Ibrk at the NTCIR-16 QA Lab-PoliInfo-3 Budget Argument Mining Subtask

Kohei Seguchi
Department of Computer and Information Sciences
Faculty of Engineering, Ibaraki University
4-12-1, Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan
22nm728r@vc.ibaraki.ac.jp

Minoru Sasaki
Department of Computer and Information Sciences
Faculty of Engineering, Ibaraki University
4-12-1, Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan
minoru.sasaki.01@vc.ibaraki.ac.jp

ABSTRACT

In this study, we construct a system to predict argument labels for statements in meeting minutes by using sequence labeling methodology and validate the effectiveness of prediction performance for various input methods of utterance data to predict argument labels for money expressions effectively. To evaluate the validation of the system, we will use the Budget Argument Mining task data in NTCIR-16 QA Lab-Poliinfo-3. We train an argument label prediction model on the training data that exists in the data, dividing it into two types: data for model training and data for model validation. As the prediction model, we use the Bi-directional LSTM-CNNs-CRF model to predict argument labels for each word in the input data and output a series of argument labels. In the experiment, we compare the prediction accuracy of models obtained by changing the data input method, such as the range of sentences containing money expressions. As a result of the experiment, we found that the prediction accuracy of argument labels was higher when each sentence was entered into the model rather than when all the statements of the assembly member were entered into the model. Furthermore, we found that the prediction accuracy of the argument labels can be improved by replacing the numbers in the money expression with special tokens.

KEYWORDS

Sequence labeling, Money expression, Minutes

TEAM NAME

Ibrk

SUBTASK

Budget Argument Mining Subtask (Japanese)

1. INTRODUCTION

This study examined whether the word-level labeling task using the LSTM model was an effective method for the minutes of four parliaments, and whether the method of replacing numbers and money expressions in the data with tokens was a useful operation for improving accuracy under the same conditions.

In this study, we use as a data set the publicly available minutes of the National Diet of Japan and three local councils (Otaru City in Hokkaido, Ibaraki Prefecture, and Fukuoka City in Fukuoka Prefecture), and split them into morphemes by morphological analysis. The purpose of this study is to identify money expressions and to estimate seven classes of labels, such as “Claim” for subject and “Premise” for evidence, for the statements in the divided minutes.

For the label estimation system, we use sequence labeling based on the LSTM model. The model is trained using the data from the conference proceedings assigned to the training data, which is divided into morphemes by morphological analysis, and then each morpheme is assigned one of seven classes such as “Claim” or “Premise”. The effectiveness of the system is verified by applying the model to test data and making predictions.[3]

In addition, there are some numbers and words that are replaced by <_UNK>, a token representing an unknown word, when the data to be used is vectorized by word2vec for prediction. Since these tokens may affect the accuracy of the prediction, the effect of the <_UNK> tokens are verified by replacing the money expressions and numbers that are the main prediction parts with 0 or with <MNY> tokens representing money expressions in advance.

The rest of the paper is organized as follows. Section 2 introduces related methods. Section 3 shows the preparation of the data set. Section 4 shows the overview of the system. Section 5 shows the actual experiment ant its results. Section 6 discusses the results presented in section 5. Section 7 gives a summary of the previous sections and concludes.

2. RELATED METHODS

This section discusses the research and methods related to the experiment.

2.1 word2vec

In this study, word vectors and character vectors were obtained by word2vec. For the model, we used the word2vec model trained in Japanese by Inui Laboratory at Tohoku University.

2.2 Bi-directional LSTM-CNNs-CRF

Bi-directional LSTM-CNNs-CRF is a model that combines CNN, which largely acquire representations of character units, and Bi-directional LSTM-CRF, which performs sequence labeling.[1]

2.2.1 CNN

CNN is used to obtain a distributed representation of character units. Only embeddings of the characters used in a word are used as input, not character features.

First, for each word, the embeddings of the letters that make up the word are used as input, and the character-by-character representation is computed by the CNN.

2.2.2 Bi-directional LSTM-CRF

LSTM is one of the RNNs that can consider the previous context in a single direction context. It is a network that address the problem in RNNs that important contexts that are far away from the target will eventually have little information.

