

Synthesis of Multiple Answer Evaluation Measures using a Machine Learning Technique for a QA System

Yasuharu MATSUDA

Takashi YUKAWA

Nagaoka University of Technology

1603-1, Kamitomioka-cho, Nagaoka-shi, Niigata 940-2188, Japan

yasuharu@stn.nagaokaut.ac.jp

yukawa@vos.nagaokaut.ac.jp

Abstract

The present paper proposes a new method that synthesizes answer evaluation rules using layered neural networks. A Base Question Answering System that employs a combined conventional method (NUT-BASE system) is implemented and evaluated in the NTCIR-5 workshop Question Answering Challenge 3 (QAC3). Based on the evaluation results, the authors focus on performance improvement for the list task and propose a new method using a neural-network-based machine learning technique for synthesizing answer candidate evaluation measures. There are several measures by which to evaluate the likelihood of the answer candidate, so the system must synthesize these measures in order to determine the answer set. However, the rule for synthesizing the measures in the NUT-BASE system was not effective because it was based on an empirical intuition. Therefore, a performance improvement is expected by the proposed method because it is based on quantitative reason. The experimental evaluation showed that the proposed method achieves a performance improvement, with a value of 0.01 for the mean F-measure.

Keywords: *Question Answering System, List Task, Machine Learning, Layered Neural Network*

1. Introduction

As a participant of the NTCIR-5 workshop Question Answering Challenge (QAC) track, the authors, i.e. the NUT (Nagaoka University of Technology) team, implemented a first-stage question answering system (NUT-BASE). The system applies a vector-space model and a phrase attribute analysis technique (Question Focus; QF)

[2], and is implemented with newly developed QF-based heuristic rules and information retrieval modules using GETA [1]. The evaluation results show that the NUT-BASE system recorded a value of 0.101 for the mean of the modified F-measure (MF1) [3].

As described above, the NUT-BASE system was comprised of conventional methodologies and newly developed heuristic rules. These rules are based on empirical knowledge of Japanese grammar. From the results, several issues are extracted. Among these issues, poor accuracy of an answer candidate evaluation reduces the system performance for the list task significantly. Therefore, the focus of the present paper is the improvement of the answer candidate evaluation.

In the answer candidate evaluation phase of the QA system, there are several measures of likelihood of answer candidates (ACs). The measures include the position of the ACs in the retrieved documents, the relevance of the QF and the ACs, the number of documents relevant to the ACs, and a number of more detailed measures. The importance of these measures varies depending on the interrogative type of the query and presence of the QF.

In the list task, if the difference between the score for a correct answer set and that for an incorrect answer set is remarkable, then the discriminability of correct answers will rise.

Taking this into consideration, a new evaluation measure synthesis method using the machine learning technique with layered neural networks is proposed and implemented. The present paper describes the details of the proposed method and shows the performance improvement compared with a slightly tuned NUT-BASE (NUT-BASE2) system.

2. Development of the Base QA System

First, as a basis for discussion, the base QA system (NUT-BASE) was developed. This system is comprised of the ‘Question Focus’ method [2] and newly developed heuristic rules.

Figure 1 shows an overview of the NUT-BASE system. Generally, in a QA system, the answer set is extracted through four phases as follows:

1. Query Analysis Phase,
2. Document Retrieval Phase,
3. Answer Candidate Extraction Phase,
4. Answer Candidate Evaluation Phase.

