

Summarizing Utterances from Japanese Assembly Minutes using Political Sentence-BERT-based Method for QA Lab-PoliInfo-2 Task of NTCIR-15

Daiki Shirafuji

Graduate School of Information Science and Technology,
Hokkaido University, Japan
d_shirafuji@ist.hokudai.ac.jp

Rafal Rzepka

Faculty of Information Science and Technology,
Hokkaido University, Japan
rzezka@ist.hokudai.ac.jp

Hiromichi Kameya

School of Engineering, Hokkaido University, Japan
hkameya@eis.hokudai.ac.jp

Kenji Araki

Faculty of Information Science and Technology,
Hokkaido University, Japan
araki@ist.hokudai.ac.jp

ABSTRACT

There are many discussions held during political meetings, and a large number of utterances for various topics is included in their transcripts. We need to read all of them if we want to follow speakers' intentions or opinions about a given topic. To avoid such a costly and time-consuming process to grasp often longish discussions, NLP researchers work on generating concise summaries of utterances. Summarization subtask in QA Lab-PoliInfo-2 task of the NTCIR-15 addresses this problem for Japanese utterances in assembly minutes, and our team (SKRA) participated in this subtask. As a first step for summarizing utterances, we created a new pre-trained sentence embedding model, i.e. the Japanese Political Sentence-BERT. With this model, we summarize utterances without labelled data. This paper describes our approach to solving the task and discusses its results.

TEAM NAME

SKRA

SUBTASKS

Dialog Summarization task (Japanese)

1 INTRODUCTION

Researchers have recently been focusing on summarizing meetings [7, 24, 35]. Li et al. [24] work on summarizing meetings with multiple participants, which is significantly different from single documents such as news articles or reviews because a statement from a meeting participant may differ from statements of others. These studies aim to summarize many utterances from different meeting participants into concise sentences considering contradictions between their utterances. Tokyo Metropolitan Assembly Minutes dataset (Tokyo minutes dataset) is similar to the data in the aspect that several people share their opinions or ask questions to other participants (examples are shown in Table 1). The most different facet of this minute dataset from other discussions is the importance of who said what. In the Tokyo minutes dataset, local politicians discuss on political topics, which are related to the management of the Tokyo metropolitan government. They are usually required to be consistent in their statements and to follow their parties' stance to a

ID	Utterance
1	平成二十三年第二回都議会定例会の開会に当たりまして、都政運営に対する所信の一端を申し述べ、都議会の皆様と都民の皆様のご理解、ご協力を得たいと思います。(Before beginning the 2nd regular session of Tokyo Metropolitan Assembly of 2011, I would like to express my opinion on the management of the Tokyo metropolitan government and I want the members of the Tokyo Metropolitan Assembly and the people living in Tokyo for cooperation.)
2	戦後、我々は、アメリカ依存の平和に安住しながら繁栄を謳歌し、かつてない物質的な豊かさと引きかえに、日本人としての価値の基軸を失ってまいりました。(After the war, we enjoyed our prosperity with the peace which depends on the U.S. In exchange for unprecedented material wealth, we have lost the foundation of our values as Japanese people.)

Table 1: Examples of utterance from the Tokyo Metropolitan Assembly Minutes dataset

given topic during a meeting. The consistency of these statements will influence how Tokyo citizens will choose the candidates in the following Tokyo assembly elections. Therefore, who said what is more important in Tokyo minutes dataset than in other types of usual meetings.

Kimura et al. [18] were first to work on unique aspects of Tokyo minutes dataset and proposed three tasks, i.e. Segmentation, Summarization, and Classification. They also organized QA Lab-PoliInfo task including those tasks during NTCIR-14 conference, and ask participants to solve them. In the Segmentation task, participants extract the relevant sentences from a set of assembly utterances related to one summary. In the Summarization task, its participants summarize utterances. This is the subtask we address in this paper. The purpose of the Classification task is to classify assembly utterances into three stances (pro, con, or neutral) for a given topic (e.g. "The Tsukiji Market should move to Toyosu."). These three tasks help to quickly grasp the content of assembly meetings. Summarization task will be more useful than other two tasks in order to promptly understand what politicians declare during a meeting because we can read a gist

Topic	避難所の自立電源確保せよ (Tokyo metropolitan government should ensure self-sustaining power supply for evacuation shelters)
Subtopic	避難所の自立電源 (Self-sustaining power for evacuation shelters)
Question Speaker	山口拓 (立憲・民主) (Taku Yamaguchi from Constitutional Democratic Party)
Answer Speaker	知事 (Tokyo Governor)
Question Summary	長期停電や平常時の温室効果ガス削減に役立つ電源確保の推進を。(We should promote securing power supplies to help reduce greenhouse gases during long-term power outages and during normal operation.)
Answer Summary	補正予算では、庁有車を災害時に非常用電源としても活用できる外部給電器の配備等を提案。引き続き非常時の電源確保を図る。(In the supplementary budget, we proposed the deployment of an external power supply unit that can be used as an emergency power source in the case of a disaster for agency-owned vehicles. We will continue to work to secure power sources for emergency situations.)

