Three Systems and One Verifier –
– HOKUM's Participation in QAC3 of NTCIR-5

Yasutomo Kimura
Department of Information
and Management Science
Otaru University of Commerce
3-5-21, Midori, Otaru, Japan
kimura@res.otaru-uc.ac.jp

Kenji Ishida Hirotaka Imaoka Fumito Masui
Department of Information Engineering
Faculty of Engineering
Mie University
Kurimamachiya, Tsu, Japan
{ishida,imaoka,masui}@ai.info.mie-u.ac.jp

Marcin Skowron Rafal Rzepka Kenji Araki
Graduate School of Information Science and Technology
Hokkaido University
Kita-ku Kita 14 Nishi 9, Sapporo-shi, Japan
{ms,kabura,araki}@media.eng.hokudai.ac.jp

Abstract

This paper is a report from collective participation in NTCIR-5 Question Answering Challenge between researchers from Mie University, Hokkaido University and Otaru University of Commerce. Although our results were not impressive, we would like to share our experiences with everyone who think about participating in the challenge but is afraid of his or her lack of experience in the field. Understanding the problems of QA from the practical side was very instructive and gave us a stronger base for future trials. We briefly introduce our preparations and participation then conclude with analysis what can be simply done with freely available tools.

Keywords: *NTCIR, Question Answering Challenge, hybrid system.*

1 Introduction

Very large data sets opened new frontiers for Question Answering (QA) field. Statistical methods used to retrieve knowledge from such sets proved that there are automatic methods for achieving desired information. But QA goes beyond common searching and brings the end-user an answer instead of documents which have to searched again, this time physically, which is really troublesome if the size of a document is large. But by using a QA system, we can easily notice that effectiveness depends on the data, even if the domain is open. This makes the evaluation of QA systems difficult and the ideal situation is to check their effectiveness on the same data set and in the possibly shortest time what should eliminate laborious and expensive tuning efforts where rich companies would have advantages over university laboratories, for instance.

2 Basic Idea

The most famous QA effectiveness competition mentioned in previous section is famous American TREC[2]. Its Japanese equivalent is called NTCIR[7] and most of our group decided to participate in its QAC[1] task for the first time, though only one debutante had a QA background. Otaru University of Commerce and Hokkaido University Teams decided to join the QAC frequenter - Mie University Team and HOKUM Group was born. Our basic idea was to probe first-time participants' ideas while having one fix system which could be a safety valve if things went wrong. Therefore main part of newly built hybrid system was created form Mie's MAIQA, Otaru's Baseline and Hokudai's Baseline Plus. Three subsystems output was to be filtered by Web-Based Verifier (also created in Hokudai) in one version of the system and simply cleaned up with majority decision in another (only the answers repeating themselves in different subsystems were supposed to be selected).

3 Baseline System

Our baseline system was developed as the departure point for experimenting with our ideas. It helped members without QA experience to get their ideas involved in the project. Most of it was based on the

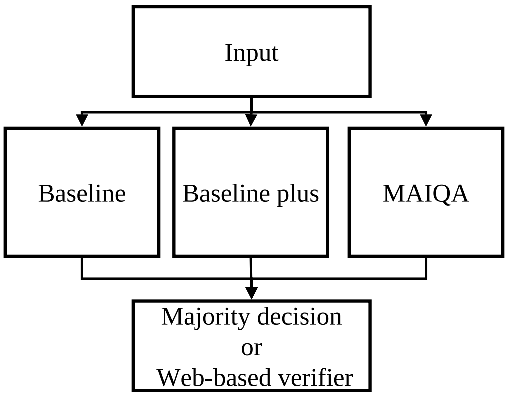

Figure 1. The subsystems of HOKUM Hybrid System

baseline presented in [4] though for this task similarity measure using tf-AoI was not used. We decided to use Named Entities and SVM this time. Processing steps of this systems are as following.

Input Sentence Processing

First part of processing is to recognize if the input is a related question and after identifying the ID it decides question which could be used. The type of answer is decided by SVM[10]. We used 1218 questions to make system learn the answer types, which were as follows: H: Humans, L: Locations, O: Organizations, N-umerical:(p: periods, d: dates, t: time, m: money, a: percentages, h: number of people, x: other), E: Entity, X: Others.

Searching

We used Namazu[5] for our default search engine which we also used for indexing Mainichi and Yomiuri newspapers for years 2000-2001. We were retrieving answers from at least 1 and no more than 100 results.

Query Formation

In the next part, a query is formed. The algorithm works as follow.