LSTM uses three gates to make important contexts that are far away in a unidirectional context have information. The forget gate adjusts information judged to be unimportant in previous memories so that it does not affect subsequent operations. In the input modulation gate, the information is converted into a form that focuses on the main points of the expression by tanh, and the unimportant information in the input gate is adjusted so that it does not affect subsequent operations and is matched with the information in the forgetting gate. The output gate combines the information from the forgetting gate and the input gate to produce an output. The LSTM process is shown in Figure 1.

Bi-LSTM considers not only the context in one direction but also the context in the opposite direction, and outputs a score of which labels is most appropriate for the input and estimates the label by selecting the label with the highest score. Furthermore, by combining the CRF layer, it is possible to input the score of each label output, calculate the combination with the highest score, and consider what kind of labeling is most natural when considering the context.[1]

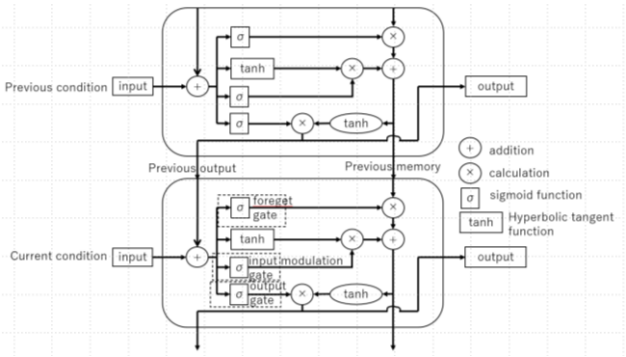

Figure 1: The processing of LSTM

3. DATASET

In this study, the dataset used in the experiment was created from the data of the minutes of the four municipalities from which the money expressions produced by Poliinfo3 were extracted.[3]

The dataset was divided into arbitrary lengths by extracting the utterances from the minutes, and then morphological analysis was performed by MeCab.

MeCab is a Japanese morphological engine developed jointly by the Graduate School of Informatics, Kyoto University and NTT Communication Science Laboratories. The creation of the dataset is shown in Figure 2.

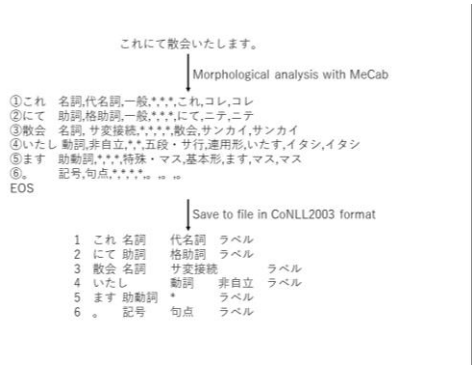

Figure 2: The creation of the dataset

4. OVERVIEW OF THE DISCUSSION LABEL PREDICTION SYSTEM

In this study, we use the method introduces in section 3 to create a dataset for data from which money expressions have been extracted in advance. This data is then passed to the system as input for sequence labeling. The output is obtained in the same format as the input and the output is formatted into json format for evaluation.

This system consists of two parts: a label prediction system and a system for linking money expressions to IDs. These two systems are processed independently of each other. The label prediction system uses NeuroNLP2 by Xuezhe Ma. The system for linking money expressions to IDs was created using data provided by Poliinfo3 to perform the process.

This section is organized as follows. 4.1 shows how the data were formatted. 4.2 shows the processing of NeuroNLP2. 4.3 shows the evaluation method for the prediction results obtained by NeuroNLP2. In 4.4, we show the method of linking money expressions and related-IDs. In section 4.5, we show how to format the results obtained by the label prediction system and the linking system between monetary expressions and related IDs into Json format. In section 4.6, we show the final evaluation method.

4.1 DATA FORMATTING

The dataset is formatted as shown in Figure 2.

We created nine different of data using the nine conditions presented in Section 5. In addition, the data does not distinguish

between parliamentary and local councils. The data is extracted from the minutes as a set of statements and money expressions, and statements for which money expressions have not been extracted are given the label “O” after the data is formatted. For the statements for which money expressions have been extracted, the respective labels are assigned to the money expressions and the “O” labels are assigned to the non-money Expressions. In the case of separating statements into sentences, punctuation marks were used to separate them. To format the test data into json format for evaluation, we assigned “B-None” to the beginning of the money expression labels and “I-None” to the middle. For the test data, the labels were assigned in a one-to-one correspondence between the extracted money expressions and the order of extraction of the money expressions in the utterances since the label information was not used. An example of label assignment for test data is shown below.

