Forst: A Challenge to the NTCIR-15 QA Lab-PoliInfo-2 Task

Hiromu Onogi
Yokohama National University, Japan
onogi-hiromu-kb@ynu.jp

Kiichi Kondo
Yokohama National University, Japan
kondo-kiichi-jc@ynu.jp

Younghun Lim
Yokohama National University, Japan
lim-younghun-vs@ynu.jp

Xinnan Shen
Yokohama National University, Japan
shen-xinnan-vk@ynu.jp

Madoka Ishioroshi
National Institute of Informatics, Japan
ishioroshi@nii.ac.jp

Hideyuki Shibuki
National Institute of Informatics, Japan
shib@nii.ac.jp

Tatsunori Mori
Yokohama National University, Japan
mori@forest.eis.ynu.ac.jp

Noriko Kando
National Institute of Informatics, Japan
SOKENDAI, Japan
kando@nii.ac.jp

ABSTRACT

In this paper, we describe the development of a system for stance classification, two systems for dialog summarization and a system for entity linking. We submitted 5 results including 3 late submissions for the stance classification, 10 results including 5 late submissions for the dialog summarization and 4 results for the entity linking. As a result, an accuracy of .9388 for the stance classification, a ROUGE-1 score of .2410 for the dialog summarization and an F-measure of .3910 for the entity linking were obtained.

TEAM NAME

Forst

SUBTASK

Stance Classification (Japanese), Dialog Summarization (Japanese), Entity Linking (Japanese)

1 INTRODUCTION

We tackled the stance classification, dialog summarization, and entity linking subtasks in the NTCIR-15 QA Lab-PoliInfo-2 task[9]. In this paper, we describe the development of a system for stance classification, two systems for dialog summarization, and a system for entity linking. Section 2 describes the stance classification system and results. Sections 3 and 4 describes the dialog summarization systems and results, respectively. Section 5 describes the entity linking system and results. Finally, Section 6 provides some concluding remarks.

2 STANCE CLASSIFICATION

2.1 Approach

We applied a rule-based approach using four rules on the proceedings and questions, and then conducted a dependency analysis on the sentences that satisfied the conditions, and obtained the output on the pros and cons. The first condition is whether the date is valid. The second condition is whether the speaker is valid. Here, we created and used a dictionary containing phrases that are commonly used by members of each parliamentary group to express their pros and cons to the bill. The third condition is whether the sentence mentions the pros and cons. The fourth condition is whether the statement mentions a bill for which we want to check the pros and cons. Here, we created and used a dictionary containing phrases such as "all bills" and "other bills" that are related to bills in general but do not refer to a specific example. We use CaboCha[17] for the dependency analysis.

2.2 Related Work

Inoue et al.[6] proposed a method for extracting the favorable and unfavorable opinions from the Web to specific topics such as products and current affairs. From each web page, the method extracts opinions that include both topical phrases and key phrases expressing such pros or cons as "sansei-suru" or "hantai-suru." We applied this technique to the Stance Classification task. Sakaji et al.[15] collected local political corpora from the records of local councils and extracted statements expressing opinions and intentions. They proposed a method for extracting statements expressing intention by using bootstrapping. Nishimura et al.[14] proposed a method to extract and organize the opinions of specific people from newspaper articles. In this research, sentences describing which subjects are members of parliament or a political party are extracted because they may express opinions. We applied this method, focusing on the subject ' s expression and predicting the parliament or political party to which each speaker belongs. The above is the related research that we referenced. In addition, we introduced a method to determine whether the date of each statement is appropriate, as well as a dictionary containing phrases that are commonly used by members of each parliamentary group to express their pros and cons to a bill and a dictionary containing phrases that are related to bills but do not refer to specific examples.

2.3 Method

2.3.1 preparation. None of the values in SpeakerList, ProsConsListBinary, or ProsConsListTernary of the given question file could be parsed by the Json parser used because expressions that should appear as values appear as keys. To solve this, we converted SpeakerList into a standard JSON style by introducing unified keys, as shown in Figure 2. In the same way, ProsConsPartyListBinary and ProsConsPartyListTernary are converted into JSON data, as shown in Figures 3 and Figure 4, respectively. After parsing, for all questions, we changed the value in ProsConsListBinary to those for which the key is binary, and the value in ProsConsListTernary without mentioning which key is ternary.

2.3.2 Checking the validity of dates. In this process, the StartDate and EndDate values are obtained from the parsed question file, and the Date value is obtained from the parsed proceedings file. This process uses the proceedings of one meeting as one unit. If the date is valid, the process proceeds to the checking process, described in Section 2.3.3, and if not, the check of the proceedings is terminated. The validity of date is defined as follows: if the date obtained from the target proceedings is between StartDate and EndDate for a question, the date is valid for the question; otherwise, it is invalid. If the statement is not within the range of the date indicated in the question, it is meaningless as a reference to the pros and cons.

2.3.3 Checking the validity of the speaker. This process is divided into two steps. The input of the first step is the value of the Member in the SpeakerList obtained from the parsed question file, the value