- Retrieving nouns from question sentence with related ID;
- Calculating amount of information of every word;
- Choosing the word combinations;
- Calculating total amount of information for prepared combinations;
- Putting them in descent order;
- Trying out queries according to the list - if at least one searching result exists it is treaded as candidate consisting an answer.

3.1 Answer Candidates

The answer type decided in Input Sentence Processing which was most frequent in the retrieved document is chosen for an answer candidate. Top five candidates become the baseline system's answer.

4 Baseline System Plus

Series of tests and system optimizations performed on a data from the previous NTCIR evaluations, aiming to find the error-prone modules of the baseline system were used. Processing was the same as in above described baseline system but there were differences described below.

Question Classification

Based on our experience with English Question Classification[8], we decided to use the most probable question classification information provided by SVM-based classifier, even in cases were the classification score was lower than 0.

Query Formation

We assumed that in case of task that requires to answer to a list of related questions it is important to preserve and use information contained in previous questions, related to the same topic, person, event etc. Such information relatively easily extractable from a question is a question subject.

Question Subject Words

Question subject word was extracted using a simple regular expression applied to a part-of-speech tagged question: NOUN (*suru-no*—*shita-no*)*?wa* (do/did). In the query formation phase, words that appeared in a question were selected and ordered based on their frequency of occurrences obtained from the corpora used in the previous NTCIR QAC evaluations. If an initial query did not retrieve any documents, a query formation module was gradually removing the most general words, aiming to preserve these that could be used as keywords to retrieve an answer-rich set of documents.

Lowering Frequency Scores

Considering the importance of a question subject word, we decided to ”promote” such words in a query formation stage, by artificially lowering their frequency score. Since frequently question words available in a current question were not sufficient to form a reliable query that could retrieve an answer rich set of documents (additional, important information existed in a previous question), we decided to use the subject word extracted from the previous question, as an additional keyword used to form a query. The frequency score for such words was also lowered.

Query Formation (Information contained in a longer expression)

In a baseline system longer expression (words consisting of a few Kanji character or a combination of Kanji characters and numbers) were excluded if a query was not retrieving a sufficient number of documents, and there were no other words that had a higher frequency of occurrences. Selected parts of longer expressions (dates, proper nouns etc.) often possessed important information usable to form a reliable query, which could not be used if the whole word/expression was removed from the list of words, used to form queries. To preserve at least partial information contained in such words/expressions the query formation module was splitting expressions longer than 3 Kanji characters on two words; similarly if applicable dates were divided on year, month and day.

5 MAIQA System

Parallelly with preparations in Hokkaido, the Mie University members were getting ready for their next challenge advising the debutantes from the North in the same time. MAIQA system[3] which is developed there basically consists from three parts: Question Analysis Module, Text Searching Module and Answer Choosing Module. After a question sentence is inputted, the Question Analysis Module analysis it. Next it decides the answer type and words of the biggest importance. These words create keywords which are passed to the Text Searching Module where related documents are retrieved. In the last part of processing system finds inside the documents candidates which fit the answer type and creates ranking where the highest candidates become answers. More specified description of the methods can be found in “Results” section.

6 Web-based Verifier

This engine was to check if the answer candidates from every system really answers one of three most popular type of questions - *where*, *who* and *when*. When one of these questions was discovered, the verifier was checking the frequencies of answers with expressions chosen for particular questions. If the question was *who* and the answer was a place, the string like “place was talking” was sent to Yahoo Japan[11] and the hit number was remembered. Every of the three question types was checked with six “answer noun + particle characteristic to the combination + verb characteristic to the type of answer” strings (See Tab.1, Tab.2 and Tab.3).

Table 1. WWW Query strings for “who” ("P" stands for Particle)

dare (who)	English
+ga+ita	Linking-P+was
+ga+ite	Linking-P+being
+ga+hanashita	Linking-P+talked
+ga+hanashite	Linking-P+talking
+ga+itta	Linking-P+said
+ga+itte	Linking-P+saying

6.1 Algorithm

The calculation for filtering was made as follow: if overall hit number was lower than five and if 10% of the highest hit number was higher than the sum of the rest of hits[1] then the answer was deleted from the answer candidates list. Proposed method works in most cases but still may have problems with questions where answer type is not clear. There are w-questions like “where” asking about organizations. In such cases verb “to live” can (but does not have to) spoil the calculations. The authors are thinking about adding verb from the question to the existing sets in the future.