- When the money expression extracted for the Japanese sentence 「事業費として 2 億円と 2 億円と十億円を用意しており、2 億円については予備費となっている。」 is [“2 億円”, “十億円”] and the label is “Premise”, the label is assigned only to the expression that first appeared in the utterance in the order of the extracted money expressions as follows.

  -「事業:O / 費:O / として:O / 2:Premise / 億:Premise / 円:Premise / と:O / 2:O / 億:O / 円:O / と:O / 十:Premise / 億:Premise / 円:Premise / を:O / 用意:O / し:O / て:O / おり:O / 、:O / 2:O / 億:O / 円:O / について:O / は:O / 予備:O / 費:O / と:O / なっ:O / て:O / いる:O / 。:O」

4.2 OVERVIEW OF NEURONLP2

NeuroNLP2 was created by Xuezhe Ma. This system receives three files as input: training data, development data, and test data. Initially, it creates a list of letters, words, labels, parts-of-speech, and parts-of-speech subdivisions as a vocabulary. Next, it creates and trains a Bi-directional LSTM-CNNs-CRF model to be used for sequence labeling. When the training is complete, we will use the model at that point and measure the performance of the model using the developed data. If the Fi score, which represents the performance of the model, is higher than the previous Fi score, we consider the model to be better and predict the labels for the test data. The prediction results are output in CoNLL2003 format.[2]

4.2.1 DATA LOAD

The first step in NeuroNLP2 is to prepare the necessary data.

It obtains words and their vectors by trained word2vec, and creates an ID table of words, letters, parts-of-speech, parts-of-speech subdivisions, and labels while eliminating words without vectors using word2vec information.

Finally, it reads the training data, development data, and test data, and creates a table that holds the vectors of words that exist in the data.

4.2.2 CREATE ID TABLE

First, read the training data one line at a time. Extract the characters used from the morphemes one by one and store the corresponding ID for each character as a value in a dictionary type variable. For each word that appears, the number of times it appears is recorded as a value in an independent variable. Part-of-speech, part-of-speech subdivisions, and labels are stored in their respective dictionary type variables with the corresponding ID as a value.

After performing the above operations on all the elements of the training data, words that appear only once in a word are stored as singleton. Words that are determined to be singleton and do not exist in the trained corpus are excluded, and the vocabulary is reduced to the specified number of words. Since the vocabulary of the training data alone is based, the vocabulary of the development and test data is acquired by the same operation.

With the above operations, we created an ID table of words, characters, parts-of-speech, parts-of-speech subdivisions, and labels.[2]

4.2.3 LOADING TRAINING DATA AND DEVELOPMENT/TEST DATA

The training data is divided into buckets and loaded. The data is obtained with each line divided by a delimiter. For each word, the used characters are extracted and converted to character IDs, and each word is obtained as a delimiter. For words, parts-of-speech, parts-of-speech subdivisions, and labels, each element is converted into its own ID. These IDs are stored in the NER instance. A NER instance is created for each block of data.

The training data is divided into 9 buckets and loaded into each bucket according to the maximum number of characters. The maximum number of words is 5, 10, 15, 20, 25, 30, 40, 50, 140 respectively, and the buckets are allocated according to the number of words held in the NER instance. Each bucket is assigned an ID in order from 0 to 140, starting from the front, and the bucket with the corresponding ID stores the information converted to the respective IDs of words, characters, parts-of-speech, parts-of-speech subdivisions, and labels.

For each of the nine buckets created here, the information converted to IDs is padded with 1 to make it fit into the maximum array that has been set. At the same time, we obtain the mask information, where valid IDs are represented by 1 and padded IDs by 0, and the position information where the singleton appears. The data divided into buckets is finally kept as a bucket unit, and in each bucket, the word, character, part-of-speech, part-of-speech subdivision, label, mask, location information of the singleton, and the length of the sentence when the sentence is divided into morphemes are obtained together.