Table 2: Entry example from *Togikai-dayori*, a portal providing summaries of Tokyo Metropolitan Assembly

of their utterances at a glance. Therefore, in this paper, we propose a method to summarize assembly transcripts.

Tokyo Metropolitan Assembly provides summaries for each meeting via a portal named “*Togikai-dayori*” (Tokyo Metropolitan Assembly Newsletters). A summary originating from the site consists of a topic, a subtopic², an asker (Question Speaker), a replier (Answer Speaker), and short gists of questions and answers (see Table 2). Although reading all utterances is possible, we can easily grasp what politicians claim from these summaries. However, creating these summaries from utterances for each meeting is time-consuming and costly, thus there is a need to explore methods to automatically summarize them. In addition to approaches for summarizing utterances, we have to consider adapting summarization techniques for Tokyo minutes dataset to be applicable to transcripts originating from other regions because only a few prefectures create their own summaries. Therefore, we examine summarization methods without reference summaries. We create a pre-trained model for sentence embeddings, and compute a similarity between each utterance and a given topic and a similarity between each utterance and a subtopic. Then we calculate MMR-based function and extract utterances with the highest score.

In short, the main contributions of this article are:

- (1) Proposing Japanese Political Sentence-BERT;
- (2) Adapting an embedding-based unsupervised key-phrase extraction, EmbedRank++, to summarization.

¹You can access in <https://www.gikai.metro.tokyo.jp/newsletter/>.

²A topic is the main theme of a meeting, and a subtopic is one of issues for the meeting discussion within a topic.

2 RELATED WORK

2.1 Summarization

Almost all of standard non-neural summarization methods are based on extractive summarization [5, 12, 28, 33]. Although researchers have proposed many neural methods to text summarization, those non-neural ones are still used for comparison [22]. In particular, Maximal Marginal Relevance (MMR) [5], LexRank [12] and TextRank [28] are typically the most frequently used methods. MMR can be adapted to various methods, also in neural network-based methods because it is easy to be applied to score texts [2, 13, 21]. One of these algorithms, EmbedRank [2], is an unsupervised algorithm with sentence embeddings capable of extracting a key phrase from documents. In addition, these MMR-based methods are easy to adapt not only to single document summarization (SDS), but also to multi-document summarization (MDS).

There are also abstractive summarization techniques for MDS with small datasets [1, 21], which are recently based on deep neural networks [13, 14]. Most recently, language models dedicated to summarization have also been developed, for example BERTSUM, PEGASUS and MARGE [23, 26, 37]. However, Japanese language models for summarization do not exist to the best of the authors’ knowledge.

2.2 Sentence Embedding Method

Word embedding-based methods recently became common tools in NLP field. One of the most well-known algorithms, word2vec [29] is a pre-trained skip-gram or Continuous Bag-of-Words (CBOW) model to represent words as vectors. Inspired by word2vec, pre-trained word embedding models have followed, for example Global Vectors (GloVe) [31] and fastText [3]. In order to pre-train high quality representations, models need to capture semantics and different meanings the word can represent in different contexts. To address these aspects, Embedding from Language Models (ELMo) [32] has been proposed. This model uses word representations from a bi-directional LSTM (BiLSTM) trained with a language model (LM) on a massive data.

Following pre-trained word embedding models, sentence / document embedding ones have been developed. For example, Skip-Thought Vector [19] is based on a sentence encoder that predicts the surrounding sentences of a given sentence. Skip-Thought is based on an encoder-decoder model, its encoder maps words to a sentence vector and its decoder generates the surrounding sentences.

Contrasting with Skip-Thought, InferSent [10] is trained on labelled data, namely Stanford Natural Language Inference (SNLI) dataset [4], for the sentence embeddings.

Sent2Vec [30] is an unsupervised objective to train distributed representations of sentences. This model is inspired by CBOW, and it trains and infers sentence embeddings.

Doc2vec [20] is an algorithm based on Distributed Memory Model of Paragraph Vectors (PV-DM) or Distributed Bag of Words version of Paragraph Vector (PV-DBOW) to embed documents.

Universal Sentence Encoder (USE) [6] is a convolutional neural network-based method to embed sentences proposed by Cer et al. [6]. For their method they implement Deep Averaging Network (DAN) and Transformer-based models. In addition, they create multilingual USE models which cover 16 languages including Japanese [8, 36].