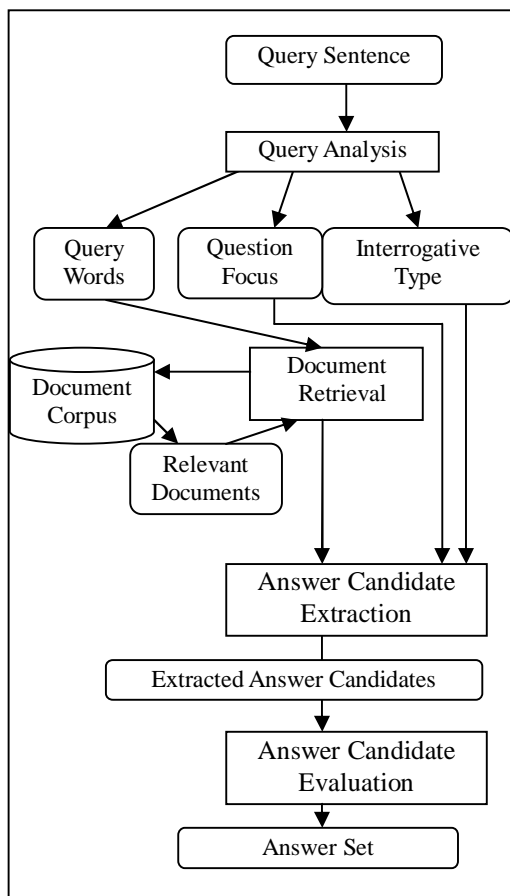


Figure 1. Overview of the base QA system

The NUT-BASE system also follows this general architecture, and so has four modules corresponding to each phase. These modules are described in detail in the following subsections.

2.1 Query Analysis Module

The query analysis module analyzes interrogative types and extracts a QF phrase.

Interrogatives are classified into seven types (what, who, when, where, how, how_many,

what-QF), which can be easily distinguished by the interrogative appearing in the query sentence. The type “what-QF” corresponds to the form of “What QF is it?”.

QF is the phrase that corresponds to the concept containing ACs. However, excessively abstract phrases such as “thing” and “one” are excluded.

The interrogative type and the QF are extracted by pattern matching with regular expressions.

2.2 Document Retrieval Module

The document retrieval module retrieves documents containing ACs exploiting the vector-space model with the TF-IDF algorithm. The modules are implemented using the GETA [1] library.

2.3 Answer Candidate Extraction Module

The processes of the answer candidate extraction module are comprised of three sub-phases. First, words in the retrieved documents are analyzed as morphemes by the ChaSen parser [4]. Second, some of the words are combined into phrases by NEXt [5] and a number of heuristic rules. Finally, the phrases that have attributes corresponding to the interrogative type of the query sentence are extracted (‘who’ and person’s name, ‘when’ and date or time, etc) as ACs.

2.4 Answer Candidate Evaluation Module

The answer candidate evaluation module gives partial scores depending on the evaluation rules as follows:

- Distance between the AC and the index term of the query in the retrieved document.
- Attribute of the AC.
- Whether a QF is included as a suffix of the AC.
- Whether there is a sentence that includes both the AC and the QF and the AC is an instance of the QF, among the corpus.
- Number of retrieved documents that contain the AC.

These partial scores are synthesized with the newly developed heuristic rules into the final score, and the AC set that has higher score is extracted as the result.

2.5 Results of QAC3

Table 1 shows the results of the QAC3 formal run and the reference-1 run for NUT-BASE system.

Table 1. Results of the QAC3 formal run and the reference-1 run for the NUT-BASE system (MF1)

Query Set	Total	First	Rest
Formal Run	0.101	0.129	0.096
Reference-1 Run	0.116	0.129	0.114

The values are the mean of the modified F-measure (MF1) [3]. The values in the ‘Total’ column indicate the results of all questions, and those in the ‘First’ column indicate the results of the first questions of each query series. The values in the ‘Rest’ column indicate the results of the questions of each query series, excluding the first questions.

These are the official records of the NUT-BASE system in QAC3. As the overall results of the formal run, this system ranked tenth among the 16 systems examined.

3. Evaluation Measure Synthesis with Layered Neural Networks

The NUT-BASE system used in QAC3 had a number of problems. Some of which were caused by the heuristic rules. Thus, a number of heuristic rules were modified to improve the performance. This modified base QA system is referred to as NUT-BASE2.

