NUL System at QA Lab-2 Task

Mio Kobayashi
Nihon Unisys, Ltd.
mio.kobayashi@unisys.co.jp

Hiroshi Miyashita
Nihon Unisys, Ltd.
hiroshi.miyashita@unisys.co.jp

Ai Ishii
Nihon Unisys, Ltd.
ai.ishii@unisys.co.jp

Chikara Hoshino
Nihon Unisys, Ltd.
chikara.hoshino@unisys.co.jp

ABSTRACT

This paper describes our strategies and the methods of NUL team on NTCIR-12 QA Lab-2 Japanese National Center Test tasks. We mainly use three strategies with four solvers. First strategy, we use Pointwise Mutual Information (PMI) and search results ranking to calculate the score of choices. Second, we convert True-or-False question to virtual factoid question by removing named entity. Third, we convert textbooks and questions to syntax tree and match them. We choose the final answer by aggregating ranks of each solver. Our system achieved 76 points in Benesse mock exam Jun 2015 (Pattern 1) of Phase 2.

Team Name

NUL

Subtasks

Japanese

Keywords

QA Lab-2, question answering, syntax tree matching, word co-occurrence, search results ranking

1. INTRODUCTION

The National Institute of Informatics (NII) launch a project called "Todai Robot Project" to develop a computer program to solve the college entrance examination problem in order to re-unify the field of artificial intelligence that has been subdivided [1]. In 2015, we took a mock exam by Todai Robot Project "World History B" section, which is Phase 2 task of QA Lab-2, and we scored 76 points (deviation value of 66.5). This score is 30 points higher than average human score.

Knowledge resources like textbook and Wikipedia contain correct information which is described as natural language sentences. To choose correct answer of world history questions, we need to match the semi-structured information.

Some of world history questions on National Center Test need to understand photo, map or illustration. Except for these questions, world history questions are classified into 4 types: "1. True-or-False", "2. Slot-filling", "3. Factoid", "4. Time Reordering".

This paper describes our system to solve questions of those types that was used in Phase 3 of QA Lab-2 [2].

2. PREVIOUS STUDIES

From previous studies, there are mainly 3 strategies for type 1 questions as following:

1. Solve questions by keyword distributions [3]
2. Convert questions into factoid questions [4] [5]
3. Convert questions into textual inference [6]

Strategy 1 has a high coverage rate but is not so strict. Strategy 2 is good at detecting wrong choice, but cannot convert every question into factoid. Strategy 3 is strict but has a low coverage rate. We implement solvers for each strategy and combine their results to complement with each other.

3. OBSERVATION OF TASK

"1. True-or-False" is most popular type in the National Center Test. 70 percent of questions are classified into this type. We observe questions of this type to implement solvers. Observation targets were World history exam of National Center Test (2007, 2009) and the mock exams of Benesse Corporation (2014 Sep, 2015 Jun). As knowledge resources, we used 4 sets of high school textbook which were provided and Wikipedia.

From observation, we built three hypotheses as follows. First, it is not necessary to read knowledge resources widely across sections or chapters, since there is sufficient information in a local portion, such as a paragraph or a sentence. Second, questions contain more abstract description of time than knowledge resources (e.g., "1453" to "15th Century", "1945" to "1940s"). Third, questions have more abstract description of location than knowledge resources (e.g., "France" to "Europe", "Japan" to "Asia"). We investigated on true choices of questions whether: (1) all of the named entities in choices appear in a single paragraph of knowledge resources, and (2) choices have a more abstract description of time and location than knowledge resources.

Table 1 shows that all of the named entities of true choices are included in a single paragraph of knowledge resources in almost cases. Therefore, we can get an optimal paragraph from knowledge resources which contains sufficient information to choose correct choice.

Table 2 shows that considering an abstract description of time and location is important.

4. RESOURCES AND COMMON MODULES

Based on above analysis, we make following dictionaries and modules.

Table 1: Rate of choice which has clustered named entities

Knowledge Resource	Count	Rate (%)
Textbook	123/135	94.1%
Wikipedia	128/135	94.8%
Textbook + Wikipedia	134/135	99.3%

Table 2: Rate of choice which has more abstract description

Type of description	Count	Rate (%)
time	39/135	28.9%
location	16/135	11.9%

4.1 Knowledge Resources

We use following knowledge resources.

1. 4 sets of high school textbook
2. Wikipedia
3. World History Ontology[1][7]
4. Web site of world history[2]

4.2 Dictionary

We make following dictionaries.

1. Named entity dictionary
2. Synonym dictionary
3. Hypernym-hyponym dictionary
4. Antonym dictionary
5. Suffix dictionary
6. Year conversion dictionary (Nations and historical events to year) made from World Historical Atlas[3]

Named entity dictionary contains class of words (time, person, etc.) information . This dictionary contains approximately 45,000 entries. We make synonym dictionary based on Wikipedia redirect and hypernym-hyponym dictionary from output of Wikipedia Hyponymy extraction tool[4]. Additionally, we also use WordNet[5], Nihongo Goi-Taikei[6]. We also make suffix dictionary for grouping words that have the same suffix using frequency of suffixes in the whole documents [5].

