WUST System at NTCIR-12 QALab-2 Task

Maofu Liu, Limin Wang, Xiaoyi Xiao, Lei Cai
College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, P.R. China
liumaofu@wust.edu.cn, smile_wlm@163.com, 876494364@qq.com, 1181502185@qq.com

Han Ren
College of Computer Science, Hubei University of Technology, Wuhan 430068, P.R. China
hanren@whu.edu.cn

ABSTRACT
This paper describes our question answering system at NTCIR-12 on QALab-2 task, which requires solving the history questions of Japanese university entrance exams and their corresponding English translations. Wikipedia of English edition is main external knowledge base for our system. We first retrieve the documents and sentences related to the question from Wikipedia. Then, the classification model has been constructed based on SVM (Support Vector Machine) in order to solve the question by choosing right or wrong sentence in multiple choice-type questions for the National Center Test, and five kinds of features about questions and choices have been extracted as inputs to the model. Finally, we choose the answer according to the score of each choice.

Categories and Subject Descriptors
H.3.4 [INFORMATION STORAGE AND RETRIEVAL]: Systems and Software – Performance evaluation (efficiency and effectiveness), Question-answering (fact retrieval) systems.

General Terms
Experimentation

Keywords
QALab-2, question answering, knowledge base, Wikipedia, support vector machine

1. INTRODUCTION
In NTCIR-12, the goal of the QALab-2 is to investigate the real-world complex QA (Question Answering) technologies using Japanese university entrance exams and their corresponding English translations on the subject of "world history"[1]. In the QALab-1 at NTCIR-11, "world history" questions are selected from the National Center Test for University Admissions (Center Test) and the secondary exams at five universities in Japan (Secondary Exam) [1]. In the QALab-1 at NTCIR-11, most of the questions are True/False or factoid ones and the QALab-2 task at NTCIR-12 increases the number of complex questions.

Question answering is the combination of information retrieval and natural language processing, which allows users to submit a question in natural language, and then returns the definitive answer through a series of processing techniques. Question answering system generally consists of three parts, i.e. question analysis, document retrieval, and answer extraction [2], and most of the question answering systems comply with these three parts. The English baseline system[2] for QALab-2 task, provided by CMU and designed based on UIMA[3] modular question answering pipeline, can automatically answer multiple-choice question for the entrance exams on world history. The Japanese baseline QA system[4], re-created from the source codes of YNU's MinerVA [3] and CMU's Javelin [4], also includes four modules, i.e. question analysis, document retrieval, extraction of answer candidates, and answer generation.

Question answering system is a hot research direction in artificial intelligence. Wang and Nyberg [5] addressed the answer sentence selection problem by a combination of the stacked bidirectional Long-Short Term Memory (LSTM) model and keyword matching. Dong et al [6] proposed a novel approach for question answering over FREEBASE[5] using multi-column CNNs. Toba et al [7] proposed a hybrid hierarchy of classifiers framework to discover high quality answer in community question answering archives. Liu et al [8] recommended related QA documents for knowledge communities of QA websites by considering factors such as the push scores, collection time of QA system and so on. Er et al [9] solved factoid question by extending queries using answer patterns matching. Shen et al [10] used a similarity matrix to combine lexical and sequential information in order to solve QA matching. Yi et al [11] built supervised classification model and extract heterogeneous features for answer selection and YES/NO response inference in community question answering. Sometimes, text can't provide intuitive answer for questions, while media data are more appropriate. Nie et al [12] enriched text answers with image and video information by image and video search reranking. Further, by processing a large set of QA pairs and adding then to pool, Nie et al [13] proposed a novel approach that can find multimedia answers by matching their questions with those in the pool.

The remainder of this paper is organized as follows. Section 2 delineates our system architecture in details. Section 3 describes our evaluation results on the formal run corpus of QALab-2. Finally, we conclude our paper in section 4.

2. SYSTEM DESCRIPTION
Our System includes six main modules, i.e. data preprocessing, question analysis, document retrieval, feature extraction, SVM classifier, answer generation. Figure 1 can illustrate our system architecture in detail.

