# KSU Systems at the NTCIR-14 QA Lab–PoliInfo Task

Tasuku Kimura[1], Ryo Tagami[1][0000−0002−8109−0962], Hikaru Katsuyama[2], Sho Sugimoto[1], and Hisashi Miyamori[1]

[1] Division of Frontier Informatics, Graduate School of Kyoto Sangyo University, Motoyama, Kamigamo, Kita-ku, Kyoto-shi 603-8555, Japan
{i1658047,i1788124,i1888097,miya}@cc.kyoto-su.ac.jp
https://www.kyoto-su.ac.jp/

[2] Faculty of Computer Science and Engineering, Kyoto Sangyo University, Motoyama, Kamigamo, Kita-ku, Kyoto-shi 603-8555, Japan
g1544377@cc.kyoto-su.ac.jp
https://www.kyoto-su.ac.jp/

**Abstract.** In this paper, the systems and the results of the team KSU for QA Lab–PoliInfo Task in NTCIR–14 are described. First, in Segmentation Task which required extracting primary information correctly from the input data, we proposed a method based on rules and vocabulary distributions. In Summarization Task which demanded generating a summary focused on a specific topic, we tried using a framework of the query–focused abstractive summarization. Finally, in Classification Task which called for classifying stances of a certain text for a specific topic, we developed a method combining deep learning and two–stage classifiers. As a result, the team KSU achieved third in five teams with the f–measure of 0.855 in Segmentation Task, and second place in 11 teams with the accuracy 0.934 in Classification Task.

**Team Name.** KSU

**Subtasks.** Segmentation task (Japanese)
Summarization task (Japanese)
Classification task (Japanese)

**Keywords:** text segmentation · queryfocused summarization · two-step classification strategy · neural network

## 1 Introduction

In recent years, due to the influence of fake news, the importance of fact verification has been reconfirmed, which verifies the authenticity about the information already posted. For example, when we verify the authenticity of an utterance by politicians who are easily subject to be a target of fake news, it is necessary to check the primary sources such as the assembly minutes. However, it is difficult to confirm the opinion of the assemblymen at a glance because there is a vast

amount of contents in such assembly minutes, including every progress of the proceedings and the materials submitted.

In NTCIR–14, QA Lab–PoliInfo Task[3] was carried out, which consists of three subtasks to develop fundamental technologies against recent fake news problems.

First, Segmentation Task is a subtask which requires identification of the corresponding primary sources, given the secondary information such as newspaper articles and microblogs. In this subtask, all the assembly minutes during a certain period and a certain text which is a summary of an utterance by an assemblyman are given as input, and it is required to determine the range of the original utterance corresponding to the given text. Here, an "utterance" represents the content which an assemblyman spoke at one rostrum. We built a method based on rules and vocabulary distributions, to utilize the cues peculiar to local assembly minutes as well as to secure the versatility.

Next, Summarization Task is a subtask which demands to generate a summary without losing its original intention from one utterance by a speaker. In this subtask, it is necessary to generate a summary, given one utterance, a topic, the length of the summary, and the set of the assembly minutes. We regarded this subtask as a query–focused summarization and constructed an automatic abstractive summarization model with deep learning. In addition, we introduced a mechanism to control the summary length proposed by Kikuchi et al. into the proposed model because controlling the summary length is also an important factor in this subtask.

Finally, Classification Task is a subtask which calls for a stance classification targeting only utterances with useful evidences. In this subtask, it is necessary to classify three kinds of labels, namely, relevance, fact verifiability and stance, from the given utterance in the minutes and the given policy topic. Also, it requires to classify the utterance to the topic into one of the followings: support with fact-verifiable reasons, against with fact-verifiable reasons, and other. For simplicity, the support with fact–verifiable reasons will be referred to as support, and the against with fact–verifiable reasons referred to as against. We developed a method to judge relevance and fact verifiability with deep learning, and constructed a method to classify stances with two stage classifiers.

This paper is organized as follows. In sections 2, 3, and 4, the methods are explained in detail and the results are shown with discussion, for Segmentation Task, Summarization Task, and Classification Task, respectively. Finally, section 5 presents our conclusion.

## 2 Segmentation Task

In this section the approach we took for segmentation task is explained. We call the datasets provided by the organizers as follows:

– MD, which stands for minutes data and corresponds to the minutes of Tokyo Metropolitan Assembly from April 2011 to March 2015,

– SD, which stands for summary data and represents a summary of an assemblyman's speech.

This sub-task needs to segment a certain range within the speech corresponding to SD as primary information from MD. In this section, we define the content of a speech in one appearance by an assemblyman as one speech.

### 2.1 Pre-processing

Before carrying out the actual segmentation process, the MD is indexed to the search engine. One sentence (or one object) in the MD is regarded as one document. Before indexing, the following two pre-processings are performed.

First, each sentence in MD is classified to the corresponding single speech. Since a speech has no clear boundary in MD, the separation of the utterance is clarified in this process. Each sentence in MD is scanned in order, and the sentences in the same speech is assigned the same section (speech) number. During the scanning, a new section number is assigned if the pre-defined rules are satisfied.