4.3 Module of Time Expressions

We extract time expressions from questions and choices and normalize them by using normalizeNumexp[7]. Additionally, we develop the module for deciding inclusion between two time expressions.

4.4 Matching of Words

In word matching process, suffixes are ignored (e.g., “Japanese (日本人)” to “Japan (日本)”). If a word in question is synonym or hypernym of another word in knowledge resources, solver determines these two words matched.

Context: “... Moreover, it was unnatural for many (5)farmers that one year begins in the fall, which is a busy farming season. ...”

Instruction: “In relation to the underlined portion (5), from 1-4 below, choose the one sentence that is incorrect with regard to the following sentences that describe the agriculture and farmers of 19-20 century.”

Choices:
1. In Prussia, farmers released (Serfs released) by the reform of the Stein-Hardenberg et al. had been carried out.
2. ...

Figure 1: Question 24 in 2011 data set

When a word does not match other word and those words have same specified class (e.g., ”America” does not match ”France” and these words have same ”Nation” class.), we define these two words hold exclusive relation.

4.5 Extracting NE from Question

Since there are many choices that lack named entities (NEs) which are key information to answer True-or-False question, it is necessary to extract NEs from instructions of a question. Thus we tried to find them by using regular expressions and word class information.

- Extract NEs from underlined portion in the context if the instruction do not include “In relation to (〜に関して)”.
- Extract **location** and **time** class NEs from the instructions if the choice lacks these class words.
- Extract **person**, **personType_nationality**, **organization** and **nation** class NEs from the instruction if subject of the choice is omitted.
- Abstract NEs (e.g, **phenomenon social**, **activity lawAndEconomics**, **personType socialRole**) are excluded.

Figure 1 (Question No.24 in 2011) shows an example. In this question, ”farmers” are underlined but ”In relation to” is contained in instruction, thus we ignore it. Moreover, ”19-20 century” is ”Time” class NE and NE of this class is lacked in No1 choice, thus we extract ”19-20 century” and complement the choice by that NE.

5. SYSTEM ARCHITECTURE

We developed four solvers for each of the three strategies as described in Chapter 2.

5.1 Solver 1

Solver 1 mainly uses words co-occurrence. It is assumed that co-occurrence between words in correct choices is high, in contrast with wrong choices. Based on this assumption, we applied Pointwise Mutual Information (PMI) [8] values between two words to score True-or-False questions.

[1] http://researchmap.jp/zoeai/event-ontology-EVT/
[2] http://www.y-history.net/
[3] http://x768.com/w/twha.ja
[4] https://alaginrc.nict.go.jp/hyponymy/
[5] http://nlpwww.nict.go.jp/wn-ja/
[6] http://www.iwanami.co.jp/hotnews/GoiTaikei/
[7] https://github.com/nullnull/normalizeNumexp

However, there are the cases that combinations of three or more words are important to solve that. Thus, we introduce search rank in order to consider the relation between an NE and the other words in choices. We used Apache Solr[8] as the base search engine and made search index by dividing textbook and Wikipedia into a sentence unit. In order to emphasize the coverage of the words in the query, we improved the scoring of search engine.

It is noted that this solver is robust to the error of NE dictionary, because this solver does not use class information of NE.

5.1.1 Score of PMI

First, we extract NEs and content words (CWs) from the choice and make pairs of two words between an NE and next NE, considering that the choice includes multiple topics. Here, we consider the pair that at least one word in the pair is NE. We also consider the CW of the pair has antonyms that are extracted from example phrases of the event in relationship of anti-meaning of the World History Ontology. Next, we calculate $PMIScore$ with average PMI of all pair S.

$PMIScore$ is defined as:

$$PMIScore = \frac{1}{|S|} \sum_{(w_i, w_j) \in S} \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

where, $|S|$ is number of S, $p(w_i, w_j)$ is the joint probability that w_i and w_j occur together in the search index and the term $p(w_i)p(w_j)$ is the probability that w_i and w_j would occur together if they were statistically independent. Let $hits(query)$ be the number of hits (the number of documents retrieved) when the query is given to Solr, $query(w_i)$ be the query expand the w_i with synonyms, and N be total number of sentence units in a search index. $p(w_i, w_j)$ is calculated as follows:

$$p(w_i, w_j) = \frac{hits(query(w_i)\ AND\ query(w_j)) + \epsilon}{N}.$$

5.1.2 Score of Search Rank

First, we make pairs of an NE ne_i and query q_i by deleting string of ne_i from the choice. Next, we get rank $Rank(ne_i, q_i)$ of sentence includes ne_i in search results by q_i. We calculate $RankScore$ as average $Rank(ne_i, q_i)$ of all pairs Q. $RankScore_q$ is defined as:

$$RankScore = \frac{1}{|Q|} \sum_{(ne_i, q_i) \in Q} -Rank(ne_i, q_i).$$

Let k ($k = 10$) be a threshold of number of search results, $Rank(ne_i, q_i)$ as follows:

$$Rank(ne_i, q_i) = \begin{cases} rank_i & rank_i < k \\ 2 * k & otherwise \end{cases}$$

where, if ne_i is added from the instruction (see 4.5), we reduce the score by half of $Rank(ne_i, q_i)$, considering the failed possibility of extracting the word from the instruction.