[1] http://research.nii.ac.jp/qalab/

[2] https://github.com/oaqa/ntcir-qalab-cmu-baseline

[3] http://uima.apache.org/

[4] https://bitbucket.org/ntcirqalab/factoidqa-centerexam/

[5] http://www.freebase.com/

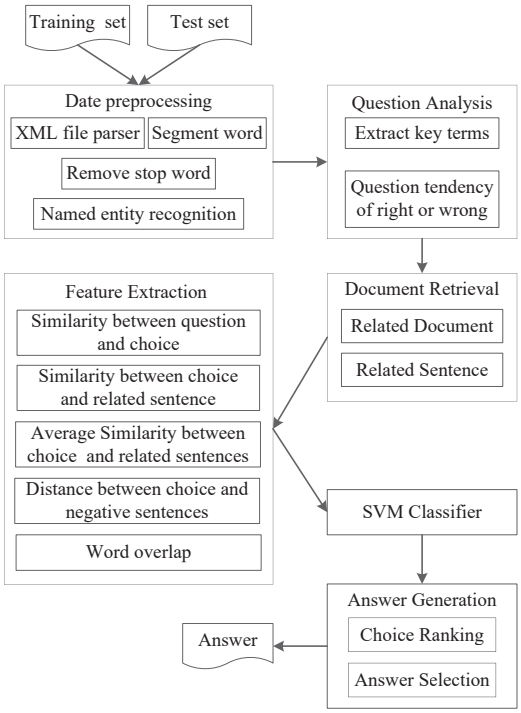

Figure 1. System architecture

2.1 Data preprocessing

In the data preprocessing, the main work of the system is to parse the question XML file and process the question sentences. The question XML file parser is mainly to obtain some nodes and attributes of the question, such as 'question', 'choices', 'anscol', 'answer_type' and so on. The question sentences processing is to segment the English words, remove the stop words and recognize the named entities. We choose Stanford tool[6] to achieve the word segmentation and named entity recognition and use the stop list from CCF (China Computer Federation) web site[7] provided by Harbin institute of technology to remove the stop word.

2.2 Question analysis

Question analysis mainly includes extracting key terms in the question and judging the question tendency of right or wrong. When we need to find the related documents of a question from Wikipedia, the query expression will be formed by the extracted key terms. We use two methods to extract key terms, one is extract proper nouns based on the results of the named entity recognition, including person names, place, organization, namely with a specific meaning of the entity, and the other is through the TF-IDF (Term Frequency–Inverse Document Frequency) method to select key terms. After segmenting word and removing the stop words, we sort the words according to their TF-IDF scores, and then select the key terms in turn.

Multiple-choice question will be asked to choose the right or wrong choice, namely the question tendency of right or wrong. We judge the question tendency through the method of manual configuration dictionary. When the question stem requires choosing the wrong answer, for example, the question contains the keywords 'mistake', 'incorrect', 'incorrectly' or 'did not exist', we regard it as a wrong tendency of question. On the contrary, if the question requires choosing the right answer, we regard it as a right tendency of question. Judging the question tendency of right or wrong has a great help for the answer generation phase.

2.3 Document retrieval

In the QALab-2 task, Wikipedia and high school textbooks about world history are provided, but the high school textbooks are only available in Japanese. Therefore, our system only utilizes Wikipedia as external resource, and all Wikipedia title and its corresponding Wikipedia document have been extracted in our system.

Before getting the Wikipedia documents related to the question, we need to get the relevant title in the question. Given a question, we use two methods to find the relevant Wikipedia title. One is to directly extract the Wikipedia titles that are included in the question and the other is to use Lucene to build a search engine to search relevant titles. Figure 2 can show the retrieval phase.

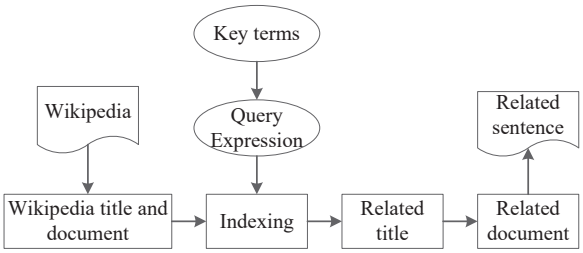

Figure 2. Retrieval phase

In retrieval phase, we first obtain Wikipedia title set and its corresponding Wikipedia document through the XPath query language from the Wikipedia resource, and establish inverted indexing file for Wikipedia title set and Wikipedia documents after segmenting sentence respectively. Then we find out the related titles from the title index file according to the query expression constructed by key terms extracted from question analysis phase. Finally, we locate the related sentence to question from the Wikipedia document index file.

2.4 Feature extraction

In this subsection, we mainly focus on five kinds of features used in our system, including the similarity between question and choice, the similarity between choice and related sentence, the average similarity between choice and related sentences, the distance between choice and negative sentences, and word overlap.

(1) Similarity between question and choice

We use cosine similarity, showed in formula (1), to calculate the similarity between the choice sentence and question sentence.

$$sim(\vec{q},\vec{c}) = \frac{\vec{q}\cdot\vec{c}}{\sqrt{\sum_{i=1}^{n} q_i^2} * \sqrt{\sum_{i=1}^{n} c_i^2}} \quad (1)$$

Where $\vec{q}$ and $\vec{c}$ are the vector representations of the question sentence and the choice sentence and the vector is built using the term frequency, the parameter n is the dimension of the vector, and the q_i and c_i denote the i[th] dimension of the vectors $\vec{q}$ and $\vec{c}$ respectively.

