knlab Team: NTCIR-15 QA Lab-PoliInfo-2 Stance Classification Task

Kazuhiro Atsumi
Shizuoka University
katsumi@kanolab.net

Yoshinobu Kano
Shizuoka University
kano@inf.shizuoka.ac.jp

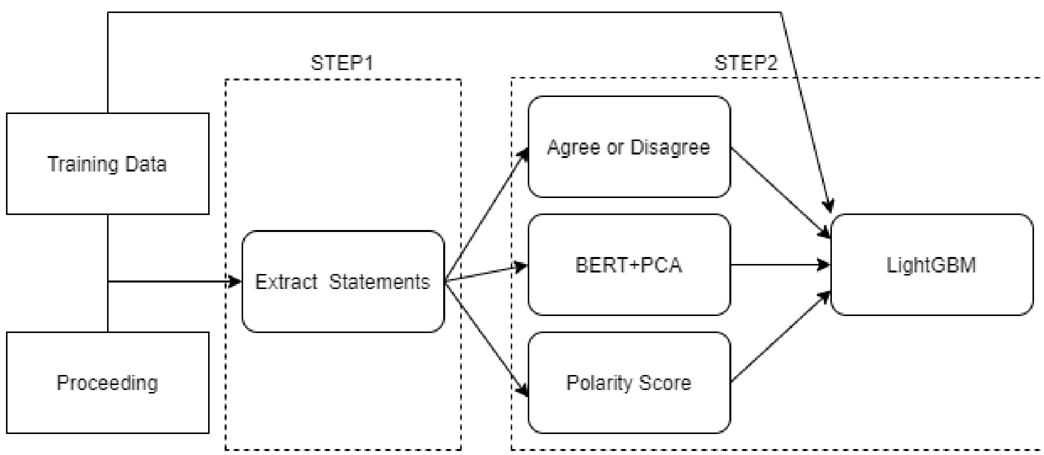

Figure 1. The overall architecture

ABSTRACT

This paper reports the knlab team's approach and results in the NTCIR-15 QA Lab-PoliInfo-2's Stance Classification Task. This task predicts stances (Agreement, Disagreement, No Mention) for each party regarding each proposal, using minutes of proceedings which includes statements of politicians. Our team designed features obtained from a sentiment dictionary and BERT, then trained LightGBM to classify the stances.

KEYWORDS

PoliInfo, Politics, Stance Classification, Machine Learning

Team Name

knlab

Subtasks

Stance Classification Task

1 Introduction

Our team participated in the NTCIR-15 QA Lab-PoliInfo-2's Stance Classification Task [1], which aims to present relevant information when discussing political problems. The Stance Classification Task aims to infer a politician's position from his or her statements, referring to the minutes of the Japanese Tokyo metropolitan parliamentary meeting in which the politician's statements are included. Given a bill, task participants are required to classify each party into one of three stances (Agreement, Disagreement, No Mention). The classification results are evaluated in two ways, either Agreement or Disagreement, and mentioned (Agreement or Disagreement) or No Mention.

Our approach is a two-step process: in the first step, we use a rules-based approach to extract the party's statements referring to the bill from the minutes. In the second step, we use machine learning methods to categorize the party's stance on the bill. Section 2 describes the proposed methods, Section 3 describes the evaluation results, and Section 4 concludes this paper.

Rule 1	If a bill number is included in a given sentence, return true.
Rule 2	If one or more patterns of “all (全て)”, “all (すべて)”, and "other (他)" is included, and if one or more patterns of "agree (賛成)" or "disagree (反対)" is included, then return true.

Table 1: List of rules which return whether a given sentence refer to a bill or not

Feature 1	Its value is “1” when there is a "agree (賛成)" immediately after the bill number, “2” when there is a "disagree (反対)", and “0” when there is no such string occurs.
Feature 2	Final layer of BERT output, dimensionally compressed by Principal component analysis (PCA).
Feature 3	Polarity scores using a Japanese Sentiment Polarity Dictionary

Table 2: List of our features

	Validation	Test
Without Feature 1-3	0.892	0.942
Without Feature 1	0.901	0.947
Without Feature 2	0.911	0.952
Without Feature 3	0.906	0.951
All Feature	0.913	0.953

Table 3: Experiment Results

2 Stance Classification Methods

Our team used a machine learning method rather than a rule-based method in order for the system to be generic. Our team performed this task in two steps. The overall architecture of the proposed method is shown in Figure 1. The first step extracts statements which refer to the bill from the minutes. The second step classifies a stance from the extracted statements. We used a morphological analyzer Mecab[1] with mecab-ipadic-NEologd[2] as its dictionary.

2.1 Extracting target's statements

The goal of this step is to obtain as many statements as possible, because we need more information in our next main step which extracts feature. This first step extracts all statements of all members of the Congress from the start date until the end date of the corresponding meeting of the bill from the proceeding. Then we extract the statements that refer to the bill by our rule-based method in a sentence-by-sentence basis after morphological analysis. Table 1 shows our rules which returns whether to refer to a bill or not as follows. Rule 1 corresponds to a sentence which explicitly refers to a bill number. If Rule 1 returns true, and if there is no other bill number in the given sentence, we concatenate following sentences until another bill number occurs. Rule 2 corresponds to a sentence that expresses a position, although it does not refer to any specific bill within that minutes. After applying Rule 1 and Rule 2, we link the extracted statements with the training data, which only include metadata of the proceedings, for the speaker's party. Finally, we concatenate extracted statements of the same party into a single document for each referred bill.