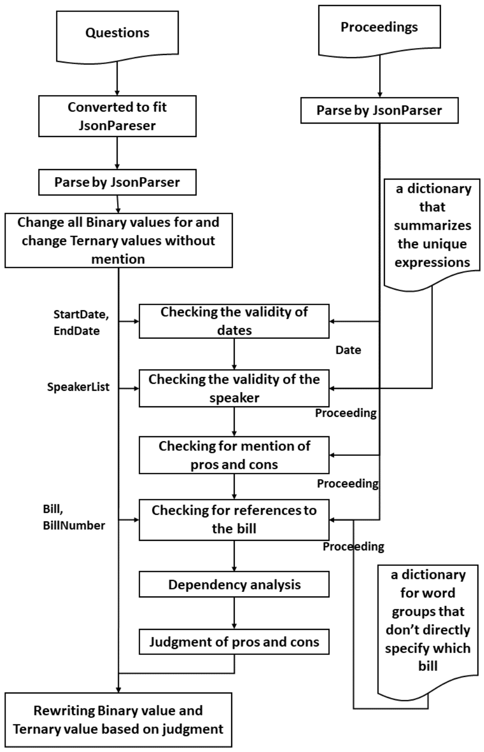

Figure 1: Stance classification pipeline

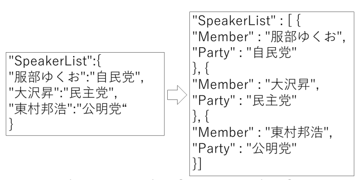

Figure 2: SpeakerList before and after conversion

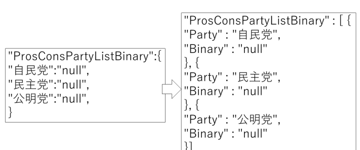

Figure 3: ProsConsPartyListBinary before and after conversion

Figure 4: ProsConsPartyListTernary before and after conversion

of Speaker in the Proceeding obtained from the parsed proceedings file, and the proceedings validated in the process described in Section 2.3.2. The input of the second step is the result of the first step and a dictionary containing phrases that are commonly used by members of each parliamentary group to express their pros and cons to the bill. This process uses an utterance as one unit. If the individual is judged to be appropriate as a speaker in the first step of processing, the process proceeds to the checking process described in Section 2.3.4, and if it is judged to be invalid, it transitions to the second step processing. If it is judged to be

party	Specific representation 1	Specific representation 2
自民党	都議会自由民主党を代表し	
民主党	都議会民主党を代表し	
日本共産党	日本共産党議団を代表し	
公明党	都議会公明党を代表し	
ネット	都議会生活者ネットワーク・みらい	都議会生活者ネットワークを代表し
民進党	都議会民進党を代表し	
民進都議団	民進党都議団を代表し	
かがやけ	かがやけＴｏｋｙｏを代表し	
東京維新	東京維新の会を代表し	
都ファースト	都民ファーストの会東京都議団を代表し	
東京改革	東京改革議員団を代表し	

Figure 5: A dictionary for unique expressions

appropriate as a speaker in the second step of processing, the process proceeds to the checking process described in Section 2.3.4, and if it is judged to be invalid, the check of this statement is terminated. In the first step of the processing, a check is applied to determine whether the speaker indicated by the value of Member in the SpeakerList of each question matches the speaker indicated by the value of Speaker associated with each statement in the proceedings for which the statement is to be checked. This checking process is required because the speaker must be a member of each group associated with the question to give an opinion regarding the pros and cons of the bill. However, there are certain cases in which it fails to extract the information of the pros and cons of the parliamentary group because of the mismatch between names in the member value and those in the speaker value. For this reason, we add a rule. As the condition, if the statement includes the named entity used by each member to state the pros and cons of the bill, it is judged that the entity is a valid speaker stating the pros and cons of the bill. In this second step of processing, if even one of these phrases that are commonly used by members of each parliamentary group is included in the statement using the dictionary shown in Figure 5, it is determined that the speaker is appropriate.

2.3.4 Checking for mentions of pros and cons. The input for this process is the Utterance value in the Proceeding obtained from the parsed proceedings and utterancess validated in the process described in Section 2.3.3. This process uses an utterance as one unit. If a statement has a reference to the pros or cons, the process proceeds to the checking process described in Section 2.3.5, and if not, the process of checking the statement is terminated. Inclusion of a reference to the pros or cons in a statement is judged according to appearance of key phrases such as "sansei (agree)" and "hantai (disagree)" in the utterance string.

2.3.5 Checking for references to the bill. The input of this process is the Bill value and BillNumber value obtained from the parsed question file, the Utterance value in the proceedings obtained from the parsed proceedings, and a dictionary containing phrases that are related to bills without references to specific bills or utterances validated during the process described in section 2.3.4. This process uses an utterance as one unit. If it is determined that there is a reference to the bill, the character string of the target utterance is used as a sentence for a dependency analysis, and if it is determined that there is no reference, the check for the target utterance is terminated. Inclusion of a reference to the bill in Utterance value is judged according to the appearance of the value of Bill or Bill Number. Even if it is not included, if any one of the words in the dictionary is included in the Utterance value, it is judged that there a reference to the bill. The dictionary contains "全ての議案 (all bills)", "すべての議案 (all bills)", "全議案 (all bills)", "他の議案 (other bills)" and "ほかの議案 (other bills)".

2.3.6 Dependency analysis. A dependency analysis was conducted using CaboCha for an utterance that meets all conditions up to Section 2.3.5. After conducting a dependency analysis, we found the word indicating the bill and the word "sansei (agree)" or "hantai (disagree)". If the word found is "sansei (agree)", we set the value of Binary in the corresponding ProsConsListBinary to "agree" and the value of ProsConsListTernary to "agree". If the word found is

Table 1: The results of stance classification

	Forst(ID164)	Forst(ID171)	Forst(ID232)	Forst(ID234)
Binary classification	0.9382	0.9388	0.9391	0.9408

"hantai (disagree)", we set the value of Binary in the corresponding ProsConsListBinary to "disagree" and the value of ProsConsListTernary to "disagree". If neither the word "sansei (agree)" nor the word "hantai (disagree)" is involved, the sentence moves to the judgment of the next sentence because a clue was not given.

2.4 Result

The results of this system are shown in table 1. In submission ID164, this system adapted a check of the validity of the dates, a check of the validity of the speaker without a dictionary for the unique expressions, a check for mentions of the pros and cons, and a check for references to the bill based only on whether the statement includes the Bill value. In submission ID171, in addition to the system indicated by ID164, it is possible to judge whether there is a reference to the bill based on whether the statement contains the BillNumber value. In submission ID232, in addition to the system indicated by ID171, a dictionary for inclusion in the utterance value was applied to check for references to the bill. In submission ID234, in addition to the system indicated by ID232, a dictionary for the unique expressions was applied to check the validity of the speaker.