If time expression $time_i$ is included in the choice, first, we get list of time expressions $list_{time_i}$ up to 20 from top 30 search results by q_i as the query. Next, we get rank of $time_i$ $rank(time_i)$ in the list and calculate $TimeScore$ in the range of -10.0 to 10.0, as follows:

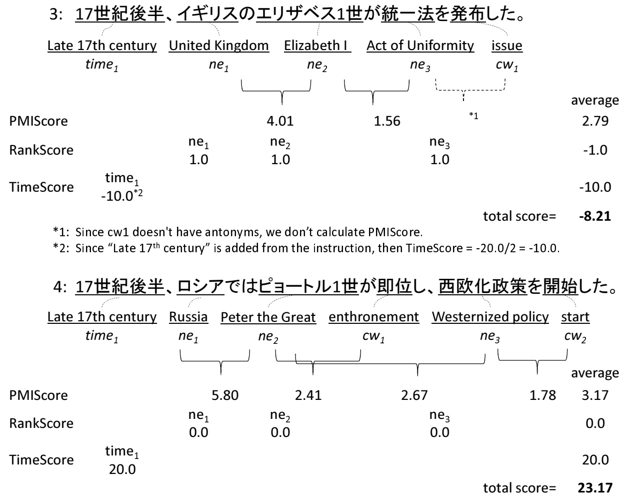
$$TimeScore_{time_i} = 10.0 - 20.0 \times \frac{rank(time_i)}{length(list_{time_i})}$$

Here, we add some rules, such as:

- If $time_i$ is not found in $list_{time_i}$, we reduce 10.0 from $TimeScore$.
- If sentences include both $time_i$ and all NEs are found in search results, we add 10.0 to $TimeScore$.
- If sentences include both some time expressions and all NEs are found in search results, however, these time expressions are not in range of $time_i$, we reduce 100.0 from $TimeScore$.
- If $time_i$ is added from the instruction, we reduce $TimeScore$ by half.

5.1.3 Judgment for True-or-False Question

We sum up $PMIScore$, $RankScore$ and $TimeScore$ as the score of a true/false choice, and choose the one that has the highest score as true from given choices. Figure 2 presents the example scores of false choice (no. 3) and true choice (no. 4). The score of no. 4 is highest, then we choose no. 4 as true.



*1: Since cw1 doesn't have antonyms, we don't calculate PMIScore.
*2: Since "Late 17th century" is added from the instruction, then TimeScore = -20.0/2 = -10.0.

Figure 2: Q41 in Jun 2015 Benesse mock exam

5.1.4 Judgment for Other Question

For Slot-Filling and Factoid question we sum up $RankScore$ and $TimeScore$, and if the words of the choice are two or more, we add $PMIScore$. For True-or-False questions with a wavy line, we also sum up the word of $RankScore$ and $TimeScore$. For Unique Image, Mixed and Other Questions, we only use $PMIScore$ of words that are extracted from annotations and the instruction.

5.2 Solver 2

In many training cases, we observed that the false choice is made by flipping one named entity in the true choice. For example, the false choice, "Charlemagne defeats the Magyar at

[8] http://lucene.apache.org/solr/

the 8th century." is made by the true choice "Charlemagne defeats the Avars at the 8th century.". In this example, the false choice is made by the flipping named entity "the Avars" to "the Magyar". Therefore, to detect such conversions, we made the solver which transforms the choice to virtual factoid questions by hiding each named entity in order. If the answer of the factoid question is different from the hidden word, we increase the fallacy of the choice.

