Label Embedding using Hierarchical Structure of Labels for Twitter Classification

Taro Miyazaki  Kiminobu Makino  Yuka Takei  Hiroki Okamoto  Jun Goto
NHK Science and Technology Research Laboratories
{miyazaki.t-jw, makino.k-gg, takei.y-ek, okamoto.h-iw, goto.j-fw}@nhk.or.jp

Abstract

Twitter is used for various applications such as disaster monitoring and news material gathering. In these applications, each Tweet is classified into pre-defined classes. These classes have a semantic relationship with each other and can be classified into a hierarchical structure, which is regarded as important information. Label texts of pre-defined classes themselves also include important clues for classification. Therefore, we propose a method that can consider the hierarchical structure of labels and label texts themselves. We conducted evaluation over the Text REtrieval Conference (TREC) 2018 Incident Streams (IS) track dataset, and we found that our method outperformed the methods of the conference participants.

1 Introduction

Twitter is used for various applications such as disaster monitoring (Ashktorab et al., 2014; Mizuno et al., 2016) and news material gathering (Vosecky et al., 2013; Hayashi et al., 2015). In these applications, each Tweet is classified into pre-defined classes. These classes have a semantic relationship with each other and can be classified into a hierarchical structure. Consider a news material gathering system that has classes such as “Tornado,” “Flood,” and “Riot.” These can be classified into two upper classes, “Natural disaster” (“Tornado” and “Flood” as lower classes) and “Incident” (“Riot” as a lower class). These relationships can be an important clue for classification. Needless to say, label texts of pre-defined classes themselves include important clues. People can understand what the criterion of the classification is by reading labels. Therefore, we propose a method that can consider the hierarchical structure of labels and the labels themselves.

Our method is based on the Label Embedding (LE) method (Zhang et al., 2018). The typical LE method embeds input text and label text, and then the embedded vectors are fed into “Matcher,” which outputs the score between the input text and label. We use a two-step attention mechanism for LE to consider the hierarchical structure of labels. We confirm the effectiveness of our method through evaluation over the Text REtrieval Conference (TREC) 2018 Incident Streams (IS) track dataset.

Our contributions are as follows: (1) we propose label embedding using the hierarchical structure of labels, and (2) our method outperformed others on the TREC 2018 IS track dataset for several metrics, including the Any-type Micro F1 score, which is the target metric of the track.

2 TREC 2018 IS track

The TREC 2018 IS track is a shared-task that aims to mature social media-based emergency response technology (McCreadie et al., 2019). The task of the track is classifying Tweets by information type, which consists of 24 classes. The list of information types and the description for each information type are given as ontology as shown in Table 1. The dataset contains approximately 1,300 Tweets for training and more than 20,000 for testing. The numbers of samples for each class are also given in the table. As shown in the table, a training dataset does not include much data and is unbalanced for classes.

The track uses two evaluation methods — Multi-type and Any-type — as follows.

Multi-type: Calculating categorization performance per information type in a 1-vs-All manner.

Note that, each sample has just one information type for training data, but more than one for testing data.

Table 1: Ontology information type.

