Wikipedia Based Essay Question Answering System for University Entrance Examination

Takaaki Matsumoto
Carnegie Mellon University
5404 GHC 5000 Forbes Ave.
Pittsburgh, Pennsylvania, USA 15213
SOC Corporation
3-16-17, Takanawa, Minato-ku
Tokyo, Japan 108-0074
takaaki@gmail.com

Francesco Ciannella
Fadi Botros
Evan Chan
Cheng-Ta Chung
Keyang Xu
Tian Tian
fciannella@cmu.edu
fbotros@andrew.cmu.edu
yiksanc@andrew.cmu.edu
chengtac@cs.cmu.edu
keyangx@andrew.cmu.edu
tian.tian@cmu.edu
Carnegie Mellon University
5404 GHC 5000 Forbes Ave.
Pittsburgh, Pennsylvania, USA 15213

Teruko Mitamura
Carnegie Mellon University
6711 GHC 5000 Forbes Ave.
Pittsburgh, Pennsylvania, USA 15213
teruko@cs.cmu.edu

ABSTRACT

This paper describes an open knowledge (Wikipedia) based question answering system that generates essays to answer the real examination questions for the admission to the Tokyo University. Questions are formulated in English and their answers are also expected in English, although they are to be found in Japanese language textbooks. This cross-lingual narrow domain question task is a hard task because most questions are based on the limited target language knowledge base which is only available in its original language. Large scale open-domain knowledge resources will certainly contain the answers, but retrieving them is difficult due to their inherent high signal to noise ratio. To overcome Wikipedia's high signal to noise ratio, we carefully calculate the weights of the keywords extracted from the question, based on a tf-idf score of the entire Wikipedia. The relevant articles are then retrieved and sets of passages are extracted based on the weighted keywords. Cherry picking, generative method, or sentence ordering strategies are subsequently used to generate short or long essays. The results of the end-to-end evaluation indicate that the proposed system succeeded to generate better essays compared with the previous research that also uses Wikipedia and the reference system that uses machine translated Japanese textbooks.

KEYWORDS

Question answering, open knowledge base, summarization, NTCIR-13, world history

1 INTRODUCTION

Question answering (QA) is one of the most notable natural language processing applications and has been heavily researched for several decades. While most research focuses on factoid, true/false and multiple choice QA tasks, essay QA has been proven to be one of the more challenging tasks since it usually requires a deeper understanding of the subject matter, information extraction from multiple sources and summarization to produce a coherent essay.

NTCIR (NII Testbeds and Community for Information access Research) [1] is a series of workshops that expand research in Information Access (IA) technologies including information retrieval, question answering, text summarization, extraction, etc. QA Lab [2], one of the tasks of NTCIR, aims to investigate complex real-world QA technologies as a joint effort of participants. The QA tasks of the NTCIR 13 QA Lab consist of three type of questions: multiple choice, named-entity and essay type questions from Japanese university entrance examination, which focus on world history [18][17].

In this paper, we present our system which participated in the essay QA portion of QA Lab 3. The rest of the paper is layed out as follows. We further explain the task in section 2. In section 3, we discuss the design of the system in detail and show evaluation for each module. Finally, in section 4 we present end-to-end system evaluation.

2 TASK AND REFERENCE SYSTEM

The essay QA task of NTCIR QA Lab 3 contains short/simple and long/complex essays. The former requires an answer essay of one or two sentences (from 15 to 60 words) with some of them containing a factoid type question. The latter expects multiple (usually from 5 to 8) sentences (225-270 words), and has 8-10 keywords that should be used in the essay. Examples of the questions are following:

Short essay The Inca Empire had no writing system, but it controlled the large territory of the Andes. Describe, in 15 English words, the transportation and information methods used by the Empire.

Long essay In answer space (A), in 225 English words or less, describe the historical significance of the philosophies of these intellectuals, including the conditions in the 18th century which led them to these conclusions, especially in France