Load development and test data without separating them into buckets. The basic structure of the data will be the same as the training data, but since it is not divided into buckets, the data will be held in a structure with a shallower layer of hierarchy.[2]

4.2.4 CREATE WORD TABLE

For each word read up to this point, create a dictionary type word table with the word as the key and the vector for the word as the value. For words that have vectors in word2vec using the ID information of the word, the learned vectors are used; for words that do not have vectors in word2vec, the elements of the vectors are randomly generated and used as vectors for the words.[2]

4.2.5 TRAINING

The training uses Bi-directional LSTM-CNNs-CRF model. The data used is divided into buckets, and training proceeds for each bucket of data. For training, we provide MASK information, which holds the IDs of words and character labels, as well as the padding information.

For training with LSTM, the word and character vectors are combined and inputted.[2]

4.3 EVALATING THE ACCURACY OF SEQUENCE LABELING

After each epoch of training, the model is evaluated using the development data. The models trained using NeuroNLP2 were given development data and the accuracy of the label predictions for the data was used as the evaluation value. The evaluation values of the development data are obtained for Accuracy, Precision, Recall, and F1 score, respectively. Accuracy was calculated as the percentage of the total data that matched the correct answer. Precision is calculated as the percentage of correct responses for the expressions that were predicted to be money expressions. Recall was calculated as the percentage of predictions that were correct for expressions for which the correct answer was a money expression. The F1 score was calculated as the harmonic mean of precision and recall. The best learning performance is determined when the F1 score exceeds the previous F1 score reconsidered, and when the F1 score is updated, the label is predicted for the test data and the prediction results are stored.

4.4 LINKING MONEY EXPRESSIONS TO RELATED-ID

NeuroNLP2 only performs sequence labeling, so the connection between money expressions and related-IDs was done in a different way. In this section, we describe our system for linking money expressions to related-IDs.

4.4.1 EMBEDDED EXPRESSIONS OF PROJECT NAMES IN BUDGET TITLES

Require embedded expressions of project names for each project name (budget title) included in budget items from determined budget items in national and local governments. We use the BERT tokenizer to split the project name into a word sequence. We use the learned model of BERT to find the embedded representation for each word in this word sequence. In this set of embedded representations, the embedded representation of the special token '[CLS]' in the final layer is used as the embedded representation of the business name.

4.4.2 EMBEDDED EXPRESSIONS OF BUSINESS CONTENT IN MEETING MINUTES STATEMENT

Extract the part related to the business contents from the sentence containing money expressions in the minutes of the meeting of the Diet and local governments and ask for the embedded expressions. To capture the business contents in which money expressions appear, the sentences including money expressions are extracted from the statements in the meeting minutes. The keywords that are likely to appear with the business contents are searched for in these sentences. However, these keywords are manually extracted from the characteristics of the training data [“計上”, “追加”, “施策”, “見込”, “実施”, “支払”, “手取り”, “値上げ”, “料”, ”負担”, “手当”, “価格”, “概算”, “増額”, “費”, “高く”, “経費”, “積算”, “予算”, “想定”, “出資金”, “貸付金”] are keywords set as. If those keywords exist, the project content existing in the budget item is judged to be included and the discussion content is extracted. Since the business contents related to the money expression tend to exist in the subject of the sentence that precedes it, the word sequence from the character following the reading point to “は” or “について” in this sentence is extracted as the business contents. We input these words into the BERT trained model and find the embedded representation for each word. In the resulting set of embedded representations, the embedded representation of the special token “[CLS]” in the final layer is used as the embedded representation of the business content. If the keyword does not appear in the sentence, it is assumed that the amount of money in the sentence is not related to the business contents in the agenda, and no association with the budget item is made.

4.4.3 ASSOCIATION OF BUSINESS NAME AND BUSINESS CONTENT

Correlate the project items in the budget title with the project details in the meeting minutes remarks. Associate the project title and the project contents with the budget items of the same year as the year of the statement in the assembly. Calculate the cosine similarity between the embedded expression of the project name and the embedded expression of the contents obtained in the previous section. The ID of the project name with the maximum cosine similarity is assigned to the Related-ID.