Next, the speech type (ST) is estimated for each clarified speech. This estimation result is used in the following segmentation processing. We defined seven STs for the local assembly as follows:

**PROGRESS** a speech for a chairperson to advance the meeting,
**QUESTION** a speech by an assemblyman to ask interpellation,
**ANSWER** a speech by an officer to respond to QUESTION,
**REQUEST** a speech by an assemblyman to further request for ANSWER,
**OPINION** a speech by an assemblyman to express their claims in favor or against a certain topic,
**REPORT** a speech by an officer to explain past events or backgrounds of agenda proposals, and
**GREETING** opening remarks, policy speeches, etc.

ST classification is done by the text classification method proposed by Joulin et al.[1] We manually labeled the following minutes collected from the Web, and constructed a model for ST estimation:

**Tokyo Metropolitan Assembly** minutes of 45 meetings randomly chosen from the ones held from 1989 to 2017,
**Itabashi City Assembly** minutes of 16 meetings held in 2017,
**Aichi Prefecture Assembly** minutes of 27 meetings in 2017, and
**Kagawa Prefecture Assembly** minutes of 22 meetings in 2017.

We used the first two sentences and the last two sentences of one speech as the features. In the preliminary experiment, we used 90% of the dataset for training data and 10% for test data, and the estimated accuracy for the test data was about 99.3%.

By this pre-processing, "section number" which the sentences belonging to the same speech share and "estimated ST" are newly added to each sentence of MD. In this approach, one sentence attached with the section number and the estimated ST is indexed in the search engine as one document.

### 2.2 Segmentation Process

Figure 1 shows an outline of the segmentation process.

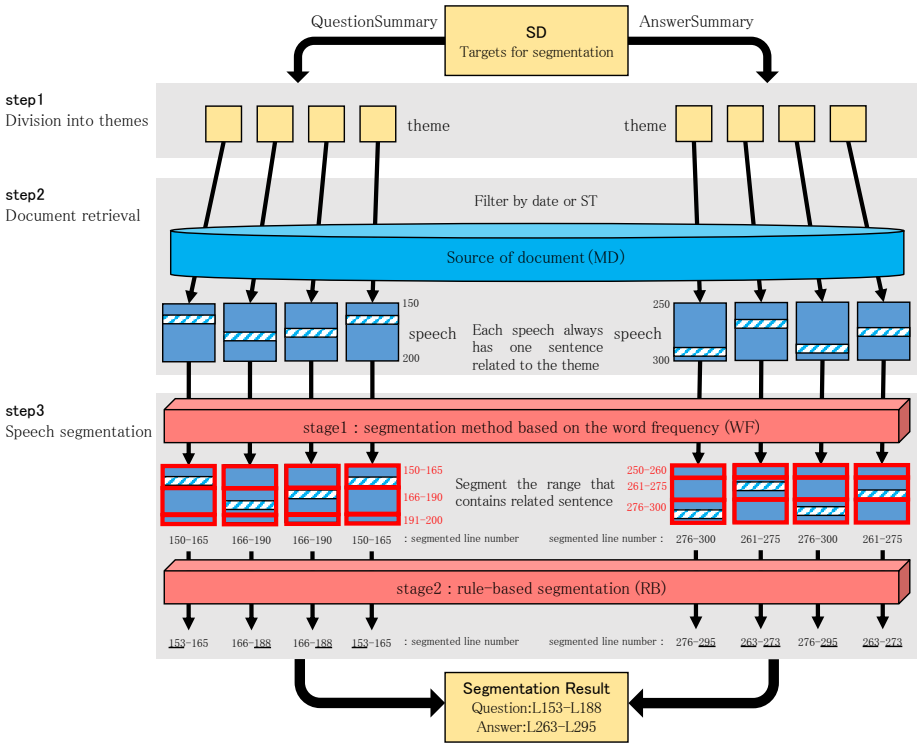


**Fig. 1.** Outline of the segmentation process.

**Step1: Division into Themes** One SD may contain multiple themes in a single summary ("QuestionSummary" or "AnswerSummary") (e.g. 1. Enhance senior sports promotion, 2. Revise the sports promotion basic plan and formulate new promotion guidelines, 3. Tell us your opinions for the success of sports festival Tokyo 2013). We divide the summary into themes, apply step2 and later for each theme, and finally integrate the result. The summary is divided by the bracketed heading numbers.

**Step2: Document Retrieval** A retrieval is performed on MD for each theme obtained in step 1. The relevant document obtained by the retrieval is one sentence in a speech. The query is made by concatenating the strings composing the theme and the strings composing the subtopic of SD. Additionally, the retrieval results are filtered according to the following two conditions:

– search only for documents having the same date as in SD, and
– search only for documents filtered by the speaker filter (SF).