<table>
<thead>
<tr>
<th>Intent type</th>
<th>Information type</th>
<th>Description</th>
<th># (train)</th>
<th># (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUEST</td>
<td>REQUEST-GOODS-SERVICES</td>
<td>The user is asking for a particular service or physical good</td>
<td>0</td>
<td>256</td>
</tr>
<tr>
<td>REQUEST</td>
<td>REQUEST-SEARCH-AND-RESCUE</td>
<td>The user is requesting a rescue (for themselves or others)</td>
<td>0</td>
<td>266</td>
</tr>
<tr>
<td>REQUEST</td>
<td>REQUEST-INFORMATION-WANTED</td>
<td>The user is requesting information</td>
<td>10</td>
<td>172</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-WEATHER</td>
<td>The user is providing a weather report (current or forecast)</td>
<td>41</td>
<td>1,125</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-FOCUS-OF-EVENT</td>
<td>The user is giving an eye-witness account</td>
<td>28</td>
<td>3,807</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-THIRD-PARTY-OBSERVATION</td>
<td>The user is reporting a information that they received from someone else</td>
<td>15</td>
<td>4,150</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-EMERGING-THREATS</td>
<td>The user is reporting a potential threat that may cause future loss of life or damage</td>
<td>36</td>
<td>666</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-SIGNIFICANT-EVENT-CHANGE</td>
<td>The user is reporting a new occurrence that public safety officers need to respond to</td>
<td>34</td>
<td>415</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-MULTIMEDIA-SHARE</td>
<td>The user is sharing images or video</td>
<td>127</td>
<td>3,974</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-SERVICE AVAILABLE</td>
<td>The user is reporting that they or someone else is providing a service</td>
<td>15</td>
<td>1,076</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-FACTOID</td>
<td>The user is relaying some facts, typically numerical</td>
<td>140</td>
<td>2,383</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-OFFICIAL</td>
<td>An official report by a government or public safety representative</td>
<td>52</td>
<td>403</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-CLEAN UP</td>
<td>A report of the clean up after the event</td>
<td>5</td>
<td>62</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT-HASHTAGS</td>
<td>Reporting which hashtags correspond to each event</td>
<td>4</td>
<td>3,363</td>
</tr>
<tr>
<td>CALL-TO-ACTION</td>
<td>CALL-TO-ACTION-ADVICE</td>
<td>The user is providing advice to the public</td>
<td>122</td>
<td>1,118</td>
</tr>
<tr>
<td>CALL-TO-ACTION</td>
<td>CALL-TO-ACTION-DONATIONS</td>
<td>The user is asking people to donate goods/money</td>
<td>15</td>
<td>804</td>
</tr>
<tr>
<td>CALL-TO-ACTION</td>
<td>CALL-TO-ACTION-MOVE-PEOPLE</td>
<td>The user is asking people to leave an area or go to another area</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>OTHER</td>
<td>OTHER-CONTINUOUS-NEWS</td>
<td>The post providing/balancing to continuous coverage of the event</td>
<td>250</td>
<td>4,871</td>
</tr>
<tr>
<td>OTHER</td>
<td>OTHER-ADVICE</td>
<td>The author is providing some advice to the public</td>
<td>39</td>
<td>1,209</td>
</tr>
<tr>
<td>OTHER</td>
<td>OTHER-SENTIMENT</td>
<td>The post is expressing some sentiment about the event</td>
<td>39</td>
<td>6,952</td>
</tr>
<tr>
<td>OTHER</td>
<td>OTHER-DISCUSSION</td>
<td>Users are discussing the event</td>
<td>51</td>
<td>2,060</td>
</tr>
<tr>
<td>OTHER</td>
<td>OTHER-IRELEVANT</td>
<td>The post is irrelevant, contains no information</td>
<td>163</td>
<td>2,605</td>
</tr>
<tr>
<td>OTHER</td>
<td>OTHER-UNKNOWN</td>
<td>Does not fit into any other category</td>
<td>28</td>
<td>77</td>
</tr>
<tr>
<td>OTHER</td>
<td>OTHER-KNOW-NOT</td>
<td>The responder already knows this information</td>
<td>112</td>
<td>1,101</td>
</tr>
</tbody>
</table>

Figure 1: Overview of our proposed method.

Methods are evaluated using four metrics: Precision, Recall, F1 (positive class), and Accuracy (overall).

Any-type: A system receives a full score for a Tweet if the system assigned any of the categories that the human assessor selected for that Tweet. Methods are evaluated using four metrics: Precision, Recall, F1 (micro average: the target metric of the track), and Accuracy (micro average).

3 Our Method

As shown in Table 1, each information type has a rather rich description, so we think that using the description can improve the performance. Also, the information types can be regarded as a hierarchical-structure label, which consists of four upper classes (equal to intent type, such as REQUEST and REPORT) and 24 lower classes (equal to information type, such as REQUEST-GOODS-SERVICES and REPORT-WEATHER). To use these useful features, we propose a label embedding using the hierarchical structure of labels.

The overview of our method is given in Figure 1. Our method is similar to the one Zhang et al. (2018) used, but it differs in that our method considers the hierarchical structure of labels, which is our key feature.

X, Lj, and Uj are the input text, label text of the j-th lower class, and label text of the upper class corresponding to the j-th lower class, respectively. We use “description” and “intent type” in Table 1 for the label text of the lower and upper classes, respectively. “Embed” in the figure is a pre-trained model such as BERT (Devlin et al., 2019) and ELMo (Peters et al., 2018) of V × m, where V and m denote the vocabulary size and embedding size, respectively. wX and wL means Bi-directional Long Short-Term Memory (Bi-LSTM), and both have a d dimension of hidden layer.