4.5 FORMAT TO JSON FORMAT

Extract the money expression and prediction label as a set using the original label that is assigned in one-to-one correspondence with the money expression that is being extracted for formatting into json format. From the original test data, the number of monetary expressions in the utterance is stored in a list, and the set of money expressions and prediction labels are grouped

together according to the number of monetary expressions. The predictive label data summarized in the original test data is embedded and formatted into json format for evaluation.

4.6 EVALUATION OF PREDICTION

The evaluation was done by the evaluation program. This program outputs the overall evaluation value, the evaluation value in the minutes of the local government, and the evaluation value in the minutes of the Diet, respectively.[3]

5. EXPERIMENT

In this section, we describe the actual experiments we conducted.

5.1 ENVIRONMENT

- System

  CPU: AMD Ryzen 7 5800X 8-Core Processor 3.80 GHz

  OS: Windows 10 Home

- Software

  Python: 3.9.7

  Pytorch: 1.10.1

  MeCab: 0.996

5.2 EXPERIMENTAL METHODS

In the experiment, the following conditions were applied to the data with modifications. In addition, the experiment was conducted on two types of data: one where the data was separated by the entire utterance, and one where the data was separated by each sentence, so that the actual number of conditions is nine.

1. Data that limits the same expression in a statement to only the first one for each money expression as a whole statement.

   -In the case where the money expression is “十兆円” for the statement “予備費として十兆円を確保しております。この十兆円は本年度使用されなければ来年度へと持ち越されます。”, we first perform morphological analysis on the statement. Next, labels are assigned to the expression “十兆円” in the utterance, but only the first “十兆円” is assigned. The label of the money expression “十兆円” is Premise.

   -”予備:O /費:O / として:O / 十:Premise / 兆:Premise / 円:Premise / を:O / 確保:O / し:O / て:O / おり:O / ます:O / 。:O / この:O / 十:O / 兆:O / 円:O / は:O / 本:O / 年度:O / 使用:O / さ:O / れ:O / なけれ:O / ば:O / 来年度:O / へ:O / 持ち越さ:O / れ:O / ます:O / 。:O”

2. Data that targets multiple expressions for a single money expression, with the entire statement as a single unit.

   - In the case where the money expression is “十兆円” for the statement “予備費として十兆円を確保しております。この十兆円は本年度使用されなければ来年度へと持ち越されます。”, we first perform morphological analysis on the statement. Next, we assign the same label to all expressions that are identical to the extracted money expression. The label of the money expression “十兆円” is Premise.

   -”予備:O /費:O / として:O / 十:Premise / 兆:Premise / 円:Premise / を:O / 確保:O / し:O / て:O / おり:O / ます:O / 。:O / この:O / 十:P / 兆:P / 円:P / は:O / 本:O / 年度:O / 使用:O / さ:O / れ:O / なけれ:O / ば:O / 来年度:O / へ:O / 持ち越さ:O / れ:O / ます:O / 。:O”

3. Data in which the entire statement is considered as a single unit, multiple targets are targeted for each money expression, ant the money expression are replaced with <MNY> tokens.

   - In the case where the money expression is “十兆円” for the statement “予備費として十兆円を確保しております。この十兆円は本年度使用されなければ来年度へと持ち越されます。”, we first perform morphological analysis on the statement. In addition, all expressions similar to the extracted money expressions are replaced by <MNY> tokens and assigned the same label during learning as follows. The label of the money Expression “十兆円” is Premise.

   -” 予 備 :O / 費 :O / と し て :O / <MNY>:Premise / <MNY>:Premise / <MNY>:Premise / を:O / 確保:O / し:O / て:O / おり:O / ます:O / 。:O / この:O / <MNY>:P / <MNY>:P / <MNY>:P / は:O / 本:O / 年度:O / 使用:O / さ:O / れ:O / なけれ:O / ば:O / 来年度:O / へ:O / 持ち越さ:O / れ:O / ます:O / 。:O”

4. Data where the entire statement is considered as a single unit, and all numbers are replaced with 0 for each money expression.