In this paragraph, SF is explained. If we search with the query described above, another assemblyman's sentence including similar surface strings may be hit. Since SD includes the name of the speaker asking questions as Question-Speaker, SF is applied in order to search only for sentences asking questions given by the corresponding assemblyman. When searching for answer speech, the retrieval is performed only for sentences which appears immediately after the obtained question speech and whose ST is ANSWER, because it is obvious that the parts we want to segment appears in the answer speech immediately after the question, considering the structure of the assembly minutes. If multiple people answer one question, each of the answers will be the target documents to be retrieved.

If no relevant documents are retrieved by applying SF partly because different notations of names are used in MD and SD, SF is not applied. If SF is intentionally spared in the first place or if SF cannot be used because of the above reason, the documents whose STs are QUESTION will be the target to be retrieved for question speech, and those whose STs are ANSWER will be for answer speech.

As a result of the above processing, the most relevant documents are obtained as for question speech and for answer speech, respectively. Since the unit of the document is one sentence instead of one speech, the corresponding speech is obtained by concatenating the adjacent documents with the same section number as of the retrieved document. In the obtained speech, the information on which sentence is the relevant document retrieved is retained for use in the next step.

**Step3: Speech Segmentation** In this step, a portion which matches with the theme obtained in step1 is extracted from the speech obtained in step 2, One speech is divided into segments, and the segments matching with the respective themes are determined as the segmentation result.

The segmentation process has two stages, at each of which the segmentation result is output. Figure 2 shows how the speech is divided and the appropriate parts are extracted.

In the first stage, the text segmentation method based on the word frequency proposed by Utiyama et al.[6] is used. Their method, hereafter referred to as WF, uses only the word frequency from the text as the indicator and selects a division that maximizes the division probability. By using WF, one speech obtained in step2 is ideally divided into some segments, each of which includes a single theme and corresponds to a question or an answer. Therefore, a segment including "the sentence obtained by the retrieval" held in step 2 is taken as a segmentation result.

In the second stage, a rule-based segmentation method, hereafter referred to as RB, is further applied to the first-stage segmentation result. This is done to cope with the case where the first stage result is segmented broader than the ideal segment result. Figure 2 shows such an example. In the Assembly, the phrases

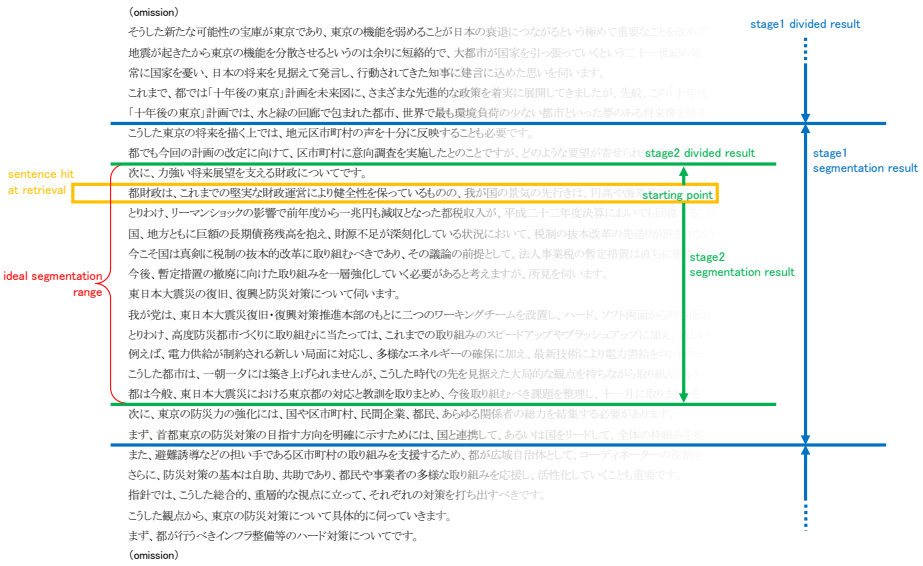


**Fig. 2.** Overview of two-step segmentation process.

such as “next” and “then I’d like to ask about” are often used when the speaker moves on to the different topics. Therefore, when a commonly used phrase for changing topics appear, a rule is applied where the position of the phrase is interpreted as a boundary of the segment. Actually, the speech is scanned from “the sentence obtained at the retrieval” in the forward and backward directions, and if a sentence matching the rule appears, the portion up to that sentence is taken as the segmentation result.

By the above process, the segmentation result for one theme divided in step 1 is determined. This processing is performed for each theme, and the results are finally integrated in one SD. Two segmentation results are output for one SD for a question and an answer.

### 2.3 Results

In the formal run of the segmentation task, we tried eight conditions corresponding to the different combination of the presence or absence of SF, WF and RB. Of these, 4 conditions are the result as official formal run. Table 1 shows the results.