5.2.1 Scoring of Answer

In factoid question answering systems, we answer the single word under given knowledge resources of texts D and the question Q. First, we tokenize the texts and the question to $D = d_1, d_2, \ldots, d_n$ and $Q = q_1, q_2, \ldots, q_d$ respectively. We consider the conditional probability of the answer A given D and Q by

$$p(A|Q, D) \equiv p(a|q_1, \ldots, q_d, d_1, \ldots, d_n).$$

If we assume the prior probability of the answer $p(a)$ is uniform and using the Bayes theorem. We get

$$p(a|q_1, \ldots, q_d) \propto p(q_1, \ldots, q_d|a, d_1, \ldots, d_n).$$

Therefore, we can transform the question to finding the word a which maximize the conditional probability of the question Q given a.

$$\arg\max_a p(q_1, \ldots, q_d|a, d_1, \ldots, d_n).$$

In the case that we can determine the answer by examining texts D locally, we should choose the range of the locality. In general, the definition of the locality is given by the sentence or the word window which are the fixed number of pre and post words of the candidate answer. However, we observe that the locality of questions in texts has a large variation. Therefore, we directly define the distance as

$$p(q_1, \ldots, q_d|a = d_k, d_1, \ldots, d_n) \propto \sum_{i=1}^{d} \alpha_{q_i, d_k} \exp(-\gamma l(q_i, d_k)^\beta)$$

where, the answer candidate a is given from the set $d_1, \ldots, d_n$ and this formula gives the kth word score. Moreover, $l(q_i, d_k)$ is the nearest neighbor distance from d_j to d_k which d_j matches q_i ($\min_{d_j = q_i} |k-j|$) and α, β, γ are hyper-parameters. Finally, our system returns the highest score with the word or the word within a given class such as **time** or **person**.

5.2.2 Judgment for True-or-False Question

We define the cost of the each factoid question as a difference from the highest scored word to the hidden expected word under the constraint of the same class of the word. For the previous example, the converted factoid question is "Charlemagne defeats **personType_nationality** at the 8th century" where **personType_nationality** is the class of the word. The solver answers "Avars" 1st place with score 3.2 and "Magyar" 5th place with score 1.1. Then, the cost of the factoid question is $2.1 (= 3.2 - 1.1)$. Additionally, we define the cost of choice as a mean of the each cost of factoid question. Finally, we answer the lowest cost choice as true from given choices.

5.3 Solver 3

It is considered that if an abstract expression for a choice is similar to an abstract expression in knowledge resources, the choice is correct. From this assumption, we convert sentences in both choices and knowledge resources to abstract expressions called "Syntax Tree", subsequently, evaluate the similarity between these abstract expressions and utilize the similarity to True-or-False questions.

5.3.1 Definition of Syntax Tree

In this paper, we define "Syntax Tree" as a tree which root is a predicate and each word in "Syntax Tree" has a semantic role. Semantic role types are pred (predicate), sbj (subject), obj (object), time, loc (location), loc-to (location to) and other.

We create sequence of Syntax Trees from predicate argument structures which are output of KNP[9] by applying simple rules.

In addition, we complement omitted words by applying following rules.

Rule-1)
If word of sbj role is omitted, complement it by word of sbj role in the just before Syntax Tree.

Rule-2)
If word of time role is omitted, complement it by word of time role in the just before Syntax Tree.

Rule-3)
If word of time role can not be complemented by Rule 2, complement it by a word of **time** class before Syntax Tree.

Rule-4)
If word of location role is omitted, complement it by word of location role in the just before Syntax Tree.

In addition, we resolve references by applying following rules.

Rule-1)
If a reference is "Demonstrative+Suffix" type like "this+trial (この+裁判)", resolve it by searching word which has the same suffix (trial).

Rule-2)
If a reference is a singular personal pronoun like "he (彼)", resolve it by searching word of which class is **person**.

Rule-3)
If a reference is a plural personal pronoun like "they (彼ら)", resolve it by searching word of which class is **personType_socialRole** or **personType_nationality** or **personType_other**.

Apart from that, we convert a passive sentence into a declarative sentence.

5.3.2 Scores

We define following scores. In this paper, let T_h be the Syntax Tree of the choice and T_t be the Syntax Tree of the sentence in knowledge resources.

Syntax Tree Similarity Score

Syntax Tree Similarity Score is defined by the following formula.

$$SimScore(T_h, T_t)$$
$$= f_m(T_h^{pred}, T_t^{pred}) * \max_{r' \in R'} f_m(T_h^{r'}, T_t^{r'}) * \frac{1}{|R|} \sum_{r \in R} f_m(T_h^r, T_t^r)$$

where T_h^{role} is set of role words in Syntax Tree T_h, $R = \{sbj, obj, time, loc, loc-to, other\}$ and $R' = \{sbj, obj\}$. The

[9] http://nlp.ist.i.kyoto-u.ac.jp/?KNP

definition of $f_m(T_h^r, T_t^r)$ is following.

$$f_m(T_h^r, T_t^r) = \begin{cases} 1.0 & (*cond) \\ 0.0 & (otherwise). \end{cases}$$

$*cond$:*Any of one word of r role in* T_h *matches a word of r role in* T_t.
If predicate words do not match, this score is 0.0. Additionally, if both words of sub and obj role do not match, this score is 0.0.