2.5 Discussion

This system cannot analyze the pros and cons of expressions such as "I agree with the 15th bill and 20 other bills." It is therefore difficult to judge the pros and cons of all 20 bills. This is because this sentence alone does not tell what the "other 20 bills" indicate because we have not applied an anaphora resolution. However, if each party has expressed its opinion on the bill, it will be possible to predict the pros and cons from what type of opinion the party has. For this reason, it is necessary to extract the remarks that are considered to express opinions on the relevant bill from the remarks of the members belonging to each political party, and to judge the pros and cons. In addition, we would like to proceed with research on the pros and cons of the judgment method based on this rule-based approach. In the future, we would also like to consider methods based on machine learning, such as an SVM, to improve the accuracy and compare them with rule-based methods.

3 DIALOG SUMMARIZATION A

3.1 Approach

The dialog summarization task involves summarizing the questions and answers recorded in local council proceedings. The goal is to summarize them, taking into account the structure of the dialogue between the assemblymen's questions and the governor's answers. We worked on this task using an original system based on an extractive summarization. Mainstream extractive summarization systems use machine-learned models of minutes to calculate the importance of candidate extraction sentences. However, we took a different approach. We assume that the representative sentence of a passage is similar to the whole passage. As a measure of similarity, we adopt the cosine similarity between the distributed representation vectors of the extracted sentences and the distributed representation vector of the passage to be summarized as the importance of the candidate extraction sentences. However, the distributed representation itself is a technology related to machine learning. We used a trained skip-gram model [1] for the distributed representation. This model uses the text of all Japanese Wikipedia articles as training data, and vectors of words and entities represented by named entities are trained in the same 200-dimensional space. Initially, the unit for calculating the similarity to the passage was not a sentence, but a substring separated by punctuation marks (commas and periods). We extracted substrings with high similarity and combined them to output a summary, but it was grammatically and contextually broken. Therefore, we decided to avoid this by turning the extraction strings into sentences. The candidate extraction sentences are pre-compressed (or pre-summarized) to avoid exceeding the specified number of characters in a sentence. For sentence compression (in-sentence summarization), we propose a new method that regards the dependency structure of the *bunsetsu*-phrase as the basic importance and applies an MMR sequentially using the cosine similarity of the distributed representation vector of each *bunsetsu*-phrase.

3.2 Related Studies

Kimura et al.[8] discussed several phrases to be considered in summarizing the local council proceedings. They pointed out that the use of key expressions (e.g., "〜を伺います" at the end of a question sentence) is useful for extracting important strings, and we also used such expressions. Noguchi et al. [13] proposed a method to estimate the importance of sentences using distributed expressions of words in summarization for question sentences. We basically follow their approach to measure the importance of question and answer statements in local council proceedings. Matsumoto et al. [19] tested whether learning word distribution representations suitable for the document summarization improves the accuracy. Specifically, their method embeds the document and the reference summary in a vector space using a distributed representation of the words. The distributed representation of words is learned to minimize the distance between a document vector and the vector of its reference summary. The obtained distributed representation of words is expected to be suitable for document summarization. Local council proceedings are documents with a unique format, mainly because they contain many political terms. The distributed representation model learned from parliamentary proceedings seems to imply these features, and using this may contribute to improving the accuracy of the summary. We did not end up implementing this approach in this system, but we hope to do so in the future. Jaggi et al. [1], tackle keyphrase extraction from single documents using EmbedRank, a novel unsupervised method that leverages sentence embeddings. EmbedRank introduces embedding-based MMR for new phrases to improve the coverage and diversity of selected key phrases. However, EmbedRank is not considered appropriate for application directly to parliamentary proceedings because it is inadequate for sentences that are too long. Therefore, we applied a similar technique to sentences and built an in-sentence summary system. In addition, whereas EmbedRank uses the similarity between the document and the phrase as the basic importance, our method uses the depth of the dependency structure of the *bunsetsu*-phrase as the basic importance. Our method attempts to generate summary sentences in the same way as the above related studies, using a sentence extraction approach with a similarity measure of the distributed representation. However, there is a difference in that we perform in-sentence summaries for each sentence in advance and then extract them using similarity measures of the distributed representation.

3.3 Method

For each question or answer passage, the following process is applied. The flow of these processes is shown in Figure 6.

1. Split the passage to be summarized into sentences.
2. Compress each sentence so that it becomes less than or equal to the specified number of characters (the details are described in Section 3.3.1).
3. From the compressed sentences in Step 2, extract a sentence containing sentence-ending expressions specific to question and answer sessions, such as '〜を伺います' and '〜をしてまいります', if any. If not, continue to step 4.

[1]Japanese Wikipedia Entity Vector Model, Inui and Suzuki Lab, Tohoku University (2017)

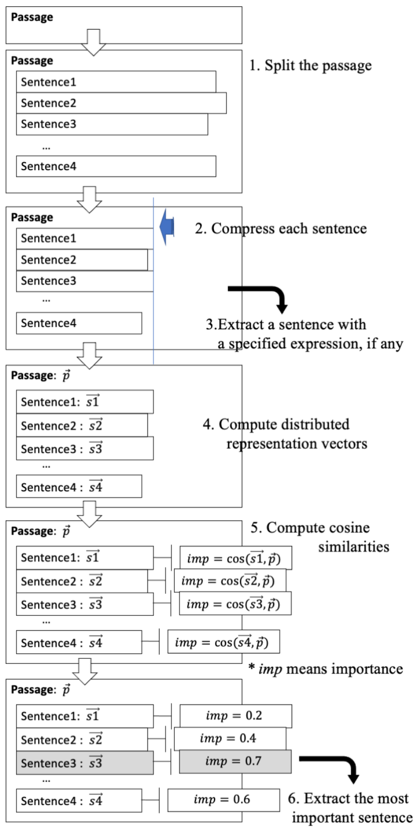

Figure 6: Flow of summarizing each question or answer passage

4. Calculate the distributed representation vector for each compressed sentence. In addition, calculate the distributed representation vector for the passage to be summarized (obtain the distributed representations of adjectives, nouns, and verbs and take their sum).
5. Compute the cosine similarity between the distributed representation vectors of each compressed sentence and the passage to be summarized and regard it as the importance of each compressed sentence.
6. Extract the most important compressed sentence.
7. If there is still room for the specified number of characters, compress the sentence again to fill in the remaining characters (go back to step 2).
8. Combine the extracted compressed sentences and output them as a summary.