### 2.4 Discussion

**Effect of SF** The segmentation results of C1 and C2 correspond to one speech by an assemblyman, including the range required by SD because there is no speech segmentation process. The required range should always be included in

**Table 1.** Conditions for exepriment and the result of segmentation

| | official | conditions | | | all | | | question | | | answer | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # | priority | SF | WF | RB | R | P | F | R | P | F | R | P | F |
| C1 | | — | — | — | 0.940 | 0.087 | 0.160 | 0.920 | 0.065 | 0.122 | **0.982** | 0.275 | 0.430 |
| C2 | | ✓ | — | — | **0.991** | 0.112 | 0.202 | **1.000** | 0.088 | 0.162 | 0.973 | 0.278 | 0.433 |
| C3 | KSU-01 | — | ✓ | — | 0.779 | 0.243 | 0.370 | 0.835 | 0.192 | 0.311 | 0.662 | 0.841 | 0.740 |
| C4 | | — | — | ✓ | 0.906 | 0.294 | 0.444 | 0.879 | 0.221 | 0.353 | 0.964 | 0.797 | 0.873 |
| C5 | KSU-03 | ✓ | ✓ | — | 0.820 | 0.661 | 0.732 | 0.899 | 0.612 | 0.728 | 0.655 | 0.857 | 0.742 |
| C6 | KSU-02 | — | ✓ | ✓ | 0.759 | 0.268 | 0.396 | 0.806 | 0.209 | 0.331 | 0.660 | 0.949 | 0.778 |
| C7 | | ✓ | — | ✓ | 0.952 | 0.857 | **0.902** | 0.953 | **0.881** | **0.916** | 0.950 | 0.812 | **0.875** |
| C8 | KSU-04 | ✓ | ✓ | ✓ | 0.797 | **0.922** | 0.855 | 0.866 | 0.905 | 0.885 | 0.651 | **0.974** | 0.780 |

one speech, and it is ideal that the recall of C1 and C2 be as close to 1.0 as possible. Since the recall of C1 "all" is 0.940, it was found that in some SD, the speech that are completely different from the required range are acquired by the retrieval. C2 which adopted SF to C1 improves the recall of "all" to 0.991. Thus, it can be seen that SF have been able to obtain more appropriate speeches. This is considered to be because the acquisition of another assemblyman's speech, which includes similar surface string sentences, was suppressed.

**Effect of Segmentation** Segmentation processing is expected to improve the precision because it eliminates unnecessary sentences from the speech to extract the required range. However, if it eliminates necessary sentences, it will lead to a decrease in the recall. So, the division at the appropriate position is important. Both C3 and C4 gave higher F-measure than C1. It is notable that C4 improved the precision of C1 while maintaining almost the same recall. C2, C5, and C7, all of which use SF, also showed the same tendency. From the above, it was confirmed that the two segmentation processings both contribute to the improvement of the accuracy and RB improves F-measure more greatly than WF.

The following example was observed, in which the segmentation result by RB was more appropriate than that by WF. The ideal segment in a speech and the following segment had similar vocabulary while their themes are different. In case of WF, the segmentation result was not appropriate because it regards the two segments as having the same theme. Meanwhile, in case of RB, the segmentation was appropriate in this example because it divides the speech at the position where the expressions often used at the boundaries of the theme appears. In contrast, another example was observed, where RB could not achieve appropriate segmentation because the phrase often used at the boundaries of the theme did not appear. In this example, WF achieved appropriate segmentation, because it identifies the difference in vocabulary used between different themes. From the above, it can be seen that the two segmentation processings have their advantages and disadvantages. The rule-based segmentation method, RB, showed better results with the minutes dataset. However, it is difficult to create

comprehensive rules manually. Hence, the segmentation method based on word frequency, WF, is considered to be more general.

## 3 Summarization Task

We regarded this task as query-focused summarization task and constructed the automatic text summarization model by deep learning.

### 3.1 Policies of model configuration

First, in order to train the summarization model, we constructed a data set using "the Assembly minutes collected from the Web" and "the Newsletters which contain highlights of the Assembly minutes" for deep learning. The Newsletter is a public relations magazine for residents, which is composed of the summarized speech from the minutes. Note that the minutes of Tokyo Metropolitan Assembly in the fiscal years used in the formal run were excluded from the data set.

- Minutes of Tokyo Metropolitan Assembly from 2001 to 2017.
- Minutes of Itabashi City Assembly from 2009 to 2017.

However, it is difficult to say that the data set of 19,689 minutes were sufficient amount for deep learning (problem 1). Also, it is difficult to deal with unknown words if only the vocabulary in the data set is used, since the data set is constructed from the minutes of the specific Assemblies (problem 2).

In this paper, in order to solve the problems of 1 and 2, we built a vocabulary using SentencePiece[4] which is a kind of subword tokenizer. Subword tokenizers treat high frequency words in the training data as one word, and divide low frequency words into shorter units such as substrings and characters. This process eliminates the unknown words in a specific language and solves the problem 2. Also, SentencePiece which can provide unigram-based tokenizers can output multiple segmentation candidates with confidence degree for the same input. Therefore, problem 2 can be solved because the training data can be sampled dynamically from the corpus to augment data.

Next, this subtask requires to generate a summary within the specified number of characters, not just a short summary. Therefore, in order to control the output length, we adopted the LenEmb mechanism that controls the output length proposed by Kikuchi et al[2].

Finally, in order to solve the problem of the Recurrent Neural Network (RNN) which repeatedly output the same phrase, we introduced into the proposed model the diversity cell proposed by Nema et al[5]. The diversity cell converts the context vectors generated by attention so that they are orthogonal to the previously generated context vector in each decoding step.