First, X, Lj, and Uj are embedded using the pre-trained model, and then the embedded vectors are fed into wX for X and wL for Lj and Uj, respectively. We gather the hidden states of Bi-LSTM for each input token and stack them, and then we get three matrices:

Text representing matrix: hX ∈ R^[|X|] × 2d ,
Lower label representing matrix: hLj ∈ R^[|Lj|] × 2d ,
Upper label representing matrix: hUj ∈ R^[|Uj|] × 2d ,

where |X|, |Lj| and |Uj| represent the number of tokens in X, Lj and Uj, respectively. Note that we have a 2d dimension for the row of each matrix because of concatenating forward and backward states.

Next, we calculate “label attention” and “hierarchical attention.” The label attention weight w_label ∈ R^[|Lj|] and label attention vector a_label ∈ R^[|X|] × 2d .

6319
are calculated as follows:
\[
    w_{\text{label}} = \text{softmax} \left( h \cdot h_j^T / \sqrt{2d} \right),
\]
\[
    a_{\text{label}} = w_{\text{label}} \cdot h_j,
\]
where \( h \in \mathbb{R}^{2d} \) is the concatenation of hidden states of the final step of both directions of \( h_{u_j} \).
We use scaled-dot product attention for label attention (Vaswani et al., 2017).

By using \( a_{\text{label}} \), the hierarchical attention weight \( w_{\text{hier}} \in \mathbb{R}^{|X|} \) and hierarchical attention vector \( a_{\text{hier}} \in \mathbb{R}^{2d} \) are obtained as follows:
\[
    w_{\text{hier}} = a_{\text{label}} \cdot h_j^T,
\]
\[
    a_{\text{hier}} = w_{\text{hier}} \cdot h_j.
\]

Note that we use normalization for label attention but not for hierarchical attention. We confirm that this condition is best by an experiment, which is detailed in the Discussion section.

Then, \( a_{\text{hier}} \) is fed to “Matcher,” which consists of a two-layer Feed-Forward Neural Network (FFNN):
\[
    v_{\text{mid}} = W_{\text{mid}} \cdot \text{ReLU} (a_{\text{hier}}) + b_{\text{mid}},
\]
\[
    s_j = W_{\text{matcher}} \cdot \text{ReLU} (v_{\text{mid}}) + b_{\text{matcher}},
\]
where \( W_{\text{mid}} \in \mathbb{R}^{d \times 2d} \) and \( W_{\text{matcher}} \in \mathbb{R}^{1 \times d} \) are weight matrices, and \( b_{\text{mid}} \in \mathbb{R}^d \) and \( b_{\text{matcher}} \in \mathbb{R}^1 \) are bias. \( s_j \) is a score between input Tweet \( X \) and information type \( j \).

We gather \( s_j \) for all \( j \in C \), where \( C \) denotes the set of the information type. Finally, we obtain the estimation result \( o \) for input Tweet \( X \) as follows:
\[
    o = \arg\max (s_1, s_2, \ldots, s_{|C|}).
\]

4 Experiments

4.1 Experimental Settings

Our experiments were based on TREC 2018 IS track in terms of the dataset and evaluation. The dataset consists of original json data of Tweets. We excluded @-mentions and URLs. We conducted 10-fold cross validation to find the best setting, and all models were used as ensemble models to predict the information type for testing data.

The models were implemented in Chainer (Tokui et al., 2015) and learned with the Adam optimizer (Kingma and Ba, 2014), on the basis of categorical cross-entropy loss with class weights \( W_c = \frac{|c_{\text{max}}|}{|c|} \), where \( |c| \) is the number of samples of information type \( c \) appearing in the training data, and \( |c_{\text{max}}| \) is that of the most-frequent information type. We used BERT-base, Uncased\(^3\) as the pre-trained model, which has a 12-layer, 768-hidden states transformer model with 12-heads attention. The BERT model was used as a frozen model without fine-tuning.

The hyperparameters used were: a minibatch size of 32; hidden layer size of 200; \( L_2 \) regularization coefficient of \( 10^{-5} \); dropout rate of 0.5; and 100 training iterations, with early stopping on the basis of the Macro F1 score of the development data.

4.2 Baseline Methods

We prepared three baseline methods. All baseline methods used the same hyperparameters as the proposed method.

Non-LE: This method is not based on LE. We use the BERT model as token embedding. The embedded vectors are fed into Bi-LSTM, and then the concatenation of hidden states of the final step of both directions is fed into the two-layer FFNN for classification into the information type.