[6] http://stanfordnlp.github.io/CoreNLP/

[7] http://www.ccf.org.cn/sites/ccf/ccfdata.jsp

(2) Similarity between choice and related sentence
Our system extracts the relevant sentences from Wikipedia document related to question. We choose the first sentence from the set of related sentences and use cosine similarity in formula (1) to calculate the similarity between a choice sentence and the first related sentence.

(3) Average similarity between choice and related sentences
We choose the top-5 sentences from the set of related sentences. The value of this feature is the average of the similarities between choice sentence and the top-5 sentences. We use formula (2) to calculate the average similarity.

$$avgsim = \frac{\sum_{i=1}^{5} sim(\vec{c}, \vec{s_i})}{5} \quad (2)$$

Where the value of $sim(\vec{c}, \vec{s_i})$ is calculate by formula (1).

(4) Distance between choice and negative sentences
Euclidean distance represents the true distance between two points in m-dimensional space. Euclidean distance is regarded as the similarity degree of the signal, and the closer distance, and the more similar. We get negative sentences that contain negative words, such as 'isn't', 'didn't', 'does not' and so on, from the related documents. An example of a negative sentence is shown as follows.

Example 1:
The film was not shown in any other Los Angeles theater during that year.

We hold the assumption that if the distance between a choice sentence and a negative sentence is smaller, and then it is more likely to be the wrong answer. We use formula (3) to calculate the distance of a choice sentence and a negative sentence.

$$D(\vec{c}, \vec{s}) = \sqrt{\sum_{i=1}^{n} (c_i - s_i)^2} \quad (3)$$

Where $\vec{c}$ and $\vec{s}$ are the vector representations of the choice sentence and the negative sentence and the vector is built using the term frequency. The parameter n is the dimension of the vector, and c_i and s_i denote the i[th] dimension of the vector $\vec{c}$ and $\vec{s}$ respectively.

(5) Word overlap
This feature considers how many same words existing in choice sentence and related sentence. This feature is used in our system because we hold the assumption that the more of the same words in the two sentences, the higher similarity between the two sentences and the meaning of the two sentences are closer. The word overlap of two sentences $W(s_1, s_2)$ can be expressed by the following formula (4).

$$W(s_1, s_2) = \frac{|Words(s_1) \cap Words(s_2)|}{|Words(s_1) \cup Words(s_2)|} \quad (4)$$

Where $Words(s_1)$ and $Words(s_2)$ express the set of the words in sentence S_1 and S_2 separately. The numerator and denominator are the intersection and union of two sets respectively.

2.5 SVM classifier
We choose LIBSVM as the classifier and the LIBSVM is a library for support vector classification (SVM) and regression. We use WEKA for SVM classification and the WEKA is an open platform for the data mining work, which brings together a large number of machine learning algorithms that can undertake the task of data mining, including data preprocessing, classification, regression, clustering, association rules and visualization on new interactive interface. After dealing with data set in LIBSVM form, our system chooses the RBF (Radical Basis Function) kernel function to do the cross-validation in WEKA.

We use SVM to solve binary classification problem. The training set are divided into two categories, positive samples and negative samples, among which the correct candidate answer choice constitutes positive sample and marked "1", the false candidate answer choice constitutes negative sample and marked "0".

2.6 Answer generation
Each choice will obtain an accuracy probability by the SVM classifier, and we use the probability as the final score of a choice. We choose the answer choice according to the question tendency. If the question requires finding the wrong choice, that is, the question belongs to the wrong tendency question, we choose the choice with lowest score as the final answer. Otherwise, if the question belongs to the right question, we choose the choice with highest score as the final answer.

3. EXPERIMENTS

3.1 Datasets
This dataset was provided by NTCIR-12 organizers, which was taken from the World History subject.

The questions were selected from the National Center Test for university admissions (multiple choice-type questions) and from secondary exams at multiple universities (complex question including essays). Our system only deals with multiple choice-type questions and this data collection contains 6 sets of training papers (230 questions) and one set of test papers (36 questions), and the question with choosing right or wrong sentence account for about two-thirds of the total, our system mainly solves this type of question. The sample in Example 2 is shown as follows.

Example 2:
*<question anscol="A1" answer_style="multipleChoice" **answer_type="sentence"** id="Q2" knowledge_type="KS" minimal="yes">*
<instruction>From (1)~(4) below, choose the one sentence that contains a mistake in regard to the underlined portion <ref comment="" target="U1">(1)</ref>.</instruction>
<choices anscol="A1" comment="">
<choice ansnum="1">Poison gas was used in trench warfare.</choice>
<choice ansnum="2">Tanks were developed in order to break through the trenches.</choice>
<choice ansnum="3">Aircraft were used for reconnaissance and bombing.</choice>
<choice ansnum="4">Unrestricted submarine warfare was carried out by the United States of America.</choice>
</choices>
</question>

We obtain "answer_type" attribute of the question by parsing the question XML file. Our system only tries to answer the question, which value of "answer_type" attribute is sentence.