These above-mentioned sentence / document embedding models are available also in Japanese in addition to English. However, most of recent sentence embedding models do not yet support Japanese language. Sentence Transformers [34] are sentence embedding models for English texts based on siamese / triplet networks [15, 16]. Reimers and Gurevych [34] insert language models (BERT [11], RoBERTa [27], or XLM-RoBERTa [9]) into the network of siamese / triplet networks. They call BERT-based siamese / triplet networks as ‘‘Sentence-BERT.’’ Although Sentence-BERT is available in English, to the best of our knowledge, no such model exists for Japanese political texts.

In this research, we introduce Japanese Political Sentence-BERT, and we test it on Japanese political utterances in Tokyo minutes dataset.

3 JAPANESE POLITICAL SENTENCE-BERT

In this section, we describe how we create Japanese Political Sentence-BERT with the given Tokyo minutes dataset provided by the task organizers [17].

3.1 Data

In this section, we describe the data for pre-training Sentence-BERT to be used for Japanese political sentences.

We utilize utterances from Tokyo Metropolitan Assembly Minutes dataset (Tokyo minutes dataset) [17]. In the dataset, the utterances are listed chronologically (see Table 1 for the examples). The IDs refer to time order, so after the utterance ID_1 is spoken followed by the utterance ID_2. In addition to the utterances, there is some other information, for example the date, or the names of speakers.

To utilize this dataset for creating Japanese Sentence-BERT, we assumed that sentences that are adjacent to each other in terms of time series are semantically related. We first compute an embedding of each utterance using Japanese NLP library, GiNZA³. With these sentence embeddings, we calculate the cosine similarity between an utterance (which we call a ‘‘target sentence’’) and the utterance that follows it. When the cosine similarity is 0.5 or more and the speaker is the same, we regard a given pair as ‘‘related utterances’’ and define the extracted utterance after the target sentence as a ‘‘positive sentence.’’

In the next step, we retrieve a non-similar utterance following the above-mentioned target sentence from the data. We extract a non-similar utterance with the following steps:

- (1) take an utterance which is spoken on a different day or in a different meeting;
- (2) compute the sentence embedding of the extracted utterance with GiNZA;
- (3) calculate cosine similarity between the target sentence and the extracted utterance;
- (4) repeat steps 1-3 until the algorithm finds an utterance whose cosine similarity with the target sentence is 0.9 or less.

After the fourth step, we finally extract one utterance which is dissimilar to the target sentence, and define it as a ‘‘negative sentence.’’ Note that thresholds (0.9 and 0.5) for sentence extraction steps are manually decided.

³<https://github.com/megagonlabs/ginza>.

triplet	Utterance
Target	このため、都は、用地取得の体制強化や都用地の活用など、一歩踏み込んだ支援メニューを示しまして、区の取り組みを促してまいりました。(Therefore, the Tokyo Metropolitan Government has presented solutions, such as strengthening the system for land acquisition and utilizing metropolitan-owned land, in order to encourage the city’s efforts.)
Positive	この結果、コア事業に位置つけた大田区糺谷駅前地区の再開発の工事が始まり、荒川区の都営住宅跡地において、生活道路整備のための従前居住者住宅の建設などが進んでおります。(As a result, work has begun on the redevelopment of the Kojiya-mae station area in Ota Ward, which was positioned as a core project, and the construction of former residents’ houses on the site of a former municipal housing complex in Arakawa Ward is under construction in order to improve roads for daily life.)
Negative	次に、ハッ場ダムの治水効果でございますが、ただいまお答えした八つの洪水について、日本学術会議が妥当と判断した計算モデルを採用して、ハッ場ダムによる効果量を試算しております。(Next, regarding the flood control effects of the Yamba Dam, we estimated the effects of this dam for the eight flooding events mentioned above by using a simulation model determined to be appropriate by the Science Council of Japan.)

Table 3: An example of [target sentence, positive sentence, negative sentence] triplet from Tokyo Metropolitan Assembly Minutes dataset

With these three types of sentences, we create a triplet, i.e. [target sentence, positive sentence, negative sentence] (see an example shown in Table 3). In the end, 27,078 triplets of [target sentence, positive sentence, negative sentence] were created. These triplets are divided into 21,662 (80%) for training, 2,708 (10%) for development, and 2,708 (10%) for testing.

3.2 Model

The pooling operation is added to Sentence-BERT model after the output from BERT in order to derive a fixed sized sentence embedding. In our model, we adopt MEAN pooling, not MAX or CLS, because Reimers and Gurevych noticed that MEAN pooling is better than two other methods [34]. In order to fine-tune BERT on our dataset, we create a triplet network [16] to update the weights that make the computed utterance embeddings more semantically meaningful and the difference between a cosine similarity of a target sentence and a positive sentence and a cosine similarity of a target sentence and negative sentence becomes larger.