Non-hier: This method is almost the same as our proposed method but does not use hierarchical attention. The vector fed into Matcher is calculated as attention between embedding vectors of the input text and label text of the lower class (Figure 2-(a)).

Transfer: This method uses the same structure as Non-hier, but we use transfer learning to consider the hierarchical structure of labels. This is inspired by Shimura et al. (2018). First, the model is learned as classifying upper class labels, and then the model is transferred to be learned as classifying lower class labels (Figure 2-(b)). We tried several settings and found that transferring only the Bi-LSTM layer is better in this task.

\(^3\)https://github.com/google-research/bert
Table 2: Experimental results. *italics* denote the best results in TREC 2018, and **bold** denotes the overall best results. For some metrics, the best results are for other TREC 2018 participants’ methods not shown in the table, so no results are in *italics* or **bold** for these metrics.

<table>
<thead>
<tr>
<th>Method</th>
<th>Multi-type, Macro Avg.</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Accuracy</td>
<td>Precision</td>
<td>Recall</td>
<td>F1 (target metric)</td>
<td>Accuracy</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Accuracy</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Non-LE</td>
<td>0.2461</td>
<td>0.1302</td>
<td>0.1523</td>
<td>0.9073</td>
<td>0.5032</td>
<td>0.7055</td>
<td>0.5874</td>
<td>0.4387</td>
<td>0.5206</td>
<td>0.6301</td>
<td>0.5701</td>
<td>0.4282</td>
<td>0.5097</td>
<td>0.6810</td>
</tr>
<tr>
<td>Non-hier</td>
<td>0.2522</td>
<td>0.1300</td>
<td>0.1497</td>
<td>0.9065</td>
<td>0.5206</td>
<td>0.6301</td>
<td>0.5701</td>
<td>0.4282</td>
<td>0.5097</td>
<td>0.6810</td>
<td>0.5830</td>
<td>0.4361</td>
<td>0.5097</td>
<td>0.6810</td>
</tr>
<tr>
<td>Transfer</td>
<td>0.2448</td>
<td>0.1324</td>
<td>0.1533</td>
<td>0.9071</td>
<td>0.5097</td>
<td>0.6810</td>
<td>0.5830</td>
<td>0.4361</td>
<td>0.5097</td>
<td>0.6810</td>
<td>0.5830</td>
<td>0.4361</td>
<td>0.5097</td>
<td>0.6810</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.2464</td>
<td>0.1300</td>
<td>0.1521</td>
<td><strong>0.9079</strong></td>
<td>0.5164</td>
<td>0.7018</td>
<td><strong>0.5950</strong></td>
<td><strong>0.4460</strong></td>
<td>0.5164</td>
<td>0.7018</td>
<td><strong>0.5950</strong></td>
<td><strong>0.4460</strong></td>
<td>0.5164</td>
<td>0.7018</td>
</tr>
</tbody>
</table>

Table 3: Comparing normalization methods.

<table>
<thead>
<tr>
<th>label attention</th>
<th>hierarchical attention</th>
<th>Macro F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>normalized</td>
<td>normalized</td>
<td>0.2676</td>
</tr>
<tr>
<td>normalized</td>
<td>none</td>
<td><strong>0.3539</strong></td>
</tr>
<tr>
<td>none</td>
<td>normalized</td>
<td>0.0303</td>
</tr>
<tr>
<td>none</td>
<td>none</td>
<td>0.0329</td>
</tr>
</tbody>
</table>

4.3 Results

Table 2 presents the results for our methods and part of the TREC 2018 participants’ results. At TREC 2018, cbunS2, KDEIS4_DM, and umdhcilfasttext had the top three results for the Any-type Micro F1 score, which is the target metric of the track. DLR_Augmented had the best Multi-type Macro F1 score.

Our proposed method outperformed others in terms of the Any-type Micro F1 score and both types of accuracy, while the best for the Multi-type F1 score is DLR_Augmented.

4.4 Discussion

Comparison with other methods

Our proposed method performs the best for several metrics including the target metric (Any-type Micro F1 score). Our hierarchical attention can give less weight for verbose phrases such as “The user is asking” in label text. These phrases include only a little information useful for classifying, so giving them less weight is reasonable and effective. This is one of the advantages of our method, which distinguishes it from LE-based baseline methods. Our method outperforms not only LE-based methods, but also the simple but strong baseline (Non-LE) and the TREC 2018 participants’ methods. Therefore, we confirmed the effectiveness of our proposed method.