**Table 2.** Comparison of the system configuration in each priority for Summarization Task

| Priority | tokenizer | diversity cell | LenEmb |
|---|---|---|---|
| KSU-01 | MeCab | ✓ | — |
| KSU-02 | MeCab | — | — |
| KSU-03 | SentencePiece | ✓ | — |
| KSU-04 | SentencePiece | — | — |
| KSU-05 | MeCab | ✓ | ✓ |
| KSU-06 | SentencePiece | ✓ | ✓ |

### 3.2 Proposed model

We constructed six models by combining the three mechanisms, namely, the tokenizer, the diversity cell, and the LenEmb. Table 2 shows the difference of the model configurations. Also, the configuration of the model of priority 5 is shown in Fig.3 as an example of the proposed model.

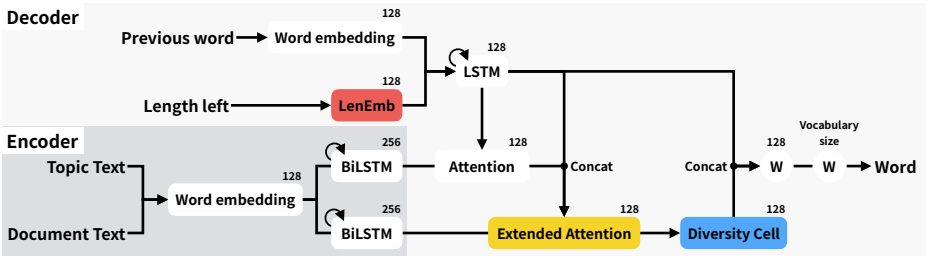


**Fig. 3.** Model configuration of priority 5.

In this section, we describe the method to give diversity to the output, the module that controls the output length, and the attention mechanism considering the topic, respectively.

**Diversity Cell** The role of this mechanism is to solve the problem of RNN which repeatedly generates the same tokens in each decoding step. Therefore, Nema et al. defined the mechanism $\mathbf{SD_2}$ that transforms the input vector into vectors orthogonal to each other in each decoding step by extending the implementation of LSTM by the following equation:

$$\begin{pmatrix} i_j \\ f_j \\ o_j \\ \hat{c}_j \\ g_j \end{pmatrix} = \begin{pmatrix} W_i & U_i \\ W_f & U_f \\ W_o & U_o \\ W_c & U_c \\ W_g & U_g \end{pmatrix} \begin{pmatrix} d_j \\ h_{j-1} \end{pmatrix} + \begin{pmatrix} b_i \\ b_f \\ b_o \\ b_c \\ b_g \end{pmatrix}, \tag{1}$$

$$c_j = \sigma(i_j) \odot \tanh(\hat{c}_j) + \sigma(f_j) \odot c_{j-1}, \tag{2}$$

$$c_j^{diverse} = c_j - \sigma(g_j) \frac{c_j^\top c_{j-1}}{c_{j-1}^\top c_{j-1}} c_{j-1}, \tag{3}$$

$$d'_j = \sigma(o_j) \odot \tanh(c_j^{diverse}), \tag{4}$$

where, $W_i$, $W_f$, $W_o$, $W_g$, $W_c \in \mathbb{R}^{l_2 \times l_1}$, $U_i$, $U_f$, $U_o$, $U_g$, $U_c \in \mathbb{R}^{l_2 \times l_2}$, $d_j$ is the vector of the $l_1$.

**LenEmb** Kikuchi et al. proposed four methods to control the output length of the encoder-decoder model. LenEmb is a method to introduce the length embedding vector to the input of LSTM. The remaining length of the summary in each decoding step is converted to the length embedding vector via the length embedding layer. With this mechanism, the model can generate a summary according to the remaining length information.

**Extended attention mechanism** The role of this mechanism is to generate a context vector of the feature vector, which is calculated from the encoding step of each encoded vector as a weighted average of the elements to be emphasized in the decoding step. The document context vector of the decoding step is calculated by the following equation: The document context vector should be controlled by the topic context vector. Therefore, the document context vector has a parameter $Z_t \in \mathbb{R}^{l_2 \times l_3}$ in order to receive the topic context vector.

$$a_{j,i}^d = v_d^\top \tanh(W_d h_j^o + U_d h_i^d + Z_t t_j), \tag{5}$$

$$\alpha_{j,i}^d = \frac{\exp(a_{j,i}^d)}{\sum_{i'=1}^{l_{dt}} \exp(a_{j,i'}^d)}, \tag{6}$$

$$d_j = \sum_{i=1}^{l_{dt}} \alpha_{j,i}^d h_i^d, \tag{7}$$

where, $W_d \in \mathbb{R}^{l_2 \times l_4}$, $U_d \in \mathbb{R}^{l_2 \times l_2}$, $v_d \in \mathbb{R}^{l_2}$, and $t_j$ is the vector of the $l_3$-dimension.