For a triplet network, the triplet loss function is defined. Assuming we have three sentences; a target sentence s_t , a positive sentence s_p ,

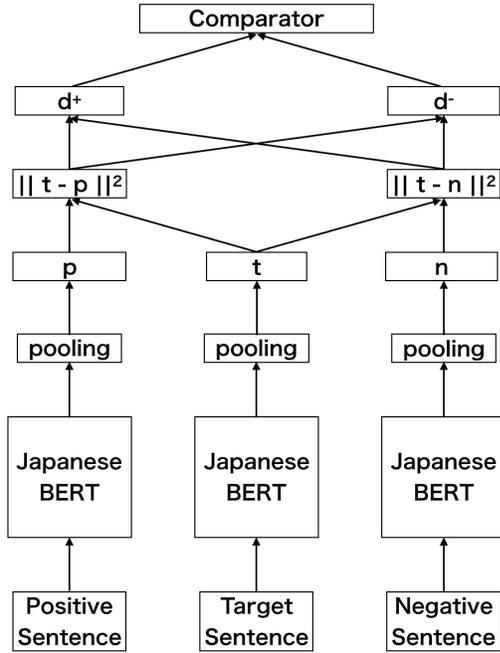


Figure 1: Creating Japanese Political Sentence-BERT with the triplet loss function

and a negative sentence s_n , the triplet loss function tunes a triplet network to maximize the difference between d_+ and d_- which are defined with Equation (1):

$$d_+ := \frac{\|e^{f s_t - f s_p}\|_2}{\|e^{f s_t - f s_p}\|_2 + \|e^{f s_t - f s_n}\|_2}, \quad (1)$$

$$d_- := \frac{\|e^{f s_t - f s_n}\|_2}{\|e^{f s_t - f s_p}\|_2 + \|e^{f s_t - f s_n}\|_2},$$

where f is an embedding function. The triplet loss is defined by the following Equation (2):

$$L(d_+, d_-) = \|d_+, d_- - 1\|_2^2 = \text{const} * d^2. \quad (2)$$

We normalize to make $d_+ + d_- = 1$, so d_- should be 1 when $d_+ = 0$. Therefore, the triplet loss can be also described by Equation (3):

$$L(s_t, s_p, s_n) = d(f(s_t), f(s_p)) + (1 - d(f(s_t) - f(s_n))) = 2d(f(s_t) - f(s_p)), \quad (3)$$

where d is Euclidean distance function. This Equation (3) shows that d_+ becomes minimized when d_- is maximized.

In our model, we insert the pre-trained Japanese BERT⁴ into the branches of the triplet network, and perform pre-training the triplet network on the triplet data described in Section 3.1. Again, we regard a target sentence as s_t , a positive sentence as s_p , and a

⁴We use the BERT-base model (12-layer, 768-hidden, 12-heads, and 110M parameters) with Whole Word Masking released by Tohoku University. The model is available at <https://github.com/huggingface/transformers>.

negative sentence as s_n . We set the margin for triplet loss ϵ to 1 following the Sentence-BERT model for English [34].

The overview of creating Japanese Sentence-BERT is shown in Figure 1. The ‘‘Comparator’’ in Figure 1 refers to the comparing function to identify which sentence, s_p or s_n , is positive / negative.

3.3 Training Details

We train Sentence-BERT on the triplet data as described in Section 3.1. As mentioned before, our data consists of 27,078 triplets ([target sentence, positive sentence, negative sentence]). These triplets are divided into 21,662 (80%) for the train data, 2,708 (10%) for the development data, and 2,708 (10%) for the test data. We use a batch size of 16, Adam optimizer with learning rate $2e-5$, and a linear learning rate warm up over 10% of the train data as in the Sentence-BERT model for English [34]. We train a Japanese triplet BERT network for three epochs with the above-mentioned hyper-parameters. The pooling strategy is MEAN.

Our dataset we train Sentence-BERT on comes from Tokyo Metropolitan Assembly Minutes dataset consisting of political utterances. Therefore, the acquired Sentence-BERT model specializes in political related sentences, hence we name the model ‘‘Japanese Political Sentence-BERT.’’

Note that our training details in the most part follows Sentence Transformers algorithms⁵ except for the epoch number.