3.3.1 Method of Sentence Compression (In-sentence Summarization). For each sentence, in-sentence summarization is conducted through the following processes. The flow of the processes is shown in figure 7.

i. Using CaboCha, analyze the dependency structure of sentences and clarify the relationship between *bunsetsu*-phrases.
ii. Assign higher importance to shallow *bunsetsu*-phrases in the dependency structure. The importance of each *bunsetsu*-phrase is set by multiplying the depth of the *bunsetsu*-phrase in the dependent structure by -1.

$$Imp(C) = -1 \cdot depth(C) \tag{1}$$

Here, C is the *bunsetsu*-phrase, $Imp(C)$ is the importance of the *bunsetsu*-phrase, and $depth(C)$ is the depth of the *bunsetsu*-phrase in the dependent structure. Therefore, the importance of the

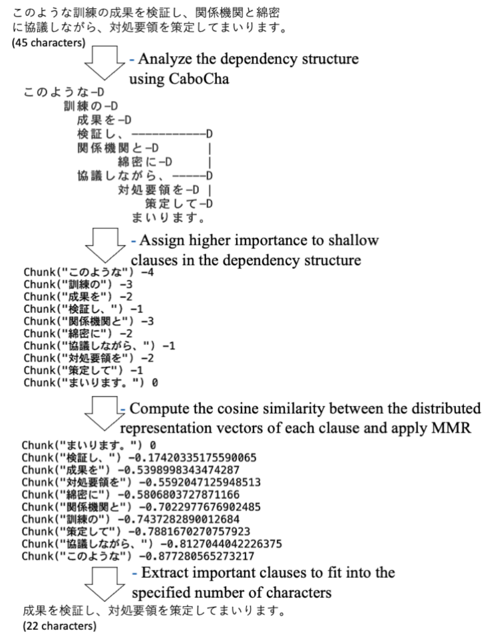

Figure 7: Method of sentence compression (in-sentence summarization)

bunsetsu-phrase at the end of a sentence, or the predicate, is always zero and is the maximum value.

iii. Calculate the distributed representation vector for each *bunsetsu*-phrase (obtain the distributed representations of adjectives, nouns, and verbs and take their sum).
iv. Compute the cosine similarity between the distributed representation vectors of each *bunsetsu*-phrase and apply the MMR (apply the MMR repeatedly, and select unique *bunsetsu*-phrases sequentially, one by one). Thus, a modification is made to the importance level previously described.
To use MMR here, we adapt the following equation:

$$MMR = \arg\max_{C_i \in \mathbf{C} \setminus \mathbf{S}} \left\{ \lambda\ Imp(C_i) - (1-\lambda) \max_{C_j \in \mathbf{S}} cos_{sim}(C_i, C_j) \right\}. \tag{2}$$

where $\mathbf{C}$ is the set of all candidate *bunsetsu*-phrases, $\mathbf{S}$ is the set of selected *bunsetsu*-phrases, and $\mathbf{C}\setminus\mathbf{S}$ is the set of non-selected *bunsetsu*-phrases. We set the lambda value to 0.15.

v. Extract the most important *bunsetsu*-phrases to fit into the specified number of characters. Some measures have been taken to avoid a grammatical breakdown. Specifically, when a *bunsetsu*-phrase containing case particles such as "〜を" or "〜に" is selected, the *bunsetsu*-phrase to which it links must also be extracted.
vi. Delete grammatically incorrect *bunsetsu*-phrases (e.g., phrases such as '、上で、' and phrases that begin with formal nouns such as 'ことを望みます').

3.4 Result

- Forst(215): Output summaries using the method described in section 3.3
- Forst(176): Output summaries extracting substrings separated by punctuation marks

3.5 Discussion

The final summary results were approximately 6% to 47% above the average for all evaluation categories, which are generally not bad (Table 2). However, our method of adopting the cosine similarity between the distributed representation vector of the extracted sentences and the distributed representation vector of the passage

Table 2: The results of a formal run

	Forst(215)	Forst(176)	average of all submissions
ROUGE	0.2410	0.0782	0.185
Content(X=2)	0.778	0.354	0.615
Content(X=0)	0.667	0.275	0.533
Well-formed	1.701	1.403	1.595
Non-twisted	1.044	0.523	0.823
evaluable Non-twisted (C>=1,WF>=1)	1.589	1.261	1.552
Sentence goodness	0.780	0.259	0.591
Dialog goodness	0.604	0.132	0.41

Table 3: Control experiments

	ROUGE
Forst(215)	0.2410
case that extracting sentences without prior in-sentence summarization	0.2275
case that not applying MMR	0.2453
case that not prioritizing sentences that indicate the intention to ask or answer questions	0.1430

to be summarized as the importance does not seem to be necessarily appropriate. Whereas the summary of council proceedings requires the extraction of sentences that indicate the intention to ask or answer questions (e.g., sentences containing "～を伺います", "～してまいります"), we have found that simply using the cosine similarity as the importance of these sentences does not make them more important. After all, the system prioritizes the sentences that contain sentence-end expressions specific to the question and answer text. Furthermore, the current system summarizes each question and answer independently. Therefore, it is not possible to summarize based on the structure of the dialogue. In the quest for a better summary, we should consider the context and logical development. This could be done by calculating the correspondence between passages in advance and using it in the summary. Our original method, which regards the depth of the dependency structure of the *bunsetsu*-phrase as the basic importance, has contributed to the improvement of our results. Sentence extraction with prior in-sentence summarization for each sentence by 0.0135 points compared to the case without prior sentence summarization. However, applying MMR sequentially using the cosine similarity of the distributed representation vector of each *bunsetsu*-phrase does not contribute to the improvement of the ROUGE score. Our subjective impression is that MMR seemed to have made the meaning of the summary easier to understand, and thus we will continue to evaluate and carefully consider the introduction of this system. In addition, in-sentence summarization often breaks down the grammar. We can also consider measures such as taking into account the strength of the relationship between the *bunsetsu*-phrases.