3.2 Experimental results

We submitted one formal runs on the phase 3 test data (Center-2011-Main-World History B) for QALab-2 to the NTCIR-12 task organization office and the official evaluation results are listed in the Table 1.

Table 1. The official evaluation result of WUQA system

Run	Center-2011-Main-World History B
Question	36
Correct	6
Total score	17
Correct answer rate	17%

According to Table 1, our system achieved 6 correct answers in a total of 36 questions and performed 17 points in the end-end run of Phase 3, the accuracy is 17%. Because our system only resolved the problem of choosing right or wrong sentence, so Table 2 shows the evaluation result for this type.

Table 2. Evaluation result for particular question

Type of questions	question with choosing right or wrong sentence (without image)
Number	23
Correct	6
Accuracy	26%

The test data contains 36 questions, of which the type of choosing right or wrong sentence has 23. Our system achieved 6 correct answers in 23 questions and the accuracy of the type of choosing right or wrong sentence is 26%.

4. CONCLUSIONS

In this paper, we introduced QALab-2 task at NTCIR-12. We construct the classification model based on SVM to solve the subtask of choosing right or wrong sentence in answer multiple choice questions of National Center Test. Our system has also extracts multiple features, including the similarity between question and choice, the similarity between choice and related sentence, the average similarity between choice and related sentences, the distance between choice and negative sentences, and the word overlap. Our approach achieved 26% accuracy for the type of choosing right or wrong sentence in the Center-2011-Main-World History B.

For this result, we find out that there are many deficiencies for our system, such as in the term extraction, question analysis, and feature extraction. In the future, we will improve these deficiencies in order to obtain higher accuracy. For example, we will introduce more features into the system, such as semantic features and so on. In addition, our system is only for the type of choosing right or wrong sentence, we will study other types of questions in the future.

ACKNOWLEDGMENTS

The work presented in this paper is partially supported by the Major Projects of National Social Science Foundation of China under Grant No. 11&ZD189 and Natural Science Foundation of China under Grant No. 61402341.

REFERENCES

[1] Shibuki, H., Sakamoto, K., Kano, Y., et al. 2014. Overview of NTCIR-11 QA-Lab Task. *Proceedings of the 11th NTCIR Conference.*

[2] Allam, A. M. N., Haggag, M. H. 2012. The Question Answering Systems: A Survey. International Journal of Research and Reviews in Information Sciences.

[3] Mori, T. 2005. Japanese question-answering system using A* search and its improvement. *ACM Trans. Asian Lang. Inf. Process.* 280-304.

[4] Shima, H., Lao, N., Nyberg, E., Mitamura, T. 2008. Complex Cross-lingual Question Answering as Sequential Classification and Multi-Document Summarization Task. *In Proceedings of NTCIR-7 workshop meeting.*

[5] Wang, D., Nyberg, E. 2015. A long short-term memory model for answer sentence selection in question answering. *ACL*.

[6] Dong, L., Wei, F., Zhou, M., et al. 2015. Question answering over freebase with multi-column convolutional neural networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. 260-269.

[7] Toba, H., Ming, Z. Y., Adriani, M., et al. 2014. Discovering high quality answers in community question answering archives using a hierarchy of classifiers. *J. Information Sciences*. 101-115.

[8] Liu, D. R., Chen, Y. H., Huang, C. K. 2014. QA document recommendations for communities of question–answering websites. *J. Knowledge-Based Systems*. 146-160.

[9] Er, N. P., Cicekli, I. 2013. A Factoid Question Answering System Using Answer Pattern Matching. *Proceedings of the 6th International Joint Conference on Natural Language Processing*. 854-858.

[10] Shen, Y., Rong, W., Sun, Z., et al. 2015. Question/Answer Matching for CQA System via Combining Lexical and Sequential Information. *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[11] Yi, L., Wang, J. X., Lan, M. 2015. ECNU: Using Multiple Sources of CQA-based Information for Answer Selection and YES/NO Response Inference. *J. SemEval.* 236.

[12] Nie, L., Wang, M., Zha, Z.-J., et al. 2011. Multimedia answering: Enriching text QA with media information, *ACM International SIGIR Conference (SIGIR)*. 695-704.

[13] Nie, L., Wang, M., Gao, Y., et al. 2013. Beyond text QA: multimedia answer generation by harvesting Web information, IEEE Transactions on Multimedia. 426-441.