3.4 Comparing Method and Metrics

We utilize Universal Sentence Encoder (USE) [6] for the comparison with our Japanese Political Sentence-BERT for a sentence embedding function f . USE is one of the most common methods to embed Japanese sentences into vectors as mentioned in Section 2.2. To show the usability of our Sentence-BERT model, we perform tests with both algorithms. The evaluation steps are as follows:

- (1) embed the sentences in a triplet of [target sentence, positive sentence, negative sentence] in test data with the BERT modules inside the triplet network;
- (2) compute cosine similarities between [target sentence, positive sentence] and [target sentence, negative sentence] from sentence embeddings;
- (3) calculate the difference between two acquired cosine similarities of [target sentence, positive sentence] and [target sentence, negative sentence], and evaluate the accuracy by counting how many of these differences are equal to zero or larger.

We define the difference between two acquired cosine similarities of [target sentence, positive sentence] and [target sentence, negative sentence] as $diff$, and $diff$ can be represented as Equation (4):

$$diff = \text{CosSim}(f(s_t), f(s_p)) - \text{CosSim}(f(s_t), f(s_n)). \quad (4)$$

We evaluate models also with a $diff$ function in addition to accuracy. The larger $diff$ indicates that the model can identify a positive sentence and a negative sentence better.

⁵Sentence Transformers are available in <https://github.com/UKPLab/sentence-transformers>. The version of the library we utilize is 0.3.2.

	<i>diff</i>	Accuracy
USE	0.2441	0.8674
JPSB (Proposed Model)	0.3705	0.9849

Table 4: Results of USE and our proposed Japanese Political Sentence-BERT (JPSB) evaluated on the test data

The USE model which we use for a sentence embedding function f is the Universal Sentence Encoder Multilingual Large module⁶ [8, 36]. USE model uses a Convolutional Neural Network-based approach to embed sentences, and covers 16 languages including Japanese. The length of its inputted text to USE is variable, and the dimension of an outputted sentence vector is 512.

3.5 Evaluation Results

This section compares the methods evaluated on the test data, and shows the results in Table 4. Japanese Political Sentence-BERT outperforms the existing sentence embedding method (USE) by approximately 0.12 in accuracy. Regarding *diff*, Japanese Political Sentence-BERT exceeds USE by around 0.13. Both of these results show that our Sentence-BERT model represents a sentence embedding more effectively than the existing sentence embedding models.

3.6 Considerations about Japanese Political Sentence-BERT

To the best of authors' knowledge, this is the first attempt to implement Sentence-BERT for Japanese language. Our model specializes in political sentences, which are the core of the QA Lab-PoliInfo-2 tasks at NTCIR-15 evaluation conference. Japanese Political Sentence-BERT outperforms the USE approach by around 0.12 in accuracy and by around 0.13 in *diff*. From this difference, it is sufficient to say our model represents sentence embeddings better than other existing models.

As the future work, we plan to create Japanese Sentence-BERT for general texts trained on Wikipedia data, and to examine the usefulness of Japanese Sentence-BERT and Japanese Political Sentence-BERT for other QA Lab-PoliInfo-2 subtasks, i.e. the stance classification task and the entity linking task.

4 SUMMARIZATION SUBTASK

In this section, we describe the summarization subtask in QA Lab-PoliInfo-2 and results of our attempt to summarize utterances in Tokyo Metropolitan Assembly Minutes dataset with Japanese Political Sentence-BERT described in Section 3.

4.1 Data

We experiment with the test data provided by the QA Lab-PoliInfo-2 task organizers [17] for the summarization task. In our method, we do not use training data provided by the task organizers. As shown in the test data example in Table 8, there is some additional information regarding one question. Each question must have at least

⁶The model is available in <https://tfhub.dev/google/universal-sentence-encoder-multilingual-large/1>.

Answer ending line	0
Answer length	150
Answer speaker	知事 (Tokyo Governer)
Answer starting line	0
Answer summary	None
Date	2013-02-26
ID	PoliInfo2-DialogSummarization-JA-Formal-Test-00001
Main topic	新知事の東京の将来像を示せ エネルギー需要側の政策進化を (Show us the future goals for Tokyo which you, the new governor, have. You should improve the policy from the perspective of the energy demanding side.)
Subtopic	都政運営 (Administration of Tokyo Metropolitan Government)
Meeting	平成25年第1回定例会 (the 1st regular session in the year 2013)
Prefecture	東京都 (Tokyo)
Question ending line	0
Question length	100
Question speaker	酒井大史 (民主党) (Daishi Sakai from Democratic Party)
Question starting line	0
Question summary	None

Table 5: Example of the inputted test data for the summarization task

one answer. The important data components for our approach to the summarization task (see Section 4.3) are:

- (1) **Answer length** showing the number of character limits of the answer summary,
- (2) **Answer Speaker** referring to the name of participant who answered to a given question,
- (3) **Main topic** being a theme for the meeting,
- (4) **Subtopic** being a category for the utterance spoken during the meeting,
- (5) **Question length** showing the number of character limits of the question summary,
- (6) **Question Speaker** referring to the name of participant who asked a question.