4 DIALOG SUMMARIZATION B

4.1 Approach

For the NTCIR-15 QA Lab-PoliInfo dialog summarization task,we need to determine the questioner's question and the corresponding answer from the minutes of parliament. However, the minutes of parliament are not the same as a normal question-and-answer dialogue, there are two problems. First, the questioner's speech will include several questions, and the replier will also answer several questions. Second, one question may have multiple answers from different persons, and the time between the question and the answer is extremely long. Thus, how to select the question and the corresponding answer is our task, as is compressing them according to the word limit. We designed the system in an extraction-like manner. The word embedding idea has been proposed for years, and it can show the relationships of words by calculating the similarity between them, such as calculating the cosine value between the words or distance in space by using word embedding. Word2vec is a machine learning model used for training word embedding, and Wikipedia data are normally used as training data. We use the newest jawikipedia file, extract text data by gensim[2], MeCab used for word segmentation, and Stopwords for removing meaningless words. Finally, We will obtain a word-embedding model. According to the information provided by the original answer sheet file, we can narrow the range of candidate sentences, and then generate the sentence vector by using word embedding. We can find the relationship between sentences based on a sentence vector similarity. Using the similarity between the original information and sentences, in which the higher similarity the better, we can find questions with high possibilities. In addition, we can find the answers based on the similarity between the found questions and the candidates of the answer sentences.

4.2 Related Studies

Le et al. (2014) [11] proposed the idea that Doc2vec can form different paragraph vectors for different documents, and thus the similarity between different paragraphs can be found, not just the similarity between words.The vector of sentences can represent the distributed expressions. Kimura et al. (2019) [18]described the tasks of PoliInfo2, and shows the data constructions of the minutes. Kazuki Terazawa et al. (2019) [7]proposed that we can use clue expressions help us find the questions and answers.For instance, "伺います" is the clue expression of a question, and "思っております" is the clue expression of an answer.

4.3 Method

Figure 8 shows the overview of the proposed method.

4.3.1 Word embedding construction. Figure 9 shows the process of word embedding construction. First, we download the newest Japanese Wikipedia data file, the most commonly used text data in word2vec training, as the training data. Second, we use gensim to extract text data from the file, and omit the punctuation. Third, we use the MeCab tokenizer with IPAdic to segment the text into words. There is a problem in that sometimes a word will be segmented into two or more words. Namely, IPAdic cannot correctly recognize compound nouns and new words because it is not updated regularly. For instance, "ピコ太郎" is divided into two words: "ピコ" and "太郎". This might influence the vector of words, and thus we use a new word dictionary called mecab-ipadic-NEologd, which will be updated every week. Finally, we use gensim for the word2vec model training to generate word vectors.

4.3.2 Minutes of parliament format change. We extract the text data of Pref13 tokyo.json file. Here, we apply a process in which one person's consecutive speeches merge into a single utterance, and log the time, namely, $\cdots$, as one element of a list, making it easier to read.Figure 10 shows the construction of previous list,and figure 11 shows the construction of the list now.

4.3.3 Find the question. Figure 12 shows the process of finding the question. Through observation,we can find that within the smallest unit, composed by an utterance with the same time and title, the questioner will explain all problems in the first speech. Thus, the information about the time, title, and speaker's name provided by the answer sheet helps us to narrow down the range to only an element of list, which is constructed in advance. Another thing we can learn from observation is that phrases such as "伺います" appear in the target sentences. In addition, the answer sheet provides a "subtopic", which is always a noun appearing before the target sentence. We therefore have a strategy.

1. According to the time, title, and name, find the unique element from list.
2. Split the speech into sentences,and put them into a new list.
3. For every element in the new list, use MeCab for word segmentation. If all "subtopic" words appear in this element, they will be added to the candidate list. From here, find the next sentence that contains "伺い", and put it into the candidate list.

[2]https://radimrehurek.com/gensim/

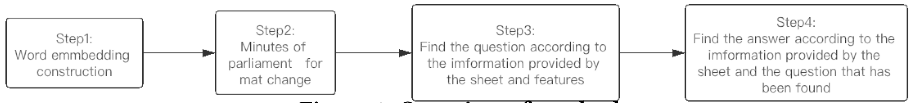

Figure 8: Overview of method

Figure 9: Word embedding construction

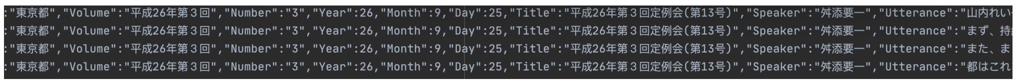

Figure 10: Previous construction

Figure 11: List construction

Table 4: The results of Summarization B

ID(Forst)	modification	ROUGE
247	Add a second sentence into the question candidate list	0.1471
235	Increase the importance of "Subtopic" words	0.1384
231	TF-idf	0.1219

4. If the number of words in the sentences in the candidate list does not exceed the limit, all sentences are considered to be questions. Otherwise, we continue to divide the sentences by commas, and take the sum of the noun vectors and average them as the vector of each part. The "subtopic" is also generated as a vector, and the cosine similarity of each part is calculated with this vector, and parts will be selected based on their similarity until reaching the limitation. Finally, those parts are sorted by original order as a question.