Word Match Score
SimScore becomes 1.0 if two Syntax Trees are same. However, we observed many cases that SimScore does not become 1.0 nevertheless the meaning of sentences are same. Thereby, we introduce Word Match Score to improve coverage of questions.
Word Match Score is defined by following formula.

$$\begin{aligned} &WMScore(T_h, T_t) \\ &= Boost(T_h, T_t) * \frac{1}{|W_h|} \sum_{w_h \in W_h} max_{w_t \in W_t} f_w(w_h, w_t) \end{aligned}$$

where W_h is set of words in Syntax Tree T_h. The definition of $Boost(T_h, T_t)$ is following.

$$Boost(T_h, T_t) = \begin{cases} 2.0 & (*cond) \\ 1.0 & (otherwise). \end{cases}$$

$*cond$:$SimScore(T_t, T_h)$ *exceeds 0.5.*
The definition of $f_w(w_h, w_t)$ is following.

$$f_w(w_h, w_t) = \begin{cases} 1.0 & (*cond) \\ 0.0 & (otherwise). \end{cases}$$

$*cond$:w_h *matches* w_t.
This score accounts for not only word level matching but also Syntax Tree Similarity through $Boost(T_h, T_t)$ term.

Word Exclusive Match Score
Word Exclusive Match Score is defined by the following formula.

$$WEMScore(T_h, T_t) = max_{w'_h \in W'_h, w'_t \in W'_t} f_e(w'_h, w'_t)$$

where W'_t is set of words which match none of words in W_h and W'_h is set of words which match none of words in W_t. The definition of $f_e(w'_h, w'_t)$ is following.

$$f_e(w'_h, w'_t) = \begin{cases} 1.0 & (*cond) \\ 0.0 & (otherwise). \end{cases}$$

$*cond$:w'_h *is exclusive or antonym word against* w'_t.

5.3.3 Evaluation of each choice

Each choice s is evaluated as below.

1. Make Syntax Trees from choice s.
   Let T_i be i-th Syntax Tree of sentence s and $T = \{T_1, \ldots, T_n\}$.
2. Search T'_i which maximizes $WMScore(T_i, T'_i)$ from knowledge resources. Let T'_i be a Syntax Tree in knowledge resources which maximizes $WMScore$.
3. Calculate $MaxWEMScore$ as below.
   $$\begin{aligned} &MaxWEMScore \\ &= \max_{i \in \{1,\ldots,n\}} WEMScore(T_i, T'_i). \end{aligned}$$
4. Calculate the true degree of s as below.
   $$TrueDegree = \begin{cases} AveWMScore & (*cond) \\ -1.0 * AveWMScore & (Otherwise). \end{cases}$$
   $cond$:$MaxWEMScore = 0$.
   Where
   $$AveWMScore = \frac{1}{|T|} \sum_{i \in \{1,\ldots,n\}} WMScore(T_i, T'_i).$$

After evaluating each choice individually, this solver chooses the answer choice based on $TrueDegree$.

5.3.4 Judgment for True-or-False Question

Solver 3 calculates $TrueDegree$ for each choice. When a question requires to choose the true choice, this solver chooses the choice which has maximum $TrueDegree$. On the other hand, when a question requires to choose the false choice, this solver chooses the choice which has minimum $TrueDegree$".

5.4 solver4

5.4.1 Factoid Scoring

In textbooks, related named entities often appear in the same paragraph, sentence or comma-separated sentence. Therefore, this solver determines word-to-word relations by co-occurrence. Unlike score of PMI, this solver calculates score without considering how many times the words co-occur. This solver considers whether a co-occurrence has occurred or not. Number of different words that co-occur with the target word is also considered. This solver regards a word which has many co-occurrence words like "France" less important than a word which has only a few co-occurrence word like "Single whip law". We make tables of co-occurrence for three types of unit: paragraph, sentence and comma-separated sentence. We consider that the table for unit of sentence is most important.

The strategy of this solver is similar to "Converting to factoid" solver (See 5.2). The solver transforms choices to virtual factoid questions.

We extract NE from the instruction and the choice. Let $Q = q_1, q_2, ..., q_n$ be set of NEs from the instruction and $C = c_1, c_2, ..., c_m$ be set of NEs from the choice . If $m = 0$ or $n + m < 2$ then the solver returns "no answer".

Solver 4 uses following score to solve virtual factoid questions. We define co-occurrence tables as $D = \{d_{paragraph}, d_{sentence}, d_{comma-separated-sentence}\}$, a candidate word as w and a target word as t. Moreover, we define that $Entry_d(t)$ is number of different words that co-occur with t on table d. A co-occurrence score f is defined as:

$$CoOccur_d(t, w) = \begin{cases} 1.0 & (t \text{ and } w \text{ co-occur in } d) \\ 0.0 & (\text{otherwise}) \end{cases}$$

$$f_d(t, w) = \frac{CoOccur_d(t, w)}{Entry_d(t) + a}$$

where a is constant to keep $f_d(t, w)$ to small value. In phase 3, a is 20.