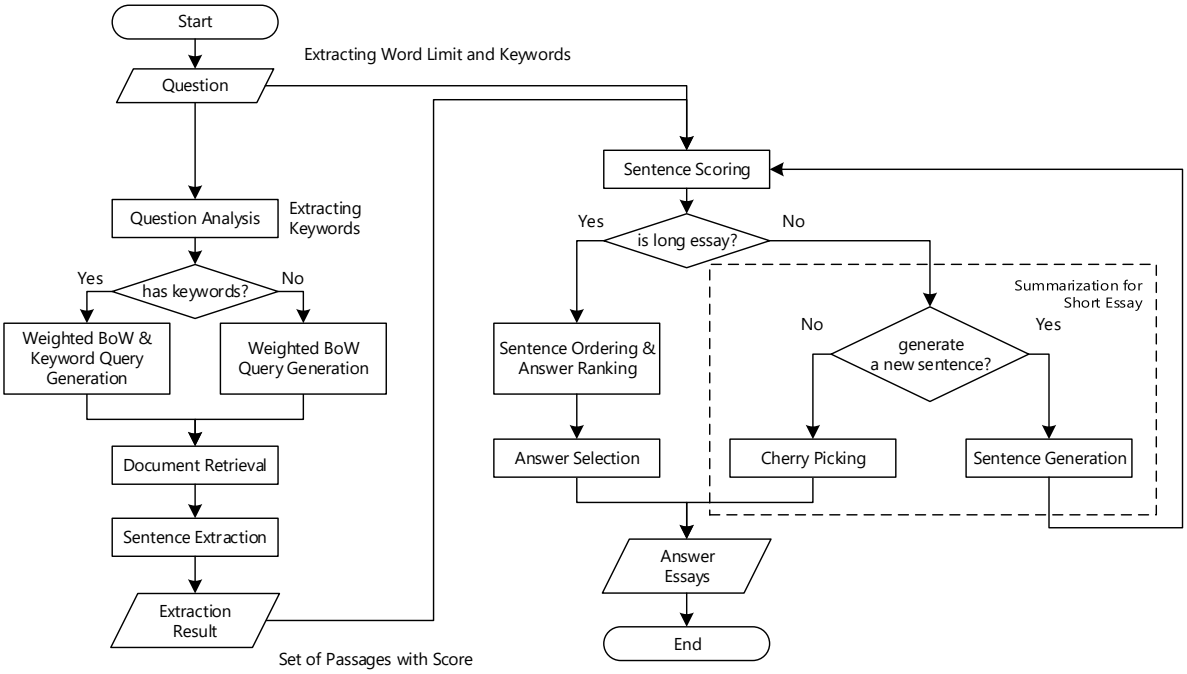

Figure 1: System Flowchart.

and China. Use each of the terms below once, and underline each term when it is used; Society of Jesus, imperial examinations, enlightenment, absolute monarchy, revocation of the Edict of Nantes, French Revolution, class system, Literary Inquisition.

A multilingual essay question answering system developed by Sakamoto's et al. [9][16] has been employed as the reference system. Knowledge resources for the reference system are five machine translated (Google Translate, in 2015) Japanese world history textbooks.

3 SYSTEM MODULES AND THEIR EVALUATIONS

3.1 Overview

Fig. 1 shows the architecture of the end-to-end system. Detailed architecture, algorithms used and experimental design for each of these modules are covered in detail in the following subsections. Communication between each module is performed using JSON files for ease of use and readability. To ensure consistency between each test iteration, we run each end to end test using a build automation software called Jenkins. Whenever a new JSON file is produced and committed to the git repository by the extraction subsystem, the build automation mechanism detects the changes and triggers the start of the summarization system which ultimately produces the results of the evaluation in an HTML report that can be consulted on-line.

The main data source we used to extract answers is Wikipedia. We experiment with a dump of all of Wikipedia and only the history section of Wikipedia [8].

3.2 Question Analysis

Question analysis module is meant to extract all useful information from question data to serve all remaining modules in our end-to-end system. This module is composed of three components, namely, information extraction, text processing and weighted keywords generation.

Information extraction refers to extracting values of a few XML tags (e.g., <instruction></instruction>) which helps solving question answering problems from three sources: qalab3-en-phase1-answersheet-essay.xml, qalab3-en-phase1-essay-extraction-GSN.xml and qalab3-en-phase1-goldstandard-essay.xml.

Extraction is followed by text processing. In NTCIR questions, redundancy for information retrieval exists and all these patterns are needed to be removed otherwise they add noise to information retrieval, e.g., "Write your answers in the answer space".

Weighted keywords generation means generating a list of reasonable keywords with corresponding weights from question text (Note that long essay questions provide a list of keywords). There are multiple ways to generate keywords and assign them different weights. After a series of experiments, we use Tf-idf metrics as the weight generator for two reasons. First of all, the algorithm make few assumption on the data. Secondly, it is properly implemented in scikit-learn.

To calculate tf-idf, we append all 27 question text (i.e., concatenated by instruction, grand_question and reference field) from XML source file to History Wikipedia corpus consisting of 11217 documents, and construct a new corpus with 11244 (=11217+27) documents. Then tf-idf weights on the corpus are calculated, and all phrases in 27 questions are sorted by tf-idf value. After that, for those question with given keywords, append these keywords into keyword list and assign them weight of 1 (heuristically). As a result, those phrases with highest tf-idf values are labeled as keywords, and be sent to information retrieval module along with their tf-idf value as weight.

3.3 Document Retrieval

The document retrieval module indexes information from Wikipedia and retrieves relevant text records against structured queries generated based on questions. The Wikipedia history subset created by Wang et al. [20] was used as the collection for constructing the index. Indri [19] was utilized for indexing, which was stopped using the default Indri stoplist and stemmed using the Krovetz stemmer.

Each Wikipedia page can be indexed as a document, which is the basic unit for retrieval. This is known as the page-level indexing. However, the question answer requires to locate the exact paragraph in extracting specific sentences relevant to the question and the whole Wikipedia page might contain too much noise. In order to increase the accuracy of sentence extraction, we also adopted passage-level indexing; It divided each Wikipedia page into sentences using Standford CoreNLP tool [1] and used a sliding window approach to combine sentences into passages [4]. Here, the sliding windows contains 10 sentences without overlapping and we heuristically chose 10 because the question answer is required to have around 40-60 words.