4.3.4 Find the answer. Figure 13 shows the process of finding the answer. As with the previous implementation, using the information on the sheet can help us narrow down the range. In addition, we have found the question, and thus we simply need to find the speech reported by a person with the corresponding position nearest to the question in the back. Here, the sheet simply tell us the title "AnswerSpeaker" representing the speaker's job, which will change over time. Therefore, we extract the job information from the question, and make a dynamical lookup table, the index of which is the name of the job, and the value is the person's name. We can then easily find the speech according to the "AnswerSpeaker". The speech text is then split into sentences, and a sentence vector is generated, along with the questions space. For every sentence vector, the distance between the between the vector of question is calculated until reaching the limitation, and the closest sentences are appended into the candidate list. Here, we apply a change in the vector generation, if the "subtopic" words appear in a sentence, it will add five times vector. This can enhance the importance. The next step is the same as before, a text compression for the candidate list.

4.4 Result

Table 4 shows the results of our method, and this system finally obtained an accuracy of 14.71 in rouge value.

4.5 Discussion

Our method can effectively work on a one-topic-one-question type, but if the topic needs to be answered from multiple aspects, it cannot extract all questions. Because it is based on an extractive approach, sometimes the text is grammatically broken. Although we try to extract parts from adjacent sentences and put them in an original order, there is a problem that we calculate the sentences vector without considering the importance of other words, and we simply enhance the "subtopic" words. Tf-idf is an idea to solve this problem, although the effect is poor. Perhaps calculating the importance based on the distance between the "subtopic" and word would be a good approach. "Bert" is a model that can generate the

Table 5: Examples of restricted condition

1) Numerous different expressions for one entity
風適法, 風営法, 風俗適正化法 –> 風俗営業等の規制及び業務の適正化等に関する法律
2) New expression combining existing expressions
カジノ法案, カジノ解禁法案 –> カジノ・賭博解禁法案
3) Abbreviation in English
環太平洋パートナーシップ協定の締結に伴う 関係法律の整備に関する法律 –> ＴＰＰ整備法
4) Similar form of non-mention and mention
過労死促進につながる戦後最悪の労働法制 : non-mention 過労死等防止対策推進法 : mention

sentences in a grammatical manner and include all information that has been read. Here, we can find the approximate positions of the questions and answers, and thus combining them into a pair is a good way to pre-train the model.

5 ENTITY LINKING

5.1 Approach

The main goal of this Entity Linking (EL) task is to extract mention from minutes of the Tokyo Metropolitan Assembly and link them to the Wikipedia page. Unlike a typical entity linking task, minutes were given as an input document; in other words, the category of the mention is restricted only to legislation. There are several challenges in this domain-specific condition. First, there are numerous different expressions of the mention, but it has to be linked to the same entity in the knowledge base. For example, mentions are similar but slightly different in the first case presented in Table5. Second, a completely new expression combining several existing expressions or using an abbreviation form in English might appear. In particular, it is quite challenging to recognize that the presented entities are identical in the third case in Table5. Finally, it is challenging to distinguish entities and descriptions, such as the fourth case in Table 5. Many different researchers have promoted essential studies in the field of EL. However, a typical EL model is designed to target universal entities, not a specific entity. Therefore, a typical EL model cannot be applied directly to address such difficulties, as stated above. We focused not on the proposal of a new universal competitive model but a model for domain-specific conditions of the EL system.

5.2 Related Work

Although a neural network method is applied to a typical Named Entity Recognition (NER) task, as indicated by Huang et al.[5] and Lample et al.[10] , the domain-specific NER task has applied a rule-based method.[3] Rule-based techniques are preferred traditionally in NER tasks owing to their explainability. However, such methods are challenging in terms of defining complex rules. These rules

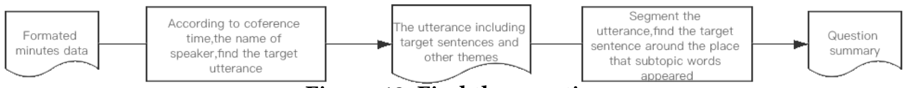

Figure 12: Find the question

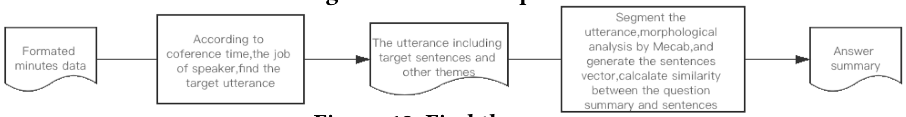

Figure 13: Find the answer

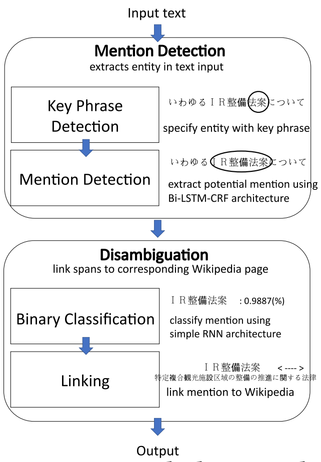

Figure 14: Entity linking pipeline

are based on textual patterns of specific entities, which are different for each property of an entity. Machine learning techniques were recently introduced to the domain-specific NER task Habibi et al.[4]; Leitner et al.[12].

5.3 Method

EL model conducts two tasks: a mention detection task, in which mentions are extracted as spans in a text input, and a disambiguation task, which links spans to the corresponding pages on Wikipedia. In addition to a typical EL model, we added filters before the mention detection process to capture a specific entity. The key phrase detection process specifies the entity in a text input. The binary classification process is applied before the linking task, conducts the classification of spans. In this section, we briefly describe the process and pipeline of our EL model depicted in Figure 14.