### 3.3 Results

Chainer was used for implementing the models. Also, SentencePiece was used as the subword tokenizer and MeCab was used as the dictionary-based tokenizer. We set the vocabulary size of SentencePiece to 8,000 and that of MeCab to 42,343. Table 3 shows the quality question scores.

**Table 3.** Quality question scores in Formal run (max is 2)

| Priority | all-topic | | | | single-topic | | | | multi-topic | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | content | | formed | total | content | | formed | total | content | | formed | total |
| | X=0 | X=2 | | | X=0 | X=2 | | | X=0 | X=2 | | |
| KSU-01 | 0.043 | 0.043 | **1.955** | 0.048 | 0.052 | 0.052 | **1.934** | 0.057 | 0.033 | 0.033 | **1.978** | 0.038 |
| KSU-02 | 0.076 | 0.121 | 1.745 | 0.071 | 0.080 | 0.156 | 1.722 | 0.104 | 0.071 | 0.082 | 1.772 | 0.033 |
| KSU-03 | 0.091 | 0.157 | 1.715 | **0.104** | 0.104 | 0.179 | 1.731 | **0.156** | 0.076 | 0.130 | 1.696 | 0.043 |
| KSU-04 | **0.111** | 0.167 | 1.419 | 0.093 | **0.118** | **0.193** | 1.420 | 0.132 | **0.103** | 0.136 | 1.418 | 0.049 |
| KSU-05 | 0.048 | 0.078 | 1.692 | 0.048 | 0.057 | 0.085 | 1.726 | 0.057 | 0.038 | 0.071 | 1.652 | 0.038 |
| KSU-06 | 0.078 | **0.169** | 1.535 | 0.091 | 0.085 | 0.151 | 1.542 | 0.094 | 0.071 | **0.190** | 1.527 | **0.087** |

### 3.4 Discussion

First, the influence of each tokenizer on the summary is discussed by comparing KSU-01 and KSU-03, KSU-02 and KSU-04, and KSU-05 and KSU-06, respectively, from Table 3. By using SentencePiece as a tokenizer, it was confirmed that the score of content increases whereas the score of formed decreases. From the above, it is considered that the model with SentencePiece could deal with unknown words appropriately while the possibility of outputting a summary containing unnatural grammar increased. Also, a detailed breakdown of the number of content showed that the number of correct answers increased, which are different from correct answers provided by the organizers.

Next, the influence of the diversity cell on the summary is discussed by comparing KSU-01 and KSU-02, and KSU-03 and KSU-04, respectively, from Table 3. By using the diversity cell, it was confirmed that the score of formed increases. This is because the problem of repeated generation of the same words has been alleviated by the diversity cell. On the other hand, it can be said that the predicted word vectors should not necessarily be orthogonal in each decoding step because the score of content decreased.

Finally, the influence of the LenEmb on the summary is discussed by comparing KSU-01 and KSU-05, and KSU-03 and KSU-06, respectively, from Table 3. By using LenEmb, it was confirmed that both the score of content and that of formed decrease. By checking the content of the generated summaries, it was also confirmed that the remaining length of the summary had a great effect on the content of output. In other words, it is considered that the content of the summary tends to change according to the remaining length, not the topic.

## 4 Classification Task

In this section, we describe the method of classifying stance of politician's remarks in the local assembly, based on the distributed representation and the features based on the expressions unique to local assembly. Given the utterance and the topic, the proposed method classifies three points of views, namely, relevance, fact-checkability, and opinion. After that, the method decides the final

stance of utterance for the given topic by considering the three classification results.

### 4.1 Dataset for Classification Task

Here, we describe the dataset of stance to each topic, which is used to construct classifiers. In this study, we attached the correct labels to the development data using the following two methods. The first method is to decide the correct labels by taking a majority vote of labels attached by multiple people. The second method was used only to decide the correct labels for relevance. The second method is to decide the correct labels as "unrelated" if even one person attached "unrelated".

### 4.2 Approach for Classification Task

**label of Relevance** Here, a binary classifier is constructed, that decides whether or not the given utterance is related to the given topic. In this paper, we adopted a classifier proposed by Joulin. This classifier is one-layered neural network that takes a sentence split into word unit as input and outputs a probability value. We used a single sentence obtained by concatenating the topic and the utterance as input.

**label of Fact-checkability** A binary classifier is built, that decides whether or not the content of the utterance can be validated as facts. The classifier is composed of a neural network that encodes an utterance with Long-Short Term Memory (LSTM) and outputs a binary probability value by fully connected layer.

**label of Opinion** Here, we construct a ternary classifier that decides whether the content of the utterance is "no opinion", "support" or "against". Opinion requires a ternary classification different from relevance and fact-checkability. Therefore, the proposed method combines two binary classifiers to achieve ternary classification. The first classifier identifies whether an utterance is "no opinion" or "having opinion". The second classifier decides whether the utterance which was classified as "having opinion" is "support" or "against". Each binary classifier used the features based on the vocabulary unique to the opinion as input. Also, Support Vecter Machine (SVM) was used for training the classifiers.