Structured queries are generated with weighted keywords extracted from the Question Analysis module(see Section 3.2). An example for keywords set "Olympia; Greek; 4th century CE " is shown as follows:

$$\#combine(\alpha_1\ Olympia\ \alpha_3\ Greek\ \alpha_3\ \#1(4th\ century\ CE))$$

where α_1,α_2,α_3 are weights generated by the Question Analysis module; And #1() operator requires all terms inside appear continuously.

The retrieval model is Indri [14], which combines statistical language models and Bayesian inference networks. All parameters were default settings. Top 20 retrieved documents are ranked with Indri scores and returned for the Sentence Extraction module in the next step.

3.4 Sentence Extraction

This module takes the output of the question analyzer and document retrieval modules and extracts sentences that could be potential answers to the question. It uses the original question, the list of retrieved documents and attempts to extract sentences that contain the answer to the question. Since long and short essay questions have different answering requirements, the system uses different strategies to answer them. This module consists of following sub-modules:

[1] https://stanfordnlp.github.io/CoreNLP/

Document Cleaning Raw Wikipedia files are highly noisy: they contain a lot of tables, links, citations, markup, etc. That is why it is important to clean the documents and remove all the unnecessary content. This sub-module also segments the documents into sentences and tokenizes each sentence. These are the steps that are taken to process each document:

(1) Remove documents that do not actually contain any useful text but rather contain a list of links to other pages (e.g. Category pages)
(2) Segment documents into sentences
(3) Filter out sentences that:
  (a) Contain links
  (b) Contain HTML or Wikipedia markup
  (c) Are image captions
(4) Tokenize each sentence:
  (a) Remove non-alphanumeric characters
  (b) Remove stopwords
  (c) Lowercase tokens
  (d) Stem tokens
(5) Remove sentences that contain two tokens or less
(6) Remove duplicate sentences

Passage Extraction There are separate passage extraction sub-modules for short and long essays since different strategies are used to answer each type of question. Each sub-module takes in the output of the question analyzer and the cleaned documents and outputs a list of sentences that are potential answers to the question. The algorithms used by these sub-modules are outlined in detail in the following section.

Evaluation The evaluation sub-module uses the given gold passages to evaluate the extracted passages. It uses both human annotations and automated methods to evaluate the performance of passage extraction. It also contains scripts that attempt to make human annotation of extractions as fast and efficient as possible.

3.4.1 Algorithms. Following algorithms were tested to select the most suitable algorithms for sentence extraction.

Jaccard Similarity Similarity is calculated between all the words in the question (introduction/instruction paragraphs and given keywords) and each sentence from the retrieved documents then the top 10 sentences with the highest scores are chosen.

Field-weighted Jaccard Similarity : Since the introduction paragraph is usually longer than the instruction paragraph, the sentences extracted using Jaccard similarity tended to be more relevant to the introduction paragraph but not to the actual instruction paragraph. Therefore, to remedy this problem, the following formula was used to give more weight to the instruction paragraph:

$$\begin{aligned}\text{Score}(\text{Question}, \text{Sentence}) = \\ 0.7 * \text{Jaccard}(\text{Instruction}, \text{Sentence}) \\ + 0.3 * \text{Jaccard}(\text{Introduction}, \text{Sentence})\end{aligned} \quad (1)$$

Field-weighted Jaccard + MMR Wikipedia contains many sentences that are very similar to each other terms of content. Therefore, sometimes the system would return 10 sentences that are all very similar. This is not very beneficial for this

task, especially for long essay questions where we want to cover a wide range of topics. To diversify the extracted sentences, MMR is used. The essence of MMR [5], which is a greedy algorithm, is in each iteration, it would pick passage that has high relevant score with question but also with little overlap with selected passages.

Field-weighted TF-IDF History questions tend to contain many names of people, events, places and special words that should be given more weight since the question is usually focused on those words. Therefore, TF-IDF and cosine similarity are used to rank sentences. IDF values are calculated using the entire Wikipedia corpus.

Field-weighted TF-IDF + PM2 Long essay questions contain keywords that have to be used and discussed in the essay. However, the previous methods cannot guarantee that all keywords were covered in the extracted passages. It is possible that all the extracted passages are only relevant to one keyword (or none at all). The PM2 diversification algorithm [6] is used to try to increase keyword coverage for long essay questions. PM2 is generally used in document retrieval when the query can have multiple intents and we want to retrieve documents that address all the intents proportionally. It gives a higher score to documents that cover multiple intents. Similarity, in long essay questions we want to retrieve sentences that cover each keyword proportionally and give extra weight to sentences that cover multiple keywords. Sentences that cover multiple keywords can connect the given concepts and potentially produce a more coherent essay.

3.4.2 Evaluation. We focused on human annotations to evaluate the extracted passages. A binary relevance metric was used to evaluate each extracted passage and precision @10 and mean reciprocal rank were then calculated for each experiment. For long essay questions, keyword recall is also evaluated by measuring the fraction of keywords that are present in the extracted passages.

Table 1 summarizes the results for each of the tested algorithms. The results show that field-weighted TF-IDF + PM2 gave the best results for all metrics.