5.3.1 Key Phrase Detection. The name of legislation contains a specific phrase indicating that it is legislation. Considering legislation in Japanese, such phrases are located in the last part of the mention. This hypothesis can be confirmed by checking some of the examples listed in Table 5. We defined these phrases as indicating legislation as a key phrase. From the fact that all the mentions contain the key phrase in common, it is reasonable to assume that there would be a high probability of a mention near the key phrase. We applied filter checking key phrases in the text input before the mention detection process to utilize this hypothesis. The key phrase detection process has the following advantages: reduced computing resources and improved accuracy as the filter narrows down the potential mention candidates. We implemented a preparation step for key phrase detection using a simple rule-based method. Training data have already been labeled with IOB tag information. We obtained key phrase data by checking the last part of all mentions in the training data. We could check that there are six key phrase patterns. After the preparation step, the system scans the text input and matches the key phrase. If the word is matched with the key phrase, then the sentence is generated temporarily, consisting of words around the key phrase. This temporary sentence becomes an input for the mention detection task.

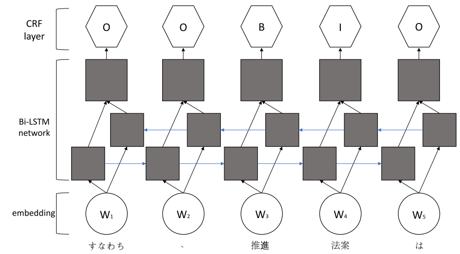

Figure 15: A Bi-LSTM-CRF architecture

5.3.2 Mention detection. We used neural networks for the Mention Detection (MD) task. We applied the LSTM-CTF architecture, which consists of LSTM networks combined with a CTF layer. We briefly describe the LSTM-CRF architecture presented by Huang et al.[5] and Lample et al.[10] used in the NER task. LSTM takes sequential data as input and returns the sequence value as output. The bidirectional LSTM network utilizes both forward and backward input features, which have shown promising NER task results. Compared with LSTM, considering the tag information independently, Conditional Random Fields(CRF) consider tag information at the sentence level. We can train the network under several conditions by taking advantage of the CRF layer. First, the network ensures the B-I pattern of the tag information in the annotation process. This means that the I-tag should come after the B-tag, i.e., an I-B pattern does not appear. Second, networks ensure that there is only one mention for one sentence. The model architecture is shown in Figure15. Collobert et al.[2] reported that word embedding based on a large number of training data has shown good performance in a tagging system. We utilized Wikipedia Entity Vectors by Suzuki et al.[16], which is a pre-trained embedding vector based on over 22,000 Wikipedia articles written in Japanese. We also used a 200-dimensional vector. We used stenographic records for the standing committee of the Tokyo Metropolitan Assembly in 2019 as the training data-set during the training step. We manually annotated the text documents and obtained data on 1,800 entities. It was trained and tested on training data with an accuracy of 99.4.% We designed the MD task to capture all possible candidates regardless of their authenticity. The input text is delivered to the trained model from the key phrase detection task, and after the MD task, the output is the annotated text of the spans.

5.3.3 Binary classification. In the disambiguation task, a typical EL model does not consider the authenticity of the mention. If there is no relevant entity in the knowledge base, it is labeled as non-linkable. However, we need to discern a non-linkable mention from a non-mention. Because an extracted mention should be legislation, the purpose of a binary classification task is to classify the extracted mention, particularly to overcome the fourth problem listed in Table5. We use neural network techniques for the binary classification task. We applied a Recurrent Neural Network(RNN). Binary classification is a many-to-one problem, which takes the input sequence for all time steps and outputs the last RNN cell state. Because the sigmoid function is used as an activation function, the output becomes the degree of authenticity. We used the same date from the MD task during the training step. It was trained and tested on training data with an accuracy of 84.2%. From the output of the

Table 6: The results of Entity Linking

ID(Forst)	Method	F-score
269	LSTM+CRF, Binary Classification	0.3912
217	Rule-based, Binary Classification	0.3910
183	Rule-based, Binary Classification	0.3656
243	LSTM+CRF, Binary Classification	0.3605
147	Rule-based method	0.3389
146	Rule-based method	0.3089

trained model, we were able to obtain a score for the spans. If the score is close to 1, the span is assumed to have a high probability of the mention. By contrast, if the score is close to zero, it is assumed to be a non-mention with a high probability. Spans under 0.5 are assumed as a non-mention, are dropped out. Furthermore, only those over 0.5 are assumed as a mention, proceed to the next step.

5.3.4 Linking. The final step is linking the extracted mention to the relevant Wikipedia page. We implemented a dictionary-based method, which checks both mentions and Wikipedia pages on a phrase basis rather than an exact match of the full name. We use the score from the previous step representing the degree of authenticity. The score increases for matched phrases and decreases for unmatched phrases. Either position of the phrase can be considered. After calculating the score for one mention with all knowledge base entities, a mention is linked to the entity with the highest score. If the mention cannot find the most relevant Wikipedia page, the mention is labeled as non-linkable.

5.4 Result

The result of the entity linking model is presented in Table6. Each submitted output is shown with the corresponding method and F-score. We obtained an F-score of 0.3089 in ID146. This model is applied using the rule-based method, described in 5.3.1Key Phrase Detection. We obtained an F-score of 0.3389 in ID 147 by removing noise from the output from the ID146 model. Using binary classification techniques with the rule-based method resulted in an F-score of 0.3656 in ID183. This result was improved slightly by adding a manually annotated data-set and resulted in an F-score of 0.3910 in ID217. Using LSTM-CRF techniques with the binary classification method and rule-based method resulted in an F-score of 0.3605 in ID243. It seems that the LSTM-CRF model has a slightly lower accuracy than the rule-based method, although it eliminates the concern of being valid only for the given text document. Moreover, it is improved by applying pre-trained word embedding vectors to the LSTM-CRF architecture and results in an F-score of 0.3912 in ID269.

