selt Team's Entity Linking System at the NTCIR-15 QA Lab-PoliInfo2

Naraki Yuji
Waseda, Japan
yuji.1277@akane.waseda.jp

Sakai Tetsuya
Waseda, Japan
tetsuya@waseda.jp

ABSTRACT

The selt team participated in the entity linking task of NTCIR-15 QA Lab-PoliInfo 2. This paper describes our entity linking system for assembly member speeches using BERT and wikipedia2vec. Using this system, we can effectively preprocess and postprocess data for mention detection, and (after fixing the format of our run file) achieved the second-best performance in the leaderboard of NTCIR-15. For mention detection specifically, we achieved the best score.

TEAM NAME

selt

SUBTASKS

Entity Linking

1 INTRODUCTION

NTCIR-15 Question Answering Lab for Political Information 2 [1] (QA Lab-PoliInfo 2) deals with political information and sets forth three tasks: stance classification, dialog summarization, and entity linking. Our team participated in the entity linking task. Entity linking involves the extraction of eigenexpressions in natural language sentences and mapping of these expressions to a knowledge base such as Wikipedia and DBpedia. For example, given a sentence stating that "Paris is the capital of France," the entities of "Paris" and "France" are mapped to a knowledge base corresponding to each word. The objective of the entity linking task at NTCIR-15 was to clarify the official name of a law or bill stated through an abbreviation. Therefore, words that describe a law or bill are extracted, if the law exists in Wikipedia, the link to the law is mapped to the words.

This report describes our entity linking system and discusses not only the official results, but also some comparison results.

2 SYSTEM

We submitted four runs, the IDs of which are 173, 178, 179, and 213, and in this section, we describe the best performing system used for run 178. Our entity linking system consists of two modules: a mention detection module using BERT and a entity disambiguation module using wikipedia2vec. In Section 2.1, we describe the method of building the training data for BERT. Sections 2.2 and 2.3 describe the architectures of the mention detection and entity disambiguation modules, respectively.

2.1 Training Data for BERT

In this task, the target sentences are speeches of an assembly members. The training data have IOB2 tags and links to the laws. The preprocessing method is used to convert some special words that make little or no sense into special tokens. The correspondence between the words and special tokens is presented in Table 1.

Table 1: Correspondence between special word sequence and special tokens

Special Word Sequence	Special Token
NaN	[NULL]
"　" (Full-width space)	[SPACE]
"○" (Circle)	[SEP]
"-" (Hyphen)	[BAR]
"　　　-------------" (3 Full-width space + 13 Hyphen)	[LBAR]

2.2 Mention Detection Module

The mention detection module extracts mentions that are word sequences describing a law or bill. We did extraction via text segmentation using a pre-trained BERT. We use the IOB2 tag for the labels for prediction. We use a pre-trained BERT from huggingface [2] and fine-tune it with 80% training data for 10 epochs. We then evaluate the performance of the fine-tuned model using the remainder of the training data.

Table 2: Modification rule of IOB2 tag, described using regular expression

Before Modification	After Modification
BB	BI
OI(I)+	OB(I)+
OBO	OOO
OIO	OOO

Table 3: Modification rule for the end of the mentions

Ends of Mentions
法
法案
法制
に関する法律
に関する法律の一部を改正する法律案
の一部を改正する法律案
に関する法律案
改正案
法律

For the test data, we predict the IOB2 tag using fine-tuned BERT and modify the output with three modification rules. The first rule is to convert the output into the rules of the IOB2 tag. This rule is presented in Table 2 through regular expressions. The second rule is based on the analysis that most law names do not begin with a hiragana or special token. We remove mentions that begin with a hiragana and special token. Although there are some laws that begin with a hiragana such as "あへん法", we did not consider them. The third rule is concerned with the end of a mention. An analysis of the training data and official laws and bills shows that mentions end with specific word sequences, as presented in Table 3. These three rules process the output of the model and determine the mentions.

2.3 Entity Disambiguation Module

This module constitutes the last part of our system. The entity disambiguation module determines the official name from the detected mentions. We used wikipedia2vec [3] to apply this module. Wikipedia2vec is a tool used for obtaining the embeddings of words and entities from Wikipedia. Wikipedia2vec can also determine entities from mentions using their embeddings. We used as input not only mentions detected by the mention detection module but also their transformation. The transformation is modifying the ends of the mentions. If a detected mention ends with "法" and its entity is unknown, we replace "法" with "法案." If a detected mention ends with "法案" and its entity is unknown, we replace the mention "法案" with "法."