In the instruction Q and the choice C (removed answer of virtual factoid question c), the score that how a candidate word w ($w \in W$; W is all words included in d and same type as c) expected to appear is defined as:

$$WordScore(w, c) = \sum_{d \in D} \sum_{t \in Q \cup C - \{c\}} f_d(t, w).$$

If $\arg\max_{w \in W} WordScore(w, c) = c$, we get correct answer to the virtual factoid question.

5.4.2 Judgment for True-or-False Question

Solver 4 chooses the choice of True-or-False questions by accuracy rate of virtual factoid questions and average of $WordScore$.

5.4.3 Ontology Search

For “4. Time Reordering” type questions, we implement sub-solver. One of the typical format of the questions of this type is to arrange three historical events in order. In these questions, we convert each event into factoid question like “When this event occurred?” and arrange events based on the answer of factoid questions. Another format is to fill an event in blanks of a chronological table. Our solver converts the event into a factoid question and answers in the same way.

When answering these factoid questions, we use event ontology EVT as knowledge resource. This ontology has data of instance like historical events, persons, nations, start date and end date. We get named entities in the instruction and search them in the ontology. Usually the start date is the answer of the factoid question. However, if the question contains keywords like “dead” or “downfall”, the end date is the answer. Questions which the solver can not answer are answered by solver 2. See 5.2.

5.5 Passage Retriever

To retrieve passages that are similar to the choices of the question from the knowledge resources, the term frequency of TF-IDF scoring is not critical, but it is important that the coverage of words in the search query is high. Furthermore, since the approximation of the importance of words by the IDF is not sufficient, it is important to increase the score of the words of world history by registering these words to the morphological analysis dictionary. In that case, it is necessary to hit in a substring of compounds of the world history words to reduce leakage of retrieval.

Thus we tried two methods: registering compounds as synonyms of the words and scoring with an emphasis on coverage of the words in the query.

This passage retriever is used in solver 1 and solver 2. It is based on the Apache Solr, and all passaging is done at index time, and passages are based on paragraph or sentence boundaries.

5.5.1 Synonym Compounds

If the compound is registered to the user dictionary of morphological analysis, we keep the compound in the index as a synonym to get a rank boost. Table 3 shows the example of the index by synonym compounds.

Moreover, we change the morphological dictionary to UniDic[9] to reduce leakage of retrieval. In UniDic, all entries are based on the definition of the short unit word, which provides word segmentation in uniform size for being high in morphological stability.

Table 3: Example of synonym compounds

Index Position	1	2
Morphemes	イスラーム	世界
Synonyms	イスラーム世界	

Table 4: Preliminary experiment results of passage retrieval methods

retrieval methods	DCG_1	DCG_3	DCG_5	DCG_{10}
Baseline	2.025	5.119	6.819	9.762
All	2.216	5.680	7.574	10.850
-BM25	2.020	5.123	6.807	9.736
-UniDic	2.197	5.563	7.391	10.598
-Max	2.207	5.609	7.459	10.659
-Cmpnds -Max	2.226	5.632	7.524	10.760

5.5.2 Scoring for Emphasis on Coverage of the Words

In order to rank passages which coverage rate of the words of search query is high at the top, we adopt Okapi / BM25 weighting, and we set both the TF and document length normalization to be low by parameters of BM25. Further, in the case of registering the compounds as synonyms to the index, the score of compound words is scored double. Thus, the influence of one compound word may be too large. Therefore, we change the scoring of synonyms to MAX.

5.5.3 Preliminary Experiment

Baseline to validate the retrieval methods, we use the default setting of Solr (TF-IDF base scoring, IPA dictionary and no setting of synonyms). We use the choice of the question as the query for the search engine.

We use the data set that is correct 134 sentences of world history exam from RITE-VAL NTCIR-11[10].

The knowledge resources of retrieval are shown in 4.1. We use about 20,000 words of dictionary provided by the Todai Robot Project as user dictionary of morphological analysis. We evaluate by using discounted cumulative gain (DCG), and we calculate relevance as follows:

$$rel_i = 1.5 \times coverage_{ne} + coverage_{cw}$$

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

where rel_i is the relevance of the result at position i. $coverage_{ne}$ is the coverage rate of named entities that exist in the retrieval results of words of the query and $coverage_{cw}$ is the coverage rate of content words. DCG_p is accumulated at a particular rank position p.

Table 4 shows the results of preliminary experiment of passage retrieval methods.

5.6 Combination

Our system consists of 4 solvers (Figure 3). Question analyzer reads each question and analyzed question data (including tagged text of question, instruction and all choices) is passed to each solver. Each solver calculates the score of each choice and returns the rank of each choice. Depending on solvers, they use multiple strategies in accordance with types of questions.