   - In the case where the money expression is “十兆円” for the statement “予備費として十兆円を確保しております。この十兆円は本年度使用されなければ来年度へと持ち越されます。”, we first perform morphological analysis on the statement. In addition, all expressions that are identical to the extracted money expression are replaced by 0 and assigned the same label as follows. When replacing a number with a “0” only, the substitution is made up to the point immediately before the word that represents the unit of number in the money expression. The substitution of “0” is done in the same way for all numbers other than money expressions. The label of the money Expression “十兆円” is Premise.

-”予備:O / 費:O / として:O / 0:Premise / 兆:Premise / 円:Premise / を:O / 確保:O / し:O / て:O / おり:O / ます:O / 。:O / この:O / 0:Premise / 兆:Premise / 円:Premise / は:O / 本:O / 年度:O / 使用:O / さ:O / れ:O / なけれ:O / ば:O / 来年度:O / へ:O / 持ち越さ:O / れ:O / ます:O / 。:O”

5. Data in which the entire statement is considered as a single unit and multiple amounts are targeted for each money expression, and the money expression is replaced with a <MNY> token and the number is replaced with 0.

   - In the case where the money expression is “十兆円” for the statement “予備費として十兆円を確保しております。この十兆円は 2 度の審議を必要とします。”, we first perform morphological analysis on the statement. Furthermore, for the same expression as the extracted money expression, we assign the same label by replacing other numbers with 0 after replacing them with <MNY> token as follows. When replacing <MNY> tokens, the entire money expression is replaced, and when replacing a number with 0, the replacement is applied to the portion of the money expression immediately before the word that represents the number unit. The substitution of “0” is done in the same way for all numbers other than money expressions. The label of the money expression “十兆円” is Premise.

   -” 予 備 :O / 費 :O / と し て :O / <MNY>:Premise / <MNY>:Premise / <MNY>:Premise / を:O / 確保:O / し:O / て:O / お り :O / ま す :O / 。 :O / こ の :O / <MNY>:Premise / <MNY>:Premise / <MNY>:Premise / は:O / 0:O / 度:O / の:O / 審議:O / を:O / 必要:O / と:O / し:O / ます:O / 。:O”

6. Data for multiple targets for one money expression with one sentence.

   -For the statement “予備費として十兆円を確保しこの十兆円は来年に持ち越されます。この十兆円は 2 度の審議を必要とします。”, separate the statements into sentence like “予備費として十兆円を確保しこの十兆円は来年に持ち越されます。” and perform morphological analysis. The same label is assigned to all expressions that are identical to the extracted money expression. The label for the money expression “十兆円” is Premise.

   -”予備:O / 費:O / として:O / 十:Premise / 兆:Premise / 円:Premise / を:O / 確保:O / し:O / この:O / 十:Premise / 兆:Premise / 円:Premise / は:O / 来年:O / に:O / 持ち越さ:O / れ:O / ます:O / 。:O”

7. Data in which one sentence is considered as a whole, and multiple sentences are targeted for one money expression, and the money expression is replaced by <MNY> tokens.

   - For the statement “予備費として十兆円を確保しこの十兆円は来年に持ち越されます。この十兆円は 2 度の審議を必要とします。”, separate the statements into sentence like “予備費として十兆円を確保しこの十兆円は来年に持ち越されます。” and perform morphological analysis. In addition, all expressions identical to the extracted money expression are replaced by <MNY> tokens and assigned the same label. The label for the money expression “十兆円” is Premise.

   -” 予 備 :O / 費 :O / と し て :O / <MNY>:Premise / <MNY>:Premise / <MNY>:Premise / を:O / 確保:O / し:O / この:O / <MNY>:P / <MNY>:P / <MNY>:P / は:O / 来年:O / に:O / 持ち越さ:O / れ:O / ます:O / 。:O”

8. Data for which all numbers are replaced with 0 for each money expression as a single sentence.

   - For the statement “予備費として十兆円を確保しこの十兆円は来年に持ち越されます。この十兆円は 2 度の審議を必要とします。”, separate the statements into sentence like “予備費として十兆円を確保しこの十兆円は来年に持ち越されます。” and perform morphological analysis. Furthermore, we assign the same label to all expressions that are identical to the extracted money expression, replacing it with 0. The replacement was performed in the same way as in Method 4. The label of the money expression “十兆円” is Premise.