3 RESULTS OF FORMAL RUN

In this section, we show the description and scores of our formal runs. However, our submission format differed from the specified format and results in a significant lowering of the disambiguation score. We discuss the difference in the format and the justification for the revision in Section 3.1. We then show the official scores and the revised disambiguation scores in Section 3.2.

3.1 Submission Format

We submitted our run results in a different format and this substantially lowered our official scores. In the specified format, if the system detects mentions and fails to disambiguate, "NIL" should be output in the columns representing the entities. However, we left that part blank and the scoring system could not calculate our disambiguation scores correctly. We replaced the improper blanks with "NIL" and recalculated the scores in the same way as described in the over-view paper [1]. Note that this only modifies the run file format, not our algorithm. Hereafter, we discuss the performance of our system based on the corrected run. We apologize to the task organizers for our mistake.

3.2 Description of Results

In this section, we show the results of our formal runs. Table 4 shows the description of our official formal runs. Table 5 shows a formal, unmodified F1 score on the entity disambiguation. Table 6 shows the F1 score of the IOB2 tag and mention detection and the revised F1 score of disambiguation. We present additional explanations to the description of our formal run. Run 173 is our first run and used the method described in Section 2 without the rule on the beginning of the mentions. For BERT, the training applies 3 epochs in runs 173 and 213, 10 epochs in run 178, and 20 epochs in run 179. In run 213, we add an entity disambiguation rule based on perfect matching between detected mentions and the given Wikipedia data.

Table 4: Description of our formal run

ID	Description
173	MD: BERT, ED: wikipedia2vec
178	change of the epochs of BERT
179	another change of the epochs of BERT
213	additional dictionary based method in ED

Table 5: Official disambiguation F1 score of our formal run

ID	Disambiguation F1
173	0.297980
178	0.297767
179	0.297767
213	0.292929

Table 6: F1 score of IOB2 tag and mention detection and revised F1 score of disambiguation of our formal run

ID	IOB2 tag	Mention Detection	Disambiguation
173	0.901987	0.808081	0.505051
178	0.932150	0.903226	0.526055
179	0.932150	0.903226	0.526055
213	0.900662	0.803030	0.454545

4 DISCUSSION

As Table 6 presents, run 178 is the result of our best and most efficient system. Regarding the number of epochs, we found that 3 training epochs is insufficient and 20 is too many. We therefore conclude that training for 10 epochs is the most optimal for our model for convergence to a desirable solution. Our system achieved the highest score on the leaderboard on IOB2 tag F1 and mention detection F1. To achieve a higher score than this, we believe that the number of assembly member speeches should be increased or an architecture of the model specializing in detecting legal names should be devised. Because our model showed desirable performance in terms of mention detection, the disambiguation score is also naturally high. Notably, our model scored the second highest.

We will now discuss our system based on the failures of entity disambiguation that was performed. The test data contained 209 mentions and our best system was able to predict 106 mentions entirely correctly, out of which 25 were incorrectly predicted in terms of mention detection and 78 in terms of disambiguation. We divided the error prediction types in terms of disambiguation into three categories. Type 1 is when the correct entity is not "NIL" and the predicted entity differs from the correct entity. Type 2 is when there is no correct entity and "NIL" should have been output, but an entity was predicted and "NIL" was not received as the output. Type 3 is when the correct entity is not "NIL", but no entity can be predicted. Here, "武力紛争法" was the only mention of type 1. The mentions of type 2 are "カジノ法案" and "ＩＲ法案." The laws or bills of type 1 and 2 can be considered as exceptions that were difficult to address owing to the small number of laws or bills in the test data. Regarding type 3, we believe that, given the words appearing in the test data, it can be further divided into two types, 3A and 3B. Type 3A occurs when the formal name of the correct entity appears in the test data and type 3B when it does not. Disambiguating the mentions of type 3B requires background knowledge of the law, which can be quite difficult to achieve. With regard to type 3A, however, it is possible to connect entities without being able to disambiguate them from the mentions if it is known that they represent the same meaning as the place where the formal names appear. We believe that the method using co-occurrence frequency can help solve this type of error prediction.

5 CONCLUSIONS

In our system, we mainly used BERT and wikipedia2vec with some rules to improve the accuracy. After fixing the format of our run file, we achieved the second-best score on the leaderboard of NTCIR-15. In particular, for mention detection, our system achieved the best performance. Our mention detection system has the potential to show further improvement if further data are used for training. Moreover, the legal background knowledge and co-occurrence frequency is useful for improving the performance of the disambiguation.

REFERENCES

[1] 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).

[2] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* abs/1910.03771 (2019).

[3] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia. *arXiv preprint 1812.06280v3* (2020).