**Selection of the features** The occurrence frequency histogram of word N-grams(N=1,2,3) was made from the utterances in the development data per each label, and the top-n word N-grams having the largest difference in frequency were selected as a feature for each label. Word N-grams were extracted by morphological analysis using MeCab from each utterance. On each label, the

top 200, 400, and 600 N-grams were selected from the occurrence frequency histogram of the extracted N-grams, to be the number of basic feature dimensions used in this work. Also, we tried several kinds of features in the proposed method by using each N-gram features by themselves or by combining the multiple kinds of N-gram features. For the combined features, we used only 200 dimensions of features.

### 4.3 Results

**Result of Relevance** The parameters of fastText used for the classifier were decided from the preliminary experiment as follows: word N-gram = 2-gram, dimension of word feature = 50, the number of epochs = 50, learning rate = 1.0, error function = hierarchy softmax. Also, the correct labels were given to the data set by both methods 1 and 2. Here, the classification accuracy of method 1 is referred to as "Model:Re1", and that of method 2 as "Model:Re2". Table 4 shows the result of relevance.

**Table 4.** Test result of classifying relevance in Formal Run

| Model | SC_Rl_Acc | SC_Rl_P0 | SC_Rl_P1 | SC_Rl_R0 | SC_Rl_R1 |
|---|---|---|---|---|---|
| Re1 | 0.790 | 0.373 | 0.966 | 0.823 | 0.785 |
| Re2 | 0.873 | 0.567 | 0.893 | 0.257 | 0.969 |

**Result of Fact-checkability** The parameters of the model were decided as follows: the dimension of context vector = 50, the number of fully connected layer = 1, activation function = softmax, the number of epochs = 20, error function = cross entropy. Adam was used as an optimization algorithm. Other parameters are set to $\alpha = 0.001$, $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\epsilon = 10^{-8}$. Also, the correct labels were given to the data set by method 1, and used in the proposed method. Table 5 shows the classification results for the test data by the proposed model.

**Table 5.** Test result of classifying fact-checkability in Formal Run

| Model | SC_FC_Acc | SC_FC_P0 | SC_FC_P1 | SC_FC_R0 | SC_FC_R1 |
|---|---|---|---|---|---|
| Fc1 | 0.735 | 0.738 | 0.722 | 0.914 | 0.407 |

**Result of Opinion** Opinion is determined by combining two binary classifiers. For each of the classifiers, the top two models with highest accuracy in the preliminary experiment were used. First, for the classifier identifying whether it has "no opinion" or "having opinion", the model learned form 1-gram features of 600 dimensions (Classifier1-1) and the model learned from 600 dimensions obtained by concatenating 1-gram, 2-gram, and 3-gram features of 200 dimensions (Classifier1-2) were selected. Second, for the classifier identifying whether it has "support" or "against", the model learned from 1-gram features of 600 dimensions (Classifier2-1) and the model learned from 1-gram features of 400 dimensions (Classifier2-2) were chosen. Also, the correct labels were given to the data set by method 1. Four classifiers were constructed by combining the two classifiers having two models, respectively. "Model:St1" is the classifier combining Classifier1-1 and Classifier1-2, "Model:St2" the one combining Classifier1-1 and Classifier2-2, "Model:St3" the one combining Classifier1-2 and Classifier1-2, and, "Model:St4" the one combining Classifier1-2 and Classifier2-2. Table 6 shows the result of opinion.

**Table 6.** Test result of classifying opinion in Formal Run

| Model | SC_St_Acc | SC_St_P0 | SC_St_P1 | SC_St_P2 | SC_St_R0 | SC_St_R1 | SC_St_R2 |
|---|---|---|---|---|---|---|---|
| St1 | 0.802 | 0.829 | 0.683 | 0.402 | 0.961 | 0.230 | 0.237 |
| St2 | 0.799 | 0.829 | 0.724 | 0.370 | 0.961 | 0.201 | 0.254 |
| St3 | 0.801 | 0.820 | 0.720 | 0.420 | 0.973 | 0.171 | 0.202 |
| St4 | 0.799 | 0.820 | 0.732 | 0.404 | 0.973 | 0.153 | 0.214 |

**Result of Class** Table 8 shows the final results of classifying stance decided by three kinds of labels, i.e. relevance, fact-checkability, and opinion. Table 7 shows the specific combination for each priority

### 4.4 Discussion

**Discussion on relevance** It can be seen from Table 4 that SC_Rl_R0 of model:Re2 is lower than that of model:Re1. This is considered to be due to the fact that the number of related and unrelated data in the data set for training was imbalanced. Meanwhile, it can be seen form Table 4 that SC_Rl_P0 of model:Re1 is lower than that of model:Re2. Since the model:Re1 was learned with the development data adjusted to have a large proportion of unrelated labels, it is considered that the value of SC_Rl_R0 increased while the number classified as unrelated increased. In addition, it is confirmed that SC_Rl_R0 by model:Re1 improved because the training data was adjusted to have a larger proportion of unrelated labels.