Table 5: NUL score of formal run

Phase	Exam	Priority of runs		
		1	2	3
1	National Center Test (1999)	43	**46**	36
2	Benesse mock exam (2015 Jun/All/out of 175)	121	121	118
(2)	Benesse mock exam (2015 Jun/Pattern 1[10])	**76**	76	76
(2)	Benesse mock exam (2015 Jun/Pattern 2)	64	64	61
3	National Center Test (2011)	65	65	**68**
3	Benesse mock exam (2014 Sep/All/out of 125)	77	76	76
(3)	Benesse mock exam (2014 Sep/Pattern 1)	**60**	57	60
(3)	Benesse mock exam (2014 Sep/Pattern 2)	58	60	54

We choose the final answer by aggregating the rank of each solver using weighted Borda count.

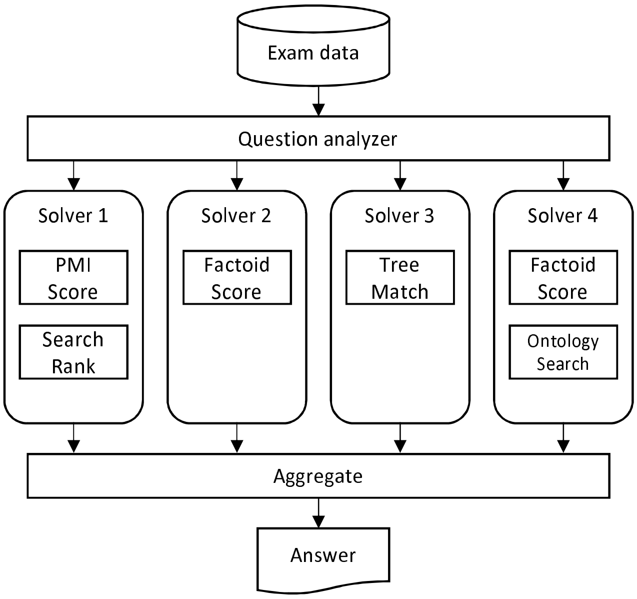


Figure 3: System architecture

6. EXPERIMENTAL RESULTS

Table 5 shows the results of our system at each phase. We changed method of aggregation for answers of four solvers at each run. For example, at phase 3, 1st run was made by an average of each solver, 2nd run was made by voting and 3rd run was made by a weighted average of each solver. The weight of each solver was correct rate of training examples.

We use different solvers on each phase because we improved solver based on results of phase 1 and 2. Therefore, we solved exams of each phase again by solvers of phase 3 and the results are shown in table 6.

Table 6: Scores of National Center Test (NCT) and Benesse mock exam (BME) by solvers of phase 3

Ph.	Exams	So.1	So.2	So.3	So.4	Combined
1	NCT (1999)	46	56	30	40	52/100
2	BME (2015 Jun)	121	104	82	56	125/175
3	NCT (2011)	62	62	43	39	65/100
3	BME (2014 Sep)	58	65	36	33	76/125
-	Total	287	287	191	168	318/500

Since our solvers were tuned for recent exams, the results of National Center Test 1999 (Phase 1) were not improved greatly, because old exams had different wordings from recent ones.

6.1 Results for each type of questions

In this section, we describe scores of each solver and combination solver for each type of questions. Task organizer classified questions into 8 types. We mapped “1. True-or-False” to ”Relative True-or-False Questions”, ”Relative True-or-False in Focus Word Questions” and ”Absolute True-or-False Questions” and “4. Time Reordering” to ”Time Reordering Questions” and ”What-Time Questions”.

The Table 7 shows the numbers of correct answers of National Center Test (2011) and Benesse mock exams (2014 Sep, 2015 Jun). The “-” indicates that the solver did not answer.

Table 7: Numbers of correct of each type of questions

type of question	So.1	So.2	So.3	So.4	Comb.
Relative TF	53/75	54/75	48/75	32/75	56/75
Relative TF in Focus Word	3/4	4/4	2/4	0/4	2/4
Absolute TF	9/17	6/17	7/17	-	9/17
Factoid	8/10	7/10	-	4/10	6/10
Slot-Filling	7/13	10/13	-	4/13	9/13
Time Reordering	-	-	-	6/9	6/9
What-Time	-	-	-	0/3	0/3
Other	7/13	-	-	-	7/13

7. DISCUSSION

Solver 1 has a problem that the PMI score between words that are included within one sentence becomes high even though the pair of the words is unrelated directly, because it does not consider the distance and the dependency of the words. There is also a problem for the search rank. When we set a sentence as a search query, it is handled as ”OR search” in the search engine, the important words in the query may not be included in the high rank of the search

[10] Benesse mock exam of phase 2 has 7 parts and phase 3 has 5 parts. Actually, students answer 4 parts of them. There are 2 patterns of choice. Pattern 1 is for students started learning from ancient history and pattern 2 is from modern history.

results. It becomes the cause that the score of false choice may be a good score.