Both numbers of character limits are decided by the task organizers. In the test data, 254 questions are included.

4.2 Evaluation Metrics

For evaluation, we used two summary evaluation metrics: recall score in ROUGE-1 [25] for both methods (described in Section 4.3.1), and human evaluation for the Japanese Political Sentence-BERT-based model. ROUGE is a recall-based metric for fixed-length summaries which is based on n-gram co-occurrence. Regarding human evaluation, the task organizers evaluate our results with the following metrics:

- **Content**: evaluates output semantically. The range of scores are $[X, 0, 1, 2]$, and the larger score, the better summary. Note that the label X means the output summary is a concise statement which summarizes utterances, but differs from the reference summary. The organizers evaluate this metric with two scores: $X=2$ (when the regarding the different content summaries as correct) and $X=0$ (regarding the different content summaries as incorrect).
- **Well-formed**: evaluates the naturalness of an output. The range of scores is $[0, 1, 2]$, and the larger score, the better summary;
- **Non-twisted**: evaluates whether the outputted texts express the speaker’s intention without distorting it. The range of scores is $[0, 1, 2]$, and the larger score, the better summary;
- **Sentence goodness**: evaluates whether an outputted text expresses a concise summary or not. The range of scores is $[0, 1, 2]$, and the larger score, the better summary;
- **Dialog goodness**: evaluates whether outputted texts express a concise summary for a set of questions and answers. The range of scores is $[0, 1, 2]$, and again the larger score, the better summary.

The human evaluation score for all summaries is calculated with the sum of scores and their averages. Each summary is scored by two of the summarization task participants chosen by the QA Lab-PoliInfo-2 organizers.

4.3 Methods

This section explains our approach for summarizing utterances with sentence embeddings. We describe the summarization method in Section 4.3.1, and show two sentence embedding models which are used in the summarization method described in Section 4.3.2.

4.3.1 Our Proposed Method. We propose the sentence embedding model-based approach for the utterance summarization which is described below. The sentence embedding models which we use for the method are described in Section 4.3.2.

We introduce a MMR-based extractive method, i.e. EmbedRank [2], to our algorithm. EmbedRank is an unsupervised method with sentence embeddings for key word extraction. Its creators also proposed EmbedRank++ in order to avoid extracting key phrases of the same meaning and acquire different words. Both algorithms differ from traditional methods of key phrase extraction based on graph representations [12, 28], because they utilize sentence embeddings for extracting phrases similar to documents. We adapt EmbedRank++ to extract sentences. Moreover, to obtain more accurate sentence embeddings, which are described in Section 4.3.2, we use our Japanese Political Sentence-BERT model and Universal Sentence Encoder (USE) model, which are described in Section 4.3.2, instead of Doc2vec [20] and Sent2Vec [30] which were used for English texts [2].

In EmbedRank++, there are two parameters; *key size* and λ . *Key size* defines the number of extracted key phrases. When *key size* equals 1, only one key phrase is extracted by EmbedRank++. λ is a parameter in the equation for Maximal Marginal Relevance (MMR) [5] which is introduced in EmbedRank++, and controls the diversity of extracted words and the relevance between documents and extracted words. With a given input query Q , the set S represents

documents that are selected as correct answers for Q . S is populated by computing MMR as described in Equation (5),

$$MMR := \arg \max_{D_i \in R \setminus S} \{ \lambda * Sim_1(D_i, Q) - (1 - \lambda) \max_{D_j \in S} Sim_2(D_i, D_j) \}, \quad (5)$$

where R is the ranked list of documents retrieved by an algorithm, S represents the subset of documents in R which are already selected, D_i and D_j are retrieved documents, Q is the averaged vector of inputted all documents and Sim_1 and Sim_2 are similarity functions. Following the existing research on MMR [5], we set λ to 0.5. Using cosine similarity for the similarity functions, we compute MMR score between D_i and Q and between D_i and D_j .

For the summarization task, we introduce two functions in addition to two terms in Equation (5). One is the cosine similarity between D_i and a given main topic MT , and the other is also the cosine similarity between D_i and a given subtopic ST . MMR score for our method is calculated with the following Equation (6):

$$MMR = \arg \max_{D_i \in R \setminus S} \{ k * \{ 0.5 * CosSim(D_i, Q) - 0.5 * \max_{D_j \in S} CosSim(D_i, D_j) \} + m * CosSim(D_i, MT) + s * CosSim(D_i, ST) \}, \quad (6)$$

where k , m and s are parameters to assign weight to each similarity. In order to obtain MMR score, Bennani-Smires et al. [2] computed embeddings of documents with Doc2vec and Sent2Vec, and we adapt Japanese Political Sentence-BERT and USE in this summarization task.