As mentioned above, using simple Jaccard similarity is naive since most of the extracted passages were relevant to the introduction paragraph but not to the actual question. Using field-weighted Jaccard doubled the precision scores which indicates that it is effective. While MMR reduced redundancy, the results show that it didn't improve precision or MRR thus it is questionable whether it is useful or not for this task. As predicted, TF-IDF was a very effective method to improve results as demonstrated by the improved precision and MRR. However, TF-IDF gave the lowest keyword recall for long essays but PM2 proved effective as it doubled keyword recall while also improving precision/MRR for long essays.

Table 1: Evaluation Result of Passage Extraction Algorithms

Algorithm	Short Essays		Long Essays		
	P@10	MRR	P@10	MRR	Keyword Recall
Jaccard	0.077	0.330	0.520	0.850	0.447
Field-weighted Jaccard	0.150	0.432	0.700	0.767	0.509
Field-weighted Jaccard + MMR	0.109	0.444	0.660	0.733	0.529
Field-weighted TF-IDF	0.191	0.447	0.679	0.750	0.376
Field-weighted TF-IDF + PM2	0.191	0.447	0.720	0.900	0.714

3.5 Sentence Scoring

The sentence scoring module gives a score to the extracted set of the passages. Since the questions are entrance examination, they need the existence of important keywords in the essay. Therefore, the simplest sentence scoring methods is measuring keyword entailment.

$$\text{Score} = \frac{k_s}{m} \tag{2}$$

where k_s is the number of keywords in the sentence, and m is the number of words of the sentence. All keywords and words of the sentence are stemmed. Stop words and punctuations are removed before calculation.

Eq.2 measures the density of the keywords in a sentence. However, not always the given keywords and words in the sentence exact match. Some words of the answer sentence could be similar to the given keywords. Hence, word level similarity between retrieved or given keywords and an extracted sentence is calculated as follows:

$$\text{Score1} = \sum_{i=1}^{m} \frac{\max(w_i \cdot k_1, w_i \cdot k_2, ...w_i \cdot k_n)}{m} \tag{3}$$

where, m is the number of words in the sentence, n is the number of keywords, w is the word vector, k is the keyword vector. Word embedding is given by GloVe (6B 100d) [13].

Eq.3 calculates similarity between given keywords and all words in an extracted passages. With this scoring method, the mean of the ROUGE-1, which is one of the official answer scoring methods of the NTCIR QA Lab [17], are improved (from 0.0598 to 0.0671) in the phase-1 data. However, dividing by the sentence length means measurement of the similarity density of a passage. In general, the longer sentence, the more information exists. Hence, we can modify the formula as follows:

$$\text{Score2} = \sum_{i=1}^{m} \frac{\max(w_i \cdot k_1, w_i \cdot k_2, ...w_i \cdot k_n)}{\log m} \tag{4}$$

The objective of the division by logarithm of sentence length is to consider the information density and amount simultaneously. The ROUGE-1 mean improved compared with the previous formula (from 0.0671 to 0.0680).

Above sentence scoring methods are keyword based (word level) approach. Today it is not difficult to calculate sentence embedding vector. Assuming that an entailment exists between questions and answers, sentence score can be given as following:

$$\text{Score3} = \max(\text{sim}(s, q_1), \text{sim}(s, q_2), ..., \text{sim}(s, q_l)) \tag{5}$$

where, sim is the function to calculate sentence similarity between two sentences, s is the extracted sentence, q_i is the i-th sentence of the question, and l is the number of sentences of the question. Sentence similarity is calculated by a siamese Long Short-Term Memory (LSTM) [12]. The siamese LSTM is one of the state of the art to assess semantic similarity between sentences. It uses word-embedding vectors supplemented with synonymic information to the LSTMs, which outputs a fixed size vector to encode the meaning expressed in a sentence. By calculating simple Manhattan metric, it gives the sentence representations to form a space which reflects semantic relationships.

3.6 Text Ordering for Long Essay

Answer candidates for long essays are generated by this module. This module has two models. The first one is K-Means model, which tries to capture the relation between sentences to generate coherent essay. The other one is MMR model, which does not aim at coherent essay. Instead, it tries to diversify the topics to generate the essay.

3.6.1 K-Means model. In [21], Zhang proposed summary generation by using global and local coherence. The intuition of this model is that there are 2 kinds of coherence: global coherence and local coherence. The global coherence means the connectivity between remote sentences. It is more like sub-topic transition, for example usually essay would cover "cause" of events first, then the "result" of events. On the other hand, local coherence indicates the connectivity between adjacent sentences, such as using some transition words to connect two sentences. Because coherence can be regarded as some kind of similarity between sentences, thus, this module adopts cosine similarity to measure the coherence.

To capture the coherence, this module applies K-means in scikit-learn package [3] to cluster the input passages. this module assumes that each cluster is related to different sub-topics, as the similarity within each sub-topics should be very similar. Each passage is represented as a word vector, whose value is tf-idf of the words in each dimension.

After the clustering, the next step would be to generate the order of these clusters, the sequence of sub-topics, to achieve global coherence. To do this, the system greedily pick most coherent cluster with ordered clusters. For local coherence, the strategy is similar that the system would greedily pick passage from the cluster with maximum coherence with selected sentences.