6.2 Filtering Example

To illustrate the verifying process let us bring an example of question number QAC3-30003-01: Hamasaki Ayumi-no seinengappi wa itsu desuka (When was Ayumi Hamasaki born). An overall answer created from all three system answers is brought to the verifier because one of three Wh-questions was recognized (if not, the majority decides the final answers set). Every answer is agglutinated with all of expressions for When-questions (see Tab. 3). Because names of singers (Utada and Masaharu Fukuyama) which appeared among answers candidates do not appear in Yahoo, they are eliminated from the list for a final answers set.

[1] It was for avoiding approving cases where an incorrect could be highly counted in one of the cases like *mura-ni-umareta* (born in a village) but are low in the rest of cases *mura-goro* (around village, which “around” is used in time expressions)

Table 2. WWW Query strings for “where” ("P" stands for Particle)

doko (where)	English
+de+okonawareru	Place-P+take+place
+de+okonawareta	Place-P+took+place
+ni+sunde	Direction-P+live
+ni+sunda	Direction-P+lived
+de+hataraita	Place-P+worked
+de+hataraite	Place-P+working

Table 3. WWW Query strings for “when” ("P" stands for Particle)

itsu (when)	English
+goro	around
+ni+okonawareta	Direction-P+was held
+ni+okonawarete	Direction-P+being held
+ni+okonawareru	Direction-P+be held
+no+aida	Possesive-P-period
+ni+umareta	Possesive-P-was+born

7 Results

The answers were produced by each system (Baseline, Baseline Plus and MAIQA) in every stage and the final answer(s) were decided from their answers by majority decision or by Web Verifier which deleted answers that seemed not to answer questions (only in case of WHO, WHERE and WHEN)). The results in both runs are presented in Fig. 2 and Fig. 3.

- **HOKUM-1** symbolizes answer set filtered by web-based verifier
- **HOKUM-2** is majority decision set
- **HOKUM-3** is later version of HOKUM-1
- **HOKUM-4** is later version of HOKUM-2

7.1 Comparison of Subsystems

We initially compared the efficiency of three subsystems scoring them with QAC-1 Task-1 questions and correct answer sets. As was expected, the Baseline Plus showed remarkable improvement over the Baseline system (see Tab. 4 and Fig. 2).

Table 4. Initial comparison of Baseline and Baseline Plus subsystems

	Quest.	Ans.	Output	Correct	MRR
B	80	94	220	11	0.103
B+	80	94	270	20	0.146

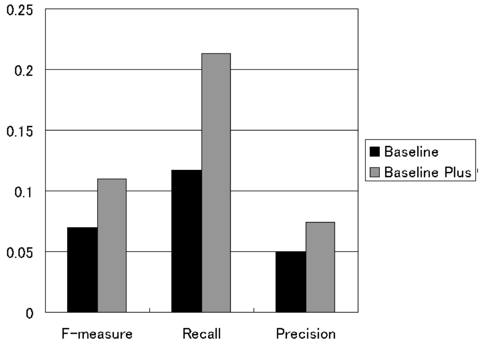

Figure 2. Scoring results for Baseline and Baseline Plus subsystems

But during the current comparison problems with using evaluation programs led to chaos in multiplied data and we failed to discover where was the biggest reason for lower results but most probably overall performance of all parts prepared by debutantes was lower than the experienced team. Although, the final results show that Web-Based Verifier seems to filter out quite a number of erroneous answers in Formal Run increasing the accuracy significantly for about 30%. There is also big possibility that web-based verifier and majority decision spoiled particular units especially MAIQA system which results are introduced below.

7.2 Results for Different Prototypes of MAIQA

a Overall comment:
S-rank answers became 0 (this is why Fig. 4 and Fig. 5 show only A-rank and B-rank answers) because the correct answer was not retrieved as a