   -”予備:O /費:O / として:O / 0:Premise / 兆:Premise / 円:Premise / を:O / 確保:O / し:O / この:O / 0:Premise / 兆:Premise / 円:Premise / は:O / 来年:O / に:O / 持ち越さ:O / れ:O / ます:O / 。:O”

9. Data in which one sentence is considered as a single unit, and multiple amounts are targeted for each money expression, and the money expression is replaced with <MNY> tokens, and the numbers are replaced with 0.

   - For the statement “予備費として十兆円を確保しこの十兆円は来年に持ち越されます。この十兆円は 2 度の審議を必要とします。”, separate the statements into sentence like “予備費として十兆円を確保しこの十兆円は来年に持ち越されます。” and perform morphological analysis. In addition, for all expressions that are identical to the extracted money expression, replace the <MNY> token, then replace all remaining numbers with 0 and assign an identical label. The replacement was performed in the same way as in Method 5. The label for the money expression “十兆円” is Premise.

   -” こ の :O / <MNY>:Premise / <MNY>:Premise / <MNY>:Premise / は P:O / 0:O / 度:O / の:O / 審議:O / を:O / 必要:O / と:O / し:O / ます:O / 。:O”

Table 1: Comparison of accuracy between different methods of creating data sets

Conditions	Accuracy
One cohesive whole utterance and only one money expression in the whole statement	0.1212
One cohesive whole utterance and multiple money expressions in the whole statement	0.4096
One cohesive whole utterance, multiple money expressions in the whole statement and replacing money expressions	0.3577
One cohesive whole utterance, multiple money expressions in the whole statement and replacing the number of the whole statement	0.3692
One cohesive whole utterance, multiple money expressions in the whole statement and after replacing the money expression, replace the number of the whole utterance	0.3577
One sentence in a cohesive phrase and multiple money expressions in a single statement	0.4385
One sentence in a cohesive phrase, multiple money expressions in a single statement and replacing money expressions	0.4385
One sentence in a cohesive phrase, multiple money expressions in a single statement and replacing the number of the whole statement	0.4019
One sentence in a cohesive phrase, multiple money expressions in a single statement and after replacing the money expression, replace the number of the whole utterance	0.4788

5.3 EXPERIMENTAL

The data obtained by the method introduced in 5.2 was passed to NeuroNLP2 as input. The results output from NeuroNLP2 are combined with the results output from the system for linking money expressions and IDs, and reworked into json format, and the data is given to the Poliinfo3 evaluation system. The results of the experiment, broken down by data creation procedure, are shown in Table 1.

5.4 EXPERIMENTAL RESULTS IN LABEL PREDICTION SYSTEM

Table 1 below shows the accuracy for each data created based on each condition.

5.5 EXPERIMENTAL RESULTS IN SYSTEM FOR LINKING MONEY EXPRESSIONS TO IDs

We tried linking money expressions to IDs using the method described in section 4.4, accuracy was 0.0000.

6. DISCUSSION

In this section, we discuss the results of the experiments and show the effectiveness and problems.

The accuracy of the label prediction using the data generated in the first condition was 0.1212, which is very low. This result can be attributed to the fact that there are multiple occurrences of the same money expression in the data, and the label of the money expression appearing after the second occurrence is “O” during the training process. This means that the label “O” is preferentially selected for money expressions that appear multiple times, resulting in low accuracy.

To solve the problem in the first condition, the second to fifth methods, which targeted all the same money expressions in the utterances, produced scores that were more than twice as high as those of the first method, suggesting that this approach was very effective. In the second condition, where the number of money expressions was expanded, the score was 0.4096, the highest among the first five conditions. On the other hand, the data for the third condition, in which the money expression was replaced with <MNY> tokens in addition to the second condition, showed 0.3577, the data for the fourth condition, in which all numbers were replaced with 0, showed 0.3692, and the data for the fifth condition, in which the money expression was replaced with <MNY> tokens and then the numbers were replaced with 0, showed 0.5096. The data for the fifth condition, in which the number was replaced with 0 after replacing the monetary expression with <MNY> tokens, was 0.3577, which was lower than the data for the second condition. In the third to fifth conditions, the experiment was conducted with the idea of identifying and predicting the nature of the money expression based on the characteristics of the surrounding words rather than on the numbers, but on the contrary, the characteristics captured in the second condition may have been lost due to the differences in the numerical values of the money expression. In the second condition, however, the differences in the numerical values of the money expressions may have made it impossible to capture the features that were captured in the first condition. In addition, the differences in the numerical values of the money expressions may have had a strong effect on the longer sentences since the cohesion was defined as the entire statement.