**Table 7.** Explain of Priority

| Priority | Relevance | Fact-checkability | Opinion |
|---|---|---|---|
| 1 | Re1 | Fc1 | St1 |
| 2 | Re1 | Fc1 | St2 |
| 3 | Re1 | Fc1 | St3 |
| 4 | Re1 | Fc1 | St4 |
| 5 | Re2 | Fc1 | St1 |
| 6 | Re2 | Fc1 | St2 |
| 7 | Re2 | Fc1 | St3 |
| 8 | Re2 | Fc1 | St4 |

**Table 8.** Test result of classifying class in Formal Run

| Priority | SC_Cl_Acc | SC_Cl_P0 | SC_Cl_P1 | SC_Cl_P2 | SC_Cl_R0 | SC_Cl_R1 | SC_Cl_R2 |
|---|---|---|---|---|---|---|---|
| 1 | 0.932 | 0.937 | 0.579 | 0.056 | 0.995 | 0.075 | 0.008 |
| 2 | 0.932 | 0.937 | 0.689 | 0.042 | 0.995 | 0.071 | 0.008 |
| 3 | 0.934 | 0.937 | 0.738 | 0.083 | 0.998 | 0.071 | 0.008 |
| 4 | 0.934 | 0.937 | 0.738 | 0.083 | 0.998 | 0.071 | 0.008 |
| 5 | 0.932 | 0.937 | 0.579 | 0.111 | 0.995 | 0.075 | 0.019 |
| 6 | 0.932 | 0.937 | 0.689 | 0.088 | 0.995 | 0.071 | 0.019 |
| 7 | 0.934 | 0.937 | 0.738 | 0.100 | 0.997 | 0.071 | 0.011 |
| 8 | 0.934 | 0.937 | 0.738 | 0.100 | 0.997 | 0.071 | 0.011 |

**Discussion on fact-checkability** Table 5 shows that the proposed method tends to judge "not fact-checkable", because SC_FC_R1 was lower than SC_FC_R0. This is considered to be the fact that there is a large proportion of "not fact-checkable" labels in the training data and that the correct labels were biased.

**Discussion on opinion** It can be observed from Table 6 that SC_St_R1 and SC_St_R2 are much lower than SC_St_R0 in each model. It is considered that the numbers of "no opinion" and "having opinion" labels in the training set were imbalanced. Also, from Table 6, it can be noticed that SC_St_P2 is much smaller than SC_St_P1. Originally, in the proposed method, only the data classified as "having opinion" by the first classifier was input to the second classifier, and the input was intended to be classified as either "support" or "against". However, it was found that at the time of participation in Formal Run on November, 2019, the data classified as "no opinion" by the first classifier was also used for training in the second classifier. As a result of examining the training data used for the second classifier, the data classified as "no opinion" by the first classifier had been treated as "against" in the second classifier. From the above,

it is considered that the number of data misclassified as "against" has much increased in the second classifier submitted to Formal Run.

**Discussion on final stance** It can be confirmed from Table 8 that in every case, the proposed model achieved high values in SC_Cl_P0 and SC_Cl_R0, whereas it gave low values in SC_Cl_P1 and SC_Cl_R1, and SC_Cl_P2 and SC_Cl_R2. It means that each proposed model has high ability to correctly estimate the final stance as Other, whereas they have low ability to accurately decide whether it is Fact-checkable Support or Fact-checkable Against. It is considered that both the recall of Fact-checkable Support and that of Fact-checkable Against in the final classification results were affected, because both the classification accuracy of "fact checkable" and that of "Support" and "Against" were low.

## 5 Conclusion

This paper described the systems and results of the team KSU for QA Lab–PoliInfo Task in NTCIR–14 First, in Segmentation Task, we proposed a method based on rules and vocabulary distributions. In Summarization Task, we tried using a framework of the query–focused abstractive summarization. Finally, in Classification Task, we developed a method combining deep learning and two–stage classifiers. As a result, the team KSU achieved third in five teams with the f–measure of 0.855 in Segmentation Task, and second place in 11 teams with the accuracy 0.934 in Classification Task.

### Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP18K11557.

## References

1. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 427–431. Association for Computational Linguistics (2017)
2. Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., Okumura, M.: Controlling Output Length in Neural Encoder-Decoders. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1328–1338 (2016)
3. Kimura, Y., Shibuki, H., Ototake, H., Uchida, Y., Takamaru, K., Sakamoto, K., Ishioroshi, M., Mitamura, T., Kando, N., Mori, T., Yuasa, H., Sekine, S., , Inui, K.: Overview of the NTCIR-14 QA Lab-PoliInfo Task. In: Proceedings of the 14th NTCIR Conference (2019)
4. Kudo, T., Richardson, J.: Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 66–71 (2018)

5. Nema, P., Khapra, M.M., Laha, A., Ravindran, B.: Diversity driven attention model for query-based abstractive summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1063–1072. Association for Computational Linguistics (2017). https://doi.org/10.18653/v1/P17-1098, http://aclweb.org/anthology/P17-1098
6. Utiyama, M., Isahara, H.: A statistical model for domain-independent text segmentation. In: In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics. pp. 491–498 (2001)