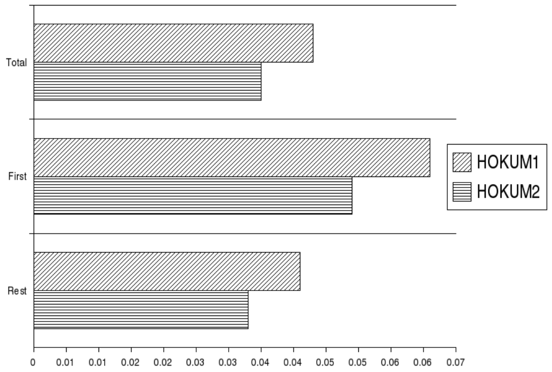

Figure 3. Results of HOKUM in Reference Run

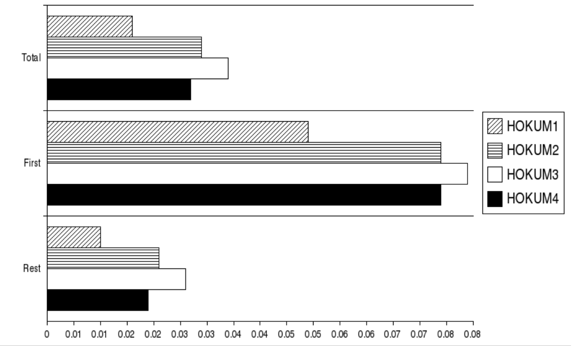

Figure 4. Results of HOKUM in Formal Run

named entity. For example a question about how much is a trip around a world, the price, which is a correct answer, could not be ranked “S” by NExT[6] Name Entity Tool used with MAIQA. The main reason why S-rank answers were not created is because of low number named entities which could created them. Probably this was influenced by big number of combined named entities which could not be correctly analyzed by NExT recognizer.

b Reference Run (RRun1) comparison:
Comparing MAIQA system output (RRun1) with results which were narrowed by WWW-based frequency check proved that the latter (RRun1 meaning MAIQA system in RRun1) had higher accuracy which shows that process of eliminating wrong answers was successful. However the number of correct answers decreases in the case of RRun1 which suggests that many of them was also eliminated. This is because there were many cross-related nouns which belonged also to other entities like names of people and names of places (as *Washinton* - “Washington”).

c Formal Run (R2) comparison:
Comparing to the RRun1, accuracy and coverage decrease drastically which shows how difficult contextual questioning is.

d FRun_A and FRun_B prototypes comparison:
Both trials had different methods for articles searching. In the first version of this prototype (called FRun_A) the first question (rootQ) was used for preparing a set of keywords which were used for forming a query retrieving an article. Next questions were producing new keywords but the searching process was done within the article retrieved by rootQ. For example (keywords in brackets):

QAC3-30009-01,How much yen 1 Euro costs? (1,Euro,How)
QAC3-30009-02,What is the symbol on the coins? (coins,symbol)
QAC3-30009-03,How many countries participate in it?(participate,it,countries,how)

then:

QAC3-30009-01 creates query “1 Euro How” and keeps the found article(s).
QAC3-30009-02 creates query “coins symbol” and use them for searching found article(s).
QAC3-30009-02 creates query “participate it countries how” and use them for searching found article(s).

In prototype called FRun_B the keywords were joined in succession after every question to form the query:

QAC3-30009-01 creates query “1 Euro How” and performs search
QAC3-30009-02 creates query “1 Euro How coins symbol” and performs search
QAC3-30009-02 creates query “1 Euro How coins symbol participate it countries how” and performs search.

In the latter method there were many cases when no article was found but MAIQA was using POS-based weighting to get rid of the keywords of the least importance and perform the search again without it.

e About FRun_A
There was many correct answers eliminated in the latest version (prototype FRun_A_Web) which happens probably due to lack of keywords (if there are two or less keywords, all the answers were deleted).

f Why prototype FRun_B produced better results than prototype FRun_A?
Probably the reason is in the lack of correspondence transition between contextual sentences. In case of FRun_A it was not possible to keep the correspondence between questions because the article found by rootQ was not enough when the topic changed. The final results for both runs of all MAIQA prototypes are introduced in Figures 3 and 4.

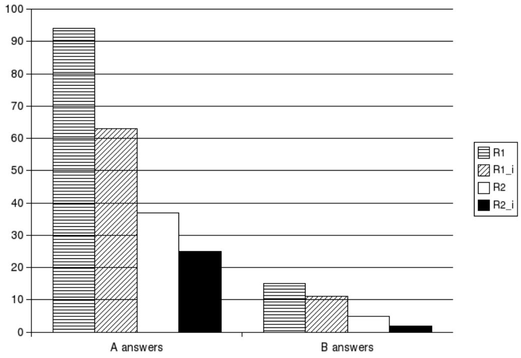

Figure 5. Results of MAIQA in Reference Run: R1 = RRun1, R1_i = RRun1_Web, R2 = RRun2, R2_i = RRun2_Web