The data according to the sixth to ninth conditions is the data that was shortened to one sentence based on the results of using the data according to the second to fifth conditions. The data for the sixth condition, in which the object of money expression is the same as the data for the second condition, has an evaluation value of 0.4385, which is an increase of 0.03 from the second data,

although it is a small money, indicating that the data for the first condition was effective. In addition, the data for the seventh condition, which is the same as the data for the third condition, the data for the eighth condition, which is the same as the data for the fourth condition, and the data for the ninth condition, which is the same as the data for the fifth condition, have values of 0.4385, 0.4019, and 0.4788, respectively, indicating that the one-sentence data was effective. Compared with the data in the fifth condition, the data in the ninth condition were 0.4385, 0.4019, and 0.4788, respectively, indicating that the method was effective, since it increased to 0 .0808, 0 .0327, and 0 .1211, respectively. Also, from the ninth data, which had the highest accuracy among the sixth to ninth conditions, it can be considered that for data for which money expressions were extracted in advance, replacing money expressions with tokens and other numbers with 0s during data formatting made it easier to capture money expressions during learning and prediction, leading to improved accuracy in sequence labeling.

When looking at the overall results, it was found that dividing the comments into a certain number of shorter ones was closer to the expected effect than dividing them into one. The evaluation results also showed that the "Claim" label was not predicted at all. One reason is that "Claim" label appears very infrequently in the minutes. In the original data by Poliinfo3, "Claim" label appears only 46 times while "Premise" label appears 943 times in the whole of the local council proceedings. In the training data, "Premise" label as a whole appears 151 times, while "Claim" label appears only 10 times in "Claim：意見・提案・質問". The problem is that there are few parts in the minutes themselves that can be judged as "Claim" labels, so it will be important to improve the prediction accuracy of labels that do not appear frequently. Even for the relatively predictable "Premise" label, past, future, and other mistakes can be seen, so it is important to find a way to create data that captures the context.

As for the system for linking money Expressions to IDs, it turned out that the method we tried this time did not produce the expected results. Although the idea itself was not bad, it is thought that it did not work well because the minutes that the system is processing are words uttered by humans. Since different people have different ways of speaking and different paths to conclusions, it is necessary to find a method to search for more effective topics around the money expressions in the speech.

7. CONCLUSION

In this study, we conducted an experiment to find out whether Japanese data can be labeled in a discussion labeling system for minutes, and to analyze how to create more effective data from the changes in labeling accuracy depending on the data given. In the experiment, we divided the data into two categories: one for the whole utterance and one for a sentence in the utterance, and the other for the replacement of numbers and money expressions in the data.

As a result of the experiment, we found that the accuracy was improved by using a condition in which the sentence is divided into short segment such as one sentence. In addition, since this study uses data where money expressions have been extracted beforehand, replacing the money expressions with tokens beforehand makes it easier to capture the money expressions and improves the accuracy.

Future issues include how to improve the accuracy of estimating the wrong type of "Premise" label and the "Claim" label, which originally appeared in a small number of cases, and whether the accuracy can be improved by introducing BERT for creating word embeddings than word2vec.

REFERENCES

[1]. Xuexhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceeding of the 54th Annual Meeting of the Association for Computational Linguistics* (*Volume 1:Long Papers*). ACL, Berlin, Germany, Pages 1064-1074. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF - ACL Anthology

[2]. https://github.com/XuezheMax/NeuroNLP2

[3]. Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamura, Madoka Ishioroshi, Masaharu Yoshioka, Tomoyoshi Akiba, Yasuhiro Ogawa, Minoru Sasaki, Kenichi Yokote, Kazuma Kadowaki, Tatsunori Mori, Kenji Araki, Teruko Mitamura and Satoshi Sekine, "Overview of the NTCIR-16 QA Lab-PoliInfo-3 Task", Proceedings of The 16th NTCIR Conference (2022).