The parameter *key size* is set up accordingly to the following Equation (7):

$$\begin{aligned} key\ size\ (Answer) &:= \frac{Answer\ length}{50}, \\ key\ size\ (Question) &:= \frac{Question\ length}{50}. \end{aligned} \quad (7)$$

The reason why we divide the lengths by 50 is that we assume the 50 characters may be the maximum length to express one content. Other parameters k , m and s are set to 0.2, 0.3, and 0.5, respectively. These three parameter settings are decided to bring the total to 1.

4.3.2 Sentence Embeddings for the proposed method. We utilize the Japanese Political Sentence-BERT described in the Section 3 to the sentence embedding model in our above-mentioned approach (see Section 4.3.1). We also adopt Universal Sentence Encoder (USE) as the comparison method, which is one of the common sentence embedding models and is available for Japanese language. We examine sentence embeddings with both models, and summarize utterances using the method described in Section 4.3.1.

4.4 Evaluation Results

Table 6 shows recall of ROUGE-1 scores, and Table 7 shows human evaluation results. Note that the manual evaluation is conducted by the task organizers, so there are no results of USE-based approach.

	ROUGE-1
USE-based EmbedRank++	0.0846
Japanese Political Sentence-BERT-based EmbedRank++	0.0696

Table 6: Results of USE-based and Japanese Political Sentence-BERT-based EmbedRank++ tested on the test data

Human Evaluation		Our Model	
num of summaries		533	
Content	(X=2)	sum	184.5
		average	0.346
	(X=0)	sum	135.5
		average	0.254
Well-formed		sum	888.5
		average	1.667
Non-twisted	all	sum	184.5
		average	0.346
	evaluable	sum	135.5
		average	0.254
Sentence goodness		sum	151.5
		average	0.284
Dialog goodness		sum	43.0
		average	0.169

Table 7: Human evaluation results of Japanese Political Sentence-BERT-based EmbedRank++ tested on the test data

Regarding ROUGE-1 scores, Sentence-BERT-based approach does not outperform USE-based one. Except well-formed scores in the human evaluation metric, our method does not perform well when compared to other task participants (all results are shown in the task overview paper [17]).

4.5 Discussion for the Summarization Subtask

From these results, we can find that our method does not improve the scores. Furthermore, our Sentence-BERT model does not contribute to extracting accurate utterances for summarizing assembly minutes. One of the main reasons why the method does not achieve high results is that we have not tuned k , m and s parameters in our model. We need to tune these three parameters with the train data provided by the task organizers, and acquire the set of parameters to maximize recall in ROUGE-1 score in the near future. Considering the output, shown in the Table 8, the generated summary #1 expresses the theme and contents similar to the reference summary. On the other hand, the summary #2 refers to a totally different theme from the reference summary. This means that Equation (6) could not keep the higher score to the more related sentence within a given theme well. We need to change parameters in Equation (6) in order to increase the parameters for a similarity between a sentence and a given main topic and a similarity between a sentence and a given subtopic.

Reference Summary #1	都の産業政策やインフラ整備との一体的取組、川崎港・横浜港との連携に加え、都が責任を持って経営に関わる体制を確保。(We should ensure that the Tokyo Metropolitan Government will take responsibility for management of the Port of Kawasaki and the Port of Yokohama, in addition to coordinated efforts with Tokyo’s industrial policies, infrastructure development and cooperation with the ports of Kawasaki and Yokohama.)
Output #1	東京の産業政策やインフラ整備と一体的に取り組むことが求められております。/現場の実態を熟知した東京都が責任を持って港湾経営にかかわっていく体制を確保してまいります。(At the same time, we need to work on Tokyo’s industrial policies and infrastructure development. We should ensure that the Tokyo Metropolitan Government, which knows well the actual situation in the field, will take responsibility for port management.)
Answer Summary #2	都道について新たな整備目標示す。区市町村支援、国への財源拡充要求、電線事業者等との連携を強化し積極的に推進する。(Let us explain the new purposes for Tokyo metropolitan roads. We will actively promote support for municipalities, also demand expansion of financial resources from the government, and strengthen cooperation with wire-line operators.)
Output #2	一層の支援に努めてまいります。/風格ある都市景観の形成と高度防災都市の実現を目指し世界に誇れる都市空間を創出してまいります。(We will make every effort to support the city. We will create a world-class urban space with the aim of creating a distinctive urban landscape and a highly disaster-resistant city.)

Table 8: Example of the outputted summaries for the summarization subtask

Regarding the comparison of USE-based approach and Sentence-BERT-based one, it cannot be said that our pre-trained Japanese Political Sentence-BERT performs well in a downstream task, i.e. a summarization task. Our model requires changing the hyper-parameters and the data. We plan to implement Sentence-BERT modules trained on news corpus for political domain tasks. In addition, for general NLP tasks, we will also train our model on Japanese Wikipedia.