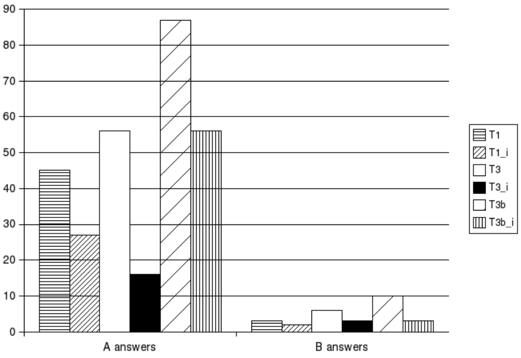

Figure 6. Results of MAIQA in Formal Run: T3 = FRun_A, T3_i = FRun_A_Web, T3b = FRun_B, T3b_i = FRun_B_Web, T1 = FRun, T1_i = FRun_Web

8 Discussion and Future Work

It was extremely important team and project experience. We discovered the problems of competition-style workshops, own lack of organization and found how powerful and fruitful a brain-storming discussion could be. These of us who had no experience in QA understood the difficulty of this task and painfully felt their lack experience which hopefully will be connected to the next year competition. Briefly pointing out the things we missed and want to work on:
a) Our lack of experience and not full understanding of the challenge system led to the lack of results and inability to fully evaluate the proposed methods in every stage.
b) We noticed the importance of language dependent features for the QA system based on relatively small corpora, compared to the statistical methods successfully applicable in systems based on large corpora (e.g. exploiting the redundancy of WWW).
c) An application of more sophisticated methods should work better:

- Question Classification - richer taxonomy, more training examples, feature space creation extending a Bag-Of-Words approach used in current system.
- Query Formation - application of the Query Generation Patterns[9] method, to obtained a set of generation rules that transform a given question to a query that retrieves an optimal set of answer-rich documents.
- Answer Extraction - extending the Named Entity Recognizer, with a list of regular expressions usable to extract various numeric expressions (weigh, distance, amount of money, speed, temperature etc.), and lists containing instances of possible answers for a specific type of question (list of countries, cities, provinces, lakes, film titles etc.).
- Application of the Internet-based QA-system to provide set of answer candidates to a corpora-based QA system, for a further verification and localization of related documents.

d) It was also noticed that in cases where the AoI (Amount of Information) difference was used without extracting named entities, accuracy was bad. Analyzing words (nouns) that appeared in documents with answer candidates, we discovered that the method where answers were chosen from documents containing words with a big difference between “amount of information calculated from the frequency of words appearing in whole set of documents” and “amount of information calculated from the frequency of words appearing in documents with answer candidates" did not show improvement. Probably language depended information might improve this part and should be considered for testing in the future challenges.

The Question Answering Challenge 2005 (QAC3) made us think also about the real-world question answering applications. What environment such system works? Is it a good idea to use a keyboard? If sound recognition is involved – is it better to approach the problem from the point of view of using a robot as a question answering body? We would like to initiate a discussion about environments in QA, how they might influence challenges its evaluation and also evaluating everyday life QA applications.

References

[1] J. Fukumoto, T. Kato, and F. Masui. Question answering challenge (qac-1): An evaluation of question answering tasks at the ntcir workshop 3. In *AAAI Spring Symposium: New Directions in Question Answering*, pages 122–133. AAAI, February 2003.

[2] D. Harman. Overview of the second text retrieval conference. The Second Text Retrieval Conference (TREC-2), Gaithersburg, MD, Special Publication 500-215, 1994.

[3] N. Hidaka, F. Masui, and K. Toaski. Question answering system based on expanded answer types and multi-scores. In *Proceedings of the Fourth NTCIR Workshop on Research in Information Access TechnologiesInformation Retrieval, Question Answering and Summarization*, April 2004.

[4] Y. Kimura, K. Tochinai, and K. Araki. Evaluation of japanese dialogue processing method based on similarity measure using tf - aoi. In *Proceedings of 5th International Conference, CICLing 2004, Lecture Notes in Computer Science, vol 2945*, pages 371–382. Springer-Verlag, February 2004.

[5] Namazu. http://www.namazu.org/.

[6] NExT. http://www.ai.info.mie-u.ac.jp/ next/next.html.

[7] NTCIR. http://research.nii.ac.jp/ntcir/.

[8] M. Skowron and K. Araki. Effectiveness of combined features for machine learning based question classification. *Special Issue of the Journal of the Natural Language Processing Society Japan on Question Answering and Automatic Summarization*, 12, November 2005. (In Printing).

[9] M. Skowron and K. Araki. Learning the query generation patterns. In *Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science*, pages 620–624. Springer-Verlag, February 2005.

[10] V. Vapnik and C. Cortes. Support-vector networks. *Machine Learning*, 20, September 1995.

[11] Yahoo. http://www.yahoo.co.jp/.