5.5 Discussion

As the mention detection and binary classification tasks use the neural network model, the output from the trained model could be varied when the model has trained again. The result of the EL system is different each time. This makes the system unstable in terms of accuracy, which is accelerated by the corpus condition. The number of legislations in the corpus is biased to only a few. If the neural network model fails to capture one entity initially, the El system eventually fails to capture that entity. When an uncaptured entity accounts for a large portion of entities, the accuracy of the model decreases dramatically. This fault can be fixed by improving the accuracy of each neural network model. To lower the number of uncaptured entities, it is necessary to reduce the risk to the system. There is still room for improvement in the linking task. Our current system does not consider the context information of the knowledge base but only the title of the knowledge base (Wikipedia). However, this approach is not sufficient to address the first problem listed in Table 5.

6 CONCLUSION

In this paper, we described the development of a system for stance classification, two systems for dialog summarization and a system for entity linking. As a result, an accuracy of .9388 for the stance classification, a ROUGE-1 score of .2410 for the dialog summarization and an F-measure of .3910 for the entity linking were obtained.

ACKNOWLEDGEMENT

We would like to thank Advanced Media, Inc. for providing a skip-gram model.

REFERENCES

[1] Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple Unsupervised Keyphrase Extraction using Sentence Embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Brussels, Belgium, 221–229. https://doi.org/10.18653/v1/K18-1022

[2] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural Language Processing (almost) from Scratch. *CoRR* abs/1103.0398 (2011). arXiv:1103.0398 http://arxiv.org/abs/1103.0398

[3] Christopher Dozier, Marc Light, Arun Vachher, Sriharsha Veeramachaneni, and Ramdev Wudali. 2010. Named Entity Recognition and Resolution in Legal Text. 27–43. https://doi.org/10.1007/978-3-642-12837-0_2

[4] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. 2017. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics* 33, 14 (07 2017), i37–i48. https://doi.org/10.1093/bioinformatics/btx228

[5] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *CoRR* abs/1508.01991 (2015). arXiv:1508.01991 http://arxiv.org/abs/1508.01991

[6] YUi Inoue and Atsushi Fujii. 2008. Extraction of opinions from Web public opinion and classification based on pros and cons. *anlp* NLP2008 (march 2008). (in Japanese).

[7] Tomoyoshi Akiba Kazuki Terazawa, Daiki Shirato and Shigeru Masuyam. 2019. AKBL at NTCIR-14 QA Lab-PoliInfo Task. *Proceedings of The 14th NTCIR Conference* (6 2019).

[8] Yasutomo Kimura, Satoshi Sekine, and Kentaro Inui. 2018. Towards Summarizing Local Council Proceedings. *Proceedings of the Twenty-fourth Annual Meeting of the Association for Natural Language Processing (NLP2018)* P5-3 (3 2018), 596–599. (in Japanese).

[9] Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, 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 Task. *Proceedings of The 15th NTCIR Conference* (12 2020).

[10] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 260–270. https://doi.org/10.18653/v1/N16-1030

[11] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *ArXiv* abs/1405.4053 (2014).

[12] Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2019. Fine-Grained Named Entity Recognition in Legal Documents. In *Semantic Systems. The Power of AI and Knowledge Graphs*, Maribel Acosta, Philippe Cudré-Mauroux, Maria Maleshkova, Tassilo Pellegrini, Harald Sack, and York Sure-Vetter (Eds.). Springer International Publishing, Cham, 272–287.

[13] Noguchi Masaki, Tanizuka Taichi, and Kobayashi Hayato. 2015. Summarization of Yahoo! Answers with Distributed Representation. *Proceedings of the Twenty-first Annual Meeting of the Association for Natural Language Processing (NLP2015)* (3 2015). (in Japanese).

[14] Masakazu Nishimura, Takayuki Sakai, and Harumi Murakami. 2009. Extracting opinions on the pros and cons of the bill from newspaper articles. *CiNii* JSAI2009 (2009), 3I34–3I34. https://doi.org/10.11517/pjsai.JSAI2009.0_3I34 (in Japanese).

[15] Hiroki Sakaji, Sakai Hiroyuki, Kobayashi Akio, Uchida Yuzu, Ototake Hokuto, Takamaru Keiichi, and Kimura Taichi. 2017. Extraction of remarks expressing opinions and intentions from the minutes of the prefectural assembly. *anlp* NLP2017 (2017), 3I34–3I34. (in Japanese).

[16] Masatoshi SUZUKI, Koji MATSUDA, Satoshi SEKINE, Naoaki OKAZAKI, and Kentaro INUI. 2018. A Joint Neural Model for Fine-Grained Named Entity Classification of Wikipedia Articles. *IEICE Transactions on Information and Systems* E101.D, 1 (2018), 73–81. https://doi.org/10.1587/transinf.2017SWP0005

[17] Yuji Matsumoto Taku Kudo. 2002. Japanese Dependency Analysis using Cascaded Chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*. 63–69.

[18] Hokuto Ototake Yuzu Uchida Keiichi Takamaru Kotaro Sakamoto Madoka Ishioroshi Teruko Mitamura Noriko Kando Tatsunori Mori Harumichi Yuasa Satoshi Sekine Yasutomo Kimura, Hideyuki Shibuki and Kentaro Inui. 2019. Overview of the NTCIR-14 QA Lab-PoliInfo Task. *Proceedings of The 14th NTCIR Conference* (6 2019).

[19] Taguchi Yuya, Shigeto Yutaro, Shimbo Masashi, and Matsumoto Yuji. 2017. Learning Distributed Representation in Extractive Document Summarization - Minimizing the Distance between Document and Summarization. *Proceedings of the Twenty-third Annual Meeting of the Association for Natural Language Processing (NLP2017)* (3 2017). (in Japanese).