3.6.2 MMR Model. For this model, the idea is that while K-Means model may generate coherent sentence sequence, the gold standard essay is not usually coherent because it has not only to cover all specified keywords but also to fulfill the words length constraint as well. Therefore, it may be useful to select sentences that cover keywords from different aspects but also be relevant to the question. Another reason is that although MMR may not be able to generate coherent essay, the evaluation metric does not consider the coherence either. Thus, it would still be beneficial if the system can select good candidate sentences.

3.6.3 Evaluation. The dataset for the evaluate of this long essay generation module are the gold standard passages and gold standard essays provided by NTCIR. There are 5 long essay questions, and each of them is associated with several passages and 3 gold standard essays. In this evaluation, the gold standard passages are used as input to the system, and gold standard essays are used to evaluate the essay generation system. The evaluation metrics is ROUGE-1 and ROUGE-2 means[10].

Table 2 shows the performance for these 2 models, and different parameter K for K-Means models. The combined method is to pick an essay generated from above methods that has highest relevant score with question. We can see that ROUGE-1 score is the same for all K-Means methods, it is because the sentence removal strategy would remove almost the same sentences, and ROUGE-1 only measures on single words. For ROUGE-2, the score is different as it measures on bi-gram, it improves when K grows from 1 to 3, then decreases gradually after that. It indicates that the clustering is effective, while the number of clusters should not be too large, as there are generally around 7 to 9 sentences in the gold standard essays.

Table 2: Results of long essay models evaluation

Method	ROUGE-1	ROUGE-2
K-Means (K=1)	0.584	0.356
K-Means (K=3)	0.584	0.358
K-Means (K=5)	0.584	0.357
K-Means (K=7)	0.584	0.352
MMR	0.596	0.396
Combined	0.596	0.396

3.7 Summarization for Short Essay

The Summarization for short essay module provides a way to summarize a set of sentences coming from the upper layers to produce a fixed length short essay, following the directions provided in the question. The summarization paradigm that has been used is the abstractive summarization, which tries to leverage on the semantics of the sentences to achieve the text compression. The module uses three possible summarization techniques, returning only the best result to the evaluation module. Two of the techniques are actually pure abstractive summarization techniques, the third one is a trivial NLP based summarization technique. The first two techniques are implementations of the two main research approaches in abstractive summarization: AMR graph merging and Neural Network with attentional model.

3.7.1 AMR model [11]. Abstractive summarization is one of the hard NLP tasks that is still an open field of research with very few techniques, unlike other NLP tasks. It it a task that cannot be decoupled from semantics: to be able to create an abstract summary of a passage, one needs to have a deep insight into what is the meaning that the passage bears. Therefore we thought to use AMR, which is one of the resources available in NLP for implementing semantics. A thorough description of the algorithms that we used can be found in [11], and we remind the reader to that paper for the details. We implemented the algorithms described in that paper, and on top of it we laid down the basis to add to the pipeline the generative model (in the paper the generation of the summary from the summarized AMR graph is left to a mere bag of words). The generative model is able to create a well formed sentence from

an AMR graph (with the limitations of AMR, like for instance the impossibility of using verbs tenses).

AMR provides a whole-sentence semantic representation, represented as a rooted, directed, acyclic graph. Nodes of an AMR graph are labeled with concepts, and edges are labeled with relations. Concepts can be English words, PropBank event predicates, or special keywords. The core semantic relations are adopted from the PropBank annotations; other semantic relations include "location," "mode," "name," "time," and "topic."

In the AMR summarization framework, summarization consists of three steps

(1) parsing the input sentences to individual AMR graphs,
(2) combining and transforming those graphs into a single summary AMR graph
(3) generating text from the summary graph.

The graph summarizer, first merges AMR graphs for each input sentence through a concept merging step, in which coreferent nodes of the graphs are merged; a sentence conjunction step, which connects the root of each sentence's AMR graph to a dummy "ROOT" node; and an optional graph expansion step, where additional edges are added to create a fully dense graph on the sentence level. These steps result in a single connected source graph. A subset of the nodes and arcs from the source graph are then selected for inclusion in the summary graph. Ideally this is a condensed representation of the most salient semantic content from the source.

We used the proxy report section of the AMR Bank because a dataset for a summarization task should include inputs and their summaries, each with gold-standard AMR annotations. A proxy report is created by annotators based on a single newswire article, selected from the English Gigaword corpus.

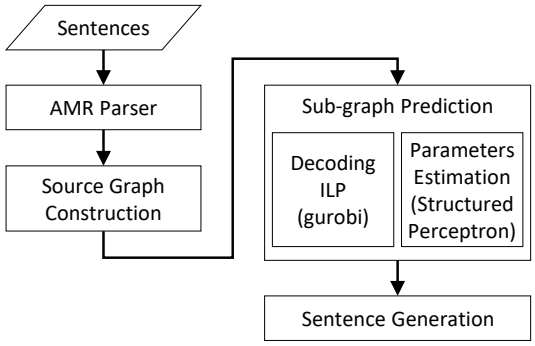

Figure 2: AMR Model Architecture

Fig. 2 shows the architecture of the AMR model. The summaries of components are following:

Source Graph Construction The "source graph" is a single graph constructed using the individual sentences' AMR graphs by merging identical concepts. Concept merging involves collapsing certain graph fragments into a single concept, then merging all concepts that have the same label.
Ideally, a source graph should cover all of the gold standard edges, so that summarization can be accomplished by selecting a subgraph of the source graph

Subgraph Prediction This steps selects a summary subgraph from the source graph. This is done with a structured prediction algorithm that enforces the following constraints in the statistical model for subgraph selection: include information without altering the meaning, maintain brevity, and produce fluent language. The selection of the graphs is done using ILP (Integer Linear Programming)

Decoding Decoding is performed as an ILP task with the constraints that the output forms a connected subcomponent of the source graph.
The length constraint is used to fix the size of the summary graph (measured by the number of edges). This is an important parameter in that the performance of a summarization systems depends strongly on their compression rate, and it is important for the NTCIR purpose because of the length limitations on the essays. An exact ILP solver called Gurobi is used.

Parameter Estimation Source graphs and summary graphs, represent a collection of input and output pairs, therefore we can use a Machine Learning algorithm like the structured perceptron to learn the parameters of the objective function designed in the previous set.

Generation Generation is the weakest link in the current chain. At the moment it is no more than a bag of words, but the plan is to plug it into a language generator from AMR.

3.7.2 Neural model with attention [15]. The idea of using a neural attentional model for summarization comes after the recent success of neural machine translation. The idea is to combine a neural language model with a contextual input encoder that learns a latent soft alignment over the input text to help inform the summary. Both the encoder and the generation model are trained jointly on the sentence summarization task.

Given an input sentence, the goal is to produce a condensed summary. The key takeaway is that the abstractive summarization task can be formulated mathematically as in finding the output sequence $\boldsymbol{y}$ that maximizes a scoring function over the input sequence $\boldsymbol{x}$ and $\boldsymbol{y}$ itself.

The problem boils down in modeling the following probability:

$$p\left(\boldsymbol{y}_{i+1}|\boldsymbol{x}, \boldsymbol{y}_c; \theta\right) \tag{6}$$

where $\boldsymbol{y}_c$ is a window of size c over the previous tokens in the output sequence and θ is the parameter of the neural network. Instead of using a noisy-channel approach, the original distribution is directly parametrized as a neural network. The network contains both a neural probabilistic language model and an encoder which acts as a conditional summarization model.

The attentional based model can be regarded as a model that learns a soft alignment, P, between the input and the summary.

Once we have the augmented language model, the generation of a summary is a search problem over a scoring function:

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y} \in Y} \sum_{i=0}^{N-1} g\left(\boldsymbol{y}_{i+1}, \boldsymbol{x}, \boldsymbol{y}_c\right) \tag{7}$$

where N is the length of the output sentence and g is the scoring function. This is the decoding problem that can be accomplished using beam search.

The training dataset is the Gigaword dataset. The golden standard summary is the headline of the news article and the body of the document is represented by the first two sentences in the article.

3.7.3 Pick one sentence. The pick one sentence method simply selects one sentence among the candidate ones, based on the relevance score provided by the scoring system and by the closeness to the required sentence's length.

This model was also designed to implement some basic NLP features, like for instance providing the sentence up to a punctuation mark or by pruning the lexical pare tree of the sentence ad hoc but so far we figured that the simple greedy method worked better in terms of evaluation scores.

4 END-TO-END EVALUATION

Table 3 shows the settings of the proposed system. The combination of "sentence similarity scoring (SentSimScoring)" and "generative short essay (Generative)" was not attempted because of both algorithm takes very long time.

Table 4 shows the summary of our system result. NTCIR employs machine evaluation and human experts to score essays. The ground truth essays are provided from the NTCIR official data. Since ROUGE-1 and 2 [10] are used one of the evaluation methods of the NTCIR QA Lab[17], ROUGE-1 and 2 scores of the proposed system were calculated using the evaluation function of the reference system [16].

Comparing the all systems in the Table 4, the Wiki-WordSimScore-PickOne has the best end-to-end ROUGE-1 mean. In addition, even though it should be noted that the results of the proposed system and the previous research cannot be simply compared because of the different questions, the ROUGE-1 mean score of the answers generated by Wiki-ExtractionScore-PickOne is about four times larger than that of the previous study that also uses Wikipedia (0.0326 in ROUGE-1 mean) [7].

The previous research by Day et al. [7] is the only available result for the NTCIR QA Lab essay QA task. The reference system (FelisCatusZero) developed by Sakamoto et al. is also the only open source software for the NTCIR QA Lab task. Compared with these two studies, the proposed system achieved high ROUGE-1 mean for the NTCIR QA Lab 3 phase-1 data. However, it also should be noted that the number of the question is only a few (5 long essays and 22 short essays). Since the NTCIR QA Lab uses the real past entrance examination of University of Tokyo, the provided data was very small. Considering the standard deviations in the table 4, the performance differences are not statistically significant.

Table 5 shows the comparison of end-to-end, short and long essay task ROUGE-1 and 2 means. It indicates that most of the ROUGE-1 and 2 mean progress comes from the short essays. Generative algorithm for short essay, Wiki-WordSimScore-Generative, was relatively worse than cherry picking (Wiki-WordSimScore-PickOne) for short essay task, however, in some questions the generative model worked better than the cherry picking.