At True-or-False questions, solver 2 and 4 were confused by questions contain many-to-many relationship. For example, choice "Indian Empire governed by Queen Victoria was founded. (ヴィクトリア女王を皇帝とするインド帝国が成立した。)" is converted to two virtual factoid questions, "What country did Queen Victoria govern?" and "Who did govern Indian Empire?". However, Queen Victoria was also the queen of the United Kingdom and there were other Emperors of India. Therefore, these virtual factoid questions have more than one answer and solver 2 and 4 were confused among them.

At Time Reordering questions and What-Time questions, solver 4 gets time information from all NEs. Frequently, the instruction has two or more NEs which contain time information in ontology. In that case, the solver could not choose the appropriate information.

In solver 3, we observed some typical error patterns described as below.

- There are some cases that words which are defined as different meaning in the dictionaries represent same meaning by consulting the context of questions and knowledge resources.

- Failure of complementing omitted words or complemented by wrong word.

- Meaning of sentences are same but Syntax Trees are different, because there are a lot of patterns of rephrasing.

Thus, to overcome these errors, we need to implement the mechanism to identify word meaning in the context, to resolve smarter co-reference and to handle valid rephrasing patterns.

In addition, about True-or-False questions, we analyzed 10 questions that were mistaken by all solvers in exams (2011, 2014 Sep and 2015 Jun). We observed some error patterns.

1. The named entity is too frequent in knowledge resources (4 cases)
2. Failure of matching synonyms (3 cases)
3. Failure of extracting important named entity from the instruction (2 cases)
4. Lack of recognizing causal relationship (2 cases)
5. The named entity is not contained in the dictionaries (1 cases)
6. Failure of recognizing replacement of subject and object (1 cases)
7. Failure of extracting time expression (1 cases)

From above observations, we found some residual problems such as strict analysis of sentence in questions and knowledge resources, determination of important word of the choice and enhancement of dictionary of named entities, verbs which often used in textbooks of history, synonyms and hypernyms.

Combination scores of 4 solvers are almost higher than each solver. Considering the results, the combination strategy was effective to reduce variance in generalization error.

Numbers of questions without True-or-False questions are not enough in order to determine a tendency of each type. However, there are types of questions that combined score is lower than the best solver. The optimal strategy of combination is future works.

8. CONCLUSIONS

To solve the National Center Test we implemented different 4 solvers and combined them. We obtained 76 points in phase 2 Benesse mock exam (on pattern 1). On the other exams, we obtained about 50 to 70 percent scores. It is the best result among other participants.

9. REFERENCES

[1] Noriko Arai, Takuya Matsuzaki. Can a Robot Get into the University of Tokyo? : The Artificial Intelligence Project at the National Institute of Informatics, Transactions of the Japanese Society for Artificial Intelligence : AI, 2012

[2] Hideyuki Shibuki, Kotaro Sakamoto, Madoka Ishioroshi, Akira Fujita, Yoshinobu Kano, Teruko Mitamura, Tatsunori Mori, Noriko Kando. Task Overview for NTCIR-12 QA Lab-2. Proceedings of the 12th NTCIR Conference, June 7-10, 2016, Tokyo, Japan

[3] Yoshinobu Kano. Solving History Exam by Keyword Distribution: KJP. Proceedings of the 11th NTCIR Conference, December 9-12, 2014, Tokyo, Japan

[4] 金山博, 宮尾祐介. ファクトイド型質問応答を用いた正誤判定問題の解決, 言語処理学会第 19 回年次大会, 発表論文集, 2013

[5] Okita Tsuyoshi, Liu Qun. The Question Answering System of DCUMT in NTCIR-11 QA Lab, Proceedings of the 11th NTCIR Conference,2014

[6] Ran Tian, Yusuke Miyao. Answering Center-exam Questions on History by Textual Inference, The 28th Annual Conference of the Japanese Society for Artificial Intelligence, 2014

[7] Ai Kawazoe, Yusuke Miyao, Takuya Matsuzaki, Hikaru Yokono, Noriko Arai. Event ontology to support reasoning existence/non-existence of events, The 28th Annual Conference of the Japanese Society for Artificial Intelligence, 2014

[8] Church, K. and Hanks, P. 1989. Word association norms, mutual information, and lexicography. In Proceedings of ACL89. pp. 76-83.

[9] Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso, and Hideki Ogura. 2008. A proper approach to Japanese morphological analysis: Dictionary, model, and evaluation. In Proc. of LREC ' 08, pp. 1019-1024.

[10] S. Matsuyoshi, Y. Miyao, T. Shibata, C.-J. Lin, C.-W. Shih, Y. Watanabe, and T. Mitamura. Overview of the NTCIR-11 Recognizing Inference in TExt and Validation (RITE-VAL) Task. In Proceedings of the 11th NTCIR Conference, 2014

[11] 宮下洋, 石井愛, 小林実央, 星野力. センター試験『世界史B』文の正誤判定問題ソルバー, 言語処理学会第 22 回年次大会, 発表論文集, 2016