On the other hand, our method did not perform the best for the Multi-type Macro F1 score. DLR_Augmented, which performed the best for this metric, uses augmented data. This is effective in the learning process especially for classes that have small training data. Our method does not use augmented data, so it has lower accuracy for these classes than DLR_Augmented. Comparing our proposed method with the baseline methods, differences in the Multi-type Macro F1 score were small, and the proposed method achieved the best score for Multi-type accuracy. We think that the differences come from labels that have a small number of samples in test set because the F1 scores for these classes are sensitive to the output, which greatly affects the Macro F1 score.

Insight of our proposed method

To determine which combination of normalization is effective for our task, we conducted a small experiment using development data. Results are shown in Table 3. We used the Macro F1 score as the metric. We can see that using normalization for label attention but not for hierarchical attention is best. One reason is that the norm of the output vector of hierarchical attention directly affects the score $s_j$. Hierarchical attention without normalization can make distinctions for the output vector of each class in the norm, so it works well. On the other hand, using normalization is better in label attention. We observed that not using normalization for label attention makes the model diverge, so normalization needs to be used for label attention.

Table 4 shows the macro F1 scores summarized for each upper class of the proposed method and Non-hier. $\Delta$ in the table means the difference between two methods. Interestingly, our method works best for the upper class OTHER, which contains the fewest meanings in the upper class label

---

4 Each sample has just one information type for development data and more than one for testing data, so the Macro F1 score between Table 2 and 3 cannot be compared directly.
text. The lower class label text for the upper class OTHER includes less verbose phrases such as “The user requesting,” so label attention can give higher weight for important parts of the label text more precisely.

On the other hand, the F1 score of the proposed method for the upper class REPORT is worse than that of Non-hier. This is caused by the lower class REPORT-SIGNIFICANTEVENTCHANGE (the $\Delta$ is −0.05, which is the worst in all lower classes). The label text of the lower class includes a misspelling (“occurence” instead of “occurrence”$^5$). This confuses our model when calculating label attention weight, so the F1 score is lower.

We found that the number of tokens in the lower class label text affects the effectiveness of the proposed method. The proposed method has a better F1 score than Non-hier for nine lower classes with a mean number of tokens in the lower class labels of 11.2. On the other hand, the proposed method has a lower F1 score than Non-hier for seven lower classes with a mean number of tokens of 8.0. This shows that the more tokens the lower class label text has, the more effective the proposed method becomes. Of course, this does not mean that our method works for only classes that have rich label text. For example, our method improves the F1 score for some classes that have short label text such as OTHER-DISCUSSION, which has only five tokens in the label text.

Overall, using hierarchical structure of labels is effective in many cases, but it is sensitive to the quality and quantity of label texts of the lower classes.

5 Related Work

Label embedding has attracted attention, especially for few- and zero-shot learning tasks (Socher et al., 2013; Akata et al., 2013; Kodirov et al., 2017). Now, many methods applied to natural language processing are proposed. Zhang et al. (2018) used multi-task learning, and Xia et al. (2018) used capsule networks for LE and obtained good results. Our method differs in that it considers the hierarchical structure of labels.

There are many studies using a pre-defined hierarchical structure of labels (Wu et al., 2014; Li et al., 2015; Bilal et al., 2018), and some methods are used with LE (Bengio et al., 2010; Ren et al., 2016). These approaches are effective but large-scale hierarchical data from a thesaurus need to be prepared. Our method does not need large-scale data, so it is advantageous when it comes to calculating cost and flexibility. Shimura et al. (2018) used transfer learning to consider the hierarchical structure of labels, which is the basis of our baseline method (Transfer).

6 Conclusion and Future Work

In this paper, we proposed a method of label embedding using the hierarchical structure of labels and confirmed its effectiveness through evaluation over the Text REtrieval Conference (TREC) 2018 Incident Streams (IS) track dataset. Our method outperformed other methods and obtained the best results on the dataset for several metrics including the Any-type Micro F1 score, which was the target metric of TREC 2018 IS track.

We used rather long sentences for lower label text. Typical applications use shorter labels such as “Tornado” and “Riot.” Confirming whether our method works well for other datasets including ones that have only shorter label texts is left for our future work.

Acknowledgements

We would like to thank Prof. Timothy Baldwin and Dr. Ichiro Yamada for valuable discussions, and the TREC 2018 IS track organizers for providing the dataset. We also thank the anonymous reviewers for their helpful comments.

References


Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining twitter to


Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1369–1378.