In addition, a new method was proposed to improve the performance for the list task.

3.1 Concept of the Proposed Method

In the NUT-BASE system, the rule for synthesizing the partial scores in the answer candidate evaluation module was not derived quantitatively, but rather by empirical intuition.

In the phases of distinguishing the interrogative types and extracting the QFs, empirically-derived rules are effective, because these phases are grounded on natural language grammar. On the other hand, humans only have empirical intuition for evaluating multiple partial scores. Therefore, an automatic rule construction method is required for this phase.

If the method can derive a rule for synthesizing the partial scores that gives a higher total score to the correct ACs and a lower total score to the incorrect ACs, then the threshold value that distinguishes the correct and incorrect ACs would be determined more easily and an improvement of system performance for the list task can be expected.

Figure 2 shows an overview of the proposed system.

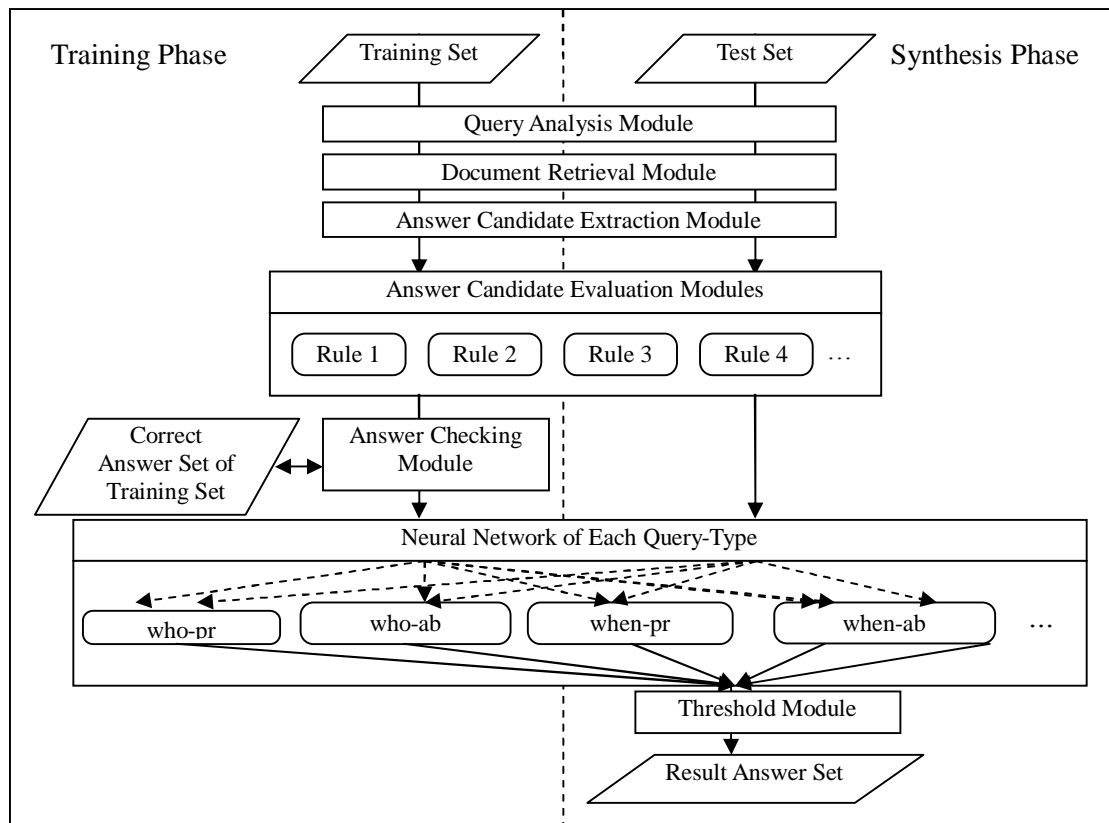


Figure 2. Overview of the proposed system