As for the long essay question, the ROUGE-1 means of all four end-to-end conditions (FelisCatusZero, Wiki-ExtractionScore-PickOne, Wiki-WordSimScore-PickOne, and Wiki-SentSimScore-PickOne) are approx. 0.2. These results indicates that the effectiveness of the sentence scoring methods are almost the same, even if their methodologies are different. However, the ROUGE-1 mean of GSN-WordSimScore-PickOne which used the gold standard extraction result was 0.58. The difference between the gold standard and end-to-end runs indicates that knowledge resource or document retrieval can be improved to write a good essay.

In all settings, short essay performances are lower than those of long essays. This difference is attributed to the lack of keywords of the answer in short essay. In short essay, necessary important terms (mainly proper noun) are not given in contrast to long essay question.

4.1 Answer Examples

The system answers and gold standards for the example questions shown in Section 2 are following:

Gold Standard for Short Essay
It used roads around Cuzco and knotted ropes called quipu.

System Answer (Pick One) for Short Essay
Inca road system.

System Answer (Generative) for Short Essay
road developed system

Gold Standard for Long Essay
The Society of Jesus, which engaged in missionary work overseas, was also active in China, bringing information about China to Europe. The scientific revolution of 18th century Europe brought about the Enlightenment, especially in France, with its focus on reason and equality. Voltaire praised China for lacking doctrines which were contrary to reason. This was in response to Catholic control of France since the reign of Louis XIV, who abolished the Edict of Nantes, which granted Protestant the same rights as Catholics. Reynal praised China for not having hereditary nobility. His aim was to contrast France, with its fixed class system, to China, whose appointment of ministers under the imperial examination system ensured some degree of social mobility. Montesquieu, however, criticized China's tyrannical authoritarian system. By criticizing China's restriction of free speech through the Literary Inquisition, he meant to implicitly criticize France's system of absolute monarchy. In these ways, the Enlightenment criticized France's authoritarian religion, class system, and absolute monarchy, and created the philosophical foundation of the French Revolution which overturned the absolute monarchy.

System Answer for Long Essay
For de Tocqueville, the Revolution was the inevitable result of the radical opposition created in the 18th century between the monarchy and the men of letters of the Enlightenment. It was instead the French Revolution, by destroying the old cultural and economic restraints of patronage and corporatism (guilds), that opened French society to female participation, particularly in the literary sphere.All this is not to say that intellectual interpretations no longer exist. By the end of the

Table 3: System Settings

System Name	Extraction Source	Scoring Method	Short Essay
Wiki-ExtractionScore-PickOne	Wikipedia	Extraction Score	Cherry Picking
Wiki-WordSimScore-PickOne	Wikipedia	Word Similarity (Eq. 4)	Cherry Picking
Wiki-WordSimScore-Generative	Wikipedia	Word Similarity (Eq. 4)	Generative (AMR)
Wiki-SentSimScore-PickOne	Wikipedia	Sentence Similarity (Eq. 5)	Cherry Picking
GSN-WordSimScore-NA	Gold Standard	Word Similarity (Eq. 4)	N.A. (Long essay only)

Table 4: End-to-end Evaluation Result of Each System

System	Evaluation Method	Number of Questions	Mean	Max	Median	Min	Variance	Standard Deviation
FelisCatusZero	ROUGE-1	27	0.063	0.244	0	0	0.007	0.081
	ROUGE-2	27	0.009	0.067	0	0	0.000	0.018
Wiki-ExtractionScore-PickOne	ROUGE-1	27	0.118	0.261	0.143	0	0.009	0.093
	ROUGE-2	27	0.030	0.143	0	0	0.002	0.041
Wiki-WordSimScore-PickOne	ROUGE-1	27	0.123	0.32	0.1	0	0.008	0.088
	ROUGE-2	27	0.025	0.167	0	0	0.002	0.042
Wiki-WordSimScore-Generative	ROUGE-1	27	0.079	0.234	0.057	0	0.007	0.081
	ROUGE-2	27	0.013	0.105	0	0	0.001	0.026
Wiki-SentSimScore-PickOne	ROUGE-1	27	0.107	0.348	0.095	0	0.010	0.098
	ROUGE-2	27	0.023	0.174	0	0	0.002	0.043

Table 5: Comparison of End-to-end, Short and Long essay task ROUGE-1 and 2 Means.

System	End-to-end ROUGE-1 Mean	End-to-end ROUGE-2 Mean	Short Essay ROUGE-1 Mean	Short Essay ROUGE-2 Mean	Long Essay ROUGE-1 Mean	Long Essay ROUGE-2 Mean
FelisCatusZero	0.063	0.010	0.032	0.004	0.202	0.032
Wiki-ExtractionScore-PickOne	0.118	0.030	0.097	0.028	0.210	0.041
Wiki-WordSimScore-PickOne	0.123	0.023	0.105	0.021	0.203	0.040
Wiki-WordSimScore-Generative	0.079	0.025	0.051	0.007	0.203	0.040
Wiki-SentSimScore-PickOne	0.107	0.012	0.086	0.017	0.201	0.05
GSN-WordSimScore-NA					0.584	0.359

18th century, prominent French philosophers and literary personalities of the day, including Anne-Robert-Jacques Turgot, were making persuasive arguments to promote religious tolerance. The edict paved the way for the most far-reaching reforms in terms of their social consequences, including the creation of a national education system and the abolition of the imperial examinations in 1905.

