \\ 0.1 & \text{if } n_{Q_i} = 0 \text{ and } f_{Q_i} \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where IDF_{Q_i} is the inverse document frequency of query word Q_i. N is the total number of documents in the corpus (i.e., QA4MRE background collections). n_{Q_i} is the number of documents in the corpus which Q_i appears. f_{Q_i} is the number of Q_i that appears in the stemmed document. We can't ignore the importance of a query word which doesn't exist in the corpus while counting TF-IDF. So, when $n_{Q_i} = 0$ and if Q_i exists in the document, we give the inverse document frequency a value of 0.1 for the smoothing reason.

The TF-IDF formula is shown as follows:

$$TF - IDF_{Q_i} = TF_{Q_i} \times IDF_{Q_i} \quad (3)$$

3.4 Sentence Weighting Scores

If a query word matches words in the relevant sentence which we found at the sentence retrieval step, then the sentence gets the weight of that word. A sentence weighting score is calculated as Equation (4) and Equation (5):

$$SW\_TF_j = \sum_{Q_i \in S_j} TF_{Q_i} \quad (4)$$

$$SW\_TFIDF_j = \sum_{Q_i \in S_j} (TF_{Q_i} \times IDF_{Q_i}) \quad (5)$$

where SW_TF_j is the sum of TF for all query words appearing in the sentence S_j. SW_TFIDF_j is the sum of TF-IDF for all query words appearing in the sentence S_j.

3.5 Answer Selection Algorithm

According to the sentence weighting scores, we can compute each answer's score in this phase. If an answer word matches words in the sentence S_j, then its weighting value is recorded by the sentence. Each answer's score is the sum of the above values. We choose the answer with the highest score to be the final answer. If there are

multiple answers with the same highest scores, we select the different answer in the different runs.

3.6 Query Expansion

In this study, we use the OMIM Alzheimer-related terms as our extra knowledge base in Run 4 and Run 5. As shown in Fig. 2, OMIM terms are first preprocessed through stopword removal, punctuation removal and stemming. We call them as expanded query words. These expanded query words will combine with query words to compute the new weighting scores. The answer selection algorithm is the same as the approach in Section 3.5.

4 Results and Discussion

We have submitted total five runs. Run 1 uses TF of the query words to weight the sentences. Run 2 and Run 3 use TF-IDF of the query words to weight the sentences. Runs 2 and 3 differ in the way that when multiple answers have the same scores in our system, we view them as different runs. Run 4 uses TF weighting scheme and takes OMIM terms about Alzheimer for query expansion. Run 5 uses TF-IDF weighting and takes OMIM terms about Alzheimer for query expansion. In summary, the weighting methods for each run are listing as follows.

TF: Run 1
TF-IDF: Run 2, Run 3
OMIM+TF: Run 4
OMIM+TF-IDF: Run 5

The main measure used in this evaluation campaign is called c@1, which is defined as follows.

$$c@1 = \frac{1}{n(n_R + n_U \frac{n_R}{n})} \tag{6}$$

where n_R is the number of correctly answered questions, n_U is the number of unanswered questions, and n is the total number of questions.

Table 1 presents the evaluation results at question-answering level. In Table 1, Column “Run ID” identifies five runs we have submitted. Column “C1” is the number of questions our system answered. Column “C2” is the number of questions our system are unanswered. Column “C3” is the number of questions answered with right candidate answer. Column “C4” is the number of questions answered with wrong candidate answer. Column “C5” is the number of questions unanswered with right candidate answer. Column “C6” is the number of questions unanswered with wrong candidate answer. Column “C7” is the number of questions unanswered with empty candidate. Column “c@1” is the value calculated in Equation (6).

Table 1. Evaluation results at question-answering level

Run ID	C1	C2	C3	C4	C5	C6	C7	c@1
1	40	0	7	33	0	0	0	0.18
2	35	5	6	29	0	0	5	0.17
3	35	5	7	28	0	0	5	0.20
4	40	0	7	33	0	0	0	0.18
5	40	0	8	32	0	0	0	0.20

Table 2. Evaluation results at reading-test level

Run ID	R1	R2	R3	R4	Median	Average	Standard Deviation
1	0.00	0.40	0.20	0.10	0.15	0.18	0.17
2	0.00	0.40	0.13	0.12	0.13	0.16	0.17
3	0.00	0.40	0.13	0.24	0.19	0.19	0.17
4	0.00	0.50	0.20	0.00	0.10	0.18	0.24
5	0.00	0.40	0.20	0.20	0.20	0.20	0.16

Table 2 presents the evaluation results at reading-test level. In Table 2, Columns "R1", "R2", "R3", "R4" represent the c@1 measure over 4 reading tests respectively. Columns "Median", "Average" and "Standard Deviation" are the median, average and standard deviation values for the c@1 values for all questions.

From Tables 1 and 2, we observe that Run 5 is the best run over other runs. Although Run 3 has the same c@1 measure as Run 5, it also remains some questions unanswered. It shows that using OMIM terms about Alzheimer as expansion has some positive effect in this experiment.

5 Conclusion

In this study, we utilize TF, TF-IDF and OMIM terms with background collections to help for machine reading comprehension. We observe that the OMIM terms are good features for answering questions in this task and the best c@1 measure is 0.20. The results also show some improvement space.

Our future work will focus on the query expansion part. Trying to extract some related words to the questions from corpus may improve the performance of the system. Also the anaphora resolution and some semantic inference are considered in the future.

References

1. Hamosh, A., Scott, A.F., Amberger, J.S., Bocchini, C.A., McKusick, V.A.: Online Mendelian Inheritance in Man (OMIM), a Knowledgebase of Human Genes and Genetic Disorders. Nucleic Acids Res. vol. 30 (1), pp.52–55 (2002)
2. Stopword list, `http://www.lextek.com/manuals/onix/stopwords1.html`
3. Porter, M.F.: An Algorithm for Suffix Stripping. In: Jones, K.S., Willet, P. (eds.) Readings in Information Retrieval. Morgan Kaufmann, San Francisco, 313−316 (1997)