2.2 Classfication by LightGBM

Our stance classification is performed by LightGBM [2], which is a machine learning method based on the decision tree. Note that this is a binary classification of Agreement or Disagreement.

We extract three features from the training data and the portions of the senator's statements created in the previous step. These features are shown in Table 2.

Feature 1 indicates whether there is an affirmative or negative utterance immediately after the bill number as a categorical feature. This feature indicates whether the legislator is Agreement or Disagreement the target proposal.

Feature 2 uses BERT [3] with a pre-trained Japanese model, compressing 512-dimensional intermediate vectors in its final layer into 5 dimensions by principal component analysis (PCA) . Feature 2 represents the linguistic features of a given sentence.

Feature 3 uses a Japanese Sentiment Polarity Dictionary[3] to represent the scores of the polarity values within an utterance. The score of Feature 3 is calculated as follows. Firstly, we calculate a sum of the polarity values for words in a given sentence that match

[1] https://taku910.github.io/mecab/

[2] https://github.com/neologd/mecab-ipadic-neologd

[3] http://www.cl.ecei.tohoku.ac.jp/index.php?Open%20Resources%2FJapanese%20Sentiment%20Polarity%20Dictionary

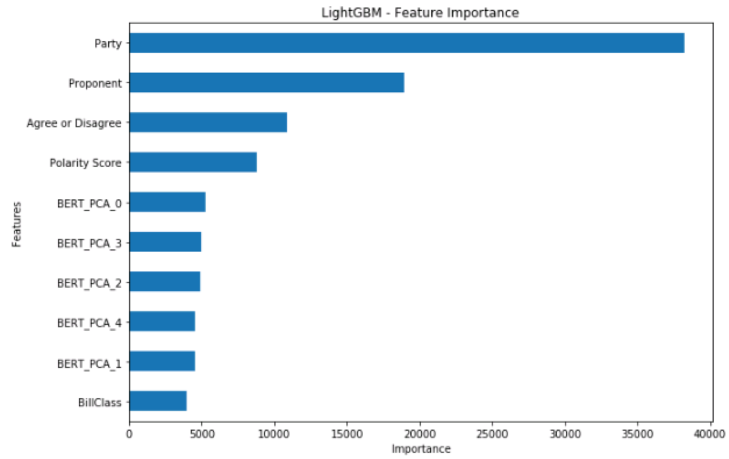

Figure 2 Feature Importance

the Japanese Sentiment Polarity Dictionary. Then we divide this sum by the number of words that match the Japanese Sentiment Polarity Dictionary words. We designed this feature to extract negative statements which do not explicitly utter "disagree (反対)".

In addition to our original features above, we use the Proponent, BillClass, and Party features as categorical features of the training data, when training the LightGBM. In the case of binary classification (Agreement or Disagreement), we use the LightGBM output directly. In the case of the trinary classification (Agreement, Disagreement, and No Mention), we regard a given sentence as No Mention when none of our rules were matched. When any of the rules matched, we used LightGBM to perform the binary classification.

3 Experiment

3.1 Setting

Since the training data is imbalanced, including 21,109 Agreements and 2,212 Disagreements over all of the parties in the proceedings, we retained this fraction when performing a 5-fold cross validation. We used the default parameters of LightGBM when training, employed an early stopping of the model training (50 counts) using the validation data accuracy of the cross-fold validation.

In order to examine the contribution of our features, we also performed a ablation study by removing some of the features.

3.2 Results

The results are shown in Table 3, which shows the accuracy scores of our proposed model (all features) and ablation models. We observed that each feature contributed to the performance in both validation (cross-fold validation) and test scores. We analyzed the importance of each feature using the cross-validation results (Figure 2), where “Agree or Disagree” corresponds to Feature 1, BERT_PCA_0 to BERT_PCA_4 corresponds to Feature 2, and Polarity Score corresponds to Feature 3. Among all features, Party showed the largest contribution. Among our original features, Feature 3 (Polarity Score) showed the largest contribution. These results suggest that the Polarity Score works.

We also performed manual error analysis using the cross-fold validation results. We found a couple of samples in which our first step extracted non-relevant statements. This is because our first step tries to extract more statements to obtain more information, remaining non-relevant statements as well. Excluding such non-relevant statements is our future work.

4 Conclusion

We proposed a machine learning based method using LightGBM for the Stance Classification Task of NTCIR-15. We designed our features which includes linguistic information using BERT, and a polarity score using the Japanese Sentiment Polarity Dictionary. The experimental result showed our machine learning method and our features were effective.

REFERENCES

[1] Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takmaru, Madoka Ishioroshi, Teruko Mitamura, Masaharu Yoshioka, Tomoyoshi Akiba, Yasuhiro Ogawa, Minoru Sasaki, Kenichi Yokote, Tatsunori Mori, Kenji Araki, Satoshi Sekine and Noriko Kando, 2020. Overview of the NTCIR-15 QA Lab-PoliInfo−2 Task. In Proceedings of The 15th NTCIR Conference.
[2] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu, 2017. A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems.
[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805