5 CONCLUSIONS

In this paper, the Wikipedia based essay question answering system for world history subject question of university entrance examination was discussed. Six modules; question analysis, document retrieval, sentence extraction, sentence scoring, short essay generation, and sentence ordering are described and tested. The proposed system extracts keywords from the question text, and weights of the keywords are determined based on tf-idf score of the entire Wikipedia. Related articles are retrieved in whole Wikipedia and important sentences are extracted based on the weighted keywords. Cherry picking or generative method are attempted to generate for short essay. For a long essay, sentence ordering is used. The results of the end-to-end evaluation indicated that the proposed system succeeded to generate better essays compared with the the only reference system which uses machine translated textbooks as the knowledge resource. However, the performance difference was not statistically significant because the number of provided dataset was small. In addition, even though it should be noted that the results of the proposed system and the previous research cannot be simply compared because of the different questions, the ROUGE-1 mean score of the answers generated by the proposed system is about three times larger than that of the previous study that also uses Wikipedia, 0.0326 [7]. Failure analysis of the proposed system is future work.

ACKNOWLEDGMENTS

We thank Prof. Eric Nyberg for assistance with the development of the essay QA system.

REFERENCES

[1] 2017. *The 13th NTCIR.* http://research.nii.ac.jp/ntcir/ntcir-13/.

[2] 2017. *NTCIR-13 QA Lab-3.* http://research.nii.ac.jp/qalab/.

[3] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. 2013. API design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238* (2013).

[4] James P Callan. 1994. Passage-level evidence in document retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., 302–310.

[5] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 335–336.

[6] Van Dang and W Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 65–74.

[7] Min-Yuh Day, Cheng-Chia Tsai, Wei-Chun Chuang, Jin-Kun Lin, Hsiu-Yuan Chang, Tzu-Jui Sun, Yuan-Jie Tsai, Yi-Heng Chiang, Cheng-Zhi Han, Wei-Ming Chen, Yun-Da Tsai, Yi-Jing Lin, Yue-Da Lin, Yu-Ming Guo, Ching-Yuan Chien, and Cheng-Hung Lee. 2016. IMTKU Question Answering System for World History Exams at NTCIR-12 QA Lab2. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016.*

[8] Dheeru Dua, Bhawna Juneja, Sanchit Agarwal, Kotaro Sakamoto, Di Wang, and Teruko Mitamura. 2016. CMUQA: Multiple-Choice Question Answering at NTCIR-12 QA Lab-2 Task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016.*

[9] Sakamoto Kotaro. 2017. *FelisCatus Zero-multilingual.* https://github.com/ktr-skmt/FelisCatusZero-multilingual.

[10] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, Vol. 8. Barcelona, Spain.

[11] Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman M. Sadeh, and Noah A. Smith. 2015. Toward Abstractive Summarization Using Semantic Representations. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar (Eds.). The Association for Computational Linguistics, 1077–1086. http://aclweb.org/anthology/N/N15/N15-1114.pdf

[12] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *AAAI*. 2786–2792.

[13] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*, Vol. 14. 1532–1543.

[14] Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 275–281.

[15] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. *CoRR* abs/1509.00685 (2015). http://arxiv.org/abs/1509.00685

[16] Kotaro Sakamoto, Takaaki Matsumoto, Madoka Ishioroshi, Hideyuki Shibuki, Tatsunori Mori, Noriko Kando, and Teruko Mitamura. 2017. FelisCatusZero: A world history essay question answering for the University of TokyoâĂŹs entrance exam. In *Proceedings of Open Knowledge Base and Question Answering Workshop at SIGIR*. (to appear).

[17] Hideyuki Shibuki, Kotaro Sakamoto, Madoka Ishioroshi, Akira Fujita, Yoshinobu Kano, Teruko Mitamura, Tatsunori Mori, and Noriko Kando. 2016. Overview of the NTCIR-12 QA Lab-2 Task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016.*

[18] Hideyuki Shibuki, Kotaro Sakamoto, Yoshinobu Kano, Teruko Mitamura, Madoka Ishioroshi, Kelly Y. Itakura, Di Wang, Tatsunori Mori, and Noriko Kando. 2014. Overview of the NTCIR-11 QA-Lab Task. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, Japan, December 9-12, 2014.*

[19] Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. 2005. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, Vol. 2. Amherst, MA, USA, 2–6.

[20] Di Wang, Leonid Boytsov, Jun Araki, Alkesh Patel, Jeff Gee, Zhengzhong Liu, Eric Nyberg, and Teruko Mitamura. 2014. CMU Multiple-choice Question Answering System at NTCIR-11 QA-Lab. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, Japan, December 9-12, 2014.* http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/QALab/05-NTCIR11-QALAB-WangD.pdf

[21] Renxian Zhang. 2011. Sentence ordering driven by local and global coherence for summary generation. In *Proceedings of the ACL 2011 Student Session*. Association for Computational Linguistics, 6–11.