There are some cases where the outputted texts were meaningless as a summary. For example, “(拍手)” (annotation for hand clapping) or “-” were found in the output. We need to delete such noise in the pre-processing.

5 CONCLUSIONS

Our work was the first attempt to create Sentence-BERT for Japanese texts, and to adopt it to utterances included in the Japanese minute data summarization task. This model outperformed the existing sentence embedding model USE by around 0.12 in accuracy. Regarding the utterance summarization, we adopted MMR-based unsupervised approach, EmbedRank++, for utterance summarization, and our Japanese Sentence-BERT for sentence embeddings. As comparison

with the Sentence-BERT, we also utilized USE as a sentence embedding model for EmbedRank++. Additionally, we changed the MMR function for EmbedRank++ and added two cosine similarity functions: one was a similarity between a given main topic and an utterance, and the other was a similarity between a given subtopic and an utterance. Our method succeeded to output natural sentences for summaries. However, in spite of algorithm improvements, we discovered that our EmbedRank++ method based on Japanese Political Sentence-BERT without parameter tuning does not contribute to generating accurate texts for summaries of utterances. In the future, we plan to change parameters during the training, and use the parameters to acquire better results. Regarding Sentence-BERT, our model did not perform well in this subtask, so we plan to pre-train it with other data, for example with news corpus or Wikipedia.

REFERENCES

- [1] Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-Document Abstractive Summarization Using ILP based Multi-Sentence Compression. *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)* (2015).
- [2] Kamil Bennani-Smires, Claudiu Musat, Andreea Hosmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple Unsupervised Keyphrase Extraction using Sentence Embeddings. *Proceedings of the 22nd Conference on Computational Natural Language Learning* (2018), 221–229.
- [3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.
- [4] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).
- [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*, 335–336.
- [6] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 169–174.
- [7] Ping Chen, Fei Wu, Tong Wang, and Wei Ding. 2017. A semantic QA-based approach for text summarization evaluation. *arXiv preprint arXiv:1704.06259* (2017).
- [8] Muthu Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yunhsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Learning Cross-Lingual Sentence Representations via a Multi-task Dual-Encoder Model. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, 250–259.
- [9] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116* (2019).
- [10] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 670–680. <https://www.aclweb.org/anthology/D17-1070>
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [12] Günes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* 22 (2004), 457–479.
- [13] Alexander R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749* (2019).
- [14] Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-Up Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4098–4109.
- [15] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. IEEE, 1735–1742.
- [16] Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*. Springer, 84–92.
- [17] Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ootake, Yuzu Uchida, Keiichi Takamaru, Madoka Ishioroshi, Teruko Mitamura, Masaharu Yoshioka, Tomoyoshi Akiba, Yasuhiro Ogawa, Minoru Sasaki, Kenichi Yokote, Tatsunori Mori, Kenji Araki, Satoshi Sekine, and Noriko Kando. 2020. Overview of the NTCIR-15 QA Lab-PoliInfo-2 Task. *Proceedings of The 15th NTCIR Conference* (12 2020).
- [18] Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ootake, Yuzu Uchida, Keiichi Takamaru, Kotaro Sakamoto, Madoka Ishioroshi, Teruko Mitamura, Noriko Kando, Tatsunori Mori, et al. [n.d.]. Overview of the NTCIR-14 QA Lab-PoliInfo task.
- [19] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, 3294–3302.
- [20] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, 1188–1196.
- [21] Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the Neural Encoder-Decoder Framework from Single to Multi-Document Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4131–4141.
- [22] Daniel Lee, Rakesh Verma, Avisha Das, and Arjun Mukherjee. 2020. Experiments in Extractive Summarization: Integer Linear Programming, Term/Sentence Scoring, and Title-driven Models. *arXiv:2008.00140* (2020).
- [23] Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via Paraphrasing. *arXiv preprint arXiv:2006.15020* (2020).
- [24] Manling Li, Lingyu Zhang, Heng Ji, and Richard J Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2190–2196.
- [25] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.
- [26] Yang Liu. 2019. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318* (2019).
- [27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
- [28] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 404–411.
- [29] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
- [30] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*.
- [31] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
- [32] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.
- [33] Dragomir R Radev, Hongyan Jing, Małgorzata Stys, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management* 40, 6 (2004), 919–938.
- [34] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. <http://arxiv.org/abs/1908.10084>
- [35] Suzan Verberne, Emiel Kraemer, Sander Wubben, and Antal van den Bosch. 2020. Query-based summarization of discussion threads. *Natural Language Engineering* 26, 1 (2020), 3–29.
- [36] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2019. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307* (2019).
- [37] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777* (2019).